Eray Gündoğdu 22101962

EEE424 HW3 FALL 2024-25

Q1)

(i) $\frac{d^n x(t)}{dt^n} \xrightarrow{CTFT} (j\omega)^n \cdot X(j\omega)$, take CTFT of both left hand and right hand site of LCCDE, $-\omega^2 Y(j\omega) + 6j\omega Y(j\omega) + 8Y(j\omega) = 2X(j\omega)$

$Y(j\omega)(j\omega+2)(j\omega+4) = 2X(j\omega)$, $H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{2}{(j\omega+2)(j\omega+4)}$

$H(j\omega) = \frac{A}{j\omega+2} + \frac{B}{j\omega+4}$, $Aj\omega+4A+Bj\omega+2B = 2$

$A=-B$, $4A+2B=2$, $B=-1$, $A=1$, $H(j\omega) = \frac{1}{j\omega+2} - \frac{1}{j\omega+4}$

Take Inverse fourier transform, $h(t) = (e^{-2t} - e^{-4t})u(t)$.

(ii) $t\,x(t) \xrightarrow{CTFT} j\frac{dX(j\omega)}{d\omega}$, $e^{-2t} \xrightarrow{CTFT} \frac{1}{j\omega+2}$, $t.e^{-2t} \xrightarrow{CTFT} j\cdot\frac{d(j\omega+2)^{-1}}{d\omega} = X(j\omega)$

$X(j\omega) = \frac{1}{(j\omega+2)^2}$, $H(j\omega).X(j\omega) = \underbrace{\frac{1}{(j\omega+2)^3}}_{Y_1(j\omega)} - \underbrace{\frac{1}{(j\omega+2)(j\omega+4)}}_{Y_2(j\omega)} = Y(j\omega)$

$Y_2(j\omega) = \frac{A}{(j\omega+2)} + \frac{B}{(j\omega+4)} = -\frac{1}{2}\cdot\frac{1}{(j\omega+2)} + \frac{1}{2}\cdot\frac{1}{(j\omega+4)}$

$Y_2(j\omega) \xrightarrow{IFT} (-\frac{1}{2}\cdot e^{-2t} + \frac{1}{2}e^{-4t})u(t)$, $Y_1(j\omega) \xrightarrow{IFT} (\frac{1}{2}\cdot t^2 e^{-2t})u(t)$

$y(t) = \frac{1}{2}u(t)\left(e^{-4t} - e^{-2t} + t^2 e^{-2t}\right)$

Q2) Take DTFT of the difference equation,

$3Y(e^{j\omega}) - e^{-j\omega}Y(e^{j\omega}) + e^{-2j\omega}Y(e^{j\omega}) = \frac{1}{1-\frac{1}{2}e^{-j\omega}}$, $\alpha^n u[n] \xrightarrow{DTFT} \frac{1}{1-\alpha e^{-j\omega}}$

$Y(e^{j\omega})(e^{-2j\omega} - e^{-j\omega} + 3) = \frac{1}{1-\frac{1}{2}e^{-j\omega}}$, let $e^{-j\omega} = \beta$

$Y(e^{j\omega}) = \frac{1}{(\beta^2-\beta+3)(1-\frac{1}{2}\beta)}$ solve for $\beta^2-\beta+3$, $\Delta = -11$

$\beta_1 = \frac{1-j\sqrt{11}}{2}$, $\beta_2 = \frac{1+j\sqrt{11}}{2}$

$Y(e^{j\omega}) = \frac{1}{(\beta-\beta_1).(\beta-\beta_2).(1-\frac{1}{2}\beta)} = \frac{1}{\beta_1\beta_2}\cdot\frac{1}{(1-\frac{1}{\beta_1}\cdot\beta)(1-\frac{1}{\beta_2}\cdot\beta)(1-\frac{1}{2}\beta)}$

$= \frac{A}{(1-\frac{1}{\beta_1}\beta)} + \frac{B}{1-\frac{1}{\beta_2}\beta} + \frac{C}{(1-\frac{1}{2}\beta)} \xrightarrow{IDTFT} \left[A\beta_1^n + B.\beta_2^n + C.(\frac{1}{2})^n\right]u[n] = y[n]$

Q3) Prove $H(e^{jw}) = \pi \cdot \delta(w) + \frac{1}{1-e^{-jw}}$

$$H(e^{jw}) = \sum_{n=0}^{\infty} e^{-jwn} = \lim_{N\to\infty} \sum_{n=0}^{N-1} e^{-jwn} = \lim_{N\to\infty} \frac{1-e^{-jwN}}{1-e^{-jw}}$$

$= \frac{1}{1-e^{-jw}} - \lim_{N\to\infty} \frac{e^{-jwN}}{1-e^{-jw}}$, it is given that $\lim_{M\to\infty} j\frac{\cos(w(M+\frac{1}{2}))}{\sin(\frac{w}{2})} = 0$

$$j\frac{\cos(w(M+\frac{1}{2}))}{\sin(\frac{w}{2})} = -\frac{e^{jwM} \cdot e^{j\frac{w}{2}} + e^{-jwM} \cdot e^{-j\frac{w}{2}}}{e^{j\frac{w}{2}} - e^{-j\frac{w}{2}}} = \frac{-e^{jwM}}{1-e^{-jw}} + \frac{e^{-jwM}}{(1-e^{jw})}$$

$\lim_{M\to\infty} \frac{e^{jwM}}{1-e^{-jw}} = \lim_{M\to\infty} \frac{e^{-jwM}}{1-e^{jw}}$ and from website, $\pi\delta(w) = \frac{1}{2} \cdot \lim_{N\to\infty} \frac{\sin(w(N+\frac{1}{2}))}{\sin(\frac{w}{2})}$

$$\frac{\sin(w(N+\frac{1}{2}))}{\sin(\frac{w}{2})} = \frac{e^{jwN} \cdot e^{j\frac{w}{2}} - e^{-jwN} \cdot e^{-j\frac{w}{2}}}{e^{j\frac{w}{2}} - e^{-j\frac{w}{2}}} = \frac{e^{jwN}}{1-e^{-jw}} + \frac{e^{-jwN}}{1-e^{jw}}$$

$\pi\delta(w) = \frac{1}{2} \lim_{N\to\infty} \frac{e^{jwN}}{1-e^{-jw}} + \lim_{N\to\infty} \frac{e^{-jwN}}{1-e^{jw}}$, Since $\lim_{M\to\infty} \frac{e^{jwM}}{1-e^{-jw}} = \lim_{M\to\infty} \frac{e^{-jwM}}{1-e^{jw}}$

$\pi\delta(w) = \frac{1}{2} \cdot \lim_{N\to\infty} \frac{2e^{-jwM}}{1-e^{jw}}$, $\pi\delta(w) = \lim_{N\to\infty} \frac{e^{-jwM}}{1-e^{jw}}$

since w is around zero, $-\pi\delta(w) = \lim_{N\to\infty} \frac{1}{\underset{\to e^{j(0)}}{1}} \cdot \frac{e^{-jwM}}{1-e^{-jw}}$, Hence

$$H(e^{jw}) = \frac{1}{1-e^{-jw}} - \lim_{N\to\infty} \frac{e^{-jwN}}{1-e^{-jw}} = \frac{1}{1-e^{-jw}} + \pi \cdot \delta(w)$$

Q4) (i)

Type 1, $n \in \{0, \ldots, M-1\}$, $h[n] = h[M-1-n]$, $H(e^{jw}) = \sum_{n=0}^{M-1} h[n] \cdot e^{-jwn}$

$$H(e^{jw}) = \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot e^{-jwn} + \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot e^{-jw(M-1-n)}$$

$$= \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot \left(e^{-jwn} + e^{-jw(M-1)} \cdot e^{jwn} \right)$$

$$= \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot 2 e^{-jw\frac{(M-1)}{2}} \left(\frac{e^{jw\frac{(M-1)}{2}} \cdot e^{-jwn} + e^{-jw\frac{(M-1)}{2}} \cdot e^{jwn}}{2} \right) = e^{-jw\frac{(M-1)}{2}} \sum_{n=0}^{\frac{M-1}{2}} 2h[n] \cdot \cos\left(w\left(\frac{M-1}{2} - n\right)\right)$$

$A(w) = \sum_{n=0}^{\frac{M-1}{2}} 2h[n] \cdot \cos\left(w\left(\frac{M-1}{2} - n\right)\right)$, $a = \frac{-(M-1)}{2}$, $b = 0$, $H(w) = A(w) \cdot e^{ja+b}$

Type 2, $H(w) = \sum_{n=0}^{M/2-1} h[n] \cdot e^{-jwn} + \sum_{n=0}^{\frac{M}{2}-1} h[n] \cdot e^{-jw(M-1-n)}$ due to symetry.

$$H(w) = \sum_{n=0}^{\frac{M}{2}-1} h[n] \, 2 \cdot e^{-jw\frac{(M-1)}{2}} \left(\frac{e^{jw\frac{(M-1)}{2}} \cdot e^{-jwn} + e^{-jw\frac{(M-1)}{2}} \cdot e^{jwn}}{2} \right) = e^{-jw\frac{(M-1)}{2}} \sum_{n=0}^{\frac{M}{2}-1} 2h[n] \cdot \cos\left(w\left(\frac{M-1}{2} - n\right)\right)$$

$H(w) = A(w) \cdot e^{j(aw+b)}$ $\Rightarrow A(w) = \sum_{n=0}^{\frac{M}{2}-1} 2h[n] \cdot \cos\left(w\left(\frac{M-1}{2} - n\right)\right)$, $a = -\frac{(M-1)}{2}$, $b = 0$

Type 3, $h[n] = -h[M-1-n]$

$$H(w) = \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot e^{-jwn} + \sum_{n=0}^{\frac{M-1}{2}} -h[n] \cdot e^{-jw(M-1-n)}$$

$$H(w) = \sum_{n=0}^{\frac{M-1}{2}} h[n] \cdot 2j \cdot e^{-jw\frac{(M-1)}{2}} \left(\frac{e^{jw\frac{(M-1)}{2}} e^{-jwn} + e^{-jw\frac{(M-1)}{2}} \cdot e^{jwn}}{2j} \right) = e^{-jw\frac{M-1}{2}} \sum_{n=0}^{\frac{M-1}{2}} 2j\, h[n] \cdot \sin\left(w\left(\frac{M-1}{2} - n\right)\right)$$

$H(w) = j \cdot A(w) \cdot e^{j(aw+b)}$ $\Rightarrow A(w) = \sum_{n=0}^{\frac{M-1}{2}} 2h[n] \cdot \sin\left(w\left(\frac{M-1}{2} - n\right)\right)$, $a = -\frac{(M-1)}{2}$, $b = 0$

Type 4, Similar to previous ones,

$$H(w) = \sum_{n=0}^{\frac{M}{2}-1} h[n] \, 2j \cdot e^{-jw\frac{(M-1)}{2}} \left(\frac{e^{jw\frac{(M-1)}{2}} \cdot e^{-jwn} + e^{-jw\frac{(M-1)}{2}} \cdot e^{jwn}}{2j} \right) = e^{-jw\frac{(M-1)}{2}} \sum_{n=0}^{\frac{M}{2}-1} 2j\, h[n] \cdot \sin\left(w\left(\frac{M-1}{2} - n\right)\right)$$

$H(w) = j\, A(w) \cdot e^{j(aw+b)}$ $\Rightarrow A(w) = \sum_{n=0}^{\frac{M}{2}-1} 2h[n] \cdot \sin\left(w\left(\frac{M-1}{2} - n\right)\right)$, $a = -\frac{(M-1)}{2}$, $b = 0$

Q4)

ii) According to part i) all of the phase responses equal to each other hence, the phase is linear and all same as expected. It is known that abnormalities in the context of 'side-lobes' are a result of sudden change in the filters impulse response (i.e. smoother filter).To reduce this abnormalities, in the linear phase type 1 LPF an impulse response which is symmetric around (M-1)/2 and a smooth transition function which is sinc in this case is used as the h[n] (in the expression A(w)) found in part i). HPF, Band-Pass and Stop filters are designed by using the LPF design. The HPF design is the multiplication of low pass filter with $(-1)^n$ which shifts the LPF by π. The band-pass design is substraction of two high pass filters, wheras band-stop design is summation of one LPF and one HPF with appropriate cutoff frequencies.

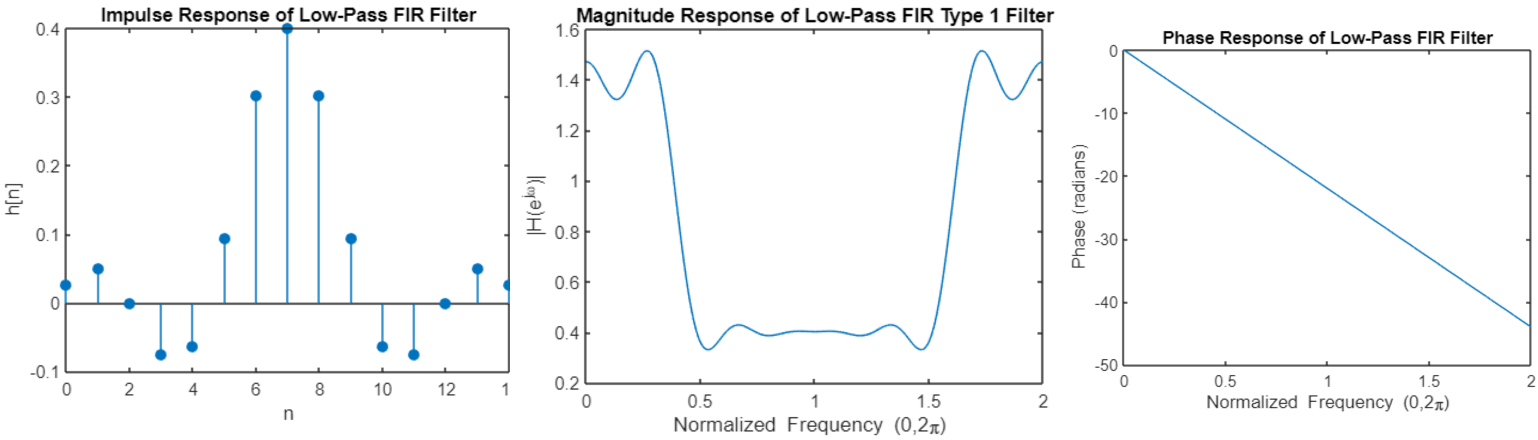

Figure 1 Plots of Linear Phase Type 1 Low-Pass Filter (cutoff freq. 0.4π)

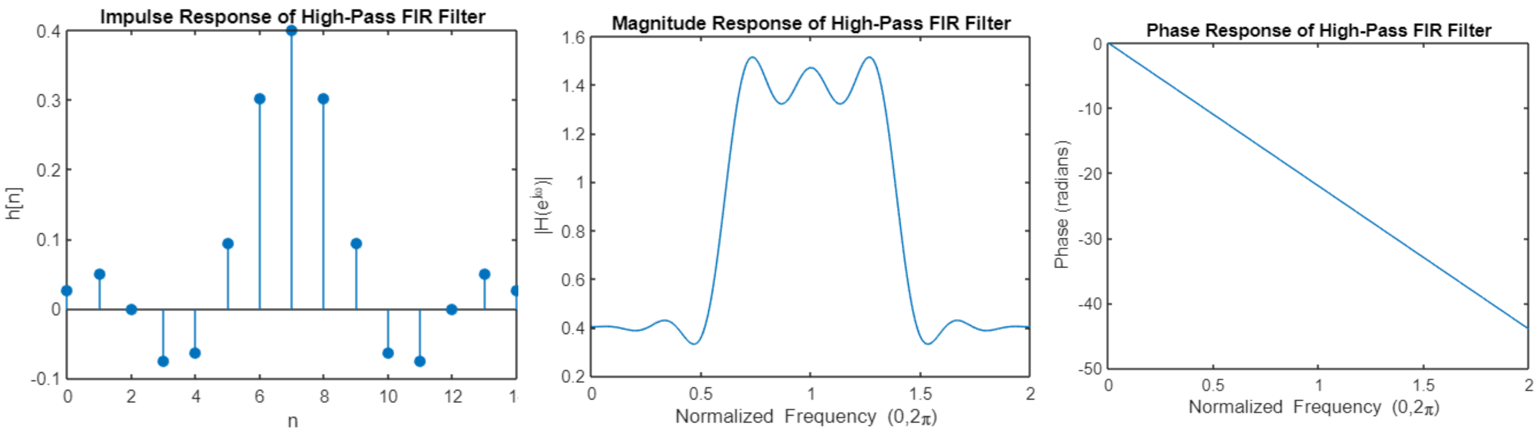

Figure 2 Plots of Linear Phase Type 1 High-Pass Filter (cutoff freq 0.6π)

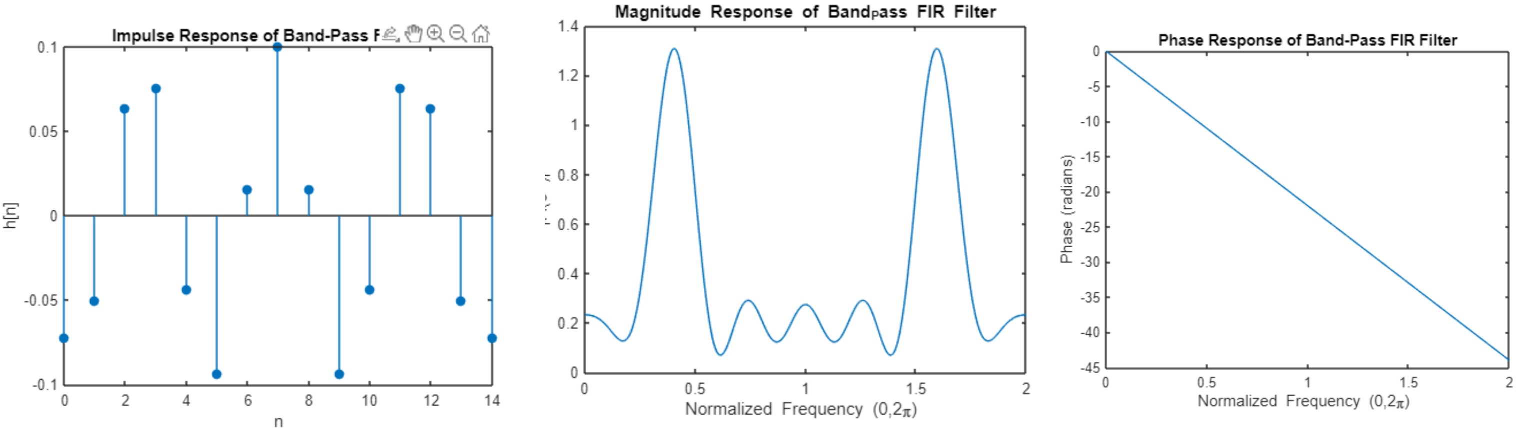

Figure 3 Plots of Linear Phase Type 1 Band-Pass Filter (cutoff freq.0.3π to 0.5π)

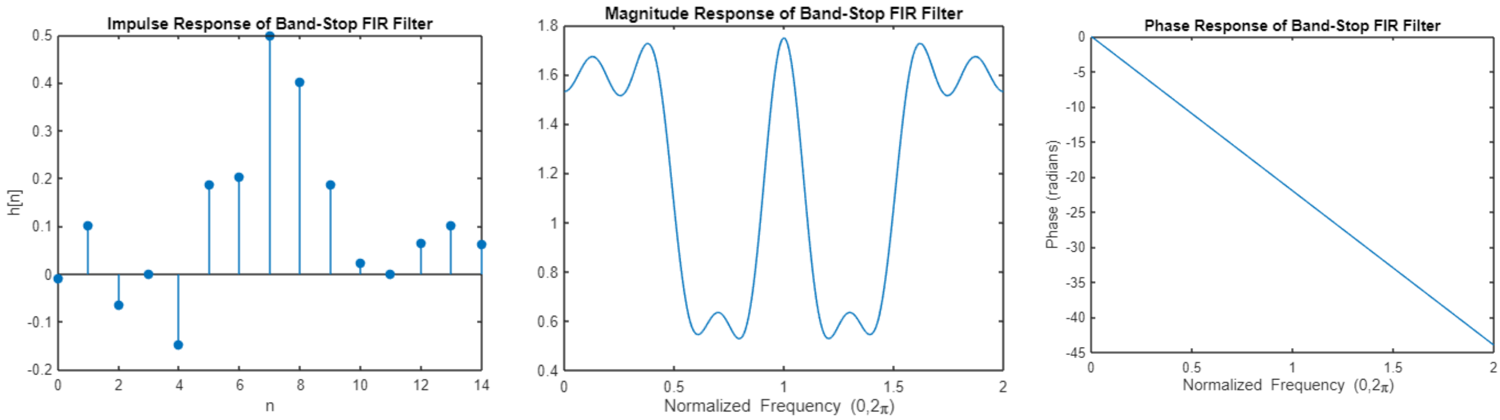

Figure 4 Plots of Linear Phase Type 1 Band-Stop Filter (0.5π to 0.9π)

iii)

At total 12 test samples are generated for chirp, speech and song signals and each of the class is apllied to all four lineaar phase type 1, LPF, HPF, Band-pass, Band-stop, filters. Which can be found in the submission. Aftter observing the samples, it is clear that filters are not working ‘perfectly’ however they work reasonable. With linear phase type 1 filters it is impossible to construct perfect filters however it is observable that for intsance in high pass filter the speech or song amplitude decreases which is parallel with the plots in which the low frequency content is not fully canceled out but significantly decreased in magnitude. Another observation is that while testing the song the notes are different for each output which implies every filter conserves different frequencies. It is important to mention that the chirp signal extents until 0.4π which is a great test tool that demonstartes the filters are working reasonable. It is important to mention taht length of the FIR filter significantly alters the filter performance. As M increases the rate of change at hte cutoff frequency increases (see Figure 5 below). Throughtout the testing M is taken as 15.

In all samples, linear phase type 1 FIR filters are used. Eventough there is a phase the speech or song remains unchanged to the human ear. This is a result of linear phase since all of the phase components of an auido or image is shifted linearly there is no visible or hearable difference in the inputs. This is the reason linear phase filters are advantageous as it introduces constant time delay for all frequencies.

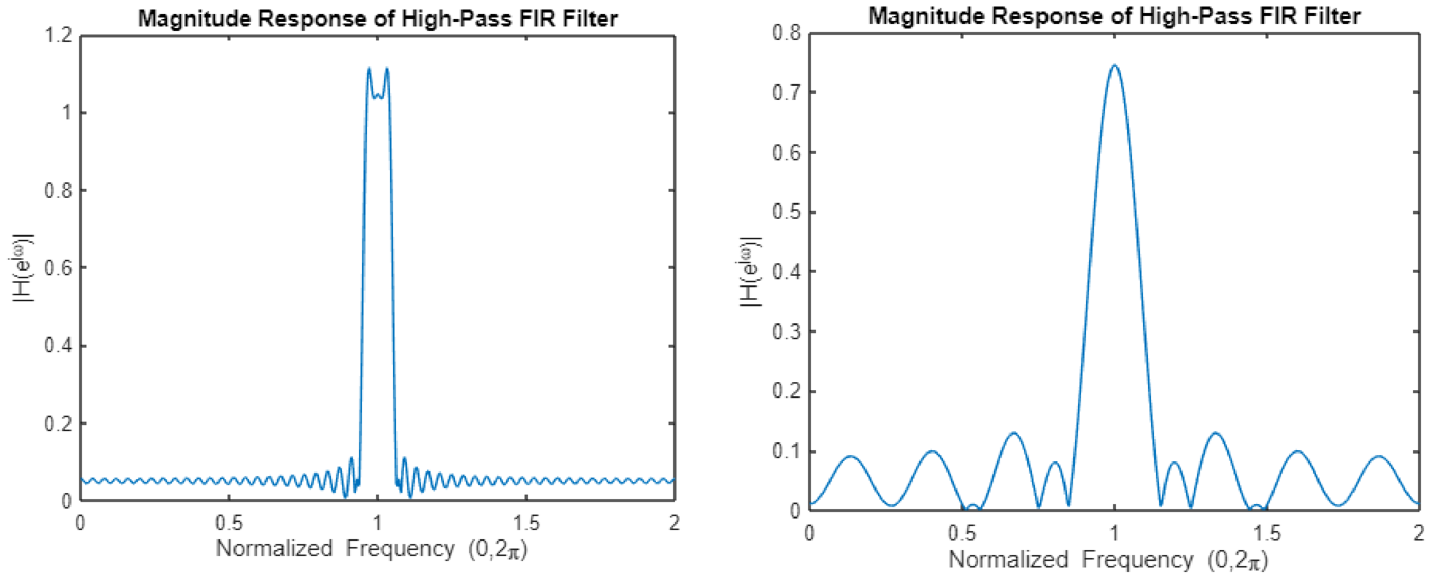

Figure 5 M=101(left) vs M=15(right) with cutoff frequency

MATLAB Codes:

```
Low-Pass Code:

M = 15;
wc = 0.4 * pi;
N = (M - 1) / 2;


h = zeros(1, M);
for n = 0:M-1
    if n == N
        h(n+1) = wc / pi;
    else
        h(n+1) = sin(wc * (n - N)) / (pi * (n - N));
    end
end
omega = linspace(0, 2*pi, 500);
A_omega = zeros(size(omega));
for k = 1:length(omega)
    for n = 0:N
        A_omega(k) = A_omega(k) + 2 * h(n+1) * cos(omega(k) * (N - n));
    end
end
phase_response = -omega * N;
magnitude_response = abs(A_omega);
figure;
stem(0:M-1, h, 'filled');
title('Impulse Response of Low-Pass FIR Filter');
xlabel('n');
ylabel('h[n]');
figure;
plot(omega/pi, magnitude_response);
title('Magnitude Response of Low-Pass FIR Type 1 Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('|H(e^{j\omega})|')
figure;
plot(omega/pi, (phase_response));
title('Phase Response of Low-Pass FIR Filter');
xlabel('Normalized Frequency (0,2\pi)');
ylabel('Phase (radians)')
%%disp('Impulse Response h[n]:');
%%disp(h);

Band-Pass Code:
M = 15;
wc_high = 0.5 * pi;
wc_low = 0.3*pi;
N = (M - 1) / 2;
h1 = zeros(1, M);
```

```
h2 = zeros(1, M);
for n = 0:M-1
    if n == N
        h1(n+1) = wc_high / pi;
    else
        h1(n+1) = sin(wc_high * (n - N)) / (pi * (n - N));
    end
end
for n = 0:M-1
    if n == N
        h2(n+1) = wc_low / pi;
    else
        h2(n+1) = sin(wc_low * (n - N)) / (pi * (n - N));
    end
end
h_high = h1-h2;
omega = linspace(0, 2*pi, 500);
A_omega = zeros(size(omega));
for k = 1:length(omega)
    for n = 0:N
        A_omega(k) = A_omega(k) + 2 * h_high(n+1) * cos(omega(k) * (N - n));
    end
end
phase_response = -omega * N;
magnitude_response = abs(A_omega);
figure;
stem(0:M-1, h_high, 'filled');
title('Impulse Response of Band-Pass FIR Filter');
xlabel('n');
ylabel('h[n]');
figure;
plot(omega/pi, magnitude_response);
title('Magnitude Response of Band_Pass FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('|H(e^{j\omega})|');
figure;
plot(omega/pi, (phase_response));
title('Phase Response of Band-Pass FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('Phase (radians)');

High-Pass Code:
M = 15;
wc = 0.4 * pi;
N = (M - 1) / 2;


h = zeros(1, M);
for n = 0:M-1
    if n == N
        h(n+1) = wc / pi;
    else
```

```
        h(n+1) = sin(wc * (n - N)) / (pi * (n - N));
    end
end
omega = linspace(0, 2*pi, 500);
A_omega = zeros(size(omega));


for k = 1:length(omega)
    for n = 0:N
        A_omega(k) = A_omega(k) + 2 * ((-1)^(n+1))*h(n+1) * cos(omega(k) * (N
- n));
    end
end
phase_response = -omega * N;
magnitude_response = abs(A_omega);
figure;
stem(0:M-1, h, 'filled');
title('Impulse Response of High-Pass FIR Filter');
xlabel('n');
ylabel('h[n]');
figure;
plot(omega/pi, magnitude_response);
title('Magnitude Response of High-Pass FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('|H(e^{j\omega})|');
figure;
plot(omega/pi, (phase_response));
title('Phase Response of High-Pass FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('Phase (radians)');
```

Band-Stop Code:

```
M = 15;
wc_high1 = 0.5 * pi;
wc_low1 = 0.1*pi;
N = (M - 1) / 2;
h_highp = zeros(1, M);
h_highp1 = zeros(1, M);
for n = 0:M-1
    if n == N
        h_highp(n+1) = wc_high1 / pi;
    else
        h_highp(n+1) = sin(wc_high1 * (n - N)) / (pi * (n - N));
    end
end
for n = 0:M-1
    if n == N
        h_highp1(n+1) = wc_low1 / pi;
    else
        h_highp1(n+1) = sin(wc_low1 * (n - N)) / (pi * (n - N));
    end
end
```

```
for n = 0:N
     h_highp1(n+1)= ((-1)^(n+1))*h_highp1(n+1);
     %%h_highp(n+1) = ((-1)^(n+1))*h_highp(n+1);
end
h=h_highp+h_highp1;
omega = linspace(0, 2*pi, 500);
A_omega = zeros(size(omega));
for k = 1:length(omega)
    for n = 0:N
        A_omega(k) = A_omega(k) + 2 *h(n+1) * cos(omega(k) * (N - n));
    end
end
phase_response = -omega * N;
magnitude_response = abs(A_omega);
figure;
stem(0:M-1, h, 'filled');
title('Impulse Response of Band-Stop FIR Filter');
xlabel('n');
ylabel('h[n]');
figure;
plot(omega/pi, magnitude_response);
title('Magnitude Response of Band-Stop FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('|H(e^{j\omega})|');
figure;
plot(omega/pi, (phase_response));
title('Phase Response of Band-Stop FIR Filter');
xlabel('Normalized Frequency (0,2\pi) ');
ylabel('Phase (radians)');
```

Codes for data extraction(music is same with the speech):

```
H_omega = magnitude_response .* exp(1j * phase_response);
```

```
[music, fs_music] = audioread('samplesong.wav');
music_fft = fft(music(1:500000));
music_filtered_mag = abs(music_fft).*magnitude_response;
music_filtered_phase = exp(1j * (phase_response+angle(music_fft)));
music_filtered_fft = music_filtered_mag.*music_filtered_phase;
music_filtered = ifft(music_filtered_fft, 'symmetric');
audiowrite('filtered_music5.wav', music_filtered, fs_music);

fs_chirp = 1000;
t_chirp = 0:1/fs_chirp:2;
chirp_signal = chirp(t_chirp, 0, 2, 0.4);
chirp_fft = fft(chirp_signal);
chirp_filtered_fft = chirp_fft .* H_omega(1:length(chirp_fft));
```

```
filtered_chirp = ifft(chirp_filtered_fft, 'symmetric');
audiowrite('filtered_chirp.wav', filtered_chirp, fs_chirp);
```