Lecture 10: Ideal Filters

Mark Hasegawa-Johnson

ECE 401: Signal and Image Analysis, Fall 2020

Outline

Review: DTFT

The DTFT (discrete time Fourier transform) of any signal is $X(\omega)$, given by

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

Particular useful examples include:

$$f[n] = \delta[n] \leftrightarrow F(\omega) = 1$$
$$g[n] = \delta[n-n_0] \leftrightarrow G(\omega) = e^{-j\omega n_0}$$

Properties of the DTFT

Properties worth knowing include:

0. Periodicity: $X(\omega + 2\pi) = X(\omega)$
1. Linearity:

$$z[n] = ax[n] + by[n] \leftrightarrow Z(\omega) = aX(\omega) + bY(\omega)$$

2. Time Shift: $x[n - n_0] \leftrightarrow e^{-j\omega n_0}X(\omega)$
3. Frequency Shift: $e^{j\omega_0 n}x[n] \leftrightarrow X(\omega - \omega_0)$
4. Filtering is Convolution:

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

Outline

What is "Ideal"?

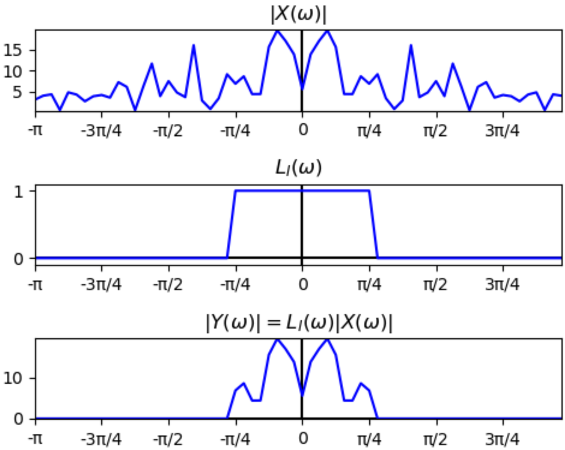
The definition of "ideal" depends on your application. Let's start with the task of lowpass filtering. Let's define an ideal lowpass filter, $Y(\omega) = L_I(\omega)X(\omega)$, as follows:

$$Y(\omega) = \begin{cases} X(\omega) & |\omega| \le \omega_L, \\ 0 & \text{otherwise}, \end{cases}$$

where ω_L is some cutoff frequency that we choose. For example, to de-noise a speech signal we might choose $\omega_L = 2\pi 2400/F_s$, because most speech energy is below 2400Hz. This definition gives:

$$L_I(\omega) = \begin{cases} 1 & |\omega| \le \omega_L \\ 0 & \text{otherwise} \end{cases}$$

Ideal Lowpass Filter



How can we implement an ideal LPF?

1. Use `np.fft.fft` to find $X[k]$, set $Y[k] = X[k]$ only for $\frac{2\pi k}{N} < \omega_L$, then use `np.fft.ifft` to convert back into the time domain?
    - It sounds easy, but...
    - `np.fft.fft` is finite length, whereas the DTFT is infinite length. Truncation to finite length causes artifacts.
2. Use pencil and paper to inverse DTFT $L_I(\omega)$ to $l_I[n]$, then use `np.convolve` to convolve $l_I[n]$ with $x[n]$.
    - It sounds more difficult.
    - But actually, we only need to find $l_I[n]$ once, and then we'll be able to use the same formula for ever afterward.
    - This method turns out to be both easier and more effective in practice.

Inverse DTFT of $L_I(\omega)$

The ideal LPF is

$$L_I(\omega) = \begin{cases} 1 & |\omega| \le \omega_L \\ 0 & \text{otherwise} \end{cases}$$

The inverse DTFT is

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

Combining those two equations gives

$$l_I[n] = \frac{1}{2\pi}\int_{-\omega_L}^{\omega_L} e^{j\omega n} d\omega$$

Solving the integral

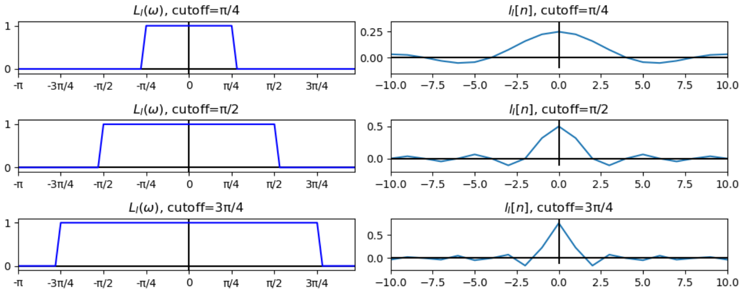
The ideal LPF is

$$l_I[n] = \frac{1}{2\pi}\int_{-\omega_L}^{\omega_L} e^{j\omega n} d\omega = \frac{1}{2\pi}\left(\frac{1}{jn}\right)\left[e^{j\omega n}\right]_{-\omega_L}^{\omega_L} = \frac{1}{2\pi}\left(\frac{1}{jn}\right)(2j\sin(\omega_L n))$$

So

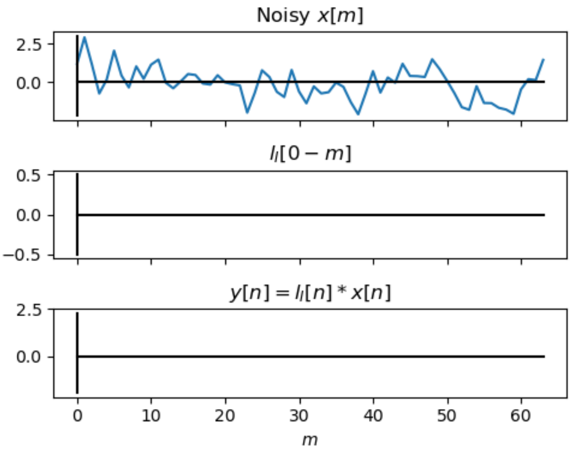
$$l_I[n] = \frac{\sin(\omega_L n)}{\pi n}$$

$l_l[n] = \frac{\sin(\omega_L n)}{\pi n}$



- $\frac{\sin(\omega_L n)}{\pi n}$ is undefined when $n = 0$
- $\lim_{n\rightarrow 0} \frac{\sin(\omega_L n)}{\pi n} = \frac{\omega_L}{\pi}$
- So let's define $l_l[0] = \frac{\omega_L}{\pi}$.

$l_l[n] = \frac{\sin(\omega_L n)}{\pi n}$



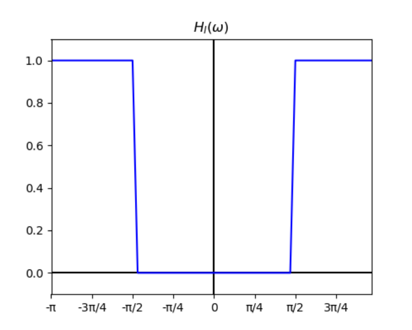
Outline

Ideal Highpass Filter

An ideal high-pass filter passes all frequencies above ω_H:

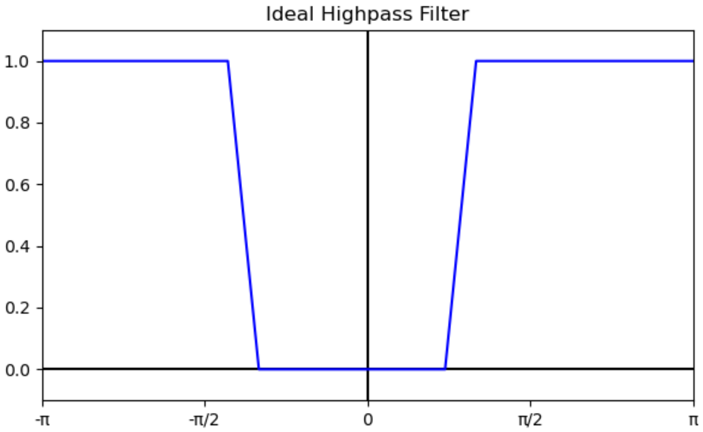
$$H_I(\omega) = \begin{cases} 1 & |\omega| > \omega_H \\ 0 & \text{otherwise} \end{cases}$$

Ideal Highpass Filter



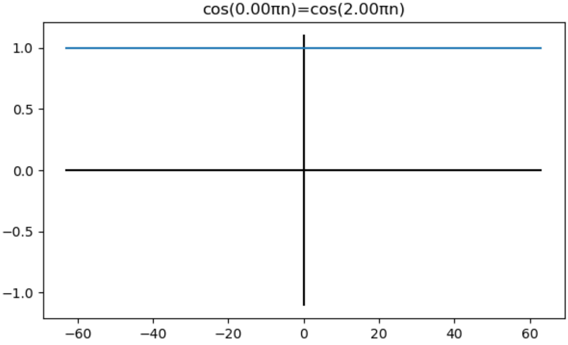
Ideal Highpass Filter

. . . except for one problem: $H(\omega)$ is periodic with a period of 2π.



The highest frequency is $\omega = \pi$

The highest frequency, in discrete time, is $\omega = \pi$. Frequencies that seem higher, like $\omega = 1.1\pi$, are actually lower. This phenomenon is called "aliasing."



Redefining "Lowpass" and "Highpass"

Let's redefine "lowpass" and "highpass." The ideal LPF is

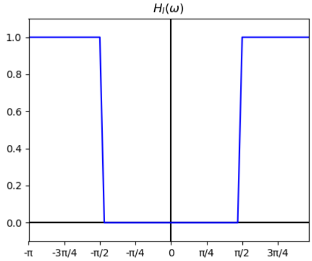
$$L_I(\omega) = \begin{cases} 1 & |\omega| \leq \omega_L, \\ 0 & \omega_L < |\omega| \leq \pi. \end{cases}$$

The ideal HPF is

$$H_I(\omega) = \begin{cases} 0 & |\omega| < \omega_H, \\ 1 & \omega_H \leq |\omega| \leq \pi. \end{cases}$$

Both of them are periodic with period 2π.

Inverse DTFT of $H_I(\omega)$



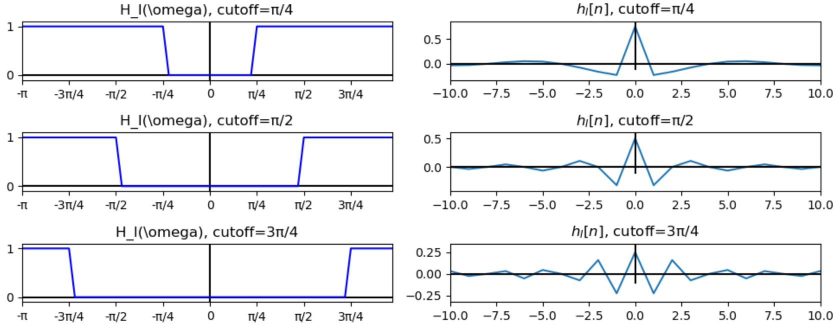
The easiest way to find $h_I[n]$ is to use linearity:

$$H_I(\omega) = 1 - L_I(\omega)$$

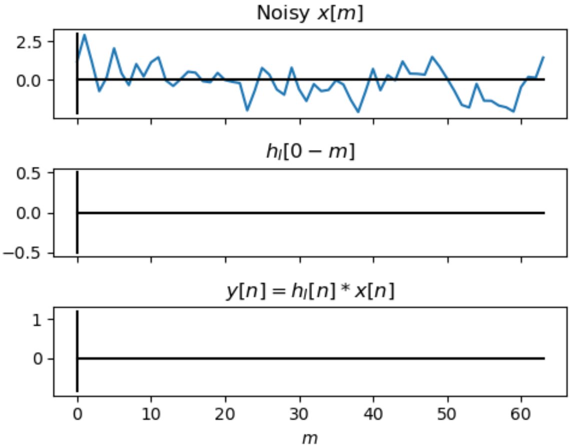
Therefore:

$$\begin{aligned} h_I[n] &= \delta[n] - l_I[n] \\ &= \delta[n] - \frac{\sin(\omega_H n)}{\pi n} \end{aligned}$$

$h_l[n] = \delta[n] - \frac{\sin(\omega_H n)}{\pi n}$



$h_l[n] = \delta[n] - \frac{\sin(\omega_L n)}{\pi n}$



Outline

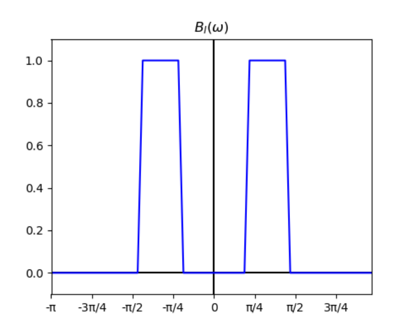
Ideal Bandpass Filter

An ideal band-pass filter passes all frequencies between ω_H and ω_L:

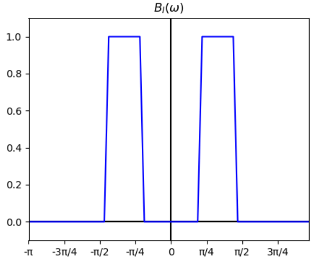
$$B_I(\omega) = \begin{cases} 1 & \omega_H \le |\omega| \le \omega_L \\ 0 & \text{otherwise} \end{cases}$$

(and, of course, it's also periodic with period 2π).

Ideal Bandpass Filter



Inverse DTFT of $B_I(\omega)$



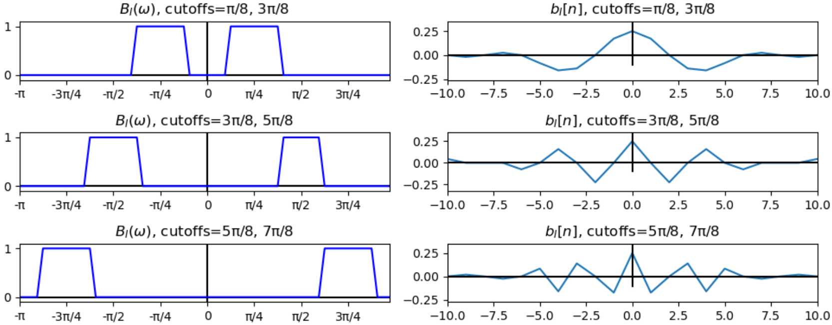
The easiest way to find $b_I[n]$ is to use linearity:

$$B_I(\omega) = L_I(\omega|\omega_L) - L_I(\omega|\omega_H)$$

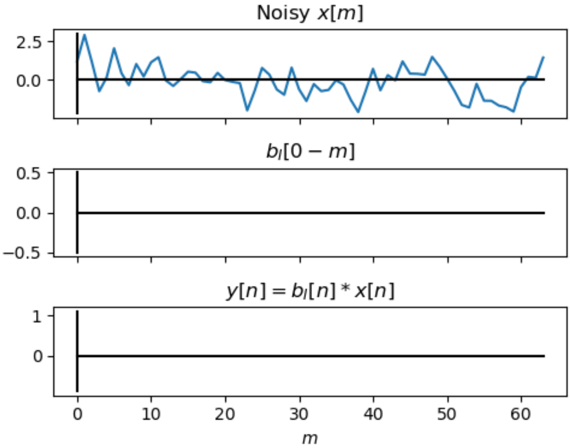
Therefore:

$$b_I[n] = \frac{\sin(\omega_L n)}{\pi n} - \frac{\sin(\omega_H n)}{\pi n}$$

$b_l[n] = \frac{\sin(\omega_L n)}{\pi n} - \frac{\sin(\omega_H n)}{\pi n}$



$b_l[n] = \frac{\sin(\omega_L n)}{\pi n} - \frac{\sin(\omega_H n)}{\pi n}$



Outline

Ideal Filters are Infinitely Long

- All of the ideal filters, $l_I[n]$ and so on, are infinitely long.
- In videos so far, I've faked infinite length by just making $l_I[n]$ 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)

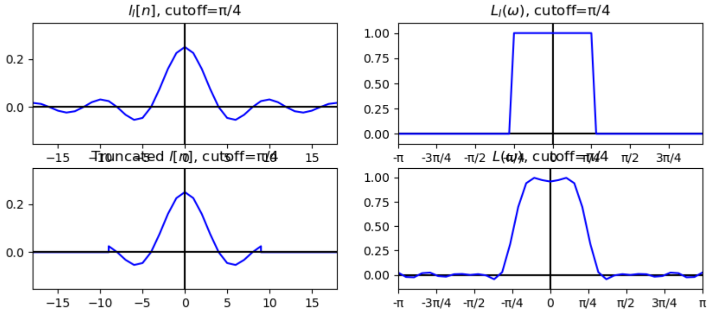
Finite Length by Truncation

We can force $l_I[n]$ to be finite length by just truncating it, say, to $2M+1$ samples:

$$l[n] = \begin{cases} l_I[n] & -M \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

Truncation Causes Frequency Artifacts

The problem with truncation is that it causes artifacts.



Windowing Reduces the Artifacts

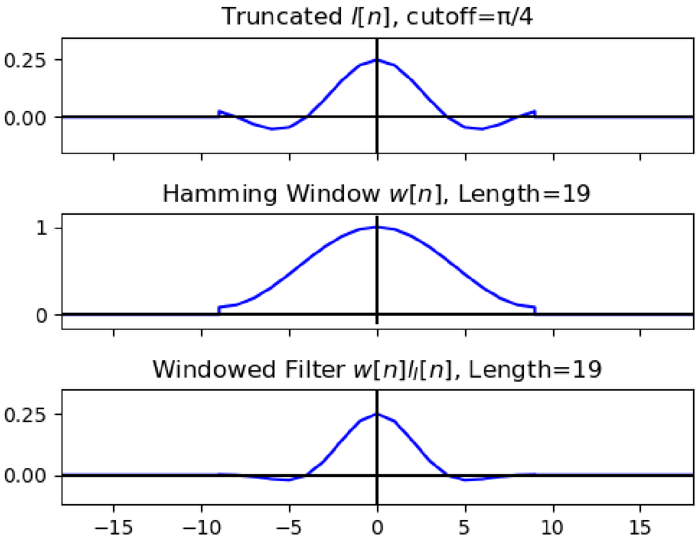
We can reduce the artifacts (a lot) by windowing $l_I[n]$, instead of just truncating it:

$$l[n] = \begin{cases} w[n]l_I[n] & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

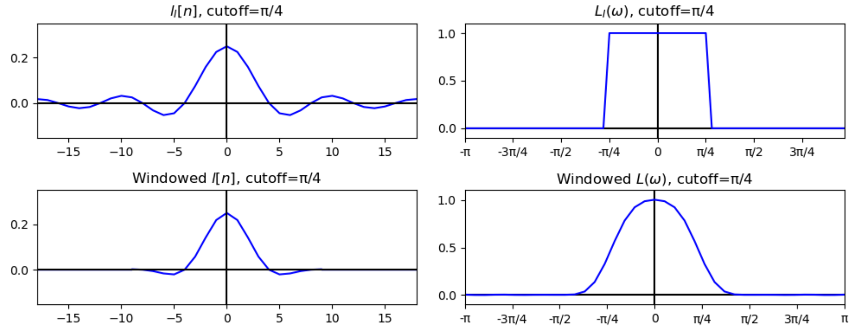
where $w[n]$ is a window that tapers smoothly down to near zero at $n = \pm M$, e.g., a Hamming window:

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

Windowing a Lowpass Filter



Windowing Reduces the Artifacts



Outline

Even Length Filters

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

$$l[n] = \begin{cases} w[n]l_I[n] & -M \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

...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

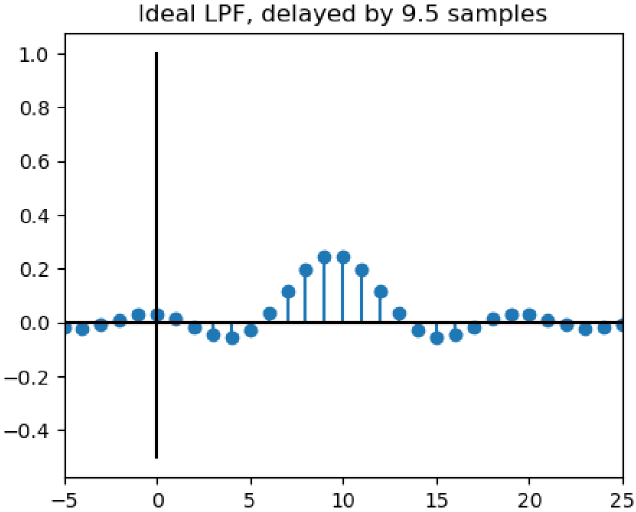
Let's delay the ideal filter by exactly $M - 0.5$ samples, for any integer M:

$$z[n] = l_I\left[n - (M - 0.5)\right] = \frac{\sin\left(\omega\left(n - M + \frac{1}{2}\right)\right)}{\pi\left(n - M + \frac{1}{2}\right)}$$

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 one, all the way out to

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

Even Length Filters using Delay



Even Length Filters using Delay

Apply the time delay property:

$$z[n] = l_I[n-(M-0.5)] \quad \leftrightarrow \quad Z(\omega) = e^{-j\omega(M-0.5)}L_I(\omega),$$

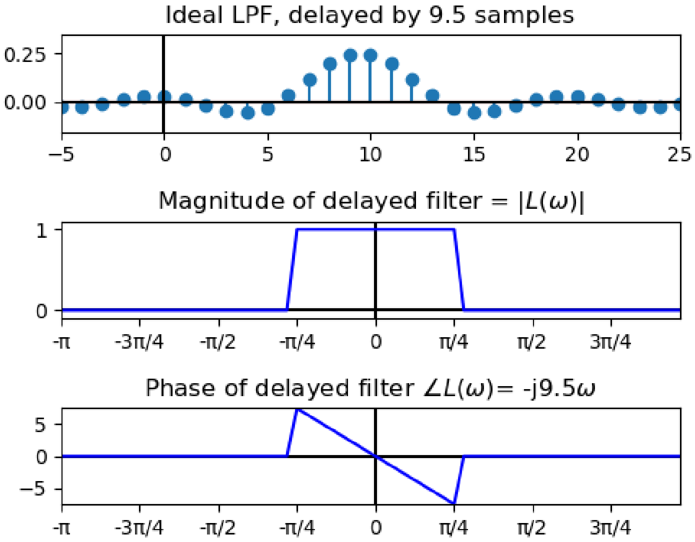
and then notice that

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

So

$$|Z(\omega)| = |L_I(\omega)|$$

Even Length Filters using Delay



Even Length Filters using Delay and Windowing

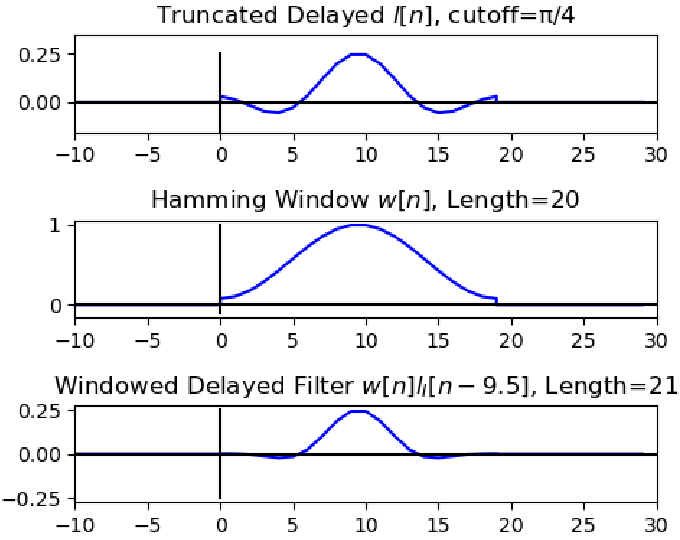
Now we can create an even-length filter by windowing the delayed filter:

$$l[n] = \begin{cases} w[n] l_I\left[n - (M - 0.5)\right] & 0 \le n \le (2M - 1) \\ 0 & \text{otherwise} \end{cases}$$

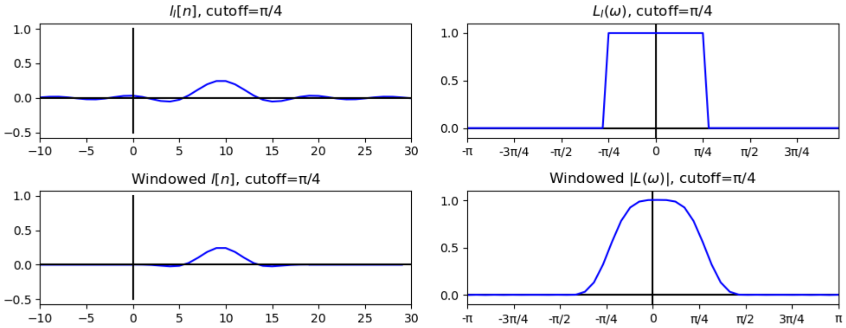
where $w[n]$ is a Hamming window defined for the samples $0 \le m \le 2M - 1$:

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

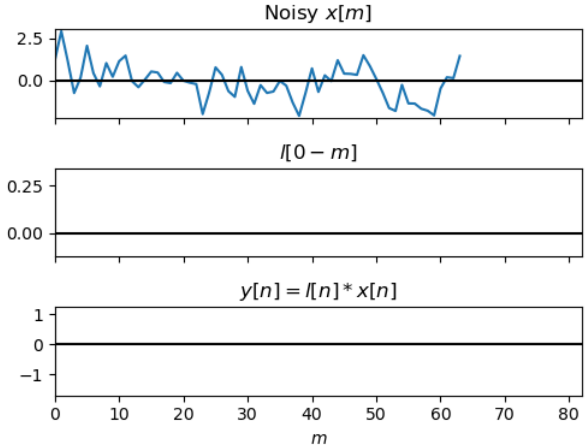
Even Length Filters using Delay and Windowing



Even Length Filters using Delay and Windowing



$l_I[n] = \frac{\sin(\omega_L n)}{\pi n}$



Outline

Summary: Ideal Filters

- Ideal Lowpass Filter:

$$L_I(\omega) = \begin{cases} 1 & |\omega| \le \omega_L, \\ 0 & \omega_L < |\omega| \le \pi. \end{cases} \quad \leftrightarrow \quad l_I[m] = \frac{\sin(\omega_L n)}{\pi n}$$

- Ideal Highpass Filter:

$$H_I(\omega) = 1 - L_I(\omega) \quad \leftrightarrow \quad h_I[n] = \delta[n] - \frac{\sin(\omega_H n)}{\pi n}$$

- Ideal Bandpass Filter:

$$B_I(\omega) = L_I(\omega|\omega_L) - L_I(\omega|\omega_H) \quad \leftrightarrow \quad b_I[n] = \frac{\sin(\omega_L n)}{\pi n} - \frac{\sin(\omega_H n)}{\pi n}$$

Summary: Practical Filters

- Odd Length:

$$h[n] = \begin{cases} h_I[n]w[n] & -M \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

- Even Length:

$$h[n] = \begin{cases} h_I\left[n - (M - 0.5)\right] w[n] & 0 \le n \le 2M - 1 \\ 0 & \text{otherwise} \end{cases}$$

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

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