Chapter 8

Design of Digital Filters

Contents

So far our treatment of DSP has focused primarily on the **analysis** of discrete-time systems. Now we finally have the analytical tools to begin to **design** discrete-time systems. All LTI systems can be thought of as filters, so, at least for LTI systems, to “design” a system means to design a digital filter.

(The design of nonlinear or time-varying systems is generally more complicated, and often more case specific.)

Goal: given desired
- magnitude response $|\mathcal{H}_\mathrm{d}(\omega)|$
- phase response $\angle\mathcal{H}_\mathrm{d}(\omega)$
- tolerance specifications (how far from ideal?),

we want to choose the filter parameters N, M, $\{a_k\}$, $\{b_k\}$ such that the system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-1}}{\sum_{k=0}^{N} a_k z^{-1}} = g\frac{\prod_i (z - z_i)}{\prod_j (z - p_j)},$$

where $a_0 = 1$, yields a frequency response $\mathcal{H}(\omega) \approx \mathcal{H}_\mathrm{d}(\omega)$.

Rational $H(z)$, so LTI system described by a constant-coefficient difference equation, so can be implemented with finite # of adds, multiplies, and delays.

In other words, **filter design** means choosing the number and locations of the zeros and poles, or equivalently the number and values of the filter coefficients, and thus $H(z)$, $h[n]$, $\mathcal{H}(\omega)$.

Overview
- $N = 0$, FIR or all-zero. linear passband phase.
- $N > 0$, IIR. lower sidelobes for same number of coefficients.

8.1 General Considerations
- Ideally would like N and M or $N + M$ to be as small as possible for minimal computation / storage.
- Causal (for now)
- Poles inside unit circle for stability

8.1.1 Causality

We will focus on designing **causal** digital filters, since those can be implemented in real time. Noncausal filter design (*e.g.*, for off-line applications) is much easier and many of the same principles apply anyway.

An LTI system is causal iff
- input/output relationship: $y[n]$ depends only on current and past input signal values.
- impulse response: $h[n] = 0$ for $n < 0$
- system function: number of finite zeros $\leq$ number of finite poles
- frequency response: What can we say about $\mathcal{H}(\omega)$?

Fact: if $h[n]$ is causal, then
- **Paley-Wiener Theorem**: $\mathcal{H}(\omega)$ cannot be exactly zero over any *band* of frequencies. (Except in the trivial case where $h[n] = 0$.)
- Furthermore, $|\mathcal{H}(\omega)|$ cannot be flat (constant) over any finite band.
- $\mathcal{H}_R(\omega)$ and $\mathcal{H}_I(\omega)$ are **Hilbert transform pairs**. Therefore they are not independent. Hence magnitude and phase response are interdependent.

Thus those **ideal** filters with finite bands of zero response cannot be implemented with a causal filter.

Instead, we must design filters that *approximate* the desired frequency response $\mathcal{H}_\mathrm{d}(\omega)$.

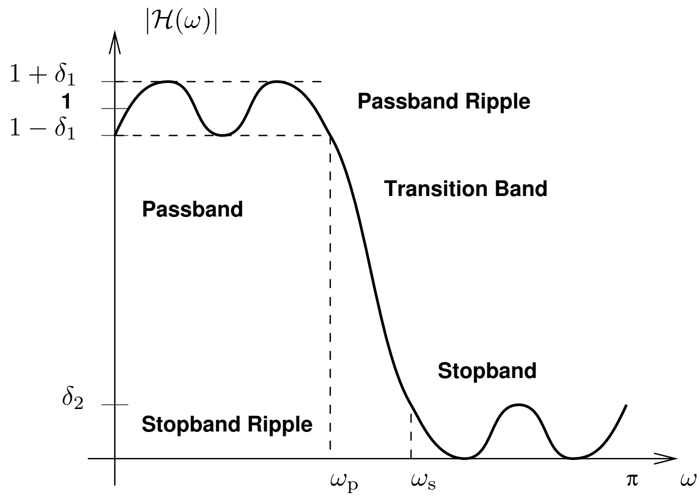
8.1.2 Characteristics of practical frequency-selective filters

No perfectly flat regions

Fact: since causal filters cannot have a band of frequencies with zero response, nor can they have *any* band of frequencies over which the frequency response is a constant.

Proof by contradiction. Suppose $\mathcal{H}(\omega) = c$ for $\omega_1 \leq \omega \leq \omega_2$, with corresponding impulse response $h[n]$. Now define a new filter $g[n] = h[n] - c\,\delta[n]$. Then certainly $g[n]$ is also causal. But $G(\omega) = \mathcal{H}(\omega) - c = 0$ for $\omega_1 \leq \omega \leq \omega_2$, which is impossible if $g[n]$ is causal. □

Causal filters cannot have a band of frequencies with zero response. Nor can they have an infinitely sharp transition between the passband and the stopband. Nor can they have perfectly flat passband. So a typical realistic magnitude response looks like the following.



Practical filter design means choosing δ_1, δ_2, ω_c and ω_s, and *then* designing N, M, $\{a_k\}$, $\{b_k\}$ to satisfy those requirements.

Often one must iterate.

Often we plot $|\mathcal{H}(\omega)|$ using dB, *i.e.*, $20\log_{10}|\mathcal{H}(\omega)|$, and express the ripple in dB as well.

Note that *stopband ripple is not defined peak-to-peak*, since the highest magnitude response in the stopband is more important than how wiggly the response is in the stopband.

Example application: CD digital crossover

Separate woofer and tweeter signals digitally inside CD player, rather in analog at speaker.
Why is passband ripple tolerable in this context (distortion)? Room acoustics act as another filter. Speaker response is not perfectly flat. Keep filter ripple below these other effects.

Now imagine that we have specified δ_1, δ_2, ω_c and ω_s, and we wish to design N, M, $\{a_k\}$, $\{b_k\}$ to satisfy those requirements. As mentioned above, we have two broad choices: FIR and IIR. We focus first on FIR.

8.2
Design of FIR Filters

An FIR filter of length M is an LTI system with the following difference equation[1]:

$$y[n] = \sum_{k=0}^{M-1} b_k \, x[n-k] \, .$$

Note that the book changes the role of M here. Earlier, when discussing rational system functions, M was the number of zeros. Now M is the number of "nonzero" elements of $h[n]$, which corresponds to at most $M-1$ zeros. (More precisely, we assume $b_{M-1} \neq 0$ and $b_0 \neq 0$, but some of the coefficients in between could be zero.)

The problem: given δ_1, δ_2, ω_c, and ω_s, we wish to choose M and $\{b_k\}_{k=0}^{M-1}$ to achieve that goal.

We focus on linear-phase FIR filters, because if linear phase is not needed, then IIR is probably preferable anyway.

We focus on lowpass filters, since transformations can be made to form highpass, bandpass from lowpass, as discussed previously.

Impulse response

Clearly for an FIR filter

$$h[n] = \begin{cases} b_n, & n = 0, \ldots, M-1 \\ 0, & \text{otherwise.} \end{cases}$$

So rather than writing everything in terms of b_k's, we write it directly in terms of the impulse response $h[n]$.

In fact, for FIR filter design we usually design $h[n]$ directly, rather than starting from a pole-zero plot.
(An exception would be notch filters.)

8.2.1 Symmetric and antisymmetric FIR filters

I focus on the symmetric case.

System function:

$$H(z) = \sum_{n=0}^{M-1} h[n] \, z^{-n}.$$

How do we make a filter have **linear phase**?
We previously answered this in the pole-zero domain. Now we examine it in the time domain.

An FIR filter has linear phase if $h[n] = h[M-1-n], \; n = 0, 1, \ldots, M-1.$

Example. For $M = 5$: $h[n] = \{b_0, b_1, b_2, b_1, b_0\}$.

Such an FIR filter is called **symmetric**. Caution: this is not "even symmetry" though in the sense we discussed previously.

This is related, but not exactly the same as circular symmetry.

<u>Example</u>. For $M = 3$ and $h[n] = \left\{\underline{1/2}, 1, 1/2\right\}$. Does this filter have linear phase? Is it lowpass or highpass?

$$\mathcal{H}(\omega) = \frac{1}{2} + e^{-\jmath\omega} + \frac{1}{2} e^{-\jmath 2\omega} = e^{-\jmath\omega} \left[\frac{1}{2} e^{\jmath\omega} + 1 + \frac{1}{2} e^{-\jmath\omega}\right] = e^{-\jmath\omega} \, (1 + \cos\omega),$$

so since $1 + \cos\omega \geq 0$, $\angle \mathcal{H}(\omega) = -\omega$, which is linear phase.

So it works for this particular example, but why does the symmetry condition ensure linear phase in general?

[1] Caution. At this point the book switches from $\sum_{k=0}^{M}$ to $\sum_{k=0}^{M-1}$ apparently. This inconsistent with MATLAB, so there are "$M-1$" factors that appear frequently in the MATLAB calls. I think that MATLAB is consistent and the book makes an undesirable switch of convention here.

Symmetric real FIR filters, $h[n] = h[M-1-n],\ n = 0, \ldots, M-1$, are linear phase.

Proof. Suppose M is even:

$$\begin{aligned}
H(z) &= \sum_{n=0}^{M-1} h[n]\, z^{-n} \\
&= \sum_{n=0}^{M/2-1} h[n]\, z^{-n} + \sum_{n=M/2}^{M-1} h[n]\, z^{-n} && \text{(split sum)} \\
&= \sum_{n=0}^{M/2-1} h[n]\, z^{-n} + \sum_{n'=0}^{M/2-1} h[M-1-n']\, z^{-(M-1-n')} && (n' = M-1-n) \\
&= \sum_{n=0}^{M/2-1} h[n]\, z^{-n} + \sum_{n=0}^{M/2-1} h[n]\, z^{-(M-1-n)} && \text{(symmetry of } h[n]) \\
&= \sum_{n=0}^{M/2-1} h[n] \left[z^{-n} + z^{-(M-1-n)}\right] && \text{(combine)} \\
&= z^{-(M-1)/2} \sum_{n=0}^{M/2-1} h[n] \left[z^{(M-1)/2-n} + z^{-((M-1)/2-n)}\right] && \text{(split phase).}
\end{aligned}$$

Thus the frequency response is

$$\begin{aligned}
\mathcal{H}(\omega) &= H(z)\Big|_{z=e^{\jmath\omega}} \\
&= e^{-\jmath\omega(M-1)/2} \sum_{n=0}^{M/2-1} h[n] \left[e^{\jmath\omega((M-1)/2-n)} + e^{-\jmath\omega((M-1)/2-n)}\right] \\
&= 2\, e^{-\jmath\omega(M-1)/2} \sum_{n=0}^{M/2-1} h[n] \cos\left[\omega\left(\frac{M-1}{2} - n\right)\right] \\
&= \boxed{e^{-\jmath\omega(M-1)/2}\, \mathcal{H}_r(\omega)} \\
\mathcal{H}_r(\omega) &= 2 \sum_{n=0}^{M/2-1} h[n] \cos\left[\omega\left(\frac{M-1}{2} - n\right)\right]. \text{ (Real since } h[n] \text{ is real.)}
\end{aligned}$$

Phase response:

$$\angle\, \mathcal{H}(\omega) = \begin{cases} -\omega\frac{M-1}{2}, & \mathcal{H}_r(\omega) > 0 \\ -\omega\frac{M-1}{2} + \pi, & \mathcal{H}_r(\omega) < 0. \end{cases}$$

The case for M odd is similar, and leads to the same phase response but with a slightly different $\mathcal{H}_r(\omega)$.

In fact, the odd M case is even easier by noting that $h[n + (M-1)/2]$ is an even function, so its DTFT is real, so the DTFT of $h[n]$ is $\mathrm{e}^{-\jmath\omega(M-1)/2}$ times a real function. This proof does not work for M even since then $(M-1)/2$ is not an integer so we cannot use the shift property.

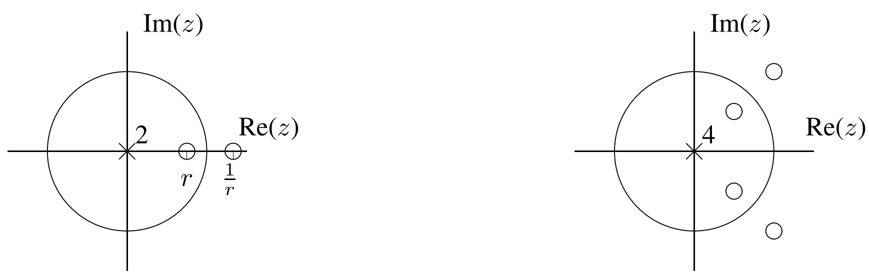
What about pole-zero plot? (We want to be able to recognize FIR linear-phase filters from pole-zero plot.)

From above, $H(z) = z^{-(M-1)/2} \sum_{n=0}^{M/2-1} h[n] \left[z^{(M-1)/2-n} + z^{-((M-1)/2-n)}\right]$ so
$H(z^{-1}) = z^{(M-1)/2} \sum_{n=0}^{M/2-1} h[n] \left[z^{-(M-1)/2-n} + z^{((M-1)/2-n)}\right] = z^{M-1} H(z)$. Thus

$$H(z) = z^{-(M-1)} H(z^{-1}) .$$

So if q is a zero of $H(z)$, then $1/q$ is also a zero of $H(z)$. Furthermore, in the usual case where $h[n]$ is real, if q is a zero of $H(z)$, then so is q^*.

Example. Here are two pole-zero plots of such linear-phase filters.



What is difference between this and all-pass filter? It was poles and zeros in reciprocal relationships for all-pass filter.

Now we know conditions for FIR filters to be linear phase. How do we design one?

Delays

In continuous time, the delay property of the Laplace transform is

$$x_\mathrm{a}(t-\tau) \overset{\mathcal{L}}{\leftrightarrow} \mathrm{e}^{-s\tau} X_\mathrm{a}(s) .$$

How do we build a circuit that delays a signal? Since $\mathrm{e}^{-s\tau}$ is not a rational function, so it cannot be implemented exactly using RLC circuits. So even though a time delay system is an LTI system, we cannot build it using RLC components!

We can make an approximation, *e.g.*,

$$\mathrm{e}^{-s\tau} \approx 1 - s\tau + \frac{1}{2}s^2\tau^2$$

which *is* rational in s, so we can design such an RLC circuit.

Or, we can use more mechanical approaches to delay like a tape loop. ***Picture*** . write, read, erase head. delay $\propto$ 1 / tape velocity.

Another approach is to rely on signal propagation time down a long wire, and **tap** into the wire at various places for various delays.

What about in discrete time? We just need a digital latch or buffer (flip flops) to hold the bits representing a digital signal value until the next time point.

8.2.2

Design of linear-phase FIR filters using windows

Perhaps the simplest approach to FIR filter design is to take the ideal impulse response $h_d[n]$ and **truncate** it, which means multiplying it by a rectangular window, or more generally, to multiply $h_d[n]$ by some other window function, where

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{H}_\mathrm{d}(\omega)\, \mathrm{e}^{\jmath\omega n} \,\mathrm{d}\omega \,.$$

Typically $h_d[n]$ will be noncausal or at least non-FIR.

Example. As shown previously, if $\mathcal{H}_\mathrm{d}(\omega) = \begin{cases} 1, & |\omega| \le \omega_\mathrm{c}, \\ 0, & \text{otherwise}, \end{cases} = \mathrm{rect}\left(\frac{\omega}{2\omega_\mathrm{c}}\right)$ then $h_d[n] = \frac{\omega_\mathrm{c}}{\pi} \,\mathrm{sinc}\left(\frac{\omega_\mathrm{c}}{\pi} n\right)$.

We can create an FIR filter by **windowing** the ideal response:

$$h[n] = w[n]\, h_d[n] = \begin{cases} h_d[n]\, w[n], & n = 0, \ldots, M-1 \\ 0, & \text{otherwise}, \end{cases}$$

where the **window function** $w[n]$ is nonzero only for $n = 0, \ldots, M-1$.

What is the effect on the frequency response?

$$\mathcal{W}(\omega) = \sum_{n=0}^{M-1} w[n]\, \mathrm{e}^{-\jmath\omega n}$$

and by **time-domain multiplication property** of DTFT, aka the **windowing theorem**:

$$\mathcal{H}(\omega) = \mathcal{W}(\omega) \circledast_{2\pi} \mathcal{H}_\mathrm{d}(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{H}_\mathrm{d}(\lambda)\, \mathcal{W}(\omega - \lambda) \,\mathrm{d}\lambda, \tag{8-1}$$

where $\circledast_{2\pi}$ denotes 2π-**periodic convolution**.

In words, the ideal frequency response $\mathcal{H}_\mathrm{d}(\omega)$ is *smeared out* by the frequency response $\mathcal{W}(\omega)$ of the window function.

What would the frequency response of the "ideal" window be? $\mathcal{W}(\omega) =$ "$2\pi\,\delta(\omega)$" $= \sum_{k=-\infty}^{\infty} 2\pi\,\delta(\omega - 2\pi k)$, a Dirac impulse. Such a "window" would cause no smearing of the ideal frequency response.

However, the corresponding "window" function would be $w[n] = 1$, which is noncausal and non-FIR.

So in practice we must make tradeoffs.

Example. **rectangular window**. $w[n] = \begin{cases} 1, & n = 0, \ldots, 2 \\ 0, & \text{otherwise}, \end{cases}$ with $\mathcal{H}_\mathrm{d}(\omega) = \mathrm{e}^{-\jmath\omega}\, 2\,\mathrm{rect}\left(\frac{\omega}{\pi}\right)$, *i.e.*, $\omega_\mathrm{c} = \pi/2$.

So by the shift property of the DTFT: $h_d[n] = \mathrm{sinc}\left(\frac{1}{2}(n-1)\right)$ and $h[n] = \left\{\underline{\mathrm{sinc}\left(\frac{1}{2}\right)}, 1, \mathrm{sinc}\left(\frac{1}{2}\right)\right\}$ where $\mathrm{sinc}\left(\frac{1}{2}\right) = 2/\pi \approx 0.64$.

So the resulting frequency response is $\mathcal{H}(\omega) = \mathrm{e}^{-\jmath\omega}\left[1 + \frac{4}{\pi}\cos(\omega)\right]$. ***Picture***

This is only a 3-tap design. Let us generalize next.

Example. **rectangular window**.

$$w[n] = \begin{cases} 1, & n = 0, \ldots, M-1 \\ 0, & \text{otherwise,} \end{cases}$$

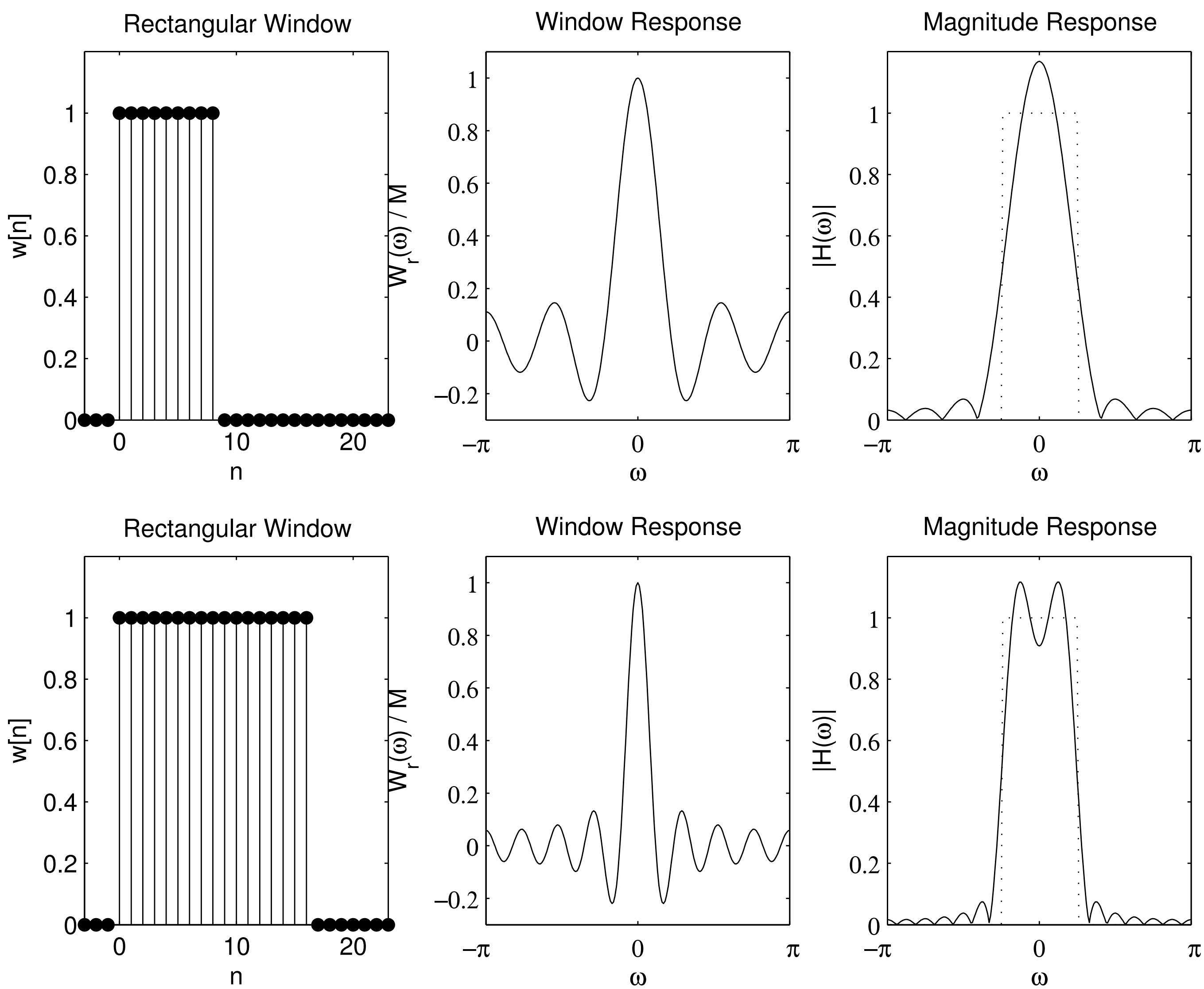
$$\mathcal{W}(\omega) = \sum_{n=0}^{M-1} \mathrm{e}^{-\jmath\omega n} = \cdots = \mathrm{e}^{-\jmath\omega(M-1)/2}\, \mathcal{W}_r(\omega), \text{ where } \mathcal{W}_r(\omega) = \begin{cases} \dfrac{\sin(\omega M/2)}{\sin(\omega/2)}, & \omega \neq 0 \\ M, & \omega = 0. \end{cases}$$

In this case the ideal frequency response is smeared out by a sinc-like function, because $\mathcal{W}_r(\omega) \approx M \operatorname{sinc}\left(M \frac{\omega}{2\pi}\right)$.

The function $\frac{\sin(\omega M/2)}{M \sin(\omega/2)}$ is available in MATLAB as the `diric` function: the **Dirichlet** or **periodic sinc** function.

How much is the ideal frequency response smeared out?
- The width of the main lobe of $\mathcal{W}(\omega)$ is $\approx 4\pi/M$, because the first zeros of $\mathcal{W}_r(\omega)$ are at $\omega = \pm 2\pi/M$.
- As M increases, width of main lobe $\mathcal{W}(\omega)$ decreases, so narrower transition band.

Example. Here is the case $\omega_c = \pi/4$ for various values of M.



How did I make these figures? Using `H = freqz(h, 1, om)` since `b = h` for causal FIR filters.

Time-delay in desired response

The term $e^{-\jmath\omega(M-1)/2}$ in $\mathcal{W}(\omega)$ above comes from the fact that the rectangular window is not centered around $n = 0$, but rather is time-shifted to be centered around $n = (M-1)/2$. This phase term will cause additional distortion of $\mathcal{H}_d(\omega)$, unless $\mathcal{H}_d(\omega)$ is also phase-shifted to compensate.

For a lowpass filter with cutoff ω_c, windowed by a length-M window function, the appropriate desired response is:

$$\mathcal{H}_d(\omega) = \begin{cases} e^{-\jmath\omega(M-1)/2}, & |\omega| \le \omega_c, \\ 0, & \text{otherwise}, \end{cases} \implies |\mathcal{H}_d(\omega)| = \begin{cases} 1, & |\omega| \le \omega_c, \\ 0, & \text{otherwise}, \end{cases}$$

so that

$$h_d[n] = \frac{\omega_c}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi}\left[n - \frac{M-1}{2}\right]\right).$$

This is illustrated below. Note that from (8-1),

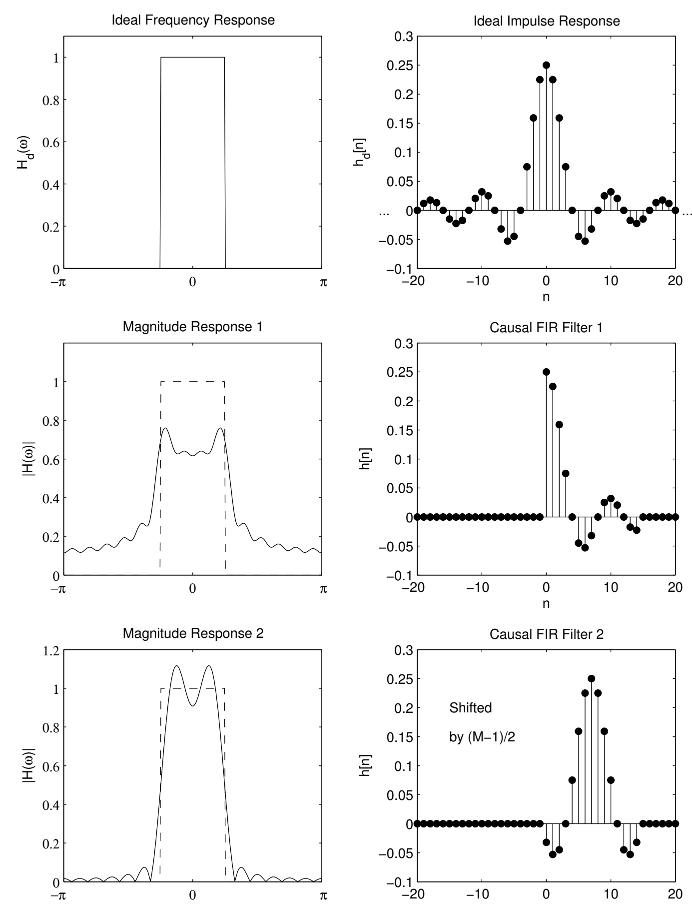
$$\mathcal{H}(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathcal{H}_d(\lambda)\, \mathcal{W}(\omega-\lambda)\, d\lambda = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-\jmath\lambda(M-1)/2}\, e^{-\jmath(\omega-\lambda)(M-1)/2} \frac{\sin((\omega-\lambda)\, M/2)}{\sin((\omega-\lambda)\,/2)}\, d\lambda$$

$$= e^{-\jmath\omega(M-1)/2} \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} \frac{\sin((\omega-\lambda)\, M/2)}{\sin((\omega-\lambda)\,/2)}\, d\lambda.$$

Thus there is an overall delay of $(M-1)/2$ samples from such a length-M causal FIR filter.

Without the phase term, the convolution integral is severely affected

Audio application, $F_s = 44kHz$ and say $M = 45$. The delay in samples is $(M-1)/2 = 22$. The time delay is $\frac{M-1}{2}T$ = 22/44kHz = 0.5msec. The speed of sound in air is about 330meters / second, so 3.3 meters away takes about 10msec. Thus a 0.5msec delay is well within the tolerable range for audio.



Sidelobes

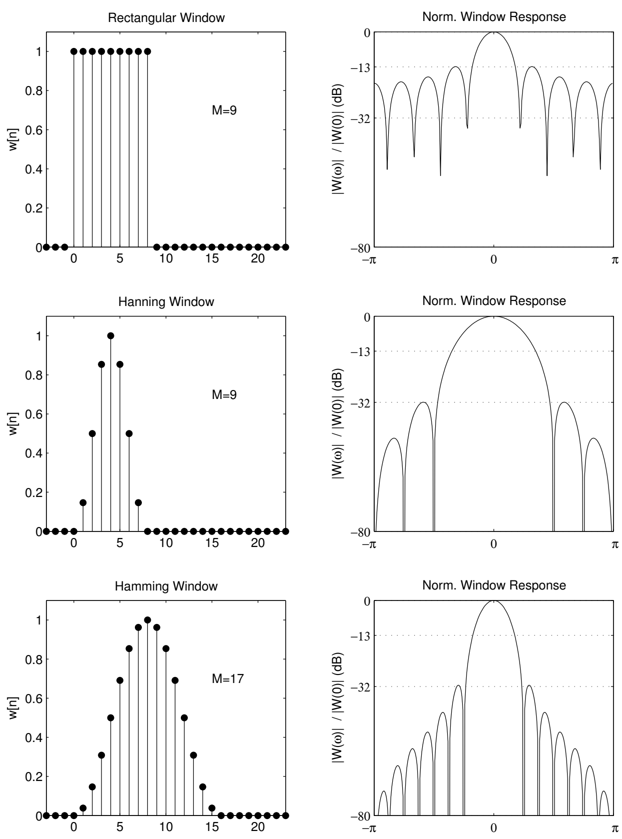
The rectangular window
- has high sidelobes in $\mathcal{W}(\omega)$, and
- the sidelobe amplitude is relatively unaffected by M.
- Sidelobes cause large passband ripple, related to **Gibbs phenomena**.
- Sidelobes caused by abrupt discontinuity at edge of window.

Solution: use some other window function.
- Examples: Bartlett, Blackman, Hanning, Hamming, Kaiser, Lanczos, Tukey. MATLAB's `window` function has 16 choices!
- All have lower sidelobes than rectangular, hence less passband ripple.
- Tradeoff? Wider transition band for the same M compared to rectangular window.

Example. **Hanning window**:

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

What is the frequency response $\mathcal{W}(\omega)$? Somewhat messy.
How do we plot $\mathcal{W}(\omega)$? `W = freqz(w, 1, om)` which in turn uses the DFT/FFT with $N \gg M$ via zero padding. Caution: this is the `hann` function in MATLAB, not the `hanning` function!



- As M increases, main lobe width decreases.
- As M increases, some of the sidelobe amplitudes decrease, but peak sidelobe amplitude remains approximately constant.
- Again, the "price" is need larger M than for rectangular window for same transition width.
- The various windows tradeoff main lobe width with peak sidelobe amplitude.
- Hanning main lobe width is $8\pi/M$, but peak sidelobe is -32dB compared with -13dB for rectangular.

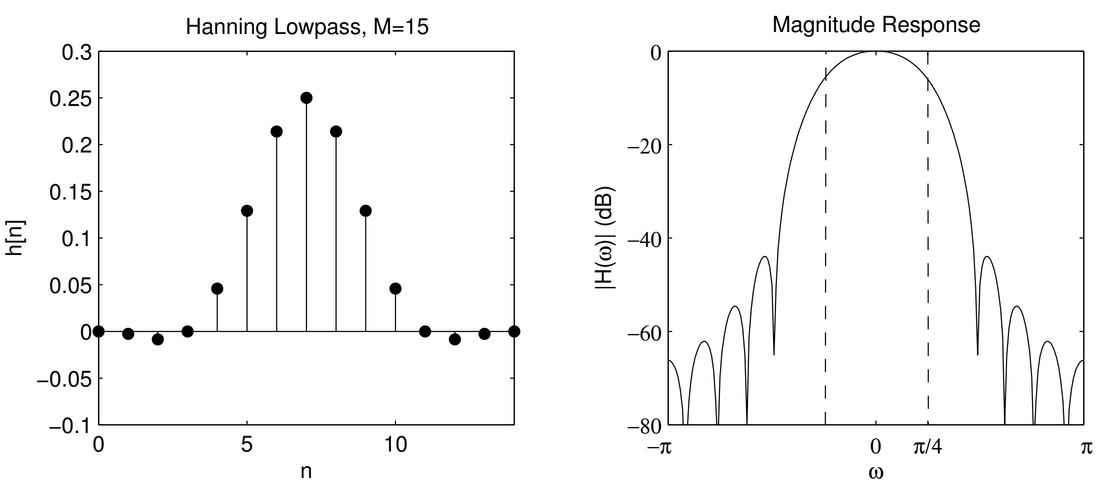
Example. Suppose we want an FIR filter design of a lowpass filter with $\mathcal{H}_{\mathrm{d}}(\omega) = \begin{cases} 1, & |\omega| < \pi/4 \\ 0, & \text{otherwise,} \end{cases}$ designed using the Hanning window with $M = 15$.

Then the desired impulse response is:

$$h_d[n] = \frac{1}{4}\,\mathrm{sinc}\left(\frac{1}{4}(n - (M-1)/2)\right).$$

In MATLAB, here is how we compute and display the impulse response and the frequency response.

```
% fig_win_example1.m

M = 15;
n = [0:(M-1)];
om = linspace(-pi, pi, 201); % for displaying frequency response
oc = pi/4; % cutoff frequency
% desired impulse response:
hd = inline('oc/pi * sinc(oc/pi*(n-(M-1)/2))', 'n', 'oc', 'M');
Hd = inline('1*(abs(om) < oc)', 'om', 'oc');

hn = hd(n, oc, M) .* hann(M)'; % Hanning window applied to ideal impulse resp.

clf, subplot(321)
stem(n, hn, 'filled'), stem_fix
axis([0 M-1 -0.1, 0.3]), xlabel 'n', ylabel 'h[n]'
title(sprintf('Hanning Lowpass, M=%d', M))

subplot(322)
H = freqz(hn, 1, om);
plot(om, 20*log10(abs(H)), '-', om, 20*log10(max(Hd(om,oc),eps)), '--')
xlabel '\omega', ylabel '|H(\omega)| (dB)'
title 'Magnitude Response'
axisy(-80, 0), xaxis_pi '-p 0 p/4 p'

% savefig 'fig_win_example1'
```



If we are unsatisfied with the width of the transition band, or the sidelobe amplitude, or the passband ripple, then what could we do? Increase M, try other window functions, and/or try other filter design methods.

Filter design by windowing in MATLAB

The `fir1` command in MATLAB is its tool for window-based FIR filter design. In the preceding example, we could have simply typed `h = fir1(14, 0.25, hann(15), 'noscale')` to get exactly the same design.

Example. Digital **phaser** or **flange**.

Time varying pole locations in cascade of 1st and 2nd-order allpass filters, Sum output of allpass cascade with original signal, creating time-varying notches.

8.2.5 FIR differentiator

$\mathcal{H}_\mathrm{d}(\omega) = \jmath\omega$

skim

8.2.6 Hilbert transform (90° phase shift)

$\mathcal{H}_\mathrm{d}(\omega) = -\jmath \,\mathrm{sgn}(\omega)$

skip

8.2.3 Design of FIR filters by frequency sampling

For the **frequency sampling** method of FIR filter design, to design a M-point FIR filter we specify the desired frequency response at a set of equally-spaced frequency locations:

$$\mathcal{H}_\mathrm{d}(\omega)\Big|_{\omega=\frac{2\pi}{M}k}, \quad k = 0, \ldots, M-1.$$

In other words, we provide equally spaced samples over $[0, 2\pi)$. ***Picture***

Recall from the DTFT formula that if $h_d[n]$ is nonzero only for $n = 0, \ldots, M-1$, then

$$\mathcal{H}_\mathrm{d}(\omega) = \sum_{n=0}^{M-1} h_d[n]\,\mathrm{e}^{-\jmath\omega n}.$$

Thus, at the given frequency locations, we have

$$\mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}k\right) = \sum_{n=0}^{M-1} h_d[n]\,\mathrm{e}^{-\jmath\frac{2\pi}{M}kn}, \qquad k = 0, \ldots, M-1.$$

This is the formula for the M-point DFT discussed in *Ch. 5* (and in EECS 206).
So we can determine $h_d[n]$ from $\left\{\mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}k\right)\right\}_{k=0}^{M-1}$ by using the **inverse DFT** formula (or `h = ifft(H)` in MATLAB):

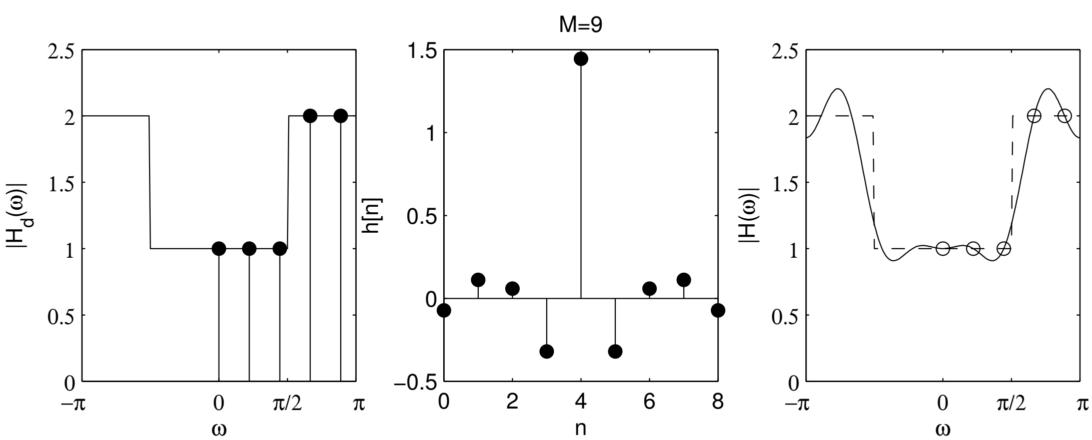
$$h_d[n] = \frac{1}{M}\sum_{k=0}^{M-1} \mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}k\right)\mathrm{e}^{\jmath\frac{2\pi}{M}kn}.$$

This will be the impulse response of the FIR filter as designed by the frequency sampling method.

- If we want $h_d[n]$ to be real, then $\mathcal{H}_\mathrm{d}(\omega)$ must be Hermitian symmetric, *i.e.*, $\mathcal{H}_\mathrm{d}^*(\omega) = \mathcal{H}_\mathrm{d}(-\omega) = \mathcal{H}_\mathrm{d}(2\pi-\omega)$. So if we specify $\mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}k\right)$ to be some value, we know that $\mathcal{H}_\mathrm{d}^*\left(\frac{2\pi}{M}k\right) = \mathcal{H}_\mathrm{d}\left(2\pi - \frac{2\pi}{M}k\right) = \mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}(M-k)\right)$, so $\mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}(M-k)\right)$ is also specified.
- Thus, software such as MATLAB's `fir2` command only requires $\mathcal{H}_\mathrm{d}(\omega)$ only on the interval $[0, \pi]$.
- If $h_d[n]$ is to be real, then it also follows that $\mathcal{H}_\mathrm{d}(\pi) = \mathcal{H}_\mathrm{d}\left(\frac{2\pi}{M}\frac{M}{2}\right)$ must be real valued when M is even.
- If you choose $\mathcal{H}_\mathrm{d}(\omega)$ to be linear phase, then the designed $h[n]$ will be **linear phase**.
  But you are not required to choose $\mathcal{H}_\mathrm{d}(\omega)$ to be linear phase!
- The book discusses many further details, but the above big picture is sufficient for this class.

Example. The treble boost revisited. Double the amplitude of all frequency components above $\omega_c = \pi/2$.

```
% fig_freq_sample1.m

Hd = inline('exp(-1i*om*(M-1)/2) .* (1 + (abs(om) > pi/2))', 'om', 'M');

M = 9;
ok = [0:(M-1)]/M * 2*pi;
Hk = Hd(mod(ok+pi,2*pi)-pi, M); % trick: [-pi,pi] specification of H(\omega)

h = ifft(Hk);
h = reale(h, 'warn');
% h = fir2(M-1, [0 0.5 0.5 1], [1 1 2 2], boxcar(M) );

om = linspace(-pi,pi,201);
clf, pl = 230;
subplot(pl+1), plot(om, abs(Hd(om,M)))
hold on, stem(ok(ok >= 0), abs(Hk(ok >= 0)), 'filled'), stem_fix, hold off
xlabel '\omega', ylabel '|H_d(\omega)|'
axisy([0 2.5]), xaxis_pi '-p 0 p/2 p'

subplot(pl+2), stem(0:(M-1), h, 'filled'), stem_fix, title(sprintf('M=%d',M))
xlabel 'n', ylabel 'h[n]', axis([0 M-1 -0.5 1.5])

H = freqz(h, 1, om);
subplot(pl+3), plot(om, abs(H), '-', om, abs(Hd(om,M)), '--', ok, abs(Hk), 'o')
xlabel '\omega', ylabel '|H(\omega)|', axisy([0 2.5]), xaxis_pi '-p 0 p/2 p'

% savefig fig_freq_sample1
```



In MATLAB, use `h = fir2(M-1, [0 0.5 0.5 1], [1 1 2 2], boxcar(M) )` to design (almost) the above filter.

8.2.4

"Optimum" equiripple linear-phase FIR filters

The window method has a minor disadvantage, that it is difficult to precisely specify ω_p and ω_s, since these two result from the smearing. All we really specify is ω_c, the cutoff.

An "ideal" linear-phase design procedure would be as follows.

> Specify ω_p, ω_s, δ_1, δ_2, and run an algorithm that returns the *minimum* M that achieves that design goal, as well as the impulse response $h[n],\ n = 0, \ldots, M-1$, where, for linear phase, $h[n] = h[M-1-n]$.

To my knowledge, there is no such procedure that is guaranteed to do this perfectly. However, we can come close using the following iterative procedure.
- Choose M, and find the linear-phase $h[n]$ whose frequency response is as "close" to $\mathcal{H}_\mathrm{d}(\omega)$ as possible.
- If it is not close enough, then increase M and repeat.

How can we measure "closeness" of two frequency response functions? ***Pictures of*** $\mathcal{H}_\mathrm{d}(\omega)$ ***and*** $\mathcal{H}(\omega)$.
Possible options include the following.
- $\frac{1}{2\pi}\int_{-\pi}^{\pi} |\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)| \,\mathrm{d}\omega$, average absolute error
- $\frac{1}{2\pi}\int_{-\pi}^{\pi} |\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)|^2 \,\mathrm{d}\omega$, average squared error
- $\frac{1}{2\pi}\int_{-\pi}^{\pi} |\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)|^2 \,\mathcal{W}(\omega)\,\mathrm{d}\omega$, average weighted squared error
- $\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)|^p \,\mathcal{W}(\omega)\,\mathrm{d}\omega\right]^{1/p}$, weighted L_p error, $p \geq 1$
- $\max_\omega |\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)|$, maximum error ($p = \infty$)
- $\max_\omega |\mathcal{W}(\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega))|$, maximum weighted error ($p = \infty$)

In this section, we focus on the last choice, the maximum weighted error between the desired response and the actual frequency response of an FIR filter. We want to find the FIR filter that minimizes this error.
How to find this? Not by brute force search or trial and error, but by analysis!

If $\mathcal{W}(\omega) \geq 0$ then

$$\mathcal{E}(\omega) \triangleq |\mathcal{W}(\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega))| = \mathcal{W}(\omega)\,|\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)|\,.$$

Consider the case of a lowpass filter. Note that

$$\mathcal{H}_\mathrm{d}(\omega) = \begin{cases} \mathrm{e}^{-\jmath\omega(M-1)/2}, & |\omega| \leq \omega_\mathrm{c}, \\ 0, & |\omega| > \omega_\mathrm{c} \end{cases} = \mathrm{e}^{-\jmath\omega(M-1)/2}\,\mathcal{H}_{dr}(\omega) \quad \text{where} \quad \mathcal{H}_{dr}(\omega) = \begin{cases} 1, & |\omega| \leq \omega_\mathrm{c}, \\ 0, & |\omega| > \omega_\mathrm{c}. \end{cases}$$

Also recall that for M even and $h[n]$ linear phase and symmetric,

$$\mathcal{H}(\omega) = \mathrm{e}^{-\jmath\omega(M-1)/2}\,\mathcal{H}_r(\omega), \quad \text{where} \quad \mathcal{H}_r(\omega) = \sum_{n=0}^{M/2-1} h[n]\, 2\cos\left(\omega\left(\frac{M-1}{2} - n\right)\right).$$

So $\mathcal{H}_r(\omega)$ is the sum of $M/2$ cosines.

The error can be simplified as follows:

$$\begin{aligned} \mathcal{E}(\omega) &= \mathcal{W}(\omega)\,|\mathcal{H}_\mathrm{d}(\omega) - \mathcal{H}(\omega)| \\ &= \mathcal{W}(\omega)\left|\mathrm{e}^{-\jmath\omega(M-1)/2}\,\mathcal{H}_{dr}(\omega) - \mathrm{e}^{-\jmath\omega(M-1)/2}\,\mathcal{H}_r(\omega)\right| \\ &= \mathcal{W}(\omega)\,|\mathcal{H}_{dr}(\omega) - \mathcal{H}_r(\omega)|\,. \end{aligned}$$

Thus we only need to consider the "real" parts of the desired vs actual frequency response.

The logical approach to specifying the error weighting function $\mathcal{W}(\omega)$ is as follows:

$$\mathcal{W}(\omega) = \begin{cases} 1/\delta_1, & 0 \le \omega \le \omega_{\mathrm{c}} & \text{pass band} \\ 0, & \omega_{\mathrm{c}} < \omega < \omega_{\mathrm{s}} & \text{transition band} \\ 1/\delta_2, & \omega_{\mathrm{s}} < \omega < \pi & \text{stop band.} \end{cases}$$

However, the effect of the weighting will be the same if we multiply $\mathcal{W}(\omega)$ by a constant, such as δ_2. So the following weighting function has the same effect:

$$\mathcal{W}(\omega) = \begin{cases} \delta_2/\delta_1, & 0 \le \omega \le \omega_{\mathrm{c}} \\ 0, & \omega_{\mathrm{c}} < \omega < \omega_{\mathrm{s}} \\ 1, & \omega_{\mathrm{s}} < \omega < \pi. \end{cases}$$

A small δ_2 value means we want very low sidelobes, and are willing to sacrifice uniformity over the passband, so less weight is given to the errors in the passband.

As a designer you choose $\mathcal{H}_{dr}(\omega)$, the filter length M, the ratio δ_2/δ_1, and the passband and stopbands, *i.e.*, ω_{c} and ω_{s}. The procedure finds the impulse response that minimizes the maximum weighted error between the desired response and the actual response.

Mathematically, the min-max or Chebyshev design uses the following criterion:

$$\min_{\{h[n]\}_{n=0}^{M/2-1}} \max_{\omega} \mathcal{E}(\omega) \,.$$

Using the **alternation theorem**, from the theory of Chebyshev approximation, Parks and McClellan in 1972 showed that a necessary and sufficient condition for a M-tap filter to be optimal (in the maximum weighted error sense) is that the error $\mathcal{E}(\omega)$ must reach its maximum at least $M/2+1$ point over the intervals where $\mathcal{W}(\omega) \neq 0$.

This theorem guarantees that there is a unique optimal filter. It also tells us that the error will alternate back and forth, *i.e.*, there will be *ripples* in passband and stopband. The resulting filters are called **equiripple** because all ripples in passband have the same peak-to-peak amplitude, and likewise for the stop band.

The actual procedure for finding the best filter is iterative, and it called the **Remez exchange algorithm**. It is implemented by the `remez` function in Matlab. The algorithm first guesses where the extremal frequencies are, and then computes $\mathcal{H}_r(\omega)$ from that, then finds new estimates of the extremal frequencies and iterates.

- Increasing M reduces passband ripple and increases stopband attenuation
- MATLAB's `remezord` command gives approximation to required M, based on *(8.2.95)*:

$$\hat{M} = \frac{D_\infty(\delta_1, \delta_2)}{\Delta f} - f(\delta_1, \delta_2)\Delta f + 1,$$

  where $\Delta f = (\omega_{\mathrm{s}} - \omega_{\mathrm{p}})/(2\pi)$ and D_∞ and $f(\delta_1, \delta_2)$ are defined in text (8.2.96-97).
- Another formula is also given in text:

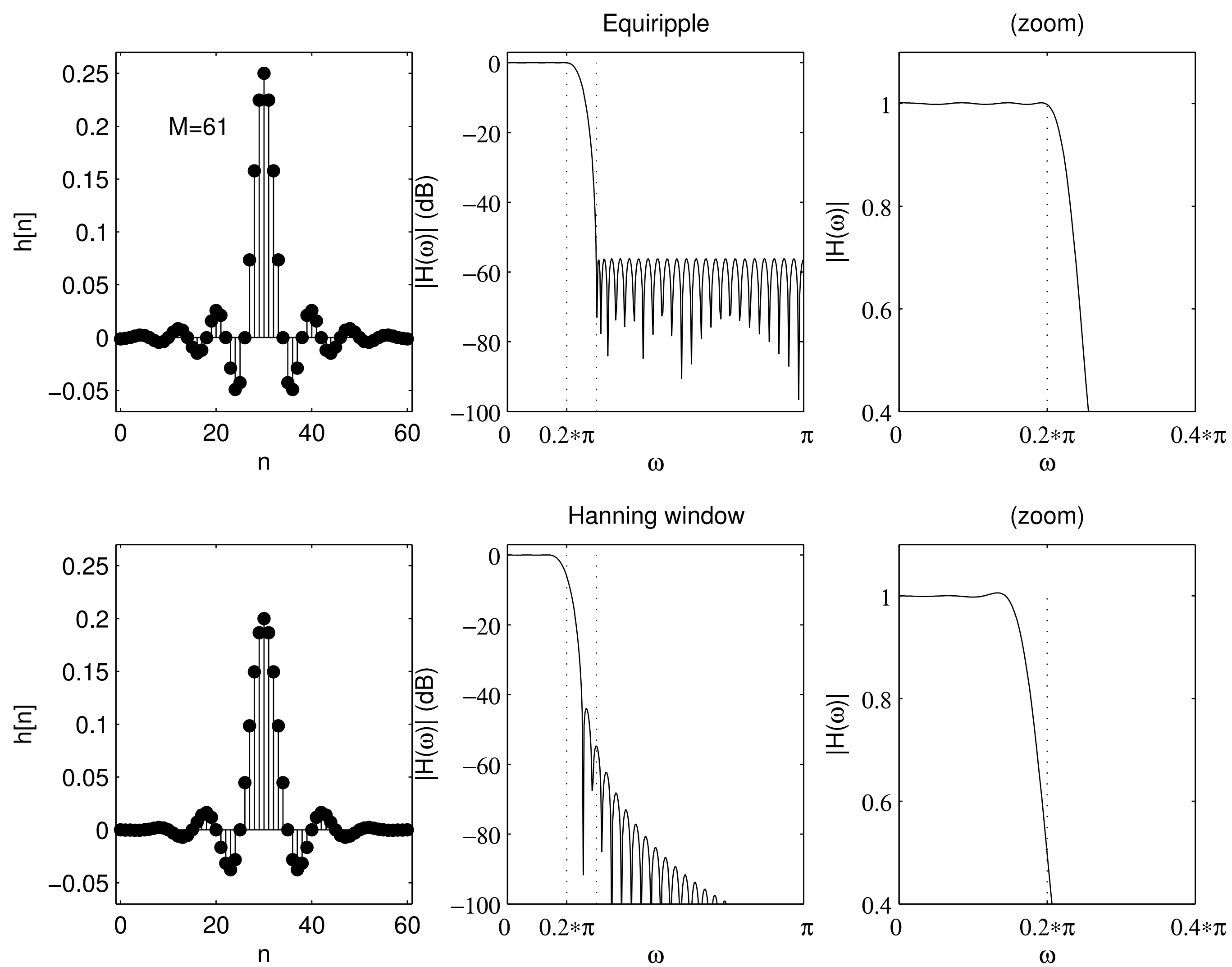
$$\hat{M} = \frac{\log_{10} \dfrac{1}{(\delta_1\delta_2)^{10}} - 13}{14.6\Delta f} + 1.$$

- As transition band Δf decreases, M increases
- As ripples δ_j decrease, M increases
- This can be explored graphically using MATLAB's `fdatool` or `filtdes`.

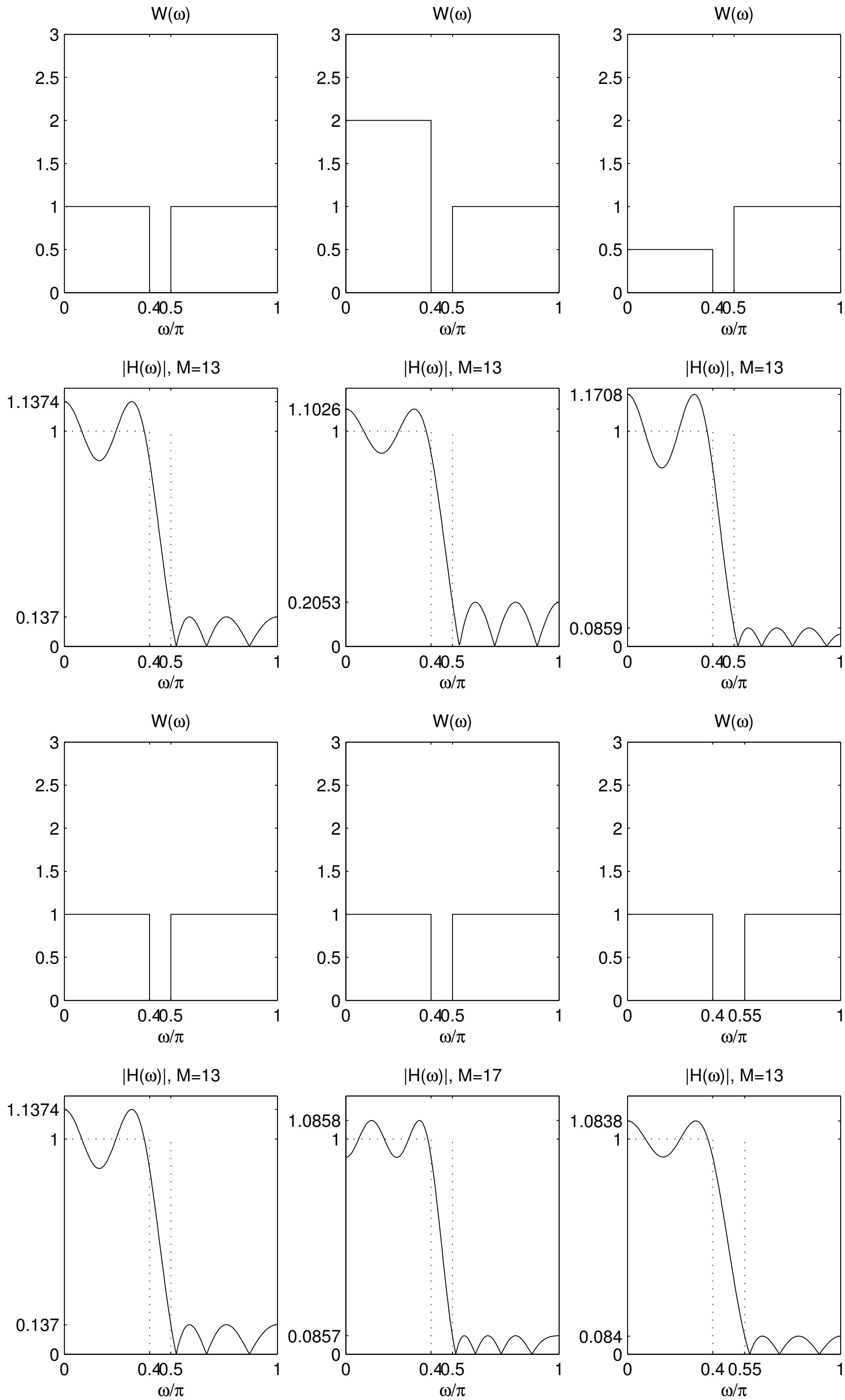
Example. digital crossover for audio

- $F_s = 44$kHz
- $F_{pass} = 4.4$kHz $\omega_p = 2\pi F_{pass}/F_s = 0.2\pi$
- $F_{stop} = 6.6$kHz $\omega_s = 2\pi F_{stop}/F_s = 0.3\pi$
- 3 bands: pass, transition, stop
- $M = 61$ for now; revisit shortly.

```
% fig_remez_example1.m
M = 61;
f = [0 0.2 0.3 1];
Hdr = [1 1 0 0];
W = [1 1];
h = remez(M-1, f, Hdr, W);
stem(0:M-1, h, 'filled')
freqz(h, 1, 201)
```



Further illustrations of the tradeoffs. The error has to go somewhere: passband(s), stopband(s), and/or transition band(s).



8.2.7 Comparison of FIR methods *skim*

8.3

Design of IIR Filters from Analog Filters

Why IIR? With IIR designs we can get the same approximation accuracy (of the magnitude response) as FIR but with a lower order filter. The tradeoff is nonlinear phase.

Analog filter design is a mature field. There are well known methods for selecting RLC combinations to approximate some desired frequency response $H_d(F)$.

Generally, the more (passive) components used, the closer one can approximate $H_d(F)$, (to within component tolerances).

Essentially *all* analog filters are IIR, since the solutions to linear differential equations involve infinite-duration terms of the form $t^m \, \mathrm{e}^{\lambda t} \, u(t)$.

One way to design IIR digital filters is to piggyback on this wealth of design experience for analog filters.

What do analog filters look like?

Any RLC network is describe by a **linear constant coefficient differential equation** of the form

$$\sum_{k=0}^{N} \alpha_k \frac{d^k}{dt^k} \, y_{\mathrm{a}}(t) = \sum_{k=0}^{M} \beta_k \frac{d^k}{dt^k} \, x_{\mathrm{a}}(t)$$

with corresponding **system function** (Laplace transform of the impulse response)

$$H_{\mathrm{a}}(s) = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k}.$$

Each combination of $N, M, \{\alpha_k\}, \{\beta_k\}$ corresponds to some arrangement of RLCs. Those implementation details are unimportant to us here.

The frequency response of a general analog filter is

$$H_{\mathrm{a}}(F) = H_{\mathrm{a}}(s) \Big|_{s=\jmath 2\pi F}.$$

Overview

- Design $N, M, \{\alpha_k\}, \{\beta_k\}$ using existing methods.
- Map from s plane to z-plane somehow to get a_k's and b_k's, *i.e.*, a rational system function corresponding to a discrete-time **linear constant coefficient difference equation**.

Can we achieve linear phase with an IIR filter? We should be able to use our analysis tools to answer this.

Recall that linear phase implies that

$$H(z) = z^{-N} \, H\left(z^{-1}\right),$$

so if z is a pole of the system function $H(z)$, then z^{-1} is also a pole. So any finite poles (*i.e.*, other than 0 or ∞) would lead to instability.

There are no causal stable IIR filters with linear phase.

Thus we design for the magnitude response, and see what phase response we get.

8.3.1 IIR by approximation of derivatives — ***skim***

$$\boxed{s = \frac{1 - z^{-1}}{T}}$$

8.3.2 IIR by impulse invariance — ***skim***

$h[n] = h_{\mathrm{a}}(nT)$

$$\boxed{z = \mathrm{e}^{sT}}$$

8.3.3

IIR filter design by bilinear transformation

Suppose we have used existing analog filter design methods to design an IIR analog filter with system function

$$H_a(s) = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k}.$$

For a sampling period T_s, we now make the **bilinear transformation**

$$\boxed{s = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}}.}$$

This transformation can be motivated by the trapezoidal formula for numerical integration. See text for the derivation.

Define the discrete-time system function

$$\boxed{H(z) = H_a(s)\Big|_{s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}}.}$$

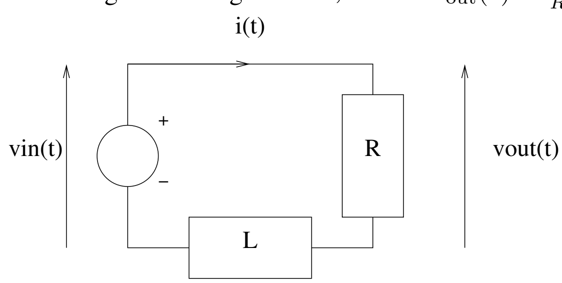
This transformation yields a **rational system function**, *i.e.*, a ratio of polynomials in z.

This $H(z)$ is a system function whose frequency response is related to the frequency response of the analog IIR filter.

Example. Consider a 1st-order analog filter with a single pole at $s = -\alpha$ ***Picture*** where $\alpha > 0$, with system function

$$H_a(s) = \frac{1}{\alpha + s}.$$

How would you build this? Using the following RL "voltage divider," where $V_{out}(s) = \frac{R}{R+sL} V_{in}(s) \Longrightarrow H_a(s) = \frac{R/L}{R/L+s}$.



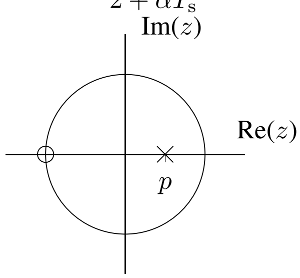
Applying the bilinear transformation to the above (Laplace domain) system function yields:

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}} = \frac{1}{\alpha + \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}} = \frac{1+z^{-1}}{\alpha(1+z^{-1}) + \frac{2}{T_s}(1-z^{-1})} = \frac{1+z^{-1}}{(\alpha + 2/T_s) + (\alpha - 2/T_s)z^{-1}}$$

$$= \frac{1}{\alpha + 2/T_s} \frac{1+z^{-1}}{1 - \frac{2/T_s - \alpha}{2/T_s + \alpha} z^{-1}} = \frac{1}{\alpha + 2/T_s}\frac{1+z^{-1}}{1 - pz^{-1}} = \frac{1}{\alpha + 2/T_s}\left[\frac{-1}{p} + \frac{1 + 1/p}{1 - pz^{-1}}\right],$$

where $p = \frac{2 - \alpha T_s}{2 + \alpha T_s} \in (-1, 1)$.

What is $h[n]$? $h[n] = \frac{1}{\alpha + 2/T_s}\left(-\frac{1}{p}\,\delta[n] + (1 + 1/p)p^n\,u[n]\right)$.



Where did zero at $z = -1$ come from?

Solving bilinear transformation for z in terms of s yields: $\boxed{z = \frac{2 + sT_\mathrm{s}}{2 - sT_\mathrm{s}},}$ so zero at $s = \infty$ maps to $z = -1$.

Note also that pole at $s = -\alpha$ maps to $z = p$.

- In general, the (finite) poles and zeros follow the mapping.
- Poles (or zeros) at $s = \infty$ map to $z = -1$.
- Real poles and zeros remain real, complex-conjugate pairs remain pairs.

What type of filters are $H_\mathrm{a}(s)$ and $H(z)$? Both are poor lowpass filters.
This illustrates the method, but not the power! The utility is for more sophisticated analog IIR designs.

Picture : more s-domain poles and corresponding z-domain poles

The bilinear transformation

More generally, if the system function $H_\mathrm{a}(s)$ is rational, *i.e.*, $H_\mathrm{a}(s) = g\frac{\prod_i (s-s_i)}{\prod_j (s-p_j)}$, then for each term of the form $s - p$ we have

$$s - p \mapsto \frac{2}{T_\mathrm{s}} \frac{1 - z^{-1}}{1 + z^{-1}} - p = \ldots = \frac{2}{T_\mathrm{s}} \left(1 - \frac{pT_\mathrm{s}}{2}\right) \frac{z - \frac{2+pT_\mathrm{s}}{2-pT_\mathrm{s}}}{z + 1},$$

so we will have one root at $z = \frac{2+pT_\mathrm{s}}{2-pT_\mathrm{s}}$ and another one at $z = -1$.

$$z = \frac{2 + sT_\mathrm{s}}{2 - sT_\mathrm{s}}, \qquad s = \frac{2}{T_\mathrm{s}} \frac{1 - z^{-1}}{1 + z^{-1}}.$$

See plot on next page.

- $\mathrm{real}(s) > 0$ maps to z outside unit circle
- $\mathrm{real}(s) < 0$ maps to z inside unit circle
- $s = \jmath 2\pi F$ maps to z on unit circle
$$z = \frac{2 + \jmath 2\pi F T_\mathrm{s}}{2 - \jmath 2\pi F T_\mathrm{s}} = \frac{r\,\mathrm{e}^{\jmath\phi}}{r\,\mathrm{e}^{-\jmath\phi}} = \mathrm{e}^{\jmath 2\phi},$$
where $r = |2 + \jmath 2\pi F T_\mathrm{s}|$ and $\phi = \angle(2 + \jmath 2\pi F T_\mathrm{s}) = \arctan\left(\frac{2\pi F T_\mathrm{s}}{2}\right)$.
Thus $z = \mathrm{e}^{\jmath\omega}$, where $\omega = 2\phi$, so $\boxed{\omega = 2\tan^{-1}\left(\frac{2\pi F T_\mathrm{s}}{2}\right).}$
- Conversely, if $z = \mathrm{e}^{\jmath\omega}$ then
$$s = \frac{2}{T_\mathrm{s}} \frac{1 - \mathrm{e}^{-\jmath\omega}}{1 + \mathrm{e}^{-\jmath\omega}} = \frac{2}{T_\mathrm{s}} \frac{\mathrm{e}^{\jmath\omega/2} - \mathrm{e}^{-\jmath\omega/2}}{\mathrm{e}^{\jmath\omega/2} + \mathrm{e}^{-\jmath\omega/2}} = \frac{2}{T_\mathrm{s}} \frac{\jmath \sin\omega/2}{\cos\omega/2} = \jmath\frac{2}{T_\mathrm{s}} \tan\left(\frac{\omega}{2}\right) \Longrightarrow \boxed{F = \frac{2}{2\pi T_\mathrm{s}} \tan\left(\frac{\omega}{2}\right).}$$
- This is a *nonlinear* relationship between the analog frequency and the digital frequency. It is called **frequency warping**.
See plot on next page.

Because of this nonlinearity, a typical design procedure goes as follows.

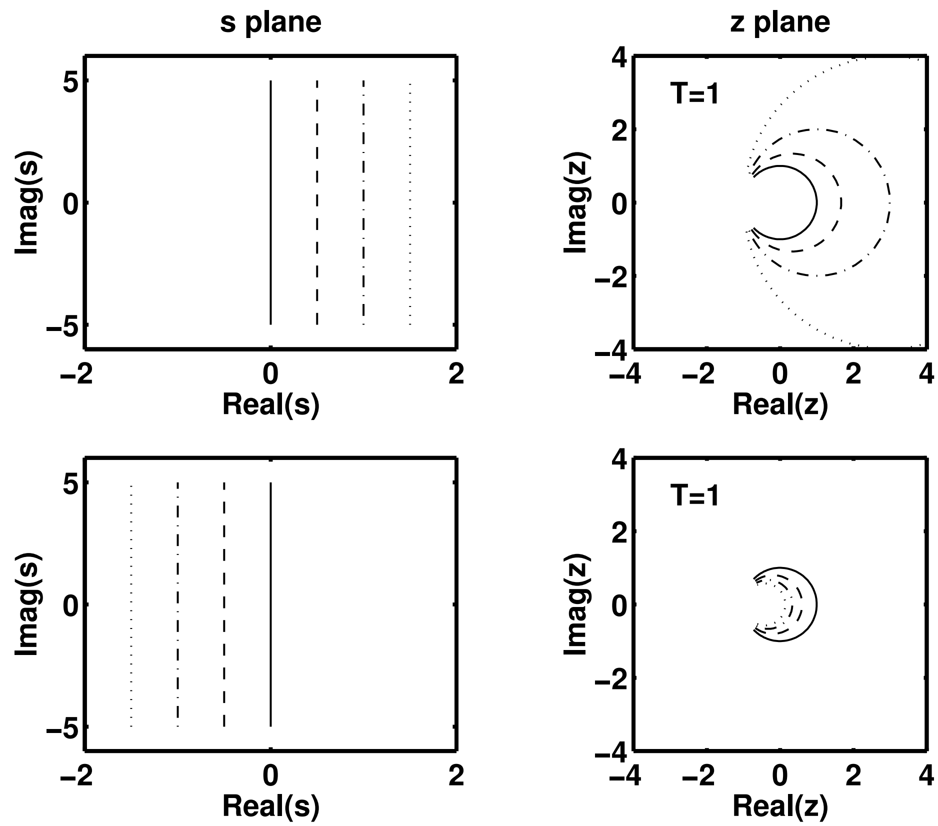
- Determine desired frequency response in terms of analog frequencies, *e.g.*, Hz.
- Convert the desired frequencies into digital frequencies using $\omega = 2\pi F/F_\mathrm{s}$, yielding ω_p, ω_s, etc.
- Map those digital frequencies into analog frequencies $F = \frac{2}{2\pi T_\mathrm{s}} \tan(\omega/2)$.
We can use any convenient T_s for this, *e.g.*, $T_\mathrm{s} = 1$.
- Design an analog filter for those frequencies.
- Transform analog filter into digital filter using the bilinear transformation with that same T_s.

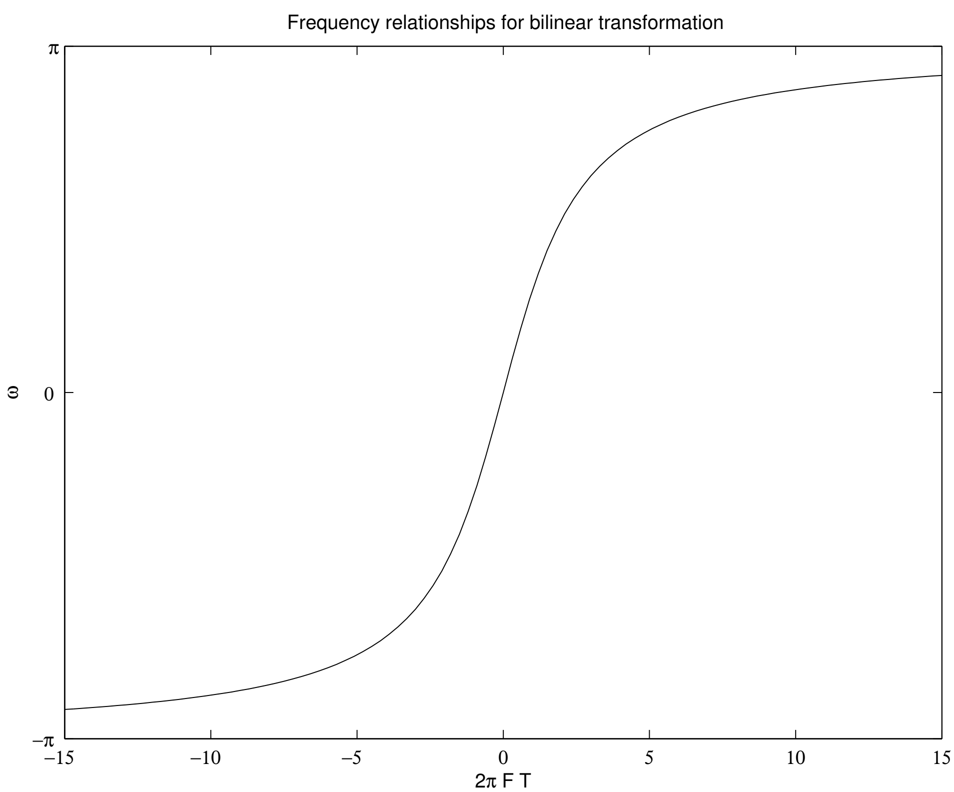
This is all built into commands such as MATLAB's `cheby1` routine.
Use MATLAB's `filtdemo` to experiment with various filter types.

The Chebyshev Type I filters are all-pole analog filters with equiripple behavior in the passband and monotonic in the stopband.

$$|H_\mathrm{a}(F)|^2 = \frac{1}{1 + \varepsilon^2 T_N^2(F/F_\mathrm{pass})} \quad \text{where} \quad T_N(x) \triangleq \begin{cases} \cos\left(N \cos^{-1} x\right), & |x| \le 1 \\ \cosh(N \cosh^{-1} x), & |x| > 1 \end{cases}$$

is the Nth order **Chebyshev polynomial**.
There is **ripple** in the passband of amplitude $1/(1 + \varepsilon^2)$ that is user-controlled.


s plane
z plane
T=1
T=1
Imag(s)
Real(s)
Imag(z)
Real(z)
5
0
−5
−2
0
2
4
2
0
−2
−4
−4
−2
0
2
4

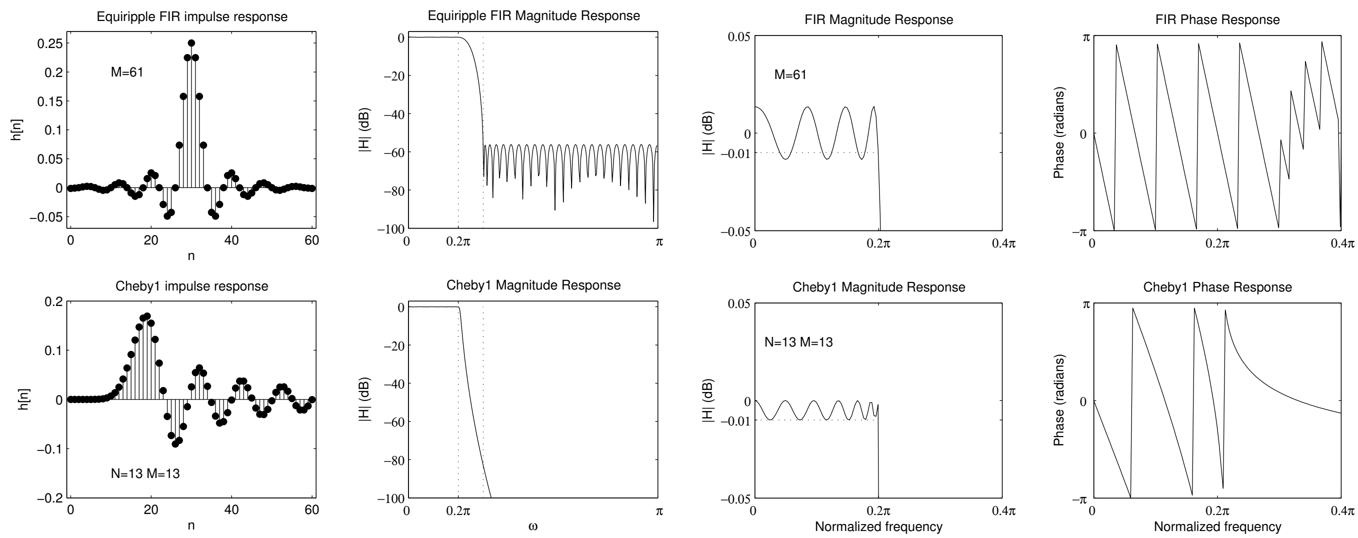

Frequency relationships for bilinear transformation
π
0
−π
ω
−15
−10
−5
0
5
10
15
2π F T

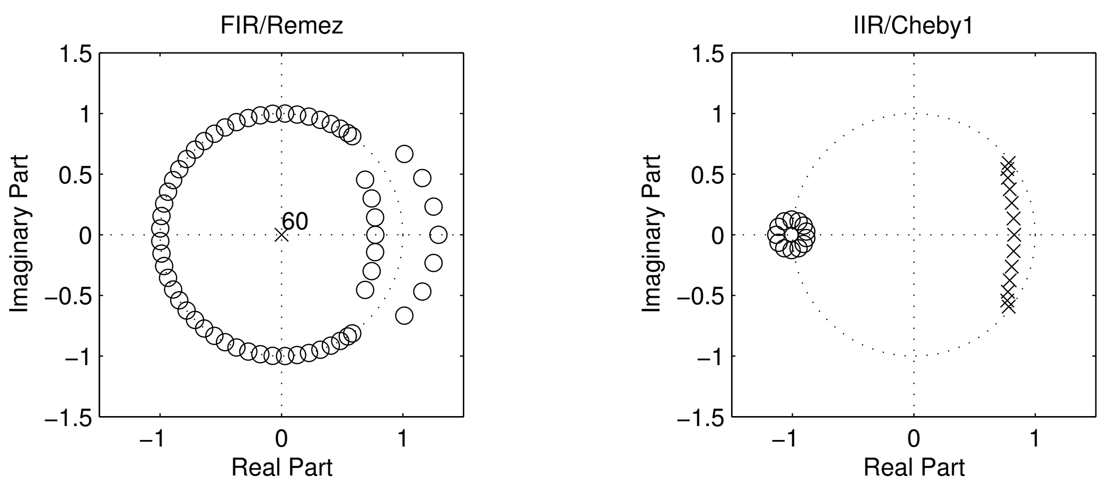
Example. Design the digital crossover considered previously, with $F_{\text{pass}} = 4.4\text{kHz}$, $F_{\text{stop}} = 6.6\text{kHz}$, $F_{\text{s}} = 44\text{kHz}$. So $\omega_{\text{p}} = 0.2\pi$. For the Chebyshev Type I design, ω_{p} is not used.

MATLAB usage:

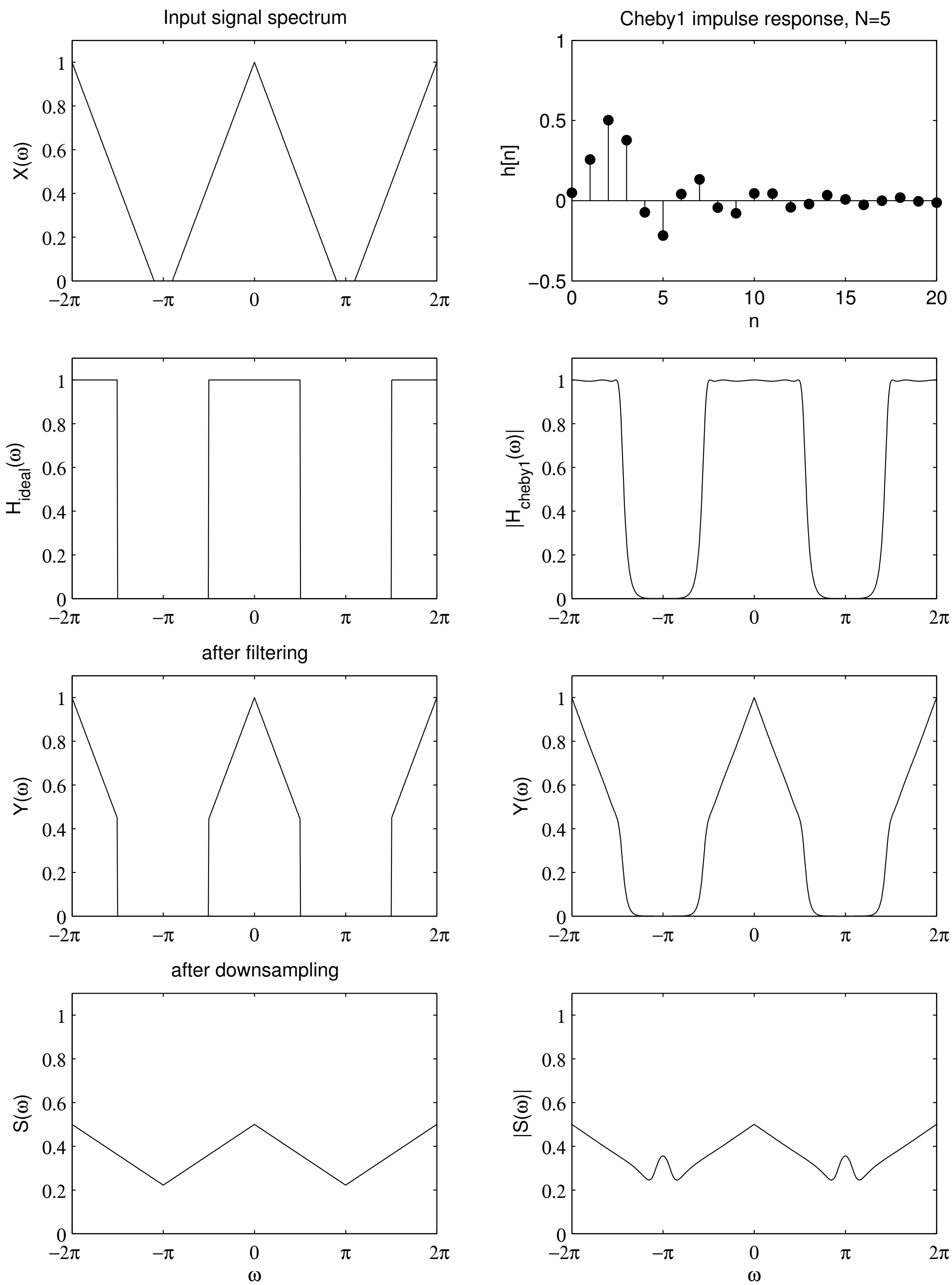
```
N=13; % number of poles
delta1=0.01; % peak-to-peak passband ripple in dB
fc=0.2; % passband cutoff frequency parameter
[b a] = cheby1(N, delta1, fc);
```

Returns coefficients of linear constant coefficient difference equation, with $M = N$.
How would you use `a` and `b` to filter a digital signal in MATLAB? Using the `filter` command.





Example. Apply IIR lowpass filter in context of downsampling. (Details in lecture.)



8.4

Frequency Transformations

skim

Convert lowpass to highpass, bandpass, etc.

8.5

Design of Digital Filters Based on Least-Squares Method

skim

8.6

Summary

- FIR can provide exactly linear phase
- IIR can provide similar magnitude response with fewer coefficients, or lower sidelobes for same number of coefficients
- FIR designs by windowed sinc or frequency sampling are simple
- FIR equiripple designs allow better control of frequency breakpoints and of passband and stopband ripple.
  it is more complicated to implement, but that was solved 30 years ago...
- IIR can be designed by mapping analog filter to digital. The bilinear transformation is particularly flexible, but frequency warping must be taken into account.
- more coefficients: closer to desired response
- wider transition band: less ripple in passband, more stopband attenuation

Summary of tradeoffs

FIR
- + linear phase if symmetric: $h[n] = h[N-1-n], n = 0, \ldots, N-1$
- + easy to design (especially windowed FIR)

IIR
- - never exactly linear phase
- + comparable (or even better!) magnitude response as FIR but with lower order (using poles and zeros)

Windowed FIR

Larger M (vs smaller M)
- + $|\mathcal{H}(\omega)|$ better approximation to $\mathcal{H}_d(\omega)$
- + Sharper transition band
- + some sidelobes have reduced amplitude
- - peak sidelobe in stopband relatively unaffected
- - more computation (or more hardware)
- - longer delay

Nonrectangular windows (vs rectangular window)
- + lower sidelobe amplitude
- + less passband ripple
- - wider transition band