Topic 2

Signal Processing Review

(Some slides are adapted from Bryan Pardo’s course slides on Machine Perception of Music)
Recording Sound

Mechanical Vibration

Pressure Waves

Motion->Voltage Transducer

Voltage over time
Microphones

Cross-Section of Dynamic Microphone

http://www.mediacollege.com/audio/microphones/how-microphones-work.html
Pure Tone = Sine Wave

\[ x(t) = A \sin(2\pi ft + \varphi) \]

time | amplitude | frequency | initial phase

Period T

Amplitude

Time (ms)

440Hz

ECE 477 - Computer Audition, Zhiyao Duan 2017
Reminders

• Frequency, $f = 1/T$, is measured in cycles per second, a.k.a. Hertz (Hz).

• One cycle contains $2\pi$ radians.

• Angular frequency $\Omega$, is measured in radians per second and is related to frequency by $\Omega = 2\pi f$.

• So we can rewrite the sine wave as

$$x(t) = A \sin(\Omega t + \varphi)$$
Fourier Transform

\[ X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} \, dt \]
We can also write

$$X(\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt$$
Complex Tone = Sine Waves

\[
\begin{align*}
&220 \text{ Hz} \\
+ & 660 \text{ Hz} \\
+ & 1100 \text{ Hz} \\
= &
\end{align*}
\]
Frequency Domain

\[ X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \]
Harmonic Sound

• 1 or more sine waves
• Strong components at integer multiples of a fundamental frequency (F0) in the range of human hearing (20 Hz ~ 20,000 Hz)

• Examples
  – 220 + 660 + 1100 is harmonic
  – 220 + 375 + 770 is not harmonic
Noise

- Lots of sines at random freqs. = NOISE
- Example: 100 sines with random frequencies, such that $100 < f < 10000$. 

![Graph showing noise with random frequencies](image)
How strong is the signal?

- Instantaneous value?
- Average value?
- Something else?

\[ x(t) \]
Acoustical or Electrical

• Acoustical

Average intensity

\[ I = \frac{1}{\rho c} \frac{1}{T_D} \int_0^{T_D} x^2(t) \, dt \]

- View \( x(t) \) as sound pressure
- \( \rho \): density
- \( c \): sound speed

• Electrical

Average power

\[ P = \frac{1}{R} \frac{1}{T_D} \int_0^{T_D} x^2(t) \, dt \]

- View \( x(t) \) as electric voltage
- \( R \): resistance
Root-Mean-Square (RMS)

\[ x_{RMS} = \sqrt{\frac{1}{T_D} \int_0^{T_D} x^2(t) \, dt} \]

- \( T_D \) should be long enough.
- \( x(t) \) should have 0 mean, otherwise the DC component will be integrated.
- For sinusoids

\[ x_{RMS} = \sqrt{\frac{1}{T} \int_0^T A^2 \sin^2(2\pi ft) \, dt} = \sqrt{A^2/2} = 0.707A \]
Sound Pressure Level (SPL)

- Softest audible sound intensity
  0.000000000001 watt/m²
- Threshold of pain is around 1 watt/m²
- 12 orders of magnitude difference
- A log scale helps with this
- The decibel (dB) scale is a log scale, with respect to a reference value
The Decibel

• A logarithmic measurement that expresses the magnitude of a physical quantity (e.g. power or intensity) relative to a specified reference level.

• Since it expresses a ratio of two (same unit) quantities, it is dimensionless.

\[
L - L_{\text{ref}} = 10 \log_{10} \left( \frac{I}{I_{\text{ref}}} \right) \\
= 20 \log_{10} \left( \frac{x_{\text{RMS}}}{x_{\text{ref,RMS}}} \right)
\]
Lots of references!

- **dB-SPL** – A measure of sound pressure level. 0 dB-SPL is approximately the quietest sound a human can hear, roughly the sound of a mosquito flying 3 meters away.

- **dbFS** – relative to digital full-scale. 0 dbFS is the maximum allowable signal. Values typically negative.

- **dBV** – relative to 1 Volt RMS. $0 \text{dBV} = 1V$.

- **dBu** – relative to 0.775 Volts RMS with an unloaded, open circuit.

- **dBmV** – relative to 1 millivolt across 75 $\Omega$. Widely used in cable television networks.

- ......
Typical Values

- Jet engine at 3m: 140 dB-SPL
- Pain threshold: 130 dB-SPL
- Loud motorcycle, 5m: 110 dB-SPL
- Vacuum cleaner: 80 dB-SPL
- Quiet restaurant: 50 dB-SPL
- Rustling leaves: 20 dB-SPL
- Human breathing, 3m: 10 dB-SPL
- Hearing threshold: 0 dB-SPL
Digital Sampling

AMPLITUDE

sample interval

TIME

quantization increment

RECONSTRUCTION

ECE 477 - Computer Audition, Zhiyao Duan 2017
More quantization levels = more dynamic range

AMPLITUDE

sample interval

TIME

quantization increment

ECE 477 - Computer Audition, Zhiyao Duan 2017
Bit Depth and Dynamics

• More bits = more quantization levels = better sound

• Compact Disc: 16 bits = 65,536 levels

• POTS (plain old telephone service): 8 bits = 256 levels

• Signal-to-quantization-noise ratio (SQNR), if the signal is uniformly distributed in the whole range
  \[ SQNR = 20 \log_{10} 2^Q \approx 6.02Q \text{ dB} \]
  – E.g. 16 bits depth gives about 96dB SQNR.
RMS

The red dots form the discrete signal $x[n]$

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x^2[n]}$$
Aliasing and Nyquist

sample interval

AMPLITUDE

TIME
Aliasing and Nyquist

Sample interval

TIME

AMPLITUDE

ECE 477 - Computer Audition, Zhiyao Duan 2017
Nyquist-Shannon Sampling Theorem

- You can’t reproduce the signal if your sample rate isn’t faster than twice the highest frequency in the signal.

- Nyquist rate: twice the frequency of the highest frequency in the signal.
  - A property of the continuous-time signal.

- Nyquist frequency: half of the sampling rate
  - A property of the discrete-time system.
Discrete-Time Fourier Transform (DTFT)

\[ X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \]

The red dots form the discrete signal \( x[n] \), where \( n = 0, \pm 1, \pm 2, \ldots \)

\( X(\omega) \) is Periodic. We often only show \([-\pi, \pi]\)

\( \omega \) is a continuous variable

ECE 477 - Computer Audition, Zhiyao Duan 2017
Relation between FT and DTFT

Sampling: \( x[n] = x_c(nT) \)

FT: \( X_c(\Omega) = \int_{-\infty}^{\infty} x_c(t)e^{-j\Omega t} \, dt \)

DTFT: \( X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \)

\[
X(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c \left( \frac{\omega}{T} + \frac{2\pi k}{T} \right)
\]

- Scaling: \( \omega = \Omega T \), i.e., \( \omega = 2\pi \) corresponds to \( \Omega = \frac{2\pi}{T} = 2\pi f_s \), which corresponds to \( f = f_s \).
- Repetition: \( X(\omega) \) contains infinite copies of \( X_c \), spaced by \( 2\pi \).
Aliasing

Complex tone
900Hz + 1800Hz

Sampling rate
= 8000Hz

Sampling rate
= 2000Hz

ECE 477 - Computer Audition, Zhiyao Duan 2017
Fourier Series

- FT and DTFT do not require the signal to be periodic, i.e., the signal may contain arbitrary frequencies, which is why the frequency domain is continuous.

- If the signal is periodic:

\[ x(t + mT) = x(t) \quad \forall m \in \mathbb{Z} \]

- It can be reproduced by a series of sine and cosine functions:

\[ x(t) = A_0 + \sum_{n=1}^{\infty} [A_n \cos(\Omega_n t) + B_n \sin(\Omega_n t)] \]

- In other words, the frequency domain is discrete.

ECE 477 - Computer Audition, Zhiyao Duan 2017
Discrete Fourier Transform (DFT)

- FT and DTFT are great, but the infinite integral or summations are hard to deal with.
- In digital computers, everything is discrete, including both the signal and its spectrum.

\[
X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}
\]

- \(X[k]\): frequency domain index
- \(x[n]\): time domain index
- \(N\): Length of the signal, i.e. length of DFT
DFT and IDFT

**DFT:** \[ X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \]

**IDFT:** \[ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} \]

- Both \( x[n] \) and \( X[k] \) are discrete and of length \( N \).
- Treats \( x[n] \) as if it were infinite and periodic.
- Treats \( X[k] \) as if it were infinite and periodic.
- Only one period is involved in calculation.
Discrete Fourier Transform

- If the time-domain signal has no imaginary part (like an audio signal) then the frequency-domain signal is conjugate symmetric around N/2.

Time domain $x[n]$

<table>
<thead>
<tr>
<th>Real portion</th>
<th>Imaginary portion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N-1</td>
<td>N-1</td>
</tr>
</tbody>
</table>

Frequency domain $X[k]$

<table>
<thead>
<tr>
<th>Real portion</th>
<th>Imaginary portion</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>fs/2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>N/2</td>
<td>N/2</td>
</tr>
<tr>
<td>N-1</td>
<td>N-1</td>
</tr>
</tbody>
</table>

DFT
IDFT
# Kinds of Fourier Transforms

<table>
<thead>
<tr>
<th>Type of Transform</th>
<th>Example Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier Transform</td>
<td><img src="image1.png" alt="Fourier Transform Example" /></td>
</tr>
<tr>
<td>Signals: continuous, aperiodic</td>
<td></td>
</tr>
<tr>
<td>Spectrum: aperiodic, continuous</td>
<td></td>
</tr>
<tr>
<td>Fourier Series</td>
<td><img src="image2.png" alt="Fourier Series Example" /></td>
</tr>
<tr>
<td>Signals: continuous, periodic</td>
<td></td>
</tr>
<tr>
<td>Spectrum: aperiodic, discrete</td>
<td></td>
</tr>
<tr>
<td>Discrete Time Fourier Transform</td>
<td><img src="image3.png" alt="Discrete Time Fourier Transform Example" /></td>
</tr>
<tr>
<td>Signals: discrete, aperiodic</td>
<td></td>
</tr>
<tr>
<td>Spectrum: periodic, continuous</td>
<td></td>
</tr>
<tr>
<td>Discrete Fourier Transform</td>
<td><img src="image4.png" alt="Discrete Fourier Transform Example" /></td>
</tr>
<tr>
<td>Signals: discrete, periodic</td>
<td></td>
</tr>
<tr>
<td>Spectrum: periodic, discrete</td>
<td></td>
</tr>
</tbody>
</table>
Duality

<table>
<thead>
<tr>
<th>Time domain</th>
<th>Frequency domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous</td>
<td>continuous</td>
</tr>
<tr>
<td>discrete</td>
<td>discrete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fourier Transform</th>
<th>DTFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>aperiodic</td>
<td>aperiodic</td>
</tr>
<tr>
<td>periodic</td>
<td>periodic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fourier Series</th>
<th>DFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>aperiodic</td>
<td>periodic</td>
</tr>
</tbody>
</table>

Frequency domain
The FFT

• Fast Fourier Transform
  – A much, much faster way to do the DFT
  – Introduced by Carl F. Gauss in 1805
  – Rediscovered by J.W. Cooley and John Tukey in 1965
  – The Cooley-Tukey algorithm is the one we use today (mostly)
  – Big O notation for this is $O(N \log N)$
  – Matlab functions `fft` and `ifft` are standard.
Windowing

- A function that is zero-valued outside of some chosen interval.
  - When a signal (data) is multiplied by a window function, the product is zero-valued outside the interval: all that is left is the "view" through the window.

\[ x[n] \times w[n] = z[n] \]

Example: windowing \( x[n] \) with a rectangular window
Some famous windows

- **Rectangular**
  \[ w[n] = 1 \]

- **Triangular (Bartlett)**
  \[ w[n] = \frac{2}{N-1} \left( \frac{N-1}{2} - \left| n - \frac{N-1}{2} \right| \right) \]

- **Hann**
  \[ w[n] = 0.5 \left( 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right) \]

Note: we assume \( w[n] = 0 \) outside some range \([0, N]\)
Why window shape matters

• Don’t forget that a DFT assumes the signal in the window is periodic
• The boundary conditions mess things up...unless you manage to have a window whose length is exactly 1 period of your signal
• Making the edges of the window less prominent helps suppress undesirable artifacts
Fourier Transform of Windows

We want
- Narrow main lobe
- Low sidelobes
Which window is better?

Hann window

\[ w[n] = 0.5 \left( 1 - \cos \left( \frac{2\pi n}{N - 1} \right) \right) \]

Hamming window

\[ w[n] = 0.54 - 0.46 \times \cos \left( \frac{2\pi n}{N - 1} \right) \]
**Multiplication v.s. Convolution**

<table>
<thead>
<tr>
<th>Time domain</th>
<th>Frequency Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x[n] \cdot y[n]$</td>
<td>$\frac{1}{N} X[k] \ast Y[k]$</td>
</tr>
<tr>
<td>$x[n] \ast y[n]$</td>
<td>$X[k] \cdot Y[k]$</td>
</tr>
</tbody>
</table>

- Windowing is multiplication in time domain, so the spectrum will be a convolution between the signal’s spectrum and the window’s spectrum.
- Convolution in time domain takes $O(N^2)$, but if we perform in the frequency domain...
  - FFT takes $O(N \log N)$
  - Multiplication takes $O(N)$
  - IFFT takes $O(N \log N)$
Windowed Signal

```matlab
fs = 10000; % sampling rate
f1 = 1000; % first sinusoid 1000Hz
f2 = 1500; % second sinusoid 1500Hz
t = 0:1/fs:3; % 3 seconds long
x1 = sin(2*pi*f1*t); % first signal
x2 = 2*sin(2*pi*f2*t); % second signal
x = x1+x2; % mixture signal
L = 100; % window length
fftLen = L*4; % fft length
w = hamming(L); % window
wx = w'.*x(101:100+L); % windowed signal
% magnitude spectrum of windowed signal
wxf = 20*log10(abs(fft(wx, fftLen)));
% show spectrum (only the positive frequencies)
figure; h = axes('FontSize', 16);
plot(h, (0:fftLen/2)*fs/fftLen, wxf(1:fftLen/2+1));
grid on;
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
```
• Two sinusoids: 1000Hz + 1500Hz
• Sampling rate: 10KHz
• Window length: 100 (i.e. 100/10K = 0.01s)
• FFT length: 400 (i.e. 4 times zero padding)
Zero Padding

- Add zeros after (or before) the signal to make it longer
- Perform DFT on the padded signal
Why Zero Padding?

- Zero padding in time domain gives the ideal interpolation in the frequency domain.
- It doesn’t increase (the real) frequency resolution!
  - 4 times is generally enough
  - Here the resolution is always $fs/L=100\text{Hz}$
How to increase frequency resolution?

- Time-frequency resolution tradeoff

\[ \Delta t \cdot \Delta f = 1 \]  
(second) (Hz)
Short time Fourier Transform

- Break signal into frames
- Window each frame
- Calculate DFT of each windowed frame
There is a “spectrogram” function in matlab.
A Fun Example

(Thanks to Robert Remez)
Overlap-Add Synthesis

- IDFT on each spectrum.
  - Use the complex, full spectrum.
  - Don’t forget the phase (often using the original phase).
  - If you do it right, the time signal you get is real.

- (optional) Multiply with a synthesis window (e.g., Hamming) to suppress signals at edges.
  - Not dividing the analysis window

- Overlap and add different frames together.
Constant Overlap Add (COLA)

- Windows of all frames add up to a constant function. **Perfect reconstruction!**

\[ \sum_{m} w[n - mR] = \text{const} \]

- Requires special design of \( w \) and \( R \)
  - Rectangular window: \( R \leq L \)
  - Triangular window: \( R = \frac{L}{k}, k \geq 2, k \in \mathbb{N} \)
  - Hamming/hann window: \( R = \frac{L}{2k}, k \in \mathbb{N} \)
Shepard Tones

Barber’s pole

Continuous Risset scale

ECE 477 - Computer Audition, Zhiyao Duan 2017
Shepard Tones

- Make a sound composed of sine waves spaced at octave intervals.
- Control their amplitudes by imposing a Gaussian (or something like it) filter in the log-frequency dimension.
- Move all the sine waves up a musical \( \frac{1}{2} \) step.
- Wrap around in frequency.