Speech (Signal) Processing (Part II)

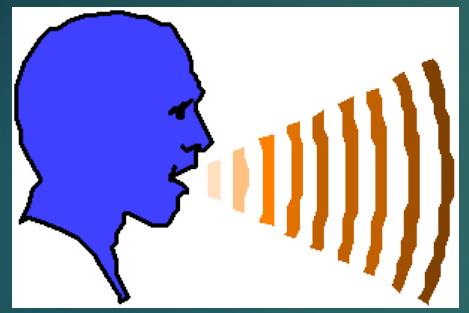
CSCI-B 659

TUESDAY, SEPTEMBER 3RD, 2019

LECTURE #3

Homework #1 Recap

Capturing Sound



Speaker

Microphone

Interface

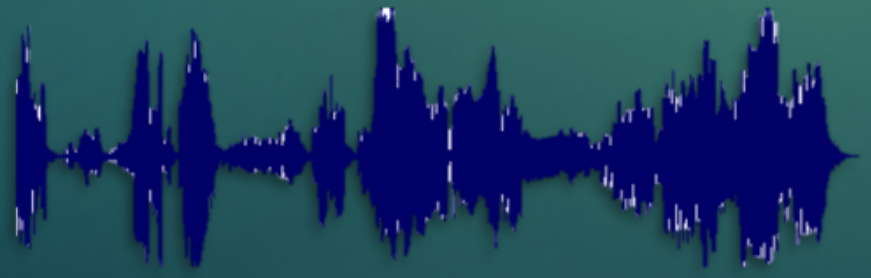
Digital Waveform

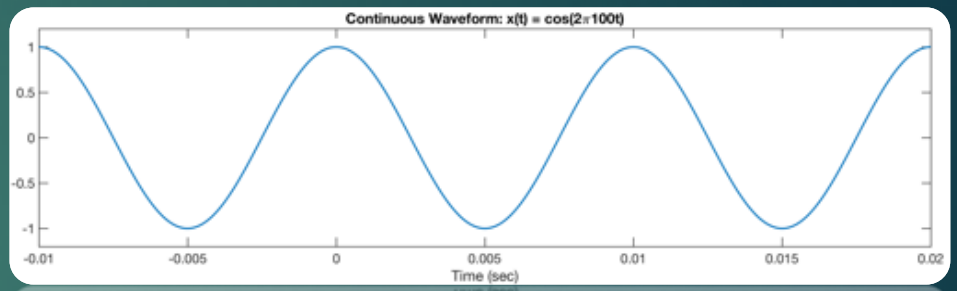
Computers have built-in converters

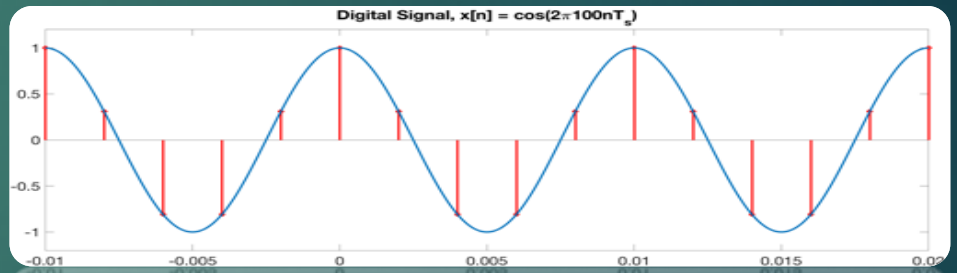
Digital Waveform: Time Domain

- x[n] and x(t) are time-domain signals
  - They are functions of time
  - *t* is continuous time
  - *n* is discrete (digital) time
- Characteristics:
  - 1-D data vector
  - Real valued

Ex. Speech signal







Learning Objectives

- **You the student will be able to:**
  - Compute power, intensity and energy of a signal
  - Convert between time and frequency domain signals
  - Understand why time-frequency representations are needed

Volume vs. Loudness

- Volume can be used to change the "loudness" of a signal

- Loudness is perceptual, not physical
  - You cannot "measure" loudness
  - Sound example
- Which sound is louder?

Measuring Signals

- Power (instantaneous) of a given signal x*[n]* at time *n*:

$$P[n] = (x[n])^2$$

- Energy is the total power over a certain time period

$$E = \sum_{n=n_1}^{n=n_2} (x[n])^2 = \sum_{n=n_1}^{n=n_2} P[n]$$

- Intensity (instantaneous)

$$I[n] = \frac{(x[n])^2}{\rho c}$$

ρ : density of the medium

c : speed of sound

Energy of the entire signal is computed over all samples

Sound Level

$$I[n] = \frac{(x[n])^2}{\rho c}$$

- Ratio of one sound to another (baseline), expressed in decibels (dB)

$$L_2 - L_1(decibels) = 10log_{10}\frac{I_2}{I_1} = 10log_{10}\frac{P_2}{P_1}$$

- Ex. 1, $I_2 = 2I_1$
  - Then the level difference is 3 dB
  - Sound 2 is 3 dB higher
- Ex. 2, $x_2 = 2x_1$
  - Then, $I_2 = 4I_1$
  - Then the level difference is 6 dB
  - Sound 2 is 6 dB higher

Common Sound Levels

$$L_2 - L_1(decibels) = \quad 10log_{10}\frac{P_2}{P_1}$$

- In air, P_1 = 1 pW {pico (10^{-12}) Watt}
  - Conversational speech is about 65 dB
  - Above 100 dB is damaging to the ear

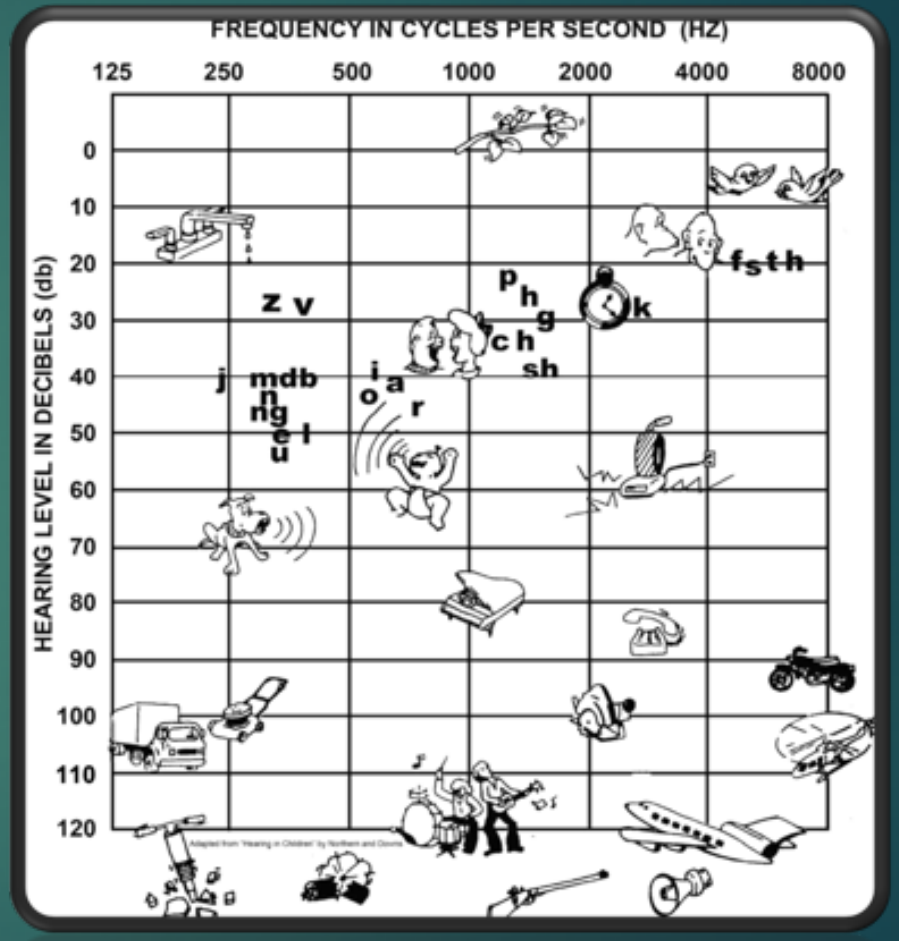


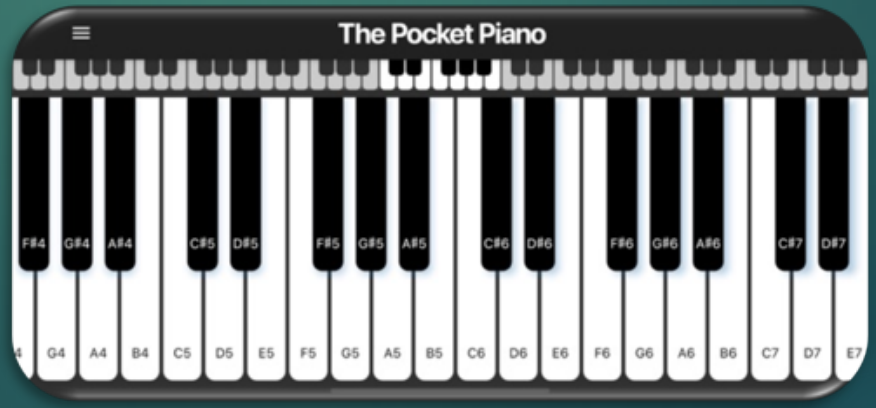
Image courtesy of http://www.handsandvoices.org/resource_guide/055_audiogram.html

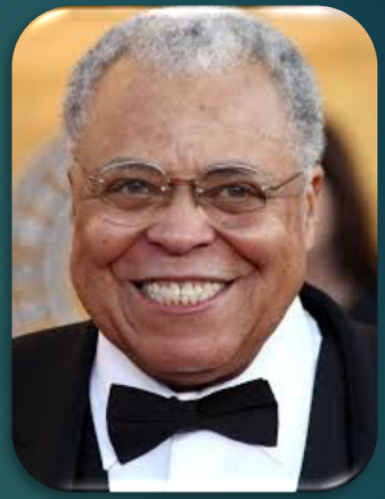
Limitations of Time Domain

- Most speech and sound processing is ***NOT*** done in the time domain. Why?
  - Visually, time domain does not convey distinguishing information (i.e. what is said)
  - Time domain processing can be more computationally expensive
- Most processing is done in the frequency domain

Frequency

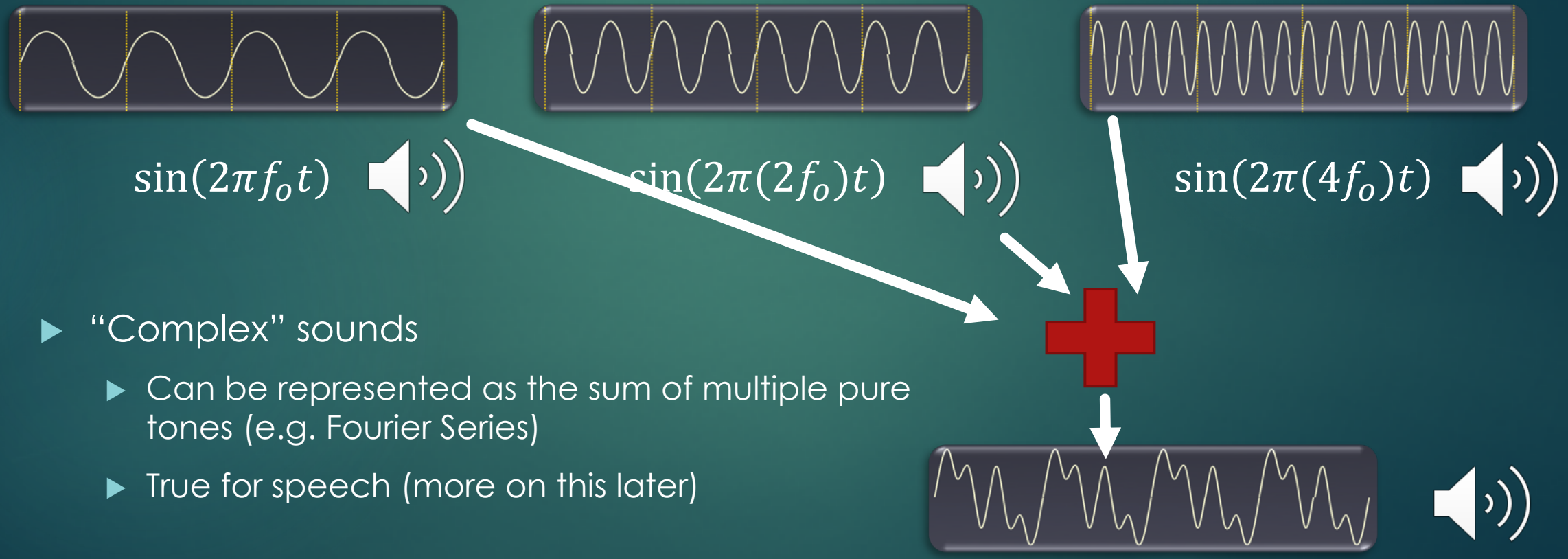
- Frequencies are discussed in multiple ways





Simple vs. Complex Sounds

- Pure tones (e.g. f_o = 250 Hz)



- "Complex" sounds
  - Can be represented as the sum of multiple pure tones (e.g. Fourier Series)
  - True for speech (more on this later)

Analyzing Signal Frequencies

- A frequency-domain signal can be computed directly from a time domain signal using the ***Discrete Fourier transform (DFT)***

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j(2\pi kn)}{N}}$$

  - x[n]: time domain signal
    - *n:* time sample index (integer)
    - *N: signal length (e.g. n is between 1 and N)*
  - X[k]: frequency domain signal
    - *k*: frequency sample index (integer)
    - *K: number of frequency samples (e.g. k is between 1 and K)*

DFT Properties

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j(2\pi kn)}{N}}$$

- $X[k]$ is complexed valued (i.e. Euler's formula $e^{jx} = cos(x) + jsin(x)$)
  - It has real and imaginary components (rectangular): $X[k] \stackrel{\text{def}}{=} X_R + jX_I$
  - It is often represented using polar coordinates: $X[k] \stackrel{\text{def}}{=} |X[k]|e^{j\angle X[k]}$

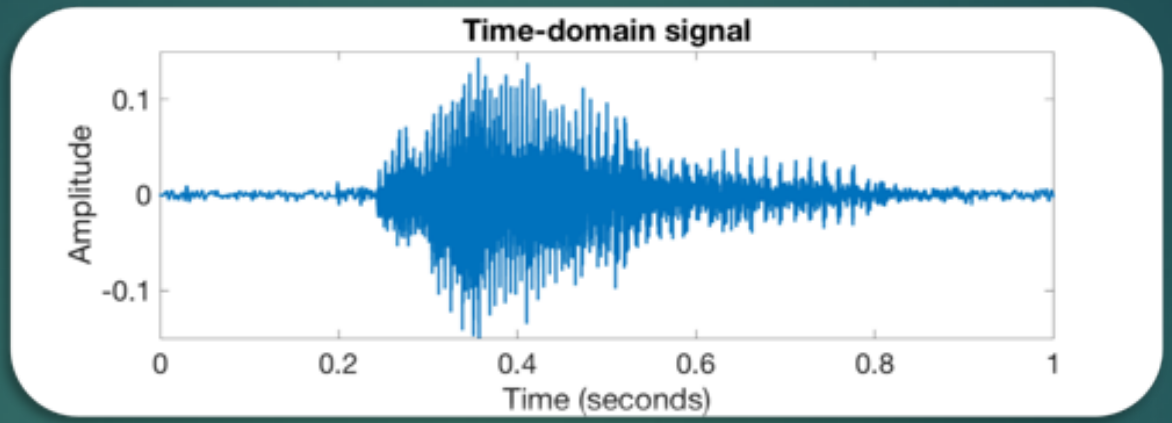
Magnitude and Phase Responses

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j(2\pi kn)}{N}} \qquad X[k] \stackrel{\text{def}}{=} |X[k]| e^{j\angle X[k]}$$

- Magnitude response
  - $|X[k]| = \sqrt[2]{X_R^2 + X_I^2}$ (numpy.*abs()* in Python)
  - Determines the relative presence of a sinusoid, $e^{\frac{-j(2\pi kn)}{N}}$, in $x[n]$
- Phase response
  - $\angle X[k] = \tan^{-1} \frac{X_I}{X_R}$ (numpy.*angle()* in Python)
  - Determines how the sinusoids line up relative to one another to form x[n]

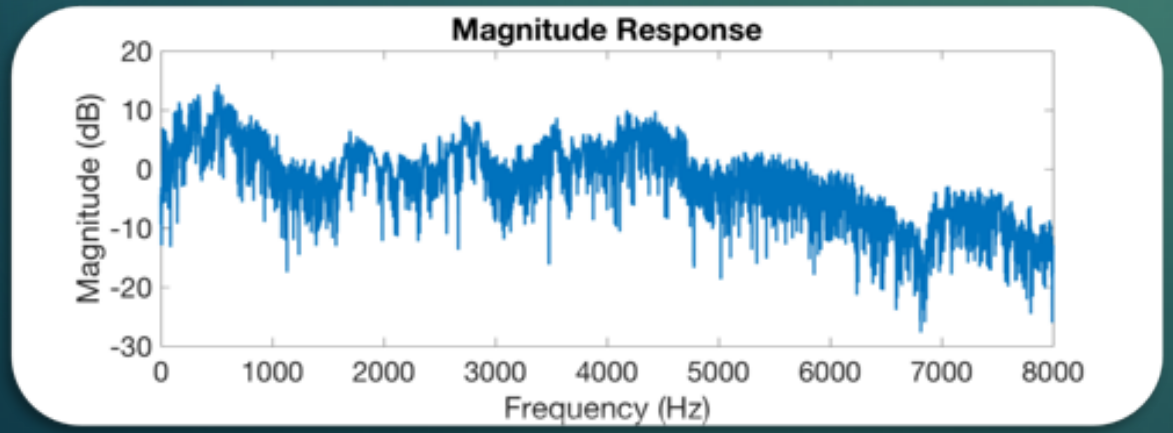
Magnitude and Phase Spectra

$x[n]$

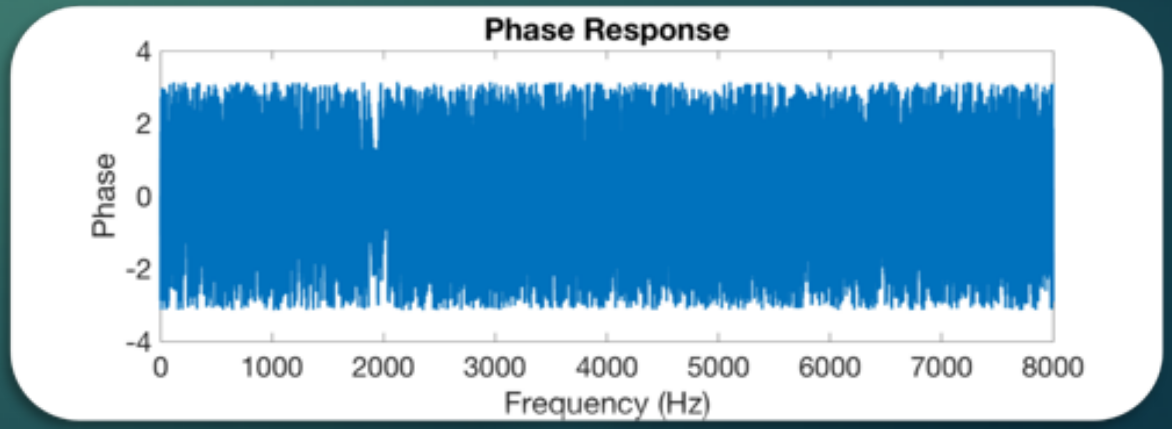


$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j(2\pi kn)}{N}}$$

$20 * \log10\ |X[k]|$



$\angle X[k]$



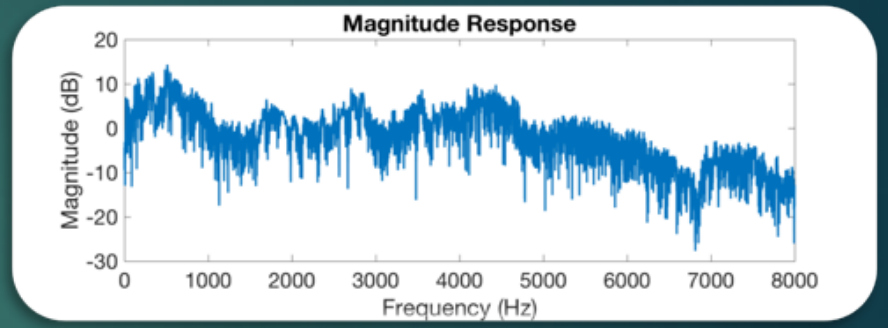
The Fourier Transform: Synthesis

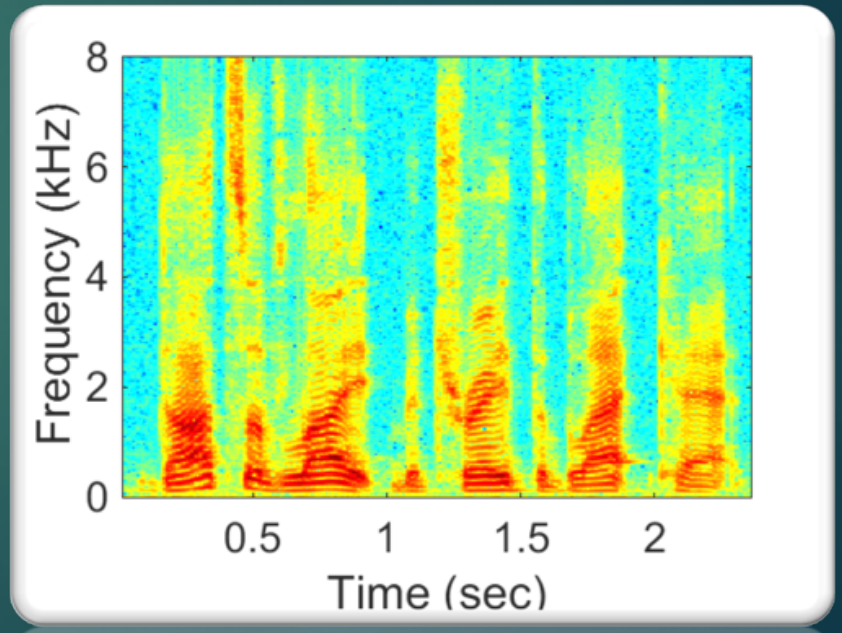
- The inverse discrete Fourier transform (IDFT) produces a time-domain signal, *x[n],* from its frequency response, *X[k]*

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi k n}{N}}$$

Frequency Domain

- Frequency domain on its own may not convey enough information. Why?
  - Frequency information for the entire signal is shown at once
  - Time is still important, but it is lost due to Fourier Analysis
- Time-Frequency (T-F) domain processing is much more widely used





T-F Domain: Fourier Transform View

- How can you capture information across time and frequency?
  - This is commonly done with the ***Short-time Fourier Transform (STFT)***

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{\frac{-j2\pi km}{N}}$$

- *X[n,k]* is the STFT of x[n]
  - *n*: time sample index
  - *k*: frequency sample index
  - x: time-domain signal
  - *N*: DFT length
  - w: analysis window (or filter) function
    - Mostly zero, except for a small region

Time-Frequency Domain

- How can you capture information across time and frequency?
  - This is commonly done with the ***Short-time Fourier Transform (STFT)***

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{\frac{-j2\pi km}{N}}$$

- (Theoretical) Steps:
  1. Define a windowing function w[m]
  2. Flip w[m] along the time axis, w[-m]
  3. Shift w[-m] by n samples, w[n-m]
  4. Multiply w[n-m] and x[m], x[m]w[n-m]
  5. Compute the DFT of x[m]w[n-m]
  6. Repeat 1-5 for all values of n

Summary

- Sound can be analyzed in terms of its power, intensity, energy, and relative level
- Time and Frequency domain are used to represent speech
- Frequency-domain is visually more informative than time (we'll see more of this later)
- Fourier analysis and synthesis are used to convert between time and frequency domains

Next class: Signal Processing (Part III)

- Main Topics:
  - Time-Frequency Domain