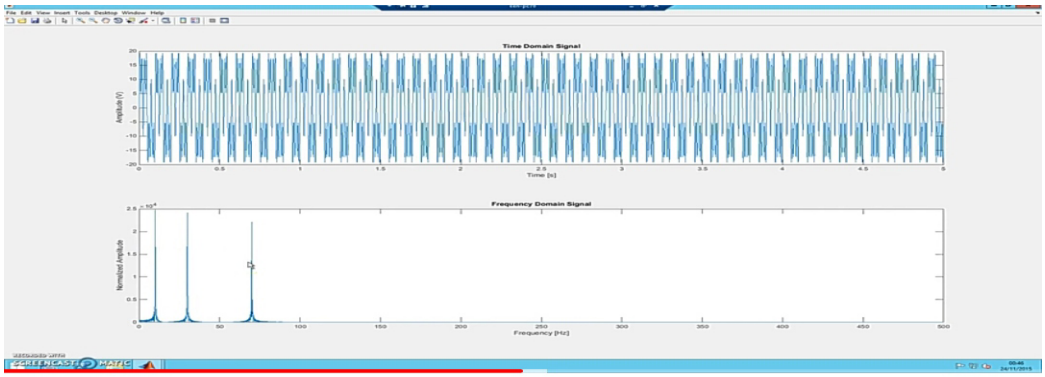
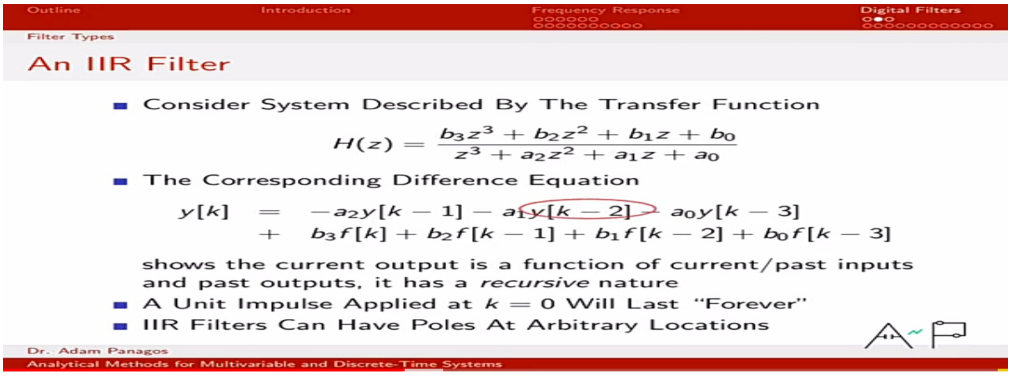
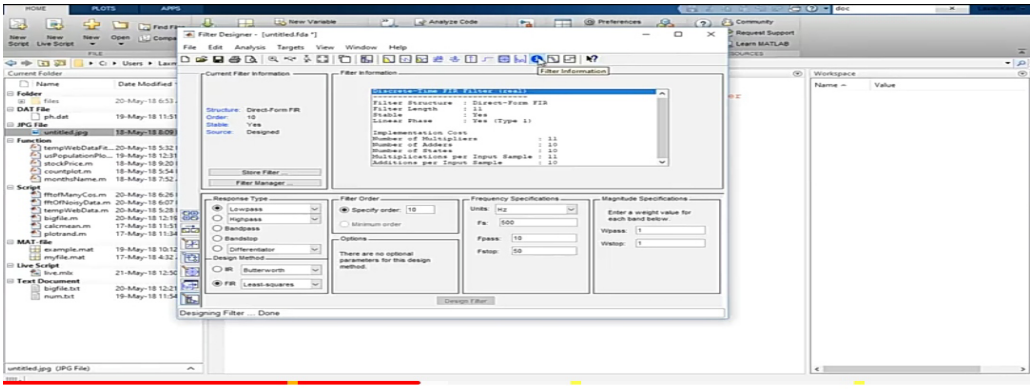
Date:	27-05-2020	Name:	Rajeshwari Gadagi
Course:	DSP	USN:	4AL17EC076
Topic:	Fourier transform,FFT USING matlab,wavelets,FIR IIR	Semester and section:	6th sem and B sec



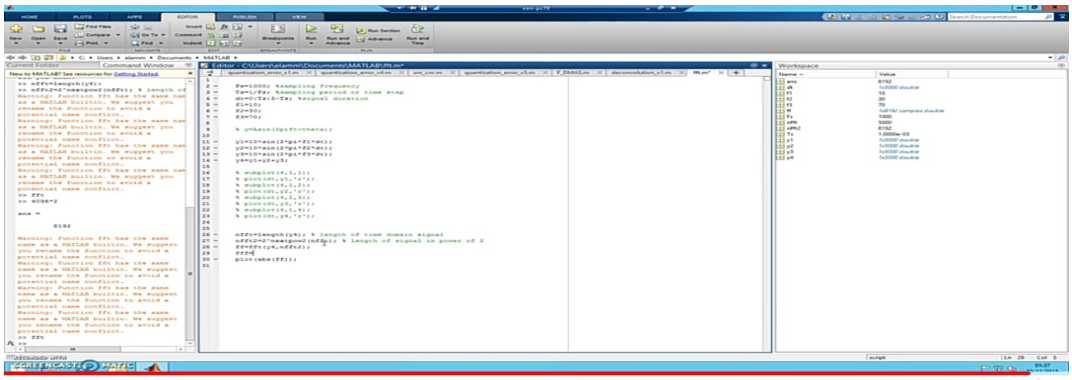
Simple and Easy Tutorial on FFT Fast Fourier Transform Matlab Part 2



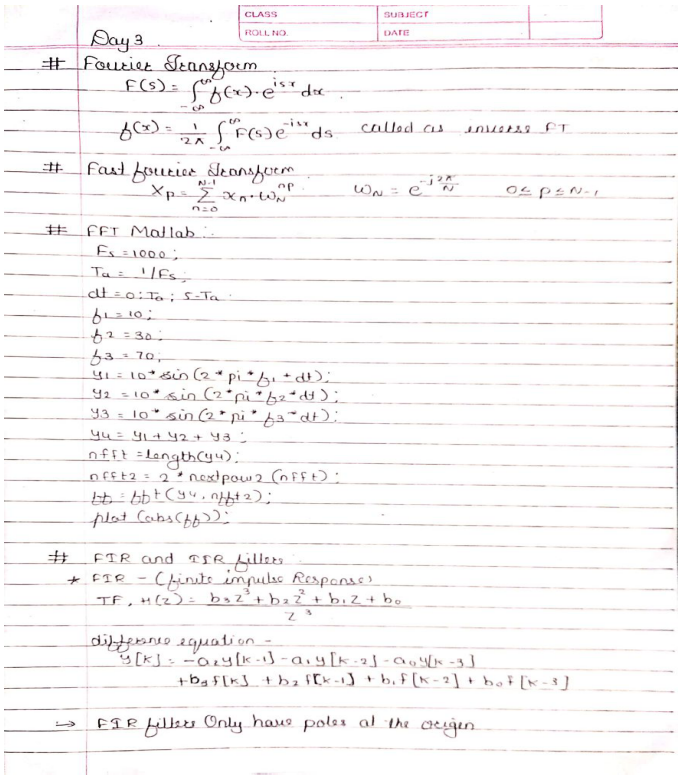
Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) Filters



42 MATLAB Beginners Tutorial- Low Pass Filter Design Using FDA Tool aka Filter Desi...



Simple and Easy Tutorial on FFT Fast Fourier Transform Matlab Part 1



Day 3

Fourier Transform

$$F(s) = \int_{-\infty}^{\infty} f(x) \cdot e^{isx}\, dx$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(s) e^{-isx}\, ds \quad \text{called as inverse FT}$$

Fast Fourier Transform

$$X_p = \sum_{n=0}^{N-1} x_n \cdot \omega_N^{np} \qquad \omega_N = e^{-j\frac{2\pi}{N}} \qquad 0 \le p \le N-1$$

FFT Matlab

```
Fs = 1000;
Ts = 1/Fs;
t = 0:Ts:1-Ts;
f1 = 10;
f2 = 30;
f3 = 70;
y1 = 10*sin(2*pi*f1*t);
y2 = 10*sin(2*pi*f2*t);
y3 = 10*sin(2*pi*f3*t);
y4 = y1 + y2 + y3;
nfft = length(y4);
nfft2 = 2^nextpow2(nfft);
ff = fft(y4, nfft2);
plot(abs(ff));
```

FIR and IIR filters

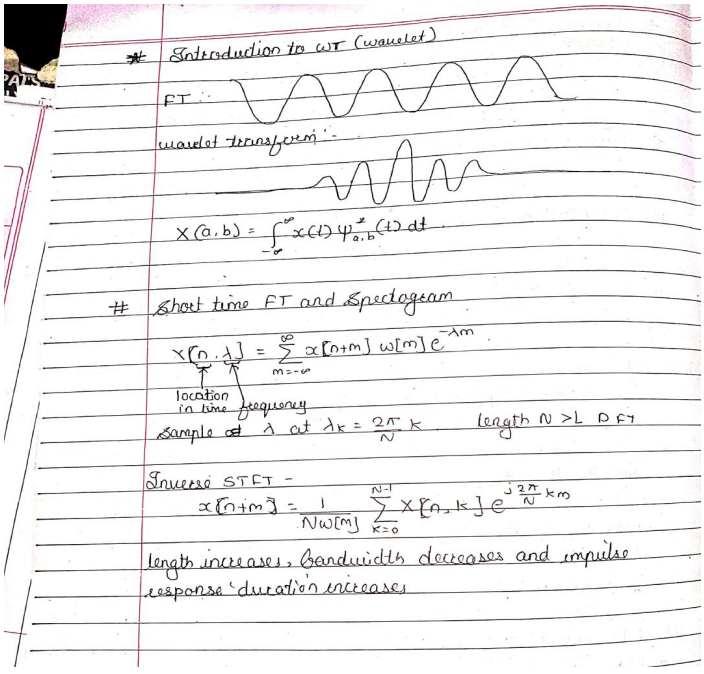
* FIR – (finite impulse Response)

TF, $H(z) = \frac{b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^3}$

difference equation –

$$y[k] = -a_2 y[k-1] - a_1 y[k-2] - a_0 y[k-3] + b_3 f[k] + b_2 f[k-1] + b_1 f[k-2] + b_0 f[k-3]$$

→ FIR filters Only have poles at the origin



Introduction to WT (wavelet)

FT :

wavelet transform :

$$X(a,b) = \int_{-\infty}^{\infty} x(t)\, \psi_{a,b}^{*}(t)\, dt$$

Short time FT and Spectogram

$$X[n, \lambda] = \sum_{m=-\infty}^{\infty} x[n+m]\, w[m]\, e^{-\lambda m}$$

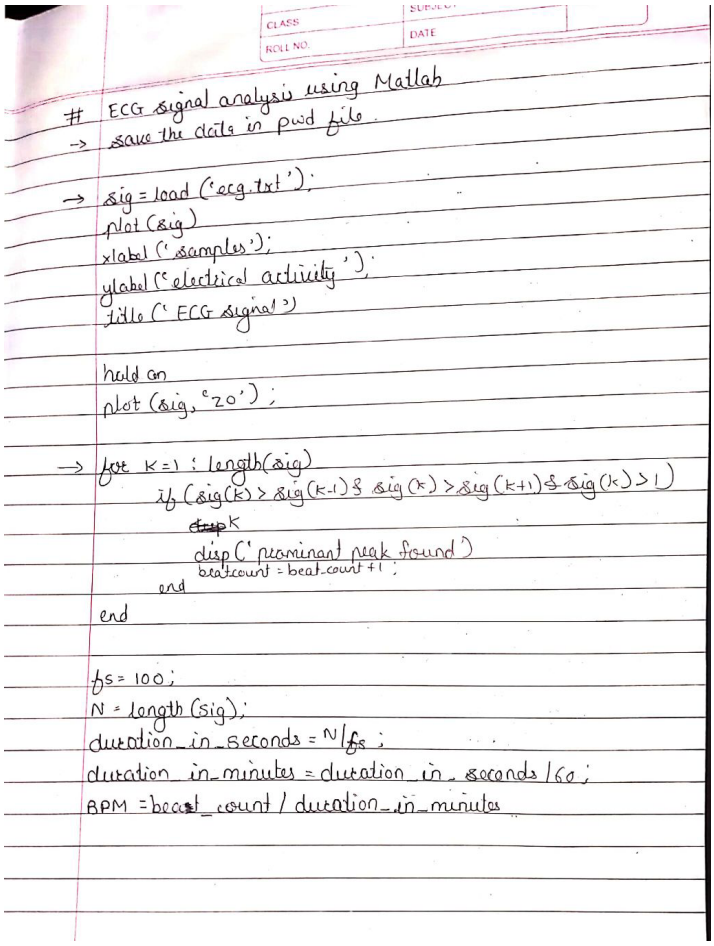
location in time, frequency

sample λ at $\lambda_k = \frac{2\pi}{N} k$ length $N > L$ DFT

Inverse STFT –

$$x[n+m] = \frac{1}{N w[m]} \sum_{k=0}^{N-1} X[n, k]\, e^{j\frac{2\pi}{N} km}$$

length increases, Bandwidth decreases and impulse response duration increases



```
# ECG signal analysis using Matlab
-> save the data in pwd file

-> sig = load ('ecg.txt');
plot (sig)
xlabel ('samples');
ylabel ('electrical activity');
title ('ECG signal')

hold on
plot (sig, 'zo');

-> for k=1 : length(sig)
    if (sig(k) > sig(k-1) & sig(k) > sig(k+1) & sig(k) > 1)
      k
      disp ('prominant peak found')
      beatcount = beat_count + 1;
    end
end

fs = 100;
N = length (sig);
duration_in_seconds = N/fs;
duration_in_minutes = duration_in_seconds / 60;
BPM = beat_count / duration_in_minutes
```

Date:	**27-05-2020**	**Name:**	**Rajeshwari Gadagi**
Course:	**Python programming**	**USN:**	**4AL17EC076**
Topic:	**Graphical user interface with tkinter,introduction to python with database.**	**Semester and section:**	**6th sem and B sec**

Day 8:

* Graphical User Interface with Tkinter

→
```
from tkinter import *
window = Tk()

b1 = Button(window, text = "Execute")
b1.grid(row = 0, column = 0)
e1 = Entry(window)
e1.grid(row = 0, column = 1)
t1 = Text(window, height = 1, width = 20)
t1.grid(row = 0, column = 2)

window.mainloop()
```

→
```
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()
```

★ Introduction to "Python with Databases".

```
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("coffee cup", 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())
```