

Design and Implementation of FIR and IIR Filters for Noise Removal

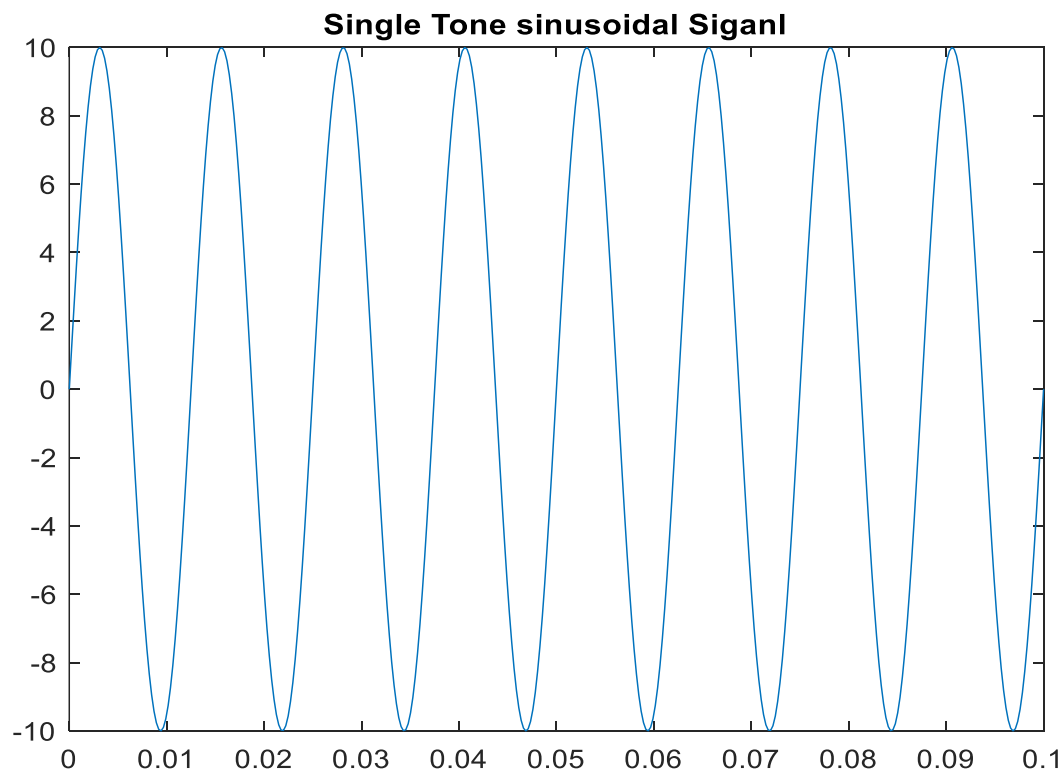
Task:

Write a MATLAB code to remove noises from a single tone sinusoidal signal using FIR and IIR lowpass filters.

Code:

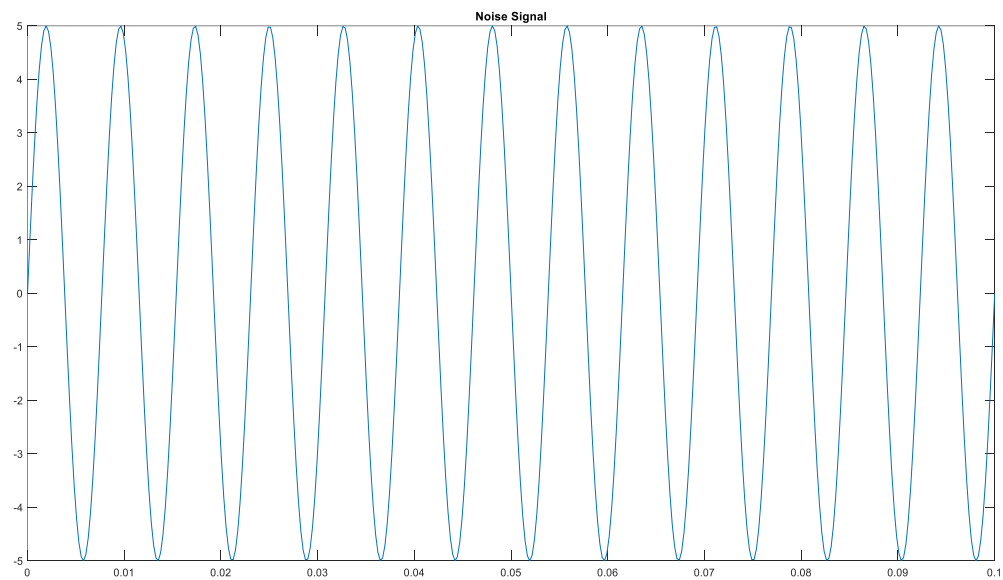
```
clc
clear all
close all
A = 10;
f = 80;
Fs = 4140;
T = 1/f;
n = 0: 1/Fs :0.1;
x = A*sin(2*pi*f*n);
plot(n,x);
title('Single Tone sinusoidal Siganl')
```

Output:



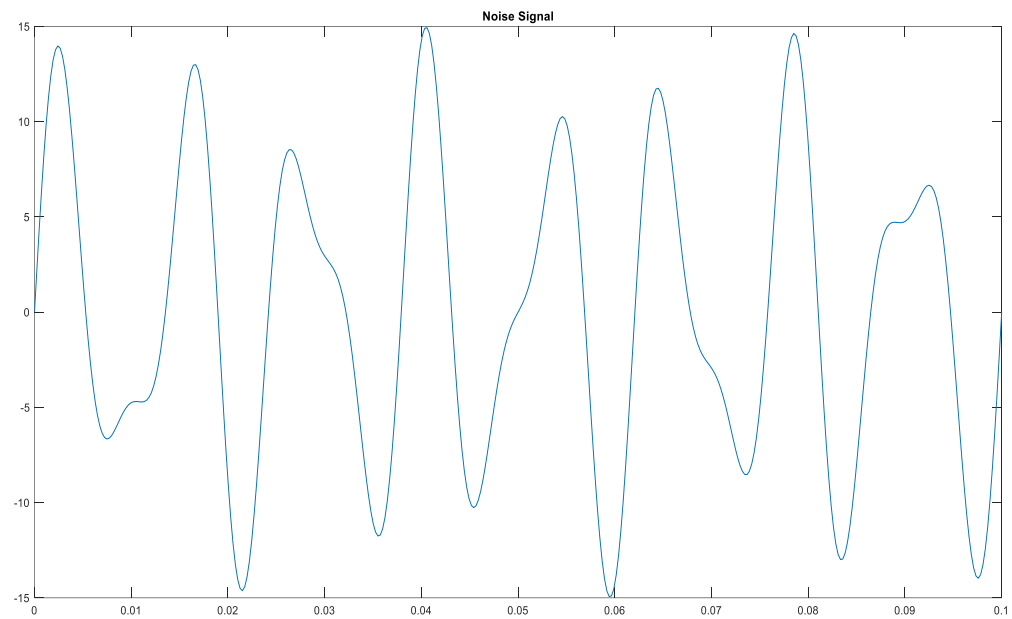
```
%Noise
An = A/2;
fn = f + 50;
y = An* sin(2*pi*fn*n);
plot(n,y);
title('Noise Signal')
```

Output:

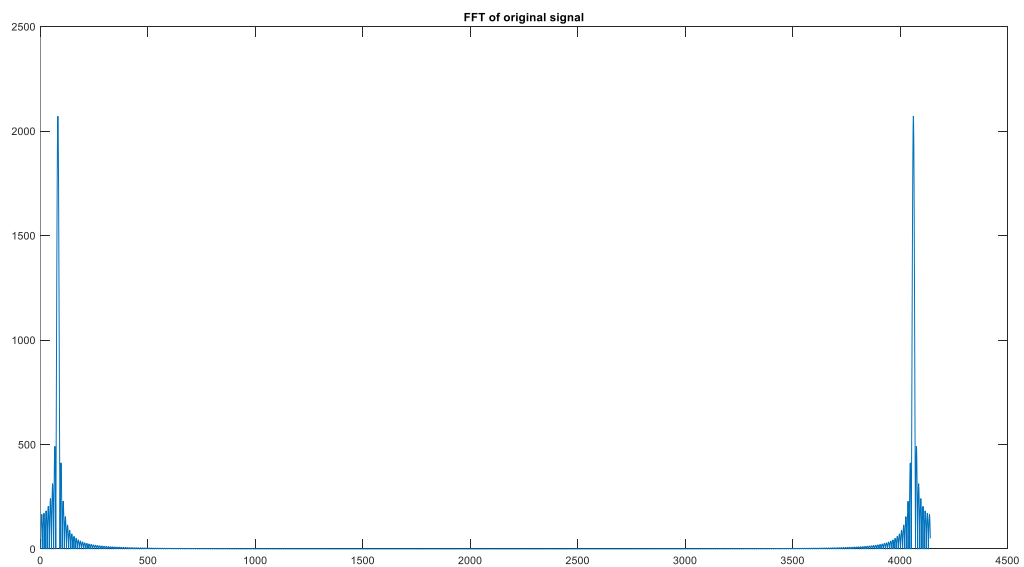


```
% Adding noise signal
z = x + y;
plot(n,z)
title('Noise Signal')
```

Output:

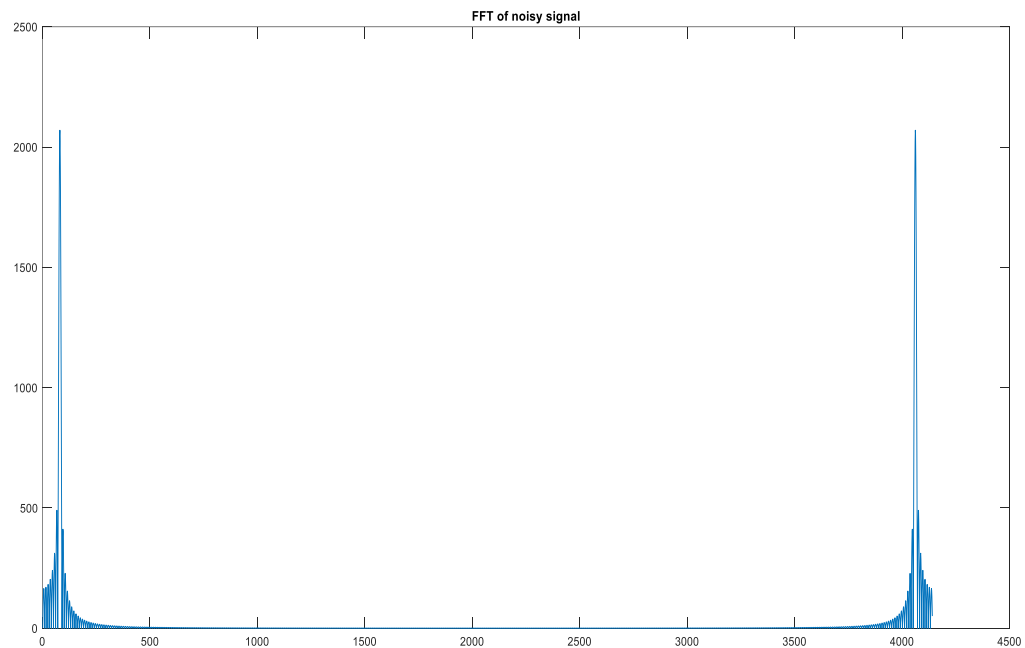


```
% plotting original signal in frequency domain  
plot(abs(fft(x,4140)));  
title('FFT of original signal')
```



```
% plotting noisy signal in frequency domain
plot(abs(fft(x,4140)));
title('FFT of noisy signal')
```

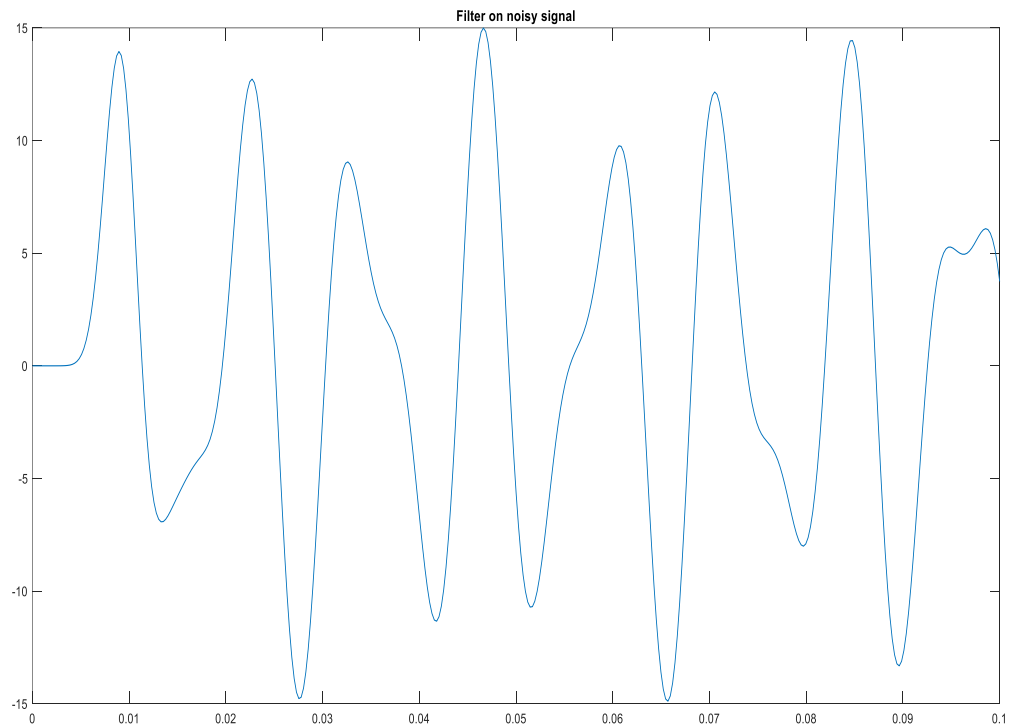
Output:



```
%IIR Butterworth Filter
O_IIR =15; % order of filter
wc = (2*pi*(f/Fs)); % normalized or digital cut-off
frequency
[b,a] = butter(O_IIR,wc,'Low'); %[b,a] neumarator and
denumarator

x_f_iir = filter(b,a,z);
plot(n,x_f_iir)
title('Filter on noisy signal');
```

Output:

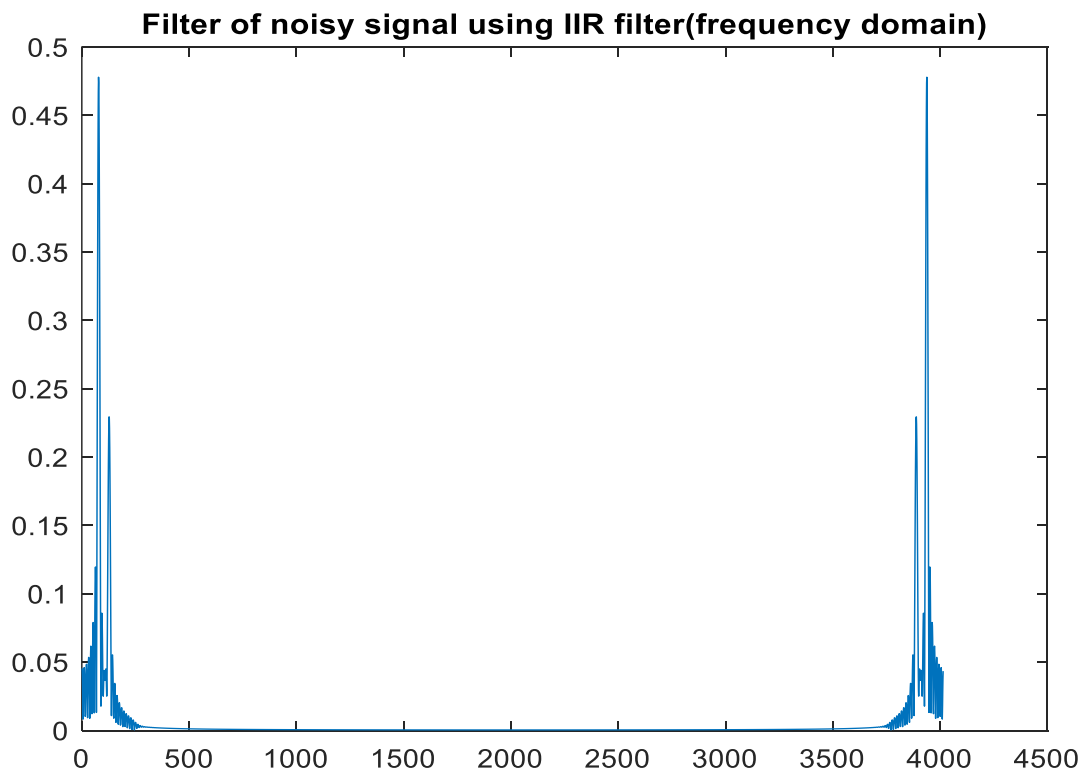


```
%IIR Butterworth Filter
O_IIR =15; % order of filter
wc = (2*pi*(f/Fs)); % normalized or digital cut-off
frequency
[b,a] = butter(O_IIR,wc,'Low'); %[b,a] neumarator and
denumarator

x_f_IIR = filter(b,a,z);

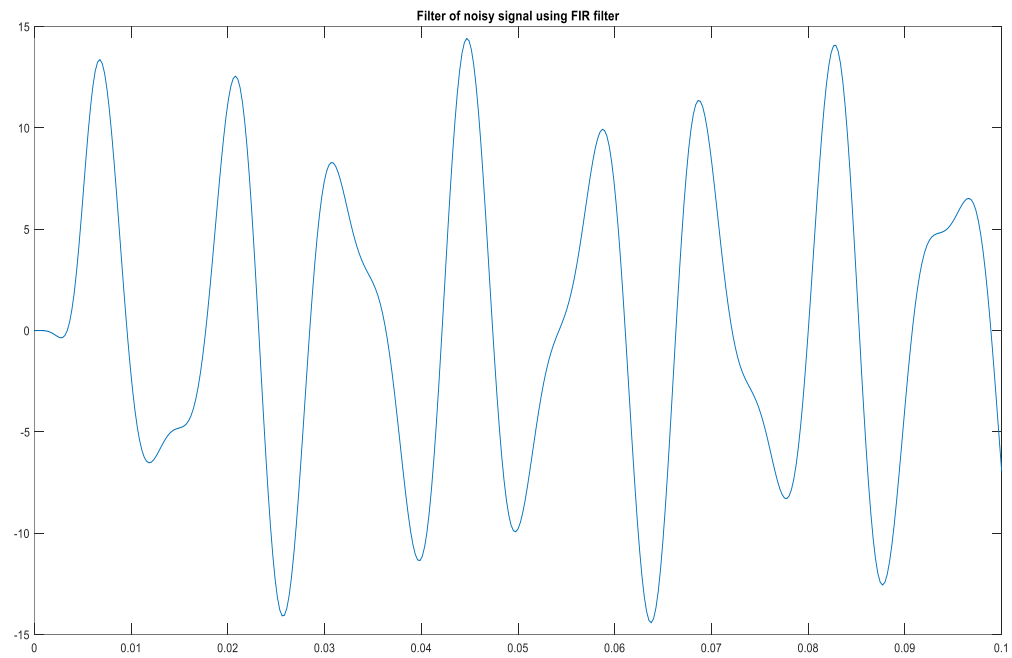
% IIR FILTER IN FREQUENCY DOMAIN
plot(abs(ifft(x_f_IIR, 4014)))
title('Filter of noisy signal using IIR filter(frequency
domain)');
```

Output:



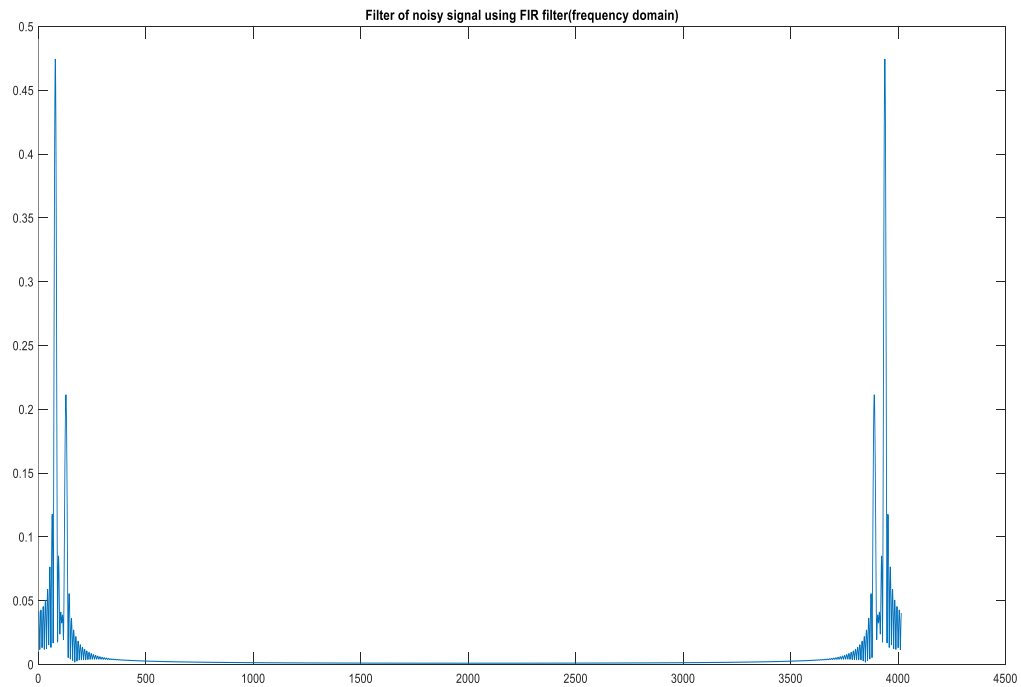
```
% FIR FILTER
O_FIR = 35;
wc = 2*pi*f/Fs;
b = fir1(O_FIR,wc,'low');
x_f_FIR = filter(b,1,z);
plot(n,x_f_FIR)
title('Filter of noisy signal using FIR filter');
```

Output:



```
% FIR FILTER IN FREQUENCY DOMAIN
plot(abs(iff(x_f_FIR, 4014)))
title('Filter of noisy signal using FIR filter(frequency
domain)');
```

Output:



```
% Changing the filter order
```

```
O_FIR = 50;
```

```
wc = 2*pi*f/Fs;
```

```
b = fir1(O_FIR,wc, 'low');
```

```
x_f_FIR = filter(b,1,z);
```

```
plot(n,x_f_FIR)
```

```
title('Filter of noisy signal using FIR filter');
```

```
% FIR FILTER IN FREQUENCY DOMAIN (changing filter order.)
```

```
plot(abs(ifft(x_f_FIR, 4014)))
```

```
title('Filter of noisy signal using IIR filter(frequency  
domain)');
```


Output:

