Topic

Filters, Reverberation & Convolution

Lecture by: Fatemeh Pishdadian
A system is called linear if it satisfies the superposition property:

\[ a x_1[n] + b x_2[n] \rightarrow ay_1[n] + by_2[n] \]
A system is called time invariant if its behavior does not change over time:

$$x[n] \xrightarrow{\text{SYS}} y[n]$$

$$x[n - k] \xrightarrow{\text{SYS}} y[n - k]$$
LTI systems

• Why do we like to think of systems we work with as LTI?

• Examples of linear systems?

• Examples of time invariant systems?
Impulse response

• An “impulse” is this signal:

\[ \delta[n] = \begin{cases} 
1 & \text{if } n = 0 \\
0 & \text{else} 
\end{cases} \]

• The *impulse response* \( h[n] \) of a system is the output of the system when the input is an impulse.

• The *frequency response* \( H(\omega) \) of a system is the Fourier transform of its impulse response \( h[n] \).
Impulse response

Impulse response

sr:100Hz  fmax:5Hz

Impulse response

Impulse response

Impulse response

Impulse response

• An LTI system is fully identified by its impulse response (or frequency response), because…

  o An arbitrary signal $x[n]$ is the sum of scaled and shifted impulse functions:
    \[ x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k] \]

  o Then if we have $h[n]$, by assuming linearity and time invariance we can find the response to $x[n]$
    \[ y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k] \]

Does this formula look familiar?
Example: Delay Operator

\[ h_d[n] = \delta[n-20] \]

\[ x[n] \]

\[ h_d[n] * x[n] = x[n-20] \]
Example: Moving Average Operator

$h_{ma}[n]$
Frequency selective filters

- We use LTI systems in this course as frequency selective filters.

- The output of filters as explained will be computed via convolution in the time domain, or equivalently, via multiplication in the frequency domain.

- Can you name a real world example of convolution as summing up attenuated and delayed copies of a signal?
Filtering in the time domain

Output Signal $y(t)$ \hspace{1cm} Filter $h(t)$ \hspace{1cm} Source Signal $x(t)$

$y(t) = h(t) \ast x(t)$

Impulse Response

Filtering in the frequency domain

\[ Y(\omega) = H(\omega)X(\omega) \]

Source Signal \( x(t) \) \( \rightarrow \) Filter \( h(t) \) \( \rightarrow \) Output Signal \( y(t) \)

Frequency Response
What is reverberation?

• Reverberation is made of echoes

• Echoes are delayed copies of the original sound

• In the physical world these are caused by reflections off of walls and other surfaces

• In the digital world, this is done with impulse response functions and convolution
They are all strongly related

• Room can be modeled as a filter

• **Reverb** is the effect of room/filter on sound signals

• Adding reverb to sound signals (filtering) is performed via **convolution**
Acting as a low-pass filter

• The moving average operator makes copies of the signal and sums them.

• You’re summing multiple, slightly-offset copies.

• This causes high-frequency events to average out in the wash (low-pass filtering).

• What would happen if the copies are offset by the exact period of some frequency in the audio?
Note: Delay and filters

• The impulse response function of your filter has some length.

• That length determines how many samples go by before the filter’s response becomes steady.

• Therefore, filtering imposes delays.
A room is a filter

Carnegie Hall
Reverberant Rooms

• …are louder (why?)

• …make speech harder to understand (why?)

• …may emphasize certain frequencies (why)?
Getting the impulse response

• Walk into a QUIET room with a recorder
• Turn on the recorder
• Clap your hands ONCE (this is your impulse)
• The recording captures the room’s impulse response
• …and other noise (air conditioner, etc.)
Noise and Distortion

Source Signal $X(w)$

Filter $H(\omega)$

Distortion $D(w)$

Noise $N(w)$

Output Signal $Y(w)$

Distortion and noise make it a lot harder to estimate $H$
Estimating the frequency response $H$

- Assume we know what $X$ is (because we made it) and what $Y$ is (‘cause we recorded it).

- Hope noise and distortion are not correlated with $X$.

- Call “noise + distortion” $N$.

- Then…

$$X[k]H[k] + N[k] = Y[k]$$
Estimating the frequency response $H$

- If there were no noise, we could estimate $H[k]$ in the frequency domain

\[ X[k]H[k] = Y[k] \]

so….

\[ H[k] = Y[k] / X[k] \]

...more or less
Estimating the frequency response $H$

- But there is noise ...
- What do we do?
- We assume noise is uncorrelated to the signal or the filter
- We assume noise is unbiased (zero mean)
- We try lots of estimates and hope that the noise "washes out" when we average
Estimating $H$

- Estimate $H$ a lot of times, in the hopes that the noise will wash out in the mix…

$$
\hat{H}[k] \approx \frac{\langle Y[k] X[k]^* \rangle}{\langle X[k] X[k]^* \rangle}
$$

where

$$
\langle Y[k] X[k]^* \rangle = \frac{1}{N} \sum_{n=1}^{N} Y_n[k] X_n[k]^*
$$

The average value over $N$ experiments.
Caveats

• This only works on the frequencies where there was energy in the input signal X.

• If there wasn’t energy at a frequency then we’re out of luck.

• So, best to use a broadband sound for X.

• Note: An impulse is broadband.
Adding room reverb to a recording

• Record an impulse in the room

• Estimate the impulse response of the room using the method described previously

• Record something in another quiet room (with the microphone up close to the source)

• Convolution between your new recording and the impulse response function of the first room
FIR Filter

- FIR means “Finite Impulse Response”

- Means it will stop making noise once you stop putting noise through it

- There are also Infinite Impulse Response (IIR) filters. These have feedback

- To find out more about IIR filters, take a DSP class.

Building a low-pass FIR filter

• Pick width of your LOW pass band
  (0 Hz to $\omega_c$ Hz, where $\omega_c$ can be up to the Nyquist rate)

• Create a desired frequency response (don’t forget the mirror frequencies above the Nyquist rate).

• Take the IFFT of the frequency response

• This is your impulse response function.
An 8-point 250 Hz low-pass filter

<table>
<thead>
<tr>
<th>Python Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>h[n]</td>
<td>0.71</td>
<td>0.26</td>
<td>-0.18</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>-0.18</td>
<td>0.26</td>
</tr>
<tr>
<td>H [k]</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

DC offset
Complex conjugates
Nyquist frequency

<table>
<thead>
<tr>
<th>Better index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4 (-4)</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of FFT</td>
<td>0</td>
<td>S/N</td>
<td>2S/N</td>
<td>3S/N</td>
<td>4S/N</td>
<td>-3S/N</td>
<td>-2S/N</td>
<td>-S/N</td>
</tr>
<tr>
<td>If S=1000 Hz and N = 8</td>
<td>0</td>
<td>125Hz</td>
<td>250Hz</td>
<td>375Hz</td>
<td>500Hz</td>
<td>-375Hz</td>
<td>-250Hz</td>
<td>-125Hz</td>
</tr>
</tbody>
</table>

The same 8-point low-pass filter
(another view)