**Heaven's Light is Our Guide**

**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

**LAB REPORT**

**Course No:** EEE 4108

**Course Name:** Digital signal processing Sessional

**Experiment No:** 03

**Experiment Name:** Study of plotting FFT of three signals as well as filter signal using FFT analysis, Butter-worth and Chebyshev Filter

**Date of experiment:** 04/03/2023

**Date of Submission:** 06/05/2023

| **Submitted By:** | **Submitted To:** |
|---|---|
| Name: Md Maruf Hassan<br>Roll: 1801105<br>Section: B<br>Registration No: 285<br>Session: 2018-2019 | Dr. Mohammod Abdul Motin<br>Assistant Professor,<br>Department of Electrical & Electronic Engineering<br>Rajshahi University of Engineering & Technology |

**Experiment No:** 03

**Experiment Name:** Study of plotting FFT of three signals as well as filter signal using FFT analysis, Butter-worth and Chebyshev Filter

**Objectives:** The following objectives of this experiment is-

- To learn about the FFT (First Fourier Transform)
- To filter noisy signal using FFT analysis, Butter-worth Filter and Chebyshev Filter.

**Theory:**

FFT (Fast Fourier Transform) is an algorithm used to efficiently compute the Fourier Transform of a signal. It analyzes the frequency components present in a signal by decomposing it into a combination of sinusoidal functions. The FFT algorithm is faster than the standard DFT algorithm, making it widely used in signal processing, image/audio processing, and other applications. It achieves its efficiency by dividing the problem into smaller sub-problems and combining the results. FFT has revolutionized many fields and enables real-time processing of signals and efficient computations involving the Fourier Transform.

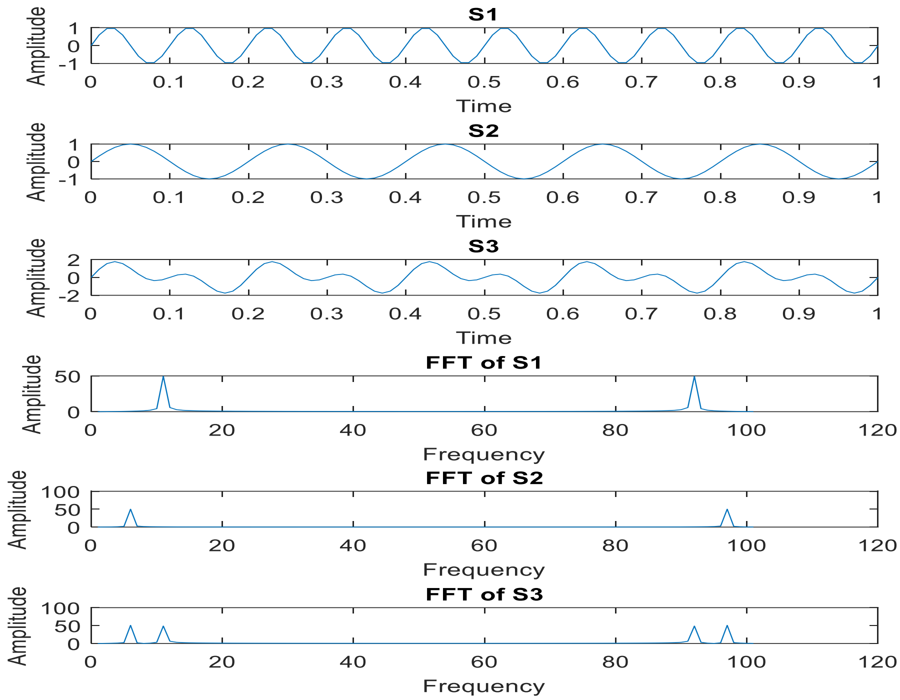
A Butterworth filter is a type of frequency-selective filter commonly used in signal processing and electronics to pass or reject specific frequencies in a signal. It belongs to the family of infinite impulse response (IIR) filters, which means it incorporates feedback in its implementation. The primary characteristic of a Butterworth filter is its maximally flat frequency response in the passband. This means that the filter's gain is constant within the passband region, making it suitable for applications where a smooth and uniform frequency response is essential, such as audio processing.

**MATLAB Code:** s1 and s2 are two separate given signals. And s3=s1+s2, plot the Fast Fourier Transform (FFT) of s1, s2 and s3

```
clc;
clear all;
t=0:0.01:1;
f=10;
s1=sin(2*pi*f*t);
subplot(611)
plot(t,s1)
ylabel('Amplitude');
xlabel('Time');
title('S1');
f1=5;
s2=sin(2*pi*f1*t);
subplot(612)
```

```
plot(t,s2)
ylabel('Amplitude');
xlabel('Time');
title('S2')
s3=s1+s2;
subplot(613)
plot(t,s3)
ylabel('Amplitude');
xlabel('Time');
title('S3')
p=fft(s1)
p1=abs(p)
subplot(614)
plot(p1)
ylabel('Amplitude');
xlabel('Frequency');
title('FFT of S1')
q=fft(s2)
q1=abs(q)
subplot(615)
plot(q1)
ylabel('Amplitude');
xlabel('Frequency');
title('FFT of S2')
r=fft(s3)
r1=abs(r)
subplot(616)
plot(r1)
ylabel('Amplitude');
xlabel('Frequency');
title('FFT of S3');
```

**Output:**

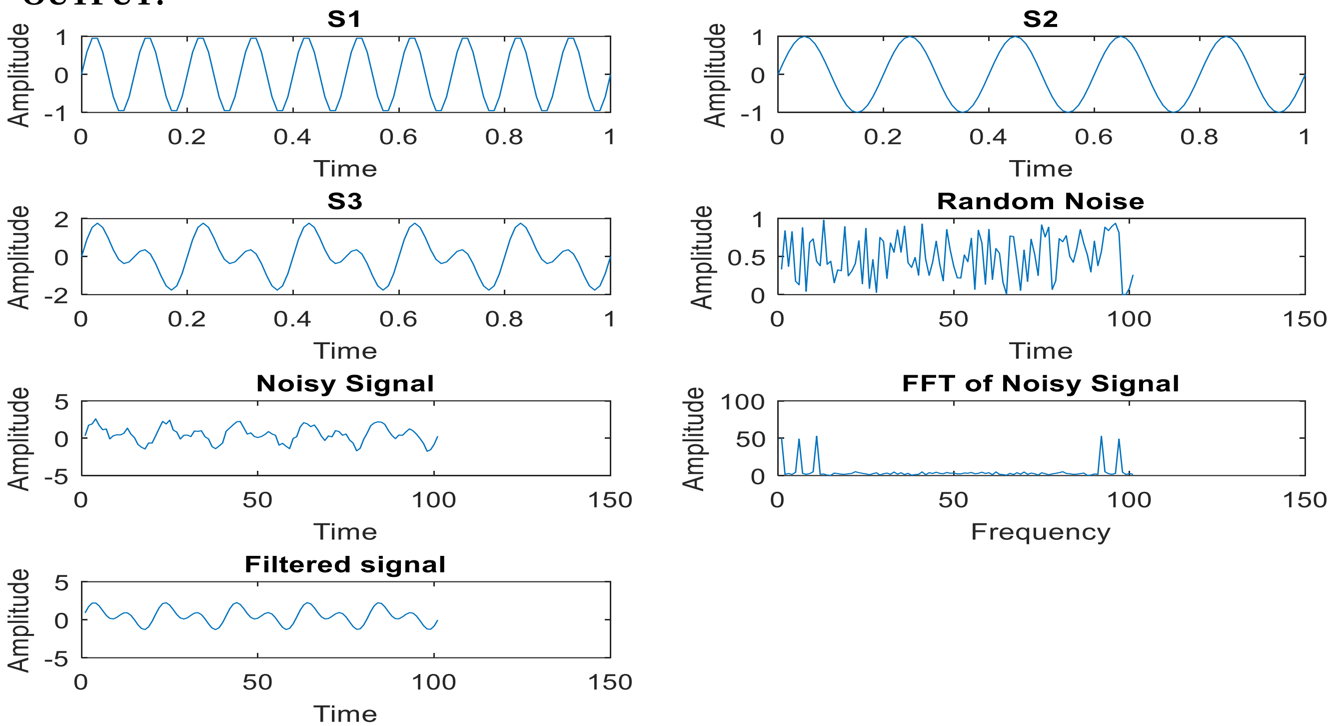



**MATLAB Code**: s4=s3+ random noise, write a MATLAB code to remove the random noise from s4 using FFT analysis.

```
clc;
clear all;
t=0:0.01:1;
f=10;
s1=sin(2*pi*f*t);
subplot(421);
plot(t,s1);
ylabel('Amplitude');
xlabel('Time');
title('S1');
f1=5;
s2=sin(2*pi*f1*t);
subplot(422);
plot(t,s2);
```

```
ylabel('Amplitude');
xlabel('Time');
title('S2');
s3=s1+s2;
subplot(423);
plot(t,s3);ylabel('Amplitude');
xlabel('Time');
title('S3');
n=rand(size(t));
subplot(424);
plot(n);
ylabel('Amplitude');
xlabel('Time');
title('Random Noise');
s4=s3+n;
subplot(425);
plot(s4);
ylabel('Amplitude');
xlabel('Time');
title('Noisy Signal');
p=fft(s4);

p1=abs(p);
subplot(426);
plot(p1);
ylabel('Amplitude');
xlabel('Frequency');
title('FFT of Noisy Signal');

k=find(p1<30);
p(k)=zeros(size(k));
p=ifft(p);
subplot(427);
plot(p);
ylabel('Amplitude');
xlabel('Time');
title('Filtered signal');
```

**OUTPUT:**

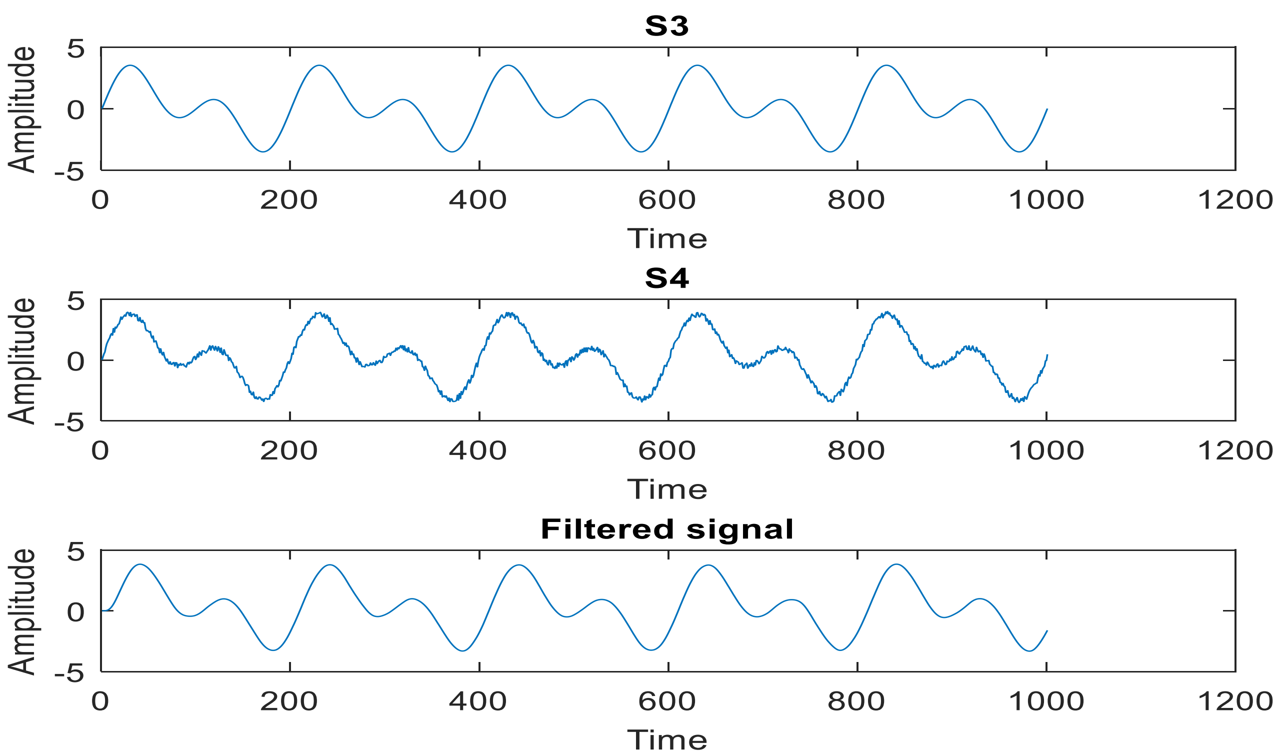



**MATLAB Code**: Denoise the signal s4 by designing a filter (Butter-worth filter) using MATLAB Code

```
clc;
clear all;
fs=1000;
a=2;
t=0:1/fs:1;
s1=a*sin(2*pi*10*t)
s2=a*sin(2*pi*5*t);
s3=s1+s2;
n=0.5*rand(size(t));
s4=s3+n;
fo=3;
fc=30;
w=2*fc/fs;
[b,a]=butter(fo,w);
filtrated_signal=filter(b,a,s4);
```

```
subplot(311)
plot(s3)
ylabel('Amplitude');
xlabel('Time');
title('S3');

subplot(312)
plot(s4)
ylabel('Amplitude');
xlabel('Time');
title('S4');

subplot(313)
plot(filtrated_signal)
ylabel('Amplitude');
xlabel('Time');
title('Filtered signal');
```

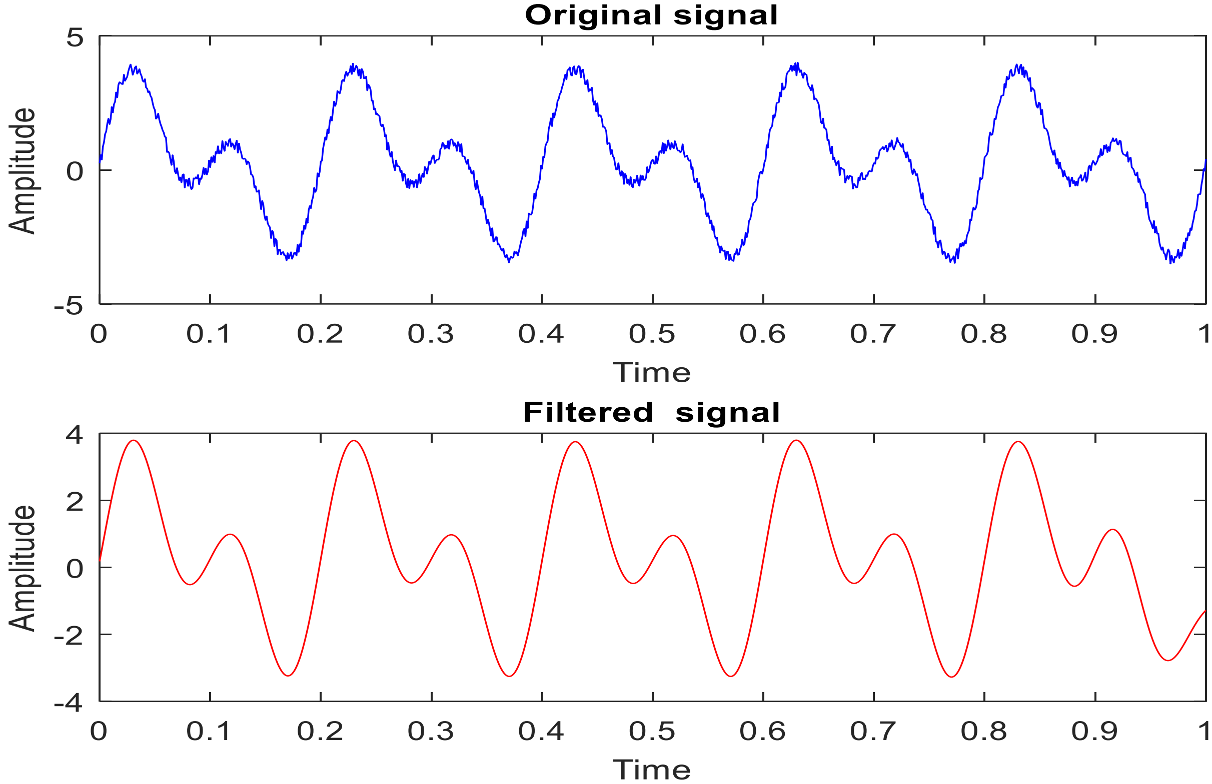
**OUTPUT:**



**MATLAB Code:** Denoise the signal s4 by designing a filter (Chebyshev filter) using MATLAB Code

```
clc;
close all;
clear all;

fs=1000;
a=2;
t=0:1/fs:1;
s1=a*sin(2*pi*10*t);

s2=a*sin(2*pi*5*t);
s3=s1+s2;
n=0.5*rand(size(t));
x=s3+n;

order=6;
passbandRipple=30;
cutoffs=2*20/fs;
[b, a] = cheby2(order, passbandRipple, cutoffs);

filteredSignal = filtfilt(b, a, x);

subplot(211);
plot(t, x, 'b', 'LineWidth', 0.5); hold on;

xlabel('Time');
ylabel('Amplitude');

title('Original signal');
subplot(212);
plot(t, filteredSignal, 'r', 'LineWidth', 0.5);
xlabel('Time');
ylabel('Amplitude');
title('Filtered  signal');
```

**OUTPUT**:




**Discussion & Conclusion:**

In this experiment, we have experimented the FFT of three signals and filtering the noisy signals. Three method is used to filter that was FFT analysis, Butter-worth Filter, Chebyshev Filter. We have successfully filtered the noisy signals by these three methods. All of case, we have obtained very smooth signal which is similar to the original signal we got before adding the random noise signal. The all-output signal was same to desired.