# Conversational UIs

## Spoken Language Processing

Dr. Dong Minghui
Institute for Infocomm Research
Agency for Science, Technology and Research (A-Star)
Email: mhdong@i2r.a-star.edu.sg

# Topics in the Course

- T1. Speech Processing Basics
- T2. Speech Synthesis
- T3. Speech Recognition
- T4. Speaker Recognition
- T5. Spoken Dialogue Processing

# Conversational UIs

## Spoken Language Processing

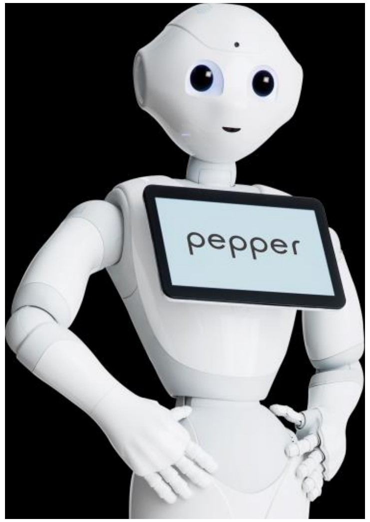
## Topic 1: Speech Processing Basics

Dr. Dong Minghui
Institute for Infocomm Research
Agency for Science, Technology and Research (A-Star)
Email: mhdong@i2r.a-star.edu.sg

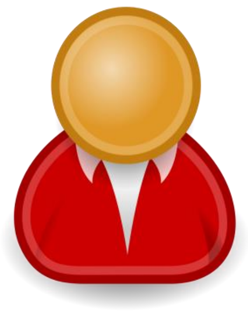
# PART 1. SPEECH SIGNALS

# Speech Applications

- **Examples**
  - Voice assistant
  - Robot
  - Navigation
  - Smart speakers

# Process of Spoken Dialogue



Speech Recognition

Speech Synthesis

Natural
language
Understanding

- **Automatic Speech Recognition**

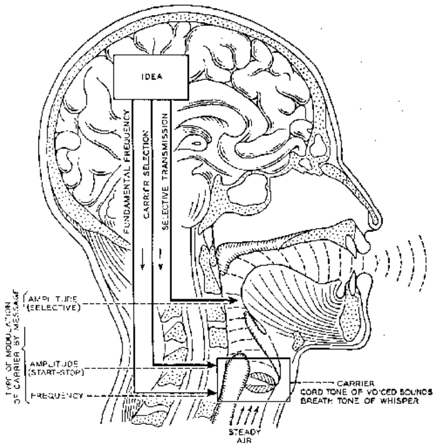
Speech (continuous time series) -> Text (discrete symbol sequence)

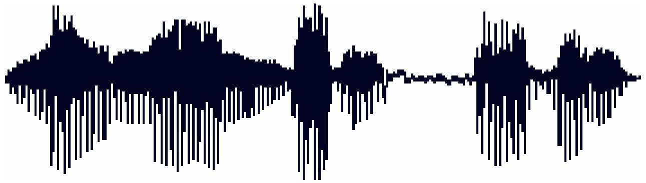
- **Speech Synthesis (Text-to-Speech)**

Text (discrete symbol sequence) -> Speech (continuous time series)

# Speech Signal

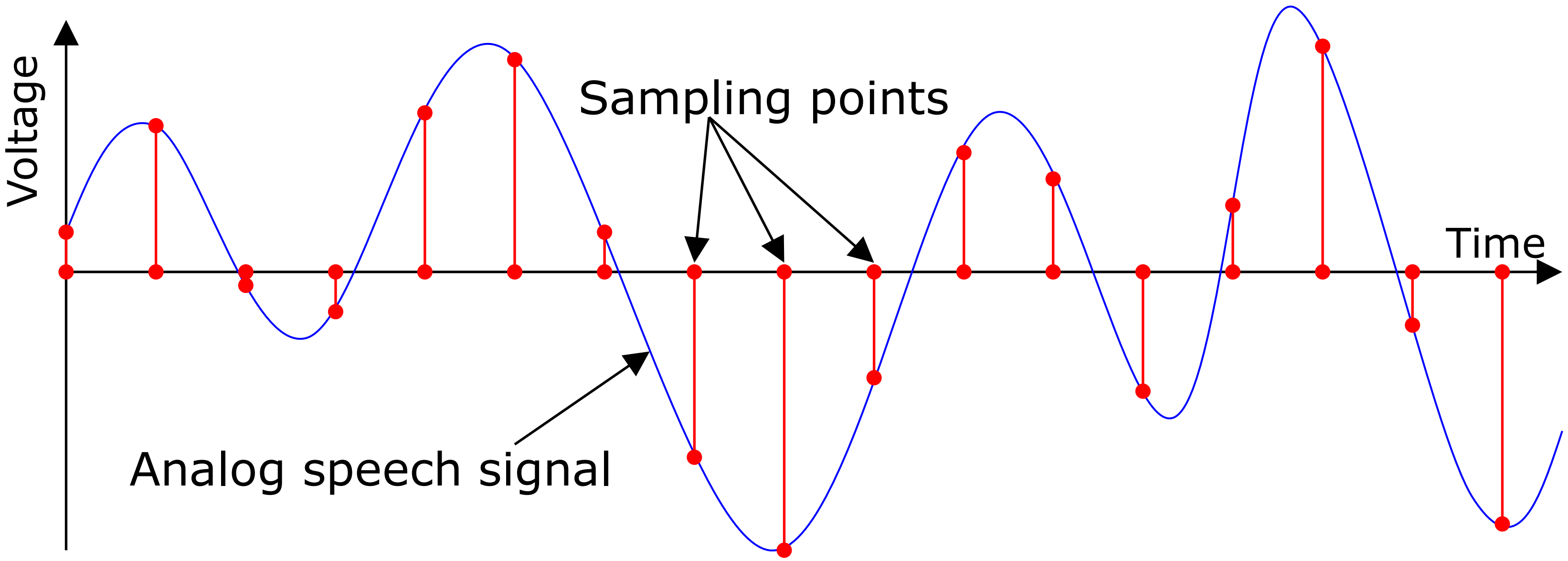
- Human voice is captured by microphone
- Speech is recorded in computer as a sequence of numbers



# Sampling Rate

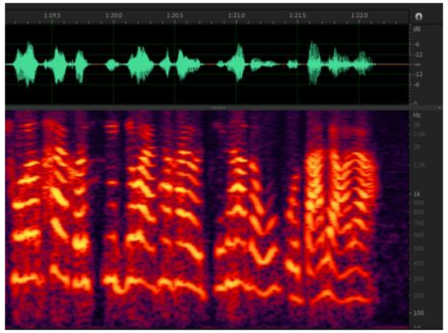
- **The analog speech signal captures pressure variations in air that are produced by the speaker.**
- **The analog speech input signal from the microphone is *sampled* periodically at some fixed *sampling rate***



# Sampling Rate

- Analog speech signal contains many frequencies
- Human ear can perceive frequencies in the range 50Hz-15kHz.
- Ideally, a sampling rate of 30kHz or more is needed to capture all the details of speech (The Nyquist theorem)
- CD recordings: 44.1kHz
- For practical reasons, 8kHz (telephone) and 16kHz (PC and smartphone) are often used.

# Speech Coding



- **Precision:**
  - Continuous signal level will be quantized to discrete values.
  - Each sample is represented with a fixed-point number in computer. (eg. 16bits, -32767 to 32767)
- **Coding:**
  - Linear coding method Pulse-Code Modulation (PCM) is commonly used.
  - Non-linear coding: *A-law* and *$\mu$-law* encoding schemes use only 256 levels (8-bit encodings)
  - 16-bit PCM is mostly used in speech processing.

# Audio File Format

- **Wav**
  - Developed by Microsoft and IBM
  - Native format: PCM, Uncompressed lossless
- **MP3**
  - A lossy data compression format.
- **In speech processing:**
  - Normally non-compressed PCM format is used for speech input.
  - Speech recognition models are often built for different sampling rates.

# Recording Quality

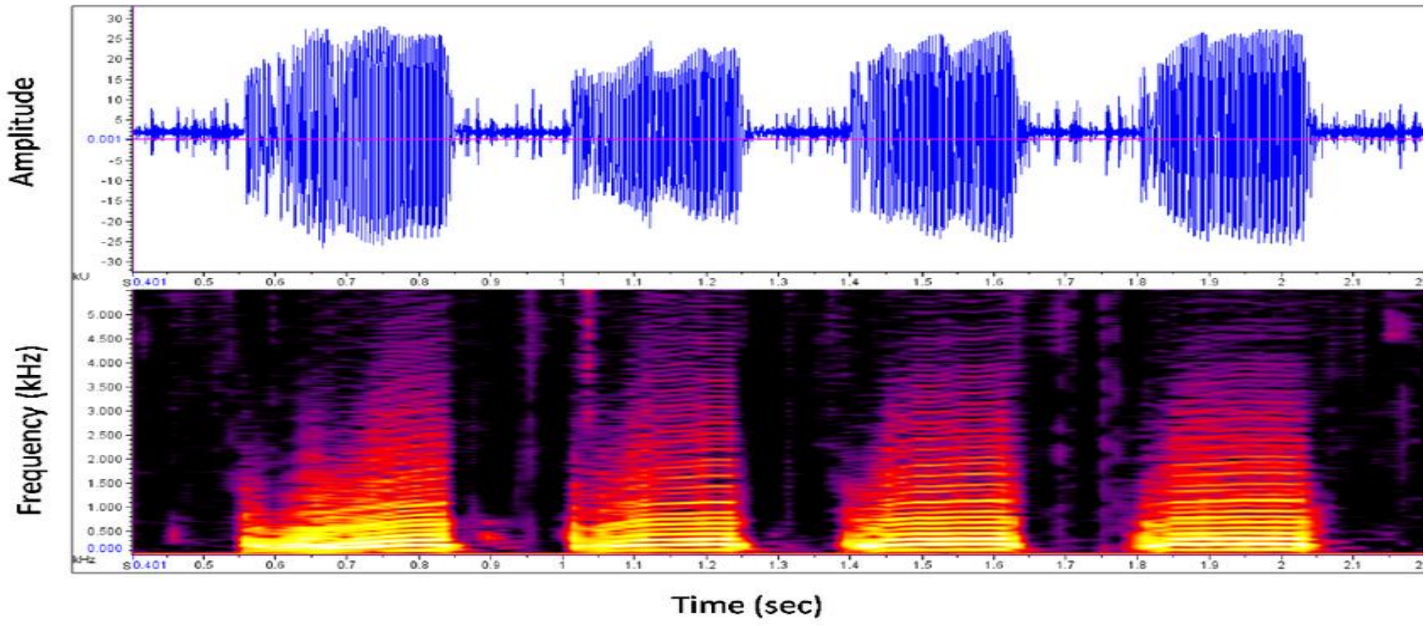
- **The microphone quality**
- **Environmental quality: ambient noise level**
- **Proper setting of the recording level**
    - Too low: losing resolution
    - To high: clipping (signal value exceeds maximum)



# PART 2. SPEECH FEATURES

# Spectrogram

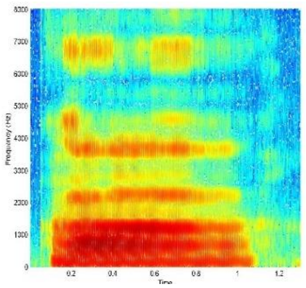
- Speech is a one-dimensional signal.
- To effectively analyse  the signal, it is often converted into a two-dimensional image called spectrogram.
- Spectrogram shows the strength of different frequencies of the signal at any time.



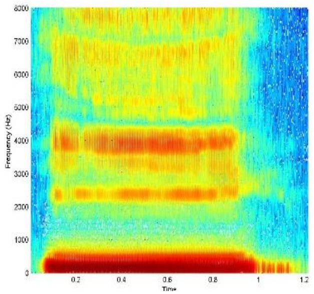
# Spectral Differences for Pronunciations

- **Different sounds show different energy levels at different frequencies.**

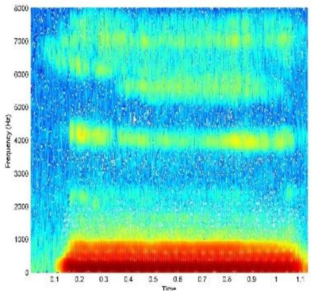
AA



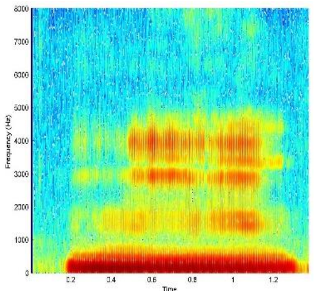
IY



UW

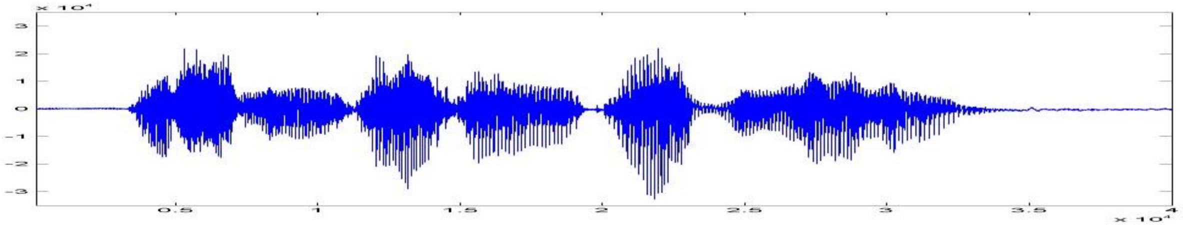


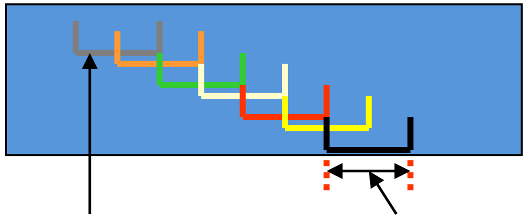
M



# Frame

- Speech signal is normally processed by segments. Each segment is called a frame.
- Frame Size: size of the speech segment
- Frame Shift: number of samples shifted to the next frame
- Frame can be overlapped with each other.



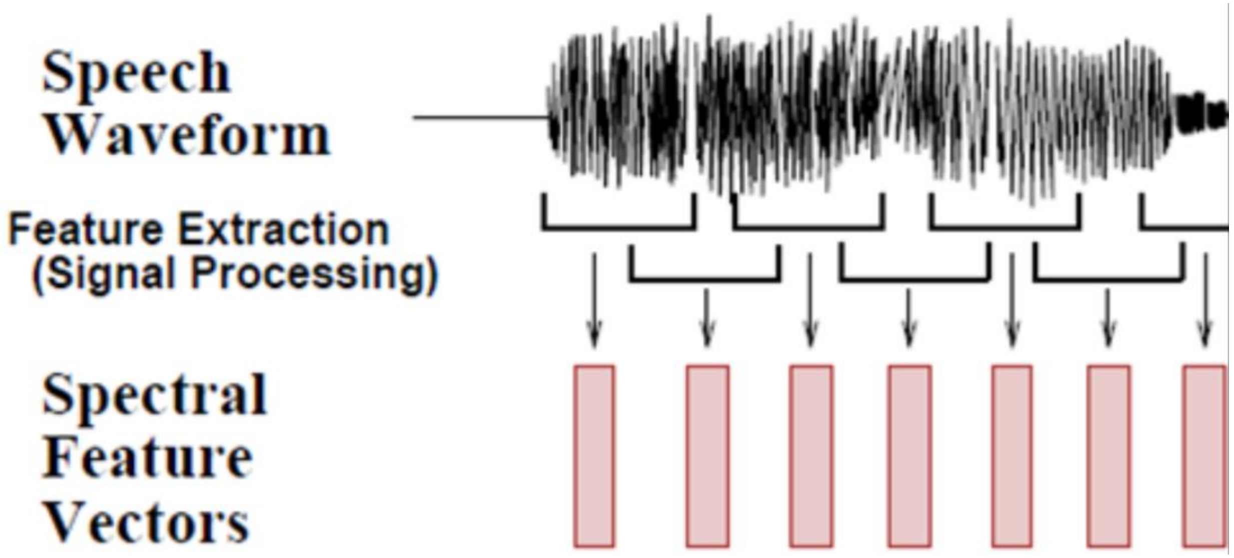


Segments shift every 10 milliseconds

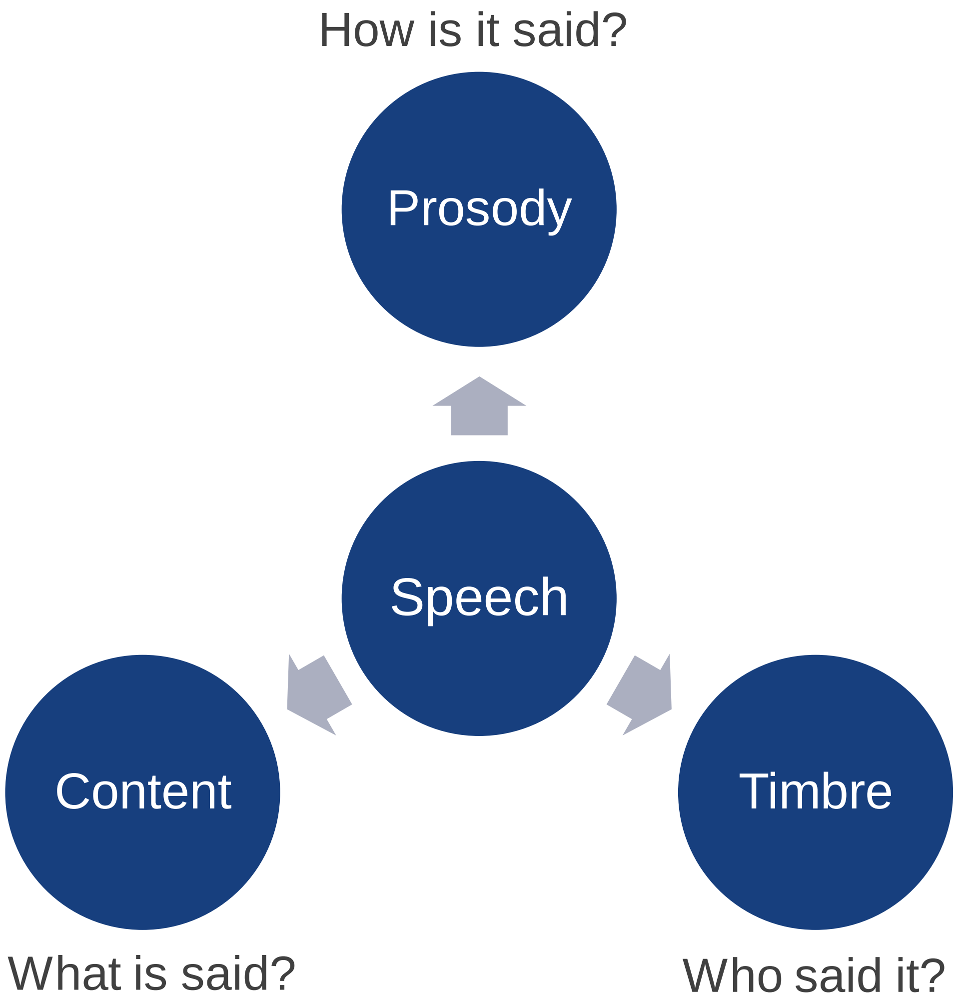
Each segment is typically 20 or 25 milliseconds wide

# Speech Feature

- Speech signal is analysed frame by frame.
- Each frame can be converted into vector.
- So, a speech signal can be represented with a sequence of feature vectors.



# Information in Speech



# Content and Timbre

- **Content:**
  - Text transcription of speech signal
  - Speech recognition is to derive the content from speech signal
  - Speech synthesis is to implement text information with speech signal.
- **Timbre**
  - Timbre represents the speaker information of the speech.
  - Different speaker has different voice timbre.

# Prosody

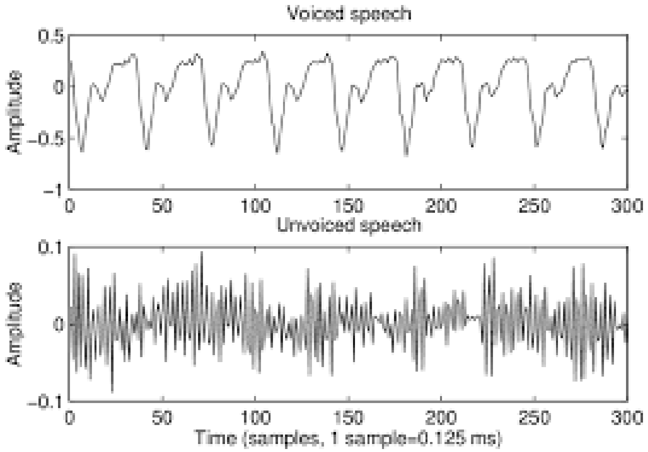
- **What is prosody (from perception level)**
  - The same text can be read in different ways. The way to read the text is determined by prosody.
  - Example: I bought two books from the shop.
  - Prosody is perceived as intonation, rhythm, pause, emotion, speaking styles, speech rate, etc.
- **Major elements (from acoustic level)**
  - Fundamental frequency (pitch of voiced signals)
  - Duration (length of each phonetic unit)
  - Energy (loudness of each phonetic unit)

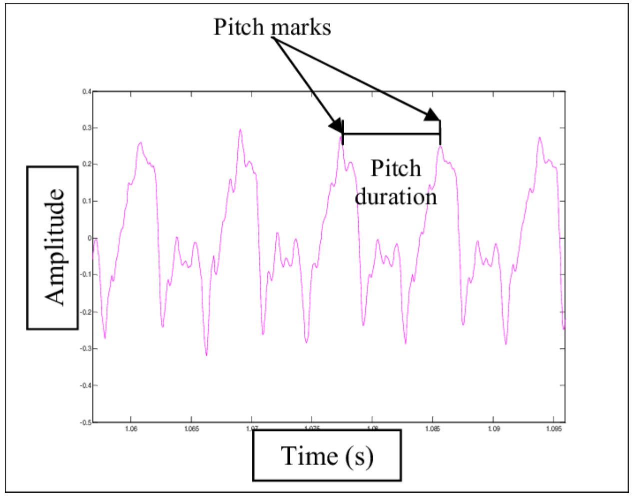
# Examples of Perceived Prosody

- **Intonation**
  - Statements normally have a falling intonation. Questions may have rising intonation.
- **Lexical tone**
  - Mandarin has four lexical tones.
- **Emphasis**
  - Some words are emphasized in speech
- **Emotion**
  - Angry and happy speeches have different prosody.
- **Phrase break**
  - Phrase break within sentence is also part of prosody.

# Fundamental Frequency (pitch)

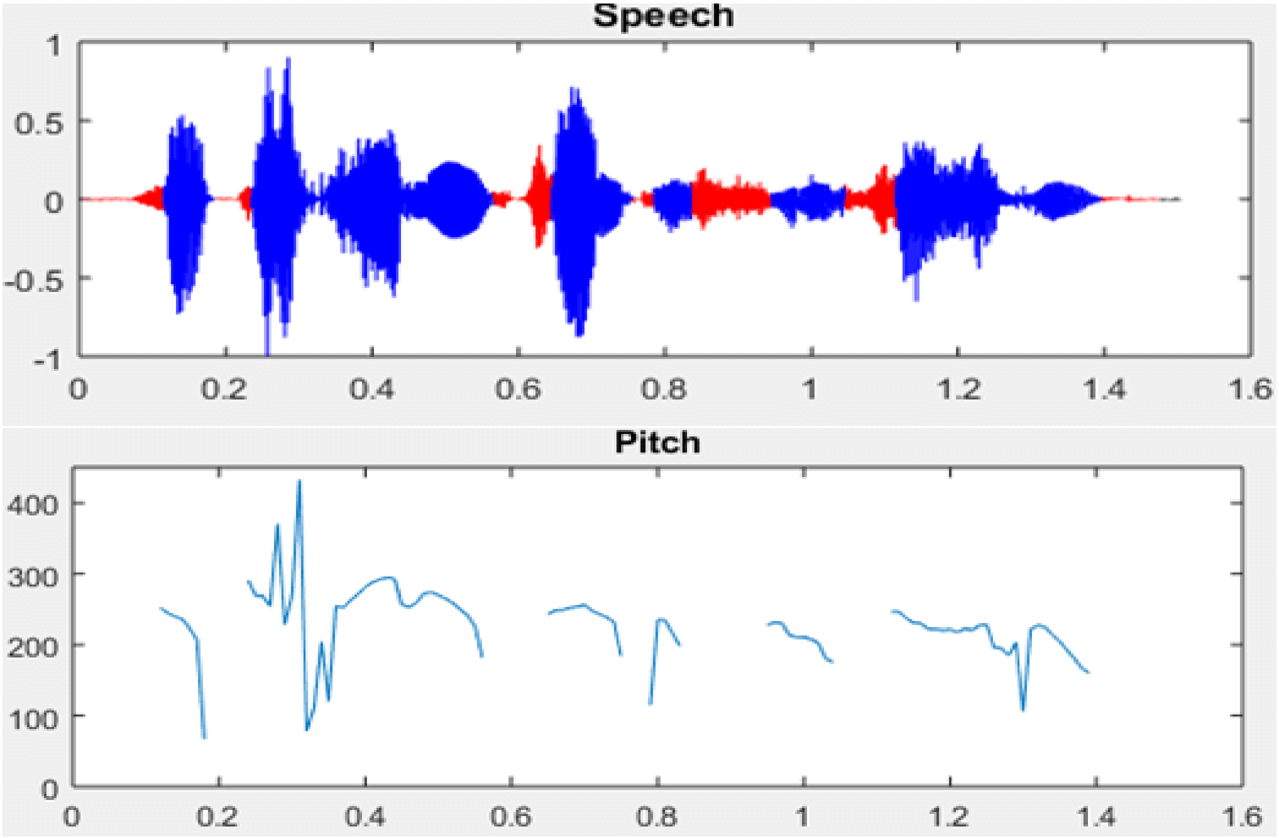
- **Voiced and Unvoiced**
  - Voiced speech signals are produced when the vocal cords vibrate.  The rest are unvoiced signals.
  - Voiced signals are periodic ones. Unvoiced signals are noise.
  - All vowels are voiced.
  - Eg. /s/, /f/ are unvoiced; /z/, /v/ are voiced
- **Pitch**
  - Pitch exists in voice signals only.
  - Fundamental Frequency (F0)
  - Pitch duration: duration between two pitch marks.
  - Frequency = 1 / period



# Fundamental Frequency (pitch)

- Fundamental frequency is referred to as Pitch or F0.
- Pitch is especially important in speech perception or generation.



# Signal Preprocessing

- **Voice Activity Detection (VAD)**
  - Microphone may be on all the time. But computer only start processing when voice is detected.
  - VAD is to find the valid speech segments to process.
- **Speech Enhancement**
  - In many cases, speech signal needs to be enhanced, and noise needs to be reduced.
  - Enhancement is to improve the speech quality with signal processing methods.
- **Speech Normalization**
  - To convert the speech signal to keep consistency.
  - Time domain normalization.
  - Frequency domain normalization

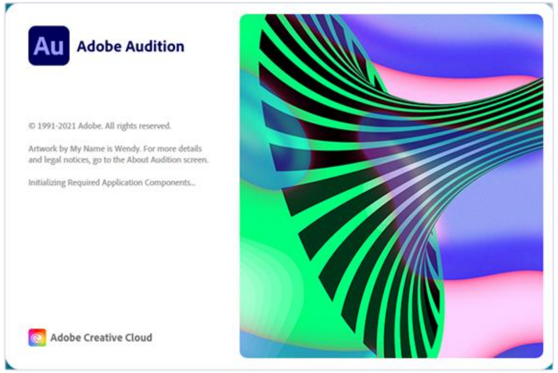
# Speech Processing Topics in this Course

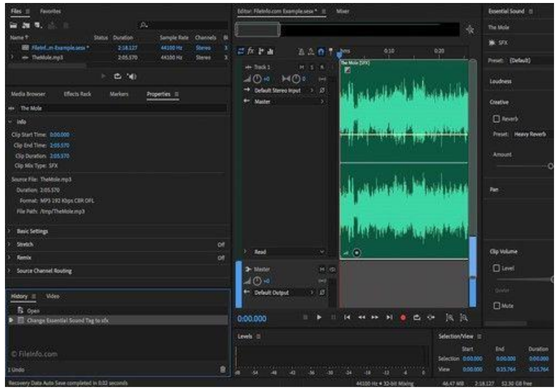
- Speech-to-Text (Speech recognition)
- Text-to-Speech (Speech synthesis)
- Voice Print (Speaker recognition)
- Chatbot (Spoken dialogue System)

# PART 3. AUDIO SOFTWARE

# Audio software

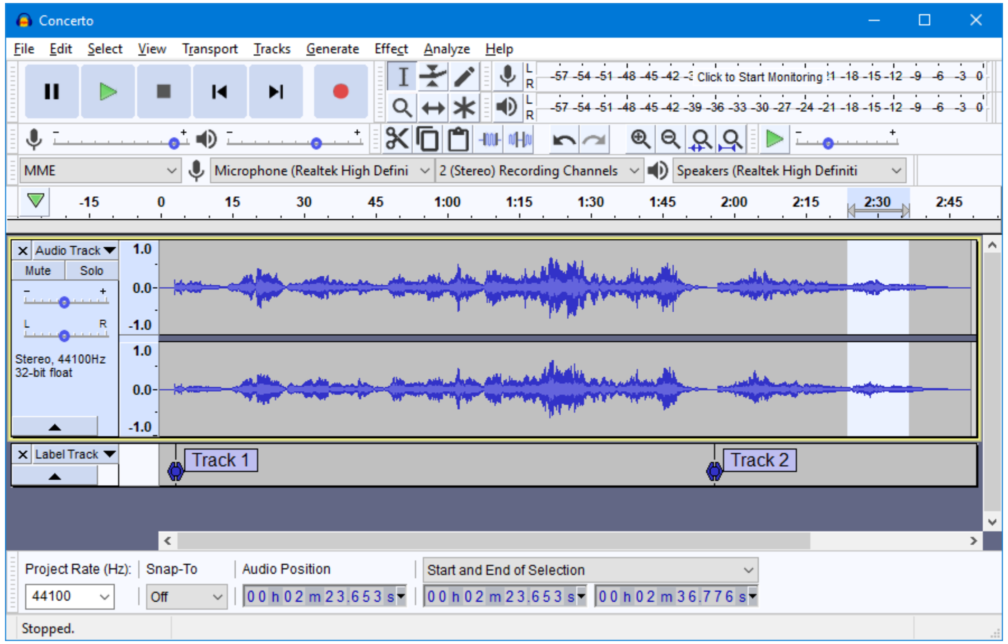
- **Adobe Audition**
  - Professional Audio Editor.
  - Record, view and play audio signals.
  - Cut, copy, paste, mix
  - Normalize, filter,  remove noise
  - Show spectrogram
  - Change sampling rate, convert file format



# Audio software

- **Audacity**
  - Audacity: https://www.audacityteam.org/
  - A free and open-source audio editor and recording software.
  - Available for Windows, macOS, Linux.



# Audacity

- **Record and save file**
  - Select sampling rate, file format, resolution
- **View and edit**
  - Zoom in, Zoom out
  - Cut, copy, past, etc
  - Generate silence, noises
- **Mix and effect**
  - Stereo to mono
  - Normalize,  noise reduction,
  - Change pitch, speed, etc

# Audio Software

- **Sox (Sound of Exchange)**
  - Sox: https://sourceforge.net/projects/sox/
  - An open-source cross-platform audio editing software
  - Command line tool.
  - Converting sampling rate, bits, stereo/mono, etc
  - Editing: Concatenate, trim, pad, repeat, reverse, volume, fade, normalise
  - Effects: chorus, flanger, echo, phaser, compressor, delay, filter
  - Adjustment of speed, pitch, tempo, etc

# Sox

- **Examples:**
  - **sox --i test1.wav**

    show information of the file
  - **sox test1.wav -r 8k test1-out-8k.wav**

    change sampling rate

    -r 16k = sampling rate:16khz,
  - **sox test2.wav -c 1 test2-out.wav**

    convert to mono wave file

    -c 1 = single channel (mono)

# Sox



- **Examples:**
  - **sox -r 8k -b 8 -c 1 -e signed test3.raw test3-out.wav**

    raw format → wav format

    -e signed = signed integer
  - **sox test3.wav test2.wav longfile.wav**

    Concatenate two files into one long file.
  - **sox test2.wav test2-fast.wav speed 1.1**

    Adjust speed of the speech

# PART 4. AUDIO PROGRAMMING

# Python tool and audio libraries

- Anaconda
  - A distribution of Python programming language.
- SoundFile, Wave, scipy.io.wavfile, audioread
  - Libraries for reading and writing audio files
- PyAudio, SoundDevice
  - Libraries for playing and recording speech files
- LibROSA
  - Library for music and audio analysis
  - Feature extraction, spectrogram display
  - Voice effects

# Anaconda - Python

- https://www.anaconda.com/
- A python distribution for scientific computing.
- Supports Windows, Linux, MacOS
- Contains basic packages and programming tools.
- Spyder: An interactive development environment (IDE) tool
- Jupyter Notebook: A Web-based IDE tool

# SoundFile – Python audio library

- **Download and Install:**
  - https://pypi.org/project/SoundFile/
  - https://pysoundfile.readthedocs.io/en/latest/
  - pip install soundfile
  - sudo apt-get install libsndfile1
- **Features:**
  - Support WAV, FLAC, OGG, MAT files.
- **Program in Python**
  - import soundfile as sf
  - data, samplerate = sf.read('existing_file.wav')
  - sf.write('new_file.flac', data, samplerate)

# PyAudio – Audio playing and recording

- **Install**
  - https://people.csail.mit.edu/hubert/pyaudio/
  - python -m pip install pyaudio (windows)
  - sudo apt-get install python-pyaudio python3-pyaudio (linux)
- **Programming**
  - import pyaudio
  - p = pyaudio.PyAudio()
  - stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
  - channels=wf.getnchannels(), rate=wf.getframerate(), output=True)
  - data = wf.readframes(1024)

# LibROSA – Feature calculation and display

- **Installation**
  - https://librosa.org/
  - pip install librosa
- **Programming**
  - D = librosa.amplitude_to_db(np.abs(librosa.stft(y)), ref=np.max)
  - plt.figure()
  - librosa.display.specshow(D, y_axis='linear')
  - plt.colorbar(format='%+2.0f dB')
  - plt.title('Linear-frequency power spectrogram')

# Thank you!

mhdong@i2r.a-star.edu.sg