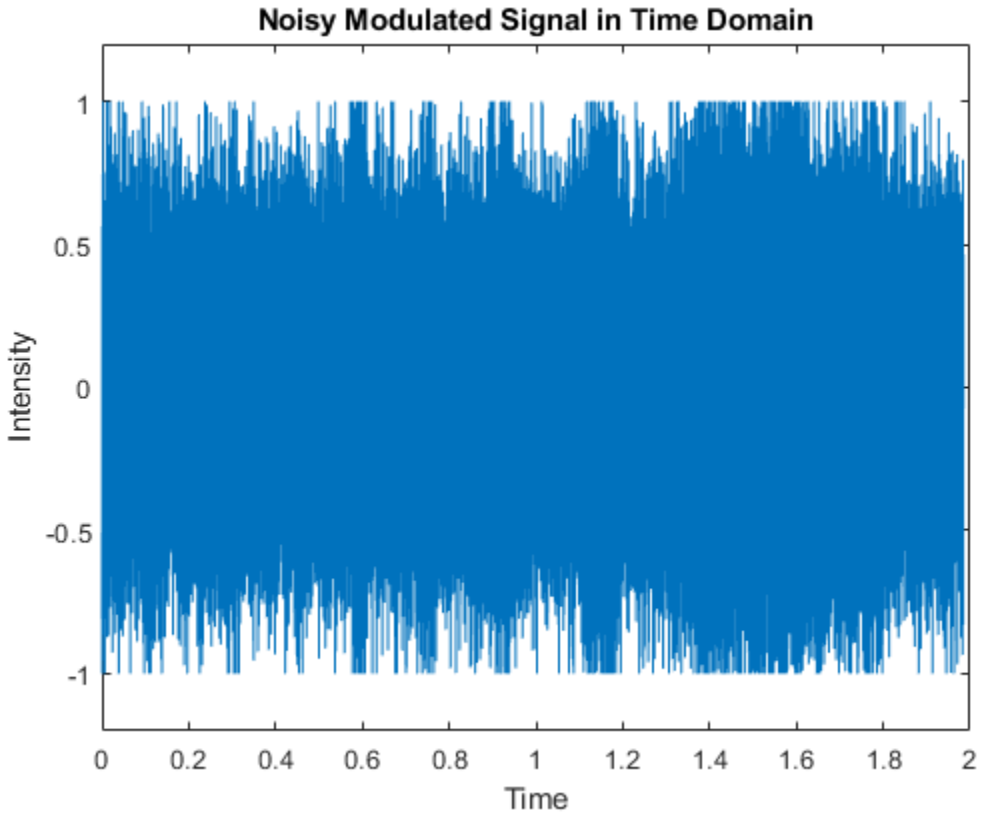
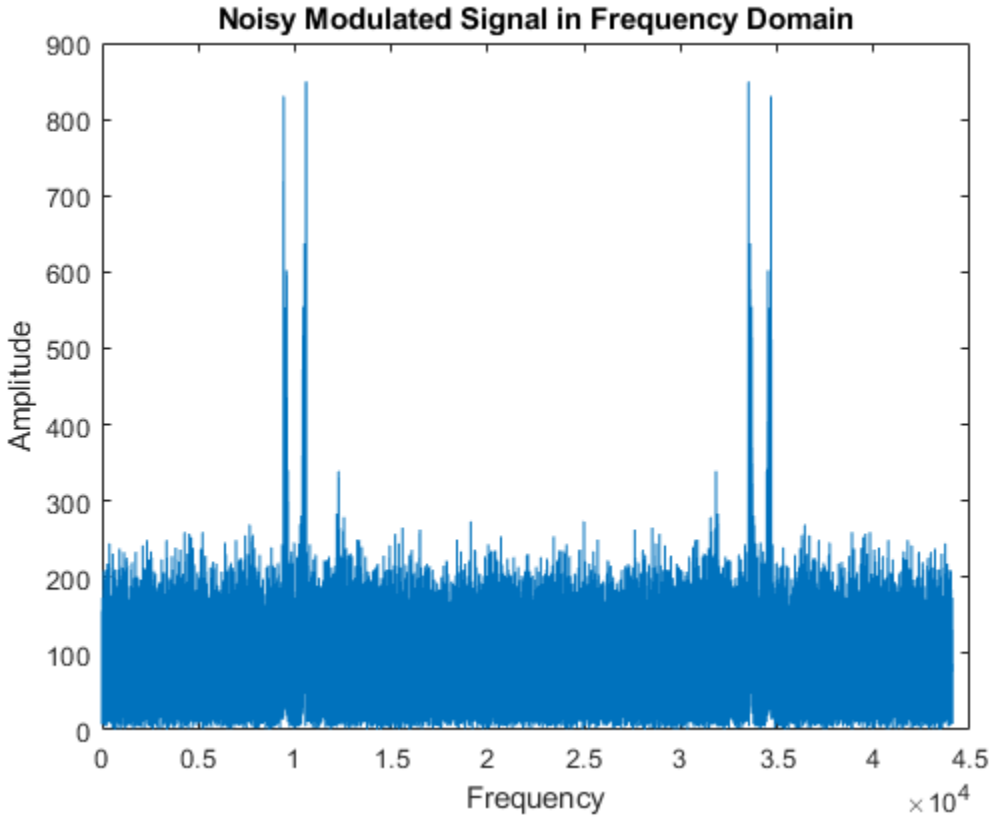
```
clc; close all; clear all;

% 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 Modulated Signal in Time Domain')
saveas(fig, 'png/Noisy Modulated Signal in Time Domain.png')
```



Frequency Domain

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

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



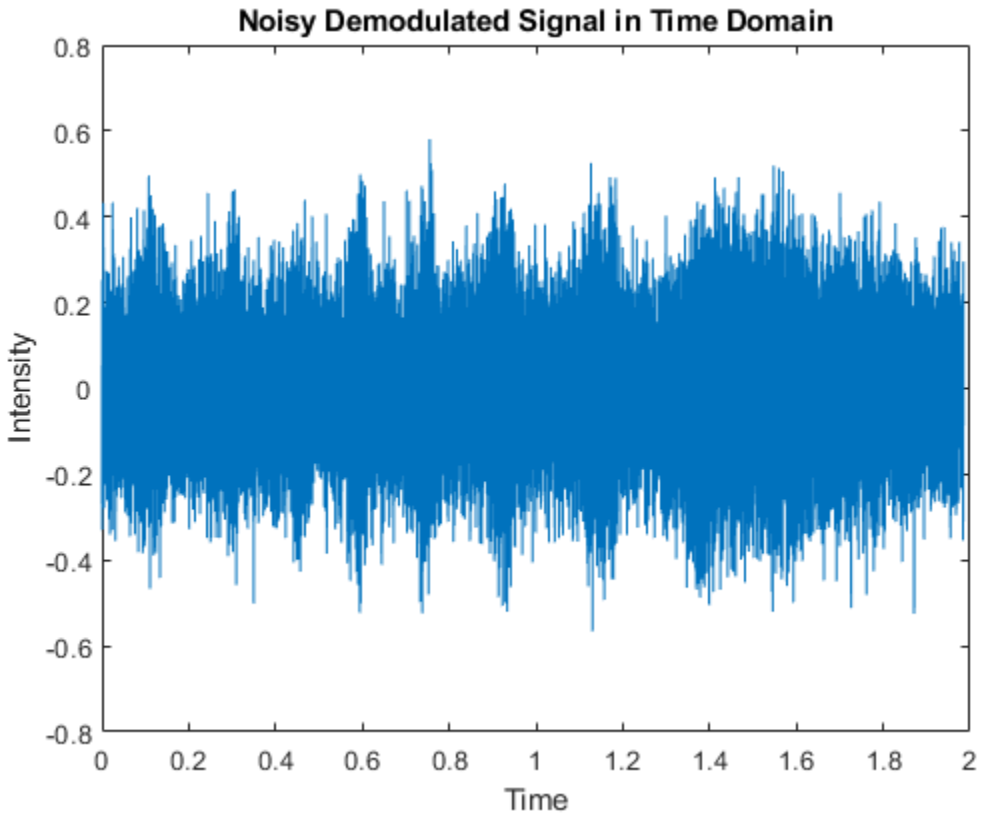
Calculating Carrier and Message Frequencies

```
[pks, indices] = findpeaks(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
```

Demodulation

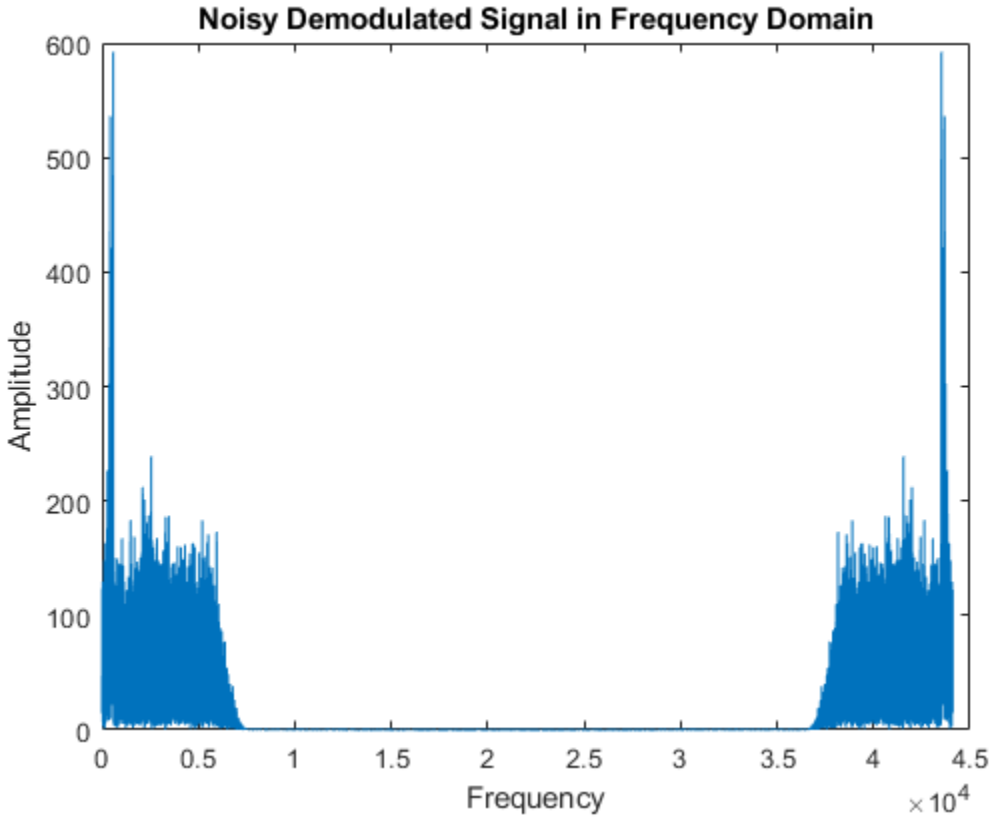
```
carrier = sin(2*pi*fc*t);
square = y .* carrier';

demod_y = lowpass(square, fc/2, fs);
```

```
fig = figure;
plot(t, demod_y);
xlabel('Time')
ylabel('Intensity')
ylim([-0.8 0.8])
title('Noisy Demodulated Signal in Time Domain')
saveas(fig, 'png/Noisy Demodulated Signal in Time Domain.png')
```



Back to Frequency Domain

```
demod_A = abs(fft(demod_y));

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

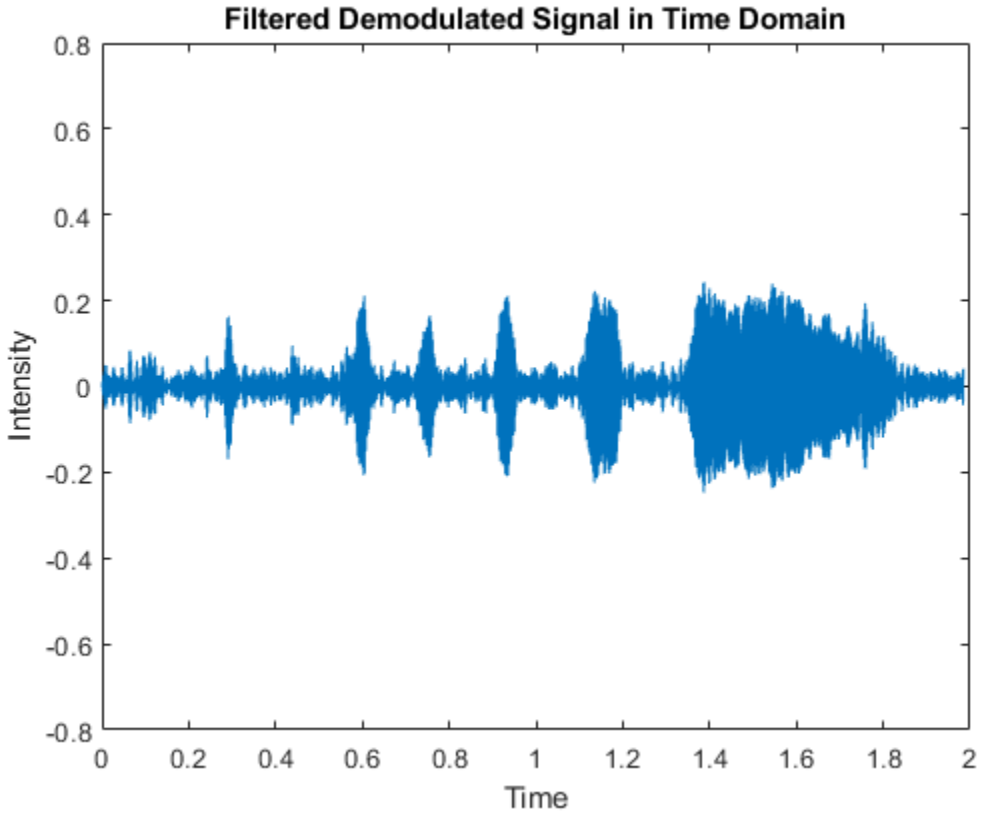


Finding bandwidth

```
Q = sqrt(3); % Qaulity Factor
[max_amp, max_amp_index] = max(demod_A(1:round(length(y)/2)));
new_arr = abs(demod_A - max_amp/Q); % new array to find f_L and f_H
[minL, min_lower_index] = min(new_arr(1:max_amp_index));
[minH, min_higher_index] = min(new_arr(max_amp_index:round(length(y)/2)));
fl = f(min_lower_index);
fh = f(max_amp_index+min_higher_index-1);
```

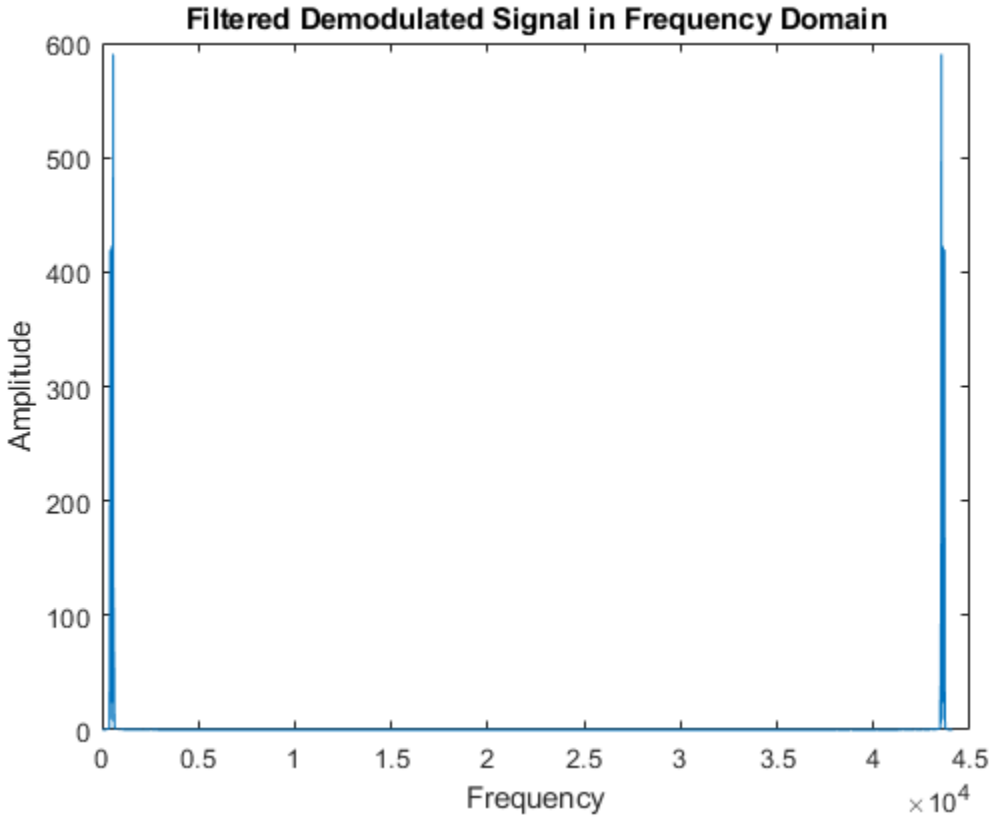
Filtering

```
demod_denoise_y = bandpass(demod_y, [fl, fh], fs);

fig = figure;
plot(t, demod_denoise_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')
```



Frequency Domain, just for fun

```
demod_denoise_amp = abs(fft(demod_denoise_y));

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



Save to file

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

Published with MATLAB® R2024b