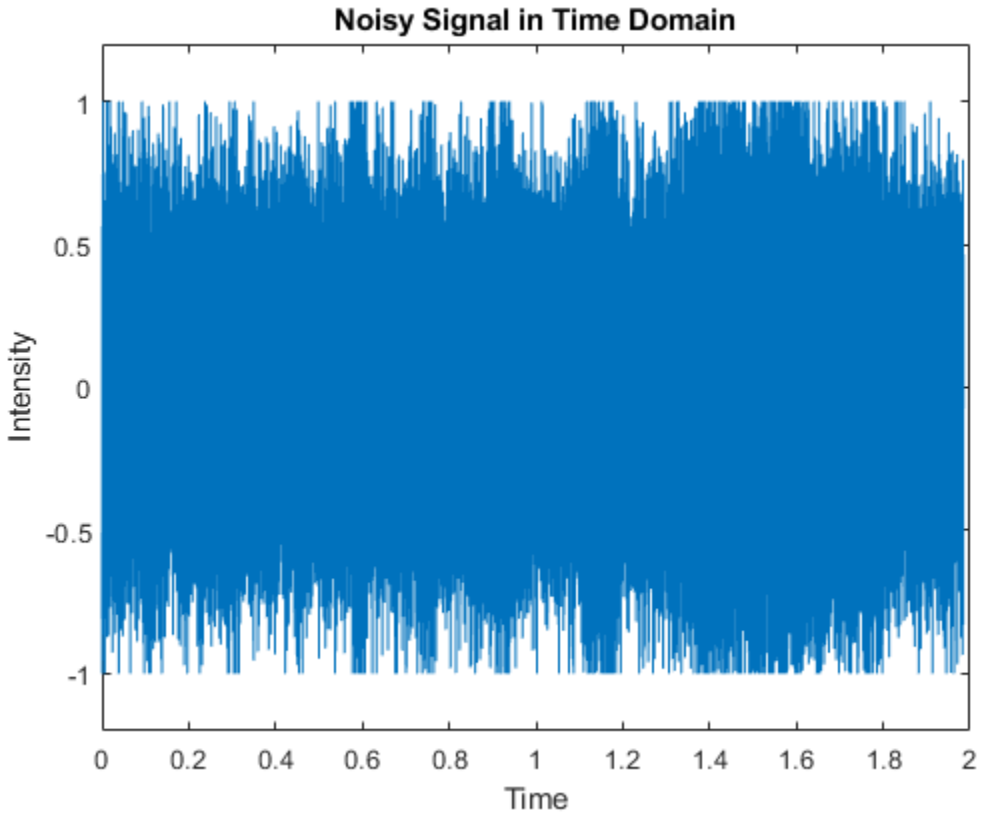
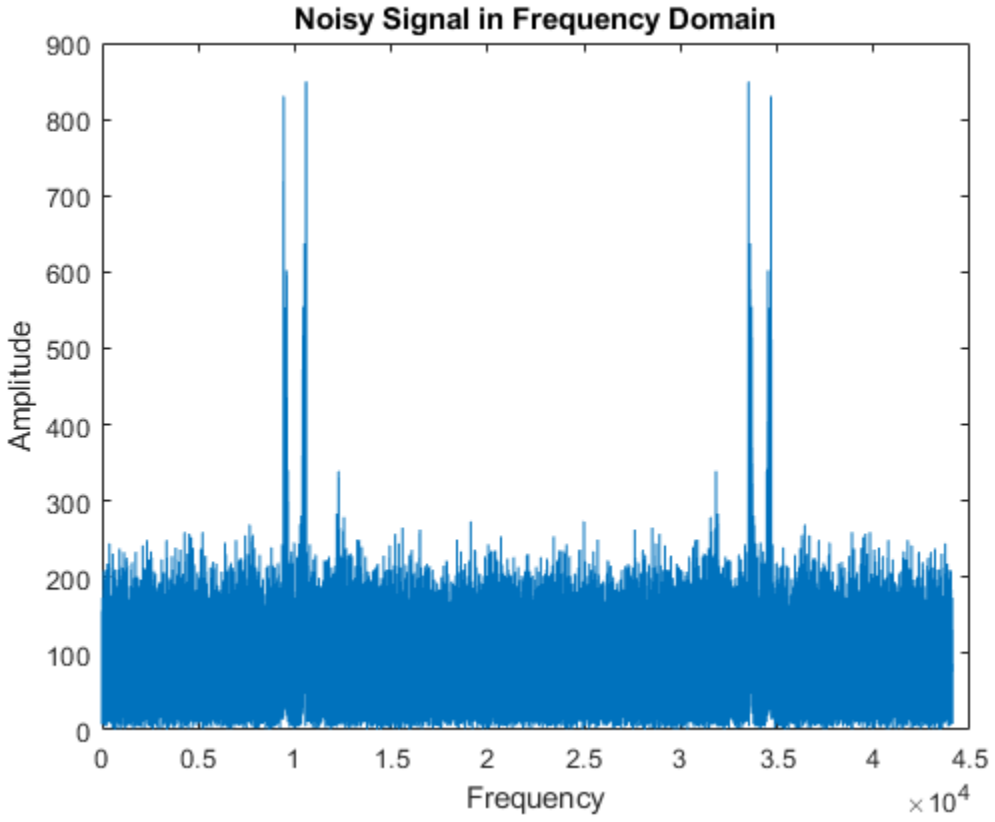
```
clc; close all; clear all;

% Initial Unfiltered Signal in time domain
[y, fs] = audioread("modulated_noisy_audio.wav");
t = linspace(0, length(y)/fs, length(y));

fig = figure;
plot(t, y);
xlabel('Time')
ylabel('Intensity')
ylim([-1.2 1.2])
title('Noisy Signal in Time Domain')
saveas(fig, 'png/Noisy Signal in Time Domain.png')
```



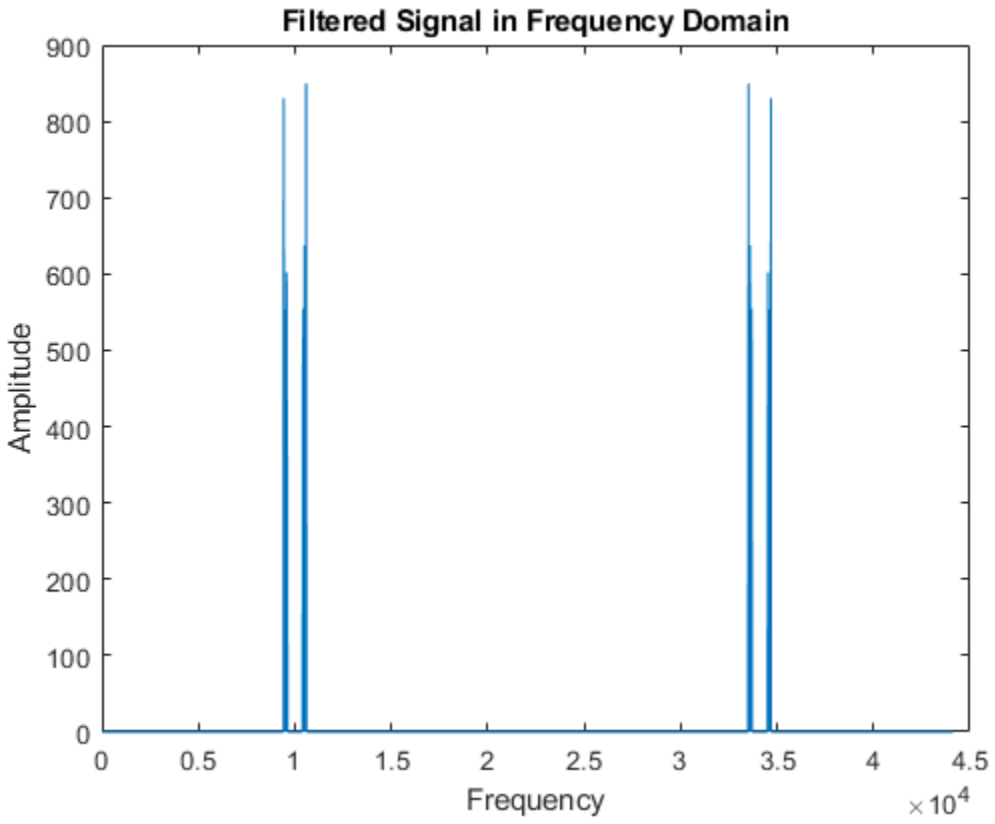
Frequency Domain

```
f = linspace(0, fs, length(y));
fourier_t = fft(y);
A = abs(fourier_t);

fig = figure;
plot(f, A);
xlabel('Frequency')
ylabel('Amplitude')
title('Noisy Signal in Frequency Domain')
saveas(fig, 'png/Noisy Signal in Frequency Domain.png')
```



Noise Filtering

```
min_A = max(A)/2;
denoise_fourier_t = fourier_t .* (A>min_A);
denoise_A = abs(denoise_fourier_t);

fig = figure;
plot(f, denoise_A);
xlabel('Frequency')
ylabel('Amplitude')
title('Filtered Signal in Frequency Domain')
saveas(fig, 'png/Filtered Signal in Frequency Domain.png')
```



Calculating Carrier and Message Frequencies

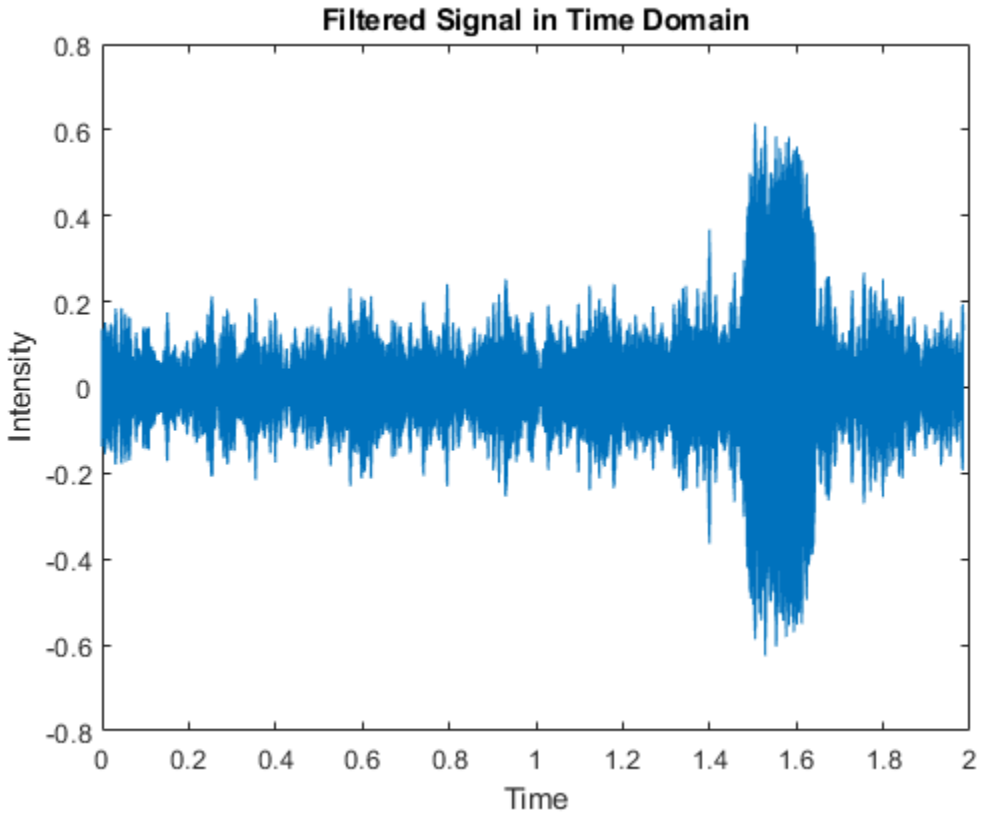
```
[pks, indices] = findpeaks(denoise_A(1:round(length(y)/2)));
[pks, I] = sort(pks, "descend");
indices = indices(I);

looper = 2;
while abs(indices(looper) - indices(1)) < 2200
    looper = looper+1;
end

a = max([indices(1) indices(looper)]); % index for larger frequency
b = min([indices(1) indices(looper)]); % index for lower frequency

fc = (f(a) + f(b))/2; % Carrier Frequency
fm = (f(a) - f(b))/2; % Message Frequency

fprintf('Carrier frequency is %fHz', fc);
```

Carrier frequency is 9999.805227Hz

Back to time domain

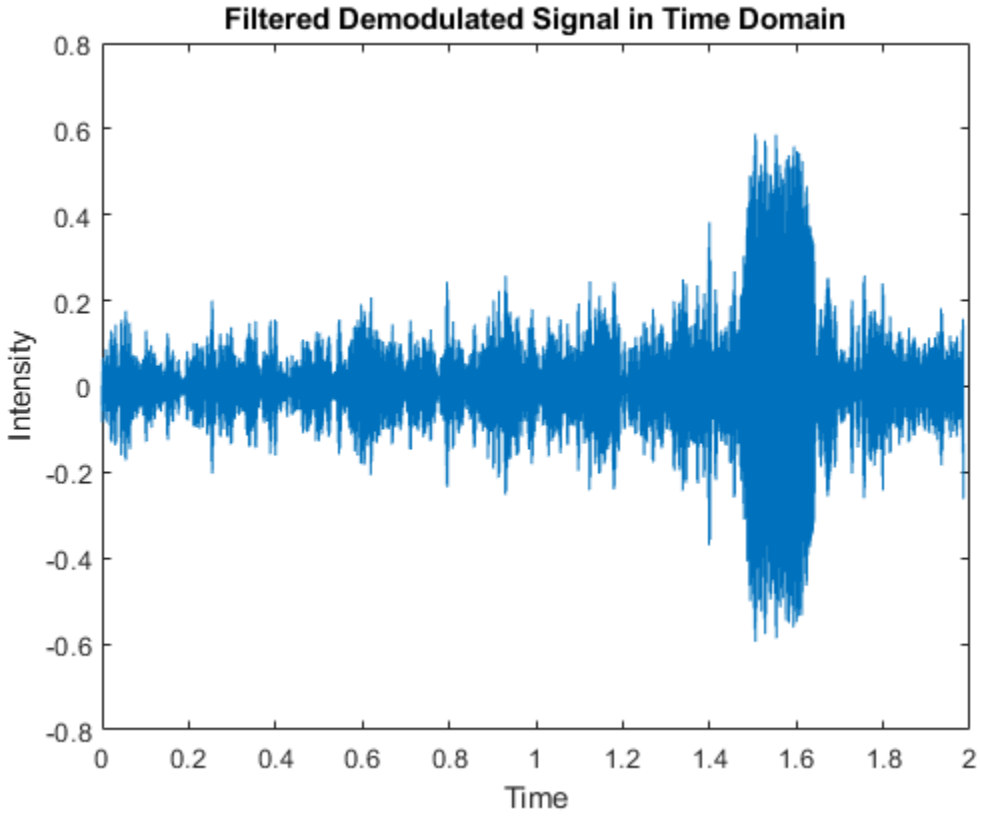
```
denoise_y = ifft(denoise_fourier_t);

fig = figure;
plot(t, denoise_y);
```

```
xlabel('Time')
ylabel('Intensity')
ylim([-0.8 0.8])
title('Filtered Signal in Time Domain')
saveas(fig, 'png/Filtered Signal in Time Domain.png')
```



Demodulation

```
denoise_demod_y = amdemod(denoise_y, fc, fs);
fig = figure;
plot(t, denoise_demod_y);
xlabel('Time')
ylabel('Intensity')
ylim([-0.8 0.8])
title('Filtered Demodulated Signal in Time Domain')
saveas(fig, 'png/Filtered Demodulated Signal in Time Domain.png')
```



Save to file

```
audiowrite('demodulated_filtered_audio.wav', denoise_demod_y, fs);
```

Published with MATLAB® R2024b