Supplementary Notes for ELEN 4810 Lecture 9
Discrete Time Filter Design

John Wright
Columbia University

December 1, 2025

Disclaimer: These notes are intended to be an accessible introduction to the subject, with no pretense at completeness. In general, you can find more thorough discussions in Oppenheim's book. Please let me know if you find any typos.

Reading suggestions: Oppenheim and Schafer 7.2-7.4, Appendix B,

In this lecture, we will describe some basic techniques for designing filters which amplify or attenuate certain frequencies. Our discussion will focus mostly on the design of low-pass filters, but many of the techniques we describe are general, and can be used to design filters with other characteristics, including band-pass and high-pass filters. Filter design is a vast area, and our discussion will not be comprehensive.

1 FIR vs. IIR Design

When designing a filter, the first question we must address is whether to design a Finite Impulse Response (FIR) system or an Infinite Impulse Response (IIR) system. These two types of filters have very different design methodologies, and offer different tradeoffs to the designer.

Generally, in filter design, we have some target magnitude response $|H_{\text{target}}(e^{j\omega})|$. In these notes, we will typically consider low-pass filters, for which the target magnitude response is a step function

$$|H_{\text{target}}(e^{j\omega})| = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \omega_c \leq |\omega| \leq \pi. \end{cases} \tag{1.1}$$

For both FIR and IIR filters, there is a tradeoff between the resources (hardware, computation) required to implement the filter, and how closely the magnitude response $|H(e^{j\omega})|$ approximates the target, $|H_{\text{target}}(e^{j\omega})|$. An FIR filter with impulse response $h[n]$ can be implemented directly, using the equation for discrete time convolution

$$y[n] = \sum_k h[k]x[n-k]. \tag{1.2}$$

We will typically be interested in causal systems, for which $h[n]$ is nonzero only when $0 \leq n \leq M$. In this situation, the convolution sum (1.2) has only $M+1$ nonzero terms. Thus, the length $M+1$

is a direct measure of the resources needed to implement the filter. The system function $H(z)$ will be a polynomial of degree M in z^{-1}. Hence, it will have exactly M roots, counted with multiplicity. Assuming $h[0] \neq 0$, these will be the only M zeros of the system. Thus, the system has *order* M – it has exactly M zeros, counted with multiplicity. The order of the system M therefore gives a rough idea of the computational resources needed to implement it.

For IIR systems, we will be interested in systems with rational system functions

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{i=0}^{N} a_i z^{-i}}. \tag{1.3}$$

The output of such a system is related to the input via a linear constant coefficient difference equation

$$\sum_{i=0}^{N} a_i y[n-i] = \sum_{k=0}^{M} b_k x[n-k]. \tag{1.4}$$

If the system is assumed to be causal, we can compute $y[n]$ from inputs and previous outputs via the expression

$$y[n] = a_0^{-1} \left(\sum_{k=0}^{M} b_k x[n-k] - \sum_{i=1}^{N} a_i y[n-i] \right). \tag{1.5}$$

The total computation required is roughly proportional to $M + N$. Counting poles and zeros at zero and infinity, there system has a total of $O = \max\{M, N\}$ poles and $O = \max\{M, N\}$ zeros. This number is the *order* of the system, and again can be taken as a rough measure of the resources required for implementation.

For both FIR and IIR systems, as the order of the system increases, we can approximate the target magnitude response $|H_d(e^{j\omega})|$ increasingly well. Nevertheless, these two classes of systems offer quite different advantages.

FIR benefits. We saw in the previous lectures that with FIR filters, it is possible to ensure linear phase[1], by enforcing certain symmetries. In contrast, achieving linear phase with a practical IIR design is almost impossible. So, for applications in which mitigating phase distortion is very important, an FIR design may be preferred. FIR designs also have the good property that they are always stable, while IIR systems need to be explicitly designed for stability. In IIR designs, we should take care that stability is maintained even when the coefficients of the system function are quantized. Finally, because FIR filters can be designed using tools from numerical linear algebra and optimization, it is relatively easy to build FIR filters which approximate rather arbitrary target magnitude responses $|H_d(e^{j\omega})|$. This is more difficult for IIR designs.

IIR benefits. The main benefit of IIR designs over FIR designs is that an IIR design can typically achieve the same quality of approximation to a desired magnitude response with a substantially lower order, which in practice means less computational cost, less hardware, or both. So, IIR designs are preferred for applications in which there is a premium on computation or hardware.

[1]Or its first cousin, generalized linear phase

Differences in design methodology. Typical design methodologies for FIR and IIR filters are very different. IIR systems have very natural analogues in continuous time, and effective discrete-time IIR designs can be obtained simply by transforming a continuous time filter in an appropriate way. We will describe how to do this in the next lecture. The resulting filters are often described by simple, closed form expressions, which are amenable to analysis.

Because almost all interesting continuous time systems are IIR, there is no natural analogue to FIR filter design in continuous time. FIR filter design is typically performed directly in the digital domain, using tools from numerical linear algebra and numerical optimization. Typically, closed form expressions are not available. However, one advantage to numerical filter design is that it is possible to apply the same methods to a wide variety of target magnitude response shapes. In contrast, for IIR design one is generally limited to filters which are already known in continuous time – in particular, to frequency selective ("X-pass") filters.

In the remainder of this lecture, we will describe two basic approaches to FIR design – windowing and Chebyschev approximation.

2 A First Attempt at FIR Design

In this section, we make a first (rather naive) attempt at FIR design. We will attempt to design a causal, FIR system, whose impulse response $h[n]$ is supported on $0 \leq n \leq M$. Since our goal is to produce a low-pass filter, we will take as our target frequency response the linear phase low-pass system

$$H_{\text{target}}(e^{j\omega}) = \begin{cases} e^{-j\omega M/2} & |\omega| < \omega_c \\ 0 & \omega_c \leq |\omega| \leq \pi. \end{cases} \tag{2.1}$$

This simply corresponds to a delay of an ideal low-pass filter by $M/2$ samples. The corresponding impulse response is

$$h_{\text{target}}[n] = \frac{\sin(\omega_c(n - M/2))}{\pi(n - M/2)}. \tag{2.2}$$

As a first attempt, let us try to design a filter $h[n]$ whose frequency response matches $H_{\text{target}}(e^{j\omega})$ as closely as possible, in the sense of energy. Namely, we solve

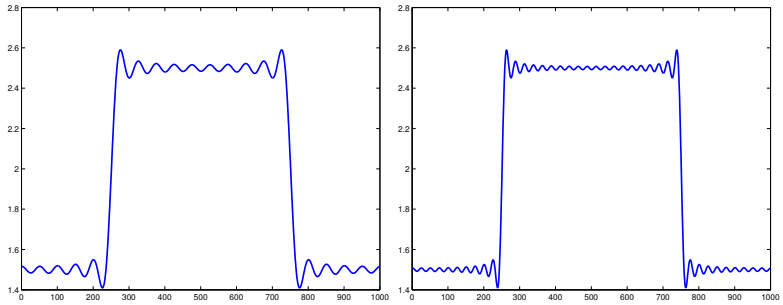
$$\min_{\text{supp}(h) \subseteq \{0,\ldots,M\}} \quad \varepsilon(h, h_{\text{target}}) \quad \equiv \quad \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega}) - H_{\text{target}}(e^{j\omega})|^2 d\omega. \tag{2.3}$$

Parseval's theorem implies that the L^2 error in frequency domain is equal to the ℓ^2 error in time domain:

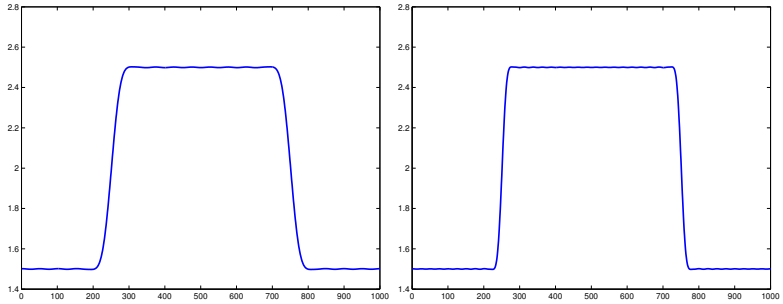
$$\varepsilon(h, h_{\text{target}}) = \sum_{n=-\infty}^{\infty} |h[n] - h_{\text{target}}[n]|^2. \tag{2.4}$$

Since this is just the sum of the squares of the approximation errors between $h[n]$ and $h_{\text{target}}[n]$, the optimal solution is to set $h[n] = h_{\text{target}}[n]$ whenever $0 \leq n \leq M$:

$$h_\star[n] = \begin{cases} h_{\text{target}}[n] & 0 \leq n \leq M, \\ 0 & \text{else.} \end{cases} \tag{2.5}$$



Frequency response magnitude of L^2 optimal low-pass filters supported on $-M \leq n \leq M$
Rectangular window. Left: $M = 20$. Right: $M = 40$.



Frequency response magnitude of windowed low-pass filter supported on $-M \leq n \leq M$
Hamming window. Left: $M = 20$. Right: $M = 40$.

Figure 1: **Gibbs phenomenon in filter design.** Top: optimal (in the sense of energy) filter designed by windowing the target impulse response with a rectangular window. Notice the Gibbs oscillations near the cutoff ω_c.

Thus, we can see $h_\star[n]$ as a *windowed* version of $h_{\text{target}}[n]$:

$$h_\star[n] = h_{\text{target}}[n] w_{\text{rect}}[n], \tag{2.6}$$

where

$$w_{\text{rect}}[n] = \begin{cases} 1 & 0 \leq n \leq m \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

This design is (i) extremely simple and (ii) optimal, in the sense of energy! Nevertheless, it has the very bad property that at the discontinuity ($|\omega| = |\omega_c|$), the magnitude response exhibits large oscillations, due to the Gibbs phenomenon. Figure 1 illustrates this behavior. This is the same disadvantage that we saw to using the rectangular window for spectral analysis.

This example illustrates an important property of optimization: it gives you exactly what you ask for, and nothing else. We asked for a good approximation in L^2 (energy). The L^2 error is not sensitive to large, highly localized oscillations, and so these show up in the solution. *Be careful what you optimize for!*

3 Filter Design Criteria

The previous example suggests that we should be more careful about our design specifications. Instead of aiming for a good approximation to $H_{\text{target}}(e^{j\omega})$ in energy, we will typically aim to approximate the magnitude response

$$|H_{\text{target}}(e^{j\omega})| = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \omega_c \leq |\omega| \leq \pi \end{cases} \tag{3.1}$$

in a *uniform* sense – i.e., to control the worst error between $|H(e^{j\omega})|$ and $|H_{\text{target}}(e^{j\omega})|$.

The ideal low-pass filter specification in (3.1) makes essentially three demands:

(i) Below the target cutoff ω_c, magnitude response is one;

(ii) Above the target cutoff ω_c, the magnitude response is zero; and

(iii) At ω_c, the magnitude response transitions abruptly from one to zero.

If we restrict our attention to stable systems, it is impossible to perfectly satisfy the third requirement: the magnitude response of a stable system is continuous in ω. Instead, we allow some "transition region" between the frequencies that pass and the frequencies that we choose to attenuate. We introduce two numbers $\omega_p < \omega_c < \omega_s$. We call $0 \leq \omega \leq \omega_p$ the *passband*, and $\omega_s \leq \omega \leq \pi$ the *stopband*. We demand that in the stopband, the magnitude response is close to zero:

$$|H(e^{j\omega})| \leq \delta_s \qquad \omega_s \leq \omega \leq \pi, \tag{3.2}$$

while in the passband, we ask the magnitude response to be close to one:

$$1 - \delta_p \;\leq\; |H(e^{j\omega})| \;\leq\; 1 + \delta_p \qquad 0 \leq \omega \leq \omega_p. \tag{3.3}$$

The tolerances δ_s and δ_p indicate how well we are approximating the ideal low-pass magnitude response in the passband and stopband.[2] In the transition region $\omega_p < \omega < \omega_s$, we allow the magnitude response to be arbitrary. In typical filter designs, however, $|H(e^{j\omega_p})| = 1-\delta_p$, $|H(e^{j\omega_s})| = \delta_s$, and $|H(e^{j\omega})|$ decreases monotonically across the transition region.

Within this tolerance specification, we can observe three conflicting design goals:

(i) Minimize the passband tolerance δ_p,

(ii) Minimize the stopband tolerance δ_s,

(iii) Minimize the width $\omega_s - \omega_p$ of the transition region.

For a fixed filter order, it is typically not possible to improve all three of these quantities simultaneously. However, if we increase the filter order, we may be able to make progress on all of them. One way to determine the appropriate filter order is to work backwards, by first deciding a family of filter types, calculating the tolerances and transition region width, and then choosing the minimum filter order needed to make these fall within the design specification.

[2] In the slightly idealized world of these lecture notes, δ_s and δ_p are simply numbers. You may encounter them expressed as multiplicative tolerances, in units of dB. For example, the stopband tolerance δ_s can also be expressed in terms of the "stop band attenuation" $20\log_{10}(1/\delta_s)$.

There is at least one additional consideration: whether to allow "ripples" in the passband, the stopband, or both. Attempting to minimize the tolerance δ_p typically results in solutions whose magnitude response within the passband oscillates between $1 + \delta_s$ and $1 - \delta_s$. In IIR filter design, knowing whether it is more important to eliminate ripples, make the transition as sharp as possible, or minimize the tolerances in the passband and stop band can help to choose an appropriate family of filters to work with.

4 FIR Design by Windowing

With this more detailed tolerance scheme in mind, we can return to the problem of designing FIR filters. The main problem with our previous design

$$h_\star[n] = w_{\text{rect}}[n] h_{\text{target}}[n] \tag{4.1}$$

was the oscillations near ω_c. Because multiplication in time domain is convolution in frequency domain, we can express $H_\star(e^{j\omega})$ as

$$H_\star(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W_{\text{rect}}(e^{j\theta}) H_{\text{target}}(e^{j(\omega-\theta)}) d\theta. \tag{4.2}$$

The Fourier transform of w_{rect} is a periodic sinc. The effect of convolving with a sinc can be thought of as follows: the mainlobe "smears out" the transition from 1 to 0, resulting in a nontrivial transition region. The sidelobes create Gibbs oscillations.

Just as we did for spectral analysis, it is possible to replace w_{rect} with other windows which trade off these two effects in different ways. For example, windows that taper near the edges tend to have wider mainlobes, but smaller sidelobes. Thus, by choosing a different window, we can achieve smaller tolerances δ_s, δ_p, at the expense of a wider transition region. Figure 1 (bottom) illustrates this with a Hamming window: the Hamming window results in a wider transition region, but much smaller oscillations near the transition.

We previously discussed tradeoffs in window selection in the context of spectral analysis of time-varying signals and the short time Fourier transform. There, our discussion was mostly qualitative. For filter design, our needs are more quantitative. We want to produce systems which satisfy a given tolerance scheme. We describe one way of doing this using a parametric family of windows called the *Kaiser windows*. These windows give an approximately optimal tradeoff between mainlobe width and sidelobe height. The Kaiser windows are parameterized by a number β:

$$w_{\text{Kaiser},\beta}[n] = \begin{cases} \dfrac{I_0\left(\beta\left\{1-\left(\frac{n-\alpha}{\alpha}\right)^2\right\}^{1/2}\right)}{I_0(\beta)} & 0 \le n \le M \\ 0 & \text{else.} \end{cases} \tag{4.3}$$

Here, I_0 is a *modified Bessel function of the first kind*. Kaiser windows for various choices of β are illustrated in Figure 2. Notice that $\beta = 0$ corresponds to a rectangular window, and that as β increases, the window tapers more and more smoothly at the edges. Figure 2 also shows the magnitude response of Kaiser windows of various orders. Notice that as β increases, the sidelobes become smaller, but the mainlobe becomes wider. Finally, Figure 3 shows the magnitude response for a fixed

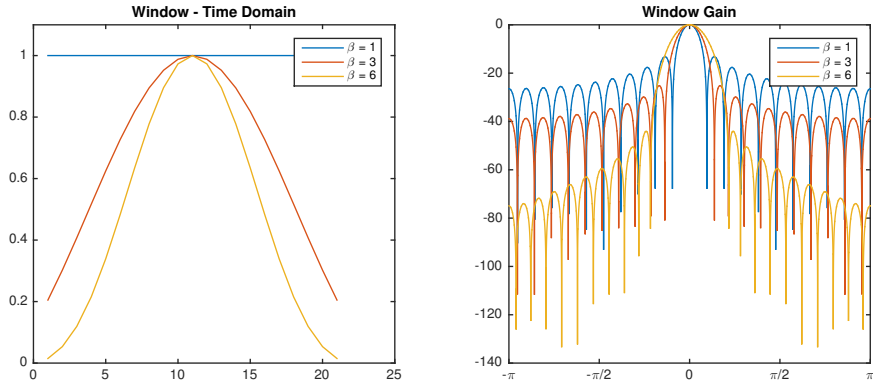


Figure 2: **Kaiser windows for varying β.** Here, the order M is fixed as 20. The Kaiser parameter β varies from 1 to 6. Left: the window in time domain. Notice that as β increases, the window tapers more gently at the edges. Right: the relative gain $20\log_{10}(|W(e^{j\omega})|/|W(e^{j0})|)$. As β increases, the sidelobes become smaller, but the mainlobe becomes wider.

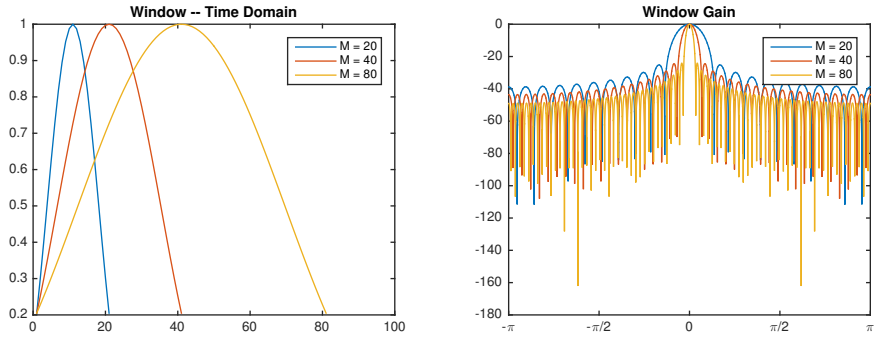


Figure 3: **Kaiser windows for varying order M.** Here, the shape parameter is fixed as $\beta = 3$. We vary the window order from $M = 20$ to $M = 80$. Notice that as the window order changes

β and varying M. Notice that as M increases, the height of the sidelobes stays fixed, but the mainline becomes narrower.

This suggests a methodology for design with the Kaiser window: first choose β based on the desired tolerances δ_p, δ_s, and then choose M based on the desired transition bandwidth $\Delta_\omega = \omega_s - \omega_p$. Unfortunately, the Kaiser window methodology does not let us set the passband and stop band tolerances independently: we have to fix $\delta_p = \delta_s = \delta$. This limitation arises because $w_{\text{Kaiser},\beta}$ is real, and hence has a symmetric magnitude response. This means that the sidelobes of $W_{\text{Kaiser},\beta}(e^{j\omega})$ on both sides of $\omega = 0$ have the same height. So, the amount of extra oscillation (error) introduced in the passband and the stop band is the same.

So, to design with a Kaiser window, we first fix $\delta_p = \delta_s = \delta$, choose β based on δ, and then choose M based on δ and Δ_ω. Experiments and experience suggests the following rules of thumb

for choosing these parameters. We first define

$$A = -20 \log_{10} \delta, \tag{4.4}$$

and then set

$$\beta = \begin{cases} 0.11(A - 8.7) & A > 50, \\ 0.58(A - 21)^{0.4} + 0.079(A - 21) & 21 \leq A \leq 50 \\ 0.0 & A < 21 \end{cases} \tag{4.5}$$

Finally, we choose

$$M = \frac{A - 8}{2.285 \Delta_\omega}. \tag{4.6}$$

5 FIR Design by Optimization: Parks-McClellan Algorithm

In the previous section, we described how to design FIR filters by windowing. A major advantage of this approach is its simplicity. However, this comes at a price. First, windowing does not allow us to control the passband and stopband tolerances independently. In some applications, we may care more about errors in one band rather than the other. Second, windowing does not produce optimal filters: for a given tolerance, the filter length may not be shortest achievable; for a given length, the tolerances may not be the smallest achievable.

In this section, we describe a way to design FIR filters which are optimal, in the sense that they minimize the tolerance, for a given filter length. Our previous attempt at optimal design (in L^2 or energy) suffered from very poor approximation behavior at certain points. To prevent this, we instead seek a filter which approximates the target magnitude response *uniformly*. To accomplish this, we draw on tools from L^∞ approximation (sometimes also referred to as "minimax" or Chebyschev approximation). We do this in two steps. First we describe a general algorithm for solving the general problem of approximating a given target function $f_{\text{target}}(x)$ with a polynomial $f(x)$ of degree at most r. We then describe how to apply this algorithm to the FIR filter design problem.

General L^∞ approximation. We consider the following general setting: we have a target function $f_{\text{target}}(x)$, which we want to approximate over some closed domain $\mathcal{D} \subset \mathbb{R}$. We will assume that

$$\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i \tag{5.1}$$

with the $\mathcal{D}_i$ disjoint, closed, bounded intervals. We assume that f_{target} is continuous over $\mathcal{D}$. We seek approximations $f(x)$ of the form

$$f(x) = \sum_{i=0}^{r} a_i x^i, \tag{5.2}$$

i.e., f is a polynomial of degree r. Letting P_r denote the set of polynomials of degree at most r, $f \in \mathcal{P}_r$. We measure how well f approximates $f_{\text{target}}(x)$ in terms of a weighted error

$$E(x) = W(x) \left(f(x) - f_{\text{target}}(x) \right), \tag{5.3}$$

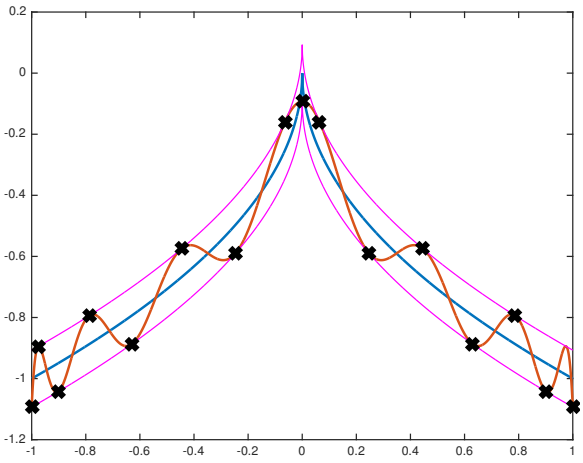


Figure 4: **Chebyschev's Alternation Theorem.** Best approximating polynomial $f_\star$ of degree ≤ 14 to the function $f_{\text{target}}(\boldsymbol{x}) = -|x|^{1/2}$ over $\mathcal{D} = [-1, 1]$. The black x's represent the points of maximum error promised by Chebyschev's theorem. Notice that the sign of the error alternates from each x_i to x_{i+1}.

where $W(x) > 0$ is a positive weight function which is continuous over $\mathcal{D}$. If $W(x)$ is large at a point x, this means that we give large weight to errors at this point – i.e., we demand a more accurate approximation.

Our goal is to find the polynomial $f_\star$ of degree at most r which minimizes the maximum value of the error over the domain $\mathcal{D}$:

$$\min_{f \in P_r} \max_{x \in \mathcal{D}} |E(x)|. \tag{5.4}$$

We sometimes write $\max_{x \in \mathcal{D}} |E(x)|$ as $\|E\|_\infty$. It is possible to show that there is a *unique* best approximating polynomial $f_\star$, which solves (5.4). Moreover, this unique optimal solution is characterized by the *Chebyschev alternation theorem*.

The alternation theorem says that the optimal solution $f_\star$ to (5.4) distributes the error somewhat evenly across $\mathcal{D}$. To be more precise, there must exist at least $r + 2$ points at which the error takes on its maximum value. Moreover, the signs of the error $E(x)$ alternate between these points:

Theorem 5.1 (Chebyschev)**.** *f is the unique optimal solution to* (5.4)*, if and only if there exist at least $r + 2$ points $x_1 < x_2 < \cdots < x_{r+2}$ at which the maximum error is attained:*

$$|E(x_i)| = \|E\|_\infty, \tag{5.5}$$

and then signs of the error alternate from point to point:

$$E(x_i) = -E(x_{i+1}) \qquad i = 1, \ldots, r + 1. \tag{5.6}$$

A proof of this result is beyond the scope of this lecture. The intuition, however, is that if the maximum error were taken at fewer than $r + 2$ points, then we could reduce the error at these points, at the expense of possibly increasing the error at other points. Figure 4 illustrates the alternation theorem, on the specific problem of approximating the function $f_{\text{target}}(x) = -|x|^{1/2}$ over

$\mathcal{D} = [-1, 1]$. f_{target} is plotted in blue. The best approximating polynomial $f_\star$ of degree $r = 14$ is plotted in brown. The black 'x' symbols represent the points x_i promised by Chebyschev's theorem. At these points, the error takes on its maximum value. Moreover, the sign of the error alternates from point to point. By Chebyschev's theorem, this is a necessary and sufficient condition for $f_\star$ to be the optimal approximation to f_{target}.

Using Chebyschev: the Remez algorithm. The alternation theorem promises that there exist points $x_1, \ldots, x_{r+2}$ at which the optimal error is obtained. For some $\delta = \pm\|E\|_\infty$, we have

$$\begin{aligned} f_\star(x_1) + \frac{1}{W(x_1)}\delta &= f_{\text{target}}(x_1) \\ f_\star(x_2) - \frac{1}{W(x_2)}\delta &= f_{\text{target}}(x_2) \\ &\vdots \\ f_\star(x_{r+2}) + (-1)^{r+1}\frac{1}{W(x_{r+2})}\delta &= f_{\text{target}}(x_{r+2}) \end{aligned}$$

We can turn this into a linear system of $r + 2$ equations the coefficients $a_0, \ldots, a_r$ of $f_\star$ and δ, and writes

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^r & \frac{1}{W(x_1)} \\ 1 & x_2 & x_2^2 & \ldots & x_2^r & -\frac{1}{W(x_2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{r+2} & x_{r+2}^2 & \ldots & x_{r+2}^r & (-1)^{r+1}\frac{1}{W(x_{r+2})} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_r \\ \delta \end{bmatrix} = \begin{bmatrix} f_{\text{target}}(x_1) \\ f_{\text{target}}(x_2) \\ \vdots \\ f_{\text{target}}(x_{r+2}) \end{bmatrix}. \tag{5.7}$$

By solving this equation, we can find a polynomial $\hat{f}$ of degree r, and a tolerance δ, such that f approximates f_{target} to tolerance δ at the points $x_1, \ldots, x_{r+2}$ with error of alternating signs. If $x_1, \ldots, x_{r+2}$ were chosen correctly – that is to say, if they actually *are* the points at which the error of the optimal solution alternates, the fitted polynomial will be $f_\star$, and $\delta = \pm\|E\|_\infty$.

On the other hand, if we have not guessed $x_1, \ldots, x_{r+2}$ correctly solving for the a_i and δ will give a polynomial $\hat{f}$ whose approximation error is larger than δ, and which has a local maximum which is not one of the x_i. The Remez algorithm exchanges the $x_1, \ldots, x_{r+2}$ for the local maxima of $\hat{f}$, and repeats this process. It is possible to prove that this algorithm produces a sequence of approximations which converge to $f_\star$, in the sense that

$$\max_x |\hat{f}(x) - f_\star(x)| \to 0. \tag{5.8}$$

Remez for Optimal FIR Filters: the Parks-McClellan Algorithm. The Parks-McClellan algorithm applies Chebyschev's alternation theorem / Remez to design FIR filters. The algorithm has several clever simplifications which avoid the need to directly solve the linear system in (5.7).

We describe how this algorithm applies to the design of Type I FIR low-pass filters; it is not difficult to work out the extension to other types. The impulse response $h[n]$ of a Type I filter is a delayed version of a symmetric sequence, arising by taking a sequence $h_e[n]$ satisfying $h_e[n] =$

$h_e[-n]$, supported on $n = -L, \ldots, L$, and delaying it by $M/2 = L$ samples. The frequency response can be written as

$$H(e^{j\omega}) = e^{-j\omega M/2} A(e^{j\omega}), \tag{5.9}$$

where

$$\begin{aligned} A(e^{j\omega}) &= \sum_{n=-L}^{L} h_e[n] e^{-j\omega n} && (5.10)\\ &= h_e[0] + \sum_{n=1}^{L} 2h_e[n] \cos(\omega n) && (5.11)\\ &= \sum_{k=0}^{L} \alpha_k \cos(\omega k). && (5.12) \end{aligned}$$

The key observation is that for each k we can express $\cos(\omega k)$ as a degree-k polynomial evaluated at $\cos(\omega)$:

$$\cos(\omega k) = T_k(\cos(\omega)), \tag{5.13}$$

where T_k is the k-th Chebyschev polynomial.[3] Using this fact, we can rewrite each of the $\cos(\omega k)$ as a superposition of powers of $\cos(\omega)$. Hence, for appropriate coefficients $a_0, \ldots, a_L$, we have

$$A(e^{j\omega}) = \sum_{k=0}^{L} a_k x^k \Big|_{x=\cos(\omega)}. \tag{5.14}$$

We want to ensure that

$$A(e^{j\omega}) \approx A_{\text{target}}(e^{j\omega}) = \begin{cases} 1 & 0 \le \omega \le \omega_p \\ 0 & \omega_s \le \omega \le \pi. \end{cases} \tag{5.15}$$

Set

$$f(x) = \sum_{k=0}^{L} a_k x^k, \tag{5.16}$$

and

$$f_{\text{target}}(x) = \begin{cases} 1 & \cos(\omega_p) \le x \le 1 \\ 0 & -1 \le x \le \cos(\omega_s), \end{cases} \tag{5.17}$$

and

$$\mathcal{D} = [-1, \cos(\omega_s)] \cup [\cos(\omega_p), 1]. \tag{5.18}$$

If we can ensure that $f \approx f_{\text{target}}$ on $\mathcal{D}$, then

$$A(e^{j\omega}) = f(\cos\omega) \approx f_{\text{target}}(\cos\omega) = A_{\text{target}}(e^{j\omega}). \tag{5.19}$$

Thus, the problem of finding an optimal Type I filter reduces to finding an optimal $A(e^{j\omega})$, which in turn reduces to a polynomial approximation problem. The Parks-McClellan algorithm applies

[3] The Chebyschev polynomials T_k are defined recursively, via the relationship $T_0(x) = 1$, $T_1(x) = x$, $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$. To prove that these polynomials indeed satisfy $T_k(\cos(\omega)) = \cos(\omega k)$, use induction on k and the angle addition relationship $2\cos(\theta)\cos(\pi) = \cos(\theta + \phi) + \cos(\theta - \phi)$.

the Remez exchange method described above to solve this problem, and then backs out the impulse response from the solution. The Remez method produces the coefficients $a_0, \ldots, a_L$ of the optimal polynomial $f_\star(x)$; the impulse response $h[n]$ can be recovered from these coefficients by solving a linear system of equations.

Note that in setting up the problem, we have assumed that ω_p and ω_s are known. The Parks-McClellan algorithm fixes these quantities (and hence, fixes the width of the transition band), and attempts to minimize the passband and stopband tolerances δ_p and δ_s. In some applications, we may care more about the quality of approximation in one band compared to the other. This preference can be incorporated through the weight function $W(x)$. Indeed, suppose that we wish to fix $\delta_p/\delta_s = K$. Then we can set

$$W(x) = \begin{cases} \frac{1}{K} & \cos(\omega_p) \leq \omega \leq 1, \\ 1 & -1 \leq \omega \leq \cos(\omega_s). \end{cases} \tag{5.20}$$

It is not difficult to prove that with this choice of weight, the optimal polynomial $f_\star(x)$ satisfies $\delta_p/\delta_s = K$. So, the Parks-McClellan method fixes ω_p, ω_s, and δ_p/δ_s and filter length, and minimizes the tolerance δ_p.

Consequences of the alternation theorem. Using the alternation theorem, it is possible to make the following observations on the magnitude response of an FIR filter designed via the Parks-McClellan algorithm:

- There are at least $L+2$ and most $L+3$ alternations (points of maximum error, with alternating signs).
- There are always alternations at ω_p and ω_s.
- The filter is *equiripple* in the sense that every local minimum or maximum $\hat{\omega}$ of $f(\cos(\omega))$ in $(0, \pi)$ satisfies $|E(\cos(\hat{\omega}))| = \|E\|_\infty$.

Numerical simplifications. The key step of the Remez algorithm is solving the linear system (5.7) to find the polynomial $f(x)$ and tolerance δ given the putative alternation points $x_1, \ldots, x_{r+2}$. This is potentially problematic: the matrix in (5.7) can be quite ill-conditioned. The Parks-McClellan algorithm avoids directly solving this system. Instead, it uses a closed form expression for δ, and then produces the polynomial $f(x)$ by Lagrange interpolation:

> **Lagrange interpolation.** *Given $r+1$ point pairs $(x_1, y_1), \ldots, (x_{r+1}, y_{r+1})$, the polynomial of degree r which passes through all these points is*
>
> $$f(x) = \sum_{\ell=1}^{r+1} y_\ell \prod_{k \neq \ell} \frac{x - x_k}{x_\ell - x_k}. \tag{5.21}$$
>
> *To observe that this is correct, simply check that $f(x_i) = y_i$ for each i.*

The Parks-McClellan algorithm is implemented in Matlab as `firpm`.

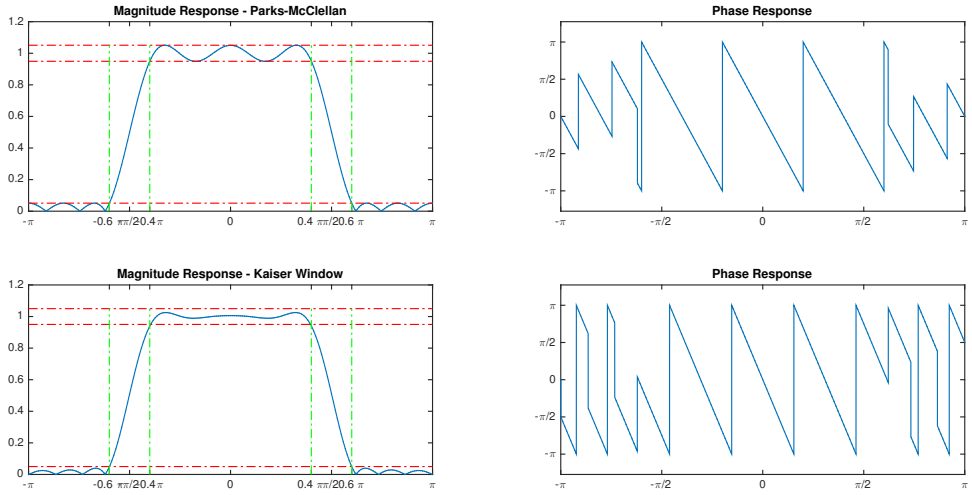


Figure 5: **Comparison of FIR design methodologies.** Top: Minimax optimal design via Parks-McClellan. Bottom: Design with the Kaiser window method. The minimax optimal design requires order $M = 10$; the Kaiser window method requires $M = 13$. The minimax optimal design distributes the error evenly across the passband and stop band, exhibiting alternations of equal height (the equiripple property). In contrast, the filter produced by Kaiser windowing exhibits larger error near the transition band.

6 A Comparison

In this section, we build FIR filters using both Kaiser window and optimization methodologies, and then compare their properties. We fix the specification as $\omega_c = \pi/2$, $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$. So, the transition bandwidth is $\Delta_\omega = 0.2\pi$. We fix $\delta_s = \delta_p = 0.05$.[4]

Applying the Parks-McClellan algorithm with $M = 10$, we obtain a filter $h_{\text{pm}}[n]$ which meets the desired specification. Figure 5 (top) plots the magnitude and phase responses of $H_{\text{pm}}(e^{j\omega})$. Notice that the designed filter is equiripple in both the passband and stopband.

Applying the design equations for the Kaiser window method, we require a filter of order $M = 13$. Figure 5 (bottom) plots the magnitude and phase response. Notice that the designed filter has larger ripples near the transition band, and smaller ripples further away. The Kaiser window methodology requires a larger M because it does not distribute the error uniformly.

7 IIR Filter Design

In filter design, the basic task is to go from design considerations (passband width, transition band width, stop band width, ect.) to a choice of $H(z)$, generally expressed in terms of its poles and zeros. Once we have $H(z)$, we can determine an appropriate block diagram for implementation. One approach to generating $H(z)$ is to design a continuous-time filter with the desired properties, and then transform it back to the discrete domain. This approach is popular mostly for historical

[4]We note again that the windowing method cannot handle situations in which $\delta_s \neq \delta_p$.

reasons: by the time digital signal processing emerged in the 1960's and 70's, continuous-time filter design was an extensively studied area, with a vast literature and an array of practical design tools. Currently, the most widely-used discrete-time IIR designs for frequency-selective ("X-pass") filters are still based on transformations of continuous-time filters. Fortunately, these design tools are standard components of software packages such as Matlab's Signal Processing Toolbox. A relatively rudimentary understanding of the tradeoffs involved is enough to get started using them.[5]

8 A Reminder on the Laplace Transform

To borrow ideas from continuous-time filter design, we need to know at least a little about that area. Because our goal is to specify the discrete-time transfer function $H(z)$, we start with its continuous-time analogue, the *Laplace transform* $H_a(s)$ of the continuous-time impulse response $h_a(t)$. The Laplace transform of a continuous-time signal $f(t)$ is a function of a complex variable $s \in \mathbb{C}$:

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}dt. \tag{8.1}$$

Note that if $s = j\Omega$ is purely imaginary, $F(s)$ is simply the (continuous-time) Fourier transform, evaluated at Ω. So, just as the DTFT is the $\mathcal{Z}$-transform evaluated around the unit circle, the Fourier transform is the Laplace transform evaluated along the imaginary axis:

$$F(j\Omega) = \int_{-\infty}^{\infty} f(t)e^{-j\Omega t}dt = F(s)\Big|_{s=j\Omega}. \tag{8.2}$$

Just as the $\mathcal{Z}$-transform summation may only converge for certain $z \in \mathbb{C}$, the Laplace transform integral (8.1) may only be well-defined for a subset of $s \in \mathbb{C}$. Also, like the $\mathcal{Z}$-transform, it happens that often the Laplace transform $F(s)$ is a rational function of s. It is very useful to understand how the region of convergence of $F(s)$, as well as the poles and zeros, relate to the properties of the system whose impulse response is $f(t)$. If $h(t)$ is the (continuous-time) impulse response of a stable and causal system, then $h(t) = 0$ for $t < 0$, and $\int_{t=0}^{\infty} |h(t)|dt < +\infty$. This implies that for any s satisfying $\mathrm{Re}[s] \geq 0$,

$$\begin{aligned}
\int_{-\infty}^{\infty} |h(t)e^{-st}|dt &= \int_{0}^{\infty} |h(t)| \underbrace{|e^{-\mathrm{Re}[s]t}|}_{\leq 1, \text{ since } \mathrm{Re}[s]t \geq 0} \underbrace{|e^{-j\mathrm{Im}[s]t}|}_{=1} dt && (8.3)\\
&\leq \int_{0}^{\infty} |h(t)|dt && (8.4)\\
&< +\infty. && (8.5)
\end{aligned}$$

So, the *poles of a stable, causal continuous-time system lie in the left half plane* ($\mathrm{Re}[s] < 0$). There is a rich theory associated with the Laplace transform, and its utility in studying continuous time systems. You probably saw some of the highlights of this in your previous signals and systems course. For our purposes, we recap the two most important facts from the above discussion:

- The continuous-time Fourier transform is equal to the Laplace transform, evaluated along the imaginary axis $s = j\Omega$.

[5] Of course, we still encourage you to study them with great depth and diligence!

- For stable, causal systems, any poles of the Laplace transform lie in the left half plane ($\mathrm{Re}(s) < 0$).

9 Turning a Continuous-Time System into a Discrete-Time System

Suppose we are given $H_a(s)$, the Laplace transform of a continuous-time impulse response $h_a(t)$.[6] For the designs of interest here, $H_a(s)$ will be a *rational function* of s. We can generate another function $H(z)$ by making a substitution

$$s = \frac{z-1}{z+1}, \tag{9.1}$$

giving

$$H(z) = H_a\left(\frac{z-1}{z+1}\right). \tag{9.2}$$

In signal processing, this mapping is referred to as *bilinear transformation*.

The mapping $z \mapsto \psi(z) = \frac{z-1}{z+1}$ is well-defined for all $z \in \mathbb{C} \setminus \{-1\}$. Strictly speaking,

$$\psi : \mathbb{C} \setminus \{-1\} \to \mathbb{C} \setminus \{1\}. \tag{9.3}$$

Solving for z in the equation $s = \psi(z)$, we obtain an inverse relationship

$$z = \frac{s+1}{1-s}, \tag{9.4}$$

which is valid for all $s \in \mathbb{C} \setminus \{1\}$. The mappings ψ and ψ^{-1} are highly nonlinear transformations of the complex plane. From the definition of ψ, it is in no way obvious why this might be the *right* way to transform the complex plane to convert a continuous time transfer function $H_a(s)$ to a discrete time transfer function $H(z)$. We make three observations below, which justify the use of the transformation (9.1).

Bilinear transformation maps rational $H_a(s)$ to rational $H(z)$. Suppose that $H_a(s)$ is rational:

$$H_a(s) = \alpha \frac{\prod_{i=1}^{M}(s-c_i)}{\prod_{\ell=1}^{N}(s-d_\ell)}. \tag{9.5}$$

The following shows that $H(z)$ is also a rational function of z, and that its poles and zeros correspond to those of $H_a(s)$ in a natural way:

Proposition 9.1. *Suppose that $H_a(s)$ satisfies (9.5), and that none of the c_i or d_ℓ are equal to 1. Then $H(z)$ is also a rational function of z, with zeros of the form $\psi^{-1}(c_i)$ and poles $\psi^{-1}(d_\ell)$.*

[6] Note that typically, filter characteristics are specified in the Fourier domain. We usually start with the form of the magnitude response $|H_a(j\Omega)|^2$, and then determine a stable, causal $H_a(s)$ whose magnitude squared is $|H_a(j\Omega)|^2$. In designing the continuous-time system $H_a(s)$, we may never actually write down the impulse response $h_a(t)$.

Proof. Write

$$\begin{aligned} H(z) &= H_a\left(\frac{z-1}{z+1}\right) && (9.6)\\ &= \alpha\frac{\prod_{i=1}^{M}\left(\frac{z-1}{z+1}-c_i\right)}{\prod_{\ell=1}^{N}\left(\frac{z-1}{z+1}-d_\ell\right)} && (9.7)\\ &= \alpha(z+1)^{N-M}\frac{\prod_{i=1}^{M}\left(z-1-c_i(z+1)\right)}{\prod_{\ell=1}^{N}\left(z-1-d_\ell(z+1)\right)} && (9.8)\\ &= \alpha(z+1)^{N-M}\left(\frac{\prod_{i=1}^{M}(1-c_i)}{\prod_{\ell=1}^{N}(1-d_\ell)}\right)\frac{\prod_{i=1}^{M}\left(z-\frac{1+c_i}{1-c_i}\right)}{\prod_{\ell=1}^{N}\left(z-\frac{1+d_\ell}{1-d_\ell}\right)} && (9.9)\\ &= \alpha(z+1)^{N-M}\left(\frac{\prod_{i=1}^{M}(1-c_i)}{\prod_{\ell=1}^{N}(1-d_\ell)}\right)\frac{\prod_{i=1}^{M}\left(z-\psi^{-1}(c_i)\right)}{\prod_{\ell=1}^{N}\left(z-\psi^{-1}(d_\ell)\right)} && (9.10) \end{aligned}$$

□

This is auspicious: practically all useful continuous-time filter design techniques produce rational designs. Applying bilinear transformation produces rational discrete-time filter, whose poles and zeros correspond to the poles and zeros of the continuous-time filter in a very natural way. As discussed briefly in the last two lectures, rational $H(z)$ are especially well-suited to implementation, and can be converted into block diagrams in a number of ways.

Bilinear transformation maps the imaginary line to the unit circle. However, we have yet to say anything about how the frequency response of the resulting discrete time filter relates to that of the continuous-time filter we started with. Recall that the $H_a(s)$ restricted to $s = j\Omega$ gives the Fourier transform $H_a(j\Omega)$. In contrast, $H(z)$ evaluated around the unit circle gives the discrete-time Fourier transform $H(e^{j\omega})$. The bilinear transformation maps the imaginary axis to the unit circle:

Proposition 9.2. *The transformation $z = \frac{s+1}{1-s}$ maps the imaginary axis $\{s \mid s = j\Omega,\ \Omega \in \mathbb{R}\}$ onto the unit circle $\left\{z \mid z = e^{j\omega}\right\} \setminus \left\{e^{j\pi}\right\}$.*

Proof. For $s = j\Omega$,

$$\begin{aligned} z &= \frac{1+s}{1-s}\\ &= \frac{1+j\Omega}{1-j\Omega}. \end{aligned} \tag{9.11}$$

Because $|1 + j\Omega| = |1 - j\Omega|$, $|z| = 1$. □

So, under this transformation, the discrete-time Fourier transform $H(e^{j\omega})$ is simply the "compression" of the continuous-time Fourier transform $H_a(j\Omega)$ to the unit circle: if $z = e^{j\omega}$, the corresponding point s is $s = j\Omega$, with

$$\Omega = \tan\left(\omega/2\right), \tag{9.12}$$

and $\omega = 2\tan^{-1}(\Omega)$.[7] In particular, low-pass continuous-time filters transform into low-pass discrete-time filters; high-pass continuous-time filters transform into low-pass discrete-time filters.

Bilinear transformation maps poles in the left half plane to poles inside the unit circle. We mentioned above that stable, causal $H_a(s)$ have their poles in the left half-plane ($\mathrm{Re}[s] < 0$). Stable, causal $H(z)$ have poles inside the unit circle ($|z| < 1$). The previous proposition can be extended to show that for any s in the left half plane, the corresponding z is inside the unit circle. Indeed,

Proposition 9.3. *The transformation $z = \frac{s+1}{1-s}$ maps the left half plane $\{s \mid \mathrm{Re}[s] < 0\}$ onto the open disc $\{z \mid |z| < 1\}$.*

Proof. For $s = \alpha + j\Omega$, we simply notice that

$$z = \frac{1+s}{1-s} = \frac{1+\alpha+j\Omega}{1-\alpha-j\Omega}. \tag{9.16}$$

The magnitude of the numerator is $\sqrt{(1+\alpha)^2+\Omega^2}$, while the magnitude of the denominator is $\sqrt{(1-\alpha)^2+\Omega^2}$. If s is in the left half plane, α is negative, and the denominator has larger magnitude. Conversely, suppose that $|z| < 1$. Then $z = re^{j\theta}$ with $|r| < 1$. The corresponding s is

$$\begin{aligned} s &= \frac{z-1}{z+1} && (9.17)\\ &= \frac{re^{j\theta}-1}{re^{j\theta}+1} && (9.18)\\ &= \frac{(re^{j\theta}-1)(re^{-j\theta}+1)}{|re^{j\theta}+1|^2} && (9.19)\\ &= \frac{r^2+r(e^{j\theta}-e^{-j\theta})-1}{|re^{j\theta}+1|^2}. && (9.20) \end{aligned}$$

The denominator is real and positive. In the numerator, $r(e^{j\theta}-e^{-j\theta})$ is purely imaginary; the real part r^2-1 is necessarily negative. □

A few additional notes. The text describes the bilinear transform via the equivalent formula $s = \frac{1-z^{-1}}{1+z^{-1}}$. This leads to equivalent expressions for $H(z)$ phrased as rational functions of z^{-1}. Identical considerations and properties apply. The text also includes an additional multiplicative factor in the expression for s, writing $s = \frac{2}{T_d}\frac{z-1}{1+z}$. This multiplier has absolutely no consequence on the design

[7] To show these relationships, note that

$$\begin{aligned} s &= \frac{e^{j\omega}-1}{e^{j\omega}+1} = \frac{(e^{j\omega}-1)(e^{-j\omega}+1)}{|e^{j\omega}+1|^2} && (9.13)\\ &= \frac{2j\sin(\omega)}{(1+\cos\omega)^2+\sin^2(\omega)} = j\frac{2\sin(\omega)}{2+2\cos(omega)} && (9.14)\\ &= j\frac{\sin(\omega)}{1+\cos(\omega)} = j\tan(\omega/2). && (9.15) \end{aligned}$$

process, and minimal consequences for the properties of the transformation, and so we omit it for simplicity.[8]

10 Four Designs for Continuous-Time Low-Pass Filters

We briefly describe four of the most popular types of continuous-time filters. In practical applications, each type can be constructed using existing computational tools – for example, Matlab's signal processing toolbox has a dedicated command for each. Using the Matlab command `bilinear`, any of these continuous-time filters can then be converted into a discrete time filter.[9] In the section, we briefly describe the properties of each of these types of filter. Figure 6 compares them on a concrete design example, and illustrates their characteristics.

Butterworth filters. A (lowpass) Butterworth filter with cutoff Ω_c and order n has squared magnitude response

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2n}} \tag{10.1}$$

This function is monotone decreasing. For any choice of $n \geq 1$, $|H_c(j0)| = 1$, $|H_c(j\Omega_c)| = 1/\sqrt{2}$, and $\lim_{\Omega\to\infty} |H_c(j\Omega)| = 0$. However, as n increases, the transition from 1 to 0 becomes increasingly abrupt.

The main advantage of the Butterworth filter is that it has no ripples – the magnitude response is monotone in both the passband and the stopband. In fact, the filter is characterized by the fact that its magnitude response is "maximally flat" at $\Omega = 0$:

$$\frac{d^k}{d\omega^k}|H_c(j\Omega)|^2\big|_{\Omega=0} = 0 \tag{10.2}$$

for $k = 1, 2, \ldots, 2n-1$. This is the largest number of derivatives that can be zero for any order n filter.

The price of this relatively flat magnitude response is a wider transition region, compared to designs that allow ripples in the passband and/or stopband. Compared to designs that are not monotone, the magnitude response of the Butterworth "rolls off" much more slowly, and hence has a wider transition region.

Chebyschev Types I and II. Chebyschev filters are built out of Chebyschev polynomials, $T_n(x)$. For each n, T_n is a polynomial of degree n. The T_n can be inductively, via the relationships

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x). \tag{10.3}$$

With a bit of trigonometry (which we did in the previous lecture), this can be shown to imply that

$$T_n(x) = \cos(n\cos^{-1}(x)), \tag{10.4}$$

[8] Roughly speaking, the parameter $\frac{2}{T_d}$ arises if we try to view the discrete-time filter as an approximation to the continuous-time filter, derived via numerical integration with the trapezoidal rule. We do not emphasize this viewpoint here.

[9] Alternatively, they can be specified directly in the discrete domain using the command `designfilter`.

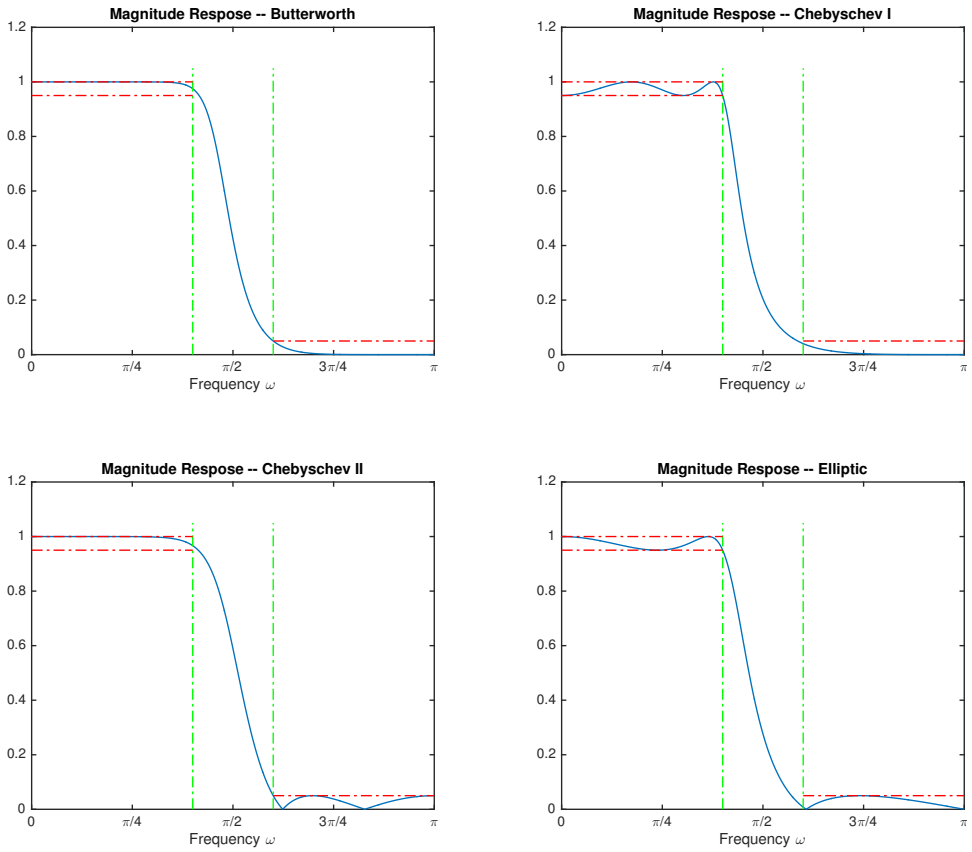


Figure 6: **Four IIR Designs.** We design IIR lowpass filters with the following characteristics: $\omega_p = 0.4 \times \pi$, $\omega_s = 0.6 \times \pi$, $\delta_p = \delta_s = 0.05$. Top left: Butterworth filter. An order $n = 7$ filter is required to meet the specification. Notice the monotone decrease in both the passband and stopband. Top right: Chebyschev I filter. An order $n = 4$ filter is required to meet the specification. Notice the presence of ripple in the passband, and monotone decrease in the stopband. Bottom left: Chebyshev II filter. Again an order $n = 4$ filter is required. Notice that now the magnitude response is monotone in the passband, but exhibits ripple in the stopband. Bottom right: an elliptic filter. Here an order $n = 3$ filter is sufficient. Notice that this is the lowest order of any of the four filter types considered here. Notice the ripple in both the passband and stopband.

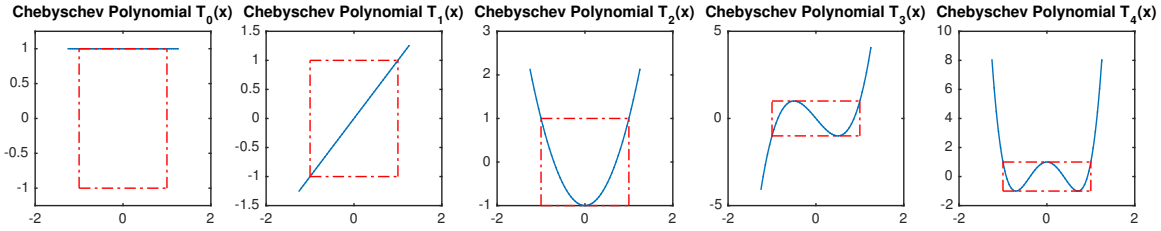


Figure 7: **Chebyshev polynomials of order** $0, \ldots 4$**.** Each figure plots $T_n(x)$ for $x \in [-1.5, 1.5]$ (blue). Also plotted is the unit box $[-1, 1] \times [-1, 1]$. Notice that for $x \in [-1, 1]$, $T_n(x)$ oscillates between -1 and 1. Moreover, for $n > 0$, $|T_n(x)|$ increases montonically when $x > 1$.

for $x \in [-1, 1]$. In particular, for $x \in [-1, 1]$, $|T_n(x)| \leq 1$. This curious family of polynomials is interesting because it minimizes (over all possible degree-n polynomials) the maximum value over $[-1, 1]$:

Fact 10.1. *Let $T_n(x)$ denote the n-th order Chebyschev polynomial, and let $f(x) = \frac{1}{2^{n-1}} T_n(x)$. Then $f(x)$ minimizes $\max_{x \in [-1,1]} |g(x)|$ over all degree-n polynomials g with leading coefficient 1.*

Figure 7 plots the first five Chebyschev polynomials $T_0, \ldots, T_4$. In each plot, the unit box $[-1, 1] \times [-1, 1]$ is outlined in red. Notice that for x between -1 and 1, $T_n(x)$ indeed oscillates between -1 and 1. As $|x|$ increases away from 1, $|T_n(x)|$ increases to $+\infty$. Moreover, the larger n is, the more rapidly $|T_n(x)|$ increases outside of $[-1, 1]$.

The Chebyschev Type I filter takes advantage of this behavior to construct a filter which is equiripple within the passband, and falls off monotonically outside the passband. The magnitude-squared function of the Chebyschev Type I filter is

$$|H_c(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 T_n^2(\Omega/\Omega_c)}. \tag{10.5}$$

Notice that for $0 \leq \Omega \leq \Omega_c$, $|H_c(j\Omega)|^2$ oscillates between $\frac{1}{1+\varepsilon^2}$ and 1. Moreover, when Ω exceeds Ω_c, $T_n^2(\Omega/\Omega_c)$ blows up to ∞, and so $|H_c(j\Omega)|^2 \to 0$.

The Type I Chebyschev filter exhibits ripples in the passband, but is monotone in the stopband. We can also achieve the opposite behavior (monotone in passband, but ripples in the stopband) by using the Chebyschev polynomials in a slightly different way. A Type II Chebyschev filter satisfies

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (\varepsilon^2 T_n^2(\Omega_c/\Omega))^{-1}}. \tag{10.6}$$

Notice that when $\Omega \to 0$, $\Omega_c/\Omega \to \infty$, and so $T_n^2(\Omega_c/\Omega) \to \infty$; this implies that as $\Omega \searrow 0$, $|H_c(j\Omega)|^2$ approaches 1. Moreover, $|H_c(j\Omega)|^2$ is monotone decreasing on $(0, \Omega_c)$. When $\Omega > \Omega_c$, $0 < \Omega_c/\Omega < 1$, and so $T_n^2(\Omega_c/\Omega)$ oscillates between zero and one in the stopband.

Elliptic Filters. Whereas Chebyshev filters either attempt to control the maximum error in the passband (Type I) or stop band (Type II), *elliptic filters* minimize the transition width for a given order N. The magnitude response has the form

$$|H_c(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 U_n^2(\Omega)}, \tag{10.7}$$

where U_n is a Jacobi elliptic function. Important properties of elliptic filters are that: (i) they exhibit ripples in both the passband and the stopband and (ii) they exhibit the sharpest possible transition between the passband and stopband, for a given choice of N, ω_p, and the filter tolerances.

11 Frequency Transformations for High-Pass and Bandpass Filters

Using the designs described above and bilinear transformation, we can produce a variety of discrete-time low-pass filters. Moreover, by performing appropriate transformations of either the s-plane or the z-plane, we can convert these low-pass filters into bandpass or even high-pass filters. Such transformations can be applied to the continuous-time filter, before bilinear transformation, or to the discrete-time filter, after bilinear transformation.

Following Section 7.4 of the text, we describe this process in the $\mathcal{Z}$-domain. Suppose we are given a low-pass filter, with system function

$$H_{lp}(z^{-1}) \tag{11.1}$$

We generate a new filter $H(z^{-1}) = H_{lp}(Z^{-1})\big|_{Z^{-1}=\varphi(z^{-1})}$. Similar to bilinear transformation, we wish to guarantee to map rational H_{lp} to rational H. We also would like to map the unit circle to itself, so that the frequency response of H can be related directly to the frequency response of H_{lp}. Finally, we wish to map stable, causal systems to stable, causal systems, and so the open disc should map to itself. It can be shown that if we wish to satisfy these requirements, we must set

$$Z^{-1} = \pm \prod_{k=1}^{N} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}}, \tag{11.2}$$

where the complex scalars α_k have magnitude strictly less than one.

For example, to map a low-pass filter to a high-pass filter, we can set

$$Z^{-1} = -\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}, \qquad \alpha = -\frac{\cos((\theta_c + \omega_c)/2)}{\cos((\theta_c - \omega_c)/2)}, \tag{11.3}$$

where θ_c is the cutoff frequency of H_{lp}, and ω_c is the target cutoff frequency. Because $\varphi(z^{-1})$ is all-pass, whenever $z = e^{j\omega}$, $Z = e^{j\theta}$, for some θ which depends on ω. For example, if $\alpha = 0$, we have $Z^{-1} = -z^{-1}$, and so we may take $\theta = \omega + \pi$. In this situations, low frequencies ω map to high frequencies θ, and vice-versa.

The text also provides formulas for converting low-pass filters to low-pass filters with different cutoff frequencies, as well as formulas for producing bandpass and bandstop filters.