1. Solve:

The Matlab code is shown as follows:

```
yt;                          % A signal of length N
Fs=88200;                    % The sampling frequency in Hz
sound(ypop,Fspop)
sound(ycou,Fscountry)
sound(yroc,Fsrock)
yout_pop=demod(yt,Fs,10000,10000);
yout_cou=demod(yt,Fs,20000,10000);
yout_roc=demod(yt,Fs,30000,10000);
function yout=demod(yt,Fs,Fc,B)
N=length(yt);                % The length of the signal
t=(0:N-1)/Fs;
wc=2*pi*Fc;                  % Fc is the carrier frequency
st=yt'.*cos(wc*t);           % Demodulate the signal
order=20;                    % Choosing the number of order is
                             a trade-off between the precision
                             and the cost of the filter
wn=B/2/Fs;                   % The cut-off frequency
w=window(@hamming,order+1);  % I choose the Hamming window to
                             guarantee less distortion in spite
                             of more transition width
b=fir1(order,wn,'low',w);
yout= filter(b,1,st);        % Filter out all signals except
                              the signal found in the baseband
sound(yout,Fs)                % Play the audio
end
```

The results after playing audios is good enough for me to figure out what sort of music it is, meaning that my choice of the order and the window of the low-pass filter is a good one. The Matlab code to draw the absolute error plots is shown as follows:

```
abs_error_pop=abs(ypop'-yout_pop);
abs_error_cou=abs(ycou'-yout_cou);
abs_error_roc=abs(yroc'-yout_roc);
figure(1)
stem(0:length(abs_error_pop)-1,abs_error_pop)
title('The absolute error plot of the Pop')
xlabel('Sample Number')
ylabel('Amplitude')
figure(2)
stem(0:length(abs_error_cou)-1,abs_error_cou)
title('The absolute error plot of the Country')
xlabel('Sample Number')
ylabel('Amplitude')
figure(3)
```

```
stem(0:length(abs_error_roc)-1,abs_error_roc)
title('The absolute error plot of the Rock')
xlabel('Sample Number')
ylabel('Amplitude')
```

Therefore, the absolute error plots of the Pop, the Country, and the Rock are shown in Fig. 1, Fig. 2, and Fig. 3 respectively.

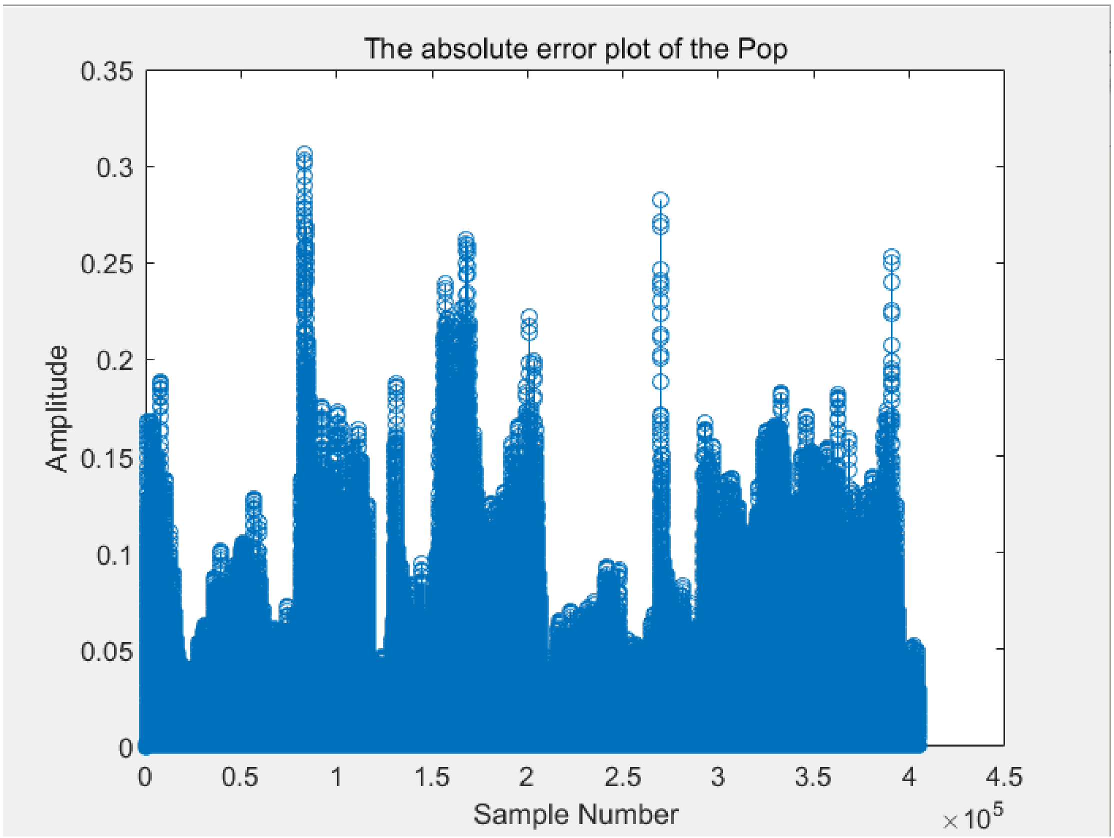

Fig. 1. The absolute error plot of the Pop

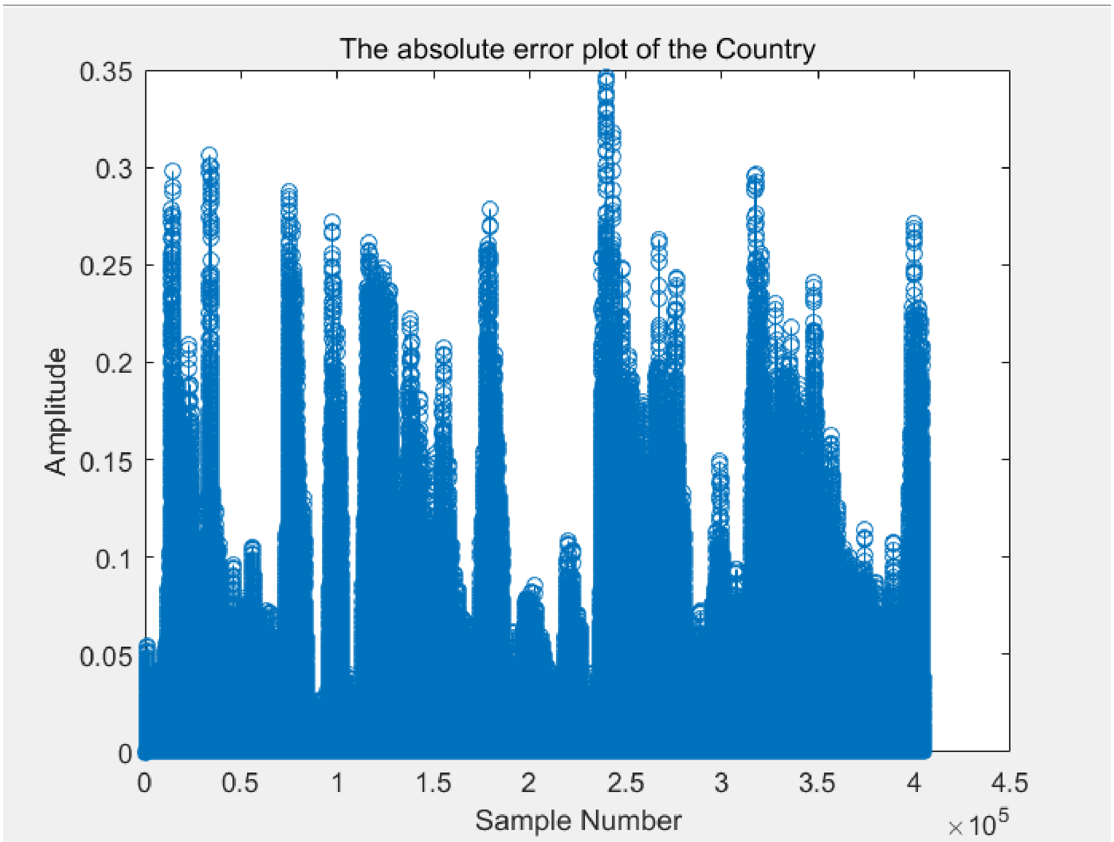

Fig. 2. The absolute error plot of the Country

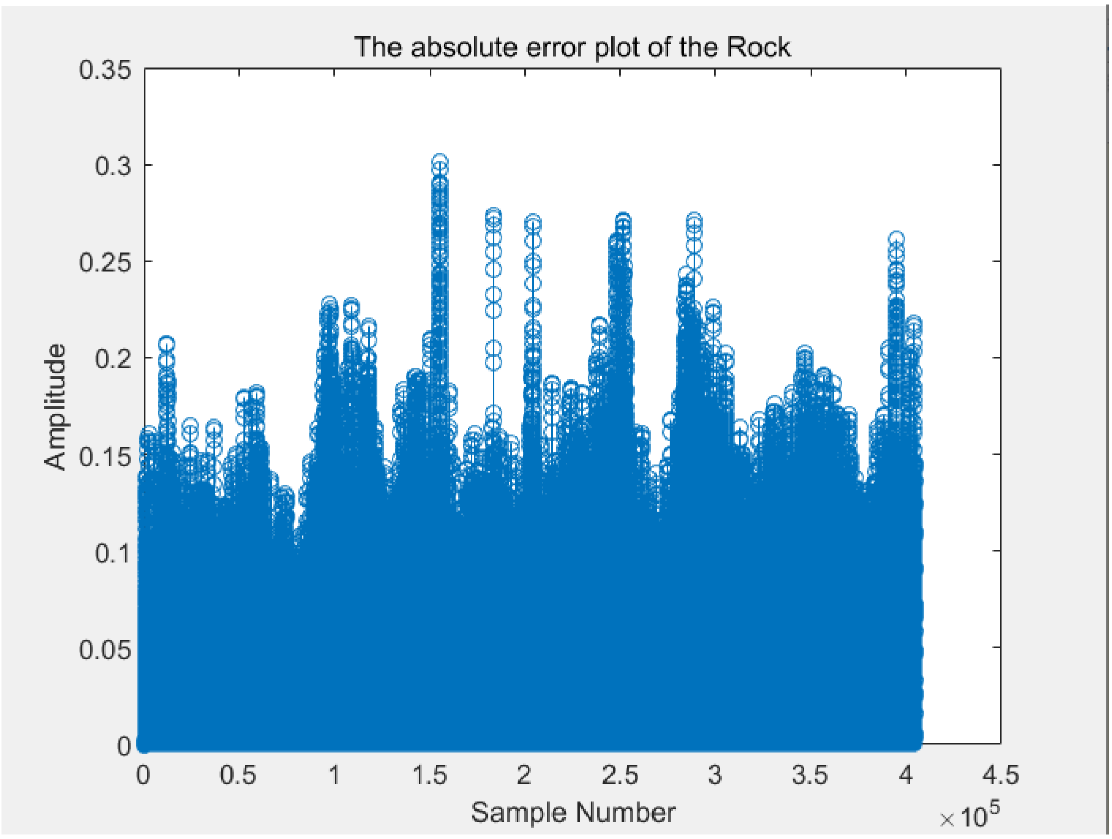

Fig. 3. The absolute error plot of the Rock

2. Solve:

(a)

The Matlab code to realize the impulse invariant approach is shown as follows:

```
B=[1.5000 6.9050 1.8880];   % The numerator of H(s)
A=[1.0000 5.0900 5.7114];   % The denominator of H(s)
[R,P,K]=residue(B,A);        % Expand H(s) into partial
                              fractions
Ts=1.15;
Fs=1/Ts;
bs_a1=R(1);                 % Operate the H1(s)
as_a1=[1 -P(1)];
[BZ_a1,AZ_a1]=impinvar(bs_a1,as_a1,Fs);
bs_a2=R(2);                 % Operate the H2(s)
as_a2=[1 -P(2)];
[BZ_a2,AZ_a2]=impinvar(bs_a2,as_a2,Fs);
bs_a3=K;                   % Operate the H3(s)
% Get the transfer function
A1=conv(AZ_a1,AZ_a2);        % The denominator of the transfer
                             function
B1=1.5*A1+[0 conv(BZ_a1,AZ_a2)]+[0 conv(BZ_a2,AZ_a1)];% The
                            numerator of the transfer function
```

The result is shown as follows:

```
A1 =

    1.0000   -0.1661    0.0029


B1 =

    1.5000   -1.0887   -0.3282
```

(b)

The Matlab code to realize the bilinear transformation is shown as follows:

```
bs_b1=R(1);                 % Operate the H1(s)
as_b1=[1 -P(1)];
[BZ_b1,AZ_b1]=bilinear(bs_b1,as_b1,Fs);
bs_b2=R(2);                 % Operate the H2(s)
as_b2=[1 -P(2)];
[BZ_b2,AZ_b2]=bilinear(bs_b2,as_b2,Fs);
bs_b3=K;                   % Operate the H3(s)
% Get the transfer function
A2=conv(AZ_b1,AZ_b2);         % The denominator of the transfer
                              function
```

```
B2=1.5*A2+conv(BZ_b1,AZ_b2)+conv(BZ_b2,AZ_b1); % The numerator
                                            of the transfer function
```

The result is shown as follows:

```
A2 =

    1.0000    0.3055   -0.0066


B2 =

    1.0481   -0.3012   -0.3175
```

(c)

The Matlab code to draw required plots is shown as follows:

```
w=pi*(0:0.005:0.5);
h=freqs(B,A,w);
h1=freqz(B1,A1,w);
h2=freqz(B2,A2,w);
hmag=abs(h);
hmag1=abs(h1);
hmag2=abs(h2);
figure(1)
plot(w,hmag,'r-');hold on;
plot(w,hmag1,'g--');hold on;
plot(w,hmag2,'b-.');hold on;
title('Magnitude Spectrum Plot')
xlabel('Normalized Frequency (radians)')
ylabel('Magnitude')
```

Therefore, the figure of the magnitude of the frequency response for three systems is shown in Fig. 4, where the red line denotes the original continuous system, the green line denotes the impulse invariant system, and the blue line denotes the bilinear transformation system.

(d)

From Fig. 4, it's easy for us to see that the bilinear approach has given us the best result because it fits the original system with far fewer errors.

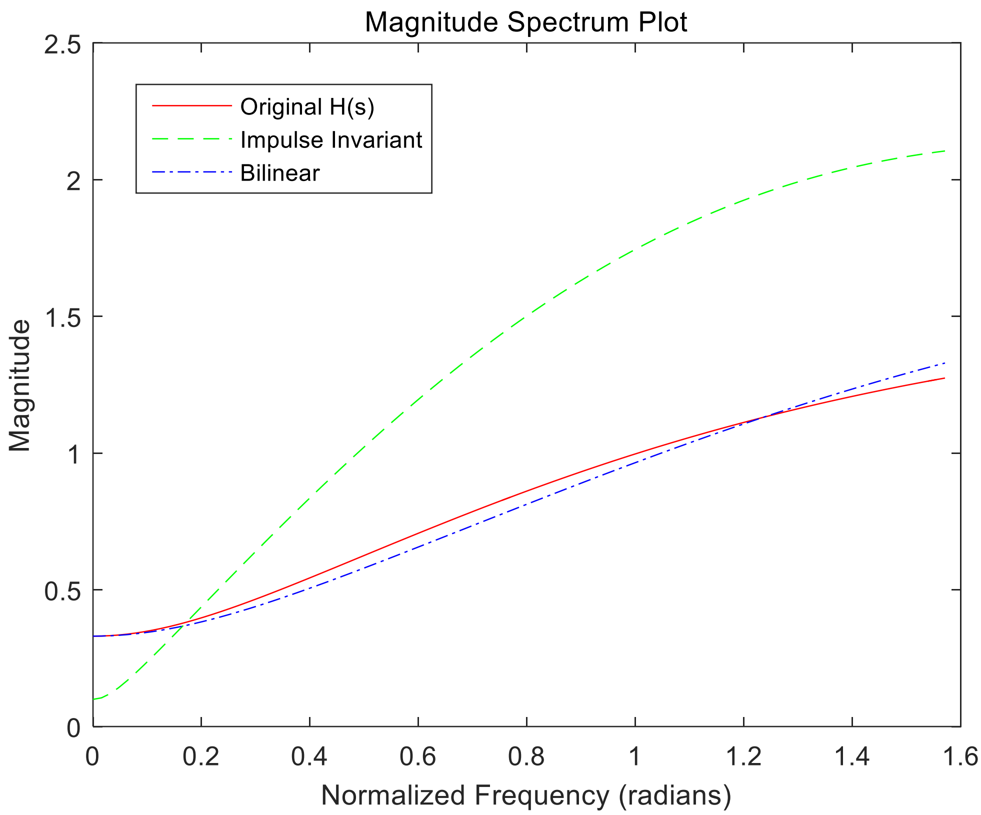

Fig. 4. The magnitude response for three systems

3. Solve:

(a)

$$\Omega_c = 1,\ H(s) = \frac{1}{\left(\frac{s}{\Omega_c}\right)^2 + \sqrt{2}\frac{s}{\Omega_c} + 1}$$

Step 1: Transform the analog lowpass prototype to an analog high-pass filter using the transformation:

$$s \to \frac{\Omega_c \cdot \Omega_c'}{s}$$

So

$$H(s) = \frac{1}{\left(\frac{\frac{\Omega_c \cdot \Omega_c'}{s}}{\Omega_c}\right)^2 + \sqrt{2}\left(\frac{\frac{\Omega_c \cdot \Omega_c'}{s}}{\Omega_c}\right) + 1}$$

Step 2: $\Omega_c = 1$, So $H(s) = \dfrac{1}{\left(\frac{\Omega_c'}{s}\right)^2 + \sqrt{2}\left(\frac{\Omega_c'}{s}\right) + 1}$

Step 3: Since the edge frequency for the high-pass filter is given as $w_c = 0.7\pi$, I can prewarp the desired discrete-time frequency to obtain

$$\hat{\Omega}_c' = \tan\frac{0.7\pi}{2} \approx 1.9626$$

Step 4: Use the transformation: $s \to \dfrac{z-1}{z+1}$,

we get

$$H(z) = \frac{1}{1.9626^2\left(\frac{z+1}{z-1}\right)^2 + \sqrt{2}\times 1.9626 \times \frac{z+1}{z-1} + 1}$$

$$= \frac{(z-1)^2}{1.9626^2 (z+1)^2 + \sqrt{2}\times 1.9626 (z^2-1) + (z-1)^2}$$

$$= \frac{z^2 - 2z + 1}{7.6273z^2 + 5.7036z + 2.0763}$$

$$= \frac{0.1311 - 0.2622z^{-1} + 0.1311z^{-2}}{1 + 0.7478z^{-1} + 0.2722z^{-2}}$$

(b)

The Matlab code to draw the magnitude response is shown as follows:

```
A_z=1.9626^2*[1 2 1]+sqrt(2)*1.9626*[1 0 -1]+[1 -2 1];
A_z=A_z/7.6273;
B_z=[1 -2 1]/7.6273;
order=2;
wc=0.7;
[b0,a0]=butter(order,wc,'high');
[h,w]=freqz(B_z,A_z,200);
hmag=20*log10(abs(h));
figure(1)
```

```
plot(w,hmag);
title('Magnitude Plot');
xlabel('Frequency (radians)');
ylabel('Magnitude (dB)');
```

The figure of the magnitude response of my digital filter is shown in Fig. 5:

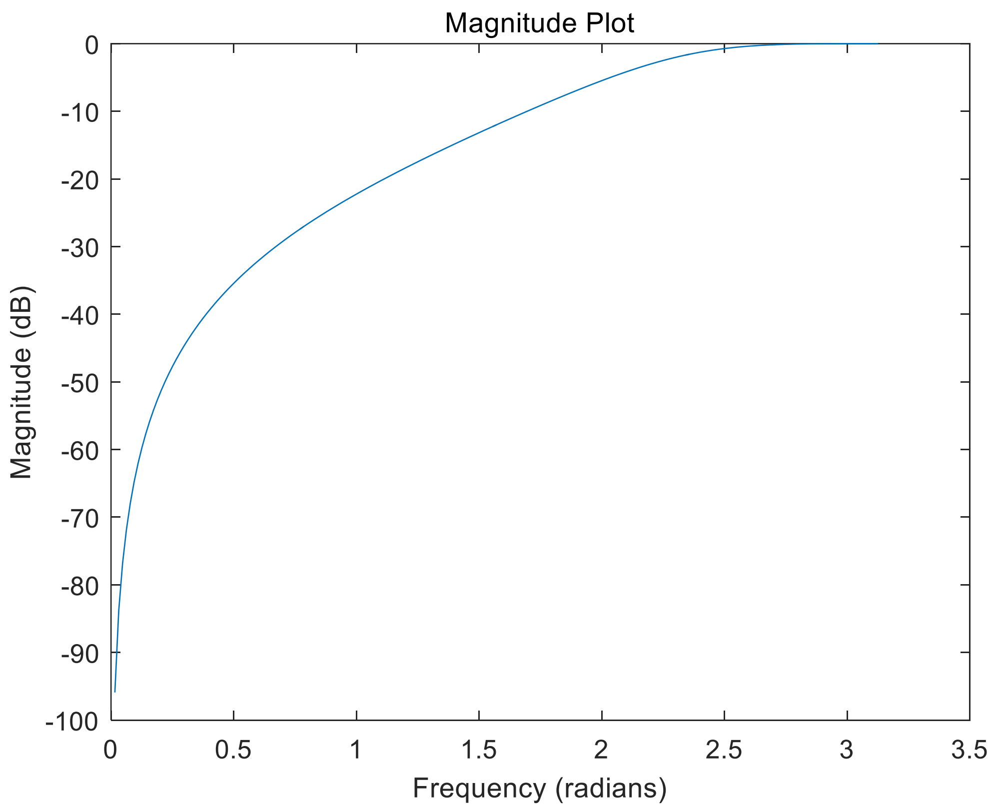

Fig. 5. The magnitude response of my designed digital filter

4. Solve:

(a)

$$H(z) = \frac{0.0803 + 0.1089 z^{-1} + 0.1666 z^{-2} + 0.1089 z^{-3} + 0.0803 z^{-4}}{1.0000 - 1.4655 z^{-1} + 1.7272 z^{-2} - 0.9687 z^{-3} + 0.3187 z^{-4}}$$

Use the low-pass to high-pass spectral transformation:

$$z^{-1} \rightarrow -\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$$

$$\alpha = -\frac{\cos[0.5(0.7\pi + 0.4\pi)]}{\cos[0.5(0.7\pi - 0.4\pi)]} \approx 0.1756$$

Therefore, we have

$$z^{-1} = -\frac{z^{-1} + 0.1756}{1 + 0.1756 z^{-1}} = -\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}$$

Next, the system transfer function for the transferred filter is given by:

$$H(z) = \frac{R(z)}{S(z)}$$

$$R(z) = 0.0803 - 0.1089 \frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}} + 0.1666 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^2 - 0.1089 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^3 + 0.0803 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^4$$

$$S(z) = 1 + 1.4655 \frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}} + 1.7272 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^2 + 0.9687 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^3 + 0.3187 \left(\frac{0.1756 + z^{-1}}{1 + 0.1756 z^{-1}}\right)^4$$

So
$$H(z) = \frac{0.05 - 0.0081 z^{-1} + 0.075 z^{-2} - 0.0081 z^{-3} + 0.05 z^{-4}}{1 + 2.2993 z^{-1} + 2.6651 z^{-2} + 1.5698 z^{-3} + 0.4186 z^{-4}}$$

(b)

The Matlab code of using function **ellip** is shown as follows:

```
% Get the transfer function
b=[0.1756 1];
a=[1 0.1756];
B_z=0.0803*conv(conv(a,a),conv(a,a))-
0.1089*conv(conv(b,a),conv(a,a))+0.1666*conv(conv(b,b),conv(a,
a))-
0.1089*conv(conv(b,b),conv(b,a))+0.0803*conv(conv(b,b),conv(b,
b));
A_z=1.0000*conv(conv(a,a),conv(a,a))+1.4655*conv(conv(b,a),con
v(a,a))+1.7272*conv(conv(b,b),conv(a,a))+0.9687*conv(conv(b,b)
,conv(b,a))+0.3187*conv(conv(b,b),conv(b,b));
```

```
B_z=B_z/A_z(1);              % The numerator of the transfer function
A_z=A_z/A_z(1);              % The denominator of the transfer function
% Use ellip to get the transfer function
order=4;
[b2,a2]=ellip(order,1,34,0.7,'high');
```

The result is shown as follows:

```
B_z =

    0.0500   -0.0081    0.0750   -0.0081    0.0500


A_z =

    1.0000    2.2993    2.6651    1.5698    0.4186


b2 =

    0.0500   -0.0082    0.0750   -0.0082    0.0500


a2 =

    1.0000    2.2992    2.6650    1.5697    0.4186
```
,

where two results are almost the same.

(c)

The Matlab code to draw the magnitude response is shown as follows:

```
[h,w]=freqz(B_z,A_z,200);
hmag=20*log10(abs(h));
figure(1)
plot(w,hmag);
title('Magnitude Plot');
xlabel('Frequency (radians)');
ylabel('Magnitude (dB)');
```

The figure of the magnitude response of my digital filter is shown in Fig. 6:

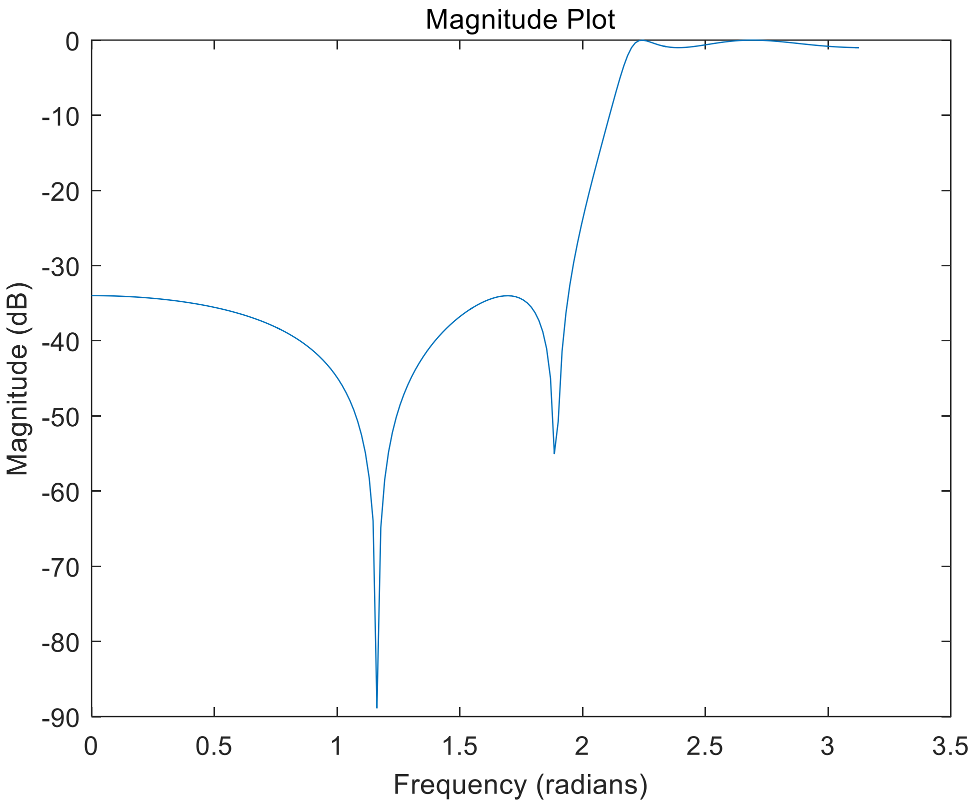

Fig. 6. The magnitude response of my designed digital filter

5. Solve:

(a)

The Matlab code to use function **firpmord** and **firpm** is shown as follows:

```
fs=44056;                         % The sampling frequency
delt=2/fs;
wc1=delt*4000;                     % The pass band edge1
wc2=delt*18000;                    % The pass band edge2
ws1=delt*2000;                     % The stop band edge1
ws2=delt*20000;                    % The stop band edge2
delp=0.15;                         % Pass band ripple
dels=0.01;                         % Stop band ripple
fedge=[ws1 wc1 wc2 ws2];
mval=[0 1 0];
dev=[0.85*dels 0.9*delp 0.85*dels];
[A,Fo,Ao,W]=firpmord(fedge,mval,dev);
b=firpm(A,Fo,Ao,W);
```

(b)

The Matlab code to plot the magnitude and phase response of my designed filter is shown as follows:

```
w=pi*(0:0.005:1.0);
h=freqz(b,1,w);
hmag=abs(h);
hphase=angle(h);
tol=0.95*pi;
figure(3)
subplot(2,1,1)                      % Plot the magnitude
                                    response of the resulting
                                    filter
hmag=unwrap(hmag,tol);
plot(w,hmag)
title('Frequency Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude')
subplot(2,1,2)                      % Plot the phase response
                                    of the resulting filter
hphase=unwrap(hphase,tol);
plot(w,hphase)
title('Phase Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (decibels)')
```

Therefore, the figure of the magnitude response and phase response of my digital filter are shown in Fig. 7:

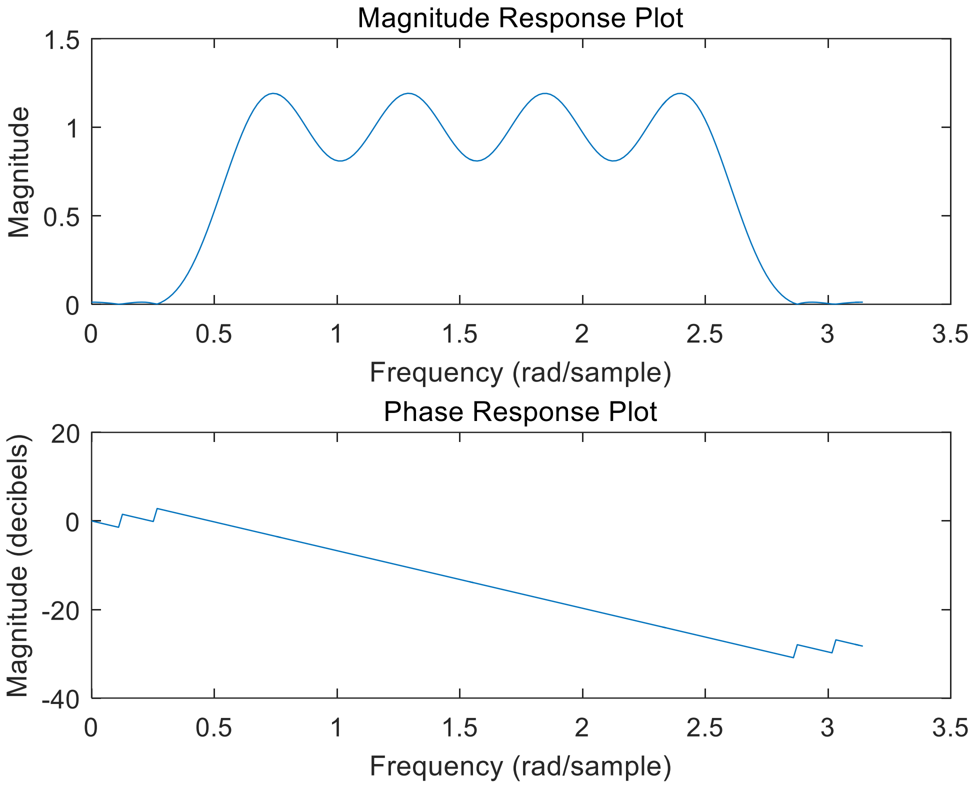

Fig. 7. The magnitude and phase response of my digital filter

(c)

The Matlab code to draw the lines is shown as follows:

```
figure(4)                              % Plot the magnitude response
                                       of the resulting filter
hmag=unwrap(hmag,tol);
plot(w,hmag)
ylim([0 1.25])
title('Magnitude Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude')
line([0,3.5],[1.19,1.19],'linestyle','--','color','g');
text(0.5,1.19,'1.19','FontWeight','bold','Color','g','horiz','
center','vert','bottom')
line([0,3.5],[0.81,0.81],'linestyle','--','color','g');
text(0.4,0.8,'0.81','FontWeight','bold','Color','g','horiz','c
enter','vert','bottom')
line([0,3.5],[0.02,0.02],'linestyle','--','color','m');
text(0.2,0.02,'0.02','FontWeight','bold','Color','m','horiz','
center','vert','bottom')
text(3,0.02,'0.02','FontWeight','bold','Color','m','horiz','ce
```

```
nter','vert','bottom')
```

Therefore, the figure of the magnitude response of my digital filter after drawing line boundaries is shown in Fig. 8:

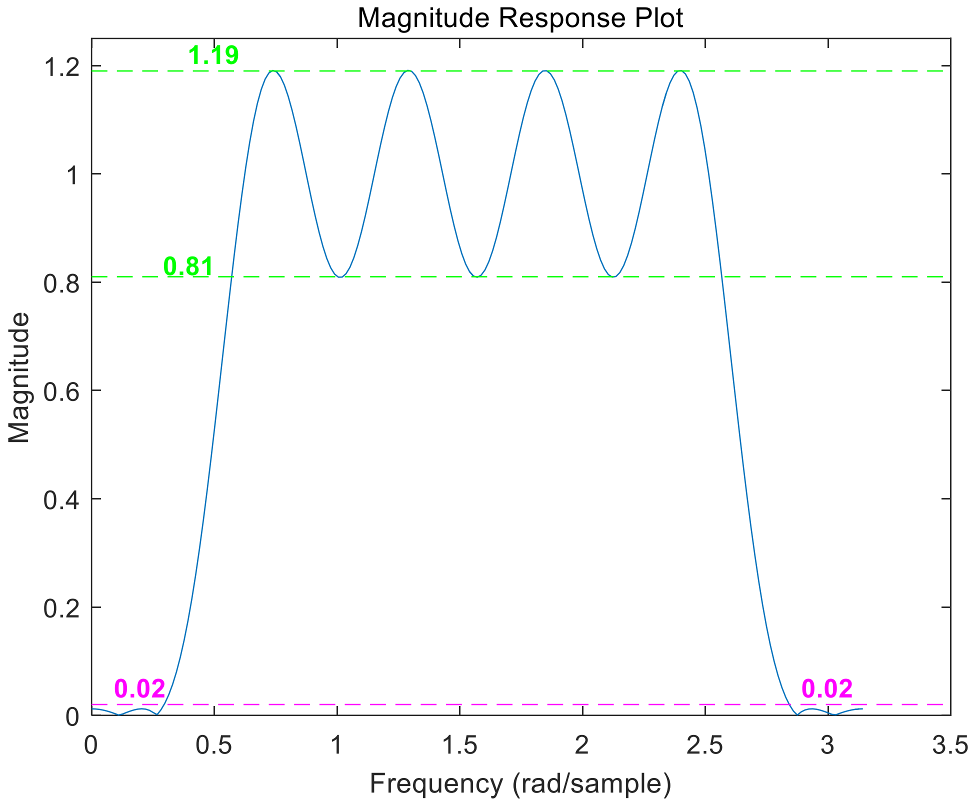

Fig. 8. The magnitude response of my digital filter after drawing line boundaries

Clearly, the digital filter doesn’t meet the ripple requirements.

(d)

The Matlab code to modify parameters, aiming at adjusting the ripples, is shown as follows:

```
fs=44056;
delt=2/fs;
wc1=delt*4000;
wc2=delt*18000;
ws1=delt*2000;
ws2=delt*20000;
delp=0.15;
dels=0.01;
fedge=[ws1 wc1 wc2 ws2];
mval=[0 1 0];
dev=[0.5*dels 0.45*delp 0.5*dels];        % Modify the weighting
                                          parameters
[A,Fo,Ao,W]=firpmord(fedge,mval,dev);
b=firpm(A,Fo,Ao,W);
w=pi*(0:0.005:1.0);
```

```
h=freqz(b,1,w);
hmag=abs(h);
hphase=angle(h);
tol=0.95*pi;
figure(5)                          % Plot the magnitude response
                                   of the resulting filter
hmag=unwrap(hmag,tol);
plot(w,hmag)
ylim([0 1.25])
title('Magnitude Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude')
line([0,3.5],[1.15,1.15],'linestyle','--','color','g');
text(0.5,1.15,'1.15','FontWeight','bold','Color','g','horiz','
center','vert','bottom')
line([0,3.5],[0.85,0.85],'linestyle','--','color','g');
text(0.4,0.85,'0.85','FontWeight','bold','Color','g','horiz','
center','vert','bottom')
line([0,3.5],[0.01,0.01],'linestyle','--','color','m');
text(0.2,0.02,'0.01','FontWeight','bold','Color','m','horiz','
center','vert','bottom')
text(3,0.02,'0.01','FontWeight','bold','Color','m','horiz','ce
nter','vert','bottom')
figure(6)
subplot(2,1,1)                      % Plot the magnitude response
                                     of the resulting filter
hmag=unwrap(hmag,tol);
plot(w,hmag)
title('Magnitude Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude')
subplot(2,1,2)                      % Plot the phase response of
                                    the resulting filter
hphase=unwrap(hphase,tol);
plot(w,hphase)
title('Phase Response Plot')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (decibels)')
```

Therefore, the figure of the magnitude response of my digital filter after modifying parameters is shown in Fig. 9. Also, the figure of the magnitude response and phase response of the digital filter are shown in Fig. 10:

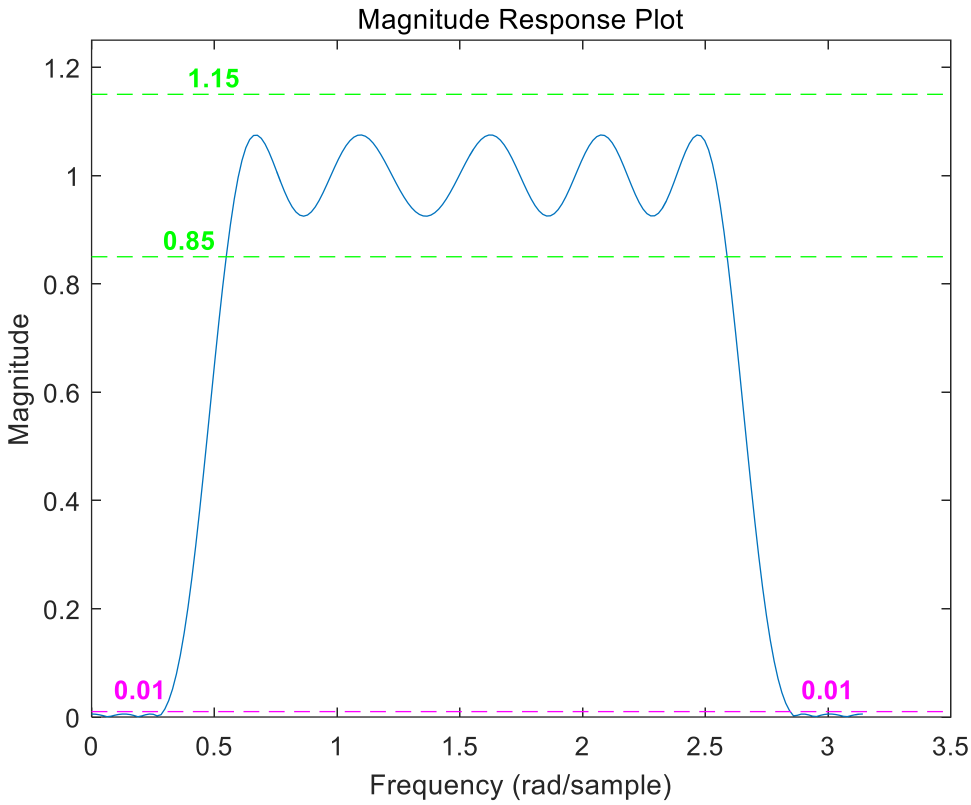

Fig. 9. The magnitude response of my digital filter after modifying parameters

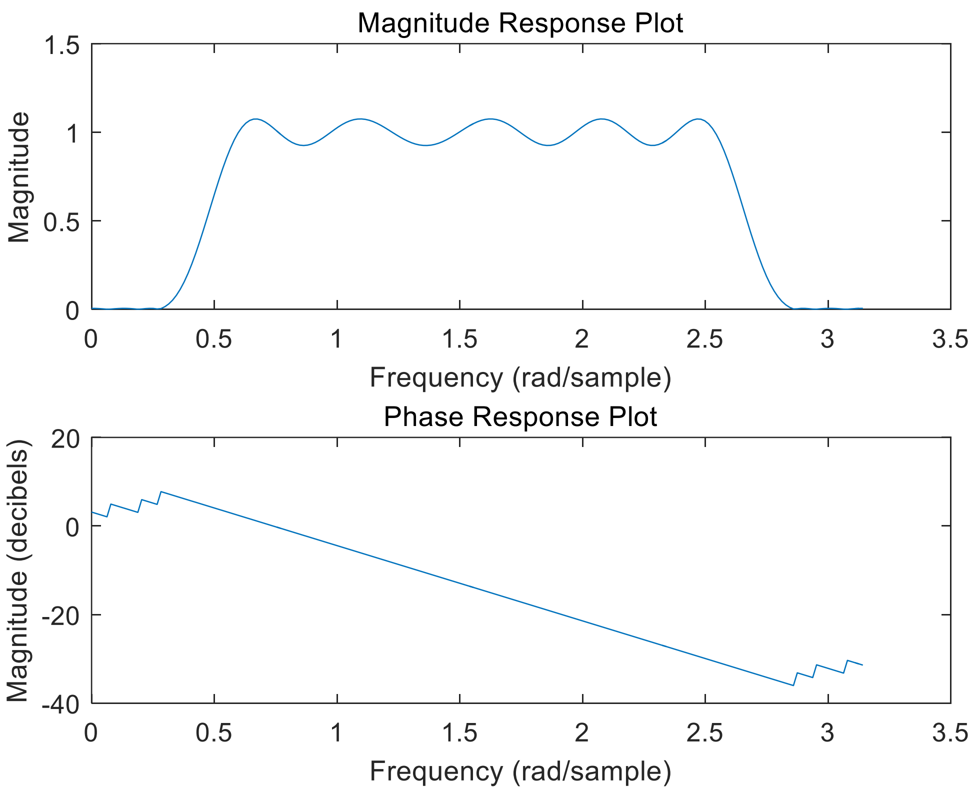

Fig. 10. The magnitude and phase response of my digital filter

(e)

The Matlab code to approximate the frequencies of the pass band edge and the stop band edge is shown as follows:

```
figure(7)
hmag=unwrap(hmag,tol);
plot(w,hmag)
ylim([0 1.25])
title('Magnitude Response Plot')
xlabel('Frequency (rad/sample)')
ylabel('Magnitude')
line([0,3.5],[0.92,0.92],'linestyle','--','color','g');
line([0.57,0.57],[0,1.25],'linestyle','--','color','g');
line([2.57,2.57],[0,1.25],'linestyle','--','color','g');
text(0.69,0.02,'0.57','FontWeight','bold','Color','g','horiz',
'center','vert','bottom')
text(2.69,0.02,'2.57','FontWeight','bold','Color','g','horiz',
'center','vert','bottom')
line([0,3.5],[0.01,0.01],'linestyle','--','color','m');
line([0.3,0.3],[0,1.25],'linestyle','--','color','m');
line([2.83,2.83],[0,1.25],'linestyle','--','color','m');
text(0.2,0.02,'0.3','FontWeight','bold','Color','m','horiz','c
enter','vert','bottom')
text(2.95,0.02,'2.83','FontWeight','bold','Color','m','horiz',
'center','vert','bottom')
```

Therefore, the resulting figure is shown in Fig. 11, by which we can approximate the stop band 1 is 0 to 2103.5 Hz, the pass band is 3996.7 to 18020.1 Hz, and the stop band 2 is 19843.2 to 22028 Hz.

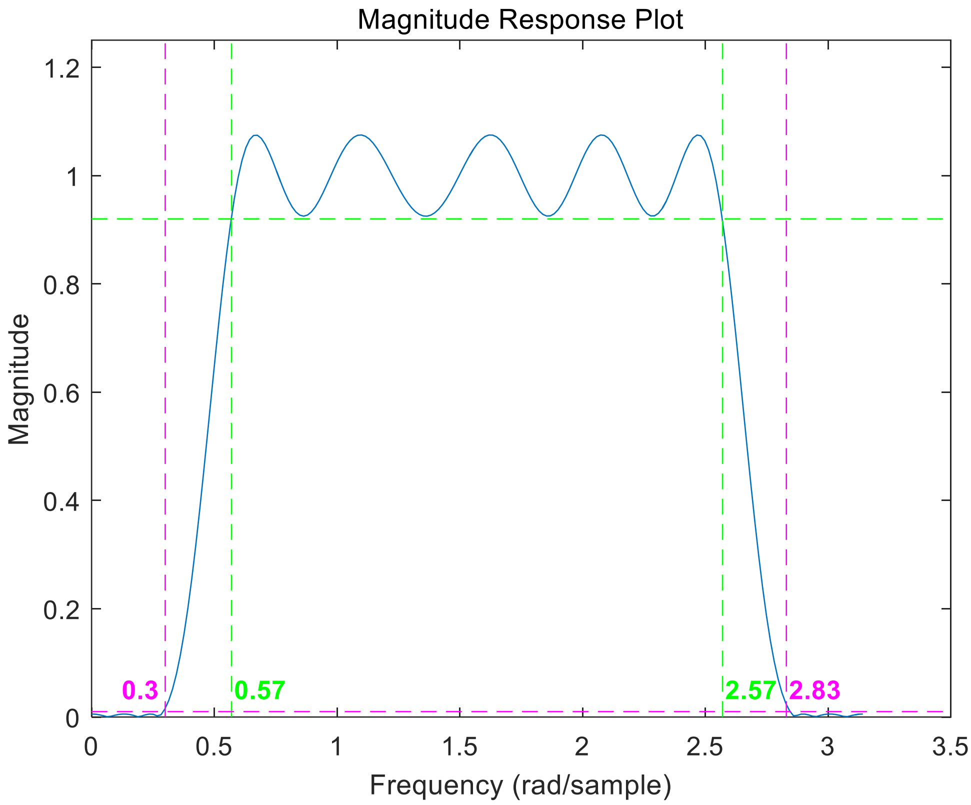

Fig. 11. The figure to approximate the edge frequencies