**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **27/05/2020** | **Name:** | **DHAMINI C L** |
| **Course:** | **DIGITAL SIGNAL PROCESSING** | **USN:** | **4AL17EC025** |
| **Topic:** | **Fourier Series and Gibbs Phenomenon using**  **Python, Laplace transform using Matlab,Z**  **Transform Using Matlab.** | **Semester & Section:** | **6TH & A** |
| **Github Repository:** | **DHAMINI-CL-Course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **Report:**  **f(x)=1a0 +∑(akcos2kt+bksin2kt)**  **−∞**  **∞**  **X(F) = ∫ x(t)e−j2Ftdt −∞**  **2**  **Fourier Series and Gibbs Phenomana Using Python import numpy as np**  **import matplotlib.pyplot as plt plt.rcParams[&#39;figure.figsize&#39;]=[8,8]**  **plt.rcParams.update({&#39;font.size&#39;:18})**  **dx=0.01**  **L=2\*np.pi**  **x=np.arange(0,L+dx,dx)**  **n=len(x)**  **nquart=int(np.floor(n/4)) f=np.zeros\_like(x) f[nquart:3\*nquart]=1**  **A0=np.sum(f\*np.ones\_like(x))\*dx\*2/L fFs=A0/2\*np.ones\_like(f)**  **for k in range(1,101):**  **Ak=np.sum(f\*np.cos(2\*np.pi\*k\*x/L))\*dx\*2/L Bk=np.sum(f\*np.sin(2\*np.pi\*k\*x/L))\*dx\*2/L**  **fFs=fFs+Ak\*np.cos(2\*k\*np.pi\*x/L)+Bk\*np.sin(2\*k\*np.pi\*x/L)**  **plt.plot(x,f,color=&#39;k&#39;,LineWidth=2) plt.plot(x,fFs,&#39;-&#39;,color=&#39;r&#39;,Linewidth=1.5) plt.show()**  **Laplace Transform [Matlab] clear all;**  **close all;**  **syms L f t; f=(exp(-3\*t)\*sin(2\*t))/t**  **L=laplace(f​)**  **Inverse Laplace Transform clear all;**  **close all;**  **syms F,s,x; F=(s+29)/(s^3+4\*s^2+9\*s+36) ilaplace(F,x)**  **Z Transform Using Matlab clear all;**  **close all;**  **syms n,w;**  **a=sin(w\*n)**  **b=ztrans(a)**  **disp(b)**  **(z\*sin(w))/(z^2 -2\*cos(w)\*z+1) pretty(b)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:26/5/2020** |  | **Name: DHAMINI C L** |  | |
| **Course:PYTHON** |  | **USN:4AL17EC025** |  | |
| **Topic: pandas** |  | **Semester & Section:6TH A SEC** |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  .  **Pandas** is a Python library that provides extensive means for data analysis. Data scientists often work with data stored in table formats like .csv, .tsv, or .xlsx. Pandas makes it very convenient to load, process, and analyze such tabular data using SQL-like queries. In conjunction with Matplotlib and Seaborn, Pandas provides a wide range of opportunities for visual analysis of tabular data.  The main data structures in Pandas are implemented with **Series** and **DataFrame** classes. The former is a one-dimensional indexed array of some fixed data type. The latter is a two-dimensional data structure - a table - where each column contains data of the same type. You can see it as a dictionary of Series instances. DataFrames are great for representing real data: rows correspond to instances (examples, observations, etc.), and columns correspond to features of these instances | | | |