Lecture 14: Windowing

ECE 401: Signal and Image Analysis

University of Illinois

3/29/2017

Outline

DTFT Review

When two signals are convolved, their DTFTs get multiplied together

$$y[n] = h[n] * x[n] \Leftrightarrow Y(\omega) = H(\omega)X(\omega)$$

where

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

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega$$

Ideal Filters

Probably the most important DTFT pairs are the ideal LPF, BPF, and HPF:

$$H_{LPF}(\omega) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \text{else} \end{cases} \Leftrightarrow h_{LPF}[n] = \begin{cases} \frac{\sin(\omega_c n)}{\pi n} & n \neq 0 \\ \frac{\omega_c}{\pi} & n = 0 \end{cases}$$

$$H_{BPF}(\omega) = \begin{cases} 1 & \omega_1 < |\omega| < \omega_2 \\ 0 & \text{else} \end{cases} \Leftrightarrow h_{BPF}[n] = \begin{cases} \frac{\sin(\omega_2 n) - \sin(\omega_1 n)}{\pi n} & n \neq \\ \frac{\omega_2 - \omega_1}{\pi} & n = \end{cases}$$

$$H_{HPF}(\omega) = \begin{cases} 1 & \omega_1 < |\omega| \leq \pi \\ 0 & \text{else} \end{cases} \Leftrightarrow h_{HPF}[n] = \begin{cases} \delta[n] - \frac{\sin(\omega_1 n)}{\pi n} & n \neq 0 \\ 1 - \frac{\omega_1}{\pi} & n = 0 \end{cases}$$

Outline

Real filters are usually FIR

- The problem with ideal filters is that $h[n] = \frac{\sin(\omega_c n)}{\pi n} = \left(\frac{\omega_c}{\pi}\right)\text{sinc}(\omega_c n)$ is infinite length. Infinite length convolution is not possible on a computer with finite memory and finite time:

$$y[n] = \sum_{m=-\infty}^{\infty} h[n]x[n-m]$$

- A real-world filter needs to have finite computation. For that, it usually needs a **finite impulse response** (FIR), e.g., a filter of length $N = 2M + 1$ requires N multiply operations per output sample:

$$y[n] = \sum_{m=-M}^{M} h[m]x[n-m]$$

Frequency Sampling Doesn't Work

You're probably thinking: why FIR? Why not just do it all in the frequency domain?

$$x[n] \rightarrow X(\omega) \rightarrow Y(\omega) = H(\omega)X(\omega) \rightarrow y[n] \tag{1}$$

The problem is that we'd need to do this with discrete omega, $\omega = \frac{2\pi k}{N}$, instead of continuous ω. The only way a computer can do this is using, effectively, a discrete Fourier series:

$$x[n] \rightarrow X_k \rightarrow Y_k = H\left(\frac{2\pi k}{N}\right) X_k \rightarrow y[n] \tag{2}$$

The problem is that Eq. 2 gives a different result from Eq. 1. Eq. 2 pretends that $x[n]$ is periodic, with a period of N samples, even if it isn't really. The pretense of periodicity causes artifacts: multiplying by $H(\omega)$ causes the imagined periods of $x[n]$ to blur into one another. This is called **temporal aliasing**.

FIR filters need to be windowed

So we need to do it all in the time domain. This is exactly what `numpy.convolve` does:

$$y[n] = \sum_{m=-M}^{M} h[m][n-m]$$

The problem is that the FIR and IIR (infinite impulse response) filters are not the same. In fact, they are related by windowing:

$$h_{FIR}[m] = w[m]h_{IIR}[m] = \begin{cases} h_{IIR}[m] & |m| \leq M \\ 0 & \text{else} \end{cases}$$

$$w[m] = \begin{cases} 1 & |m| \leq M \\ 0 & \text{otherwise} \end{cases}$$

The Result of Windowing

What does windowing do? To get a hint, remember the convolution property of DTFT:

$$y[n] = h[n] * x[n] \leftrightarrow Y(\omega) = H(\omega)X(\omega)$$

It turns out that almost the same thing works in reverse:

$$h_{FIR}[n] = w[n]h_{IIR}[n] \leftrightarrow H_{FIR}(\omega) = \frac{1}{2\pi}W(\omega) * H_{IIR}(\omega)$$

Now we need to define "convolution in frequency." We define it like this:

$$H_{FIR}(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\theta)H_{IIR}(\omega - \theta)d\theta$$

Convolution in Frequency Example: IIR

As an easy-to-compute example, suppose

$$h[n] = \left(\frac{\omega_c}{\pi}\right) \text{sinc}(\omega_c n) \leftrightarrow H(\omega) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases}$$

Suppose we square it:

$$g[n] = h^2[n] = \left(\frac{\omega_c}{\pi}\right)^2 \text{sinc}^2(\omega_c n)$$

Then

$$G(\omega) = \frac{1}{2\pi} H(\omega) * H(\omega) = \begin{cases} \left(\frac{\omega_c - |\omega|}{\pi}\right) & |\omega| < 2\omega_c \\ 0 & \text{otherwise} \end{cases}$$

Outline

Rectangular Window

The most useful window is the rectangular window:

$$w_R[n] = \begin{cases} 1 & |n| \leq M \\ 0 & \text{else} \end{cases}$$

$$W_R(\omega) = \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

where $N = 2M + 1$ is the length of the window.

Windowing with a Rectangular Window

Window with a rectangular window:

$$h_{FIR}[n] = w_R[n]h_{IIR}[n] \leftrightarrow H_{FIR}(\omega) = \frac{1}{2\pi}W_R(\omega) * H_{IIR}(\omega)$$

Causes the following effects:

- $H_{FIR}(\omega)$ has **ripples in the passband**, going up and down, crossing the value $H_{FIR}(\omega) = 1$ only once every $2\pi/N$ radians.
- $H_{FIR}(\omega)$ has **ripples in the stopband**, going up and down, crossing the value $H_{FIR}(\omega) = 0$ only once every $2\pi/N$ radians.
- $H_{FIR}(\omega)$ has a gradual **transition band** between the passband and stopband, with a width of $2\pi/N$.

Other Useful Windows

- Sidelobes can be made smaller, at the expense of a wider transition band.
- This is done by making the window less abrupt in the time domain.

Other Useful Windows

$$\textbf{Triangular (Bartlett):} w_B[n] = \left(1 - \frac{|n|}{M+1}\right) w_R[n]$$

$$\textbf{Hamming Window:} w_H[n] = \left(0.54 + 0.46 \cos\left(\frac{\pi n}{M}\right)\right) w_R[n]$$

$$\textbf{Hann Window:} w_N[n] = \left(0.5 + 0.5 \cos\left(\frac{\pi n}{M}\right)\right) w_R[n]$$

Other Useful Windows

Window	First Null	First Sidelobe	Other Sidelobes
Rectangular	$\omega = \frac{2\pi}{N}$	-13dB	$1/N$
Triangular	$\omega = \frac{4\pi}{N}$	-26dB	$1/N^2$
Hamming	$\omega = \frac{4\pi}{N}$	-44dB	flat
Hann	$\omega = \frac{4\pi}{N}$	-32dB	Very Small