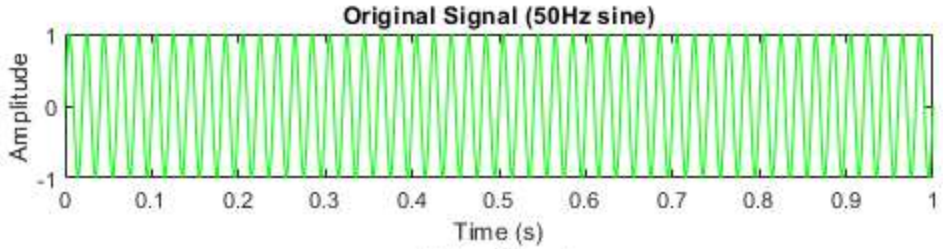
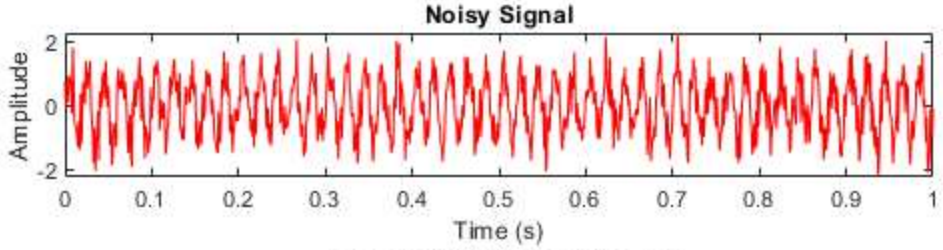
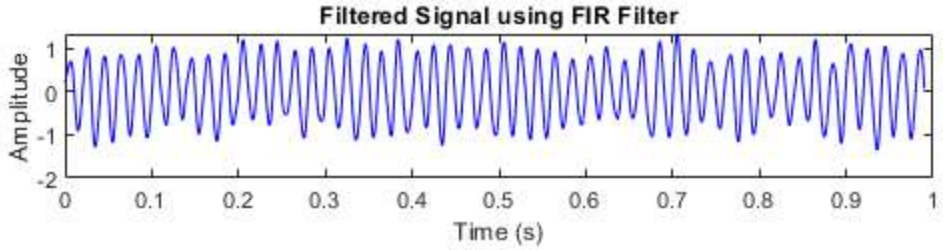
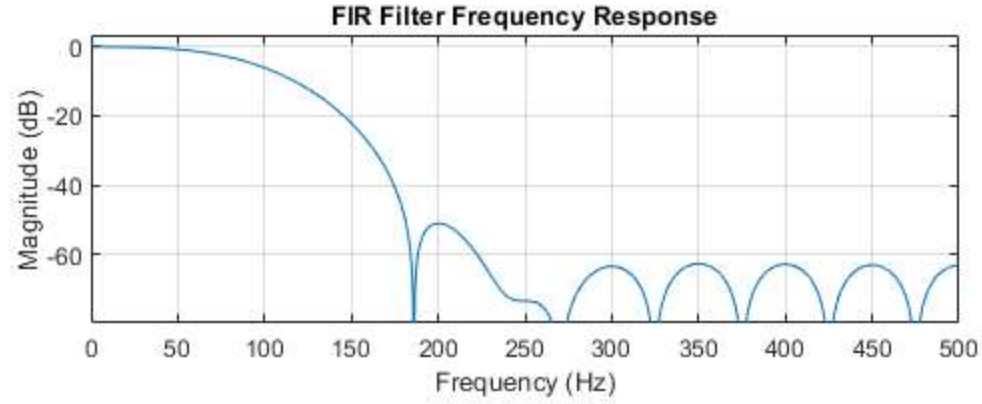
```
clc;
clear;

N = 21;                 % Filter Order
Fc = 0.2;               % Normalized Cutoff Frequency
Fs = 1000;              % Sampling Frequency (Hz)

h = fir1(N-1, Fc);      % FIR filter coefficients using Hamming window

t = 0:1/Fs:1;                         % Time vector (1 second)
x = sin(2*pi*50*t);                  % 50 Hz sine wave (original signal)
x_noisy = x + 0.5*randn(size(t));   % Add Gaussian noise

y = filter(h, 1, x_noisy);          % Apply FIR filter
delay = (N - 1) / 2;                % FIR group delay in samples
x_trimmed = x(1:end - delay);
x_noisy_trimmed = x_noisy(1:end - delay);
y_aligned = y(delay + 1:end);       % Align filtered output

figure;
subplot(3,1,1);
plot(t, x, 'g');
title('Original Signal (50Hz sine)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,2);
plot(t, x_noisy, 'r');
title('Noisy Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,3);
plot(t(1:end - delay), y_aligned, 'b');
title('Filtered Signal using FIR Filter');
xlabel('Time (s)');
ylabel('Amplitude');

figure;
freqz(h, 1, 1024, Fs);
title('FIR Filter Frequency Response');
snr_before = snr(x_trimmed, x_noisy_trimmed - x_trimmed);
snr_after = snr(x_trimmed, y_aligned - x_trimmed);

fprintf('SNR before filtering: %.2f dB\n', snr_before);
fprintf('SNR after filtering:  %.2f dB\n', snr_after);
```

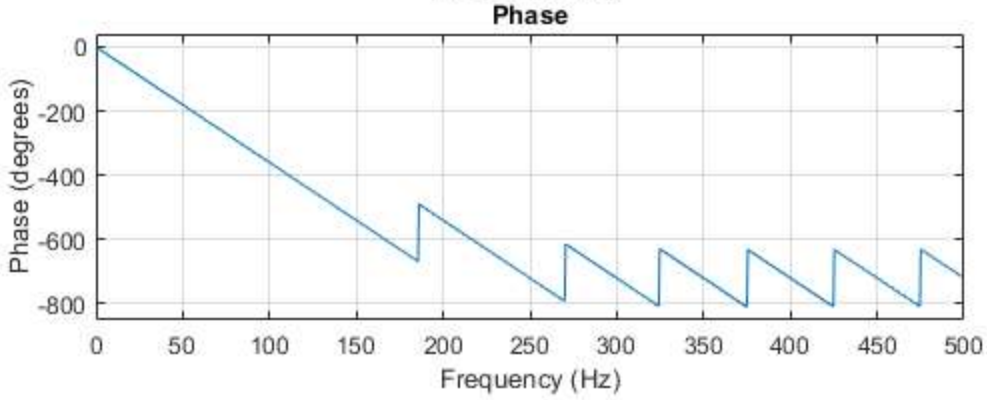
```
SNR before filtering: 3.07 dB
SNR after filtering:  11.05 dB
```











Published with MATLAB® R2024b