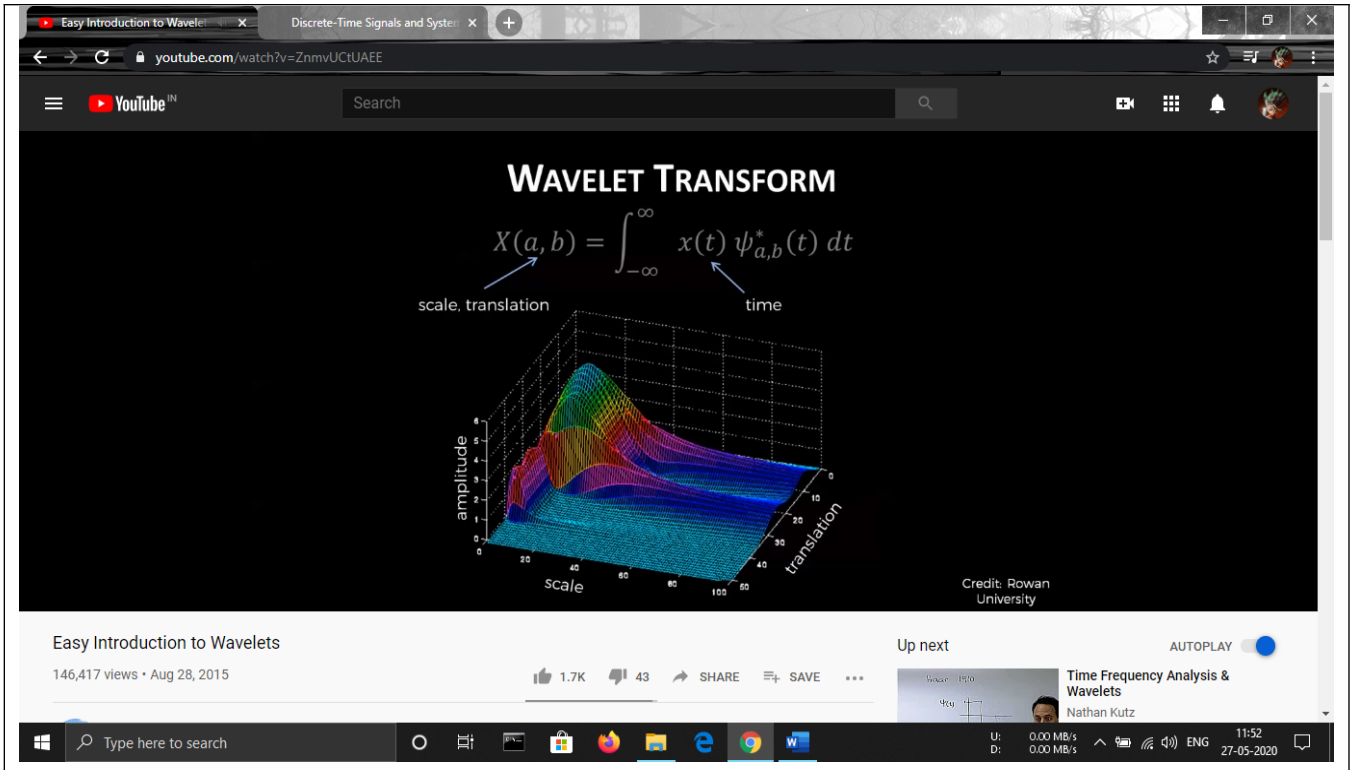
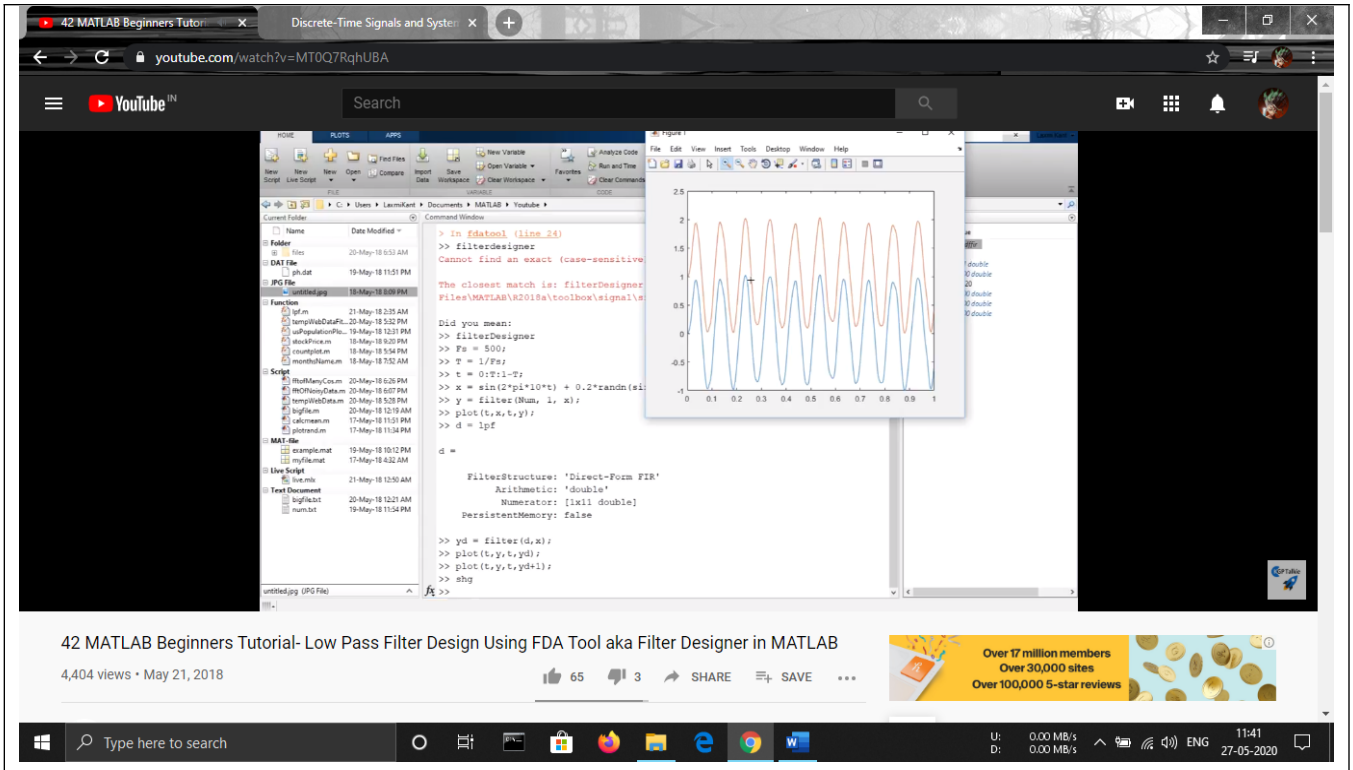
DAILY ASSESSMENT REPORT

Date:	27/05/2020	Name:	Abhishek
Subject:	Digital Signal Processing	USN:	4AL17EC001
Topic:	1] FFT, Fast Fourier Transform MATLAB 2] FIR and IIR Filters 3] Study and analysis FIR and IIR using FDA tool in MATLAB 4] Introduction to WT 5] CWT & DWT 6] Implementation of signal Filtering signal using WT in MATLAB 7] Short-time Fourier Transform and the Spectrogram 8] Welch's method and windowing 9] ECG Signal Analysis Using MATLAB	Semester & Section:	6th 'A'
Github Repository:	Abhishek-online-courses		

FORENOON SESSION DETAILS
Image of session


Easy Introduction to Wavelet
Discrete-Time Signals and Syster
youtube.com/watch?v=ZnmvUCtUAEE
YouTube IN
Search
WAVELET TRANSFORM
$X(a,b) = \int_{-\infty}^{\infty} x(t)\,\psi^*_{a,b}(t)\,dt$
scale, translation
time
amplitude
scale
translation
Credit: Rowan University
Easy Introduction to Wavelets
146,417 views • Aug 28, 2015
1.7K
43
SHARE
SAVE
Up next
AUTOPLAY
Time Frequency Analysis & Wavelets
Nathan Kutz
Type here to search
U:
D:
0.00 MB/s
0.00 MB/s
ENG
11:52
27-05-2020



Report

Fast Fourier Transform (FFT)

A **fast Fourier transform** (**FFT**) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT).

Basically, there are two types of FFT algorithm.

- Decimation in time algorithm (DIT-FFT).

The radix-2 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) _____equation into two parts: a sum over the even-numbered discrete-time indices n=[0,2,4,…,N−2] and a sum over the odd-numbered indices n=[1,3,5,…,N−1] as in Equation:

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) e^{-\left(i\frac{2\pi nk}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi\times(2n)k}{N}\right)} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} + e^{-\left(i\frac{2\pi k}{N}\right)} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} \\
&= \mathrm{DFT}_{\frac{N}{2}} \left[[x(0), x(2), \ldots, x(N-2)]\right] + W_N^k \, \mathrm{DFT}_{\frac{N}{2}} \left[[x(1), x(3), \ldots, x(N-1)]\right]
\end{aligned}
$$

- Decimation in frequency algorithm (DIF-FFT).

The radix-2 decimation-in-frequency algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: computation of the even-numbered discrete-frequency indices X(k) for k=[0,2,4,…,N−2] (or X(2r)) and computation of the odd numbered indices k=[1,3,5,…,N−1] (or X(2r+1)).

$$
\begin{aligned}
X(2r) &= \sum_{n=0}^{N-1} x(n) W_N^{2rn} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2r\left(n+\frac{N}{2}\right)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2rn} 1 \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + x\left(n+\tfrac{N}{2}\right)\right) W_{\frac{N}{2}}^{rn} \\
&= \mathrm{DFT}_{\frac{N}{2}} \left[x(n) + x\left(n+\tfrac{N}{2}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
X(2r+1) &= \sum_{n=0}^{N-1} x(n) W_N^{(2r+1)n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + W_N^{\frac{N}{2}} x\left(n+\tfrac{N}{2}\right)\right) W_N^{(2r+1)n} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left(\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right) W_{\frac{N}{2}}^{rn} \\
&= \mathrm{DFT}_{\frac{N}{2}} \left[\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right]
\end{aligned}
$$

FIR and IIR Filters

- In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time.
- An infinite impulse response (IIR) filter is a digital filter that depends linearly on a finite number of input samples and a finite number of previous filter outputs.

- The crucial difference between FIR and IIR filter is that the FIR filter provides an impulse response of finite period. As against IIR is a type of filter that generates impulse response of infinite duration for a dynamic system.

DWT and CWT

- The Discrete Wavelet Transform (DWT), simply put, is an operation that receives a signal as an input (a vector of data) and decomposes it in its frequential components.
- the Continuous Wavelet Transform (CWT) is a formal (i.e., non-numerical) tool that provides an overcomplete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously.

Welch's method and windowing

Welch's method [296] (also called the periodogram method) for estimating power spectra is carried out by dividing the time signal into successive blocks, forming the periodogram for each block, and averaging.

Denote the m^{th} windowed, zero-padded frame from the signal x by,

$$x_m(n) \stackrel{\Delta}{=} w(n)x(n+mR), \quad n = 0, 1, \ldots, M-1, \; m = 0, 1, \ldots, K-1,$$

where R is defined as the window hop size, and let K denote the number of available frames. Then the periodogram of the m^{th} block is given by,

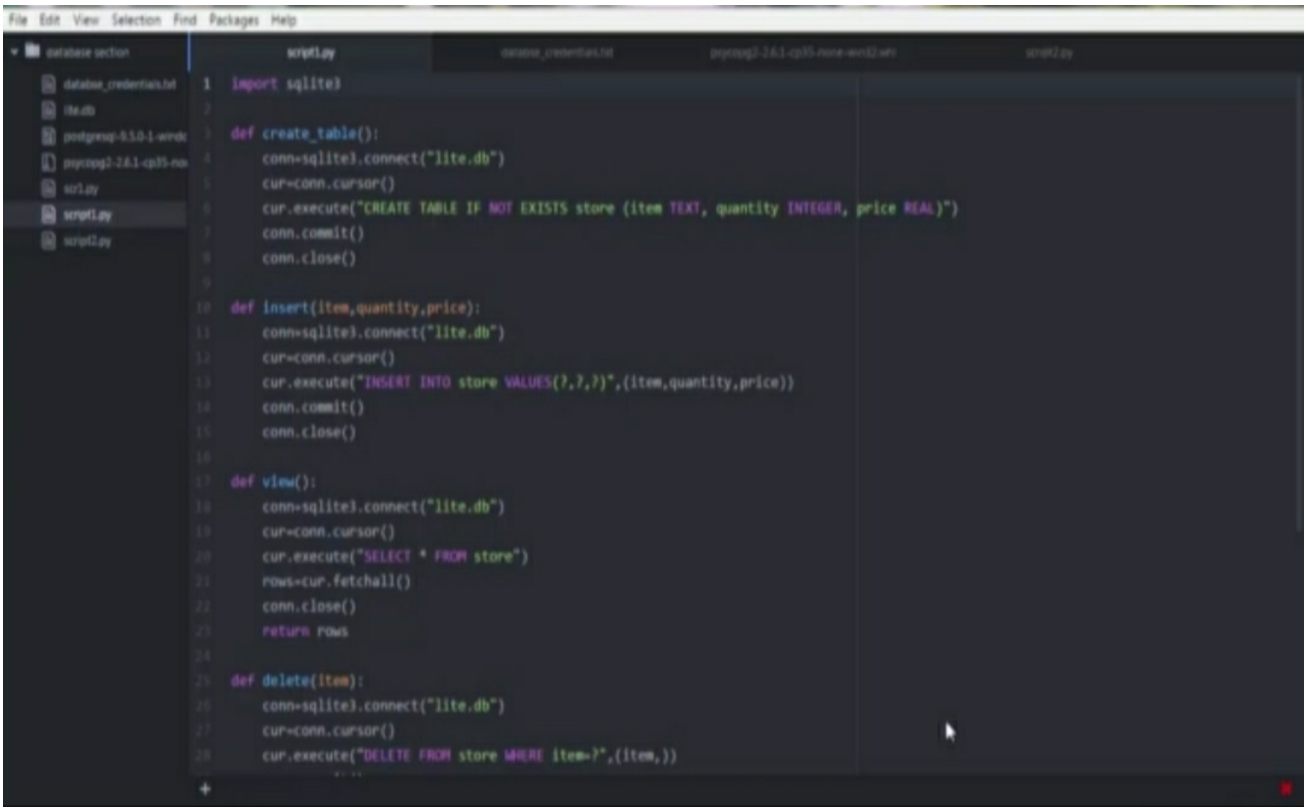
$$P_{x_m,M}(\omega_k) = \frac{1}{M} \left| \mathrm{FFT}_{N,k}(x_m) \right|^2 \stackrel{\Delta}{=} \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j2\pi nk/N} \right|^2$$

as before, and the Welch estimate of the power spectral density is given by

$$\hat{S}_x^W(\omega_k) \stackrel{\Delta}{=} \frac{1}{K} \sum_{m=0}^{K-1} P_{x_m,M}(\omega_k).$$

Date:	27/05/2020	Name:	Abhishek
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC001
Topic:	1] Interacting with Databases	Semester & Section:	6th 'A'
Github Repository:	Abhishek-online-courses		

AFTERNOON SESSION DETAILS
Image of session


File Edit View Selection Find Packages Help
database section
script1.py
database_credentials.txt
lite.db
scr1.py
script1.py
script2.py
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()
def view():
conn=sqlite3.connect("lite.db")
cur=conn.cursor()
cur.execute("SELECT * FROM store")
rows=cur.fetchall()
conn.close()
return rows
def delete(item):
conn=sqlite3.connect("lite.db")
cur=conn.cursor()
cur.execute("DELETE FROM store WHERE item=?",(item,))

Report

Interacting with Databases

- Introduction to python with database.
- A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
- Connecting and inserting Data to SQlite.
- SQLite is a relational database management system (RDBMS) contained in a C library.
- Introduction to PostgreSQL Psycopg2.
- The PostgreSQL can be integrated with Python using psycopg2 module.
- Psycopg2 is a PostgreSQL database adapter for the Python programming language. .
- To create a new table in SQLite and PostgreSQL, CREATE TABLE statement is used.
- Selecting, Inserting, Updating and Deleting SQLite records and PostgreSQL records can be done using SELECT, INSERT, UPDATE AND DELETE SQL commands respectively.