

Lecture 17: Windowing

Mark Hasegawa-Johnson

These slides are in the public domain

ECE 401: Signal and Image Analysis

- ① Review: Ideal Filters
- ② Realistic Filters: Finite Length
- ③ Multiplication is the Fourier Transform of Convolution!
- ④ Realistic Filters: Even Length
- ⑤ Summary

Outline

- 1 Review: Ideal Filters
- 2 Realistic Filters: Finite Length
- 3 Multiplication is the Fourier Transform of Convolution!
- 4 Realistic Filters: Even Length
- 5 Summary

Review: Ideal Filters

- Ideal Lowpass Filter:

$$H_{LP}(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c, \\ 0 & \omega_c < |\omega| \leq \pi. \end{cases} \quad \leftrightarrow \quad h_{LP}[m] = \frac{\omega_c}{\pi} \text{sinc}(\omega_c n)$$

- Ideal Highpass Filter:

$$H_{HP}(\omega) = 1 - H_{LP}(\omega) \quad \leftrightarrow \quad h_{HP}[n] = \delta[n] - \frac{\omega_c}{\pi} \text{sinc}(\omega_c n)$$

- Ideal Bandpass Filter:

$$H_{BP}(\omega) = H_{LP,\omega_2}(\omega) - H_{LP,\omega_1}(\omega) \\ \leftrightarrow h_{BP}[n] = \frac{\omega_2}{\pi} \text{sinc}(\omega_2 n) - \frac{\omega_1}{\pi} \text{sinc}(\omega_1 n)$$

Outline

- 1 Review: Ideal Filters
- 2 Realistic Filters: Finite Length
- 3 Multiplication is the Fourier Transform of Convolution!
- 4 Realistic Filters: Even Length
- 5 Summary

Ideal Filters are Infinitely Long

- All of the ideal filters, $h_{LP,i}[n]$ and so on, are infinitely long!
- In demos so far, I've faked infinite length by just making $h_{LP,i}[n]$ more than twice as long as $x[n]$.
- If $x[n]$ is very long (say, a 24-hour audio recording), you probably don't want to do that (computation=expensive)

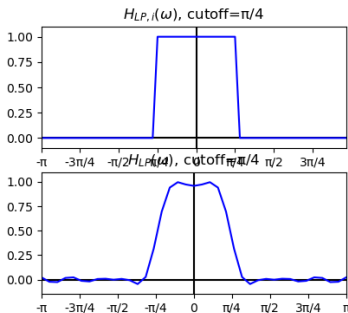
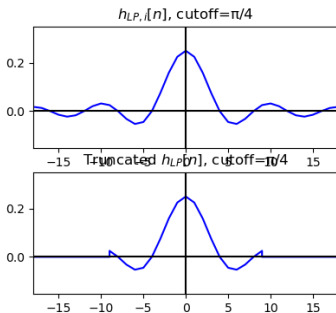
Finite Length by Truncation

We can force $h_{LP,i}[n]$ to be finite length by just truncating it, say, to $2M + 1$ samples:

$$h_{LP}[n] = \begin{cases} h_{LP,i}[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

Truncation Causes Frequency Artifacts

The problem with truncation is that it causes artifacts.



Windowing Reduces the Artifacts

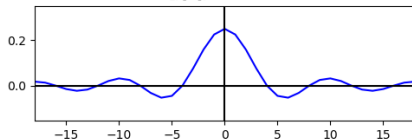
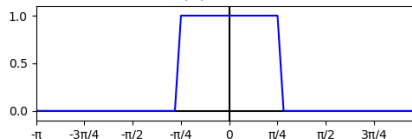
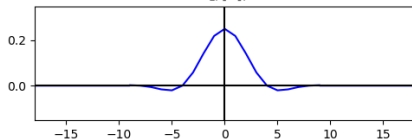
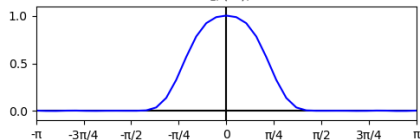
We can reduce the artifacts (a lot) by windowing $h_{LP,i}[n]$, instead of just truncating it:

$$h_{LP}[n] = \begin{cases} w[n]h_{LP,i}[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

where $w[n]$ is a window that tapers smoothly down to near zero at $n = \pm M$, e.g., a Hamming window:

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{2M}\right)$$

Windowing Reduces the Artifacts

 $h_{LP}[n]$, cutoff= $\pi/4$  $H_{LP}(\omega)$, cutoff= $\pi/4$ Windowed $h_{LP}[n]$, cutoff= $\pi/4$ Windowed $H_{LP}(\omega)$, cutoff= $\pi/4$ 

Outline

- 1 Review: Ideal Filters
- 2 Realistic Filters: Finite Length
- 3 Multiplication is the Fourier Transform of Convolution!**
- 4 Realistic Filters: Even Length
- 5 Summary

Why does truncation cause artifacts?

But why does truncation cause artifacts?

The reason is that, when we truncate an impulse response, we are (unintentionally?) multiplying it by a rectangular window:

$$\begin{aligned} h_{LP}[n] &= \begin{cases} h_{LP,i}[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases} \\ &= w_R[n] h_{LP,i}[n] \end{aligned}$$

... where $w_R[n]$ is a function called the “rectangular window:”

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

Review: DTFT of Convolution is Multiplication

Remember that the DTFT of convolution is multiplication. If

$$y[n] = h[n] * x[n]$$

... then ...

$$Y(\omega) = H(\omega)X(\omega)$$

New Stuff: DTFT of Multiplication is Convolution!

Guess what: the DTFT of multiplication is ($1/2\pi$ times) convolution!! If

$$g[n] = w[n]h[n]$$

... then ...

$$G(\omega) = \frac{1}{2\pi} W(\omega) * H(\omega)$$

Definition and proof: convolution in frequency

The previous slide used the formula “ $W(\omega) * H(\omega)$ ”. What does that even mean?

To find out, let's try taking the DTFT of $g[n]$:

$$\begin{aligned} G(\omega) &= \sum_n g[n] e^{-j\omega n} \\ &= \sum_n w[n] h[n] e^{-j\omega n} \\ &= \sum_n w[n] \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} H(\theta) e^{j\theta n} d\theta \right) e^{-j\omega n} \end{aligned}$$

In the last line, notice the difference between θ and ω . One is the dummy variable for the IDTFT, one is the dummy variable for the DTFT.

Definition and proof: convolution in frequency

Now let's complete the derivation:

$$\begin{aligned} G(\omega) &= \sum_n w[n] \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} H(\theta) e^{j\theta n} d\theta \right) e^{-j\omega n} \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\theta) \left(\sum_n w[n] e^{-j(\omega-\theta)n} \right) d\theta \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\theta) W(\omega - \theta) d\theta \end{aligned}$$

New Stuff: DTFT of Multiplication is Convolution!

So when we window a signal in the time domain,

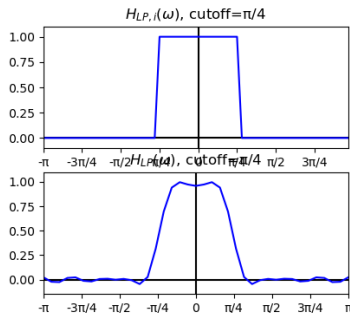
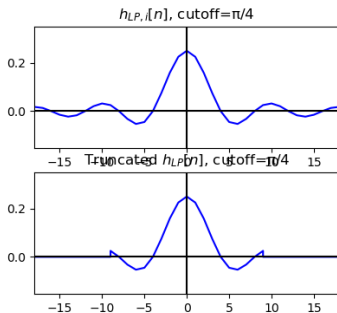
$$g[n] = w[n]h[n]$$

That's equivalent to convolving $H(\omega)$ by the DTFT of the window,

$$\begin{aligned} G(\omega) &= \frac{1}{2\pi} W(\omega) * H(\omega) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\theta) W(\omega - \theta) d\theta \end{aligned}$$

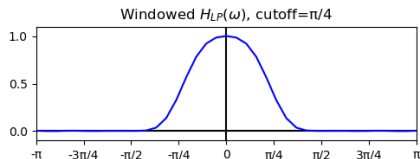
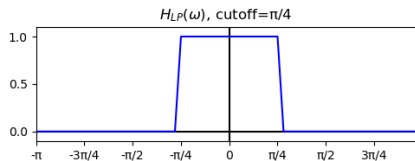
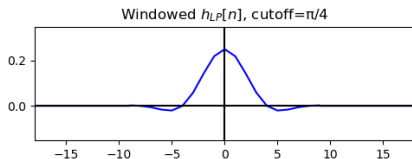
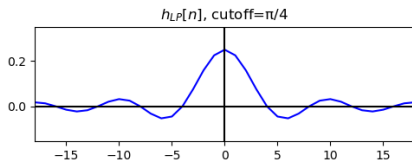
Windowing Causes Frequency Artifacts

We've already seen the result. Windowing by a rectangular window (i.e., truncation) causes nasty artifacts!



Windowing Reduces the Artifacts

... whereas windowing by a smooth window, like a Hamming window, causes a lot less artifacts:



Quiz

Go to the course web page, and try the quiz!

Outline

- 1 Review: Ideal Filters
- 2 Realistic Filters: Finite Length
- 3 Multiplication is the Fourier Transform of Convolution!
- 4 Realistic Filters: Even Length**
- 5 Summary

Even Length Filters

Often, we'd like our filter $h_{LP}[n]$ to be even length, e.g., 200 samples long, or 256 samples. We can't do that with this definition:

$$h_{LP}[n] = \begin{cases} w[n]h_{LP,i}[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

... because $2M + 1$ is always an odd number.

Even Length Filters using Delay

We can solve this problem using the time-shift property of the DTFT:

$$z[n] = x[n - n_0] \quad \leftrightarrow \quad Z(\omega) = e^{-j\omega n_0} X(\omega)$$

Even Length Filters using Delay

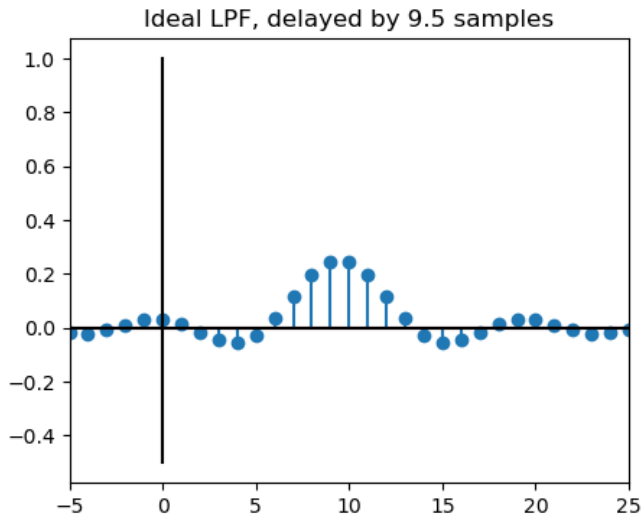
Let's delay the ideal filter by exactly $M - 0.5$ samples, for any integer M :

$$z[n] = h_{LP,i}[n - (M - 0.5)] = \frac{\omega_c}{\pi} \text{sinc} \left(\omega \left(n - M + \frac{1}{2} \right) \right)$$

I know that sounds weird. But notice the symmetry it gives us. The whole signal is symmetric w.r.t. sample $n = M - 0.5$. So $z[M - 1] = z[M]$, and $z[M - 2] = z[M + 1]$, and so on, all the way out to

$$z[0] = z[2M - 1] = \frac{\omega_c}{\pi} \text{sinc} \left(\omega \left(M - \frac{1}{2} \right) \right)$$

Even Length Filters using Delay



Even Length Filters using Delay

Apply the time delay property:

$$z[n] = h_{LP,i}[n - (M - 0.5)] \quad \leftrightarrow \quad Z(\omega) = e^{-j\omega(M-0.5)} H_{LP,i}(\omega),$$

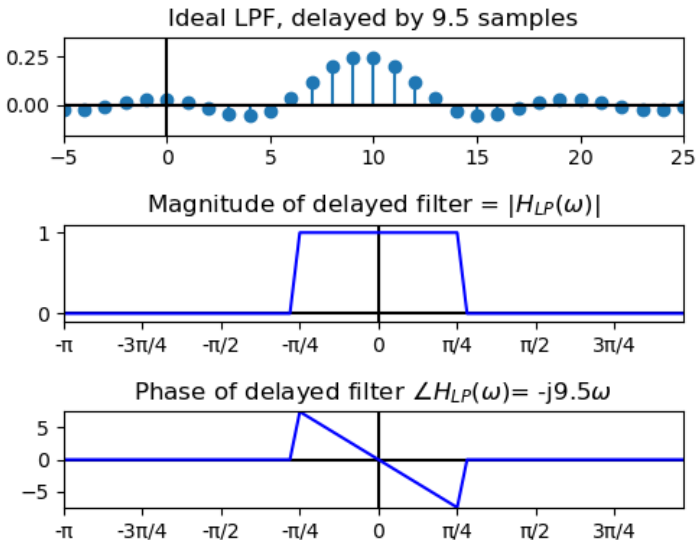
and then notice that

$$|e^{-j\omega(M-0.5)}| = 1$$

So

$$|Z(\omega)| = |H_{LP,i}(\omega)|$$

Even Length Filters using Delay



Even Length Filters using Delay and Windowing

Now we can create an even-length filter by windowing the delayed filter:

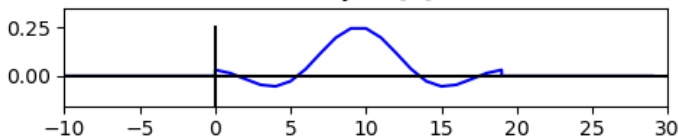
$$h_{LP}[n] = \begin{cases} w[n] h_{LP,i}[n - (M - 0.5)] & 0 \leq n \leq (2M - 1) \\ 0 & \text{otherwise} \end{cases}$$

where $w[n]$ is a Hamming window defined for the samples $0 \leq m \leq 2M - 1$:

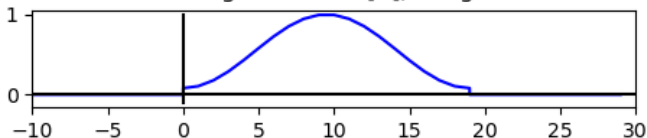
$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{2M - 1}\right)$$

Even Length Filters using Delay and Windowing

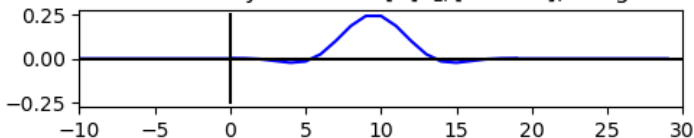
Truncated Delayed $I[n]$, cutoff= $\pi/4$



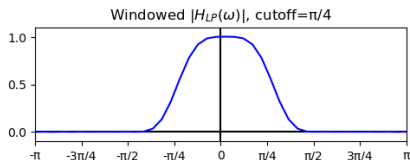
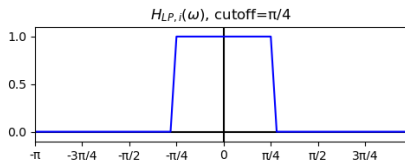
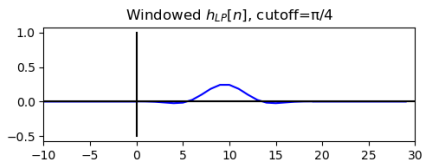
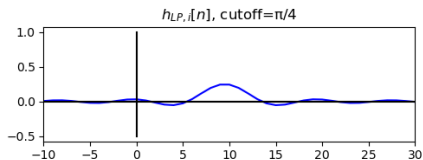
Hamming Window $w[n]$, Length=20



Windowed Delayed Filter $w[n]h_{LP}[n - 9.5]$, Length=21



Even Length Filters using Delay and Windowing



Outline

- 1 Review: Ideal Filters
- 2 Realistic Filters: Finite Length
- 3 Multiplication is the Fourier Transform of Convolution!
- 4 Realistic Filters: Even Length
- 5 Summary

Summary: Ideal Filters

- Ideal Lowpass Filter:

$$H_{LP}(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c, \\ 0 & \omega_c < |\omega| \leq \pi. \end{cases} \quad \leftrightarrow \quad h_{LP}[m] = \frac{\omega_c}{\pi} \text{sinc}(\omega_c n)$$

- Ideal Highpass Filter:

$$H_{HP}(\omega) = 1 - H_{LP}(\omega) \quad \leftrightarrow \quad h_{HP}[n] = \delta[n] - \frac{\omega_c}{\pi} \text{sinc}(\omega_c n)$$

- Ideal Bandpass Filter:

$$H_{BP}(\omega) = H_{LP, \omega_2}(\omega) - H_{LP, \omega_1}(\omega) \\ \leftrightarrow h_{BP}[n] = \frac{\omega_2}{\pi} \text{sinc}(\omega_2 n) - \frac{\omega_1}{\pi} \text{sinc}(\omega_1 n)$$

Summary: Practical Filters

- Odd Length:

$$h_{HP}[n] = \begin{cases} h_{HP,i}[n]w[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

- Even Length:

$$h_{HP}[n] = \begin{cases} h_{HP,i}[n - (M - 0.5)]w[n] & 0 \leq n \leq 2M - 1 \\ 0 & \text{otherwise} \end{cases}$$

where $w[n]$ is a window with tapered ends, e.g.,

$$w[n] = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{L-1}\right) & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$