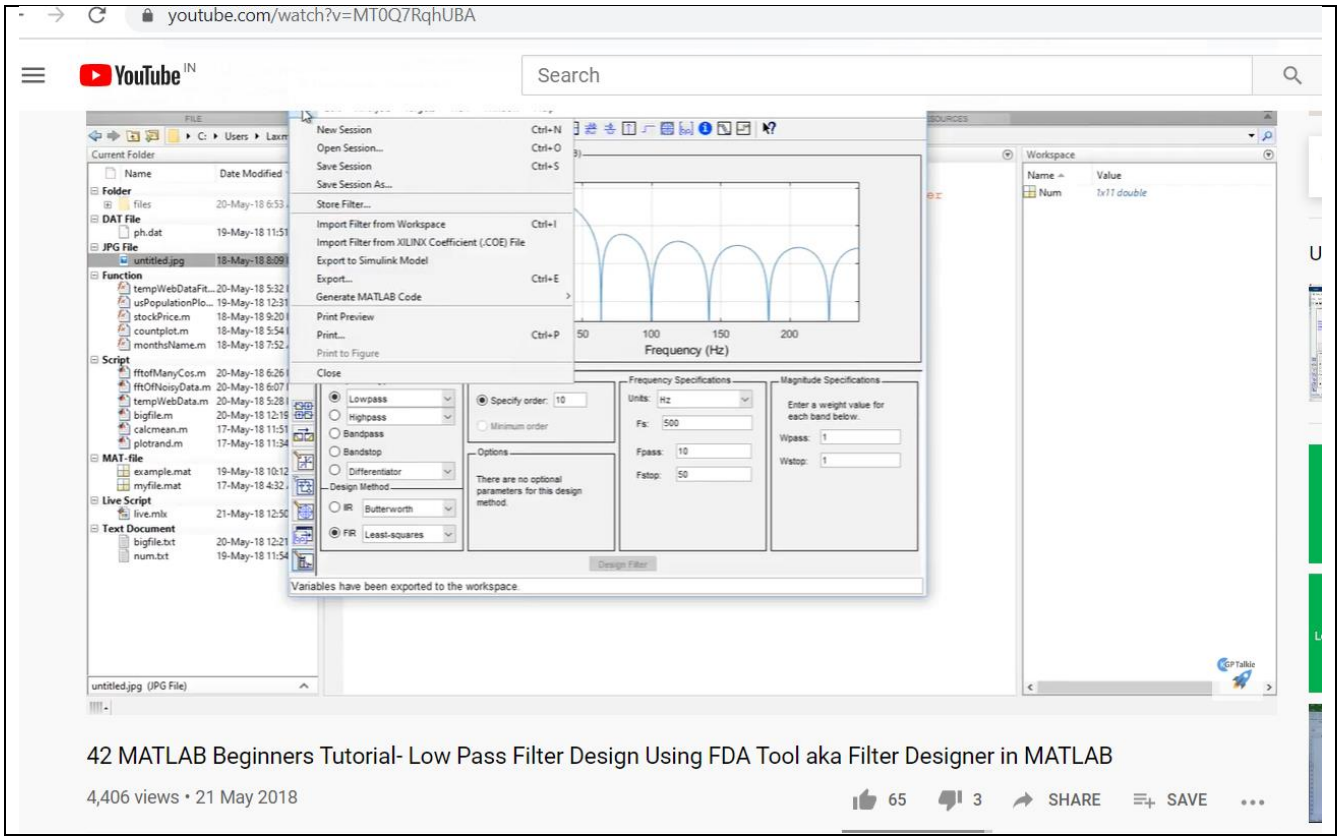
DAILY ASSESSMENT FORMAT

Date:	**27-05-2020**	**Name:**	**BINDUSHRI**
Course:	**dsp**	**USN:**	**4AL17EC011**
Topic:	**1.Fourier transform** **2.FFF fast fourier transform Matlab** **3.FFT** **4.FIR and IIR filter** **5.FIR and IIR using FDA tool in Matlab** **6.introducton to WT** **7.CWT & DWT** **8,.signal filtering using WTin Matlab** **9.short time fourier transform & spectrogram** **10.ECG signal analysis using Matlab.**	**Semester & Section:**	**6th A**
Github Repository:	**Bindushri**		

FORENOON SESSION DETAILS


youtube.com/watch?v=MT0Q7RqhUBA
YouTube IN
Search
New Session Ctrl+N
Open Session... Ctrl+O
Save Session Ctrl+S
Save Session As...
Store Filter...
Import Filter from Workspace Ctrl+I
Import Filter from XILINX Coefficient (.COE) File
Export to Simulink Model
Export... Ctrl+E
Generate MATLAB Code
Print Preview
Print... Ctrl+P
Print to Figure
Close
Frequency (Hz)
Variables have been exported to the workspace.
42 MATLAB Beginners Tutorial- Low Pass Filter Design Using FDA Tool aka Filter Designer in MATLAB
4,406 views • 21 May 2018
65
3
SHARE
SAVE

Day3 - 27-05-2020 Fourier transform

The function $F(s)$, defined by

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

↓

FT of $f(x)$.

Also function $f(x)$ defined by

$$f(x) = \frac{1}{2\pi} \int_{-\alpha}^{\alpha} F(s) \cdot e^{j s x} ds$$

↓ called IFT of $F(s)$

* FFT

DFT: $x_p = \sum_{n=0}^{N-1} x_n \cdot W_N^{np}$ $\quad 0 \le p \le N-1$

$W_N \triangleq e^{-j\frac{2\pi}{N}}$

$x_0 = x_0 e^{-j2\pi(0)(0)/4} + x_1 e^{-j2\pi(1)(0)/4} + x_0 e^{-j2\pi(2)(0)/4} + x_3 e^{-j2\pi(3)(0)/4}$.

$x_0 = x_0 + x_1 + x_2 + x_3$.

$x_1 = x_0 + x_1 e^{-j2\pi/4} - x_2 + x_3 e^{-j2\pi 3/4}$

$x_1 = x_0 - x_1 e^{-j2\pi/4} - x_2 - x_3 e^{-j2\pi/4}$.

$x_2 = x_0 - x_1 + x_2 - x_3$

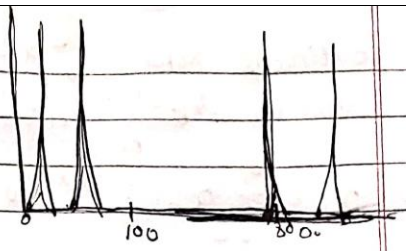
$x_3 = x_0 + x_1 e^{-j2\pi 3/4} - x_2 - x_3 e^{-j2\pi 3/4}$

FFT:- converts Time domain signal to frequency domain signal

* FFF fast fourier transform Matlab

```
fs=1000;  % sampling frequency
Ts = 1/fs; % sampling period or time step
dt = 0:Ts:2-Ts
f1 = 10;
f2 = 80;
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;
```

```
nfft = length(y4);
nfft2 = 2^nextpow2(nfft);
ff = fft(y4, nfft2);
plot(abs(ff));
```



FIR and IIR Filter

~~FIR filter~~ consider system described by the transfer function $H(z) = \frac{b_3z^3 + b_2z^2 + b_1z + b_0}{z^3}$

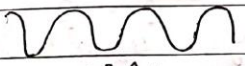
then

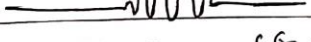
$$Y(z) = b_3 f(k) + b_2 f(k-1) + b_1 f(k-2) + b_0 f(k-3)$$

- Digital filters can be classified as Recursive or non-recursive. also called IIR or FIR

Study and analysis FIR and IIR using FDA tool in matlab

Introduction to WT

FT → 

WT → 

Correlation :- $X(a,b) = \int_{-\infty}^{\infty} x(t)\, \psi^*_{a,b}(t)\, dt$.

Vanishing moments :- $m_k = \int_{-\infty}^{\infty} f(x)\, x^k dx$

moment k vanishes if integral is zero

higher number of vanishing moments = more complex

CWT & DWT

$$f(t) = \frac{1}{2} a_0 + \sum_{k=1}^{\infty} (a_k \cos 2\pi kt + b_k \sin 2\pi kt$$

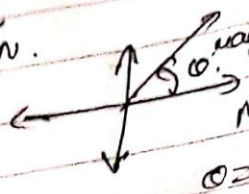
$$FT: X(F) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi ft}\, dt.$$

DFT continuous $X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$

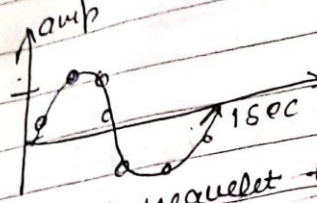
discrete: $X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{j2\pi kn}{N}}$

$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi kn/N}$.

$x_k = A_k + B_k j$ →



Ex: Sinwave: 1Hz, Amp = 1, Sampling frequency: 8Hz



· continuous wavelet transform

$f(t) * \phi(\frac{t}{s}) = \int f(\tau) \cdot \phi(s(t-\tau)\, d\tau)$

$f(t) \cdot \phi(\frac{t-u}{s})$ → Study of function at various location

$CWT(\tau, s) = \langle f(t), \psi(\frac{t-\tau}{s}) \rangle$

STFT $\langle f(t), g(t-u) e^{jst} \rangle$

$f(t) = \frac{1}{C\psi} \langle CWT, \psi(\frac{t-\tau}{s}) \rangle$

$f(t)$ —— [$\psi(t/s)$] → $CWT(t, s)$

Axes.

By Parsevals theorem:

$CWT(\tau, s) = \int f(t) \cdot \psi^*(\frac{t-\tau}{s})\, dt = \frac{1}{2\pi} \int F(\omega) \cdot \Psi^*(s\omega)\, d\omega$

* Implementation of signal filtering signal using wrin Matlab

1. clear all;
2. (x, Fs) = audioread ('man_noise.wav');

K=K * 0.5 / rms(K);
K= awgn(K, 12, 'measured');
[C,L] = wavedec(K,3, 'db4');
sound (K, Fs)   # hearing the voice
a(i) >> sound (K, Fs)

b = wthresh (C, 's', 0.15);
Y = waverec (b, L, 'db4');
Y = Y * 0.5 / rms (Y);
sound (Y, Fs);

* Short-time Fourier Transform & Spectrogram

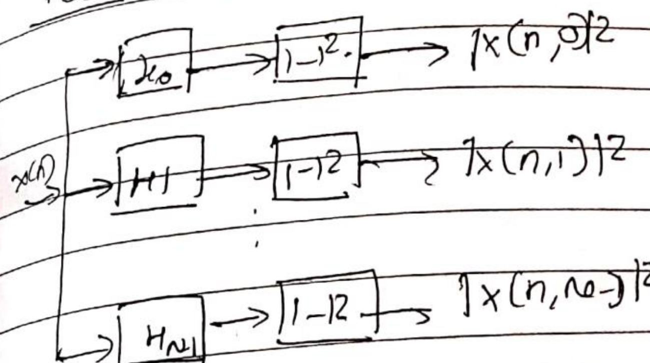
$ISTFT = x(n+m) = \frac{1}{N w(m)} \sum_{k=0}^{N-1} X(n,k)\, e^{j\frac{2\pi}{N} km}$

Saxophone Riff

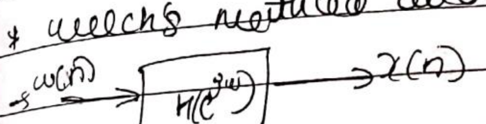
f_s = 44,100 KHz   L = 512 pts / segment N = 2048 freq samples

.192 points

FBI [Filter Bank interpretation]



* Welch's method and windowing.



$S_{xx}(\omega) = |H(e^{j\omega})|^2.$

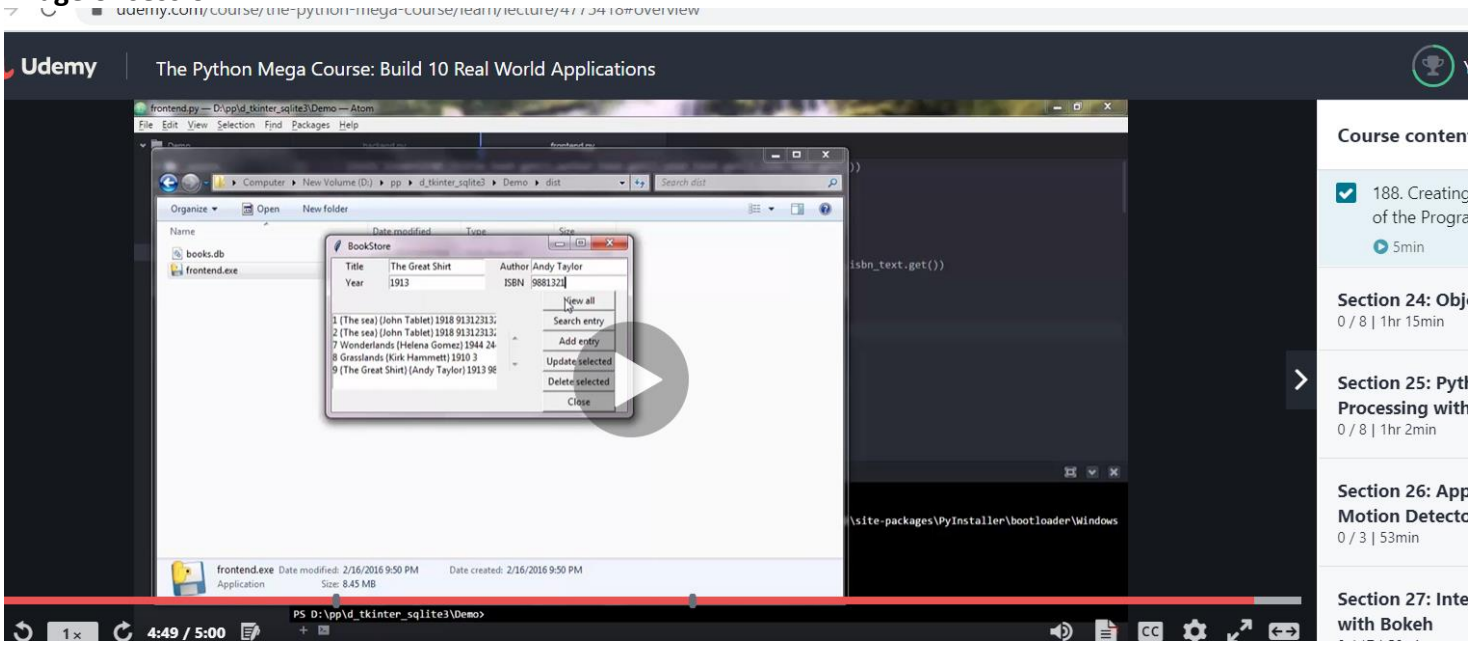
$S_{xx}(f) = |H(e^{j2\pi f/f_s})|^2.$

* ECG signal ~~~~ Analysis using matlab

Date:27may2020 **Name:Bindushri**
Course: python **USN:4AL17EC011**
Topic: **Sem&Sec:6th A**
Basics:sec23

AFTERNOON SESSION DETAILS

Image of session



Date: 07-05-2020 : A BUPLA a Desk top data base library

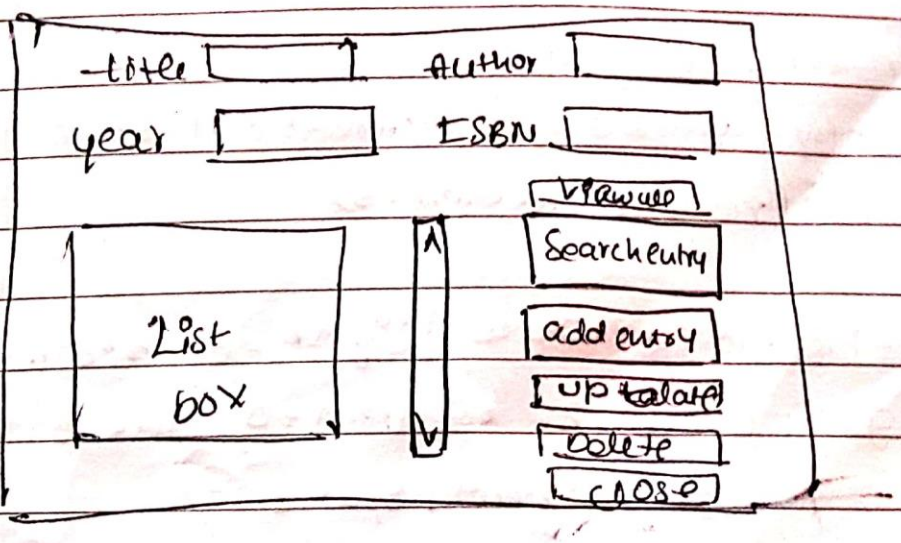
Code has 2 parts

front end :- code of tkinter library

Back end :- sqlite3 database code

Settings → wrap-guide (Enable)

Sketch labelled program



scrapt.py

Demo -> frontend.py

```
1. from tkinter import *
   import backend  -> def get_selected_row(event):
2.                    index = lst1.curse selection()[0]
   def view_command():
3.                    lst1.delete(0, END)
                      for row in backend.view():
                          lst1.insert(END, row)
4. window = Tk()
5. l1 = Label(window, text = "Title")
6. l1.grid(row=0, column=0)
7.
8. l2 = Label(window, text = "Author")
9. l2.grid(row=0, column=2)
10.
11. l3 = Label(window, text = "year")
12. l3.grid(row=1, column=0)
13.
14. l4 = Label(window, text = "ISBN")
15. l4.grid(row=1, column=2)
16.
17. title_text = StringVar()
18. e1 = Entry(window, textvariable = title_text)
19. e1.grid(row=0, column=1)
20. author_text = StringVar()
21. e2 = Entry(window, textvariable = author_text)
22. e2.grid(row=0, column=3)
23. year_text = StringVar()
24. e3 = Entry(window, textvariable = year_text)
25. e3.grid(row=1, column=1)
26.
27. isbn_text = StringVar()
28. e4 = Entry(window, textvariable = isbn_text)
29. e4.grid(row=1, column=3)
30.
31. lst1 = Listbox(window, height = 6, width = 35)
32. lst1.grid(row=2, column=0, rowspan=6,
                columnspan=2)
```

sbl=Scrollbar(window)
sbl.grid(row=2, column=2, rowspan=6)

lst1.configure(yscrollcommand=sbl.set)
sbl.configure(command=lst1.yview)
lst1.bind('<<ListboxSelect>>', get_selected_row)

b1=Button(window, text="View all", width=12, command=view_command)
b1.grid(row=3, column=3)

b2=Button(window, text="Search entry", width=12, command=search_command)
b2.grid(row=4, column=3)

b3=Button(window, text="Add entry", width=12, command=add_command)
b3.grid(row=5, column=3)

b4=Button(window, text="Update selected", width=12, command=update_command)
b4.grid(row=6, column=3)

b5=Button(window, text="Delete selected", width=12, command=delete_command)
b5.grid(row=7, column=3)

b6=Button(window, text="Close", width=12, command=window.destroy)
b6.grid(row=8, column=3)
window.mainloop()

Page → Backend.py

```
1 import sqlite3

2 def connect():
3     conn=sqlite3.connect("Books.db")
4     cur=conn.cursor()
```

Python ./backend.py

```
cur.execute("CREATE TABLE IF NOT EXISTS book
            (id INTEGER PRIMARY KEY, title text,
            author text, year integer, isbn integer)")

conn.commit()
conn.close()

def Insert(title, author, year, isbn)
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("INSERT INTO book VALUES (NULL
                ?,?,?,?)", (title, author, year,
                isbn))

    conn.commit()
    conn.close()

def view():
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book")
    rows=cur.fetchall()
    conn.close()
    return rows

Insert("The Sea", "John Tablet", 1918, 91)
def Search(title="", author="", year="", isbn="")
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book WHERE
        title=? OR author=? OR isbn=?", (title,
            author, year, isbn))
    rows=cur.fetchall()
    conn.close()
    return rows
```

```
def delete(id):
    conn = sqlite3.connect("books.db")
    cur = conn.cursor()
    cur.execute("DELETE FROM book WHERE
    id = ?", (id,))
    conn.commit()
    conn.close()

def update(id, title, author, year, isbn):
    conn = sqlite3.connect("book.db")
    cur = conn.cursor()
    cur.execute("UPDATE book SET title = ?, author = ?
        year = ?, isbn = ? WHERE id = ?", (
        title, author, year, isbn, id))
    conn.commit()
    conn.close()
connect()
```

Connecting frontend to Backend part 1

→ # Before line # 4 in frontend() add at the top of Program

```
def search_command():
    lst1.delete(0, END)
    for row in backend.search(title_text.get(),
        author_text.get(), year_text.
        get(), isbn_text.get()):
    lst.insert(END, row)

def add_command():
    backend.insert(title_text.get(), author_text.
        get(), year_text.get(), isbn_text.
        get())
    lst1.delete(0, END)
    lst1.insert(END, title_text.get(), author_text,
        get, year_text.get(), isbn_text.get())
```

def delet_comand():
    backend.deleted(get_selected_tuple()[0])

def update_command():
    backend.update(selected_tuple[0], selected_tuple[1],
    selected_tuple[2], selected_tuple[3],
    selected_tuple[4])
    print(selected_tuple[0], selected_tuple[1], selected_
    tuple[2], selected_tuple[3], selected_tuple[4])

baichend.update(selected_tuple[0], title_text.get(),
    author_text.get(), year_text.get(),
    isbn_text.get())