

DAILY ASSESSMENT REPORT

Date:	26 May 2020	Name:	PAVITHRAN S
Course:	DIGITAL SIGNAL PROCESSING	USN:	4AL17EC068
Topic:	<ul style="list-style-type: none"> • Fourier Series & Gibbs Phenomena using Python • Fourier Transform • Fourier Transform Derivatives • Fourier Transform and Convolution • Intuition of Fourier Transform and Laplace Transform • Laplace Transform of First order • Implementation of Laplace Transform using Matlab • Applications of Z-Transform • Find the Z-Transform of sequence using Matlab 	Semester & Section:	6 th B
Github Repository:	Pavithran		

FORENOON SESSION DETAILS

Image of session

The screenshot shows a YouTube video player. The main video is titled "Laplace Transform: First Order Equation" and is from the MIT OpenCourseWare channel. The video content features a lecturer standing in front of a chalkboard. The chalkboard has the following text written on it:

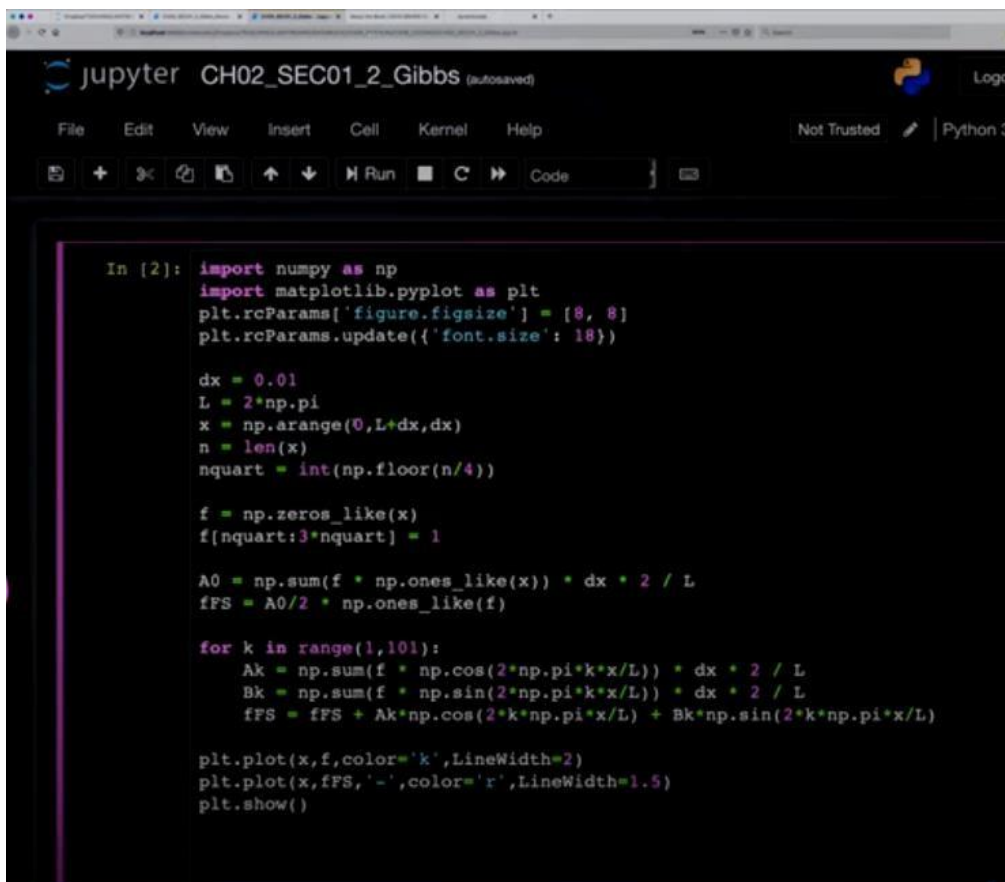
- Laplace Transform: 1st order equation
- The transforms of $f(t)$ and $y(t)$ are $F(s)$ and $Y(s)$
- Definition: $F(s) = \int_0^\infty e^{-st} f(t) dt$
- Example $f(t) = at$

The video player interface shows a progress bar at 0:04 / 22:37. On the right side, there is a list of recommended videos:

- The Laplace Transform - A Graphical Approach** by Brian Douglas (508K views • 7 years ago) - 12:24
- Laplace Transform: Second Order Equation** by MIT OpenCourseWare (33K views • 4 years ago) - 15:31
- What's a Tensor?** by Dan Fleisch (1.9M views • 8 years ago) - 12:21
- But what is the Fourier Transform? A Visual...** by 3Blue1Brown (4.2M views • 2 years ago) - 19:43
- MIT Learn Differential Equations** by MIT OpenCourseWare (68 views)

Report – Report can be typed or hand written for up to two pages.

Fourier Series & Gibbs Phenomena using Python:



```
In [2]: import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [8, 8]
plt.rcParams.update({'font.size': 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='k',LineWidth=2)
plt.plot(x,fFS,'-',color='r',LineWidth=1.5)
plt.show()
```

Fourier Transform & Fourier Transform Derivatives:

- Digital Signal Processing/Discrete Fourier Transform. As the name implies, the Discrete Fourier Transform (DFT) is purely discrete: discrete-time data sets are converted into a discrete-frequency representation. This is in contrast to the DTFT that uses discrete time, but converts to continuous frequency.

The Fourier Series \rightarrow Fourier Transform

$$f(x) = \sum_{k=-\infty}^{\infty} C_k e^{i k \pi x / L} \quad \omega_k = \frac{k \pi}{L} = k \Delta \omega \quad \Delta \omega = \frac{\pi}{L}$$
$$C_k = \frac{1}{2\pi} \langle f(x), \psi_k \rangle = \frac{1}{2\pi} \int_{-L}^L f(x) \underbrace{e^{-i k \pi x / L}}_{\psi_k} dx$$
$$f(x) = \lim_{\substack{\Delta \omega \rightarrow 0 \\ (L \rightarrow \infty)}} \sum_{k=-\infty}^{\infty} \frac{\Delta \omega}{2\pi} \int_{-\pi/\Delta \omega}^{\pi/\Delta \omega} f(\xi) e^{-i k \omega \xi} d\xi e^{i k \omega x}$$
$$= \int_{-\infty}^{\infty} \underbrace{\frac{1}{2\pi} \int_{-\infty}^{\infty} f(\xi) e^{-i \omega \xi} d\xi}_{f(\omega)} e^{i \omega x} d\omega$$

Fourier Transform and Convolution:

- Why study Fourier transforms and convolution? Each of these sinusoidal terms has a magnitude (scale factor) and a phase (shift). – Note that in a computer, we can represent a function as an array of numbers giving the values of that function at equally spaced points.

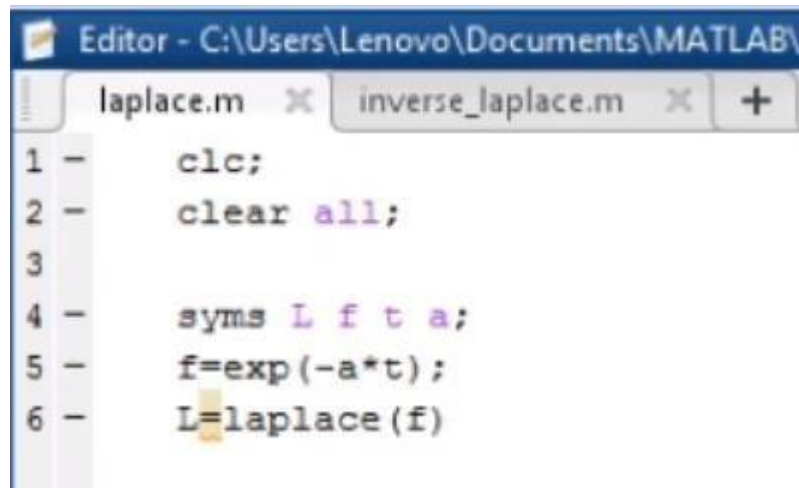
Laplace Transform of First order:

- One familiar input to a first order system is the step change or step input. A step change from 0 to 1 is equivalent to a function that is equal to 0 for time < 0, and is equal to 1 for time ≥ 0 . The Laplace transform of such a function is $1/s$.

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\theta_o(s)}{a/s}$$
$$\Rightarrow \theta_o(s) = \frac{a}{s} \frac{K}{(1+\tau s)} = Ka \frac{1}{s(1+\tau s)}$$
$$\Rightarrow \theta_o(s) = Ka \frac{1/\tau}{s(1/\tau + s)} = Ka \frac{1/\tau}{s(s + 1/\tau)}$$

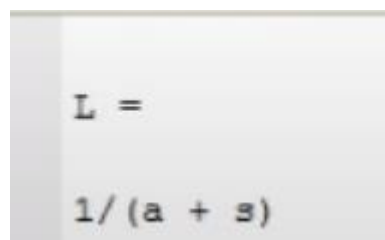
Implementation of Laplace Transform using Matlab:

Code:



```
Editor - C:\Users\Lenovo\Documents\MATLAB\I
laplace.m  X  inverse_laplace.m  X  +
1 -      clc;
2 -      clear all;
3
4 -      syms L f t a;
5 -      f=exp(-a*t);
6 -      L=laplace(f)
```

Output:

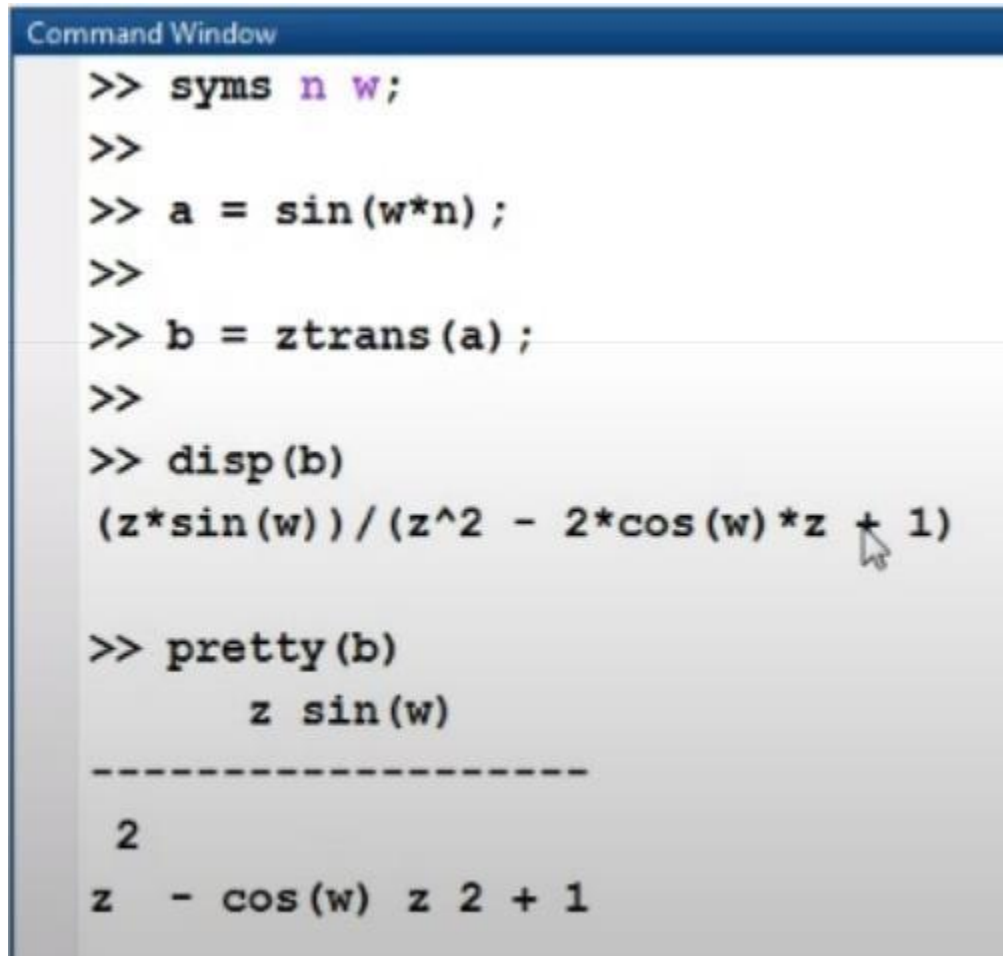


```
L =
1/(a + s)
```

Applications of Z-Transform

- Sampled systems
- Inputs and outputs are related by difference equations and Z-transform techniques are used to solve those difference equations.
- VOICE TRANSMISSION: To band-limit the signal and filter noise from the signal.
- Calculation of a signal to control a system.

Find the Z-Transform of sequence using Matlab:



```
Command Window
>> syms n w;
>>
>> a = sin(w*n);
>>
>> b = ztrans(a);
>>
>> disp(b)
(z*sin(w))/(z^2 - 2*cos(w)*z + 1)

>> pretty(b)
      z sin(w)
-----
      2
z  - cos(w) z 2 + 1
```

- The above picture is a basic example of Z-Transform using Matlab.

Date:	26 May 2020	Name:	PAVITHRAN S
Course:	The Python Mega Course	USN:	4AL17EC068
Topic:	Application 4: Build a Personal Website with Python and Flask	Semester & Section:	6th B

AFTERNOON SESSION DETAILS

Image of session:

The screenshot displays the Udeity course interface. At the top, the course title 'The Python Mega Course: Build 10 Real World Applications' is visible. A congratulatory message from the instructor reads: 'Congratulations! Hey, I just wanted to congratulate you and tell you that it's awesome that you made it this far in the course. You have completed around 50% of the course and I know it takes patience and commitment to go this far without quitting. Since you made it this far you probably know a lot about Python by now. I promise that if you go on completing the rest of the course you will be on the right path to becoming a real Python programmer. There's a ton of fun and useful content and apps awaiting for you in the next sections. Kudos and I'll keep in touch with you!'. The course content sidebar on the right lists sections 21 through 27, including 'Section 21: Graphical User Interfaces with Tkinter', 'Section 22: Interacting with Databases', 'Section 23: Application 5: Build a Desktop Database Application', 'Section 24: Object Oriented Programming', 'Section 25: Python for Image and Video Processing with OpenCV', 'Section 26: Application 6: Build a Webcam Motion Detector', and 'Section 27: Interactive Data Visualization with Bokeh'. The navigation bar at the bottom includes links for Overview, Q&A, Bookmarks, and Announcements.

Report – Report can be typed or hand written for up to two pages.

Build a Personal Website with Python and Flask:

- We have learnt to build our Personal website using Python.
- It shows how to render links from one page to another.
- It also includes HTML format for making good visuals.
- Also shows how to create virtual environment to run the app.
- This course helped in putting our website into global domain as well.
- There is scope for further improvement in the current website.
- The below shown are some images of personal website built by me.