PREMIER UNIVERSITY

Department of Computer Science & Engineering

Course Code	:	**EEE-202**
Course Title	:	**Signals & Systems Laboratory**
Report No	:	**06**
Name of Report	:	**Implementing Fourier transform in MATLAB**
Date of Performance	:	**23/12/2023**
Date of Submission	:	**20/01/2024**
Course Instructor	:	Mohammed Saifuddin Munna

Submitted By :

Name	: **Mohammad Hafizur Rahman Sakib**
ID	: **0222210005101118**
Section	: **C**
Semester	: **4th Semester**
Session	: **Fall 2023**

REMARKS

Experiment no: 06
Experiment name: Implementing Fourier transform in MATLAB
Objective: To understand and implement the Fourier transform using MATLAB.
Software Requirement: MATLAB 2014
Theory:
Fourier transform
The Fourier transform of the expression f = f(x) with respect to the variable x at the point w is defined as follows

Here, c and s are parameters of the Fourier transform. The Fourier function uses c = 1, s = –1.

Syntax
fourier(f,trans_var,eval_point)

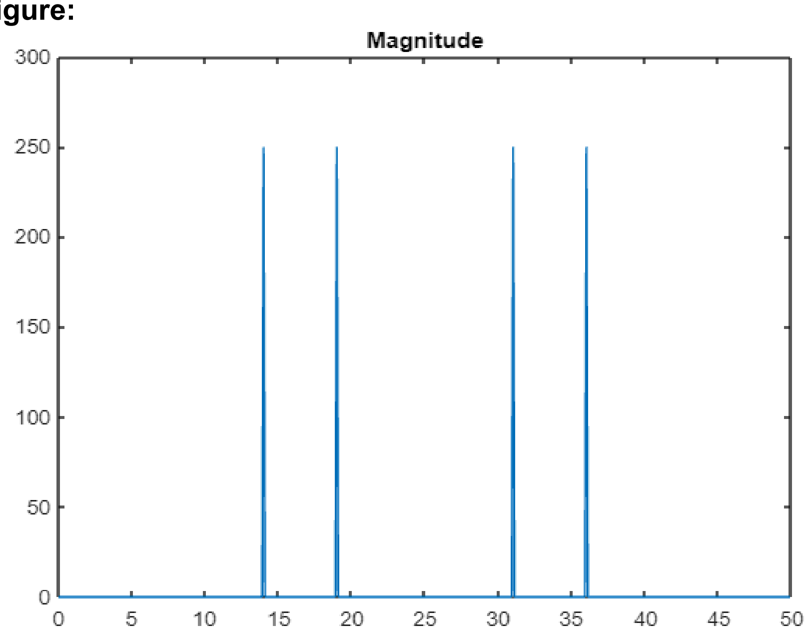
Examples:
Compute the Fourier transform of this expression with respect to the variable x at the evaluation point y:
There Are two Types of Fourier transformation:

- For symbolic Value

syms x y
f = exp(-x^2);
fourier(f, x, y)
ifourier(ans)
You can use it for any function:

- Real Value:
- **Complex wave**

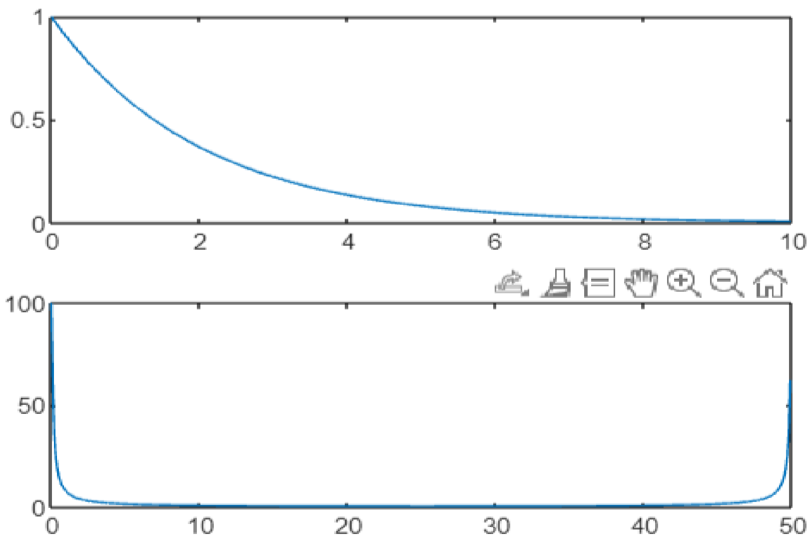
```
t = 0:1/50:10-1/50;
x = sin(2*pi*14*t) + sin(2*pi*19*t);
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
plot(f, abs(y))
title('Magnitude')
```

Figure:


Exponential Functions:

```
t = 0:1/50:10-1/50;
x = exp(-0.5*t);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

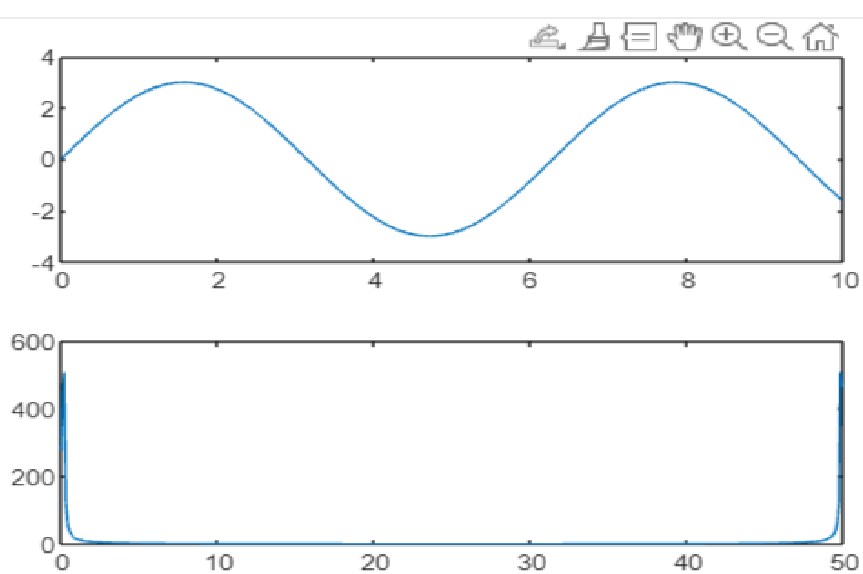
Figure:



Simple Sin Function;

```
t = 0:1/50:10-1/50;
x = 3*sin(t);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

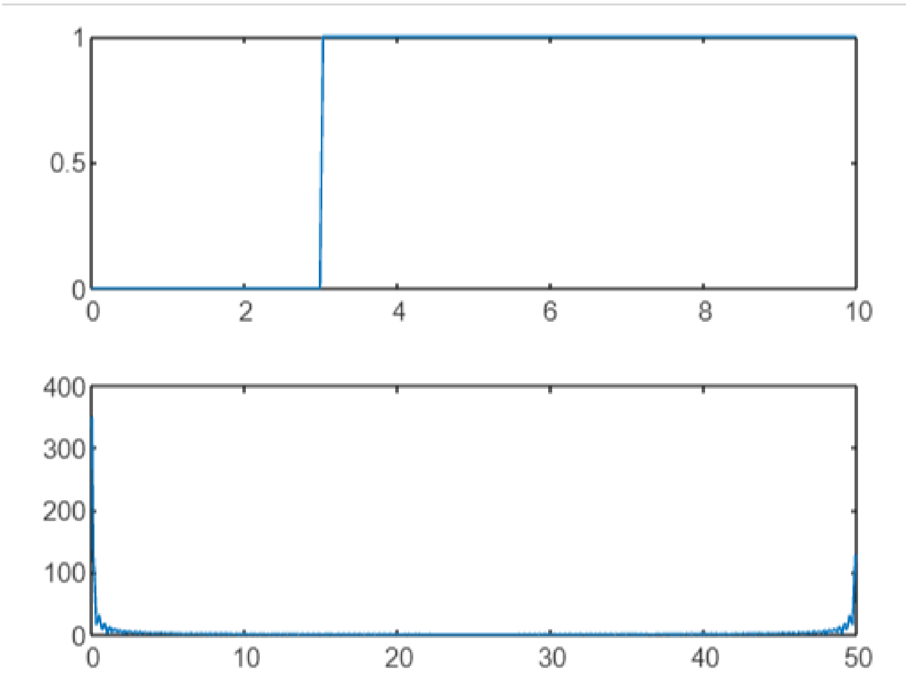
Figure:



Unit Step Function:

```
t = 0:1/50:10-1/50;
x = heaviside(t - 3);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

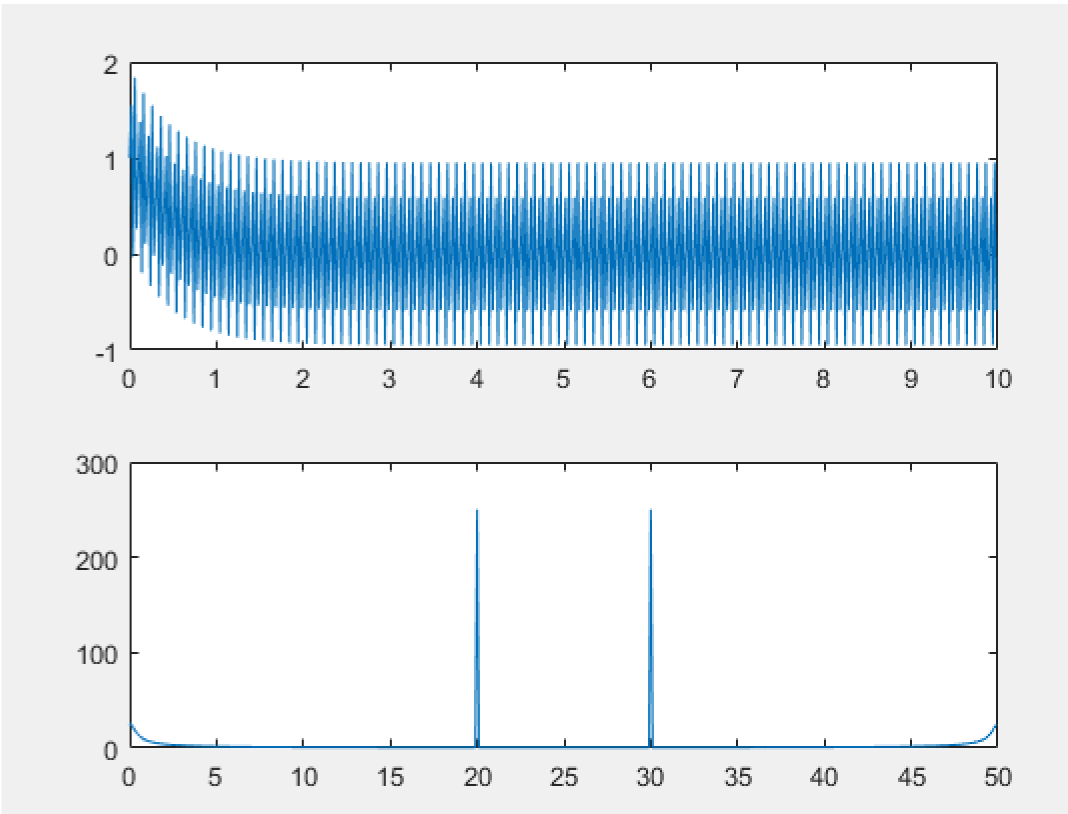
Figure:



Task: Do the same for the ramp signal & any complex Wave mixed with sin function & exponential functions.

```
t = 0 : 1/50:10-1/50;
x = sin(2*pi*20*t)+exp(-2*t);
subplot(2,2,1);
subplot(2,1,1);
plot(t,x);
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2);
plot(f,abs(y));
```

Figure:



Discussion:

In this lab, our main objective was to gain practical experience in implementing the Fourier Transform for diverse functions using MATLAB. We began by defining parameters for a sinusoidal signal and utilized the sin function to generate it. The Fourier Transform was computed using the fft function, and subplots were employed to visualize both the original signal and its frequency domain representation. A similar approach was taken with a square wave, emphasizing the impact of different parameters on the results. The analysis revealed the significance of the sampling frequency in determining frequency component resolution. Through the utilization of MATLAB functions such as fft, sin, and square, students gained hands-on insights into signal processing, encouraging further exploration of parameters for a comprehensive understanding of Fourier analysis. This lab effectively bridged theoretical concepts with practical implementation, enhancing our understanding of signal processing in MATLAB.

PREMIER UNIVERSITY

Department of Computer Science & Engineering

Course Code	:	**EEE-202**
Course Title	:	**Signals & Systems Laboratory**
Report No	:	**06**
Name of Report	:	**Implementing Fourier transform in MATLAB**
Date of Performance	:	**23/12/2023**
Date of Submission	:	**20/01/2024**
Course Instructor	:	Mohammed Saifuddin Munna

Submitted By :

Name	: Arnab Shikder
ID	: 0222210005101098
Section	: C
Semester	: 4th Semester
Session	: Fall 2023

REMARKS

Experiment no: 06
Experiment name: Implementing Fourier transform in MATLAB
Objective: To understand and implement the Fourier transform using MATLAB.
Software Requirement: MATLAB 2014
Theory:
Fourier transform
The Fourier transform of the expression f = f(x) with respect to the variable x at the point w is defined as follows

Here, c and s are parameters of the Fourier transform. The Fourier function uses c = 1, s = –1.

Syntax

fourier(f,trans_var,eval_point)

Examples:
Compute the Fourier transform of this expression with respect to the variable x at the evaluation point y:
There Are two Types of Fourier transformation:

- For symbolic Value

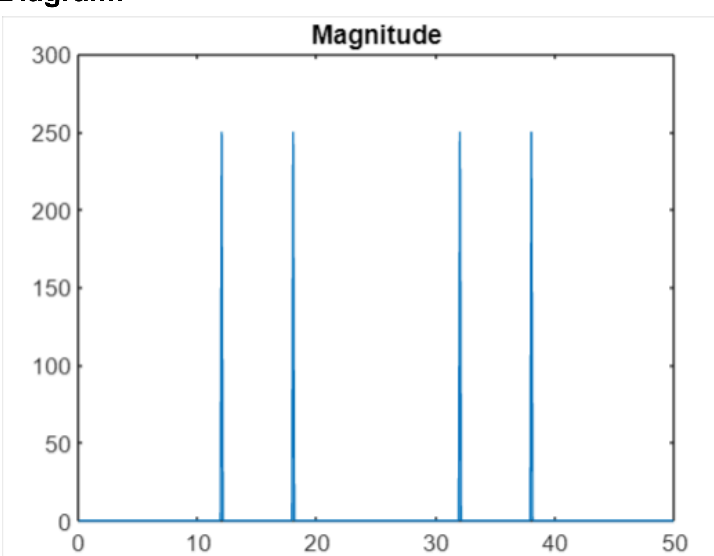
syms x y
f = exp(-x^2);
fourier(f, x, y)
ifourier(ans)

You can use it for any function:

- Real Value:
- **Complex wave**

```
t = 0:1/50:10-1/50;
x = sin(2*pi*12*t) + sin(2*pi*18*t);
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
plot(f, abs(y))
title('Magnitude')
```

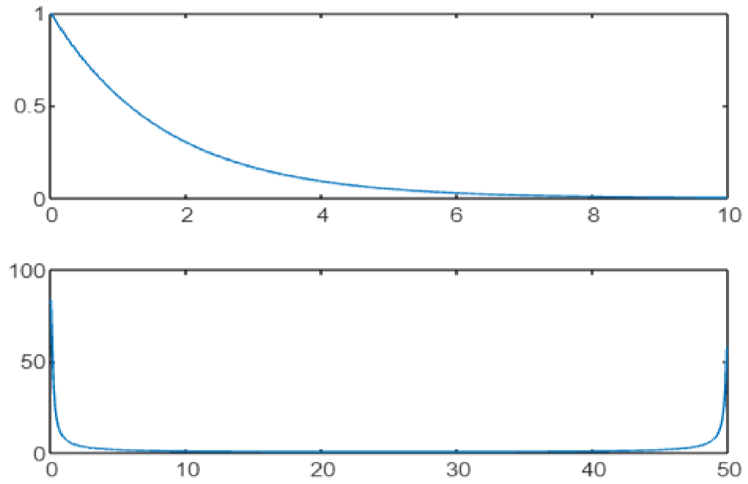
Diagram:



Exponential Functions:

```
t = 0:1/50:10-1/50;
x = exp(-0.6*t);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

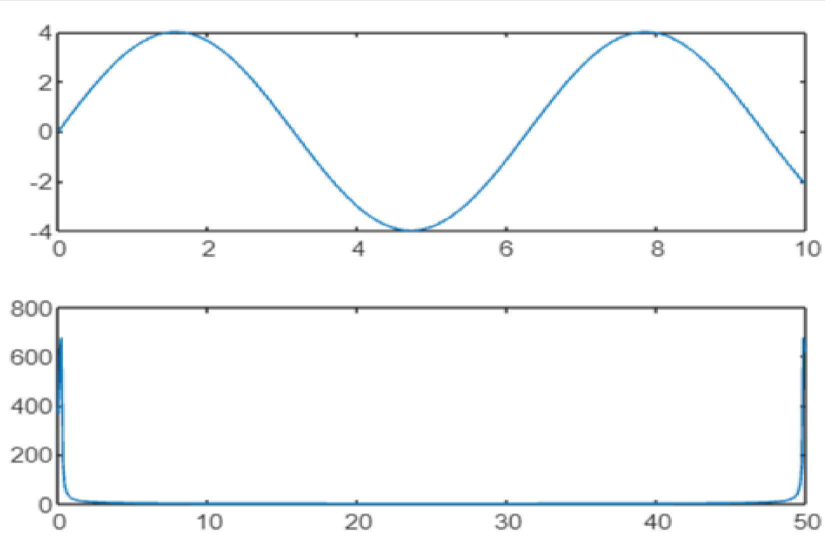
Diagram:



Simple Sin Function;

```
t = 0:1/50:10-1/50;
x = 4*sin(t);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

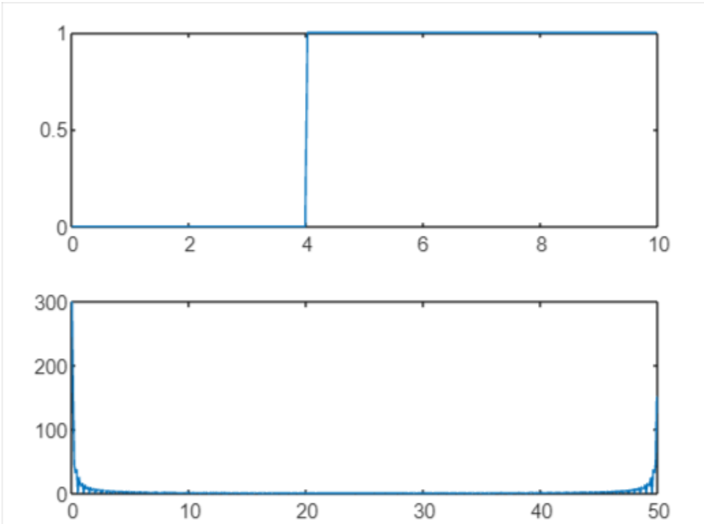
Diagram:



Unit Step Function:

```
t = 0:1/50:10-1/50;
x = heaviside(t - 4);
subplot(2,1,1)
plot(t, x)
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2)
plot(f, abs(y))
```

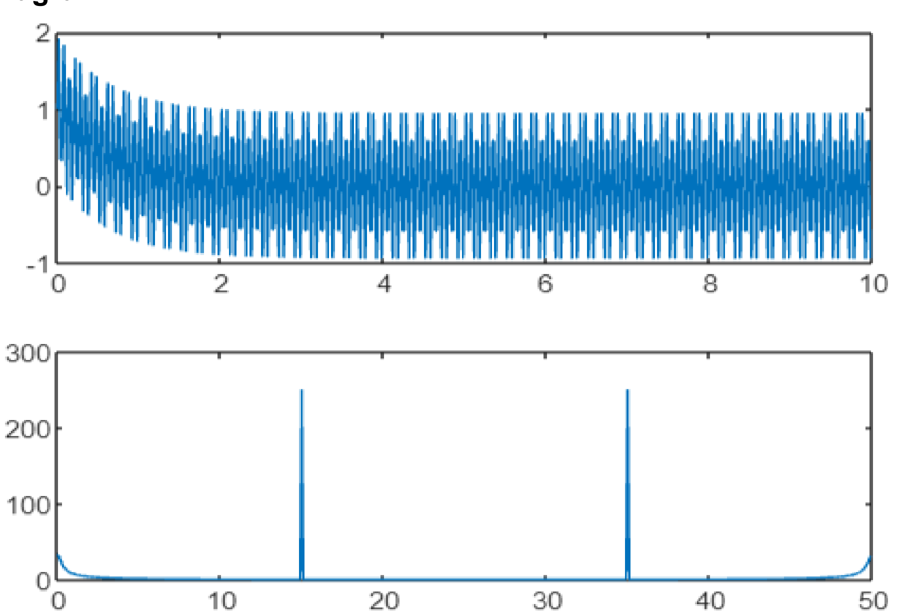
Diagram:



Task: Do the same for the ramp signal & any complex Wave mixed with sin function & exponential functions.

```
t = 0 : 1/50:10-1/50;
x = sin(2*pi*15*t)+exp(-1.5*t);
subplot(2,2,1);
subplot(2,1,1);
plot(t,x);
y = fft(x);
f = (0:length(y)-1)*50/length(y);
subplot(2,1,2);
plot(f,abs(y));
```

Diagram:



Discussion:

A signal is generated by combining a sine function and an exponential function. The original signal is plotted to visually assess its characteristics. The Fourier transform is then computed using the FFT algorithm, and the resulting frequency spectrum is displayed. Subsequently, the inverse Fourier transform is applied to reconstruct the signal from the frequency components. The reconstructed signal is plotted and compared with the original signal, providing insights into the accuracy of the Fourier representation. This code serves as a practical demonstration of implementing Fourier transform techniques in MATLAB for signal analysis and reconstruction. this MATLAB implementation demonstrates a systematic approach to Fourier analysis and signal reconstruction. The presented code is a valuable resource for users seeking hands-on experience with Fourier transforms, especially in the context of signal processing and frequency domain analysis.