Noise Reduction from Sinusoidal Signals Using FDATool

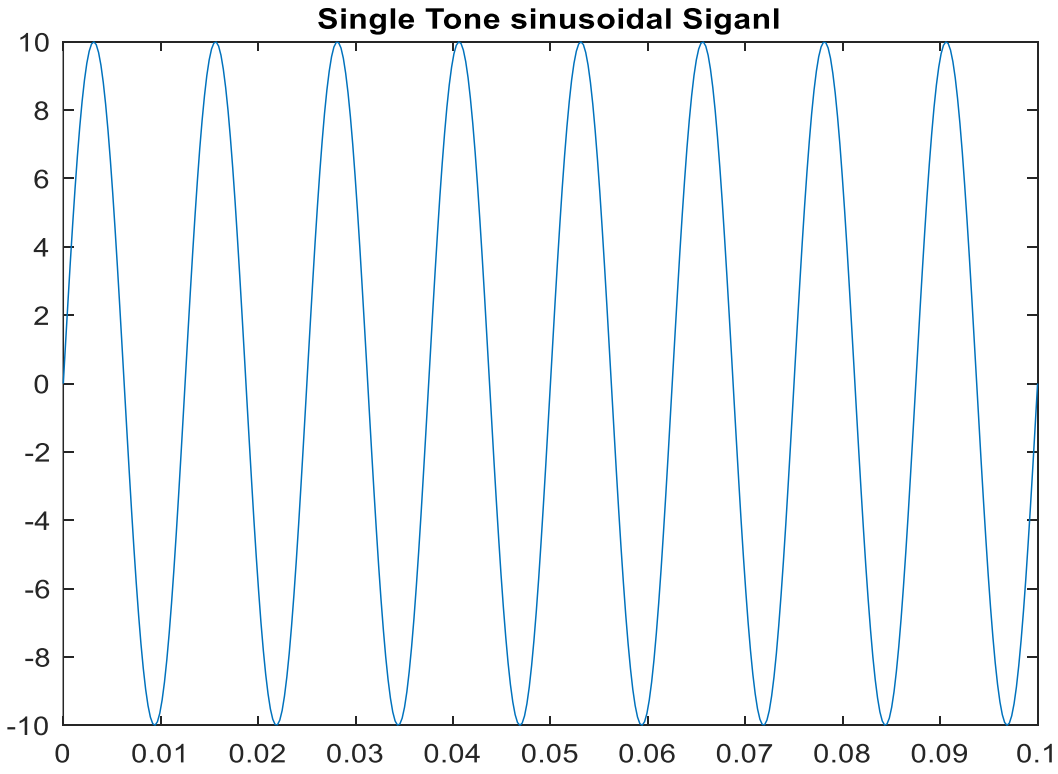
Task

Write a MATLAB code to remove noises from the single tone sinusoidal signal of the previous problem using the filterDesigner tool (FDATool). The noise should be generated using the rand function. Design LPF, HPF and BF filters (both FIR and IIR). Use different filters and select the most effective one.

Code:

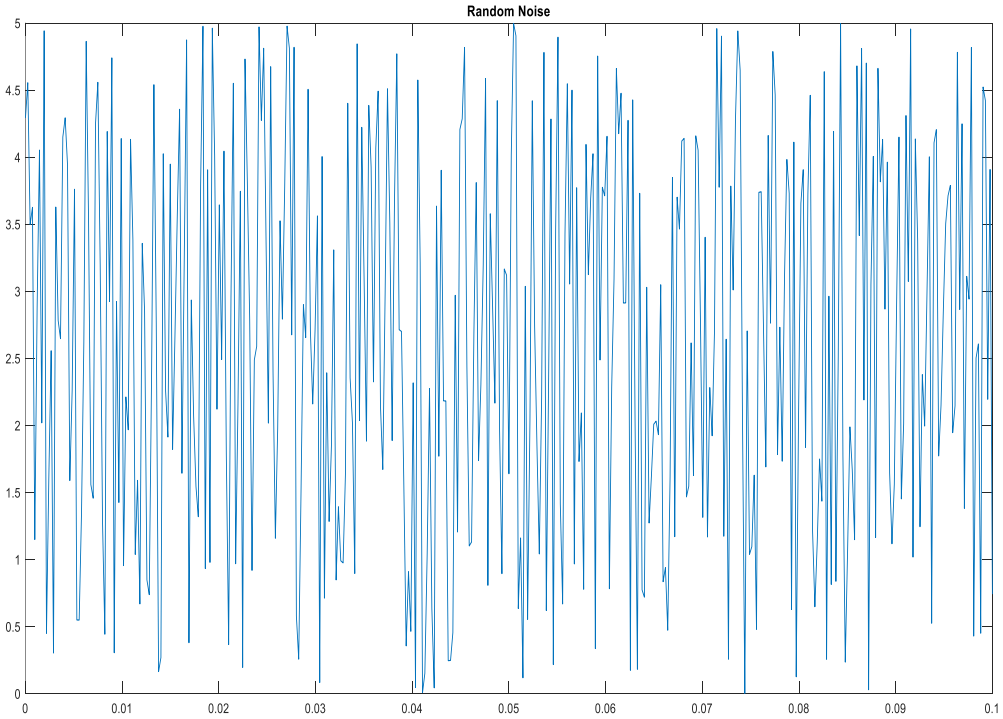
```
clc
clear all
close all
A = 10;
f = 80;
Fs = 4140;
T = 1/f;
n = 0: 1/Fs :0.1;
x = A*sin(2*pi*f*n);
plot(n,x);
title('Single Tone sinusoidal Siganl')
```

Output:



```
%Generating a random noise.
SZ= size(n);
y=5*rand(SZ)
plot(n,y)
title('Random Noise')
```

Output:



Command Window:

y =

Columns 1 through 10

4.2929 4.5553 3.4982 3.6259 1.1494 2.8803 4.0531 2.0192 4.9422 0.4500

Columns 11 through 20

1.6047 2.5570 0.3030 3.6284 2.7828 2.6468 4.1499 4.2938 3.9451 1.5892

Columns 21 through 30

2.2610 3.7611 0.5493 0.5487 1.3494 2.6232 4.8633 3.5520 1.5593 1.4573

Columns 31 through 40

4.2518 4.5582 3.1964 1.2769 0.4433 4.1913 2.9236 4.7405 0.3051 2.9232

Columns 41 through 50

1.4255 4.1387 0.9549 2.2126 1.9671 4.1329 3.3844 1.0380 1.5905 0.6691

Columns 51 through 60

3.3573 2.8550 0.8488 0.7383 2.3804 4.5405 2.7609 0.1647 0.2693 4.0253

Columns 61 through 70

2.2569 1.9132 3.9482 1.8214 2.6617 3.5583 4.3574 1.6434 3.2506 4.8742

Columns 71 through 80

0.3798 2.9351 2.0694 1.5457 1.3192 3.7938 4.9761 0.9329 3.9057 0.9790

Columns 81 through 90

4.9618 4.0113 2.1211 3.6443 2.4918 4.0450 1.7825 0.3662 2.9550 4.5509

Columns 91 through 100

0.9688 2.1618 3.7458 0.1959 4.7316 3.8184 2.7941 0.9192 2.4897 2.5892

Columns 101 through 110

4.9712 4.2743 4.8120 3.3947 2.0175 4.6749 2.3974 1.1590 1.9815 3.5254

Columns 111 through 120

2.7928 3.7832 4.9774 4.8122 2.6753 4.8194 0.5781 0.2572 1.5217 2.9010

Columns 121 through 130

2.6548 4.5060 2.7028 2.1599 2.7133 3.5621 0.0834 4.0046 0.7125 2.3924

Columns 131 through 140

1.2842 1.8455 3.3088 0.8480 1.3939 0.9911 0.9754 1.6342 4.4017 2.3555

Columns 141 through 150

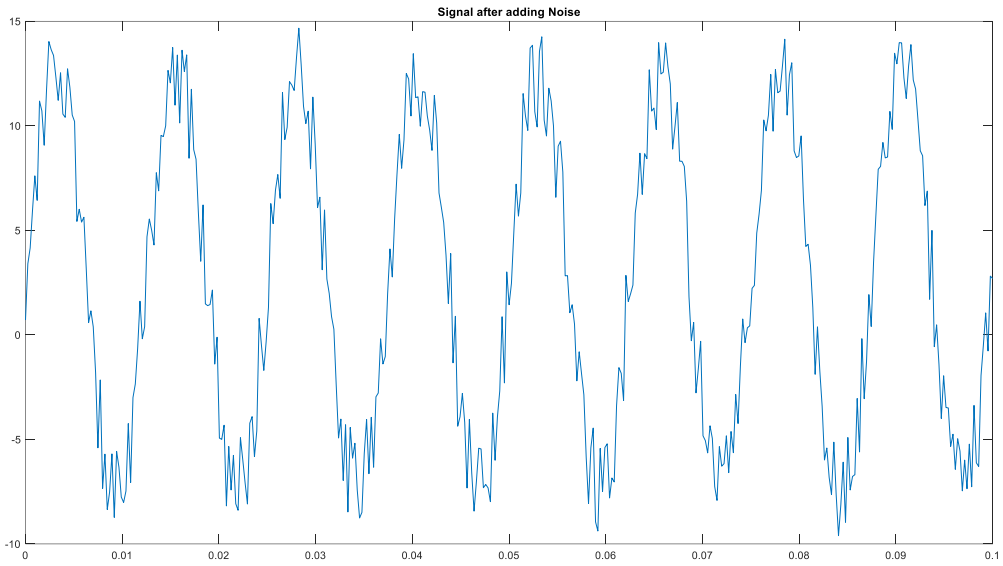
2.0198 0.8962 4.8446 2.0373 4.2224 3.0766 1.8831 4.3859 3.9243 2.3248

Columns 151 through 160

4.0699 4.4922 2.1462 1.6716 2.9832 4.5100 3.5103 1.8873 3.6748 4.7705

```
%Adding noise to the main signal.
z=x+y
plot(n,z)
title('Signal after adding Noise')
```

Output:

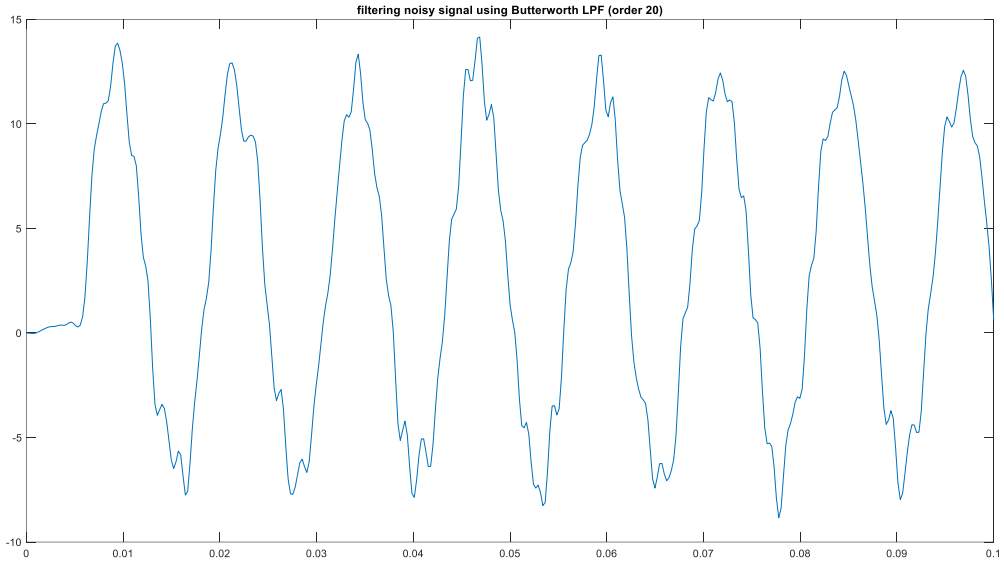


Design Filter Using filter Designer Tool:

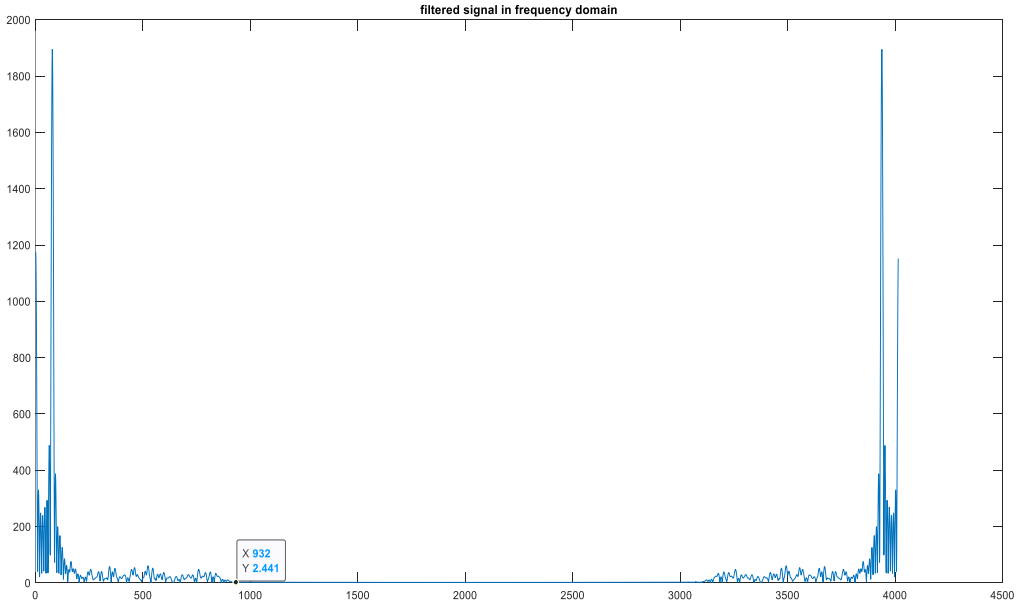
Butterworth LPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=LPF_BW_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Butterworth LPF (order
20)')
```

Output:



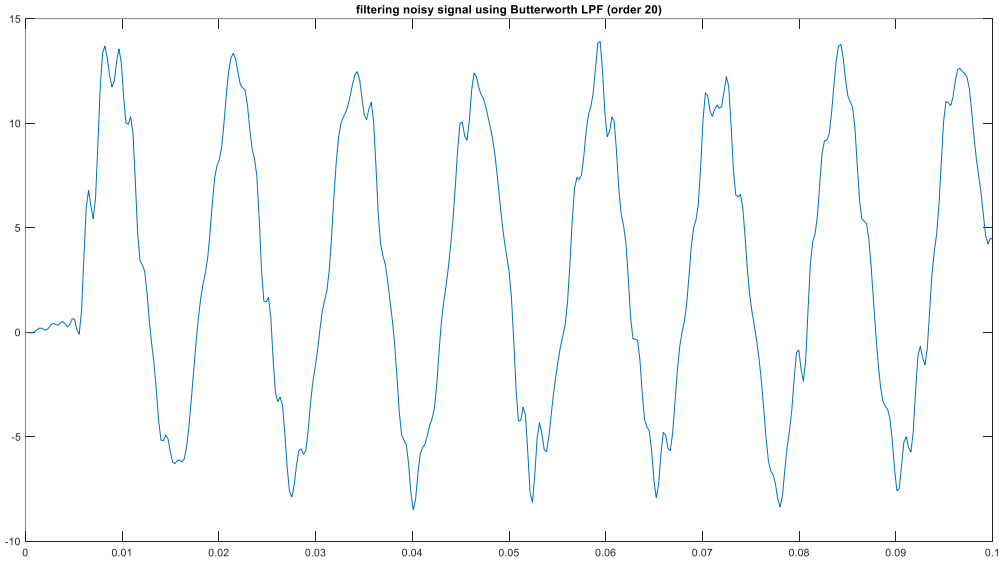

```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```



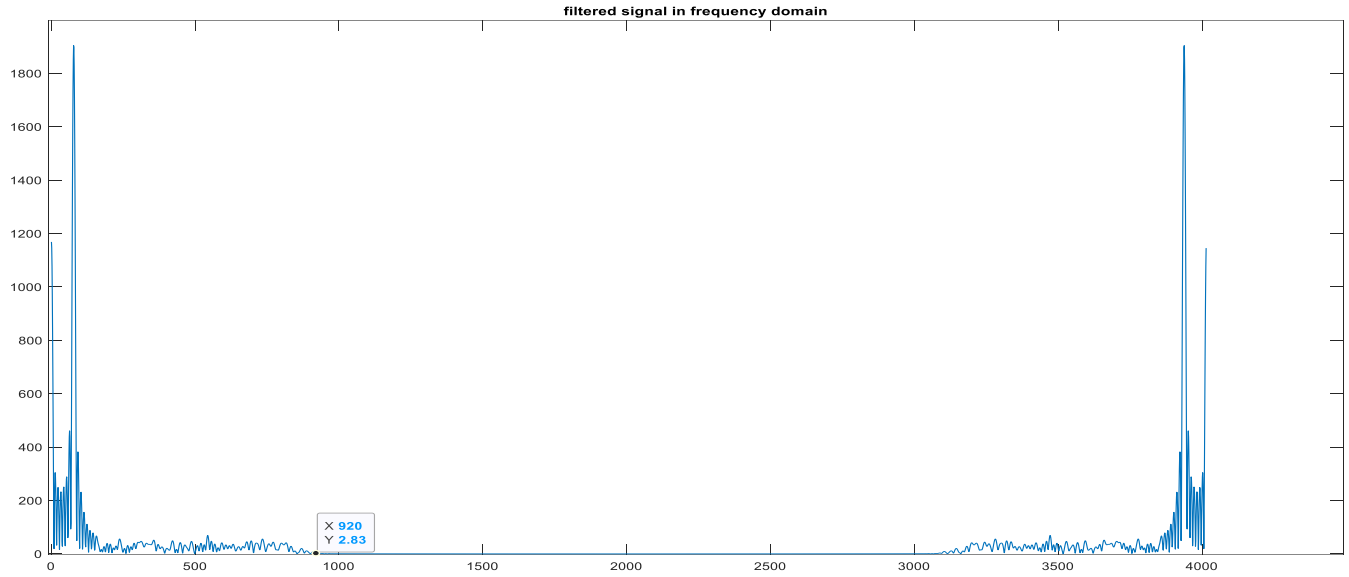
Chebyshev type-1 LPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=LPF_CS1_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using chebyshev type-1 LPF
(order 20)')
```

Output:



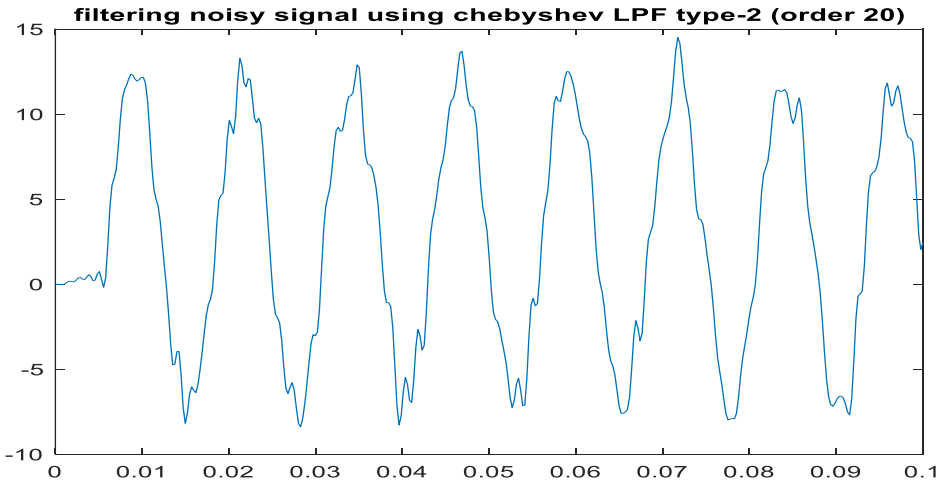
```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```



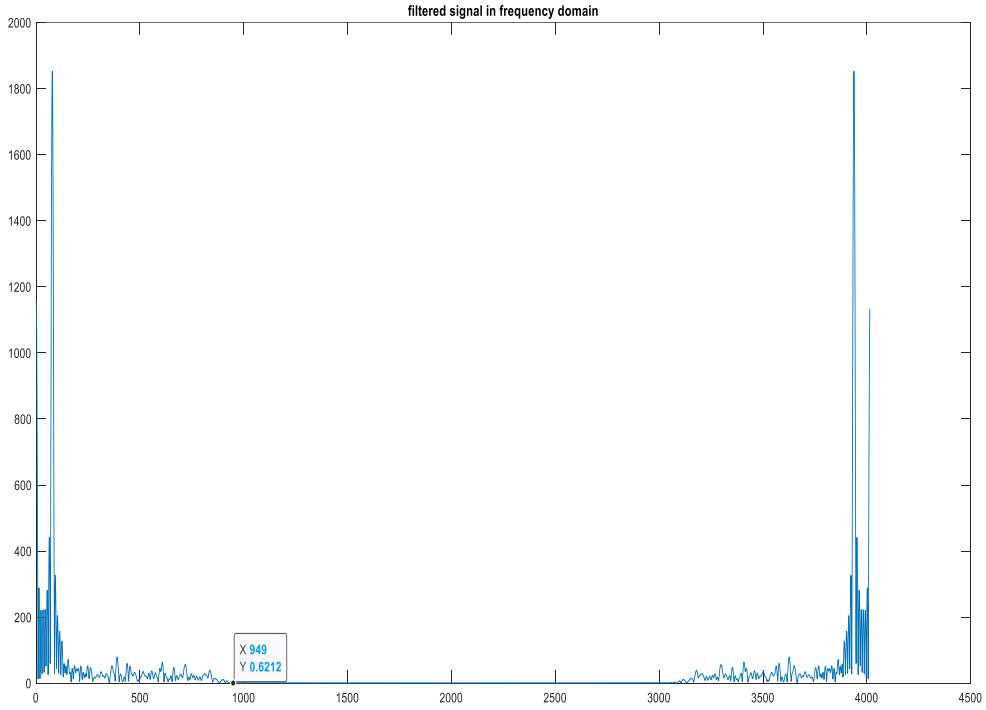
Chebyshev type-2 LPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=LPF_CS2_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using chebyshev LPF type-2
(order 20)')
```

Output:

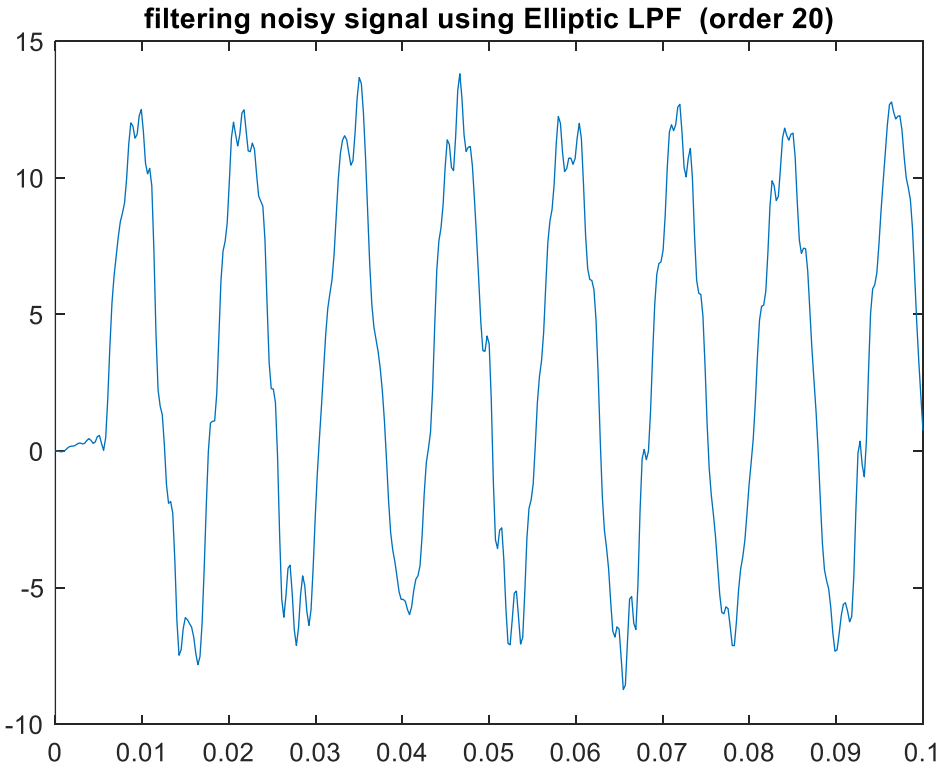


```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

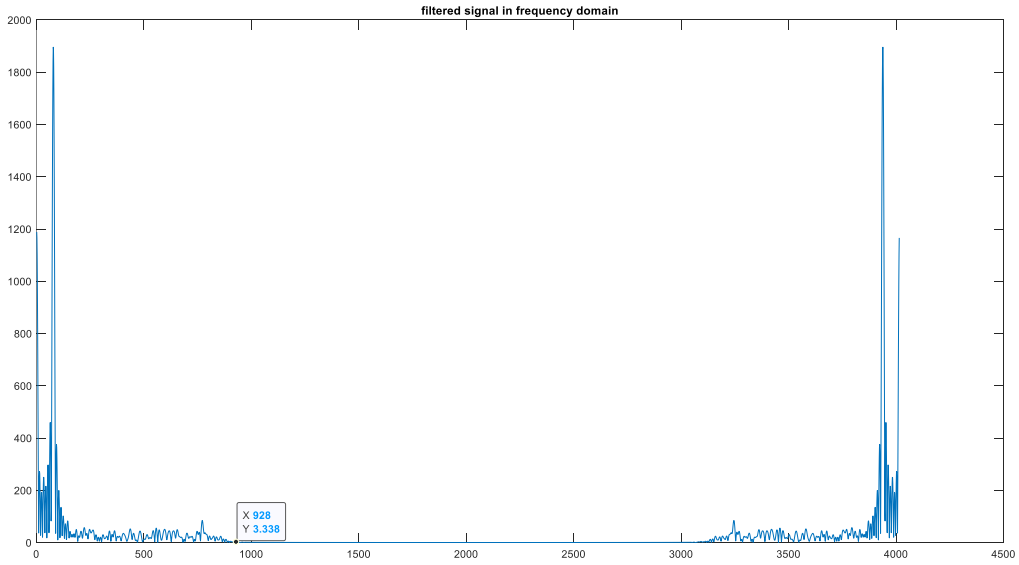


Elliptic LPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=LPF_Elliptic_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Elliptic LPF  (order
20)')
```

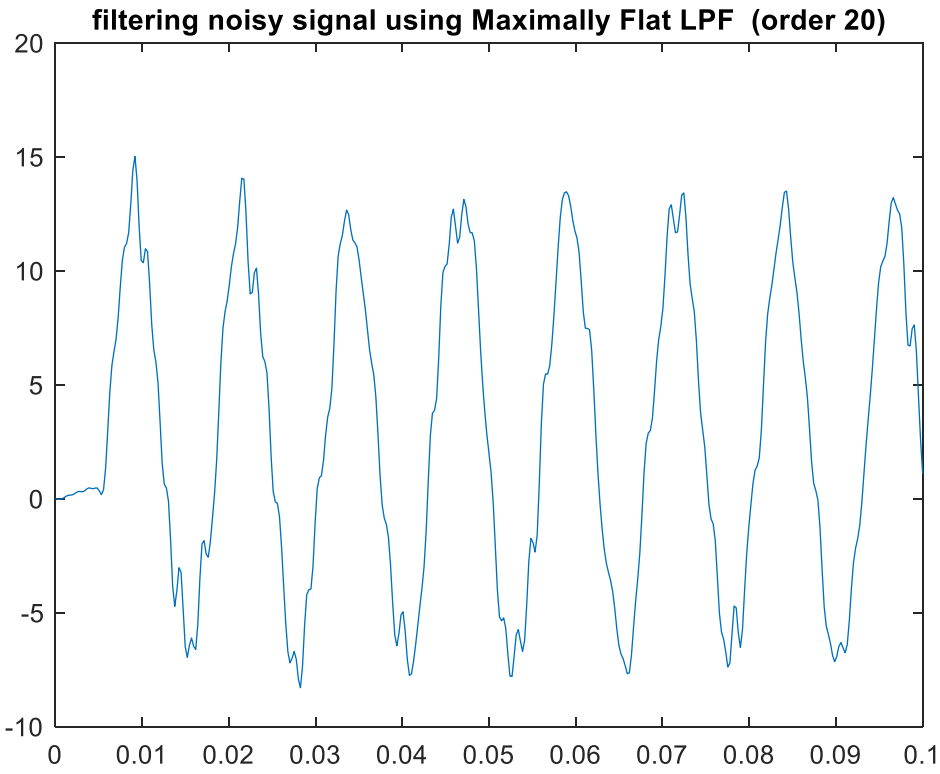



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

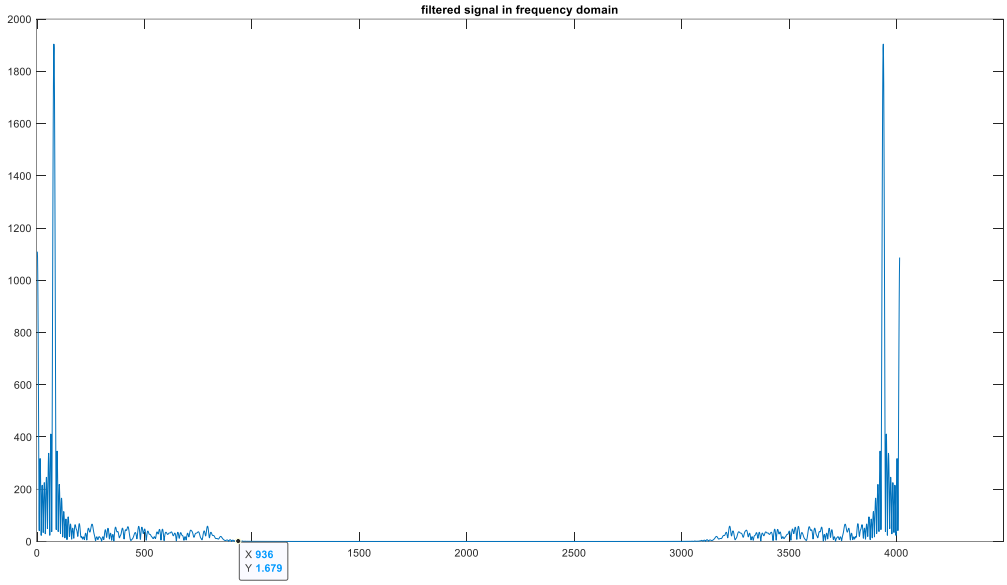


Maximally Flat LPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=LPF_MF_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Maximally Flat LPF
(order 20)')
```

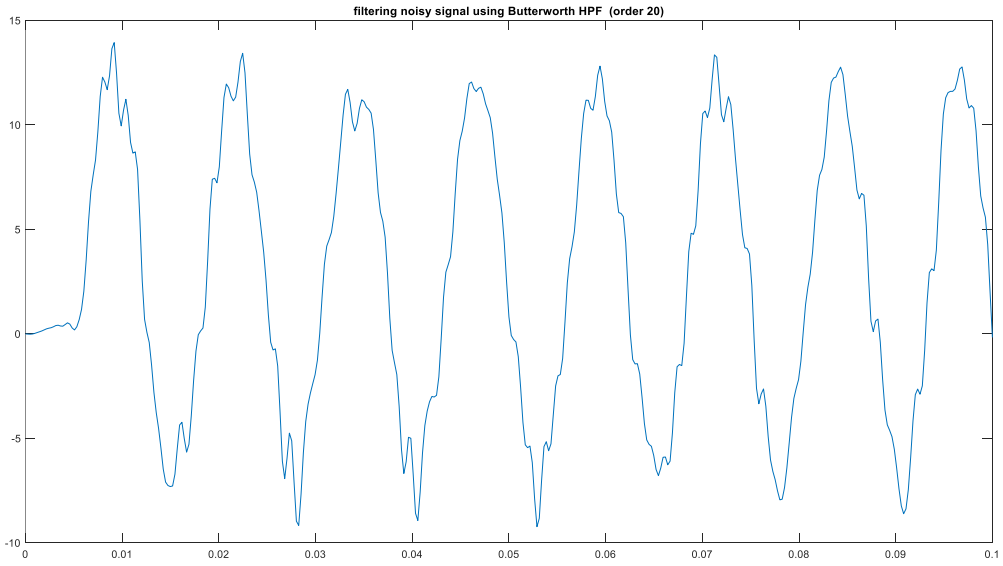



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

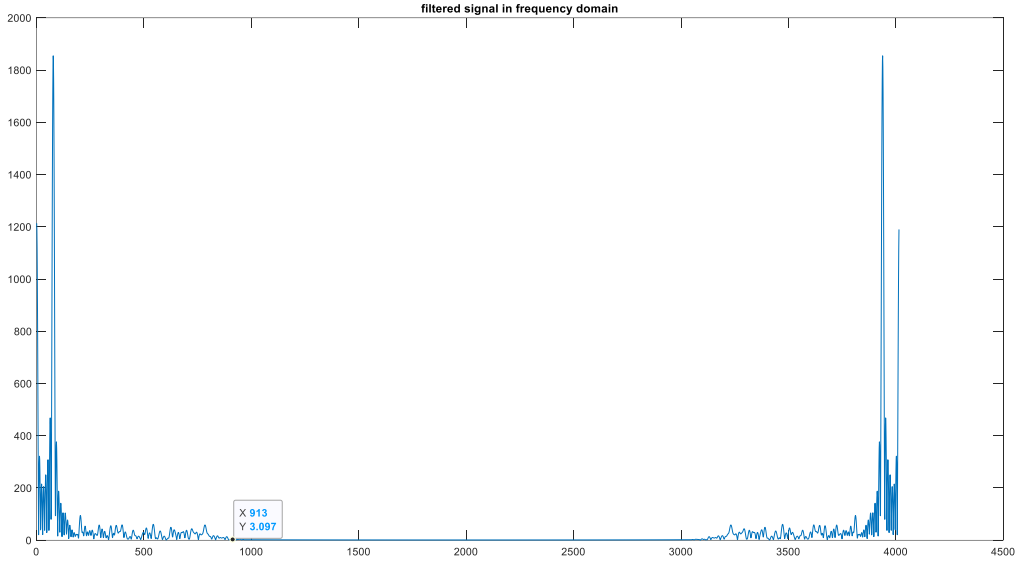


Butterwoth HPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=HPF_BW_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Butterworth HPF  (order
20)')
```

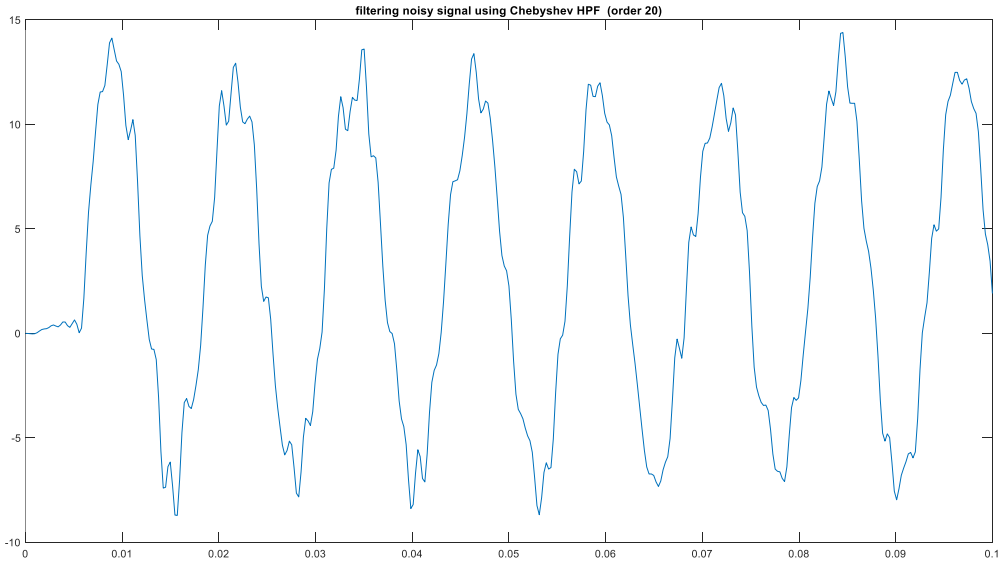


```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

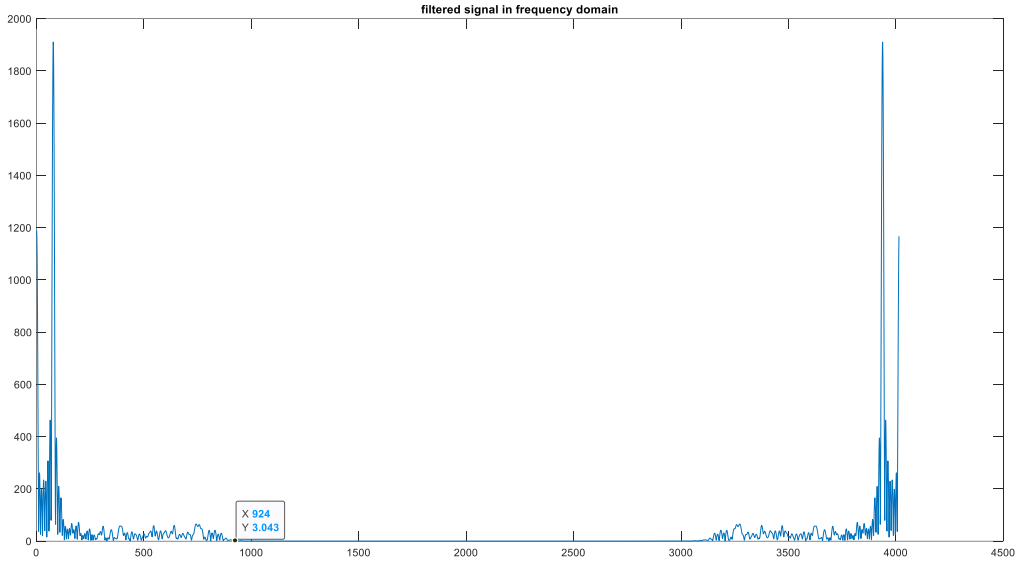


Chebyshev type-1 HPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=HPF_CS1_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Chebyshev HPF  (order
20)')
```

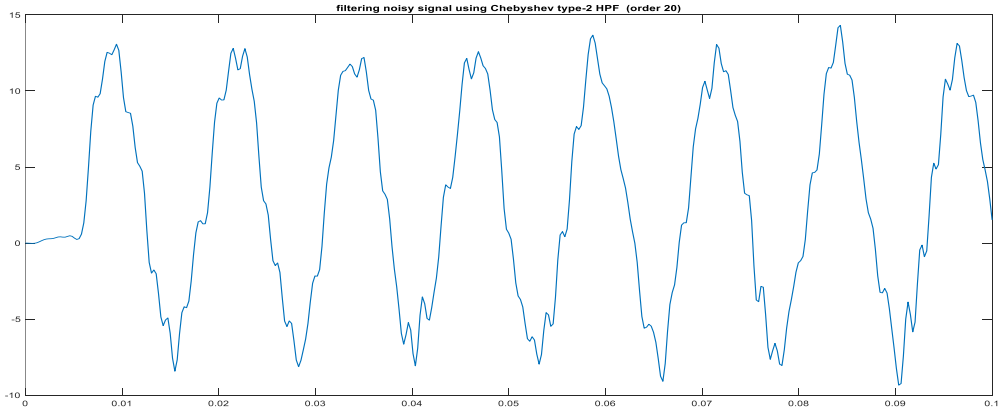


```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

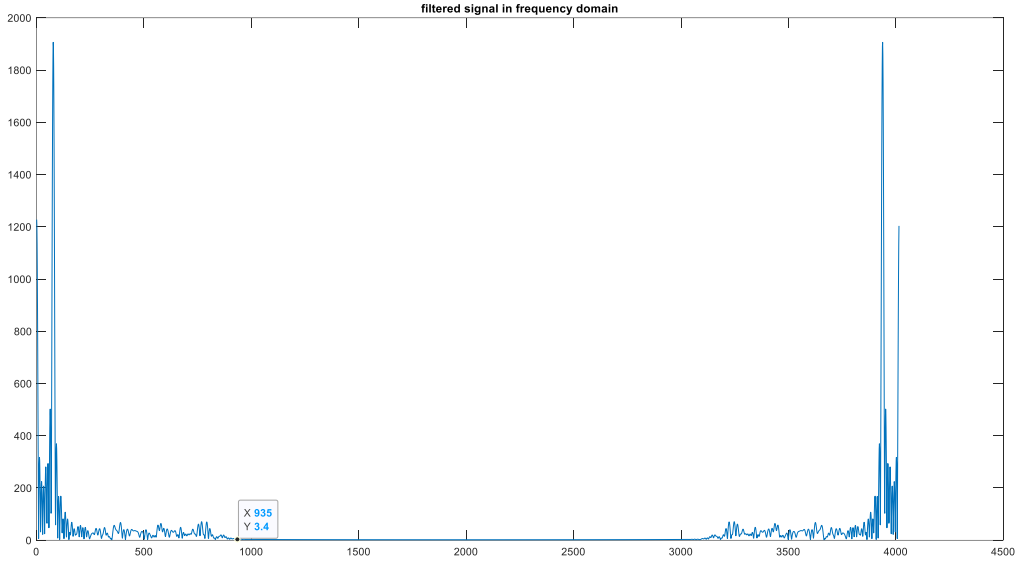


Chebyshev type-2 HPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=HPF_CS2_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Chebyshev type-2 HPF
(order 20)')
```

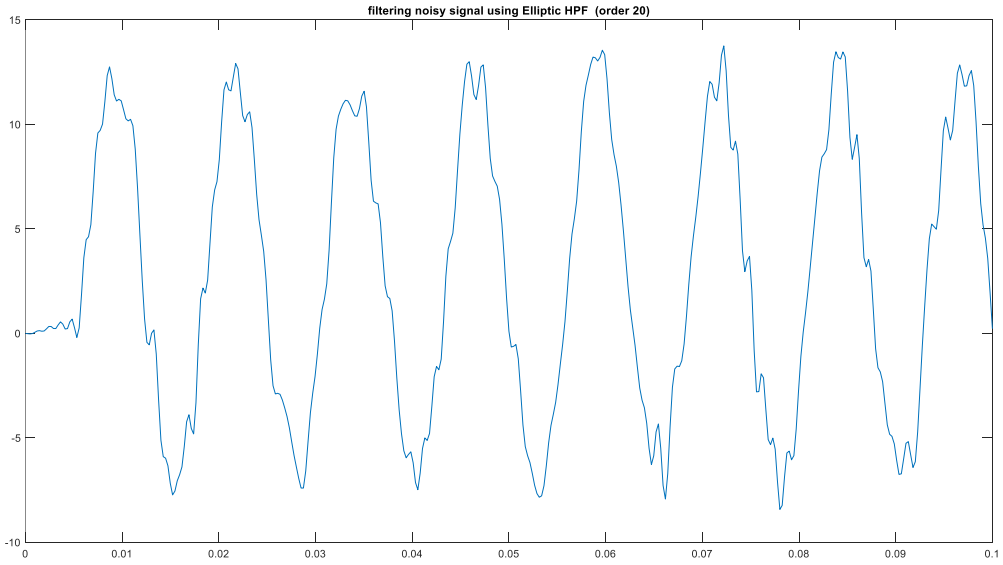


```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

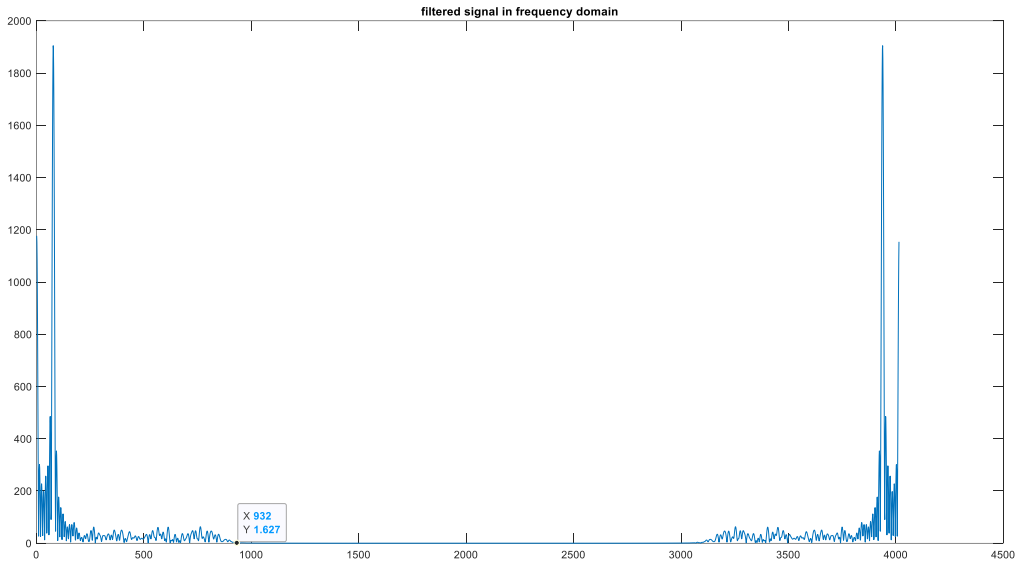


Elliptic HPF for IIR (ORDER = 2)

```
%Design filter using filterDesigner tool
filterpara_iir=HPF_Elliptic_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Elliptic HPF  (order
20)')
```

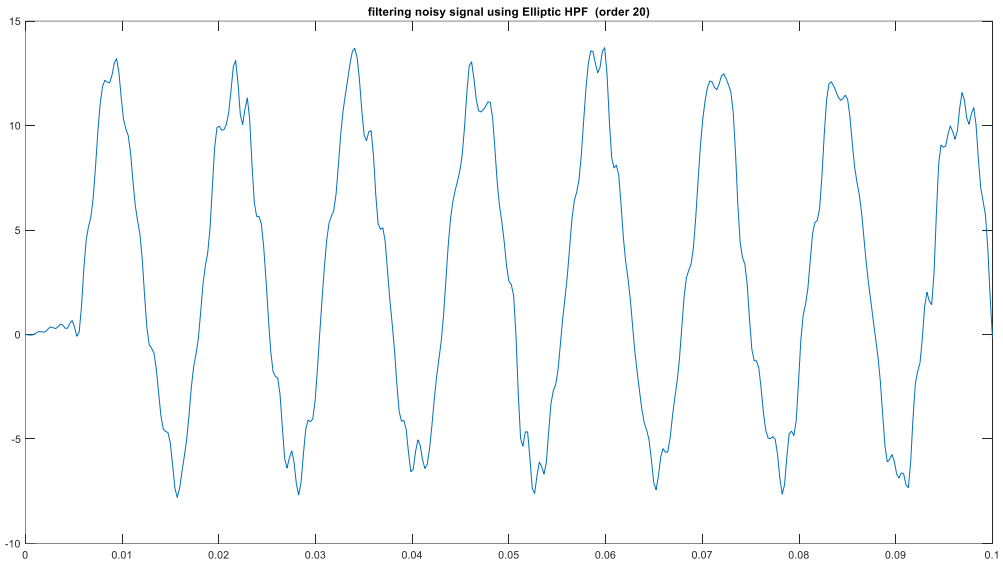


```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

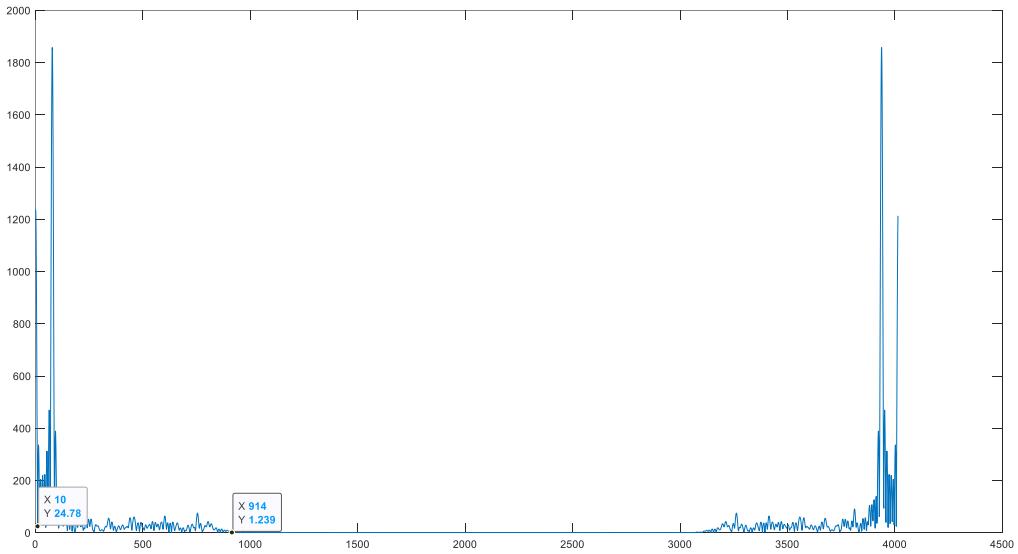


Butterwoth BPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=BPF_BW_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Elliptic HPF  (order
20)')
```



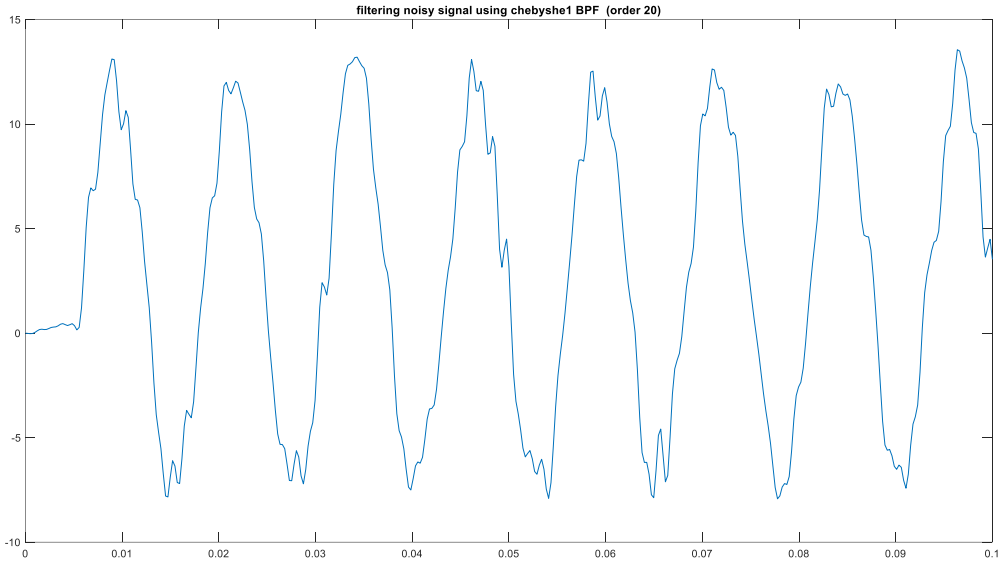

```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```



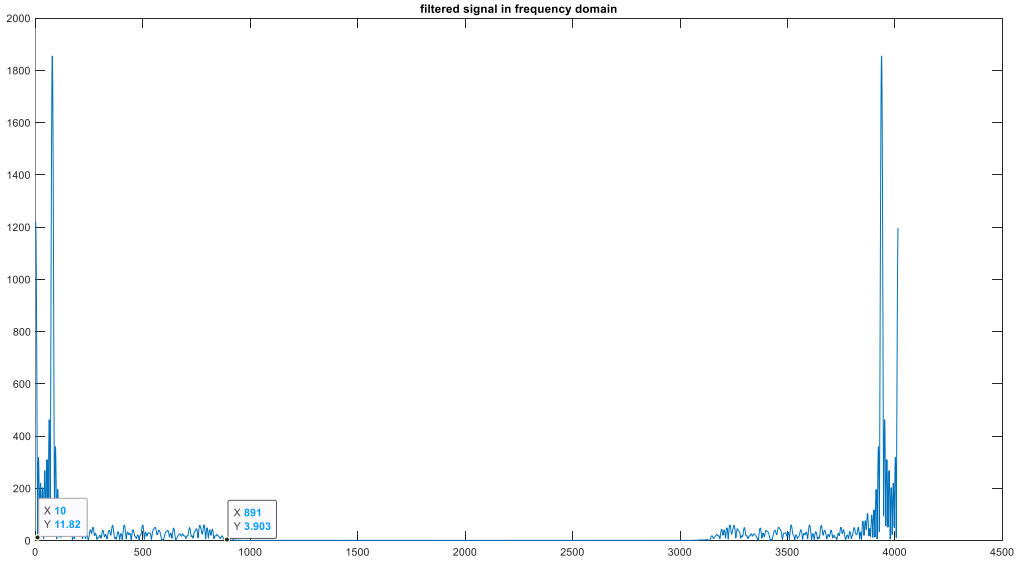
Chebyshev type-1 BPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=BPF_CS1_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
```

```
title('filtering noisy signal using chebyshe1 BPF  (order
20)')
```

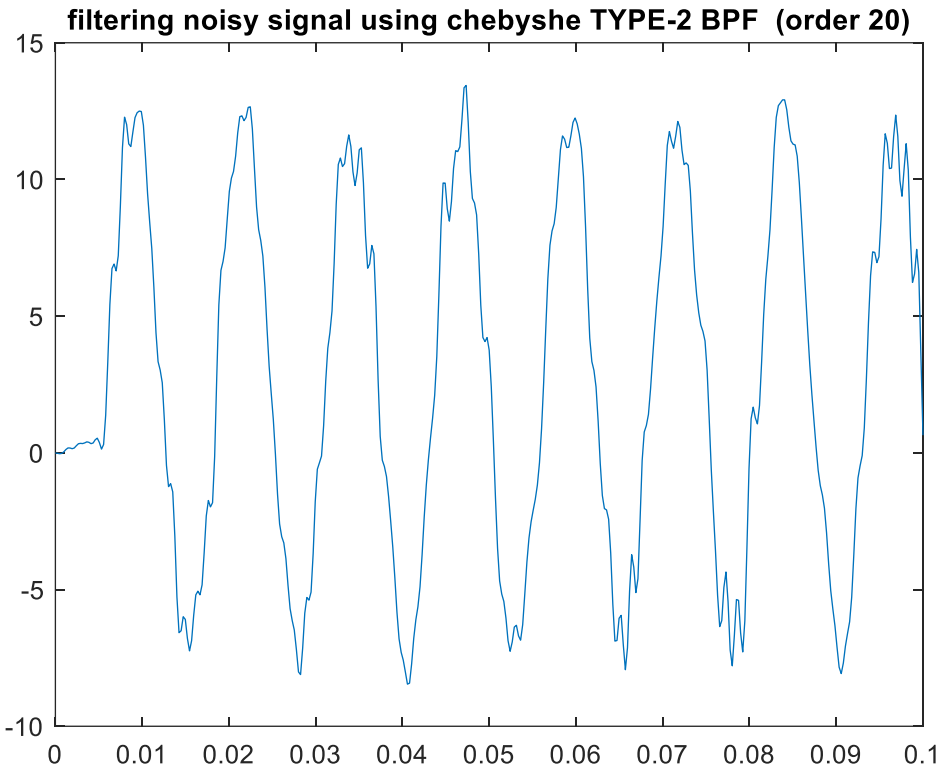



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

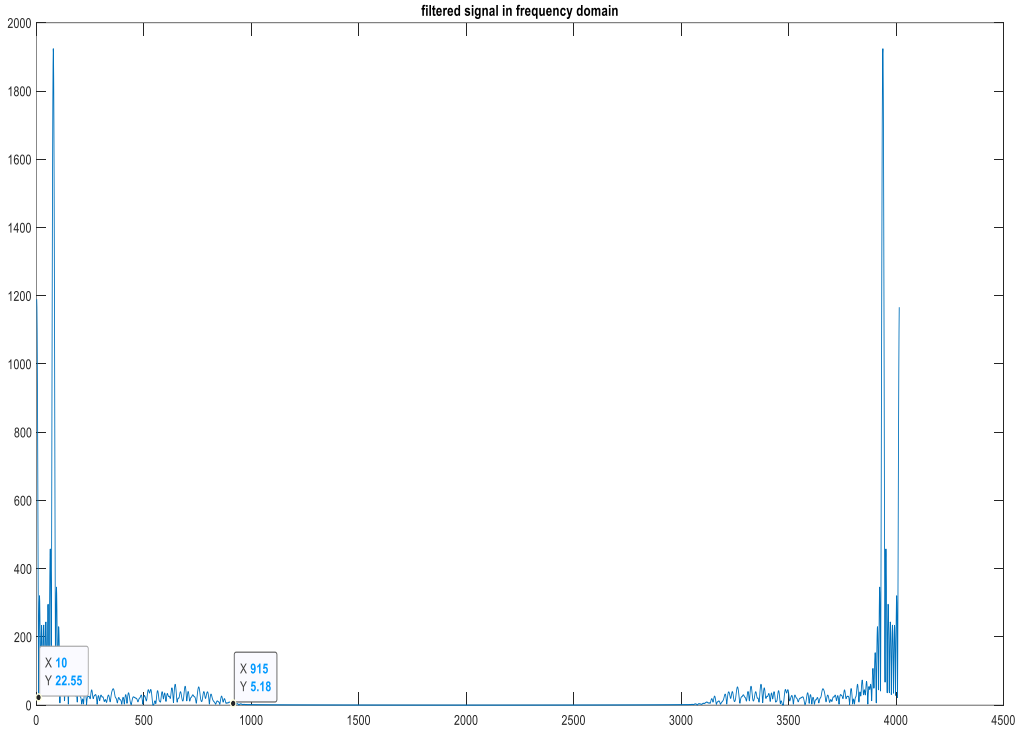


Chebyshev type-2 BPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=BPF_CS2_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using chebyshe TYPE-2 BPF
(order 20)')
```

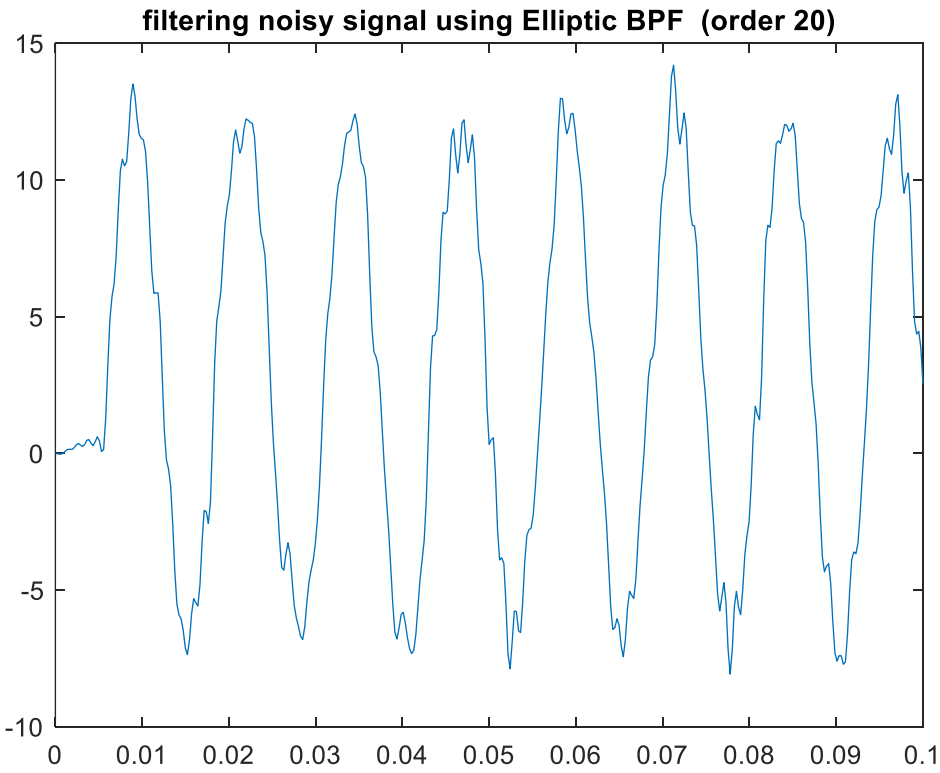



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

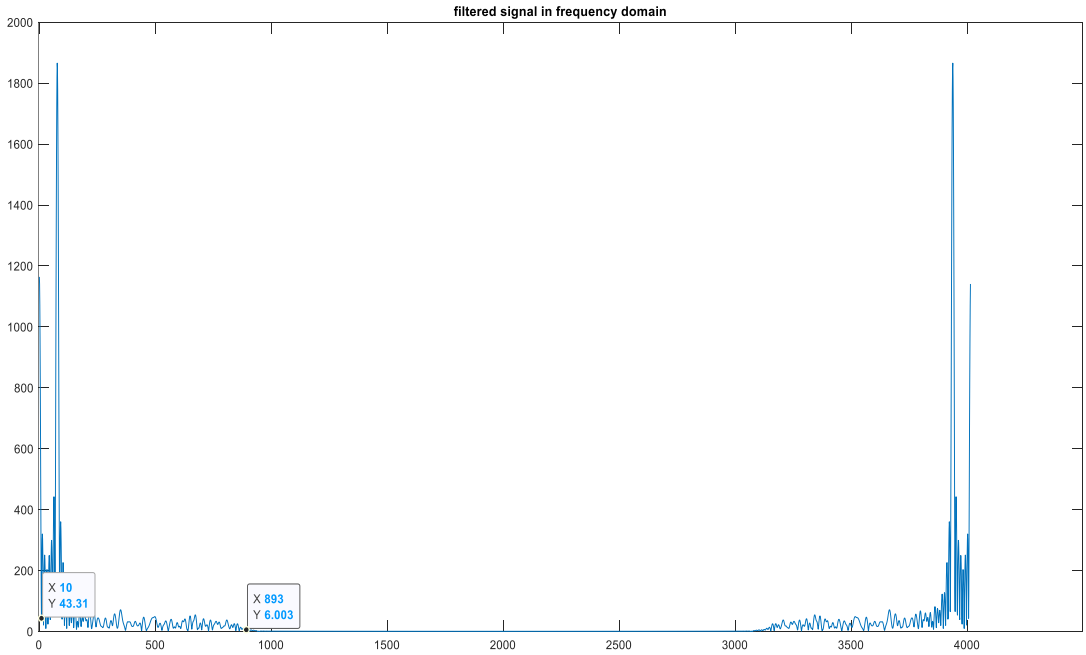


Elliptic BPF for IIR (ORDER = 20)

```
%Design filter using filterDesigner tool
filterpara_iir=BPF_Elliptic_20;
x_f_IIR = filter(filterpara_iir,z);
plot(n,x_f_IIR)
title('filtering noisy signal using Elliptic BPF  (order
20)')
```

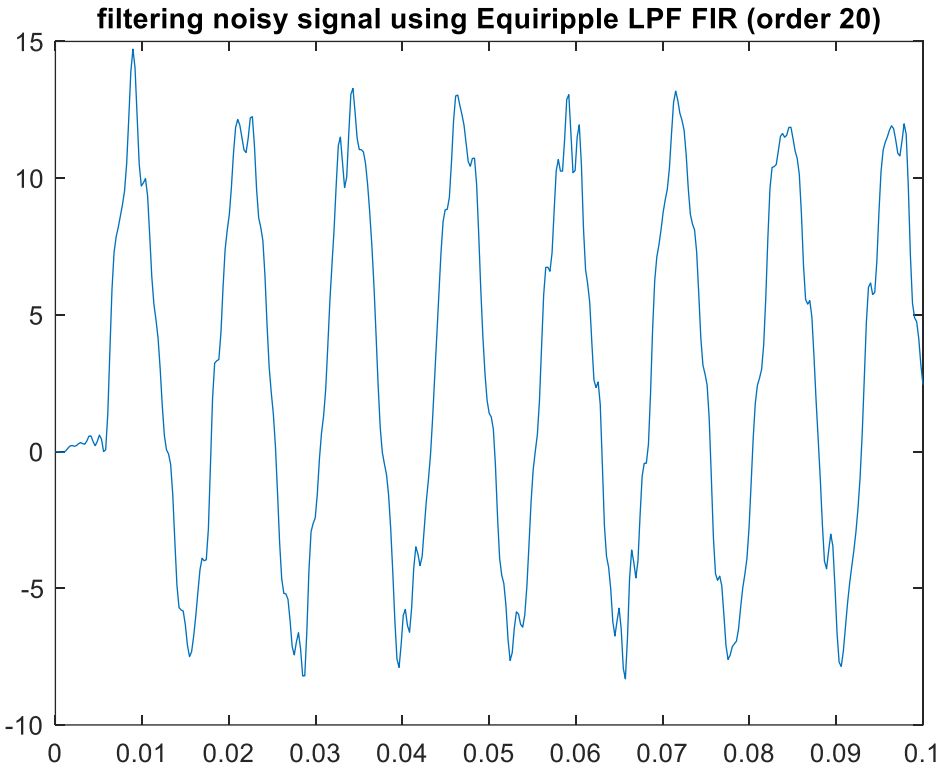



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

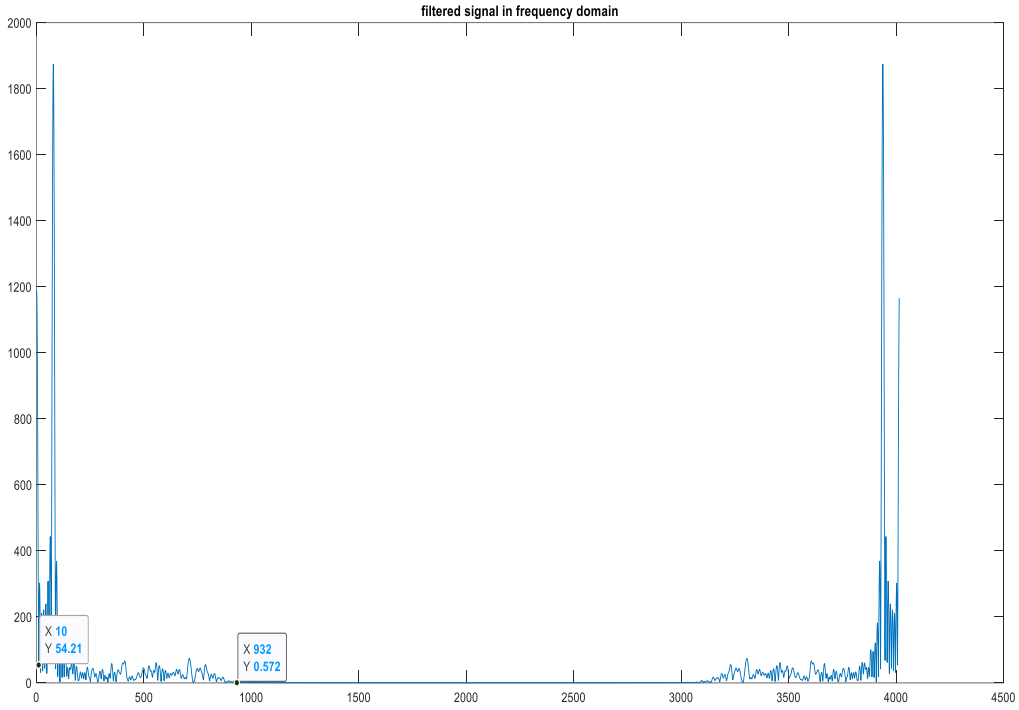


Equiripple LPF for FIR (ORDER=20):

```
%Design filter using filterDesigner tool
filterpara_fir= Equiripple_FIR__LPF;
x_f_FIR = filter(filterpara_fir,z);
plot(n,x_f_FIR)
title('filtering noisy signal using Equiripple LPF FIR
(order 20)')
```

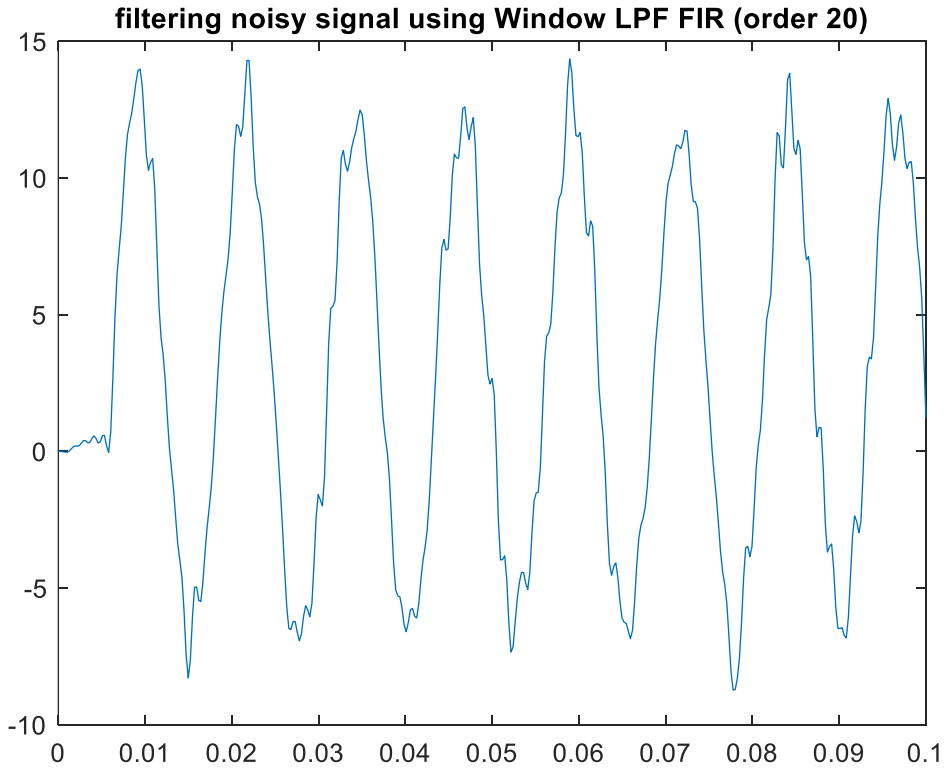



```
% Fast Fourrier Transform
plot(abs(fft(x_f_IIR,4014)))
title('filtered signal in frequency domain');
```

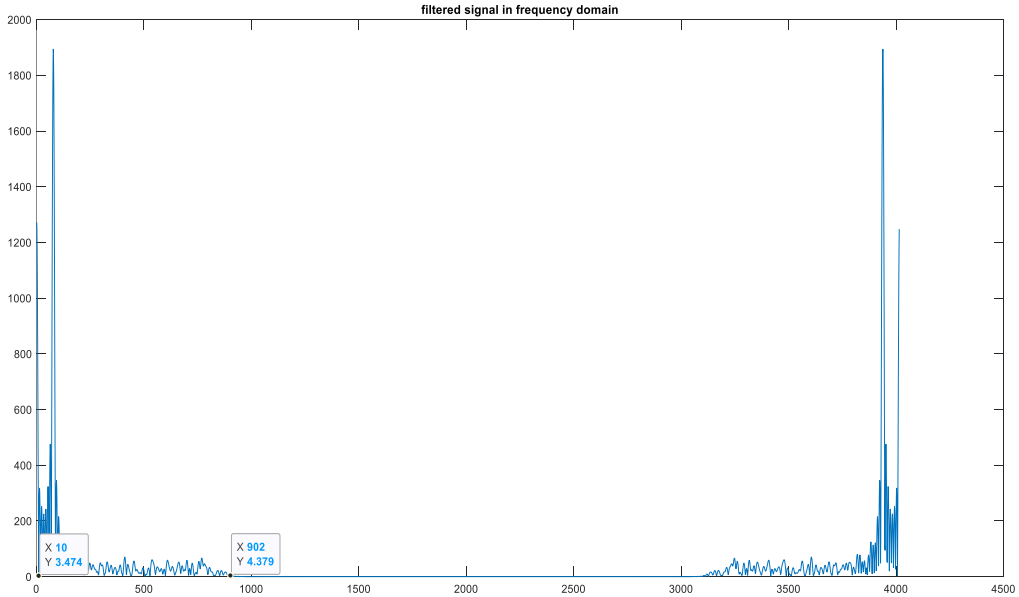


Window LPF for FIR (ORDER=20):

```
%Design filter using filterDesigner tool
filterpara_fir= Window_FIR__LPF;
x_f_FIR = filter(filterpara_fir,z);
plot(n,x_f_FIR)
title('filtering noisy signal using Window LPF FIR (order
20)')
```

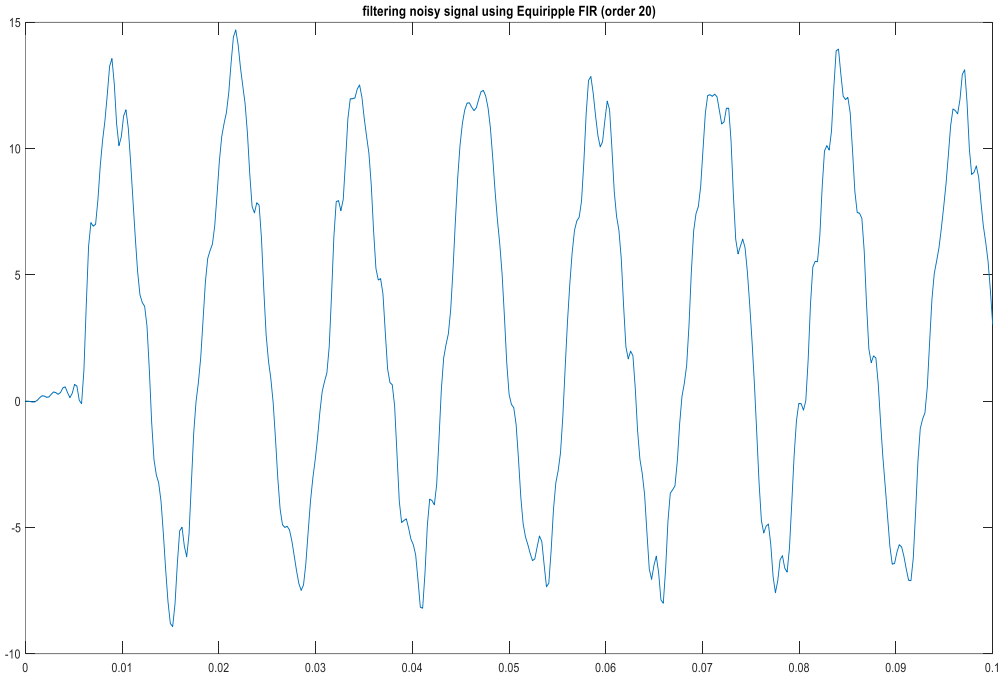



```
% Fast Fourrier Transform
plot(abs(fft(x_f_FIR,4014)))
title('filtered signal in frequency domain');
```

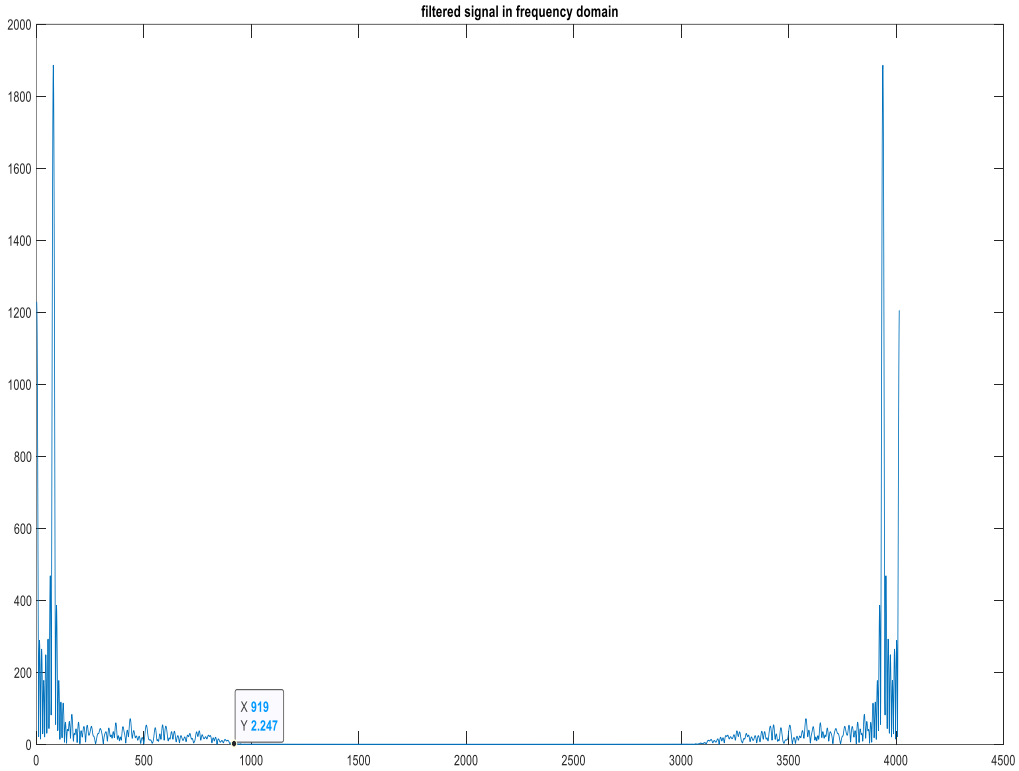


Equiripple HPF for FIR (ORDER=20):

```
%Design filter using filterDesigner tool
filterpara_fir= Equiripple_FIR__LPF;
x_f_FIR = filter(filterpara_fir,z);
plot(n,x_f_FIR)
title('filtering noisy signal using Equiripple LPF FIR
(order 20)')
```

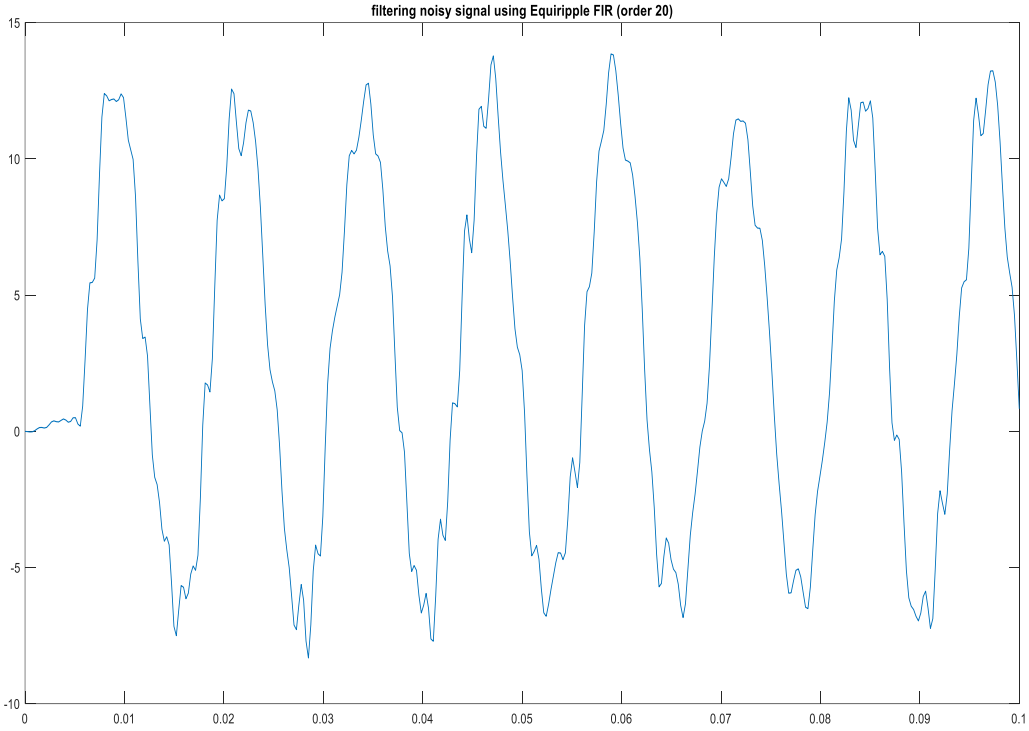



```
% Fast Fourrier Transform
plot(abs(fft(x_f_FIR,4014)))
title('filtered signal in frequency domain');
```



Equiripple BPF for FIR (ORDER=20):

```
%Design filter using filterDesigner tool
filterpara_fir= bandpass;
x_f_FIR = filter(filterpara_fir,z);
plot(n,x_f_FIR)
title('filtering noisy signal using Equiripple FIR (order
20)')
```




```
% Fast Fourrier Transform
plot(abs(fft(x_f_FIR,4014)))
title('filtered signal in frequency domain');
```