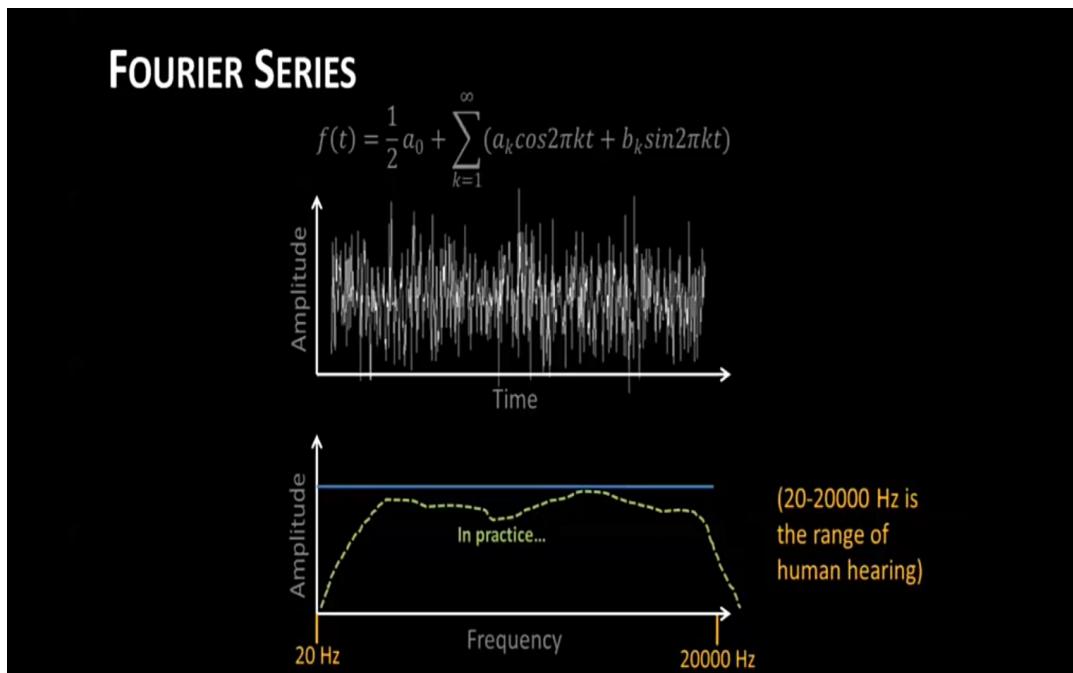


DAILY ASSESSMENT FORMAT

Date:	25/5/2020	Name:	Nichenametla Bhargavi
Course:	Digital Signal Processing	USN:	4AL17EC061
Topic:	Fourier Series	Semester & Section:	6th Sem A Sec
Github Repository:	alvas-education-foundation/Bhargavi_Nichenametla		

FORENOON SESSION DETAILS

Image of session



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

DSP

Fourier Transform: In 1800's Fourier transform was discovered. It is a coordinate transformation.

$u(x, y, t)$ SVD = Data driven

rectangular state
 $u_t = \alpha \nabla^2 u$

This function has eigen values & eigen functions. Since x-axis & y-axis forms as a base for Two-D vector space in the same way these cosines forms as a base for function space.

$f(t) = \frac{1}{2} a_0 + \sum_{k=1}^{\infty} (a_k \cos(2\pi k t) + b_k \sin(2\pi k t))$

Periodic function Fourier Transform:

$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi F t} dt$ → continuous analyzing function sinusoid

Result: one complex coefficient per frequency

$x_a(F) = \int_{-\infty}^{\infty} x(t) \cos(2\pi F t) dt \quad | \quad x_b(F) = \int_{-\infty}^{\infty} x(t) \sin(2\pi F t) dt$

Result: Two real coefficients per frequency

discrete: $x_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi k n}{N}}$

$f(x)$

$f(x) = \sum_{k=1}^{\infty} (A_k \cos(kx) + B_k \sin(kx)) + \frac{A_0}{2}$
 A_k & B_k are Fourier coefficients

$A_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx = \frac{1}{\pi} \langle f(x), \cos(kx) \rangle$

$B_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) dx = \frac{1}{\pi} \langle f(x), \sin(kx) \rangle$

$f(x)$
 $f(x) \in L_2([0, L])$
 $x \in (0, L)$

$f(x) = \frac{A_0}{2} + \sum_{k=1}^{\infty} (A_k \cos(\frac{2\pi k x}{L}) + B_k \sin(\frac{2\pi k x}{L}))$

$A_k = \frac{2}{L} \int_0^L f(x) \cos(\frac{2\pi k x}{L}) dx$

$B_k = \frac{2}{L} \int_0^L f(x) \sin(\frac{2\pi k x}{L}) dx$

inner products in Hilbert space:

$\langle f(x), g(x) \rangle = \int_a^b f(x) \bar{g}(x) dx$

$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$

$\langle f, g \rangle = g^T \cdot f$

$\langle f, g \rangle \Delta x = \sum_{k=1}^n f(x_k) g(x_k) \Delta x$

$\Delta x = \frac{b-a}{n-1}$

Complex Fourier Series:

$$\langle f(x), g(x) \rangle = \int_{-\pi}^{\pi} f(x) \cdot \bar{g}(x) dx$$

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx} = \sum_{k=-\infty}^{\infty} (c_k + i\beta_k)(\cos(kx) + i\sin(kx))$$

$c_k = \bar{c}_{-k}$ if $f(x)$ real

defines orthogonal basis of $f(x)$

$$e^{ikx} = \cos(kx) + i\sin(kx) = \Psi_k$$

$$\langle \Psi_j, \Psi_k \rangle = \int_{-\pi}^{\pi} e^{ijx} \cdot e^{ikx} dx = \int_{-\pi}^{\pi} e^{i(j-k)x} dx = \frac{1}{i(j-k)} [e^{i(j-k)x}]_{-\pi}^{\pi} = \begin{cases} 0 & \text{if } j \neq k \\ 2\pi & \text{if } j = k \end{cases}$$

$$f(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \underbrace{\langle f(x), \Psi_k \rangle}_{c_k} \underbrace{\Psi_k}_{e^{ikx}}$$

Date:	22/05/2020	Name:	Nichenametla Bhargavi
Course:	Python Bootcamp 2020 build 15 working applications and Games	USN:	4AL17EC061
Topic:	Application 1:Build an Interactive English dictionary	Semester & Section:	6th Sem A sec

AFTERNOON SESSION DETAILS

Image of session

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

The image shows handwritten Python code on a piece of paper. The code starts with importing the `difflib` module and using its `SequenceMatcher` class to find matches between words and a database. It includes a function to handle the best match and another for getting close matches. Below this, there's a section on DataFrame indexing and slicing, showing examples like `df.loc` and `df.iloc`.

```
>import difflib
>>from difflib import SequenceMatcher
>>>SequenceMatcher(None, "rain", "gain").ratio()
    ↗ used for getting similarity of strings.
>>> from difflib import get_close_matches
>>> help(get_close_matches)
get_close_matches(wold, possibilities, n=3, cutoff=0.6)
    ↗ word & possibilities are passed
    ↗ n possibilities are returned
Best match:
if w in data:
    return data[w]
elif len(get_close_matches(w, data.keys())) > 0:
    return "Did you mean %s instead?" % get_close_matches(w, data.keys())[0]
else:
    return "The word doesn't exist."
df = pandas.read_csv("file")
Indexing & slicing:
df = df.set_index(addresses)
1) label indexing
2) position indexing
Label Indexing:
df.loc["332 will st", "country"]
    ↗ for 1 item
    ↗ iloc for 2+ items
    ↗ for 1 item
    ↗ iloc for 2+ items
```

Numpy:

1. What is Numpy?
2. Numpy is the fundamental package for scientific computing in Python.
3. It is a Python library that provides a multidimensional array object, various derived objects
4. At the core of the Numpy package, is the ndarray object. This encapsulates-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance.
5. The value '0' is passed when read operation is performed on the image to give gray scale pixel values of the image.
6. The value '1' is passed when read operation is performed on the image to give Blue, Green, Red(BGR)pixel values of the image respectively.