EE4015
Digital Signal Processing

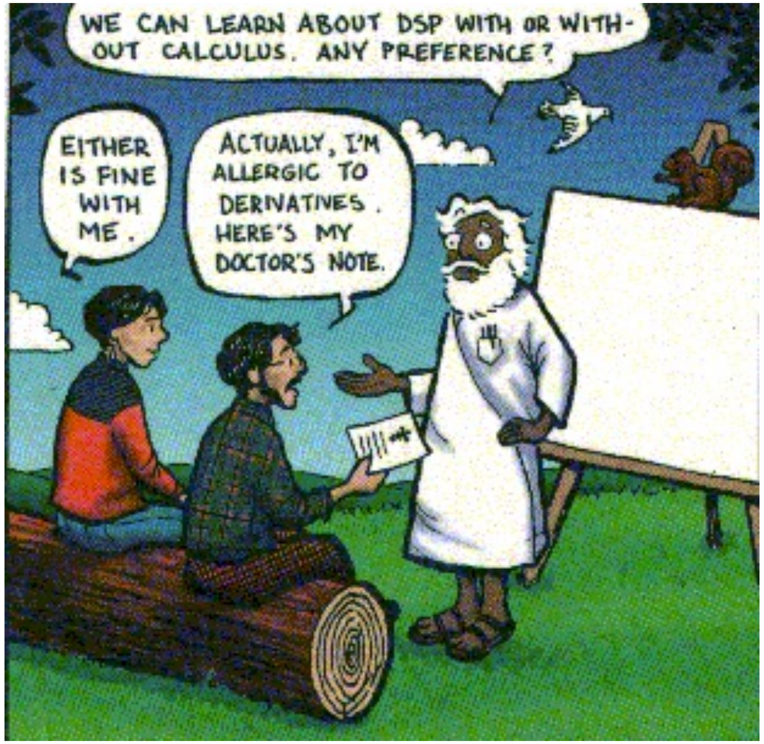
Dr. Lai-Man Po

Department of Electrical Engineering

City University of Hong Kong

http://www.ee.cityu.edu.hk/~lmpo/

EE4015 DSP is Math Class?!



Quote of the EE4015

Do not worry about your difficulties in Mathematics.
I can assure you mine are still greater.
~ Albert Einstein

Outline

- **Today**
  - Course Overview and Administration
    - Teaching Staff, Textbook, Website, Topics, Grading, Assignments, Project
  - Review topics : Math and Continuous-Time Signals and Systems
  - Reading : Chapter 1: Introduction to DSP
    - Digital Signal Processing : Fundamentals and Applications by L. Tan and J. Jiang
- **Next Lecture**
  - Review of Discrete-Time Signals and Systems

Teaching Staff

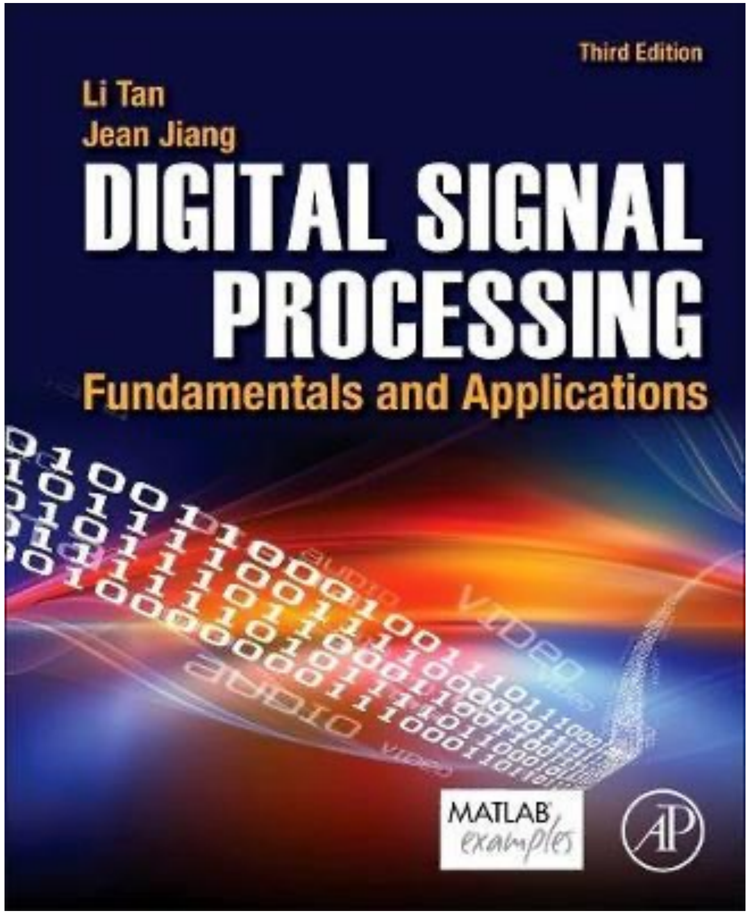
- **Course Instructor**
  - **Dr. Lai-Man Po**
  - eelmpo@cityu.edu.hk
  - Room G6506, Green Zone, 6/F, AC1
  - Phone: 3443-7779
  - http://www.ee.cityu.edu.hk/~lmpo

- **Graders**
  - **Mr. YU Wing Yin, Rocket** (wingyinyu8-c@my.cityu.edu.hk)
  - Room P1412, EE Lab, Purple Zone, 1/F, AC1

Textbook

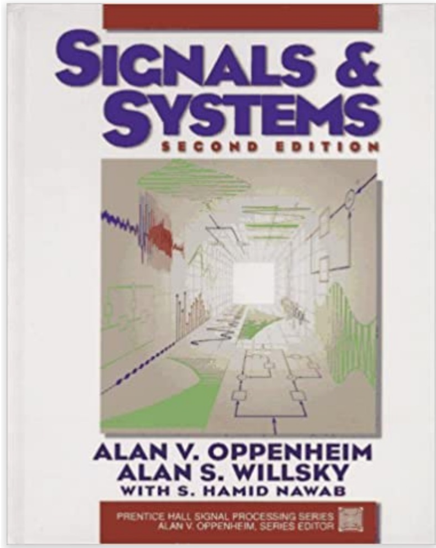
- **Digital Signal Processing : Fundamentals and Applications**
  - **Lizhe Tan & Jean Jiang**

https://books.google.com.hk/books?hl=en&lr=&id=MxlxDwAAQBAJ&oi=fnd&pg=PP1&dq=Digital+Signal+Processing+:+Fundamentals+and+Applications&ots=p8lQTfsoZC&sig=PKbIwC0IrPpAQGLQOuHaLkb37Bw&redir_esc=y#v=onepage&q=Digital%20Signal%20Processing%20%3A%20Fundamentals%20and%20Applications&f=false



Supplemental (Optional) Textbooks

- Oppenheim & Willsky, *Signals and Systems*
  - Textbook for pre-requisite signals & systems course



- J. H. McClellan, R. W. Schafer & M. A. Yoder, *DSP First: A Multimedia Approach*, 1998
  - Demos: http://users.ece.gatech.edu/~dspfirst/

- Steve Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 1997
  - Available **free** online: http://www.dspguide.com

Course Website

- Course Website:
  - http://www.ee.cityu.edu.hk/~lmpo/ee4015/
  - Students should check the course website regularly for announcements about the course.
  - Copies of the lecture, assignment, project information are all available in course website with various formats.
- Messages, Schedule, Projects, Links
  - Username: students
  - Password: dsp2022

Assessment and Grading

- 15% - Three Assignments (Week 5, Week 10, Week 13)
- 5% - Quiz (Week 7)
- 10% Mid-Term Exam (Week 11)
- 20% Group Project
  - Proposal (Week 5)
  - Oral Presentation (Week 12 or 13)
  - Final Report (Week 14)
- 50% Final Exam

Remarks:

- *To pass the course, students are required to achieve at least 35% in course work and 35% in the examination*

Attendance

- Class attendance is required, and you are held responsible for any material or announcements.

Electronic Submission

- All assignments must be submitted in electronic formats such as MS-Words, PDF and source code.
- Soft copies such as MS-Word files and related documents need to be submitted to Canvas by 11:00pm on the due date.
- **Canvas submission format:**
  - To make it easier for graders to process your electronic submissions, please submit them using the following filename format:
  - EE4015_Assignment_Number_Student_Name_Student_Number.pdf
    - For examples:
      - EE4015_Project_Proposal_Group01.pdf
      - EE4015_Assignment01_Chan_Chi_Ming_501234565.pdf

Late Submission Policy

- All assignments and reports mush be uploaded to CANVAS before 11:00PM on the due date.
- NO late assignment is accepted without previous arrangement with the instructor.
- If approved, late submission receives 20% per business day penalty.
- Students may work together on the assignment, but copying is unacceptable.
- **Work to learn. Don't work for marks**

Cheating

- Especially plagiarizing a classmate's assignment or program is a very serious crime! If you are caught cheating, you will automatically receive an F grade in this course and your conduct will be reported to the department for necessary disciplinary action.

- Do not let others copy your assignments or programs, as we cannot tell who is copying whom and you may be penalized.

Readings

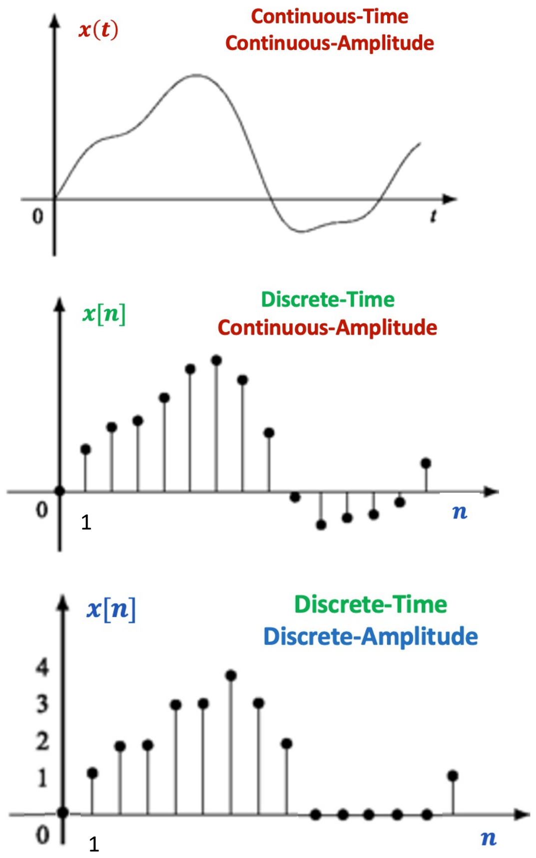
- Reading materials will be assigned to each class. They will be assigned on the course website as well as in previous lectures.

- PLEASE read over the indicated sections before class. Be prepared to discuss the subject intelligently and/or ask questions about material you don't understand.

Introduction to
Digital Signal Processing (DSP)

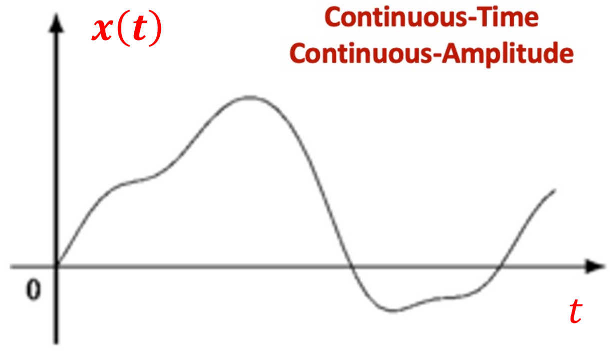
Three Types of Signal in Signal Processing

- **Continuous-Time Signals** (Analog Signals) : $x(t)$
  - Continuous values for time
  - Continuous values for amplitude
- **Discrete-Time Signals** (Sequences) : $x[n]$
  - Discrete values for time
  - Continuous values for amplitude
- **Digital Signals** : $x[n]$
  - Discrete values for time
  - Discrete values for amplitude
    - Data points can only take on a finite number of values



Continuous-Time Signals (Analog Signals)

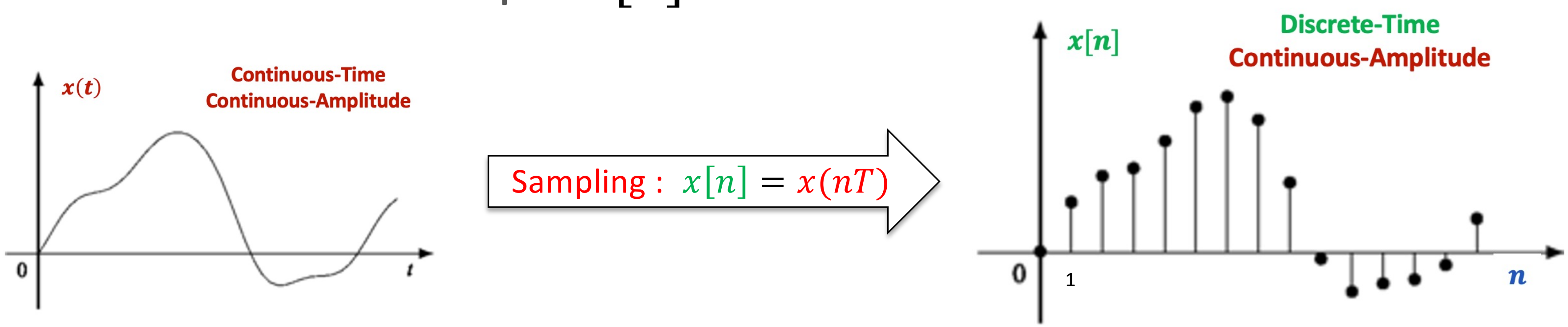
- Continuous-Time (CT) signal is a signal that exists at every instant of time
  - A CT signal is often referred to as analog signal
  - The independent variable t is a continuous variable
  - Continuous signal can assume any value over a continuous range of numbers



- Most of the signals in the physical world are CT signals.
  - Examples: voltage & current, pressure, temperature, velocity, etc.

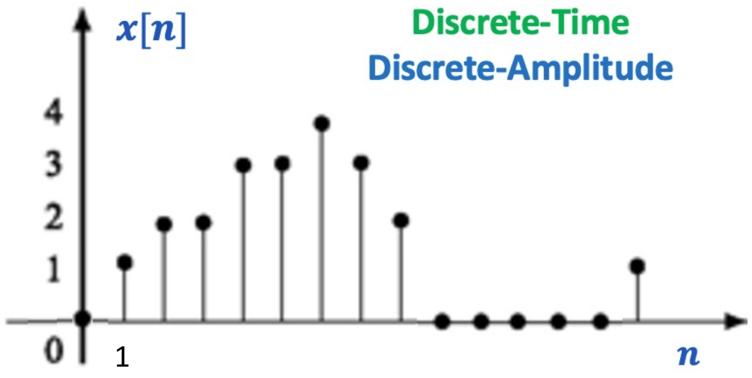
Discrete-Time Signals

- A signal defined only for discrete values of time is called a discrete-time (DT) signal or simply a sequence
- DT signal $x[n]$ can be obtained by taking samples of an analog signal $x(t)$ at discrete instants of time : $x[n] = x(nT)$
- The values of each sample $x[n]$ is continuous



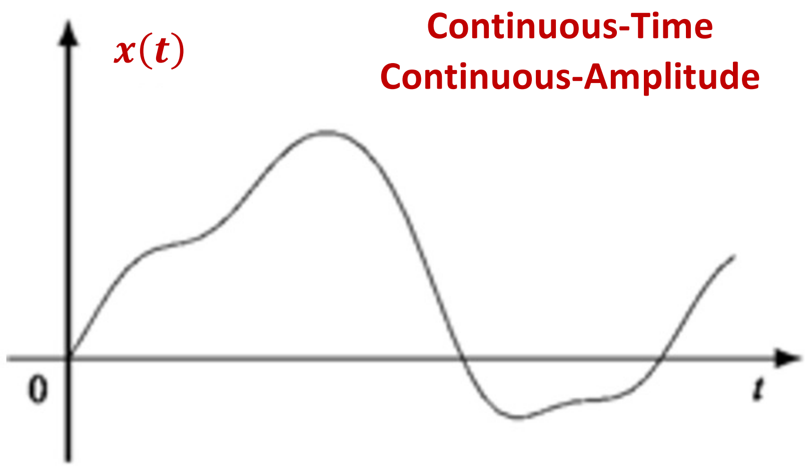
Digital Signals

- Digital signal is a discrete-time signal whose values are **quantized** and represented by digits
  - Discrete-Time
    - $n = \cdots, -3, -2, -1, 0, 1, 2, \ldots$
  - Discrete-Amplitude
    - $x[n] \in \{0,1,2,3,4\}$
    - A finite set of numbers

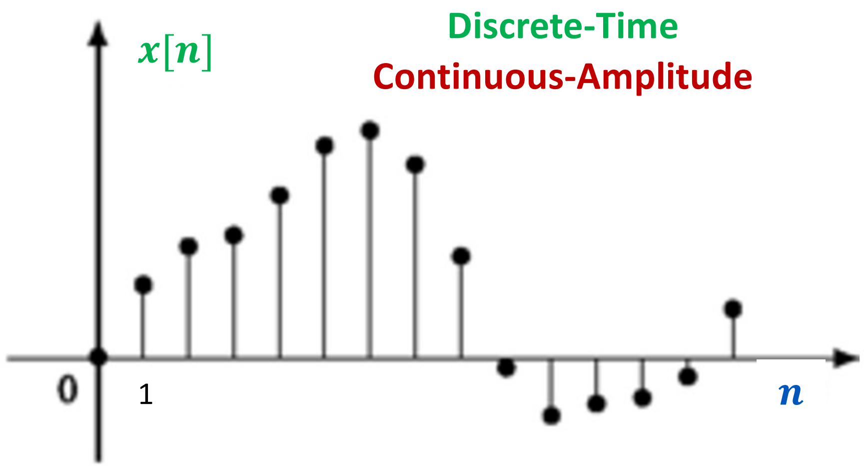


- The digital signal is the sampled and quantized (rounded) representation of the analog signal. A digital signal consists of a sequence of samples, which in this case are integers: 0, 1, 2, 2, 3, 3, 4, 3, ...

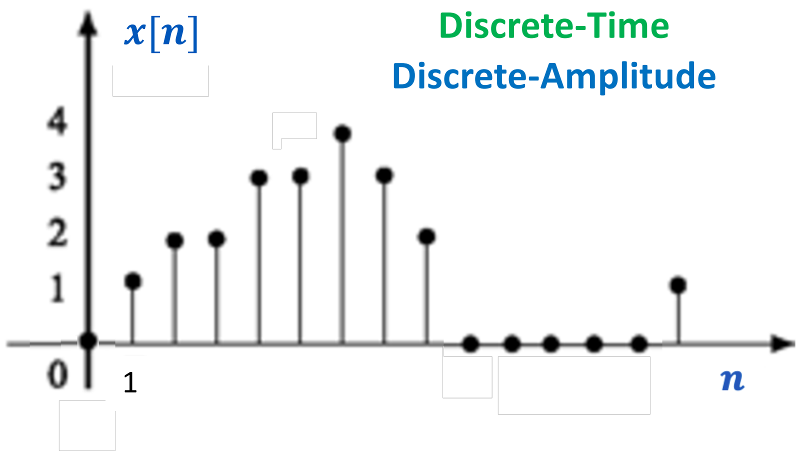
Continuous-Time Signal



Discrete-Time Signal

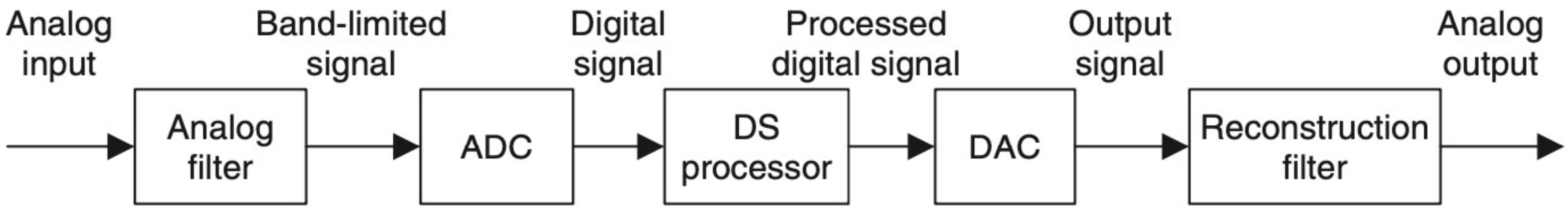


Digital Signal



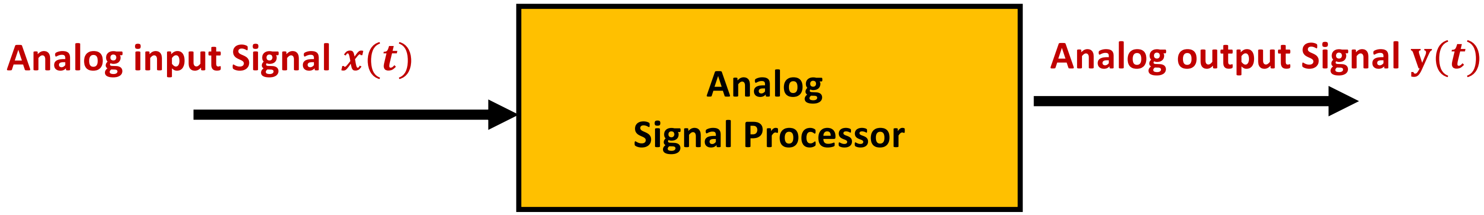
Typical Digital Signal Processing System

- A typical DSP system involving ADC and DAC with the digital signal processed by digital signal processor
  - ADC (Analog-to-Digital Conversion) : Sampling of analog signals to generate digital signals
  - DAC (Digital-to-Analog Conversion) : Reconstruction of analog signals from digital signals

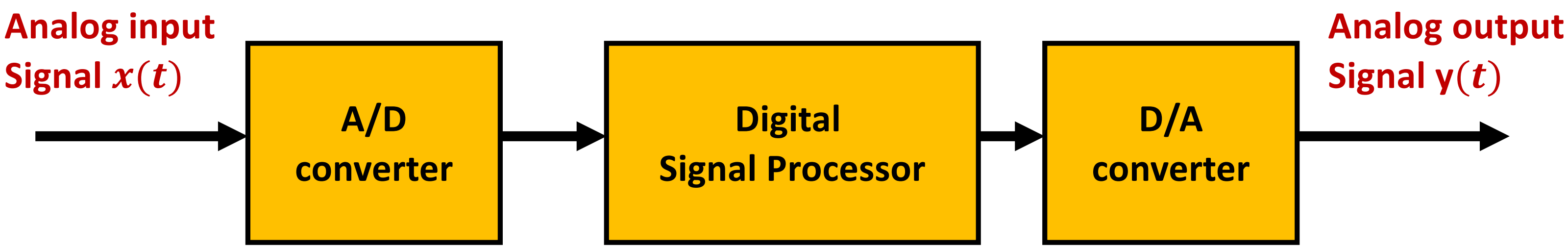


Anti-Aliasing Filter

Analog vs. Digital Signal Processing



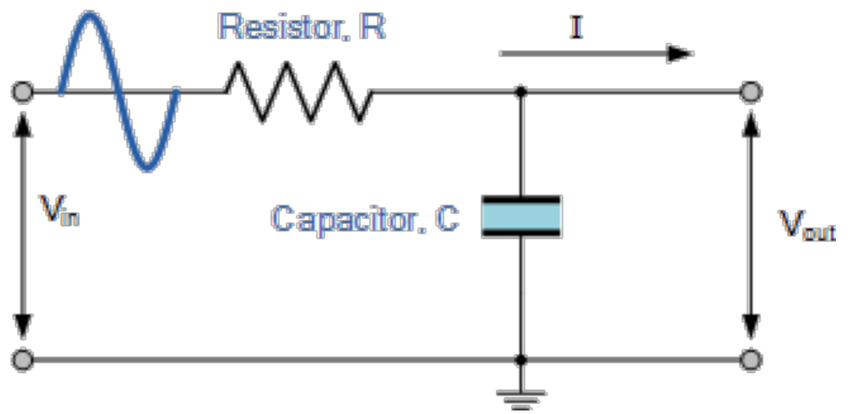
Analog Signal Processing



Digital Signal Processing

Why Not Analog Signal Processing?

- Analog signal processing is achieved by using analog components such as:
  - Resistors
  - Capacitors
  - Inductors



Analog Lowpass Filter

- The inherent tolerances associated with these components, temperature, voltage changes and mechanical vibrations can dramatically affect the effectiveness of the analog circuitry.

Advantages of DSP (1)

- A digital programmable system allows flexibility in reconfiguring the DSP operations simply by changing the program. Reconfiguration of an analogue system usually implies a redesign of hardware, testing and verification that it operates properly.
- DSP provides better control of accuracy requirements.
- Digital signals are easily stored on storage media i.e. hard disk

Advantages of DSP (2)

- The DSP allows for the implementation of more sophisticated signal processing algorithms.
- In some cases, a digital implementation of the signal processing system is cheaper than its analogue counterpart.
- DSP consume relatively less power than analog counterpart.
- DSP processor can be reuse for many applications

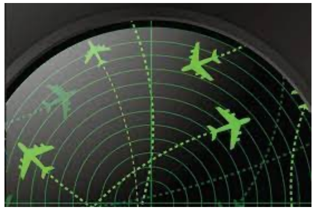
Summary of Why DSP

- **Advantages**
  - Flexible in operation
  - Accurate results
  - Stable system
  - Data storage – less expensive
  - Low cost
- **Disadvantages**
  - Limited speed of operation

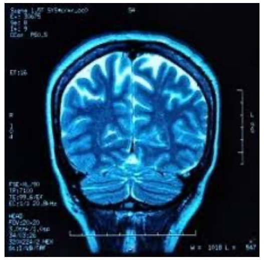
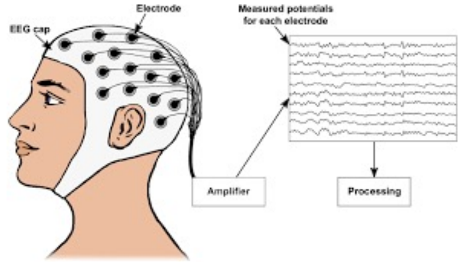
DSP Applications

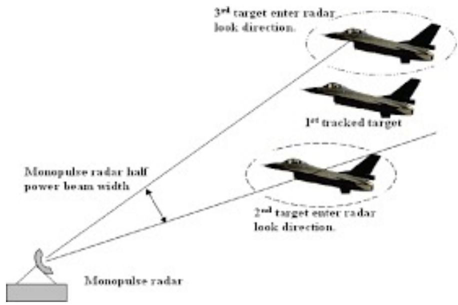




- **Consumer electronics**
  - Smartphone, HDTV, cameras, …
- **Transportation**
  - GPS, engine control, airplane tracking, …
- **Medical**
  - Imaging, monitoring (EEG, ECG), …
- **Military**
  - Target tracking , surveillance, …
- **Remote sensing**
  - Astronomy, climate monitoring, weather forecasting, ...



DSP is Everywhere (1)

- **Sound applications**
  - Compression, enhancement, special effects, synthesis, recognition, echo cancellation,…
  - **Cell Phones, MP3 Players, Movies, Dictation, Text-to-speech,…**
- **Communication**
  - Modulation, coding, detection, equalization, echo cancellation,…
  - Cell Phones, dial-up modem, DSL modem, Satellite Receiver,…
- **Automotive**
  - ABS, GPS, Active Noise Cancellation, Cruise Control, Parking,…

DSP is Everywhere (2)

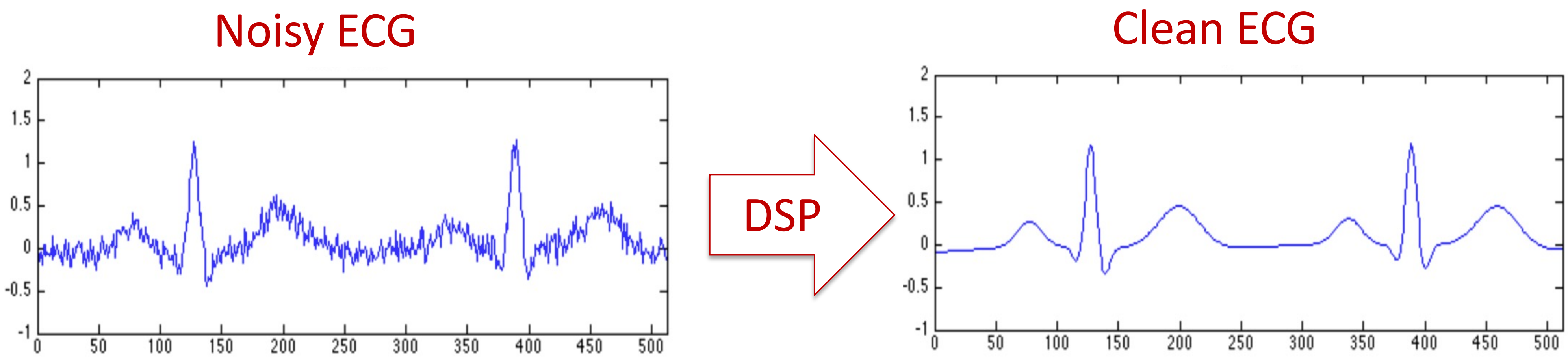
- **Medical**
  - Magnetic Resonance, Tomography, Electrocardiogram,…
- **Military**
  - Radar, Sonar, Space photographs, remote sensing,…
- **Image and Video Applications**
  - **DVD, JPEG, Movie special effects, video conferencing,…**
- **Mechanical**
  - Motor control, process control, oil and mineral prospecting,…

Typical Signal Processing Problems

1. Eliminating **Nosie**
2. Signal **Restoration** (Correcting distortion)
3. Extracting an **indirect quantity from measured signals**
4. Signal **Compression**

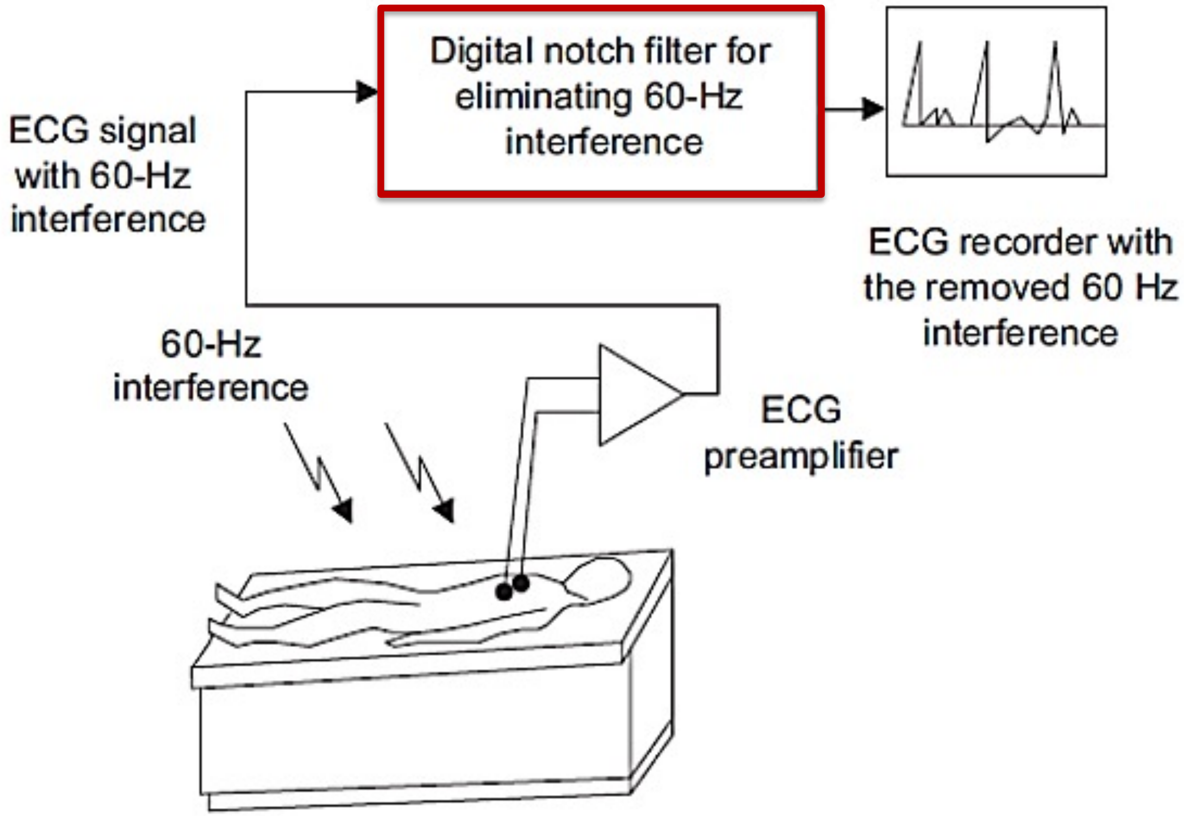
Typical Signal Processing Problems (1)

- Eliminating **Nosie**

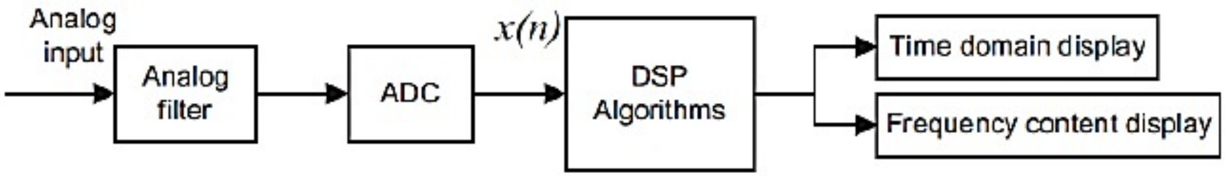


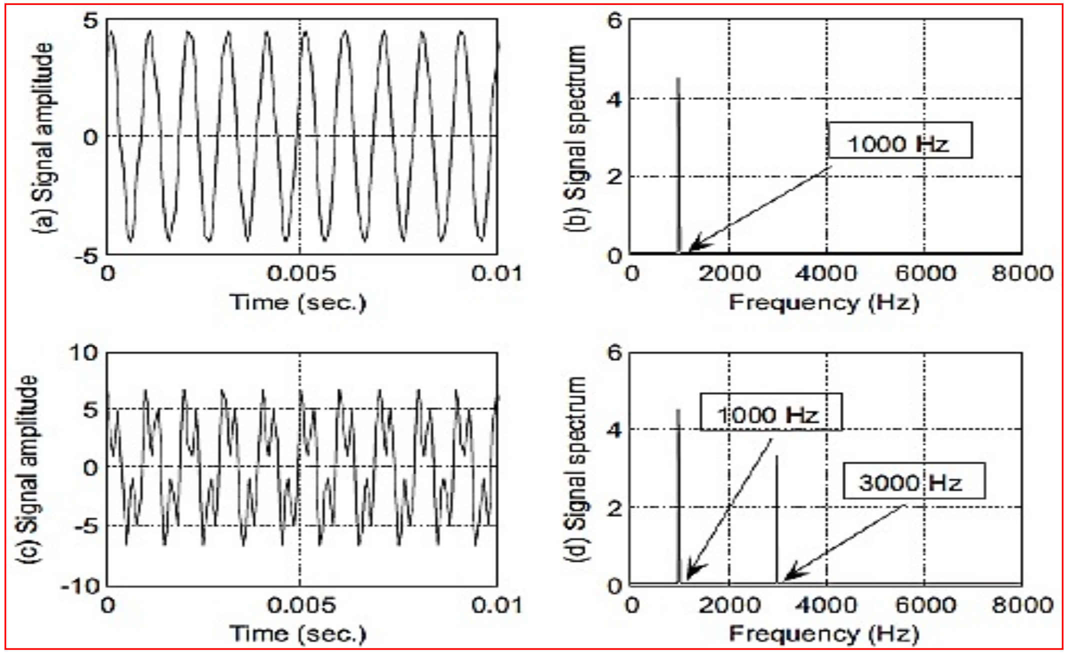
DSP Example : Nosie Removal

- 50/60 Hz Interference Cancellation in Electrocardiography



DSP Example : Signal Spectrum Analysis





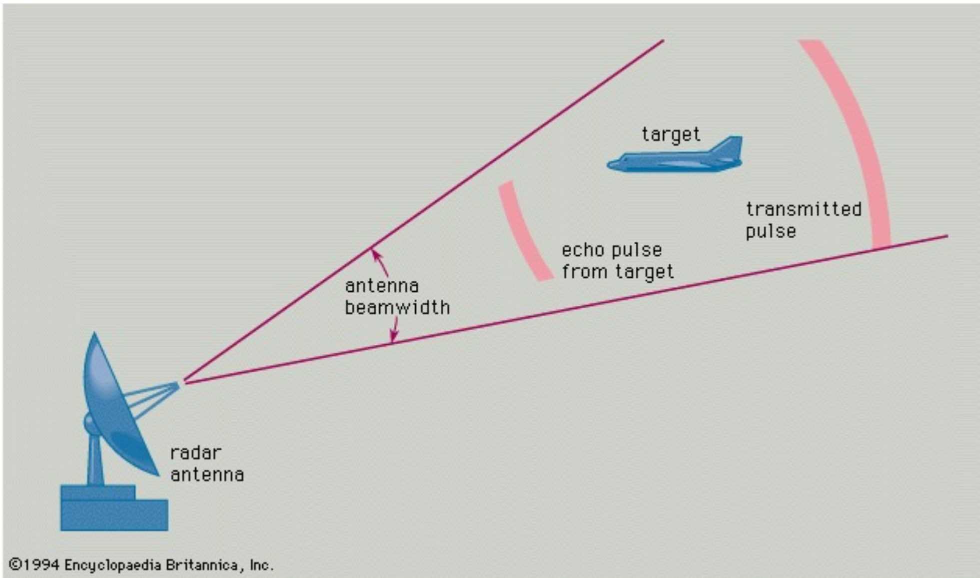
Typical Signal Processing Problems (3)

- Signal **Restoration** (Correcting distortion)

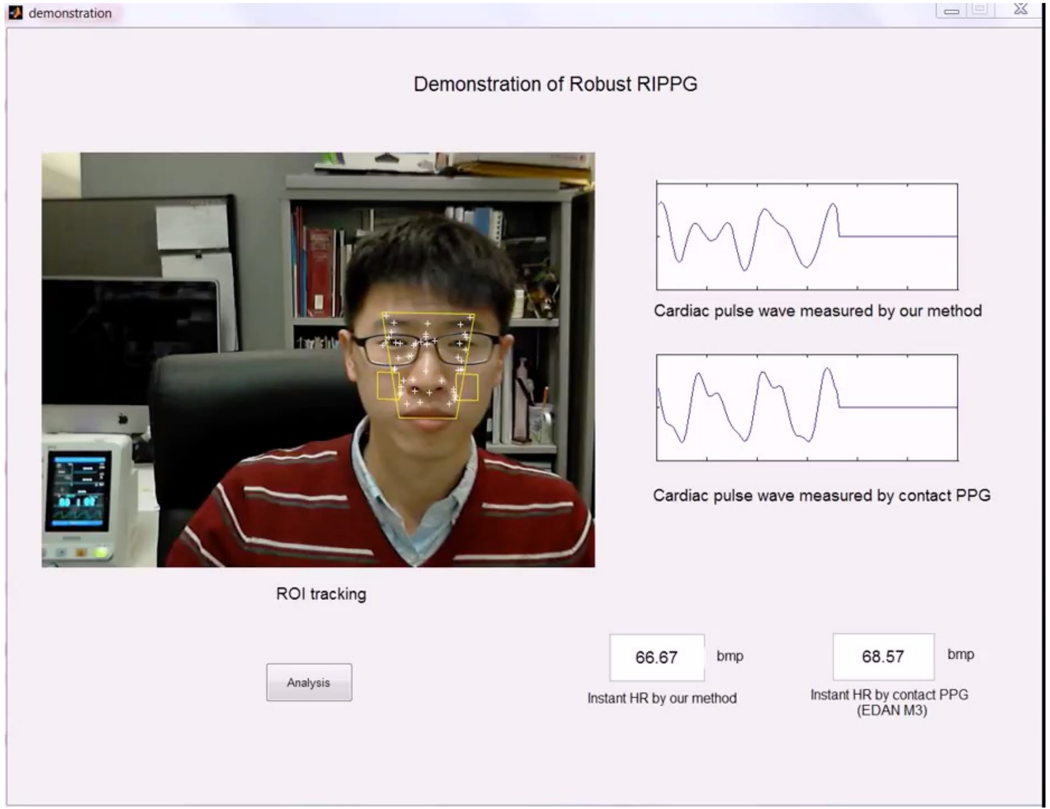
Motion Blur Restored Image

Typical Signal Processing Problems (3)

- **Extracting an indirect quantity from measured signals**
  - Determine aircraft position and velocity



Motion Resistant Remote-PPG for Heart Rate Monitoring



https://www.youtube.com/watch?v=8X2nJcVsmYQ

Real-Time Face Liveness Detection

https://www.youtube.com/watch?v=t4O2bbF61Tk

Typical Signal Processing Problems (4)

- Signal **Compression**
  - Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission.

Audio Compression Technology

- **Audio Recording Technology** (1877 to 2020)
  - From Phonograph to High-Res Music Streaming
  - Phonograph, LP Phonograph, Magnetic Tape Audio, Compact Disc Digital Audio, MP3 Music, Hi-Res Audio
- **Digital Audio Coding Technology**
  - Lossless and Lossy Audio Compression
  - Audio Coding File Format
  - Audio Coding Standards
    - MPEG-1 Audio Coding
    - MPEG-2 Layer 3 : MP3
    - Advanced Audio Coding : ACC

CD Players

- A **CD player** is an electronic device that plays audio compact discs, which are a digital optical disc data storage format.
- CD players were first sold to consumers in 1982.

Cambridge Audio AXC35

In 1984, Sony introduced the first portable CD player
Sony Discman D-50

Hi-Res Audio – 2014

- Hi-Res Audio (High Resolution Audio)
- Hi-Res Audio describes digital audio files have higher sample rate and bit dept than Compact Disc Digital Audio (CD-DA) of 44.1kHz/16-bit.
  - Hi-Res Sample Rate: 96kHz to 192kHz
  - Hi-Res Bit Depth: 24-bit
- Hi-Res audio is the favorite of audiophiles.
- Today, Hi-Res music and songs are available in special audio file formats (FLAC and ALAC), and widely are widely promoted by audio streaming services such as Tidal, Spotify, and Moov.

Hi-Res Audio File Formats

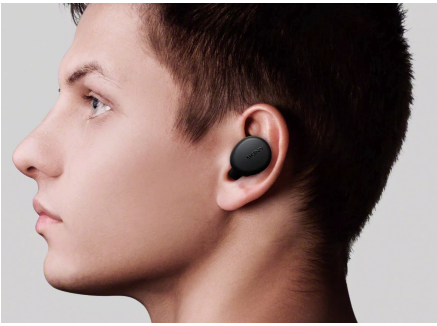
Common Hi-Res Audio File Formats can be classified as:

- Uncompressed Formats (No compression):
  - WAV, AIFF
  - DSD (Direct Stream Digital) - Single bit encoding with 64 time oversampling of CD audio
- Lossless Compressed Formats (40 – 60% compression):
  - **FLAC** – Free Lossless Audio Codec (2001)
  - **ALAC** – Apple Lossless Audio Codec (2004)
- Lossy Compressed Formats (Higher compression):
  - **MQA** – Master Quality Authenticated (2014)
  - OGG – It is a license-free and open-source container format that can multiplex a number of independent streams for audio, video, text and metadata.

Bluetooth Audio Codecs – 2020s

Mainly used for Bluetooth headphone communication especially true wireless Earbuds:

- **SBC** (Sub-Band Codec) – 192 to 320kbps
- **ACC** (Advanced Audio Coding) – Max 250kbps
- Qualcomm's **AptX**
  - aptX (48kHz/16-bit at 352kbps), aptX LL, aptX HD (48kHz/24-bit at 576kbps) and aptX Adaptive
- Sony **LDAC** – Variable bit rate
- HWA Alliance's **LHDC** and **LLAC** codecs
  - Low-latency and high-definition audio codec – 900kbps
  - Low-latency audio codec – 30ms latency



https://www.soundguys.com/understanding-bluetooth-codecs-15352/

Main Topics of this DSP Course

- **Review of Signals and Systems**
  - Review of Mathematics
  - Review of CT Signals and Systems
  - Colab Python for DSP
- **Discrete-Time Signals and Systems**
  - Discrete-Time Signals
  - Discrete-time Systems : Casual, Stable, and LTI Systems
  - Impulse Response and Convolution
- **Sampling and Reconstruction**
  - Review of CT Fourier Analysis
  - Modelling of Sampling and reconstruction Processes in CT Fourier Domain
  - Quantization
- **Discrete-Time Transformations**
  - Discrete-Time Fourier Transform (DTFT)
  - Discrete Fourier Series (DFS)
  - Discrete Fourier Transform (DFT)
  - Fast Fourier Transform (FFT)
  - The z-Transform
  - Transform Domain Analysis of LTI Systems
- **Digital Filter Design**
  - Realization of Digital Filter
  - Analog Filter Design
  - IIR and FIR Digital Filter Design
  - Multirate Digital Signal Processing (MDSP)
- **Applications of DSP**
  - Audio Recording Technology
  - Audio Compression Technology

Digitization of Analog Signals

- **Sampling and Reconstruction**
  - Time-Domain Modelling using Impulse Train Modulation
  - Frequency-Domain Analysis of the sampled signal spectrum
  - Sampling Theorem, Nyquist Sampling Rate, and Aliasing Effect
  - **Reconstruction of the continuous-time signal from discrete-time signal**
- **Quantization**
  - Resolution in terms of number of bits per sample (bit depth)
  - Quantization Effect and Quantization Noise
  - Signal-to-Quantization-Noise Ratio (SQNR)
  - Pulse Code Modulation (PCM)

A Big Picture of Transformations for Signal Processing

Continuous-Time Signals

Periodic : $\tilde{x}(t)$

- Continuous-Time Fourier Series (CTFS)
  - Commonly called Fourier Series (FS)

Non-Periodic (Aperiodic) : $x(t)$

- Continuous-Time Fourier Transform (CTFT) : $X(j\Omega)$
  - Commonly called Fourier Transform (FT)

Generalization

- Laplace Transform : $X(s) = X(\sigma + j\Omega)$
  - For system design

Discrete-Time Signals (Sequences)

Periodic : $\tilde{x}[n]$

- Discrete Fourier Series (DFS)
  - also called Discrete-Time Fourier Series (DTFS)

Non-Periodic (Aperiodic) : $x[n]$

- Discrete-Time Fourier Transform (DTFT) : $X\left(e^{j\omega}\right)$

Finite-Duration Sequences

- Discrete Fourier Transform (DTF) : $X[k]$
- Fast Fourier Transform (FFT) : $X[k]$

Generalization

- The z-Transform : $X(z) = X\left(re^{j\omega}\right)$

Digital Filter Design

Realization of Digital Filters

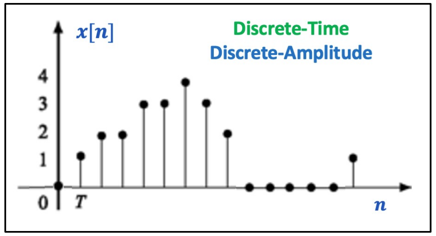
- Non-Recursive Digital Filters
  - No Feedback Path
  - Commonly used for Finite Impulse Response (FIR) filter implementation
- Recursive Digital Filters
  - At least one feedback path
  - Used for Infinite Impulse Response (IIR) filter implementation
- Structures for Digital Filter
  - Direct Form
  - Canonic Form
  - Cascade Form
  - Parallel Form

IIR and FIR Filter Design

- FIR Filter Design
  - Windowing Method
  - Frequency Sampling Method
  - The Optimal Parks-McCellan Method
- Basic Analog Filter Design
- IIR Filter Design
  - Mapping from analog filter
  - Impulse Invariant Method
  - Bilinear Transform Method
  - Digital-to-Digital Transformation
- Multirate Digital Signal Processing
  - Down-Sampler and Up-Sampler

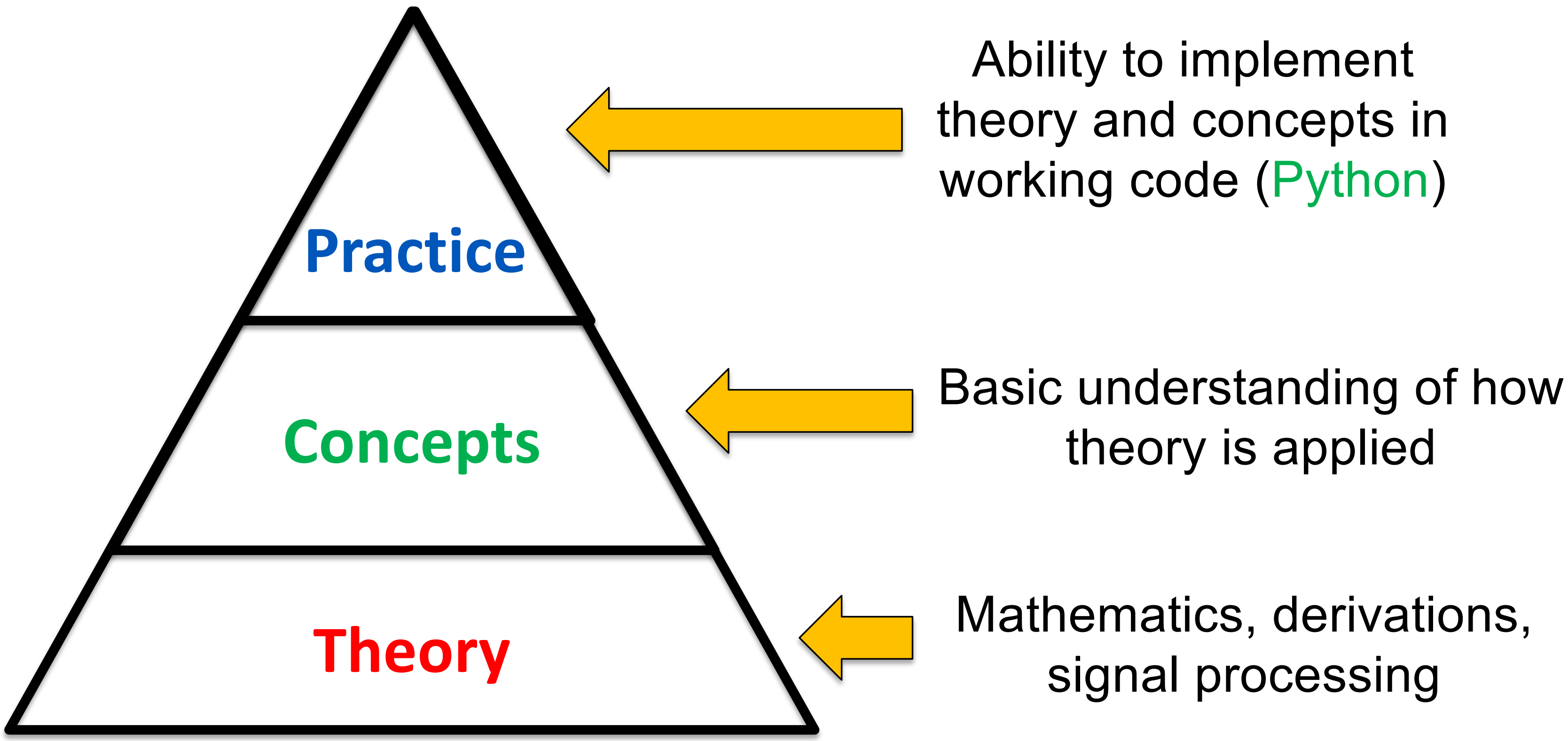
Four-Level of Understanding of DSP

- **Natural Language Understanding**
  - Use human language to describe the DSP concepts, problem and solution
  - e.g. Digital signals are easily stored on storage media i.e. hard disk
- **Visual Understanding**
  - Use figures to describe the DSP concepts, problem and solutions



- **Mathematical Understanding**
  - Use mathematical equations to represent the DSP concepts
  - e.g. DT signal $x[n]$ can be obtained by taking samples of an analog signal $x(t)$ at discrete instants of time : $x[n] = x(nT)$
- **Implementation of real DSP system**
  - Using Google Colab with Python programming language and other advanced signal processing packages to implement DSP applications

Learning Outcome of this EE4015 DSP Course



Need to understand signal processing at all three levels

Special Feature of this Course

- Students are able to apply the basic digital signal processing principles and Python based tools to create, alter and store 1-D and 2-D digital signals of audio, image or video.
- Students are able to apply digital signal processing techniques to solve practical problems by implementing algorithms in Python programming languages using powerful libraries of Pytorch, Numpy, ScipPy, OpenCV, etc.

49

Python based Signal Processing Platforms

- **Python 3** as the programming language
- **Anaconda** as the software tool
  - Anaconda is Scientific Python Distribution
  - Including many pre-installed python libraries
    - numpy, matplotLib, opencv, etc
- **Jupyter Lab (Jupyter notebook)** is a new web-based Python programming tools that enables us to work with documents and activities such as text editors, terminals, integrated, etc.
- **Google Colab** is the online version of Jupyter notebook that provided by Google.

Installation of Python Environment

- Anaconda Python Package:
  - https://www.continuum.io/downloads
- Use pip install command for additional package installation
  - **`pip install opencv-python`**
  - **`pip install opencv-python==3.4.2.17.7`**
  - **`pip install scikit-image`**
- We can use **conda** package manager of the Anaconda to install non-python packages such as OpenCV
  - **`sudo conda install -c conda-forge opencv`**

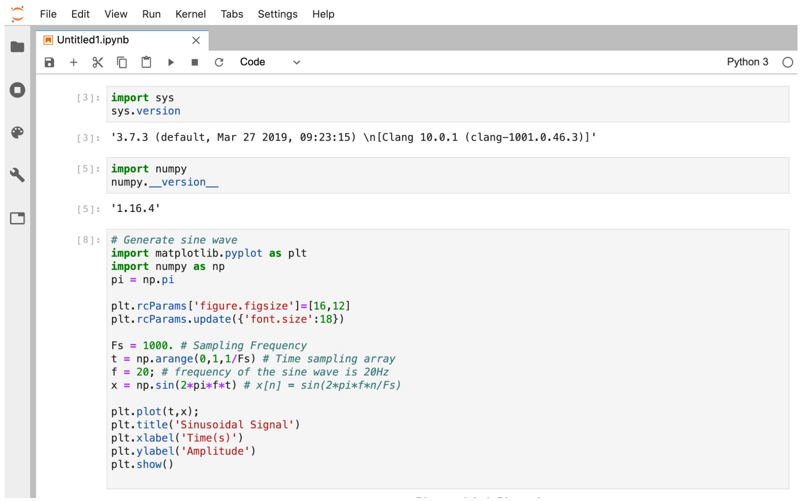
Use Python3 in Terminal

```
po — Python — 80×24
Last login: Sat Aug  8 15:36:47 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Lais-MBP-2:~ po$ python3
Python 3.7.3 (default, Mar 27 2019, 09:23:15)
[Clang 10.0.1 (clang-1001.0.46.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.version
'3.7.3 (default, Mar 27 2019, 09:23:15) \n[Clang 10.0.1 (clang-1001.0.46.3)]'
>>> import numpy as np
>>> np.__version__
'1.16.4'
>>>
```

```
import sys
import skimage
import cv2

print("Python {}".format(sys.version))
print("SKimage: {}".format(skimage.__version__))
print("OpenCV Version: {}".format(cv2.__version__))
```

Jupyter Notebook and Jupyter Lab

- **JupyterLab** is a next-generation web-based user interface for Project **Jupyter**.
- **JupyterLab** enables you to work with documents and activities such as **Jupyter**notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner.
- Installation:
  - pip install jupyternotebook
  - pip install jupyterlab
- Run :
  - jupyter notebook
  - jupyter lab



Jupyter Lab Example

File Edit View Run Kernel Tabs Settings Help

Untitled1.ipynb

Code — Python 3

[3]:
```python
import sys
sys.version
```

[3]: '3.7.3 (default, Mar 27 2019, 09:23:15) \n[Clang 10.0.1 (clang-1001.0.46.3)]'

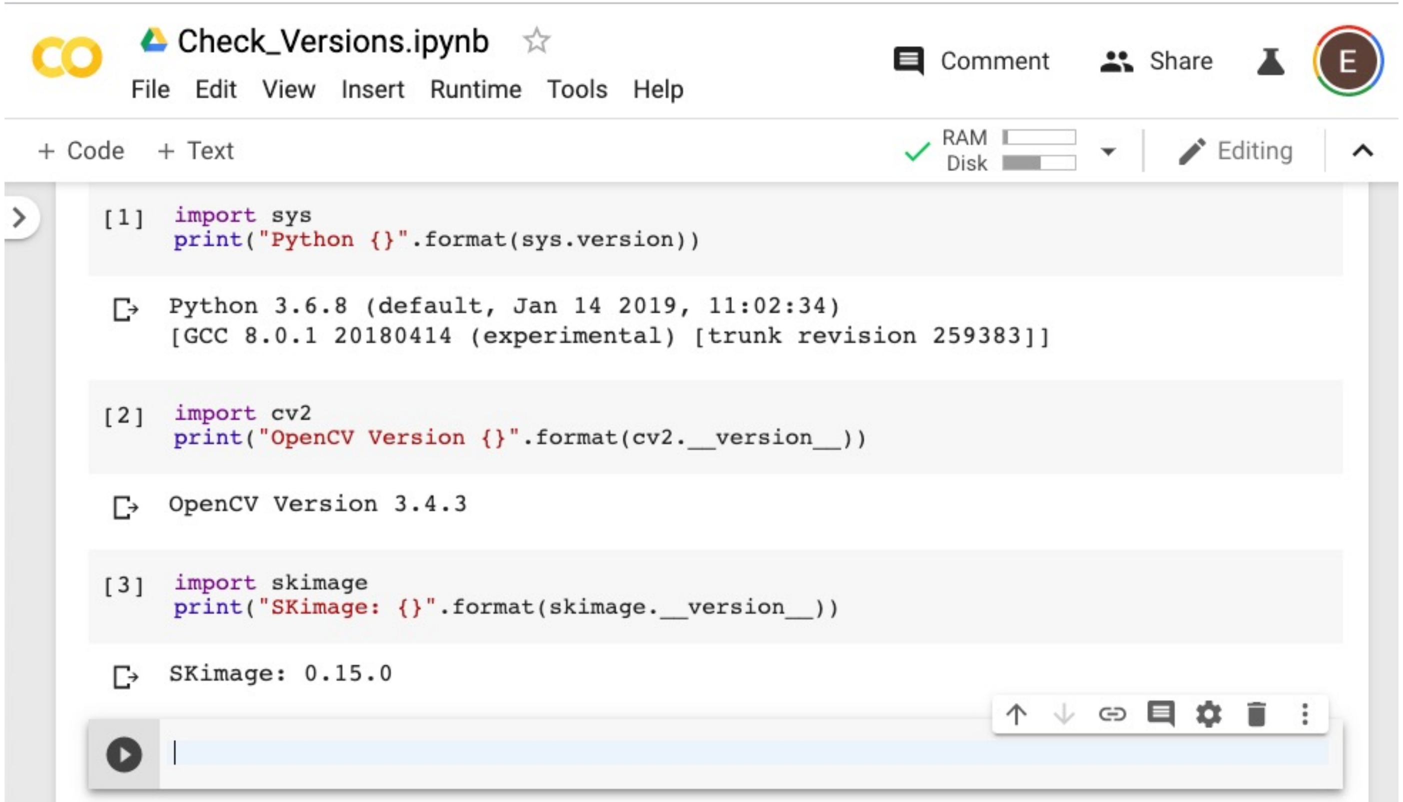
[5]:
```python
import numpy
numpy.__version__
```

[5]: '1.16.4'

[8]:
```python
# Generate sine wave
import matplotlib.pyplot as plt
import numpy as np
pi = np.pi

plt.rcParams['figure.figsize']=[16,12]
plt.rcParams.update({'font.size':18})

Fs = 1000. # Sampling Frequency
t = np.arange(0,1,1/Fs) # Time sampling array
f = 20; # frequency of the sine wave is 20Hz
x = np.sin(2*pi*f*t) # x[n] = sin(2*pi*f*n/Fs)

plt.plot(t,x);
plt.title('Sinusoidal Signal')
plt.xlabel('Time(s)')
plt.ylabel('Amplitude')
plt.show()
```

Google Colab : http://colab.research.google.com

- Automatic setting-up, getting help effectively, collaborative programming, and version control. A one-stop solution to the pain points in Python beginners' practice.



https://colab.research.google.com/drive/1XSfQVeErVSvSpMwAPIm_KsMJ1tAIvsTb?usp=sharing

Numpy and SciPy Packages

- NumPy (Numeric Python) Package
  - Provides basic functions for manipulating arrays and matrices of numeric data
- SciPy (Scientific Python) package
  - Extends the functionality of NumPy
  - Uses NumPy arrays as the basic data structure
  - Creates modules for scientific programming
    - Linear Algebra
    - Integration (Calculus)
    - Ordinary Differential Equation (ODE)
    - **Signal Processing**

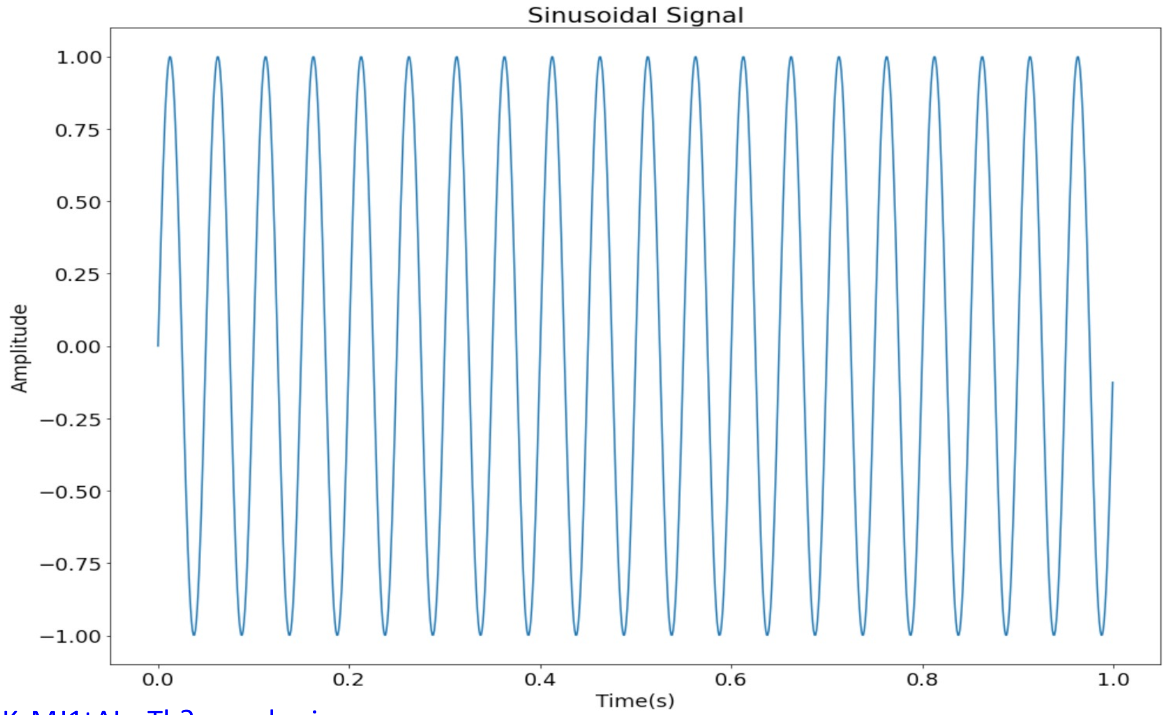
https://www.youtube.com/watch?v=b06pFMIRO0I

Python : Generation of Sinusoidal Waveform

```
# Generate sine wave
import matplotlib.pyplot as plt
import numpy as np
from math import pi

Fs = 1000. # Sampling Frequency
t = np.arange(0,1,1/Fs) # Time sampling array
f = 20 # frequency of the sine wave is 20Hz

x = np.sin(2*pi*f*t) # x[n] = sin(2*pi*f*n/Fs)

plt.rcParams['figure.figsize']=[16,12]
plt.rcParams.update({'font.size':18})

plt.plot(t,x);
plt.title('Sinusoidal Signal')
plt.xlabel('Time(s)')
plt.ylabel('Amplitude')
plt.show()
```

np.arrange( ) - Return evenly spaced time vector (1/Fs) between [0,1)

np.sin( ) - Input angle(2π x time vector) in radians and return value ranging from -1 to +1

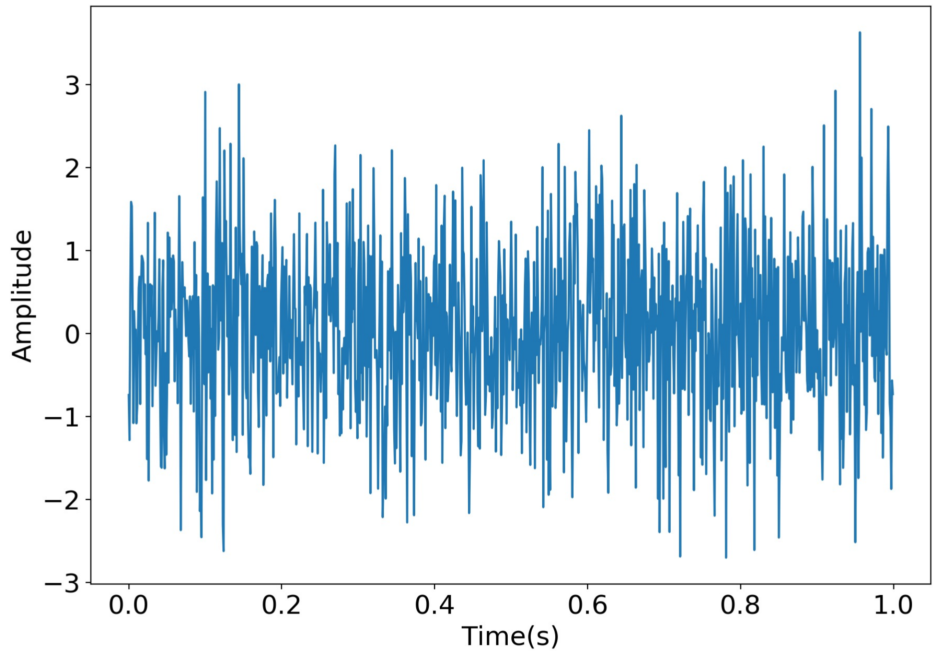


https://colab.research.google.com/drive/1XSfQVeErVSvSpMwAPIm_KsMJ1tAIvsTb?usp=sharing

Python : Generation of Nosie Waveform

```
# Generate Random Siganl
import matplotlib.pyplot as plt
import numpy as np

Fs = 1000. # Sampling Frequency
t = np.arange(0,1,1/Fs)

x = np.random.randn(t.size)

plt.rcParams['figure.figsize']=[16,6]
plt.rcParams.update({'font.size':18})
plt.plot(t,x);
plt.title('Random Noise Signal')
plt.xlabel('Time(s)')
plt.ylabel('Amplitude')
plt.show()
```

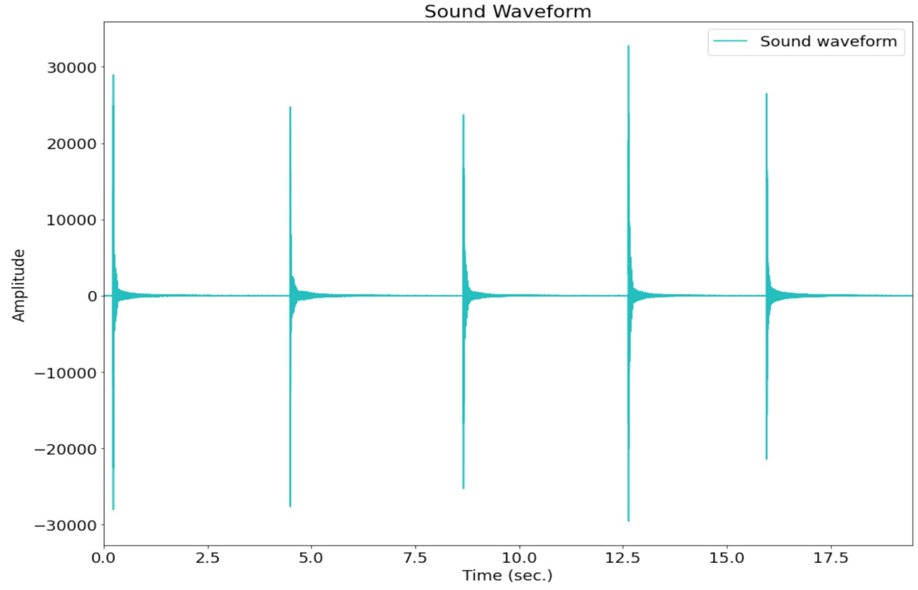
np.random.randn( ) – Return samples from the standard normal distribution of mean 0 and variance 1



Load Audio File and Play the Sound

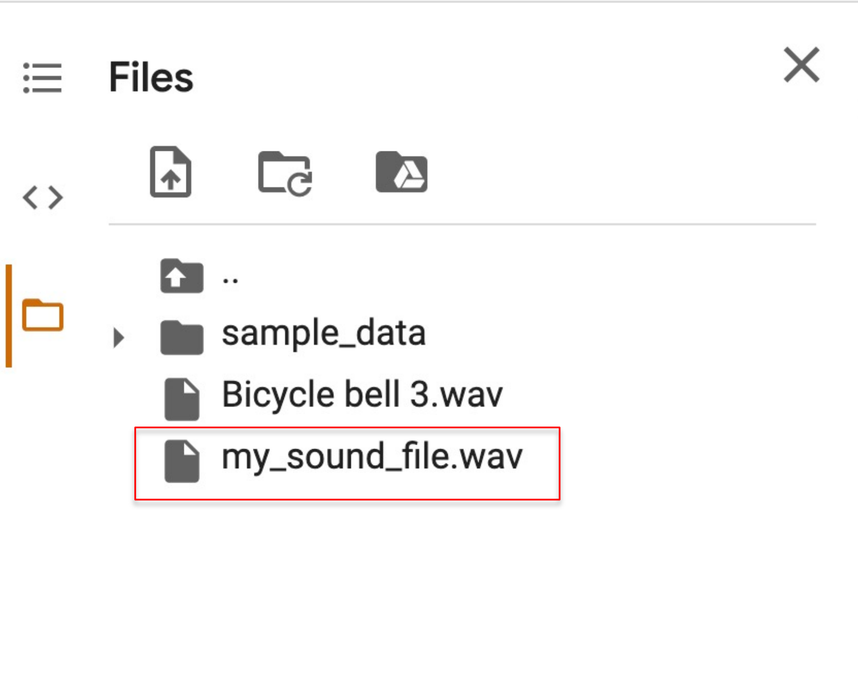
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.io import wavfile

# Load sound file from GitHub
!wget https://github.com/R6500/Python-bits/raw/master/Colaboratory/Sounds/Bicycle%20bell%203.wav

# Load the file on an object
data = wavfile.read('Bicycle bell 3.wav')

# Separete the object elements
Fs = data[0] # Sample Rate
x = data[1] # Signal Data
n = x.size # Number of samples
T = 1/Fs # Sample Period T
t = np.arange(0, n)*T # Time vector

# Show information about the object
print('Sample rate:',Fs,'Hz')
print('Total time:',n*T,'s')

plt.plot(t,x,color='c',LineWidth=1.5,label='Sound wave')
plt.xlim(t[0],t[-1])
plt.title('Sound Waveform')
plt.ylabel('Amplitude')
plt.xlabel('Time (sec.)')
plt.legend()
plt.show()
```

```
Sample rate: 48000 Hz
Total time: 19.4594375 s
```




```
[246] # Play the mono sound
      from IPython.display import Audio
      Audio(x,rate=Fs, autoplay=True)
```

Save Wave file



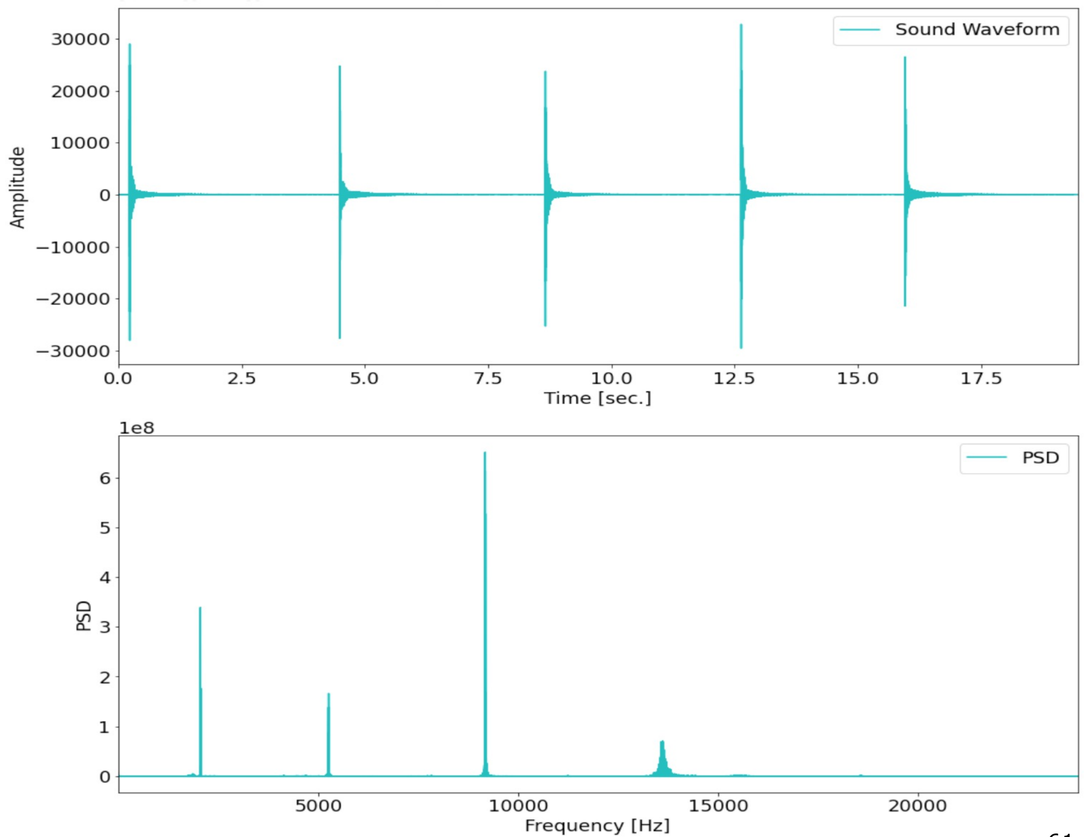
+ Code + Text

```
[13] # Save Waveform into a Wave File

     from scipy.io.wavfile import write

     # Set Sampling Frequence
     Fs = 48000

     write('my_sound_file.wav', Fs, x)
```

Frequency Analysis using SciPy fftpack

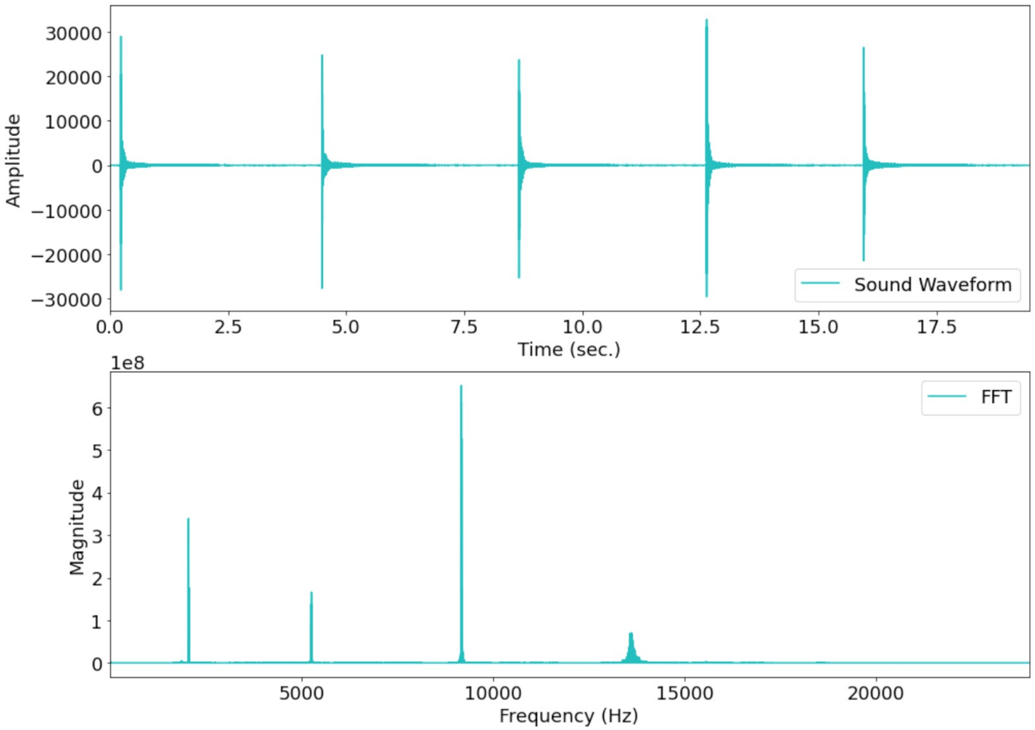
```
from scipy import fftpack

X = fftpack.fft(x)
magnitude = np.abs(X)
PSD = X * np.conj(X)/n
phase = np.angle(X)

freq = fftpack.fftfreq(n, d=T)
L = np.arange(1,np.floor(n/2),dtype='int')

fig,axs = plt.subplots(2,1)

plt.sca(axs[0])
plt.plot(t,x,color='c',LineWidth=1.5,label='Sound Waveform')
plt.xlim(t[0],t[-1])
plt.ylabel('Amplitude')
plt.xlabel('Time [sec.]')

plt.sca(axs[1])
plt.plot(freq,PSD,color='c',LineWidth=1.5,label='PSD')
plt.xlim(freq[L[0]],freq[L[-1]])
plt.ylabel('PSD')
plt.xlabel('Frequency [Hz]')
plt.legend()
plt.show()
```

scipy.fftpack.fftfreq() : Return the Discrete Fourier Transform sample frequencies



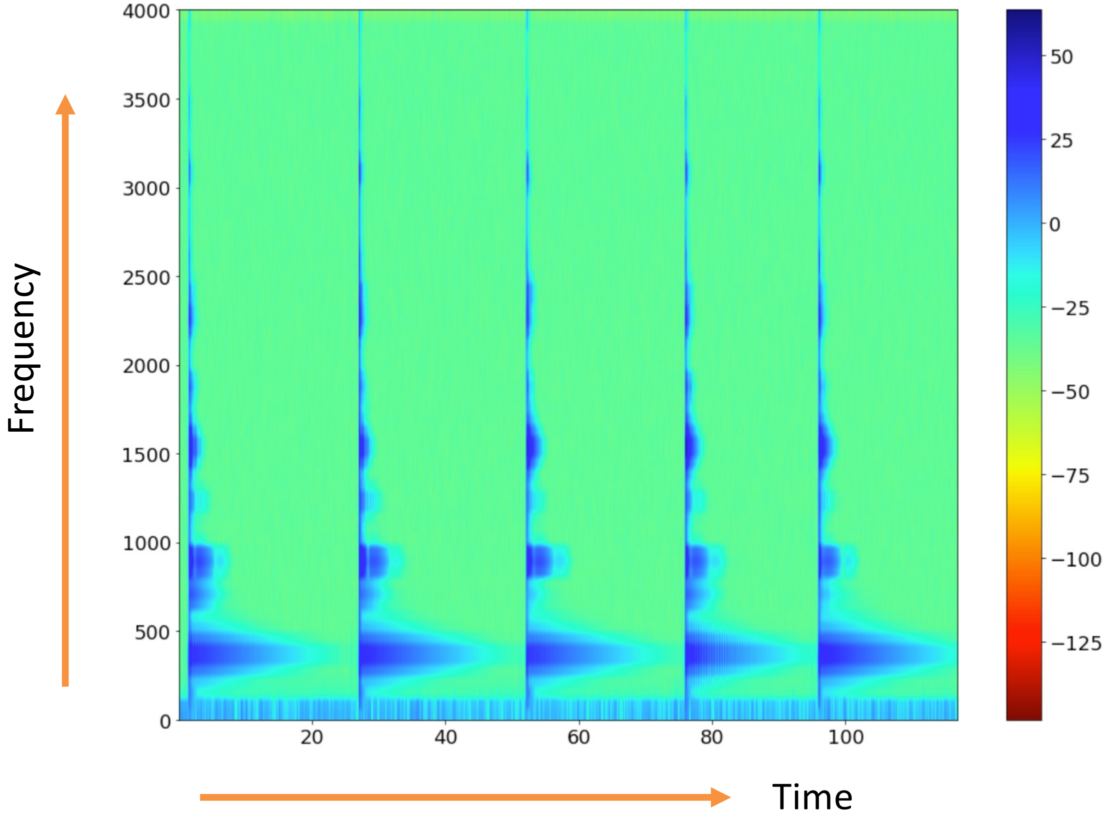
Frequency Analysis using Numpy FFT

```
n = x.size
X = np.fft.fft(x,n)
PSD = X * np.conj(X)/n
freq = (Fs/n)*np.arange(n)
L = np.arange(1,np.floor(n/2),dtype='int')

fig,axs = plt.subplots(2,1)

plt.sca(axs[0])
plt.plot(t,x,color='c',LineWidth=1.5,label='Sound Waveform')
plt.xlim(t[0],t[-1])
plt.ylabel('Amplitude')
plt.xlabel('Time (sec.)')
plt.legend()

plt.sca(axs[1])
plt.plot(freq[L],PSD[L],color='c',LineWidth=1.5,label='FFT')
plt.xlim(freq[L[0]],freq[L[-1]])
plt.ylabel('Magnitude')
plt.xlabel('Frequency (Hz)')
plt.legend()
plt.show()
```



Spectrogram : Time-Frequency Analysis

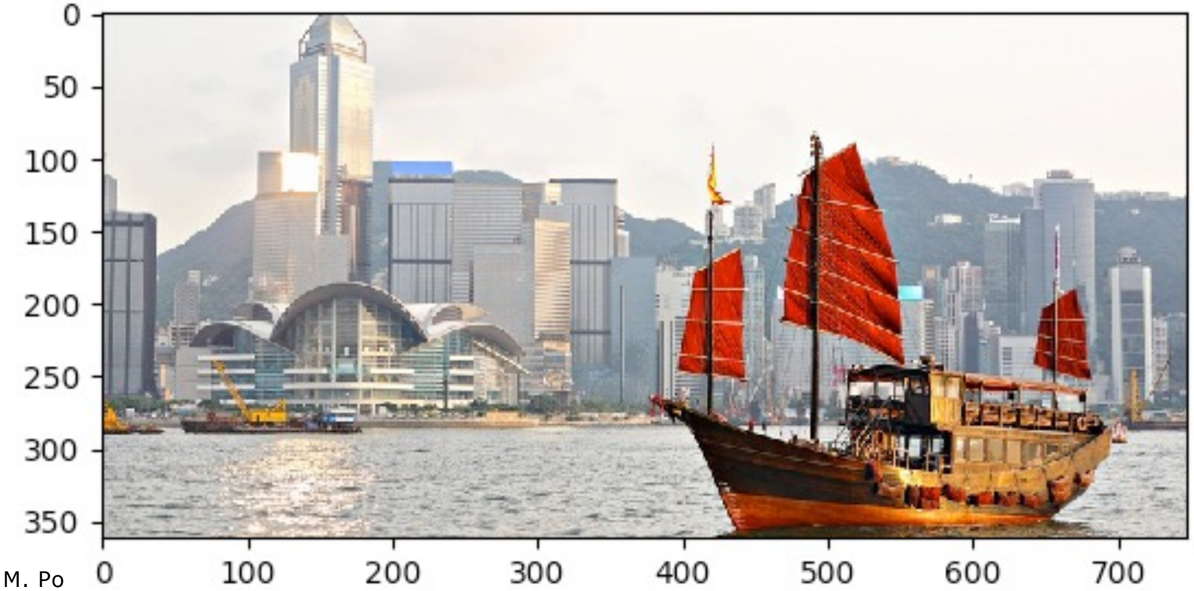
```
plt.specgram(x, NFFT=128, Fs=Fs, noverlap=120, cmap='jet_r')
plt.colorbar()
plt.show()
```



2-D Signals : Read and Display Image

```python
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
img=mpimg.imread('hongkong.jpg')
imgplot = plt.imshow(img)
plt.show()
```

display_plt.py




```python
import numpy as np
import cv2

# Load an color image in grayscale
img = cv2.imread('hongkong.jpg', 1)

# Display the image
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWinows()
```

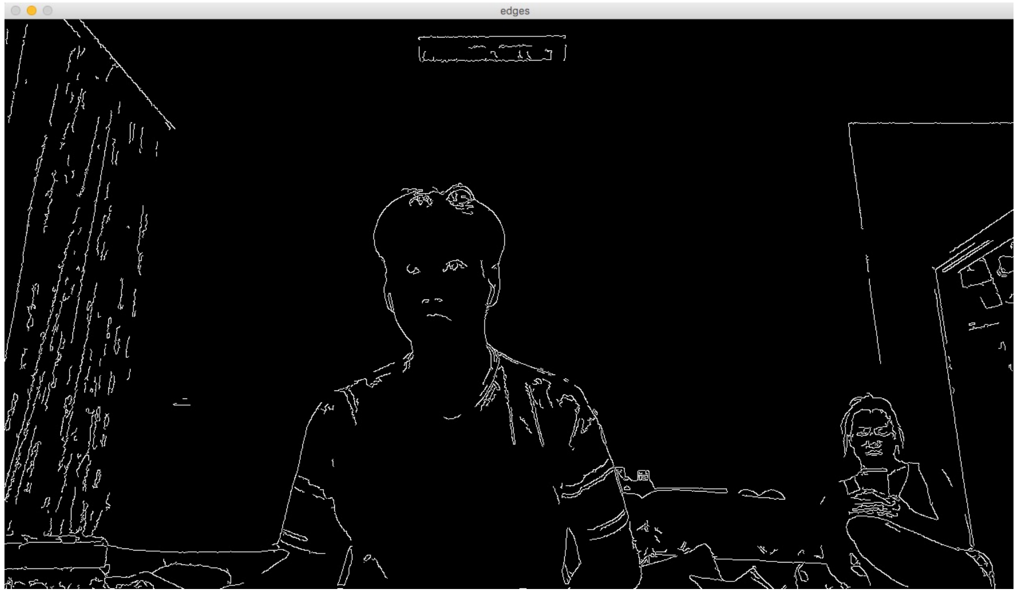
display_cv2.py

Real-Time Video Processing using OpenCV

```python
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while(1):

    # Take each frame
    _, frame = cap.read()

    # Convert BGR to GRAY
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Find the edges by the Canny Edge Detector
    edges = cv2.Canny(frame, 100,200)

    cv2.imshow('frame',frame)
    cv2.imshow('edges',edges)

    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
```

canny.edge.py

Real-Time Canny Edge Detection



Real-Time Face Detection using OpenCV

```python
import cv2

video_capture = cv2.VideoCapture(0)
detector = FaceDetector("xml/haarcascade_frontalface_default.xml")

while True:
    ret, frame = video_capture.read()
    # frame=cv2.resize(frame,(800,480))
    faces_coord = detector.detect(frame)

    for (x, y, w, h) in faces_coord:
        cv2.rectangle(frame, (x,y), (x+w, y+h), (150, 150, 0), 2)
        cv2.putText(frame, "Viola-Jones Detector",
            (x, y - 10),
            cv2.FONT_HERSHEY_PLAIN, 2, (150, 150, 0), 2)


    cv2.imshow("Viola-Jones Face Detection", frame)
    if cv2.waitKey(40) & 0xFF == 27:
        cv2.destroyAllWindows()
        break
```

Group Project

DSP Group Project

- The goal of this group project is to encourage students to explore DSP techniques to understand advanced applications in daily life.
- Students need to form a project team with 3 members to complete a DSP-related project.
- The project can analyze/evaluate/compare existing DSP technologies through literature surveys, and then implement related DSP algorithms using **Python**.
- Full project instructions can be found in the course website
  - http://www.ee.cityu.edu.hk/~lmpo/ee4015/projects.html

~ Learning is a game and group project is a good learning game.

Group Project Assessment (1)

- **Project Proposal** (Week 5)
  - A 2-page project proposal with motivation, DSP topic and implementation plan.
  - Submit the project proposal in PDF format to CANVAS proposal assignment.
  - Deadline: 28th Sep. 2022
- **Oral Presentation** (Week 12 and 13)
  - Every group is also required to make a 10-minute Power Point presentation of their term project to the entire class. The presentation must include:
    - A short description of the project and its objectives
    - An explanation of the implemented algorithm and relevant theory
    - A demonstration of the working program – i.e., results obtained when running the program

Group Project Assessment (2)

- **Final Report and Source Code** (Week 14)
  - A 30-page final report
  - Suggested structure of the final report:
    - Abstract, Introduction, Literature Survey, Theory, Implementation, Experimental Results, Conclusion, and References.
  - Students are also required to submit the Python source code of any implementation and PPT of the oral presentation for assessment.
  - All the PPT, Final Report and Source Code are required to submit to CANVAS Group Project Final Report.
  - Deadline: 29th Nov 2022

Suggestions for Group Projects

- *Your passion is your compass.*
- Something you are interested in or that may help your work or research
- Don't need to develop something original, but should
  - Read and describe relevant papers in literature
  - Do some applications of relevant techniques
- Could investigate something we have covered briefly, or that is related to some topic in the course.
- No more than two groups perform the same or very similar projects
  - Identical or similar projects may be rejected due to late submission of project proposal
  - Students are encouraged to submit the project title and short description to the instructor via email (eelmpo@cityu.edu.hk) for approval of the group project title.

Three Kinds of Group Projects

- **Application project**. Pick an application that interests you and explore how best to apply DSP algorithms to solve it.
- **Algorithmic project**. Pick a problem or family of problems, and develop a new DSP algorithm, or a novel variant of an existing algorithm, to solve it.
- **Theoretical project**. Prove some interesting/non-trivial properties of a new or an existing DSP algorithm. (This is often quite difficult, and so very few, if any, projects will be purely theoretical.)

Suggested Topics

- Audio Source Separation
- Audio Classification using Deep Learning
- Generating Songs With Neural Networks (Neural Composer)
- Wavelet Transformation for 1-D Signal Analysis Using Python
- Text to Speech in Python
- Urban Sound Classification with Librosa
- Speech Recognition Using Python
- Noise Reduction for Audio Processing
- Understanding Bluetooth codecs
- High-resolution audio
- Noise-cancelling headphones
- 3D Audio Simulation
- Sound Classification using Deep Learning
- Audio Genre Classification with Python OOP
- Two Minute Papers in Youtube
  - This AI Produces Binaural (2.5D) Audio
  - This AI Learned To Isolate Speech Signals
  - WaveNet by Google DeepMind

Mel Spectrogram Applications

- Audio Classification : Urban sound classification
- Automatic mood recognition
- Music genre classification
- Music instrument classification
- Brid sound classification
- Electrocardiogram (ECG) signal classification
- Heart sound (Phonocardiogram PCG) classification

15-Minute Break

- During the break, students are highly recommended to start forming the Term project groups.
- More information about the Group projects will be provided and discussed in the coming few weeks.