6.003 Signal Processing

Week 9, Lecture A:
Short Time Fourier Transform
Time-varying Signals

Real-world signals (i.e., speech, music, ...) often have frequency content that varies with time.

Fourier Transform: events that are local in time are global in frequency (and vice versa). Sudden changes and local variations can be difficult to detect.

Example: 2 tunes
- `cos0.wav`
- `cos1.wav`

How to tell them apart?

FFT of the two signals:
Short-Time Fourier Transform (STFT)

The short-time Fourier Transform is a tradeoff between time and frequency-domain representations, representing the frequency content of the signal at various points in time.

Conceptually, we are taking the DFT of successive windowed regions of the original signal (and these regions may overlap).
Short-Time Fourier Transform (STFT)

Formally, we define the STFT of a signal \( x \) as:

\[
STFT\{x\}[m,k] = \sum_{n=0}^{N-1} x[n + m \times s] \cdot w[n] e^{-j \frac{2\pi k}{N} n}
\]

Where:
- \( m \) is a time index, \( k \) is a frequency index
- \( N \) is the length of a window
- \( s \) is “step size”
- \( w[n] \) is window function
STFT and Spectrograms

\[ STFT\{x\}[m, k] = \sum_{n=0}^{N-1} x[n + m \times s] \cdot w[n] e^{-j\frac{2\pi k}{N} n} \]

The STFT is often visualized using a spectrogram, which is defined to be the magnitude squared of the STFT.

Spectrogram:

Frequency content of the signal at various points in time.
Setting Suitable Parameters for Spectrograms

\[ STFT\{x\}[m, k] = \sum_{n=0}^{N-1} x[n + m \times s] \cdot w[n]e^{-j\frac{2\pi k}{N}n} \]

- What window size \( N \) should be used?
- What step size \( s \) should be used?
- What type of window function should be used?

**Setting suitable window size \( N \):**
- Larger \( N \) will give rise to better frequency resolution
- Larger \( N \) will compromise time resolution

**Setting suitable step size \( s \):**
- Smaller \( s \) helps with better time resolution
- Smaller \( s \) will mean more steps to perform calculation, increase computation cost
Spectrogram of signal from cos0.wav

- Window size large, we cannot detect frequency change in time.
- Need to optimize the window size to achieve reasonable resolution both in frequency and time.
- Frequency conversion: see slide #13 of Lec 05A.

Sampling rate 8000 samples/sec, 9600 samples

- window_size=4096, step_size=256
- window_size=2048, step_size=256
- window_size=512, step_size=256
- window_size=256, step_size=128
- window_size=128, step_size=64
Onset of a note tend to have frequencies across all spectrum
Reducing step size, better resolution in time, longer time to compute
Spectrogram of signal from mystery1.wav

- Larger window size $N$ to resolve frequency better
- If already know whole frequency range, can zoom into certain frequency range to see better
- Now we can read music using spectrogram!
Spectrogram of signal from mystery2.wav

displayed with

- window_size=512, step_size=128
- window_size=1024, step_size=64

sample_rate=22050, 202341 samples

- Larger window size N to resolve frequency better to see the chords more clearly
• Now we can use this to analyze all kinds of waveforms!
Other Spectrograms

- Spectrogram gives visual representation of the spectrum of frequencies contained in signal, as a function of time. A very useful tool in many fields!
STFT in Processing Streaming Signals

- Breaking a signal into pieces so that it can be processed in small chunks is especially important in **streaming** applications where the signal may be arbitrarily long.

General procedure:
- Divide the input signal into a sequence of windows, each of length N.
- Process each window.
- Assemble the processed pieces together to form the output.

- How to do this? How to choose N (good resolution with high speed)?
- How to assemble them smoothly?
Summary

- **Short-Time Fourier Transform**: analyzing a sequence of finite-length portions of an input signal

  ➢ **Spectrogram**: visual representation of the spectrum of frequencies of a signal as it varies with time

  ➢ **Processing Streaming signals**: (See Recitation 9B)