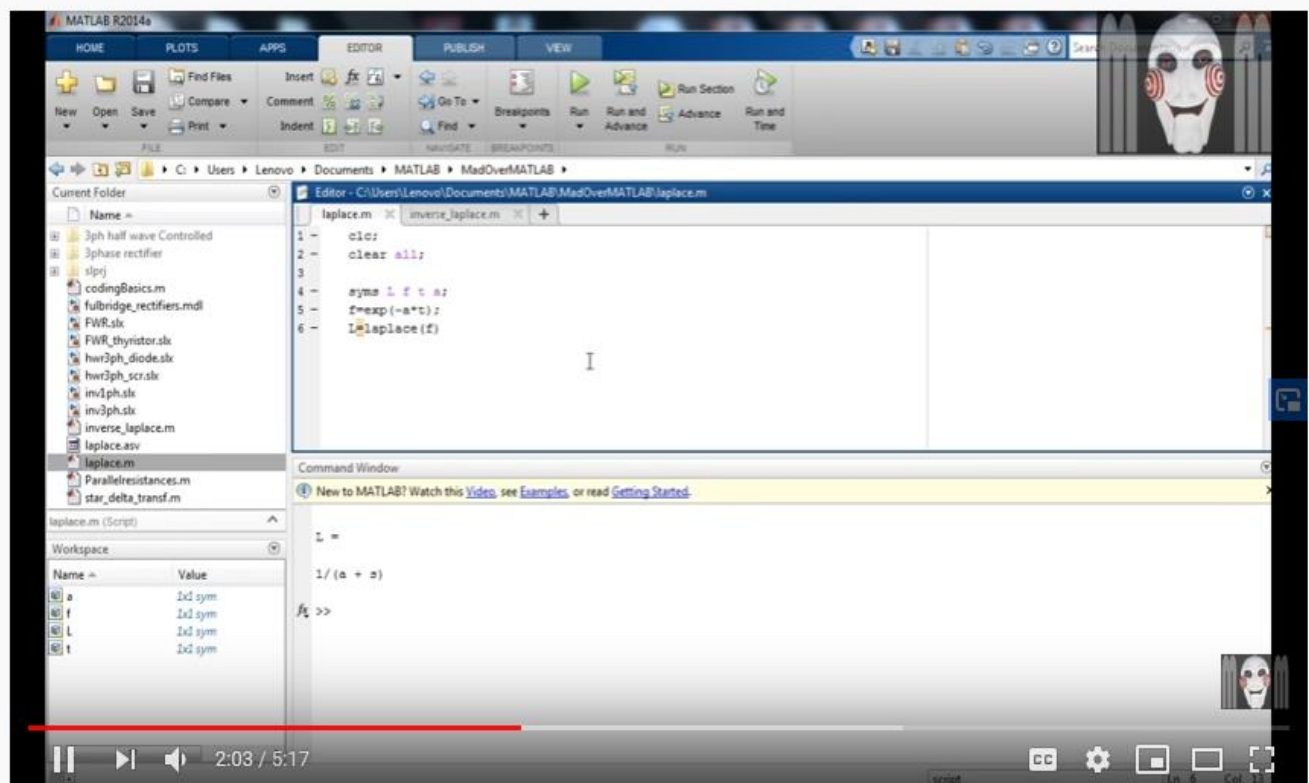


DAILY ASSESSMENT FORMAT

Date:	26 may 2020	Name:	nishanth
Course:	Digital signal processing	USN:	4a17ec063
Topic:	1.Fourier Series & Gibbs Phenomena using Python 2.Fourier Transform 3.Fourier Transform derivatives 4.Fourier Transform and Convolution 5.Intuition of Fourier Transform and Laplace Transform 6.Laplace Transform of First order 7.Implementation of Laplace Transform using Matlab 8.Applications of Z-Transform 9.Find the Z-Transform of sequence using Matlab	Semester & Section:	6th & B
Github Repository:	nishanthvr		

FORENOON SESSION DETAILS

Image of session



Laplace Transform and Inverse laplace Transform Using MATLAB | Mad Over Matlab Tutorials

Nirshanth

day-2

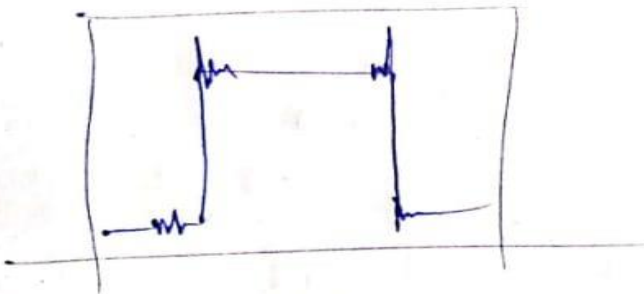
Fourier Series and Gibbs phenomenon (python)

$$f(x) \approx \sum_{k=0}^{100} a_k \cos\left(k \frac{2\pi x}{L}\right) + b_k \sin\left(k \frac{2\pi x}{L}\right)$$

$$a_k = \langle f(x), \cos\left(k \frac{2\pi x}{L}\right) \rangle$$

$$b_k = \langle f(x), \sin\left(k \frac{2\pi x}{L}\right) \rangle$$

Python code



code (python code)

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [8, 5]
plt.rcParams.update({'font.size': 18})

dx = 0.01
L = 2 * np.pi
x = np.arange(0, L + dx, dx)
n = len(x)
nquant = int(np.floor(n/4))
f = np.zeros_like(x)
f[nquant:3*nquant] = 1
A0 = np.sum(f * np.ones_like(x)) * dx * 2/L
fFS = A0/2 * np.ones_like(f)
```

for k in range $(1, 101)$:

$$A_k = np.sum(f * np.cos(2 * np.pi * k * x / h)) * dx * 2 / h$$

$$B_k = np.sum(f * np.sin(2 * np.pi * k * x / h)) * dx * 2 / h$$

$$bfs = bfs + A_k * np.cos(2 * k * np.pi * x / h) + B_k * np.sin(2 * k * np.pi * x / h)$$

plt.plot(x, f, color='k', linewidth=2)

plt.plot(x, bfs, '-', color='v', linewidth=1.5)

plt.show()



Fourier Series

Fourier transform

$$b(x) = \sum_{k=-\infty}^{\infty} C_k e^{i k \pi x / h}$$

$$C_k = \frac{1}{2h} \langle b(x), \psi_k \rangle = \frac{1}{2h} \int_{-h}^h b(x) e^{-i k \pi x / h} dx$$

$$b(x) = \lim_{\Delta \omega \rightarrow 0} \sum_{k=-\infty}^{\infty} \frac{\Delta \omega}{2\pi} \int_{-\pi \Delta \omega}^{\pi \Delta \omega} b(\xi) e^{-i k \omega \xi} d\xi e^{i k \omega x}$$

$$= \int_{-\infty}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} b(\xi) e^{-i \omega \xi} d\xi e^{i \omega x} d\omega$$

Property

$$\hat{f}(\omega) = \hat{F}(1/\omega) = \int_{-\infty}^{\infty} f(x) e^{-i \omega x} dx$$

$$g(x) = \hat{F}^T(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i \omega x} d\omega$$

the fourier transform and convolution Integral

$$(f \times g)(x) = \int_{-\infty}^{\infty} f(x-y) g(y) dy$$

$$\begin{aligned} \mathcal{F}^{-1}(\mathcal{F}f \mathcal{F}g)(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}f(\omega) \mathcal{F}g(\omega) e^{j\omega x} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}f(\omega) \left(\int_{-\infty}^{\infty} g(y) e^{-j\omega y} dy \right) e^{j\omega x} d\omega \\ &= \frac{1}{2\pi} \left(\int_{-\infty}^{\infty} g(y) \int_{-\infty}^{\infty} \mathcal{F}f(\omega) e^{j\omega(x-y)} d\omega dy \right) \\ &= \int_{-\infty}^{\infty} g(y) f(x-y) dy \\ &= f \times g \end{aligned}$$

the fourier

intuition behind fourier and laplace transform

$$\mathcal{F}f(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

$$\mathcal{F}f(\omega) = \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt - i \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt$$

laplace transform: first order Equation

$$\mathcal{F}f(s) = \int_0^{\infty} f(t) e^{-st} dt$$

laplace transform: 1st order Equation the transform of $f(t)$ and $y(t)$ are $\mathcal{F}f(s)$ and $\mathcal{F}y(s)$

$$\text{at } f(t) = e^{at} \quad \mathcal{F}f(s) = \int_0^{\infty} e^{at} e^{-st} dt = \left[\frac{e^{(a-s)t}}{a-s} \right]_0^{\infty} = \frac{1}{s-a} = \mathcal{F}(e^{at})$$

$$\frac{dy}{dt} - ay = 0 \quad \xrightarrow{\text{LT}} \int_0^{\infty} \frac{dy}{dt} e^{-st} dt = - \int_0^{\infty} y(t) (-se^{-st} dt) + [ye^{-st}]_0^{\infty}$$

$$s(Y(s)) - y(0) - a(Y(s)) = 0$$

$$Y(s) = \frac{y(0)}{s-a} \xrightarrow{\text{inv}} y(t) = y(0)$$

$$\boxed{y(t) = y(0)e^{at}}$$

$$\boxed{Y(s) = \frac{1}{(s-a)} + \frac{y(0)}{(s-a)}}$$

Laplace transform and Inverse transform using matlab

clc;

clear all;

Syms h t;

b = 5 * t => b = exp(-5*t)

h = laplace(b)

F

clc;

clear all;

Syms h t a;

t = exp(-a*t) or b = (exp(-3*t) * sin(2*t)) / t

h = laplace(b)

Inverse Laplace

clc;

clear all;

Syms F s;

F = (s+29) / (s^3 + 4*s^2 + 9*s + 36)

ifft(F, x)

Application of z-transform

Difference Equation

model process

It is relation between the difference of an unknown function at one or more general value

$$\Delta y_{n+1} + y_n = z$$

$$\Delta y_{n+1} + \Delta^2 y_{n-1} = 1$$

are difference equation

Example

$$u_{n+2} + 4u_{n+1} + 3u_n = 3^n \quad u_0 = 0, u_1 = 1$$

$$z(u_{n+2}) + 4z(u_{n+1}) + 3z(u_n) = z(3^n)$$

$$z^2 [u(z) - u_0 - \frac{u_1}{z}] + 4z [u(z) - u_0] + 3u(z) = \frac{z}{z-3}, |z| > 3$$

$$z^2 [u(z) - 0 - \frac{1}{z}] + 4z [u(z) - 0] + 3u(z) = \frac{z}{z-3}$$

$$(z^2 + 4z + 3) = \frac{z}{z-3} + z = \frac{z^2 - 2z}{z-3}$$

$$\text{Hence } \frac{u(z)}{z} = \frac{z-2}{(z-3)(z+1)(z+3)} = \frac{A}{(z-3)} + \frac{B}{(z+1)} + \frac{C}{(z+3)}$$

Hence

$$\frac{u(z)}{z} = \frac{1}{2} \frac{1}{4(z-3)} + \frac{3}{8(z+1)} + \frac{5}{12(z+3)}$$

or

$$u(z) = \frac{1}{2} \frac{z}{4(z-3)} + \frac{3}{8} \frac{z}{z+1} + \frac{5}{12} \frac{z}{z+3}$$

Here

$$\rightarrow Z^{-1}(Z)$$

$$= \frac{1}{24} Z^{-1}\left(\frac{Z}{Z-3}\right) + \frac{3}{8} Z^{-1}\left(\frac{Z}{Z+1}\right) + \frac{5}{12} Z^{-1}\left(\frac{Z}{Z+3}\right)$$

$$= \frac{1}{24} (3^n) + \frac{3}{8} (-1)^n + \frac{5}{12} (-3)^n \quad |Z| > 3$$

$$\text{then } u_n = \frac{1}{24} 3^n + \frac{3}{8} (-1)^n + \frac{5}{12} (-3)^n \quad |Z| > 3$$

How to calculate the z transform in matlab

Syms n;

$$a = n+1;$$

$$b = ztrans(a);$$

disp(b)

pretty(b)

or

Syms n w,

$$a = \sin(w \cdot n);$$

$$b = ztrans(a);$$

disp(b)

pretty(b)

Date: 26 may 2020
Course: python
Topic: Application 4: Build a Personal Website with Python and Flask

Name: nishanth
USN: 4a17ec063
Semester & Section: 6th & B

Image of session

The screenshot displays a Udemy course interface. At the top, the course title "The Python Mega Course: Build 10 Real World Applications" is visible. The main video player area shows a code editor with the following Python code:

```
script.py - C:\Program Files\Python\Python37\Scripts\Python.exe
File Edit View Selection Find Packages Help
script.py
1 from flask import Flask
2
3 app=Flask(__name__)
4
5 @app.route('/')
6 def home():
7     return "Website content goes here!"
8
9 if __name__ == '__main__':
10     app.run()
```

The right sidebar lists the course content:

- 158. Your First Website (8min)
- 159. HTML Templates (4min) [Resources]
- 160. Navigation Menu (9min) [Resources]
- 161. Note on Browser Caching (1min)
- 162. CSS Styling (6min) [Resources]
- 163. Creating a Python Virtual Environment (6min)
- 164. How to Install Git (1min)
- 165. Deploying the Website to a Live Server (22min) [Resources]

The bottom section is titled "About this course" and contains the text:

A complete Python course for both beginners and intermediates! Master Python 3 by making 10 amazing Python apps.

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

Your first website:

Program:

```
from flask import flask,render_template
App=flask(__name__)
@app.route('/')
def home():
    return render_template("home.html")
@app.route('/about')
def about():
    return render_templates("about.html")
if __name__=="__main__":
    app.run(debug=True)
```