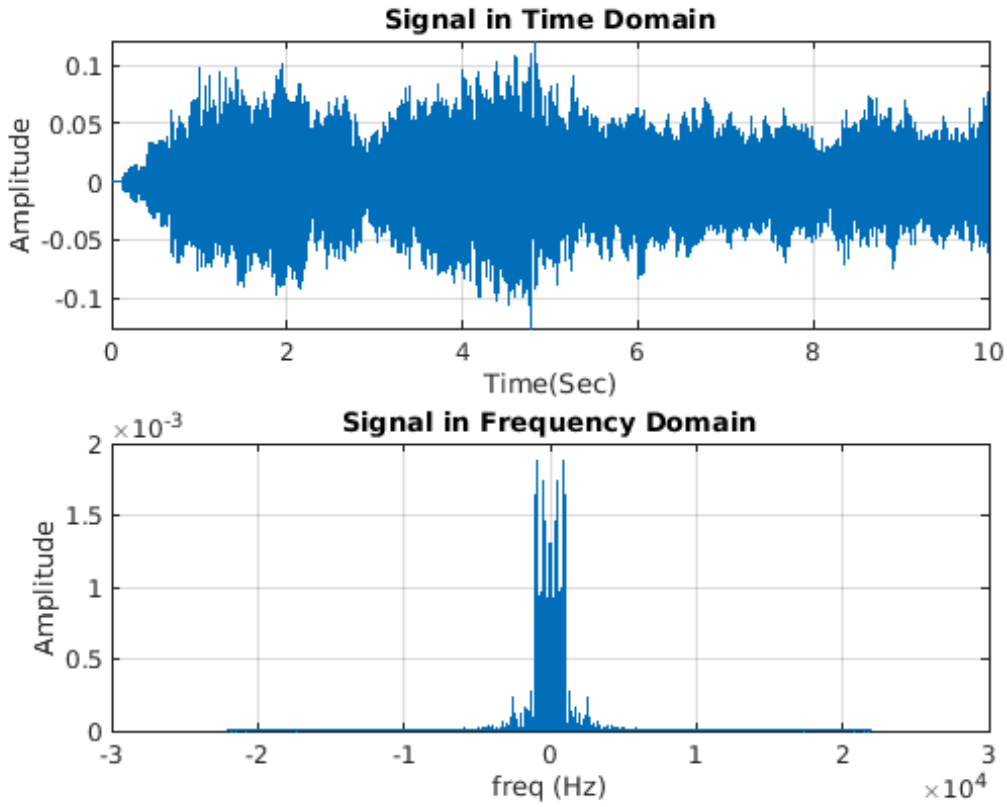
Table of Contents


```
close all;
clear;
clc;
```

2.3.a

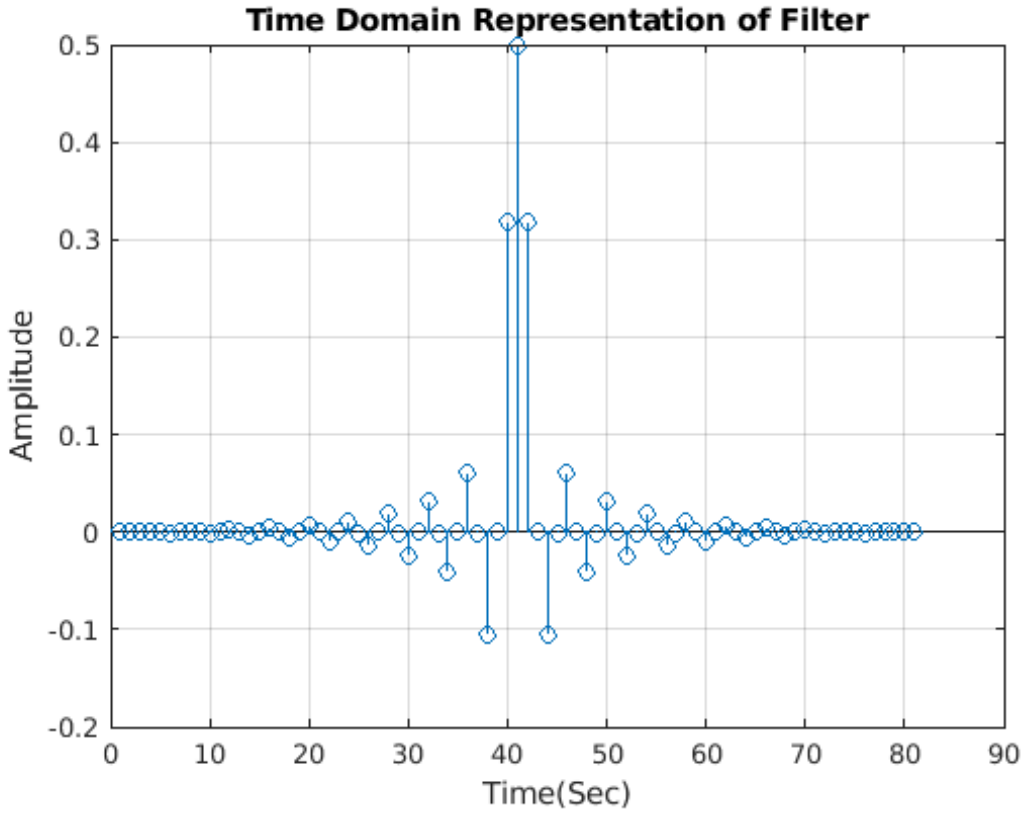
Loading the audio file

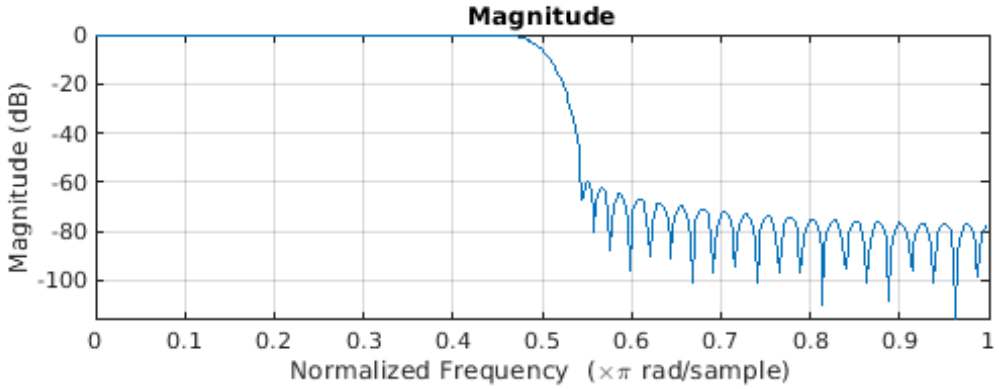
```
[x, Fs] = audioread('Audio01.wav');

% Plotting signal in time and freq domain
plot_time_freq(x, Fs, 'Original audio signal', 'Signal in Time
 Domain', 'Signal in Frequency Domain');

% Playing the audio
sound(x, Fs);
pause(length(x) * (1/ Fs));
```

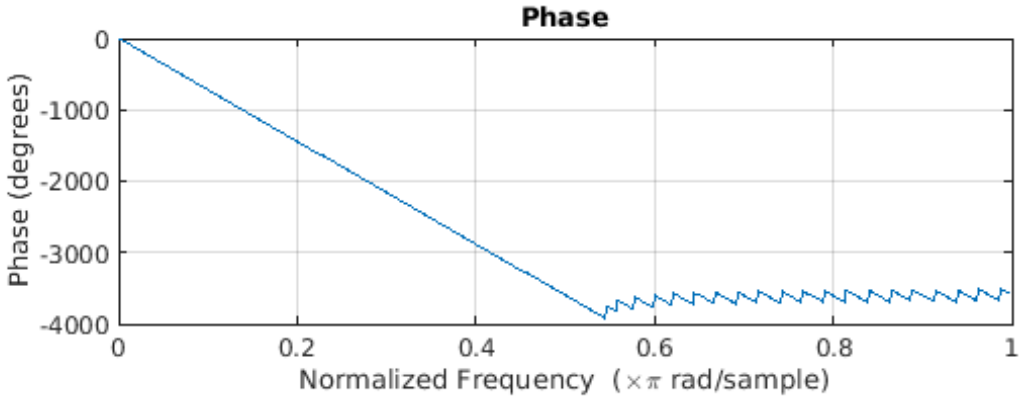


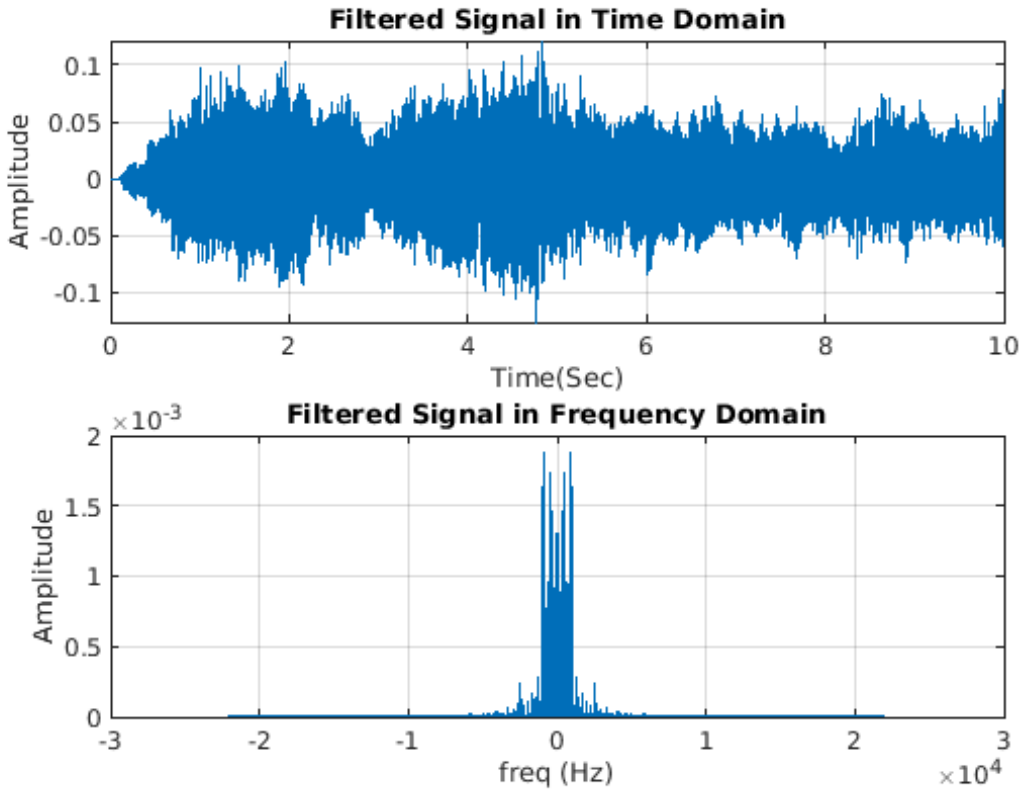
2.3.b

```
load('filter.mat');

% Plotting filter in time domain
figure('Name', 'Time Domain Representation of Filter');
stem(Num, 'LineWidth', 1.5);
grid on;
xlabel('Time(Sec)') ;
ylabel('Amplitude') ;
title('Time Domain Representation of Filter');

figure('Name', 'Filter freq-phase response');
freqz(Num); % Visualising filter freq-phase response

y_0 = filter(Num,1,x); % Applying filter

% Plotting signal in time and freq domain
plot_time_freq(y_0, Fs, 'Filtered audio signal', 'Filtered Signal in Time
 Domain', 'Filtered Signal in Frequency Domain');
```


Time Domain Representation of Filter
Amplitude
Time(Sec)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
10
20
30
40
50
60
70
80
90

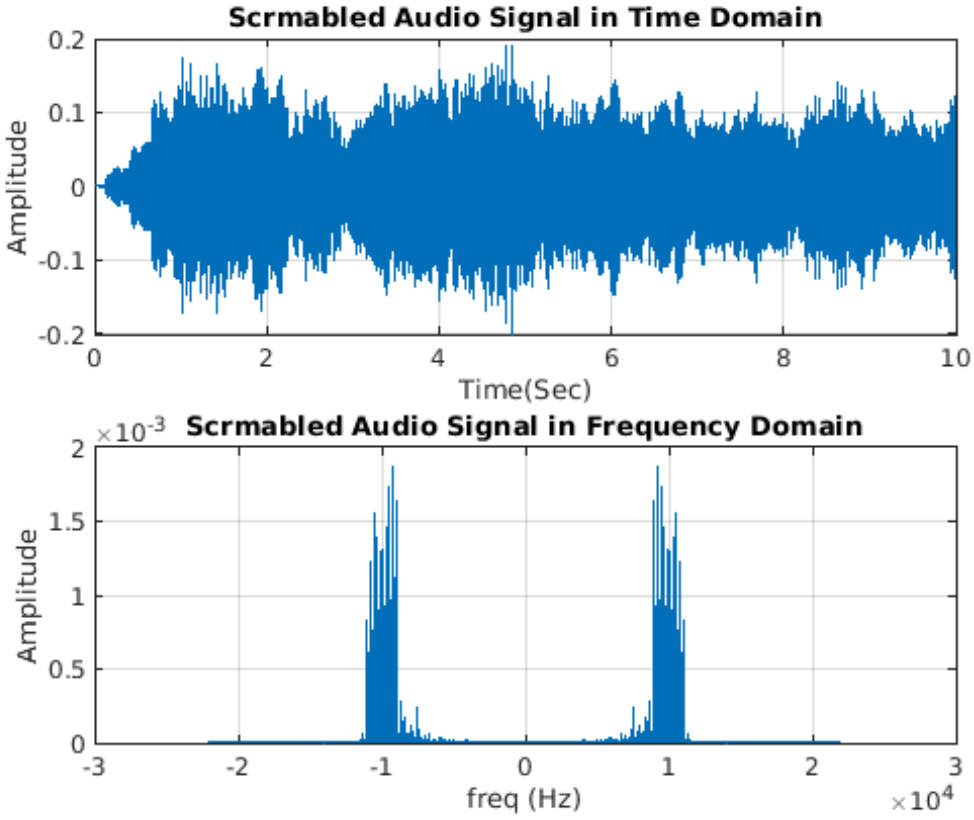

Magnitude
Magnitude (dB)
Normalized Frequency (×π rad/sample)
0
-20
-40
-60
-80
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

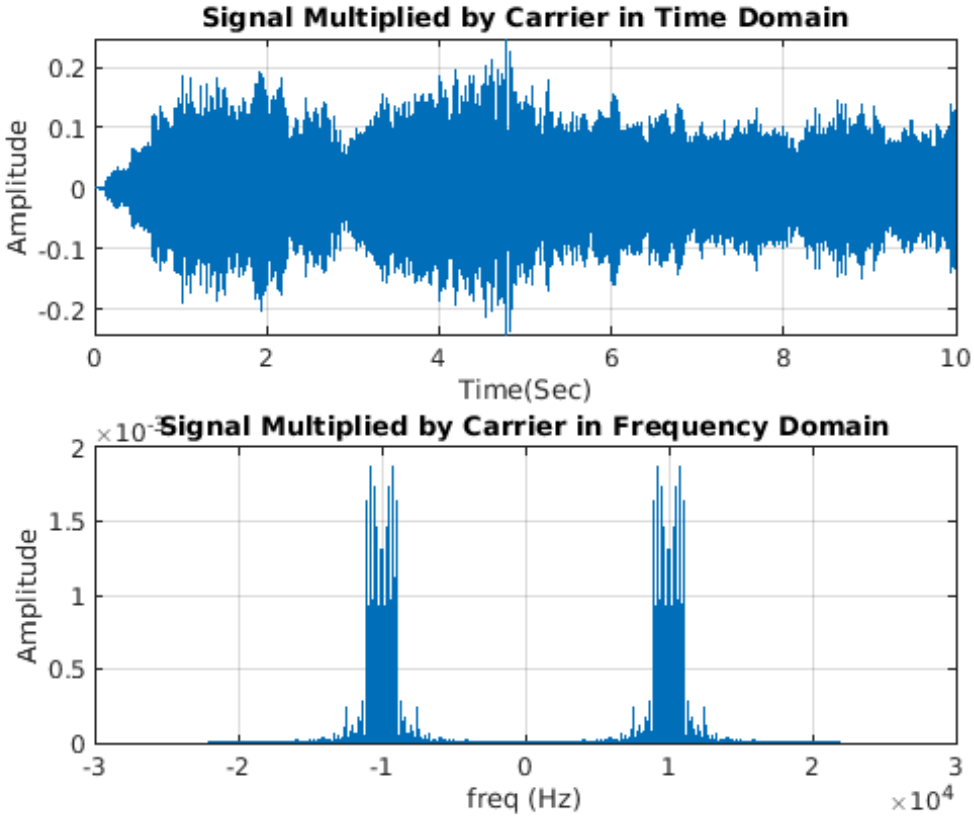

Phase
Phase (degrees)
Normalized Frequency (×π rad/sample)
0
-1000
-2000
-3000
-4000
0
0.2
0.4
0.6
0.8
1



2.3.c

```
f0 = 10000;
n   = 1:length(x);
s_n = 2*cos(2*pi*f0*n/Fs);

y_1 = y_0 .* s_n';
y_2 = filter(Num,1,y_1); % Scrambled Signal

% Plotting signals in time and freq domain
plot_time_freq(y_1, Fs, 'Signal Multiplied by Carrier', 'Signal Multiplied by
 Carrier in Time Domain', 'Signal Multiplied by Carrier in Frequency Domain');
plot_time_freq(y_2, Fs, 'Scrmabled Audio Signal', 'Scrmabled Audio Signal in
 Time Domain', 'Scrmabled Audio Signal in Frequency Domain');

% Playing the audio
sound(y_2, Fs);
pause(length(y_2) * (1/ Fs));
```


Scrmabled Audio Signal in Time Domain
Amplitude
Time(Sec)
Scrmabled Audio Signal in Frequency Domain
×10^-3
Amplitude
freq (Hz)
×10^4



2.3.d

```
y_3 = filter(Num,1,y_2);
y_4 = y_3 .* s_n';
y_5 = filter(Num,1,y_4);

% Plotting signals in time and freq domain
plot_time_freq(y_4, Fs, 'Scrmabled Signal Multiplied by Carrier', 'Scrmabled
 Signal Multiplied by Carrier in Time Domain', 'Scrmabled Signal Multiplied by
 Carrier in Frequency Domain');
plot_time_freq(y_5, Fs, 'Descrmabled Audio Signal', 'Descrmabled Audio Signal
 in Time Domain', 'Descrmabled Audio Signal in Frequency Domain');

plot_freq_freq(x, y_5, Fs, 'Frequency Spectrums', 'Frequency Spectrum Of
 Original Signal', 'Frequency Spectrum Of Descrambled Signal');

% Playing the audio
sound(y_5, Fs);

% Calculating MSE and MAE
disp('MAE and MSE:');
MAE = mean(abs(x - y_5))
MSE = mean((x - y_5) .^ 2)

pause(length(y_2) * (1/ Fs));
```

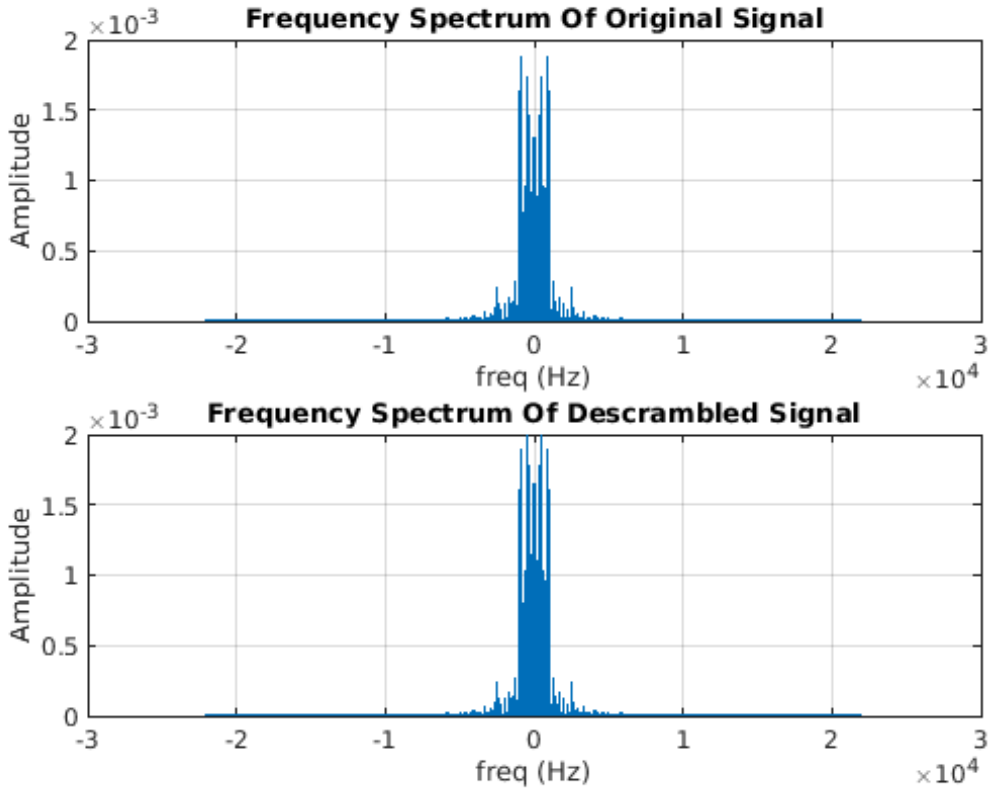
MAE and MSE:

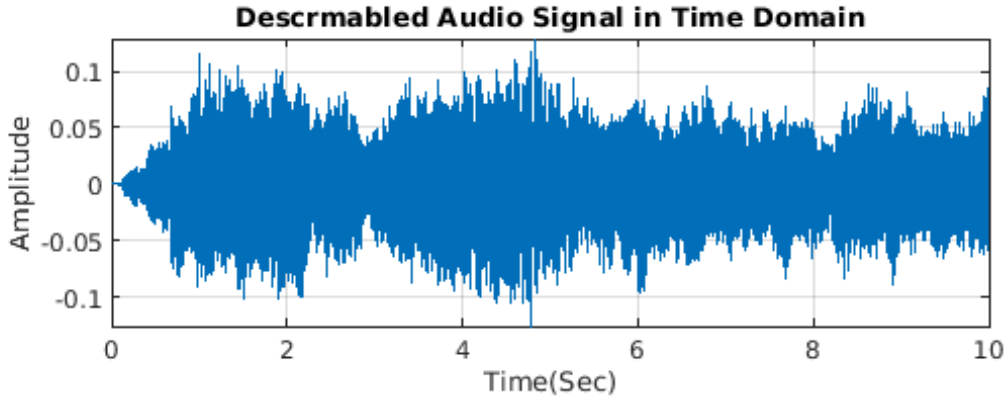
MAE =

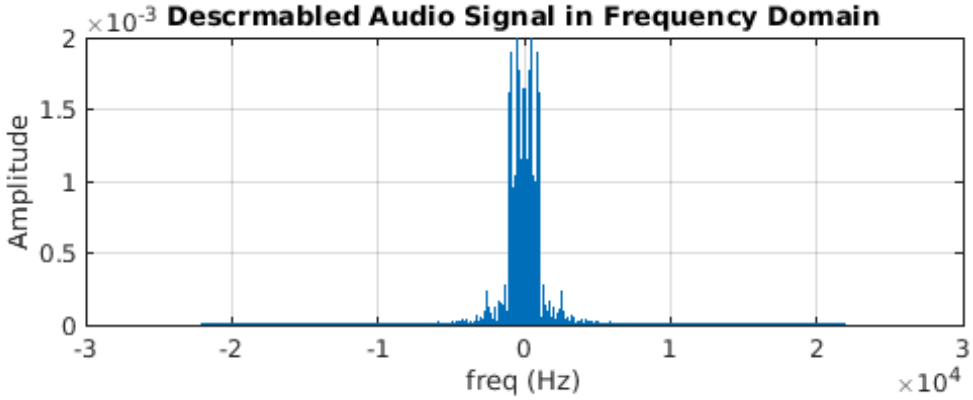
0.0311

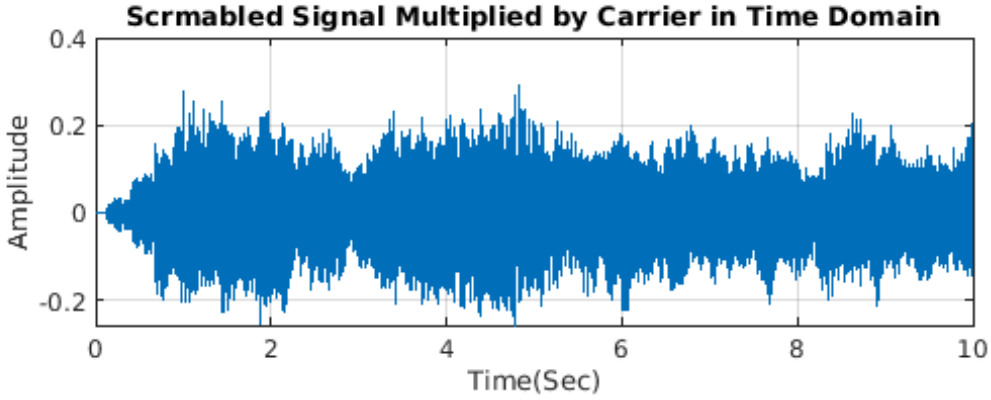
MSE =

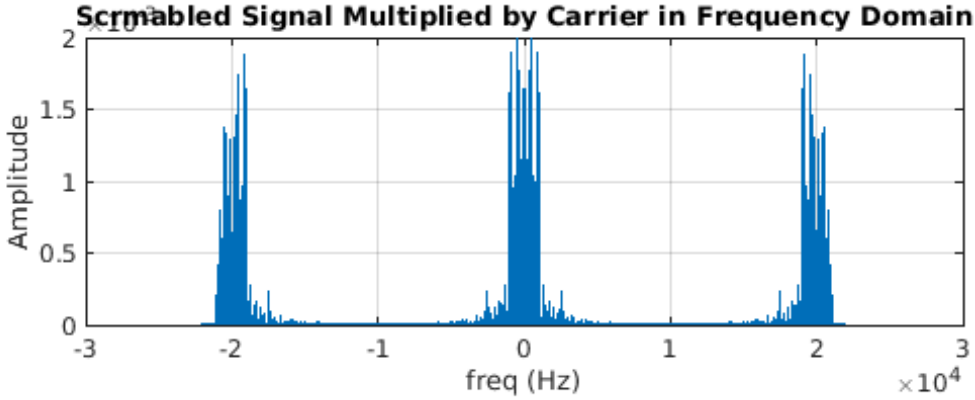
0.0017




Descrmabled Audio Signal in Time Domain
Amplitude
Time(Sec)

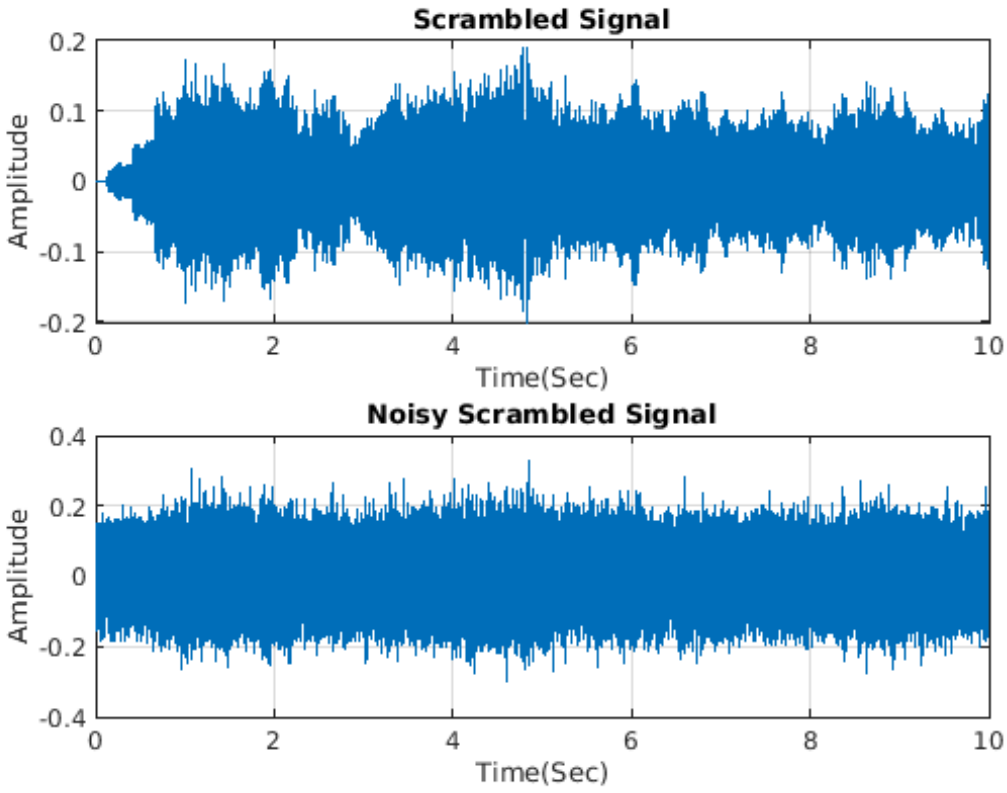

Descrmabled Audio Signal in Frequency Domain
Amplitude
freq (Hz)


Scrmabled Signal Multiplied by Carrier in Time Domain
Amplitude
Time(Sec)


Scrmabled Signal Multiplied by Carrier in Frequency Domain
Amplitude
freq (Hz)

2.3 (with noise)

```
y_2_noisy = y_2 + 0.05 * randn(1, length(y_2))';

T = length(y_2) * (1/ Fs);
t = 0: 1/Fs:T-1/Fs;

figure('Name', 'Scrambled Signal VS Noisy Scrambled Signal');
subplot(2, 1, 1);
plot(t, y_2, 'LineWidth', 1.5);
grid on;
xlabel('Time(Sec)') ;
ylabel('Amplitude') ;
title('Scrambled Signal');
subplot(2,1,2) ;
plot(t, y_2_noisy, 'LineWidth',1.5) ;
grid on;
xlabel('Time(Sec)') ;
ylabel('Amplitude');
title('Noisy Scrambled Signal');


y_3 = filter(Num,1,y_2_noisy);
y_4 = y_3 .* s_n';
y_5 = filter(Num,1,y_4);

% Plotting signals in time and freq domain
plot_time_freq(y_3, Fs, 'Low-Passed Scrmabled Noisy Signal', 'Low-Passed
 Scrmabled Noisy Signal in Time Domain', 'Low-Passed Scrmabled Noisy Signal in
 Frequency Domain');
plot_time_freq(y_4, Fs, 'Noisy Scrmabled Signal Multiplied by Carrier', 'Noisy
 Scrmabled Signal Multiplied by Carrier in Time Domain', 'Noisy Scrmabled
 Signal Multiplied by Carrier in Frequency Domain');
plot_time_freq(y_5, Fs, 'Noisy Descrmabled Audio Signal', 'Noisy Descrmabled
 Audio Signal in Time Domain', 'Noisy Descrmabled Audio Signal in Frequency
 Domain');

plot_freq_freq(x, y_5, Fs, 'Frequency Spectrums (Channel with
 noise)', 'Frequency Spectrum Of Original Signal', 'Frequency Spectrum Of
 Descrambled Signal');
% When the communication channel has noise, a noise floor is added to the
% signal which remains after descrambling and is audible in the final result.
% The noise floor can also be seen in the spectrum of the descrambled signal.

% Playing the audio
sound(y_5, Fs);

% Calculating MSE and MAE
disp('MAE and MSE for noisy signal:');
MAE = mean(abs(x - y_5))
MSE = mean((x - y_5) .^ 2)
```



Plot func

```
function plot_time_freq(y, Fs, title1, title2, title3)
    T = length(y) * (1/ Fs);
    t = 0: 1/Fs:T-1/Fs;

    % Constructing the FFT spectrum of the signal
    L_y = length(y);
    f_y = (Fs/L_y) * (-L_y/2:L_y/2-1);
    fft_y = fftshift(fft(y))/L_y;

    % Plotting signal in time and freq domain
    figure('Name', title1);
    subplot(2, 1, 1);
    plot(t, y, 'LineWidth', 1.5);
    grid on;
    xlabel('Time(Sec)') ;
    ylabel('Amplitude') ;
    title(title2);
    subplot(2,1,2) ;
    plot(f_y, abs(fft_y), 'LineWidth',1.5) ;
    grid on;
    xlabel('freq (Hz)');
    ylabel('Amplitude');
    title(title3);
end
```

```
function plot_freq_freq(y1, y2, Fs, title1, title2, title3)
    % Constructing the FFT spectrum of the signals
    L_y1 = length(y1);
    f_y1 = (Fs/L_y1) * (-L_y1/2:L_y1/2-1);
    fft_y1 = fftshift(fft(y1))/L_y1;

    L_y2 = length(y2);
    f_y2 = (Fs/L_y2) * (-L_y2/2:L_y2/2-1);
    fft_y2 = fftshift(fft(y2))/L_y2;

    % Plotting signal in time and freq domain
    figure('Name', title1);
    subplot(2, 1, 1);
    plot(f_y1, abs(fft_y1), 'LineWidth', 1.5);
    grid on;
    xlabel('freq (Hz)') ;
    ylabel('Amplitude') ;
    title(title2);
    subplot(2,1,2) ;
    plot(f_y2, abs(fft_y2), 'LineWidth',1.5) ;
    grid on;
    xlabel('freq (Hz)');
    ylabel('Amplitude');
    title(title3);
end
```

Published with MATLAB® R2023a