**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **27/5/2020** | **Name:** | **GAURAV NR** |
| **Course:** | **Digital signal processing** | **USN:** | **4AL15EC025** |
| **Topic:** | **FFT, FIR and IIR filters, CWT & DWT, Welch’s method and windowing, ECG signal analysis using MATLAB** | **Semester & Section:** | **8th sem A sec** |
| **Github Repository:** | **gaurav** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report –**  **FAST FOURIER TRANSFORM MATLAB:**  A fast Fourier transform (FFT) is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) that computes the [discrete Fourier transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) (DFT) of a sequence, or its inverse (IDFT). [Fourier analysis](https://en.wikipedia.org/wiki/Fourier_analysis) converts a signal from its original domain (often time or space) to a representation in the [frequency domain](https://en.wikipedia.org/wiki/Frequency_domain) and vice versa. The DFT is obtained by decomposing a [sequence](https://en.wikipedia.org/wiki/Sequence) of values into components of different frequencies. This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical. An FFT rapidly computes such transformations by [factorizing](https://en.wikipedia.org/wiki/Matrix_decomposition) the [DFT matrix](https://en.wikipedia.org/wiki/DFT_matrix) into a product of [sparse](https://en.wikipedia.org/wiki/Sparse_matrix) (mostly zero) factors. As a result, it manages to reduce the [complexity](https://en.wikipedia.org/wiki/Computational_complexity_theory) computing the DFT from {\displaystyle O\left(N^{2}\right)}, which arises if one simply applies the definition of DFT, to {\displaystyle O(N\log N)}, where {\displaystyle N} is the data size. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In the presence of [round-off error](https://en.wikipedia.org/wiki/Round-off_error), many FFT algorithms are much more accurate than evaluating the DFT definition directly or indirectly. There are many different FFT algorithms based on a wide range of published theories, from simple [complex-number arithmetic](https://en.wikipedia.org/wiki/Complex_number) to [group theory](https://en.wikipedia.org/wiki/Group_theory) and [number theory](https://en.wikipedia.org/wiki/Number_theory).  **SYNTAX:**  [Y = fft(X)](https://www.mathworks.com/help/matlab/ref/fft.html#d120e347344)  [Y = fft(X,n)](https://www.mathworks.com/help/matlab/ref/fft.html#d120e347387)  [Y = fft(X,n,dim)](https://www.mathworks.com/help/matlab/ref/fft.html#d120e347446) **DESCRIPTION:** [Y](https://www.mathworks.com/help/matlab/ref/fft.html#f83-998360-Y) = fft([X](https://www.mathworks.com/help/matlab/ref/fft.html#f83-998360-X)) computes the [discrete Fourier transform](https://www.mathworks.com/help/matlab/ref/fft.html#buuutyt-6) (DFT) of X using a fast Fourier transform (FFT) algorithm.   * If X is a vector, then fft(X) returns the Fourier transform of the vector. * If X is a matrix, then fft(X) treats the columns of X as vectors and returns the Fourier transform of each column. * If X is a multidimensional array, then fft(X) treats the values along the first array dimension whose size does not equal 1 as vectors and returns the Fourier transform of each vector.   [example](https://www.mathworks.com/help/matlab/ref/fft.html#buvwuf3-1)  [Y](https://www.mathworks.com/help/matlab/ref/fft.html#f83-998360-Y) = fft([X](https://www.mathworks.com/help/matlab/ref/fft.html" \l "f83-998360-X),[n](https://www.mathworks.com/help/matlab/ref/fft.html" \l "f83-998360-n)) returns the n-point DFT. If no value is specified, Y is the same size as X.   * If X is a vector and the length of X is less than n, then X is padded with trailing zeros to length n. * If X is a vector and the length of X is greater than n, then X is truncated to length n. * If X is a matrix, then each column is treated as in the vector case. * If X is a multidimensional array, then the first array dimension whose size does not equal 1 is treated as in the vector case.   [example](https://www.mathworks.com/help/matlab/ref/fft.html#buuutyt-10)  [Y](https://www.mathworks.com/help/matlab/ref/fft.html#f83-998360-Y) = fft([X](https://www.mathworks.com/help/matlab/ref/fft.html" \l "f83-998360-X),[n](https://www.mathworks.com/help/matlab/ref/fft.html" \l "f83-998360-n),[dim](https://www.mathworks.com/help/matlab/ref/fft.html#f83-998360-dim)) returns the Fourier transform along the dimension dim. For example, if X is a matrix, then fft(X,n,2) returns the n-point Fourier transform of each row.  **FIR AND IIR FILTERS:**  Both FIR and IIR are the two major classifications of digital filters used for signal filtration. 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. **Definition of FIR filter:** A type of digital filter that generates a finite impulse response of a dynamic system is known as FIR filters. More simply, we can say, here the impulse response provided by the filter is of finite duration.  In FIR filters the response gets fixed to zero in a finite period of time thus it is named so. In the case of FIR filters, the nth order filter generates n+1 samples before getting fixed to 0. In FIR filters, feedback is not present thus these operate on only present and past input values. Hence the output is the summation of a finite quantity of finite samples of input values. So, these are highly stable. The generalized equation for FIR filter is given as:  [eq1 FIR filter](https://circuitglobe.com/wp-content/uploads/2020/03/eq1-FIR-filter.jpg)  Where, N is the length of the filter.  FIR filters are used with systems having 0 response. **Definition of IIR filter:** The type of digital filter that is designed to generate infinite impulse response of a dynamic system is known as the IIR filter. As an internal feedback mechanism is present in these filters thus the filter operates for an indefinite period of time.  IIR filters are used by the systems that generate an infinite response. IIR filter operates in a way that not only the present and past inputs but the past output sample is also taken into consideration.  The equation in general form for IIR filter is given as:  [eq2 IIR filter](https://circuitglobe.com/wp-content/uploads/2020/03/eq2-IIR-filter.jpg)  As these filters support the recursive operation. Thus it never allows its response to settle to 0 for the applied impulse.  The memory requirement and computational time enhance the efficiency of the IIR filters. Also, it is quite easy to convert an analog filter into a digital IIR filter. [**WELCH'S METHOD**](https://ccrma.stanford.edu/~jos/sasp/Welch_s_Method.html)**AND WINDOWING:**As usual, the purpose of the window function  w(n) is to reduce [side-lobe](https://ccrma.stanford.edu/~jos/sasp/Rectangular_Window_Side_Lobes.html) level in the spectral density estimate, at the expense of frequency resolution, exactly as in the case of [sinusoidal](https://ccrma.stanford.edu/~jos/mdft/Sinusoids.html) [spectrum analysis](https://ccrma.stanford.edu/~jos/mdft/Example_Applications_DFT.html). When using a non-rectangular analysis window, the window hop-size R cannot exceed half the frame length M/2 without introducing a non-uniform sensitivity to the data x(n) over time. In the rectangular window case, we can set R=M and have non-overlapping windows. For Hamming, [Hanning](https://ccrma.stanford.edu/~jos/sasp/Hann_Hanning_Raised_Cosine.html), and any other generalized [Hamming window](https://ccrma.stanford.edu/~jos/sasp/Hamming_Window.html), one normally sees R=(M-1)/2 for odd-length windows. For the [Blackman window](https://ccrma.stanford.edu/~jos/sasp/Three_Term_Blackman_Harris_Window.html#10289), R~M/2 is typical. In general, the hop size R is chosen so that the analysis window $ w$ overlaps and adds to a constant at that hop size. This consideration is explored more fully in Chapter [8](https://ccrma.stanford.edu/~jos/sasp/Overlap_Add_OLA_STFT_Processing.html#chap:ola). An equivalent parameter used by most [matlab](https://ccrma.stanford.edu/~jos/matlab/) implementations is the overlap parameter M-R .  Note that the number of blocks averaged in increases as $ R$ decreases. If Nx>=M denotes the total number of [signal](http://ccrma.stanford.edu/~jos/filters/Definition_Signal.html) samples available, then the number of complete blocks available for averaging may be computed as   |  |  | | --- | --- | | $\displaystyle K = \left\lfloor \frac{N_x-M}{R}\right\rfloor + 1,$ |  |   where the floor function [x] denotes the largest integer less than or equal to x . |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:27/5/2020** |  | **Name: GAURAV** |  | |
| **Course: Python** |  | **USN:4AL15EC025** |  | |
| **Topic: Graphical user interface with Tkinter, interacting with databases** |  | **Sem :8th sem**  **Section: A sec** |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **SESSION IMAGE :**        **REPORT:**   1. **MULTI WIDGET GUI:**   Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. **To create a tkinter app:**   1. Importing the module – tkinter 2. Create the main window (container) 3. Add any number of widgets to the main window 4. Apply the event Trigger on the widgets.   Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is ‘Tkinter’ and in Python 3.x it is ‘tkinter’.  import tkinter  There are two main methods used which the user needs to remember while creating the Python application with GUI.   1. **Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1):** To create a main window, tkinter offers a method ‘Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:   m=tkinter.Tk() where m is the name of the main window object   1. There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.   m.mainloop()   1. from tkinter import \* 2. # Create an empty Tkinter window 3. window=Tk() 4. def from\_kg(): 5. # Get user value from input box and multiply by 1000 to get kilograms 6. gram=float(e2\_value.get())\*1000 7. # Get user value from input box and multiply by 2.20462 to get pounds 8. pound=float(e2\_value.get())\*2.20462 9. # Get user value from input box and multiply by 35.274 to get ounces ounce=float(e2\_value.get())\*35.274   Empty the Text boxes if they had text from the previous use and fill them again  t1.delete("1.0", END) # Deletes the content of the Text box from start to END  t1.insert(END,gram) # Fill in the text box with the value of gram variable  t2.delete("1.0", END)  t2.insert(END,pound)  t3.delete("1.0", END)  t3.insert(END,ounce)  # Create a Label widget with "Kg" as label  e1=Label(window,text="Kg")  e1.grid(row=0,column=0) # The Label is placed in position 0, 0 in the window  e2\_value=StringVar() # Create a special StringVar object  e2=Entry(window,textvariable=e2\_value) # Create an Entry box for users to enter the value  e2.grid(row=0,column=1  # Create a button widget  # The from\_kg() function is called when the button is pushed  b1=Button(window,text="Convert",command=from\_kg)  b1.grid(row=0,column=2  # Create three empty text boxes, t1, t2, and t3  t1=Text(window,height=1,width=20)  t1.grid(row=1,column=0)  t2=Text(window,height=1,width=20)  t2.grid(row=1,column=1)  t3=Text(window,height=1,width=20)  t3.grid(row=1,column=2)  # This makes sure to keep the main window open  window.mainloop()  **Querying data from a MySQL database:**  If you prefer to work with MySQL instead of PostGreSQL, the below code is used.  I set up a remote MySQL database on a server with the IP address 108.167.140.122, so you don't have to install and set up a MySQL database yourself. To connect and query data from that remote database, you need a username, password, and the name of the database. These are written inside the Python script below.  You also need a Python library that interacts with MySQL databases. Many libraries are compatible, but I prefer mysql.connector. To install mysql.connector: simply execute pip install mysql-connector  or pip3 install mysql-connector depending on whether you use pip or pip3.  import mysql.connector   1. word = input("Enter a word in English and press Enter: ") 2. con = mysql.connector.connect( 3. user="ardit700\_student", 4. password = "ardit700\_student", 5. host="108.167.140.122", 6. database = "ardit700\_pm1database" 7. ) 8. cursor = con.cursor() 9. query = cursor.execute("SELECT \* FROM Dictionary WHERE Expression = '%s'" % word) 10. results = cursor.fetchall() 11. if results: 12. for result in results: 13. print(result[1]) 14. else: 15. print("We couldn't find any results about that.") | | | |