

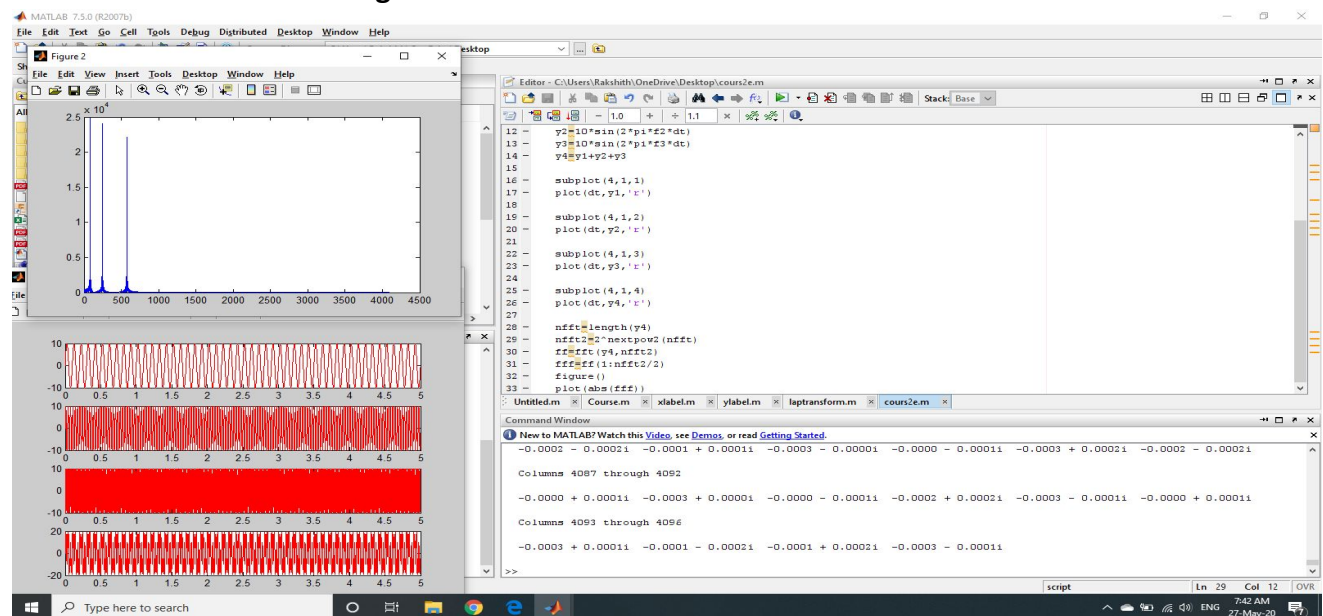
REPORT MAY 27

Date:	27 MAY 2020	Name:	Rakshith B
Course:	Digital Signal Processing	USN:	4AL16EC409
Topic:	FFT Using Matlab, Study and Analysis of FIR and IIR, Filtering Signal ECG Signal Analysis	Semester & Section:	6th SEM B
Github Repository:	Rakshith-B		

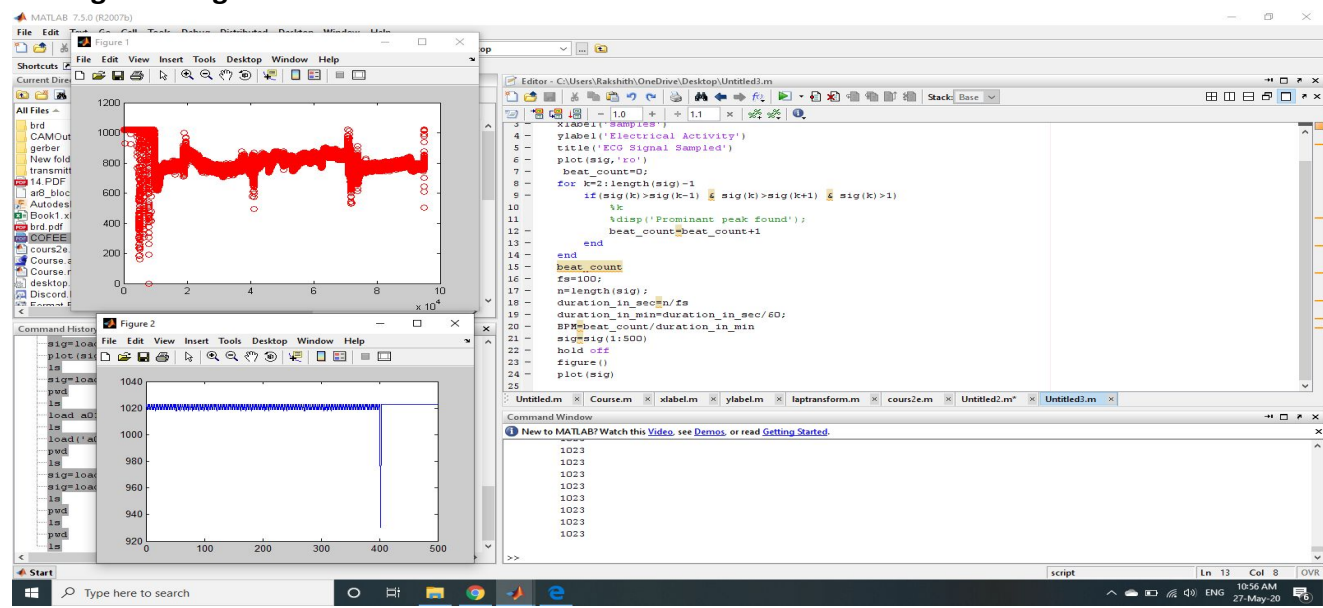
FORENOON SESSION DETAILS

Image of session

Fast Fourier Transform Using Matlab



ECG Signal Using Matlab



Report –

Fast Fourier Transform Using MATLAB

```
clear all;
close all;
clc;
fs=1000
ts=1/fs
dt=0:ts:5-ts
f1=10;
f2=30;
f3=70

y1=10*sin(2*pi*f1*dt)
y2=10*sin(2*pi*f2*dt)
y3=10*sin(2*pi*f3*dt)
y4=y1+y2+y3

subplot(4,1,1)
plot(dt,y1,'r')

subplot(4,1,2)
plot(dt,y2,'r')

subplot(4,1,3)
plot(dt,y3,'r')

subplot(4,1,4)
plot(dt,y4,'r')

nfft=length(y4)
nfft2=2^nextpow2(nfft)
ff=fft(y4,nfft2)
fff=ff(1:nfft2/2)
figure()
xfft=fs*(0:nfft2/2)/nfft2
plot(abs(fff))
```

ECG Signal Analysis Using MATLAB

```
sig=load('ecg.csv')
plot(sig)
xlabel('samples')
ylabel('Electrical Activity')
title('ECG Signal Sampled')
plot(sig,'ro')
beat_count=0;
for k=2:length(sig)-1
```

```
if(sig(k)>sig(k-1) & sig(k)>sig(k+1) & sig(k)>1)
    %k
    %disp('Prominant peak found');
    beat_count=beat_count+1
end
end
beat_count
fs=100;
n=length(sig);
duration_in_sec=n/fs
duration_in_min=duration_in_sec/60;
BPM=beat_count/duration_in_min
sig=sig(1:500)
hold off
figure()
plot(sig)
```

Date: 27 MAY 2020
Course: PYTHON On Udemy
Topic: GUI with Tkinter,Interfacing with databases

Name:RAKSHITH B
USN:4AL16EC409
Semester & Section:6 B

AFTERNOON SESSION DETAILS

Image of session:GUI with Tkinter

The screenshot displays the Visual Studio Code interface with a Python script named `script.py` open in the editor. The script is a Tkinter-based GUI application. It defines a function `km_to_miles` that takes a value from a text entry field, converts it to a float, and calculates the distance in miles. The GUI consists of a window titled `tk` with a text entry field and a button labeled `Execute`. The button is configured to call the `km_to_miles` function when clicked. The script also includes a main loop to keep the window open.

```
E> COURSE\GUI python> script.py ~
1 from tkinter import *
2 window=Tk()
3
4 def km_to_miles():
5     miles=float(e1_value.get())*1.6
6     t1.insert(END,miles)
7
8 b1=Button(window,text="Execute",command=km_to_miles)
9 b1.grid(row=0,column=0)
10
11 e1_value=StringVar()
12 e1=Entry(window,textvariable=e1_value)
13 e1.grid(row=0,column=1)
14
15 t1=Text(window,height=1,width=20)
16 t1.grid(row=0,column=2)
17 window.mainloop()
18
19
```

The `PROBLEMS` panel at the bottom shows the output of the script, indicating that the program ran successfully without any errors or warnings.

```
PS E:\COURSE\GUI python> python .\script.py
hey y=thetre
hey y=thetre
PS E:\COURSE\GUI python> python .\script.py
10
10
10
PS E:\COURSE\GUI python> python .\script.py
10
PS E:\COURSE\GUI python> python .\script.py
PS E:\COURSE\GUI python> python .\script.py
[]
```

[illegible]

Report –

GUI with Tkinter :

```
from tkinter import *

window=Tk()

def km_to_miles():
    miles=float(e1_value.get())*1.6
    t1.insert(END,miles)

b1=Button(window,text="Execute",command=km_to_miles)
b1.grid(row=0,column=0)

e1_value=StringVar()
e1=Entry(window,textvariable=e1_value)
e1.grid(row=0,column=1)

t1=Text(window,height=1,width=20)
t1.grid(row=0,column=2)

window.mainloop()
```

Interfacing with Database:

```
import sqlite3

def create_table():
    conn=sqlite3.connect("lite.db")
    cur=conn.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS store (item TEXT,quantity
INTEGER,price REAL)")
    conn.commit()
    conn.close()

def insert(item,quantity,price):
    conn=sqlite3.connect("lite.db")
    cur=conn.cursor()
    cur.execute("INSERT INTO store VALUES (?,?,?)",(item,quantity,price))
    conn.commit()
    conn.close()

insert("Plates",10,5)

def view():
```

```
conn=sqlite3.connect("lite.db")
cur=conn.cursor()
cur.execute("SELECT * FROM store")
rows=cur.fetchall()
conn.close()
return rows

print(view())
```