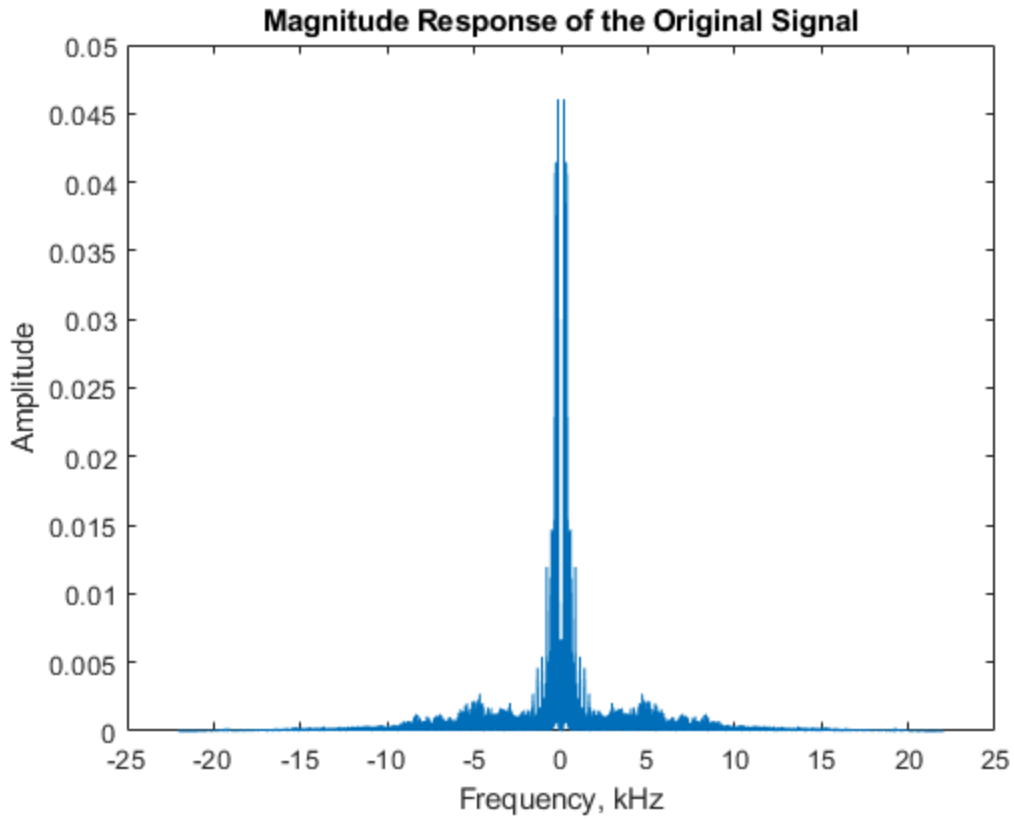
Group No: 66

Table of Contents

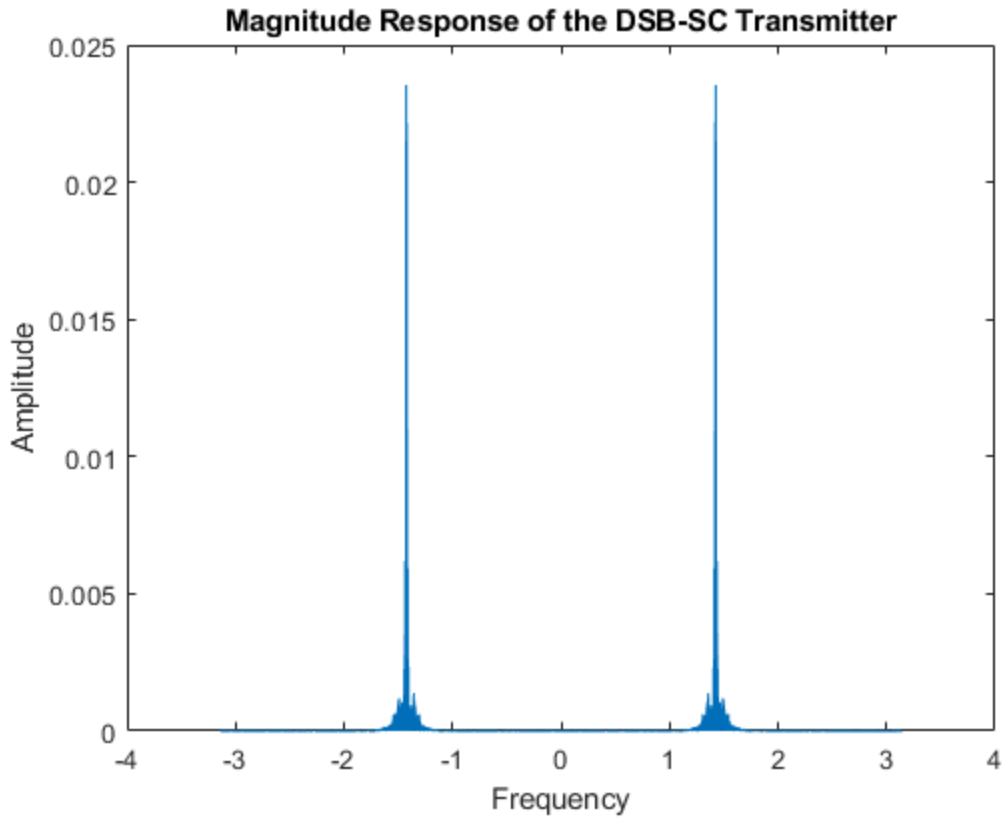
Group Members: Enes AYAZ, Ali AYDIN

```
close all, clear all;
[voice, Fs] = audioread('speech.wav'); % Read voice data from given
 file
% should be in the same directory with code
x = voice / max(abs(voice)).'; % Normalize voice data
% sound(x, Fs); % Listen to the original signal we will work on
Sampling_rate = Fs; %44.1 kHz sampling rate
voice_len = size(x,1);
w_axis = (-Fs/2:Fs/(voice_len-1):Fs/2)/1e3;
figure, plot(w_axis, fftshift(abs(fft(x)))/Fs); % Observe magnitude
 response of the sample signal
% by calculating DFT of the sequence
% fftshift(X), which is a Matlab built-in function, is used to
 rearrange a Discrete Fourier Transform X by shifting the zero-
frequency component to the center of the array
title('Magnitude Response of the Original Signal');
xlabel('Frequency, kHz'), ylabel('Amplitude');
voice_dur = voice_len / Fs;
upsmp_rate = 10;
Fs = Fs * upsmp_rate; %441 kHz sampling rate
t = 0 : voice_dur / (voice_len * upsmp_rate - 1) : voice_dur;
x = interp(x, upsmp_rate);
%A Matlab built-in function y = interp(x,r) increases the sampling
 rate of x, the input signal, by a factor of r. The interpolated
 vector, y, is r times as long as the original input, x
```



QUESTION 1

```
k = (-pi:pi/(5*(voice_len)):pi-pi/(5*(voice_len)));
 s_t = x .* cos(2 * pi * 1e5 .* t');
figure();
plot(k', fftshift(abs(fft(s_t)))/Fs);
title('Magnitude Response of the DSB-SC Transmitter');
xlabel('Frequency'), ylabel('Amplitude');
```

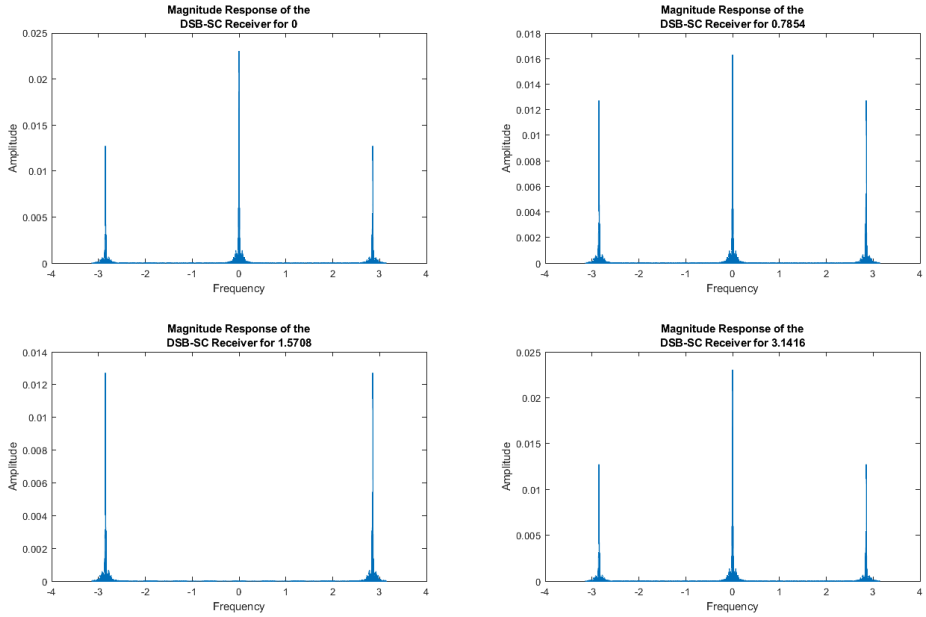


QUESTION 2

Part a

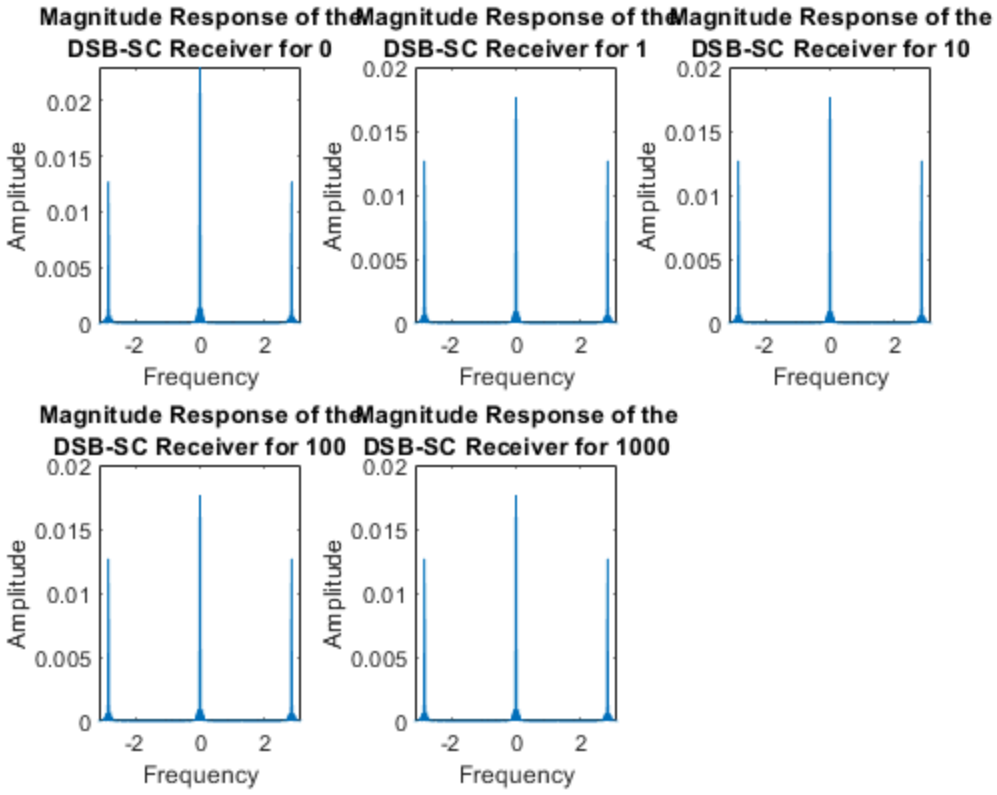
```
figure('Renderer', 'painters', 'Position', [10 10 1500 900]);
a=1;
for teta = 0 : pi /4 : pi
    if(teta ~= 3 *pi /4)
    v_t = s_t .* cos(2 * pi * 1e5 .* t' + teta);
%     hold on

    subplot(2,2,a);
    plot(k', fftshift(abs(fft(v_t)))/Fs);
    str1 = 'Magnitude Response of the' ;
    str2 = 'DSB-SC Receiver for ' +string(teta);
    title({str1; str2});
    xlabel('Frequency'), ylabel('Amplitude');
     a=a+1;
    end
end
```



Part b

```
    deltaF=0;
    a=1;
    figure();
for i=1:5
    v_t = s_t .* cos(2 * pi * (1e5+deltaF) .* t' );
%     hold on
    deltaF= deltaF * 10 ^ (i-2);
    subplot(2,3,a);
    plot(k', fftshift(abs(fft(v_t)))/Fs);
    str1 = 'Magnitude Response of the';
    str2 = 'DSB-SC Receiver for ' +string(deltaF);
    title({str1; str2});
    xlabel('Frequency'), ylabel('Amplitude');
     a=a+1;
     deltaF = 1;
end
```



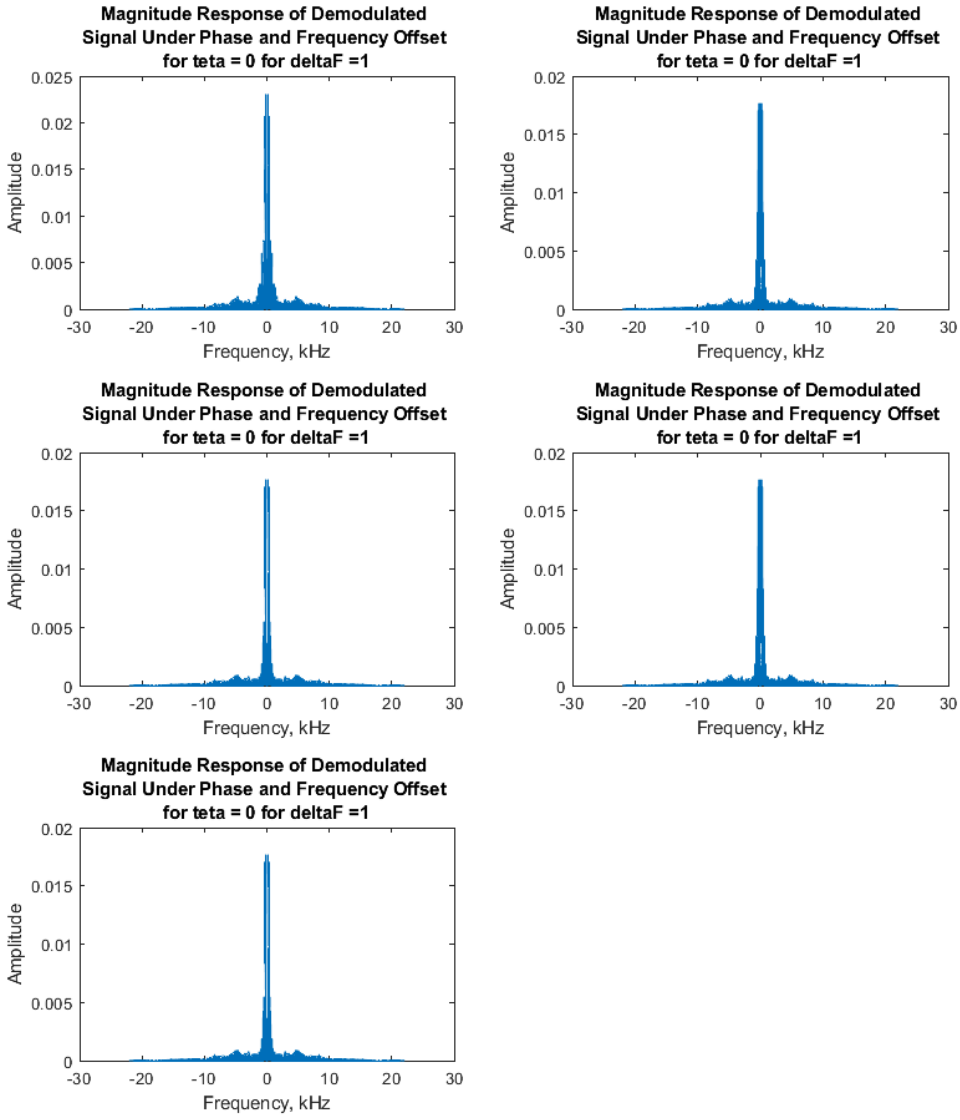
Part c

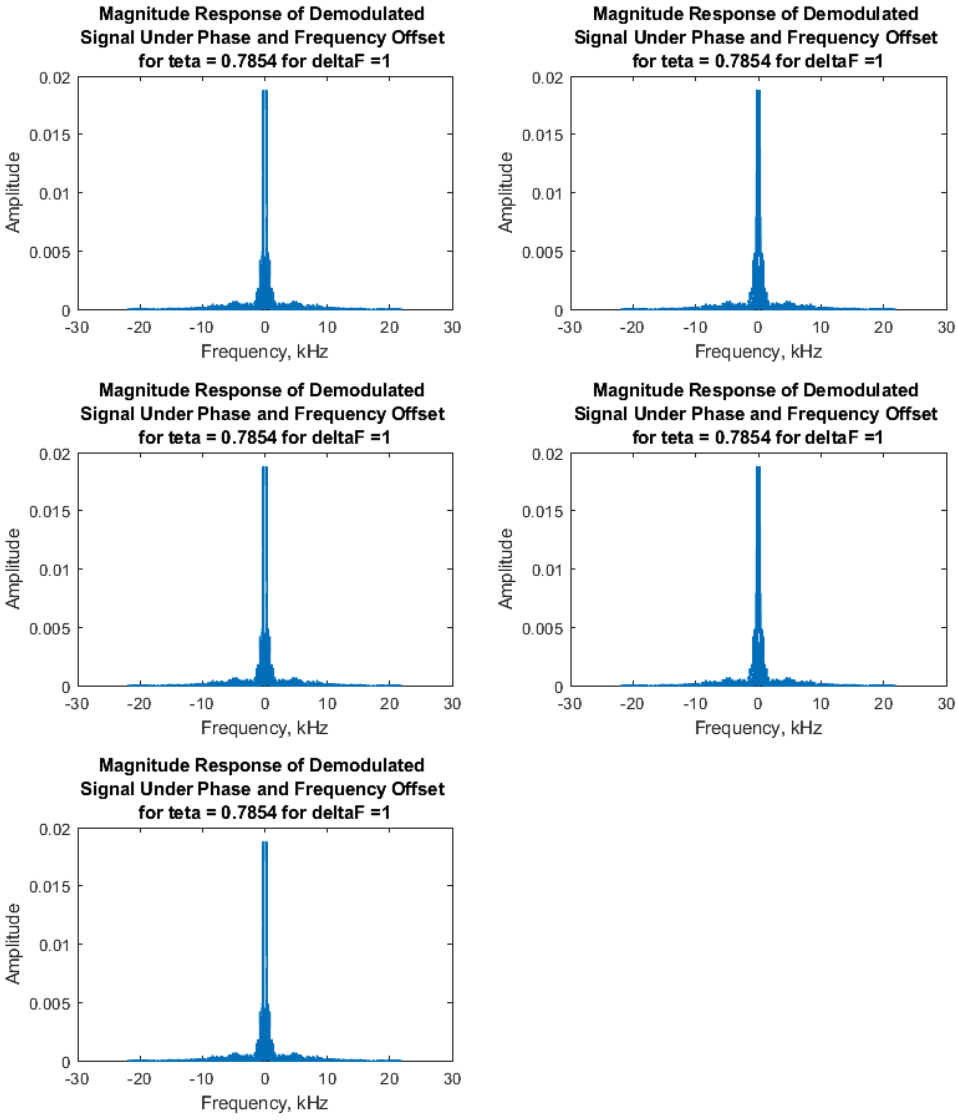
Part d

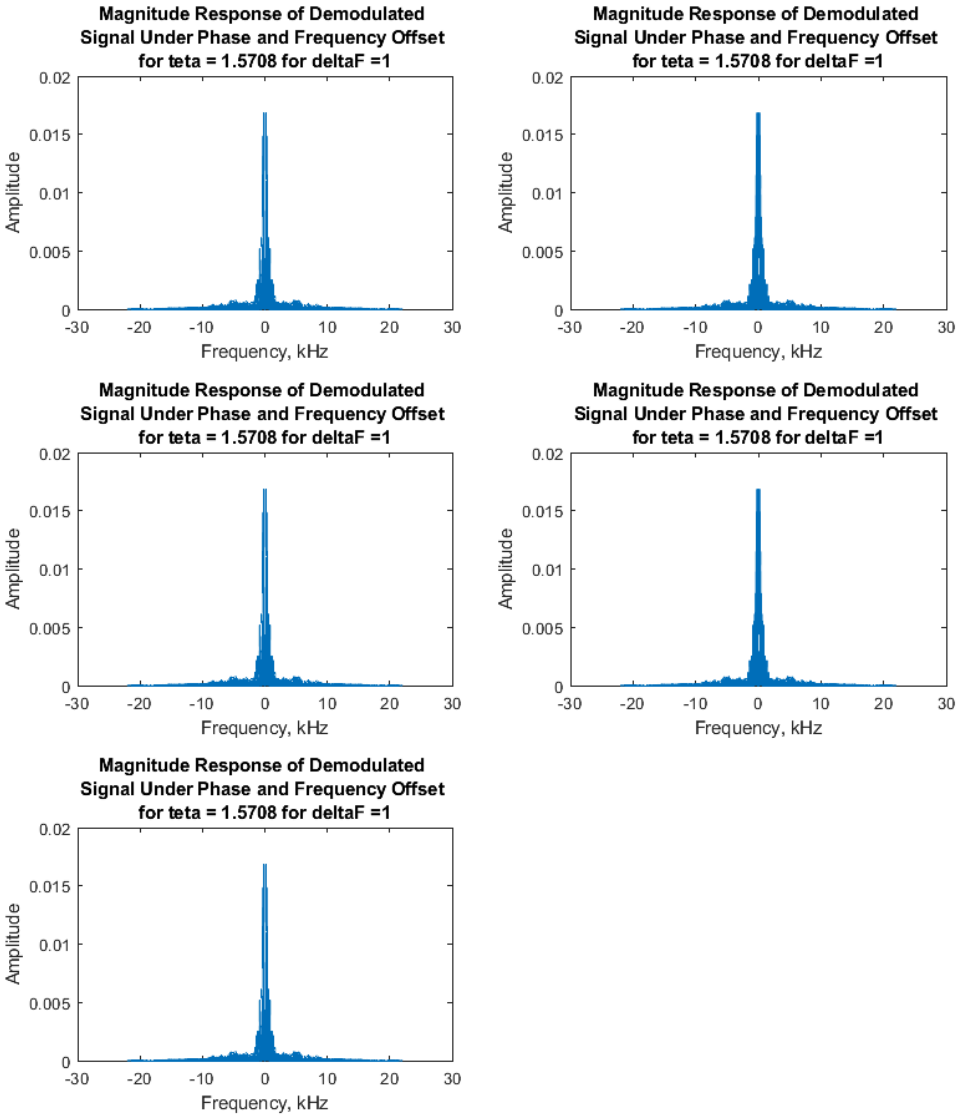
```
fir_order = 128; % The low-pass filter for demodulation will be an FIR
 filter of order 128
downsmp_rate = 10;
Fs = Fs / downsmp_rate;
deltaF=0;
figure('Renderer', 'painters', 'Position', [10 10 900 1500]);
a = 1;
teta = 0;
for i=1:5
    v_t = s_t .* cos(2 * pi * (1e5 + deltaF) .* t' + teta);
    deltaF = deltaF * 10 ^ (i-2);
    deltaF = 1;
    demodulated_signal = decimate(v_t, downsmp_rate,
 fir_order, 'fir');
    w_axis = (-Fs/2:Fs/(voice_len-1):Fs/2)/1e3;
    subplot(3,2,a);
    plot(w_axis, fftshift(abs(fft(demodulated_signal)))/Fs);
    str1 ='Magnitude Response of Demodulated ';
    str2=  'Signal Under Phase and Frequency Offset ';
    str3 =  'for teta = ' + string(teta) +' for deltaF =' +
 string(deltaF);
```

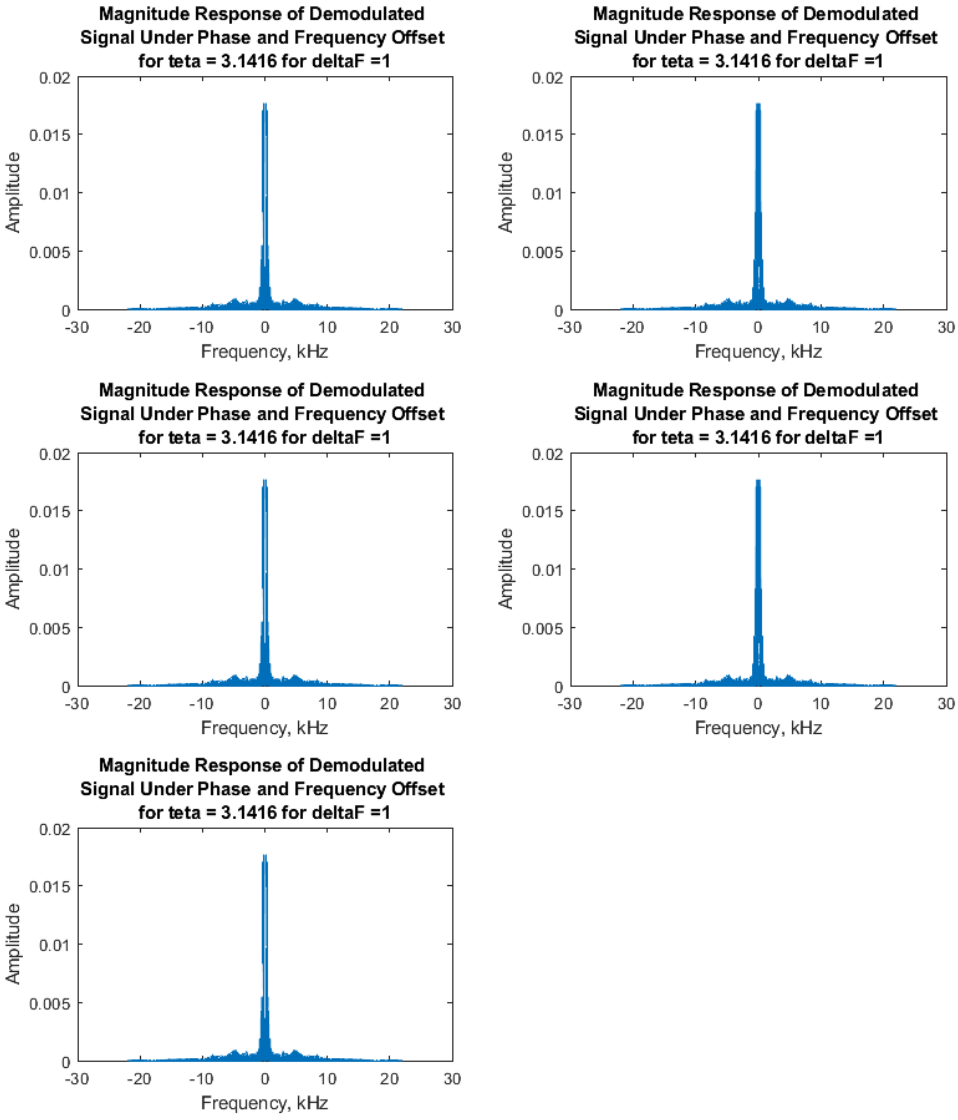
```
    title({str1 ; str2; str3});
    xlabel('Frequency, kHz'), ylabel('Amplitude');
    demodulated_signal = demodulated_signal /
 max(abs(demodulated_signal)); % Normalize demodulated message signal
    %sound(demodulated_signal, Fs);
    a = a + 1;
end
figure('Renderer', 'painters', 'Position', [10 10 900 1500]);
a = 1;
teta = pi / 4;
for i=1:5
    v_t = s_t .* cos(2 * pi * (1e5 + deltaF) .* t' + teta);
    deltaF = deltaF * 10 ^ (i-2);
    deltaF = 1;
    demodulated_signal = decimate(v_t, downsmp_rate,
 fir_order, 'fir');
    w_axis = (-Fs/2:Fs/(voice_len-1):Fs/2)/1e3;
    subplot(3,2,a);
    plot(w_axis, fftshift(abs(fft(demodulated_signal)))/Fs);
    str1 ='Magnitude Response of Demodulated ';
    str2=  'Signal Under Phase and Frequency Offset ';
    str3 =  'for teta = ' + string(teta) +' for deltaF =' +
 string(deltaF);
    title({str1 ; str2; str3});
    xlabel('Frequency, kHz'), ylabel('Amplitude');
    demodulated_signal = demodulated_signal /
 max(abs(demodulated_signal)); % Normalize demodulated message signal
    %sound(demodulated_signal, Fs);
    a = a + 1;
end
figure('Renderer', 'painters', 'Position', [10 10 900 1500]);
a = 1;
teta = pi / 2;
for i=1:5
    v_t = s_t .* cos(2 * pi * (1e5 + deltaF) .* t' + teta);
    deltaF = deltaF * 10 ^ (i-2);
    deltaF = 1;
    demodulated_signal = decimate(v_t, downsmp_rate,
 fir_order, 'fir');
    w_axis = (-Fs/2:Fs/(voice_len-1):Fs/2)/1e3;
    subplot(3,2,a);
    plot(w_axis, fftshift(abs(fft(demodulated_signal)))/Fs);
    str1 ='Magnitude Response of Demodulated ';
    str2=  'Signal Under Phase and Frequency Offset ';
    str3 =  'for teta = ' + string(teta) +' for deltaF =' +
 string(deltaF);
    title({str1 ; str2; str3});
    xlabel('Frequency, kHz'), ylabel('Amplitude');
    demodulated_signal = demodulated_signal /
 max(abs(demodulated_signal)); % Normalize demodulated message signal
    %sound(demodulated_signal, Fs);
    a = a + 1;
end
figure('Renderer', 'painters', 'Position', [10 10 900 1500]);
```

```
a = 1;
teta = pi;
for i=1:5
    v_t = s_t .* cos(2 * pi * (1e5 + deltaF) .* t' + teta);
    deltaF = deltaF * 10 ^ (i-2);
    deltaF = 1;
    demodulated_signal = decimate(v_t, downsmp_rate,
 fir_order, 'fir');
    w_axis = (-Fs/2:Fs/(voice_len-1):Fs/2)/1e3;
    subplot(3,2,a);
    plot(w_axis, fftshift(abs(fft(demodulated_signal)))/Fs);
    str1 ='Magnitude Response of Demodulated ';
    str2=  'Signal Under Phase and Frequency Offset ';
    str3 =  'for teta = ' + string(teta) +' for deltaF =' +
 string(deltaF);
    title({str1 ; str2; str3});
    xlabel('Frequency, kHz'), ylabel('Amplitude');
    demodulated_signal = demodulated_signal /
 max(abs(demodulated_signal)); % Normalize demodulated message signal
    %sound(demodulated_signal, Fs);
    a = a + 1;
end
```


Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0 for deltaF =1
Amplitude
Frequency, kHz
Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0 for deltaF =1
Amplitude
Frequency, kHz
Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0 for deltaF =1
Amplitude
Frequency, kHz
Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0 for deltaF =1
Amplitude
Frequency, kHz
Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0 for deltaF =1
Amplitude
Frequency, kHz


Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 0.7854 for deltaF =1
Amplitude
Frequency, kHz
0.02
0.015
0.01
0.005
0
-30
-20
-10
0
10
20
30


Magnitude Response of Demodulated Signal Under Phase and Frequency Offset for teta = 1.5708 for deltaF =1
Amplitude
Frequency, kHz
0.02
0.015
0.01
0.005
0
-30
-20
-10
0
10
20
30



Published with MATLAB® R2018a