

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

FILE

Current Folder

Folder

DAT File

JPG File

Function

Script

MAT-file

Live Script

Text Document

untitled.jpg (JPG File)

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

Lowpass
Specify order: 10
Minimum order

Options

Design Method

FIR Least-squares

Frequency Specifications

Units: Hz
Fs: 500
Fpass: 10
Fstop: 50

Magnitude Specifications

Enter a weight value for each band below.
Wpass: 1
Wstop: 1

Variables have been exported to the workspace.

Workspace

Name	Value
Num	1x1717 double

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 ...

Day 3 - 27-05-2020

Fourier transform

the function $F(s)$, defined by

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

↓
FT of $f(x)$.

also function $f(x)$ defined by

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

↓ called IFT of $F(s)$

* FFT

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

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

$$X_0 = x_0 e^{-j2\pi(0)(0)/4} + x_1 e^{-j2\pi(1)(0)/4} + x_2 e^{-j2\pi(2)(0)/4} + x_3 e^{-j2\pi(3)(0)/4}$$

$$X_0 = x_0 + x_1 + x_2 + x_3$$

$$X_1 = x_0 + x_1 e^{-j2\pi/4} - x_2 + x_3 e^{-j2\pi/4}$$

$$X_2 = x_0 - x_1 + x_2 - x_3$$

$$X_3 = x_0 + x_1 e^{-j2\pi/4} - x_2 - x_3 e^{-j2\pi/4}$$

FFT: converts

Time domain signal

to Frequency domain

signal

* FFT fast Fourier transform Matlab

$F_s = 1000$; % sampling frequency

$T_s = 1/F_s$; % sampling period or time step

$\Delta t = 0.9 \times T_s$; $2 - T_s$

$N = 10$;

$f_2 = 80$;

$f_3 = 70$;

$$y_1 = 10 * \sin(2 * \pi * f_1 * \Delta t);$$

$$y_2 = 10 * \sin(2 * \pi * f_2 * \Delta t);$$

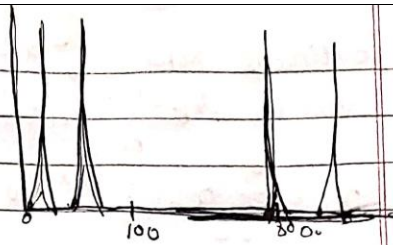
$$y_3 = 10 * \sin(2 * \pi * f_3 * \Delta t);$$

$$y_4 = y_1 + y_2 + y_3;$$

```

nfft = length(y4);
nfft2 = 2 * nextpow2(nfft);
ft = fft(y4, nfft2);
plot(abs(ft));

```



4 FIR and IIR Filter

For filter consider system described by the transfer function $H(z) = \frac{b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^3}$

then

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

- Digital filters can be classified as Recursive or Non-recursive. Also called IIR or FIR

* Study and analyse FIR and IIR using FFT tool in MATLAB

1. Introduction to WT

$f_t \rightarrow$

$w_t \rightarrow$

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

vanishing moments $\therefore m_k = \int_{-\infty}^{\infty} f(x) x^k dx$

Moment k vanishes if integral is zero.

Higher number of vanishing moments = more complex

* CWT & DWT

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

$$FT: x(f) = \int_{-\infty}^{\infty} x(t) e^{j 2\pi f t} dt$$

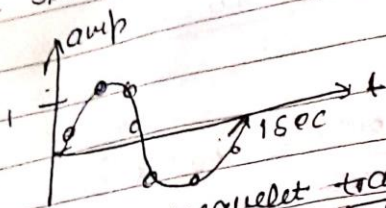
DTT continuous $x(t) = \int_{-\infty}^{\infty} x(t) e^{j2\pi f t} dt$
 discrete $x_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi f n / N}$

$$x_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi f n / N}$$

$$x_k = A_k + B_k j \rightarrow$$

$\omega = \tan^{-1} \frac{B_k}{A_k}$
 $\text{mag} = \sqrt{A_k^2 + B_k^2}$

ex: sinusoid: 1Hz, Amp=1, sampling frequency: 8Hz.



continuous wavelet transform
 $f(t) \cdot \phi\left(\frac{t-\tau}{s}\right) \rightarrow \int f(\tau) \cdot \phi\left(\frac{t-\tau}{s}\right) d\tau$

Study of function at various resolution

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

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

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

$$f(t) \xrightarrow{\psi(t/s)} CWT(t/s)$$

Parseval's theorem:

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

* Implementation of signal filtering signal using wtn in Matlab

1. ~~clear~~ clear all;
2. (k,fs) = audioread('mm_noise.wav');

$K = K + 0.5 / \text{rms}(k);$

$K = \text{awgn}(K, 12, 'measured');$

$[L, J] = \text{wavread}(K, B, 'dB');$

$\text{plot}(B, \text{sound}(K, Fs))$ # hearing the noise

$b = \text{wthresh}(L, 's', 0.13);$

$y = \text{wavwrite}(b, 1, 'dB4');$

$y = y * 0.5 / \text{rms}(y);$

$\text{sound}(y, Fs);$

* Short-time Fourier Transform & Spectrogram

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

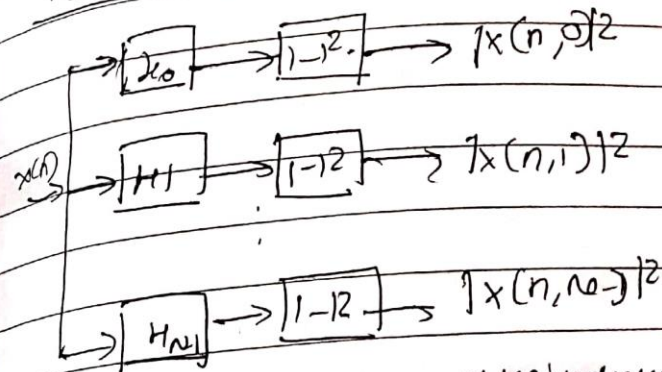
Saxophone Riff

$F_s = 44,100 \text{ KHz}$

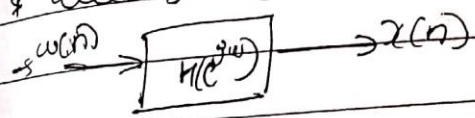
$L = 812 \text{ pts / segment}$ No. 2048 frequency

192 points

FBT [Filter Bank Interpretation]



* Welch's method and windowing



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

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

* ECG signal analysis using matlab

Date:27may2020
Course: python
Topic:
Basics:sec23

Name:Bindushri
USN:4AL17EC011
Sem&Sec:6th A

AFTERNOON SESSION DETAILS

Image of session

Udemy | The Python Mega Course: Build 10 Real World Applications

188. Creating of the Progra 5min

Section 24: Obj 0 / 8 | 1hr 15min

Section 25: Pytl Processing with 0 / 8 | 1hr 2min

Section 26: App Motion Detecto 0 / 3 | 53min

Section 27: Inte with Bokeh

BookStore

Title	Year	Author	ISBN
1 (The sea) (John Tabiet)	1918	91312313	
2 (The sea) (John Tabiet)	1918	91312313	
7 Wonderlands (Helena Gomez)	1944	24	
8 Grasslands (Kirk Hammett)	1910	3	
9 (The Great Shirt) (Andy Taylor)	1913	98	

22/03: 27-05-2020 : A Build a Desk top ^{data base application} application
Code has 2 parts

front end :- code of tkinter library
Back end :- SQLites database code.

Settings → wrap-guide (enable)

Sketch
to build
program

The sketch shows a rectangular window with a title bar. Inside the window, there are two input fields at the top: "title" and "Author". Below them are "year" and "ISBN" input fields. To the left of the center is a large rectangular area labeled "List box". To the right of the "List box" is a vertical scrollbar. On the far right, there is a vertical stack of buttons: "view all", "Search entry", "add entry", "up to date", "delete", and "close".

Demo → frontend.py

```

1. from tkinter import *
   import backend
2. import backend
   def view_command():
       lst1.delete(0, END)
       for row in backend.view():
           lst1.insert(END, row)
3. def view_command():
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=1)
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=1)
16.
17. title_text = StringVar()
18. e1 = Entry(window, textvariable=title_text)
19. e1.grid(row=0, column=2)
20. author_text = StringVar()
21. e2 = Entry(window, textvariable=author_text)
22. e2.grid(row=0, column=3)
23.
24. year_text = StringVar()
25. e3 = Entry(window, textvariable=year_text)
26. e3.grid(row=1, column=1)
27.
28. isbn_text = StringVar()
29. e4 = Entry(window, textvariable=isbn_text)
30. e4.grid(row=1, column=3)
31.
32. def view_command():
33. def view_command():
34. def view_command():
35. def view_command():
36. def view_command():
37. def view_command():
38. def view_command():
39. def view_command():
40. def view_command():
41. def view_command():
42. def view_command():
43. def view_command():
44. def view_command():
45. def view_command():
46. def view_command():
47. def view_command():
48. def view_command():
49. def view_command():
50. def view_command():
51. def view_command():
52. def view_command():
53. def view_command():
54. def view_command():
55. def view_command():
56. def view_command():
57. def view_command():
58. def view_command():
59. def view_command():
60. def view_command():
61. def view_command():
62. def view_command():
63. def view_command():
64. def view_command():
65. def view_command():
66. def view_command():
67. def view_command():
68. def view_command():
69. def view_command():
70. def view_command():
71. def view_command():
72. def view_command():
73. def view_command():
74. def view_command():
75. def view_command():
76. def view_command():
77. def view_command():
78. def view_command():
79. def view_command():
80. def view_command():
81. def view_command():
82. def view_command():
83. def view_command():
84. def view_command():
85. def view_command():
86. def view_command():
87. def view_command():
88. def view_command():
89. def view_command():
90. def view_command():
91. def view_command():
92. def view_command():
93. def view_command():
94. def view_command():
95. def view_command():
96. def view_command():
97. def view_command():
98. def view_command():
99. def view_command():
100. def view_command():

```

sb1 = scrollbar(window)

sb1.grid(row=2, column=2, rowspan=6)

opst1.config(cursor=(yscrollcommand-sb1.get))

sb1.config(command=opst1.yview)

opst1.bind("<1stboxselect>", gd_selected_row)

b1 = Button(window, text="Update all", height=12, command=b2buttoncommand)

b1.grid(row=2, column=3)

b2 = Button(window, text="Launch new", height=12, command=searchcommand)

b2.grid(row=3, column=3)

b3 = Button(window, text="Total", width=20, command=add_command)

b3.grid(row=4, column=3)

b4 = Button(window, text="Update selected", height=12, command=update_command)

b4.grid(row=5, column=3)

~~main~~

b5 = Button(window, text="Delete selected", width=20, command=main_delete_command)

b5.grid(row=6, column=3)

b6 = Button(window, text="Close", width=12, command=window.destroy)

b6.grid(row=7, column=3)

window.mainloop()

Part 2 > Backend .py

1 import sqlite3

2 def connect():

3 conn = sqlite3.connect("Books.db")

4 cur = conn.cursor()

python. • \backend.py

Date _____
Page _____

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

```
connect()
```

```
insert("The Sea", "John Galsworthy", 1918, 9)
```

```
def search(title = " ", author = " ", year = " ", isbn = " ")  
    conn = sqlite3.connect("books.db")  
    cur = conn.cursor()  
    cur.execute("SELECT * FROM book WHERE  
        title = ? OR author = ? OR year = ? OR isbn = ?", (title,  
        author, year, isbn))  
    rows = cur.fetchall()  
    conn.close()  
    return rows
```


def delete (pd):

conn = sqlite3.connect("book.db")

cur = conn.cursor()

cur.execute("DELETE FROM book WHERE
pd = ?", (pd,))

conn.commit()

conn.close()

def update (pd, title, author, year, isbn):

conn = sqlite3.connect("book.db")

cur = conn.cursor()

cur.execute("UPDATE book SET title = ?, author = ?,
year = ?, isbn = ? WHERE pd = ?", (pd,
title, author, year, isbn))

conn.commit()

conn.close()

connect()

Connecting front end to Backend part

→ Before line 4 in front end() add at the top of program

def search_command():

l1.delete(0, END)

for row in backend.search(title_text.get(),
author_text.get(), year_text.
get(), isbn_text.get()):

l1.insert(END, row)

def add_command():

backend.insert(title_text.get(), author_text.
get(), year_text.get(), isbn_text.
get())

l1.delete(0, END)

l1.insert(END, (title_text.get(), author_text.
get(), year_text.get(), isbn_text.get()))

```
def delet_command():
```

```
    backend.deleted(tupleget_selectedtuple() [0])
```

```
def update update command():
```

```
    (backend.deleted 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])
```

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



