Digital Filter Design

By:
Douglas L. Jones

Digital Filter Design

By:
Douglas L. Jones

Online:
< http://cnx.org/content/col10285/1.1/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: June 9, 2005
PDF generated: February 4, 2011
For copyright and attribution information for the modules contained in this collection, see p. 40.

Table of Contents

Overview of Digital Filter Design 1

1 FIR Filter Design

- **1.1** Linear Phase Filters 3
- **1.2** Window Design Method 7
- **1.3** Frequency Sampling Design Method for FIR filters 9
- **1.4** Parks-McClellan FIR Filter Design 11
- **1.5** Lagrange Interpolation 18
- Solutions 19

2 IIR Filter Design

- **2.1** Overview of IIR Filter Design 21
- **2.2** Prototype Analog Filter Design 22
- **2.3** IIR Digital Filter Design via the Bilinear Transform 28
- **2.4** Impulse-Invariant Design 31
- **2.5** Digital-to-Digital Frequency Transformations 32
- **2.6** Prony's Method 33
- **2.7** Linear Prediction 36
- Solutions 38

Index 39

Attributions 40

Overview of Digital Filter Design[1]

Advantages of FIR filters

1. Straight forward conceptually and simple to implement
2. Can be implemented with fast convolution
3. Always stable
4. Relatively insensitive to quantization
5. Can have linear phase (same time delay of all frequencies)

Advantages of IIR filters

1. Better for approximating analog systems
2. For a given magnitude response specification, IIR filters often require much less computation than an equivalent FIR, particularly for narrow transition bands

Both FIR and IIR filters are very important in applications.

Generic Filter Design Procedure

1. Choose a desired response, based on application requirements
2. Choose a filter class
3. Choose a quality measure
4. Solve for the filter in class 2 optimizing criterion in 3

Perspective on FIR filtering

Most of the time, people do L^∞ optimal design, using the Parks-McClellan algorithm (Section 1.4). This is probably the second most important technique in "classical" signal processing (after the Cooley-Tukey (radix-2[2]) FFT).

Most of the time, FIR filters are designed to have linear phase. The most important advantage of FIR filters over IIR filters is that they can have exactly linear phase. There are advanced design techniques for minimum-phase filters, constrained L^2 optimal designs, *etc.* (see chapter 8 of text). However, if only the **magnitude** of the response is important, IIR filers usually require much fewer operations and are typically used, so the bulk of FIR filter design work has concentrated on linear phase designs.

[1] This content is available online at <http://cnx.org/content/m12776/1.2/>.
[2] "Decimation-in-time (DIT) Radix-2 FFT" <http://cnx.org/content/m12016/latest/>

Chapter 1

FIR Filter Design

1.1 Linear Phase Filters[1]

In general, for $-\pi \leq \omega \leq \pi$

$$H(\omega) = |H(\omega)| e^{-(i\theta(\omega))}$$

Strictly speaking, we say $H(\omega)$ is linear phase if

$$H(\omega) = |H(\omega)| e^{-(i\omega K)} e^{-(i\theta_0)}$$

Why is this important? A linear phase response gives the **same time delay for ALL frequencies**! (Remember the shift theorem.) This is very desirable in many applications, particularly when the appearance of the time-domain waveform is of interest, such as in an oscilloscope. (see Figure 1.1)

[1] This content is available online at <http://cnx.org/content/m12802/1.2/>.

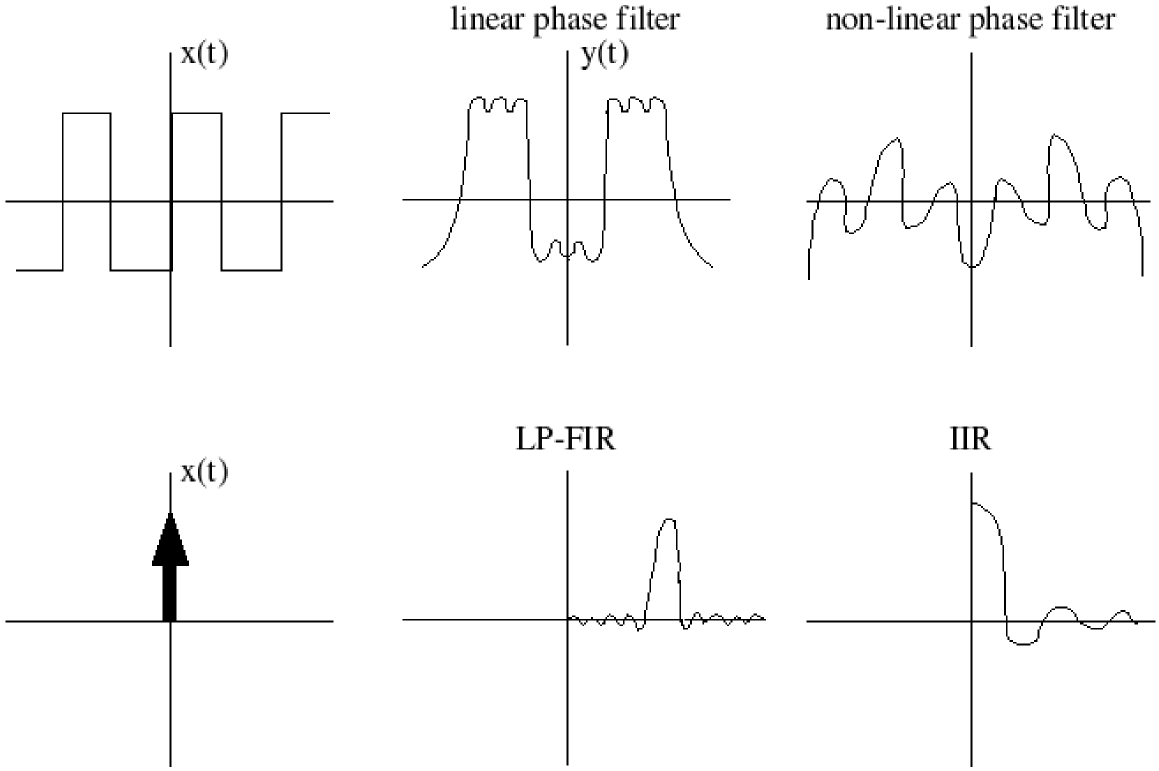


Figure 1.1

1.1.1 Restrictions on h(n) to get linear phase

$$H(\omega) = \sum_{h=0}^{M-1} h(n)\, e^{-(i\omega n)} = h(0) + h(1)\, e^{-(i\omega)} + h(2)\, e^{-(i2\omega)} + \dots + h(M-1)\, e^{-(i\omega(M-1))} = e^{-\left(i\omega\frac{M-1}{2}\right)}\left(h(0)\, e^{i\omega\frac{M-1}{2}} + \dots + h(M-1)\, e^{-\left(i\omega\frac{M-1}{2}\right)}\right) = e^{-\left(i\omega\frac{M-1}{2}\right)}\left((h(0)+h(M-1))\cos\left(\frac{M-1}{2}\omega\right) + (h(1)+h(M-2))\cos\left(\frac{M-3}{2}\omega\right) + \dots + i\left(h(0)\sin\left(\frac{M-1}{2}\omega\right) + \right.\right. \tag{1.1}$$

For linear phase, we require the right side of (1.1) to be $e^{-(i\theta_0)}$(real,positive function of ω). For $\theta_0 = 0$, we thus require

$$h(0) + h(M-1) = \text{real number}$$

$$h(0) - h(M-1) = \text{pure imaginary number}$$

$$h(1) + h(M-2) = \text{pure real number}$$

$$h(1) - h(M-2) = \text{pure imaginary number}$$

$$\vdots$$

Thus $h(k) = h^*(M-1-k)$ is a **necessary** condition for the right side of (1.1) to be real valued, for $\theta_0 = 0$.

For $\theta_0 = \frac{\pi}{2}$, or $e^{-(i\theta_0)} = -i$, we require

$$h(0) + h(M-1) = \text{pure imaginary}$$

$$h(0) - h(M-1) = \text{pure real number}$$

$$\vdots$$

$$\Rightarrow h(k) = -(h^*(M-1-k))$$

Usually, one is interested in filters with **real-valued** coefficients, or see Figure 1.2 and Figure 1.3.

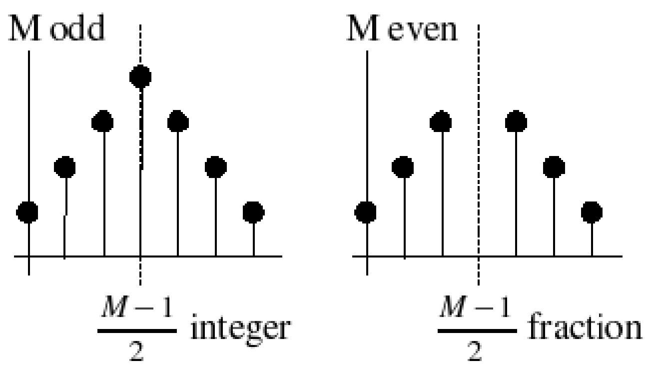


Figure 1.2: $\theta_0 = 0$ (Symmetric Filters). $h(k) = h(M-1-k)$.

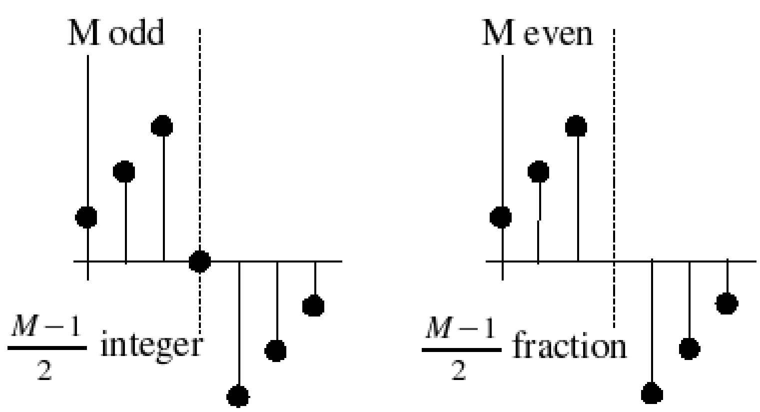


Figure 1.3: $\theta_0 = \frac{\pi}{2}$ (Anti-Symmetric Filters). $h(k) = -h(M-1-k)$.

Filter design techniques are usually slightly different for each of these four different filter types. We will study the most common case, symmetric-odd length, in detail, and often leave the others for homework or tests or for when one encounters them in practice. Even-symmetric filters are often used; the anti-symmetric filters are rarely used in practice, except for special classes of filters, like differentiators or Hilbert transformers, in which the desired response is anti-symmetric.

So far, we have satisfied the condition that $H(\omega) = A(\omega)e^{-(i\theta_0)}e^{-\left(i\omega\frac{M-1}{2}\right)}$ where $A(\omega)$ is **real-valued**. However, we have **not** assured that $A(\omega)$ is **non-negative**. In general, this makes the design techniques much more difficult, so most FIR filter design methods actually design filters with **Generalized Linear Phase**: $H(\omega) = A(\omega)e^{-\left(i\omega\frac{M-1}{2}\right)}$, where $A(\omega)$ is **real-valued**, but possible negative. $A(\omega)$ is called the **amplitude of the frequency response**.

NOTE: $A(\omega)$ usually goes negative only in the stopband, and the stopband phase response is generally unimportant.

NOTE: $|H(\omega)| = \pm(A(\omega)) = A(\omega)e^{-\left(i\pi\frac{1}{2}(1-sign(A(\omega)))\right)}$ where $sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$

Example 1.1
Lowpass Filter

Desired $|\mathbf{H}(\omega)|$

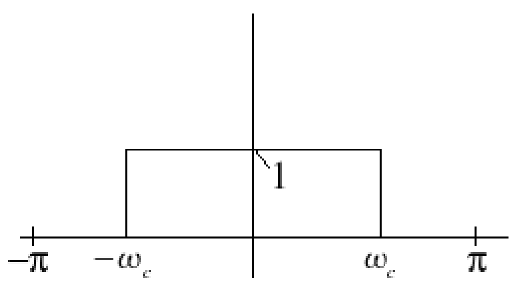


Figure 1.4

Desired $\angle\mathbf{H}(\omega)$

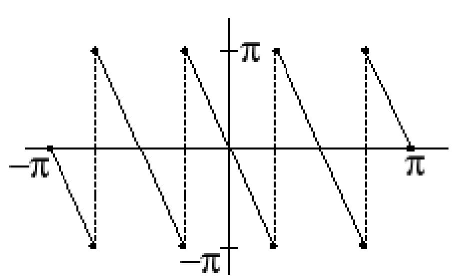


Figure 1.5: The slope of each line is $-\frac{M-1}{2}$.

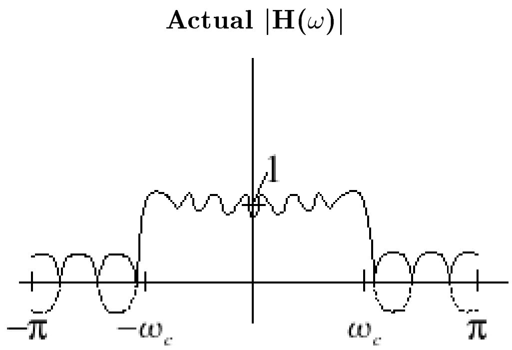


Figure 1.6: $A(\omega)$ goes negative.

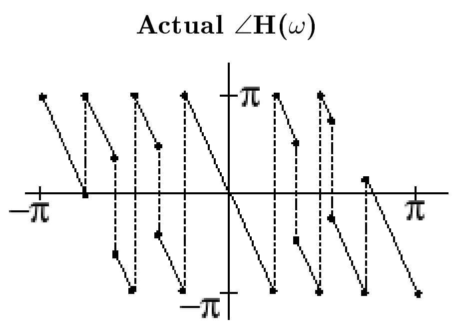


Figure 1.7: 2π phase jumps due to periodicity of phase. π phase jumps due to sign change in $A(\omega)$.

Time-delay introduces generalized linear phase.

NOTE: For odd-length FIR filters, a linear-phase design procedure is equivalent to a zero-phase design procedure followed by an $\frac{M-1}{2}$-sample delay of the impulse response[2]. For even-length filters, the delay is non-integer, and the linear phase must be incorporated directly in the desired response!

1.2 Window Design Method[3]

The truncate-and-delay design procedure is the simplest and most obvious FIR design procedure.

Exercise 1.1 ***(Solution on p. 19.)***

Is it any Good?

[2]"Impulse Response of a Linear System" <http://cnx.org/content/m12041/latest/>

1.2.1 L2 optimization criterion

find $\forall n, 0 \leq n \leq M-1 : (h[n])$, maximizing the energy difference between the desired response and the actual response: i.e., find

$$min_{\mathrm{hn}} \left\{ \mathrm{hn}, \int_{-\pi}^{\pi} (|H_d(\omega) - H(\omega)|)^2 d\omega \right\}$$

by Parseval's relationship[4]

$$min_{\mathrm{hn}} \left\{ \mathrm{hn}, \int_{-\pi}^{\pi} (|H_d(\omega) - H(\omega)|)^2 d\omega \right\} = 2\pi \sum_{n=-\infty}^{\infty} (|h_d[n] - h[n]|)^2 = 2\pi \left(\sum_{n=-\infty}^{-1} (|h_d[n] - h[n]|)^2 + \sum_{n=0}^{M-1} (|h_d[n] - h[n]|)^2 + \sum_{n=M}^{\infty} (|h_d[n] - h[n]|)^2 \right) \quad (1.2)$$

Since $\forall n, n < 0 n \geq M : (h[n])$ this becomes

$$min_{\mathrm{hn}} \left\{ \mathrm{hn}, \int_{-\pi}^{\pi} (|H_d(\omega) - H(\omega)|)^2 d\omega \right\} = \sum_{h=-\infty}^{-1} (|h_d[n]|)^2 + \sum_{n=0}^{M-1} (|h[n] - h_d[n]|)^2 + \sum_{n=M}^{\infty} (|h_d[n]|)^2$$

NOTE: $h[n]$ has no influence on the first and last sums.

The best we can do is let

$$h[n] = \begin{cases} h_d[n] & \text{if } 0 \leq n \leq M-1 \\ 0 & \text{if else} \end{cases}$$

Thus $h[n] = h_d[n] w[n]$,

$$w[n] = \begin{cases} 1 & \text{if } 0 \leq n (M-1) \\ 0 & \text{if else} \end{cases}$$

is **optimal** in a least-total-sqaured-error (L_2, or energy) sense!

Exercise 1.2 ***(Solution on p. 19.)***

Why, then, is this design often considered undersirable?

For desired spectra with discontinuities, the least-square designs are poor in a minimax (worst-case, or L_∞) error sense.

1.2.2 Window Design Method

Apply a more gradual truncation to reduce "ringing" (Gibb's Phenomenon[5])

$$\forall \mathrm{n}\ 0 \leq \mathrm{n} \leq \mathrm{M}-1\ \mathrm{hn} = \ \mathrm{h\ d\ nwn} : (\mathrm{n}\ 0 \leq \mathrm{n} \leq \mathrm{M}-1\ \mathrm{hn} = \ \mathrm{h\ d\ nwn})$$

NOTE: $H(\omega) = H_d(\omega) * W(\omega)$

The window design procedure (except for the boxcar window) is ad-hoc and not optimal in any usual sense. However, it is very simple, so it is sometimes used for "quick-and-dirty" designs of if the error criterion is itself heurisitic.

[4] "Parseval's Theorem" <http://cnx.org/content/m0047/latest/>
[5] "Gibbs's Phenomena" <http://cnx.org/content/m10092/latest/>

1.3 Frequency Sampling Design Method for FIR filters[6]

Given a desired frequency response, the frequency sampling design method designs a filter with a frequency response **exactly** equal to the desired response at a particular set of frequencies ω_k.

Procedure

$$\forall k, k = [o, 1, \dots, N-1] : \left(H_d(\omega_k) = \sum_{n=0}^{M-1} h(n) e^{-(i\omega_k n)}\right) \tag{1.3}$$

NOTE: Desired Response must incluce linear phase shift (if linear phase is desired)

Exercise 1.3 ***(Solution on p. 19.)***

What is $H_d(\omega)$ for an ideal lowpass filter, cotoff at ω_c?

NOTE: This set of linear equations can be written in matrix form

$$H_d(\omega_k) = \sum_{n=0}^{M-1} h(n) e^{-(i\omega_k n)} \tag{1.4}$$

$$\begin{pmatrix} H_d(\omega_0) \\ H_d(\omega_1) \\ \vdots \\ H_d(\omega_{N-1}) \end{pmatrix} = \begin{pmatrix} e^{-(i\omega_0 0)} & e^{-(i\omega_0 1)} & \dots & e^{-(i\omega_0 (M-1))} \\ e^{-(i\omega_1 0)} & e^{-(i\omega_1 1)} & \dots & e^{-(i\omega_1 (M-1))} \\ \vdots & \vdots & \vdots & \vdots \\ e^{-(i\omega_{M-1} 0)} & e^{-(i\omega_{M-1} 1)} & \dots & e^{-(i\omega_{M-1} (M-1))} \end{pmatrix} \begin{pmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{pmatrix} \tag{1.5}$$

or

$$H_d = Wh$$

So

$$h = W^{-1} H_d \tag{1.6}$$

NOTE: W is a square matrix for $N = M$, and invertible as long as $\omega_i \neq \omega_j + 2\pi l$, $i \neq j$

1.3.1 Important Special Case

What if the frequencies are equally spaced between 0 and 2π, i.e. $\omega_k = \frac{2\pi k}{M} + \alpha$

Then

$$H_d(\omega_k) = \sum_{n=0}^{M-1} h(n) e^{-\left(i\frac{2\pi kn}{M}\right)} e^{-(i\alpha n)} = \sum_{n=0}^{M-1} \left(h(n) e^{-(i\alpha n)}\right) e^{-\left(i\frac{2\pi kn}{M}\right)} = \text{DFT!}$$

so

$$h(n) e^{-(i\alpha n)} = \frac{1}{M} \sum_{k=0}^{M-1} H_d(\omega_k) e^{i\frac{2\pi nk}{M}}$$

or

$$h[n] = \frac{e^{i\alpha n}}{M} \sum_{k=0}^{M-1} H_d[\omega_k] e^{i\frac{2\pi nk}{M}} = e^{i\alpha n} \text{IDFT}[H_d[\omega_k]]$$

[6]This content is available online at <http://cnx.org/content/m12789/1.2/>.

1.3.2 Important Special Case #2

$h[n]$ symmetric, linear phase, and has real coefficients. Since $h[n] = h[M-n]$, there are only $\frac{M}{2}$ degrees of freedom, and only $\frac{M}{2}$ linear equations are required.

$$\begin{aligned} H[\omega_k] &= \sum_{n=0}^{M-1} h[n]\, e^{-(i\omega_k n)} \\ &= \begin{cases} \sum_{n=0}^{\frac{M}{2}-1} h[n] \left(e^{-(i\omega_k n)} + e^{-(i\omega_k (M-n-1))}\right) & \text{if M even} \\ \sum_{n=0}^{M-\frac{3}{2}} h[n] \left(e^{-(i\omega_k n)} + e^{-(i\omega_k (M-n-1))}\right) \left(h\left[\frac{M-1}{2}\right] e^{-\left(i\omega_k \frac{M-1}{2}\right)}\right) & \text{if M odd} \end{cases} \\ &= \begin{cases} e^{-\left(i\omega_k \frac{M-1}{2}\right)} 2 \sum_{n=0}^{\frac{M}{2}-1} h[n] \cos\left(\omega_k \left(\frac{M-1}{2} - n\right)\right) & \text{if M even} \\ e^{-\left(i\omega_k \frac{M-1}{2}\right)} 2 \sum_{n=0}^{M-\frac{3}{2}} h[n] \cos\left(\omega_k \left(\frac{M-1}{2} - n\right)\right) + h\left[\frac{M-1}{2}\right] & \text{if M odd} \end{cases} \end{aligned} \tag{1.7}$$

Removing linear phase from both sides yields

$$A(\omega_k) = \begin{cases} 2 \sum_{n=0}^{\frac{M}{2}-1} h[n] \cos\left(\omega_k \left(\frac{M-1}{2} - n\right)\right) & \text{if M even} \\ 2 \sum_{n=0}^{M-\frac{3}{2}} h[n] \cos\left(\omega_k \left(\frac{M-1}{2} - n\right)\right) + h\left[\frac{M-1}{2}\right] & \text{if M odd} \end{cases}$$

Due to symmetry of response for real coefficients, only $\frac{M}{2}$ ω_k on $\omega \in [0, \pi)$ need be specified, with the frequencies $-\omega_k$ thereby being implicitly defined also. Thus we have $\frac{M}{2}$ **real-valued** simultaneous linear equations to solve for $h[n]$.

1.3.2.1 Special Case 2a

$h[n]$ symmetric, odd length, linear phase, real coefficients, and ω_k equally spaced: $\forall k, 0 \leq k \leq M-1 : \left(\omega_k = \frac{n\pi k}{M}\right)$

$$\begin{aligned} h[n] &= \text{IDFT}\left[H_d(\omega_k)\right] \\ &= \frac{1}{M} \sum_{k=0}^{M-1} A(\omega_k)\, e^{-\left(i\frac{2\pi k}{M}\right)\frac{M-1}{2}} e^{i\frac{2\pi nk}{M}} \\ &= \frac{1}{M} \sum_{k=0}^{M-1} A(k)\, e^{i\left(\frac{2\pi k}{M}\left(n - \frac{M-1}{2}\right)\right)} \end{aligned} \tag{1.8}$$

To yield real coefficients, $A(\omega)$ mus be symmetric

$$(A(\omega) = A(-\omega)) \Rightarrow (A[k] = A[M-k])$$

$$\begin{aligned} h[n] &= \frac{1}{M}\left(A(0) + \sum_{k=1}^{\frac{M-1}{2}} A[k] \left(e^{i\frac{2\pi k}{M}\left(n - \frac{M-1}{2}\right)} + e^{-\left(i2\pi k\left(n - \frac{M-1}{2}\right)\right)}\right)\right) \\ &= \frac{1}{M}\left(A(0) + 2\sum_{k=1}^{\frac{M-1}{2}} A[k] \cos\left(\frac{2\pi k}{M}\left(n - \frac{M-1}{2}\right)\right)\right) \\ &= \frac{1}{M}\left(A(0) + 2\sum_{k=1}^{\frac{M-1}{2}} A[k] (-1)^k \cos\left(\frac{2\pi k}{M}\left(n + \frac{1}{2}\right)\right)\right) \end{aligned} \tag{1.9}$$

Simlar equations exist for even lengths, anti-symmetric, and $\alpha = \frac{1}{2}$ filter forms.

1.3.3 Comments on frequency-sampled design

This method is simple conceptually and very efficient for equally spaced samples, since $h[n]$ can be computed using the IDFT.

$H(\omega)$ for a frequency sampled design goes **exactly** through the sample points, but it may be very far off from the desired response for $\omega \neq \omega_k$. This is the main problem with frequency sampled design.

Possible solution to this problem: specify more frequency samples than degrees of freedom, and minimize the total error in the frequency response at all of these samples.

1.3.4 Extended frequency sample design

For the samples $H(\omega_k)$ where $0 \leq k \leq M-1$ and $N > M$, find $h[n]$, where $0 \leq n \leq M-1$ minimizing $\parallel H_d(\omega_k) - H(\omega_k) \parallel$

For $\parallel l \parallel_\infty$ norm, this becomes a linear programming problem (standard packages availble!)

Here we will consider the $\parallel l \parallel_2$ norm.

To minimize the $\parallel l \parallel_2$ norm; that is, $\sum_{n=0}^{N-1} |H_d(\omega_k) - H(\omega_k)|$, we have an overdetermined set of linear equations:

$$\begin{pmatrix} e^{-(i\omega_0 0)} & \dots & e^{-(i\omega_0 (M-1))} \\ \vdots & \vdots & \vdots \\ e^{-(i\omega_{N-1} 0)} & \dots & e^{-(i\omega_{N-1}(M-1))} \end{pmatrix} h = \begin{pmatrix} H_d(\omega_0) \\ H_d(\omega_1) \\ \vdots \\ H_d(\omega_{N-1}) \end{pmatrix}$$

or

$$Wh = H_d$$

The minimum error norm solution is well known to be $h = \left(\overline{W}W\right)^{-1}\overline{W}H_d$; $\left(\overline{W}W\right)^{-1}\overline{W}$ is well known as the pseudo-inverse matrix.

> NOTE: Extended frequency sampled design discourages radical behavior of the frequency response between samples for sufficiently closely spaced samples. However, the actual frequency response may no longer pass exactly through **any** of the $H_d(\omega_k)$.

1.4 Parks-McClellan FIR Filter Design[7]

The approximation tolerances for a filter are very often given in terms of the maximum, or worst-case, deviation within frequency bands. For example, we might wish a lowpass filter in a (16-bit) CD player to have no more than $\frac{1}{2}$-bit deviation in the pass and stop bands.

$$H(\omega) = \begin{cases} 1 - \frac{1}{2^{17}} \leq |H(\omega)| \leq 1 + \frac{1}{2^{17}} & \text{if } |\omega| \leq \omega_p \\ \frac{1}{2^{17}} \geq |H(\omega)| & \text{if } \omega_s \leq |\omega| \leq \pi \end{cases}$$

The Parks-McClellan filter design method efficiently designs linear-phase FIR filters that are optimal in terms of worst-case (minimax) error. Typically, we would like to have the shortest-length filter achieving these specifications. Figure Figure 1.8 illustrates the amplitude frequency response of such a filter.

[7]This content is available online at <http://cnx.org/content/m12799/1.3/>.

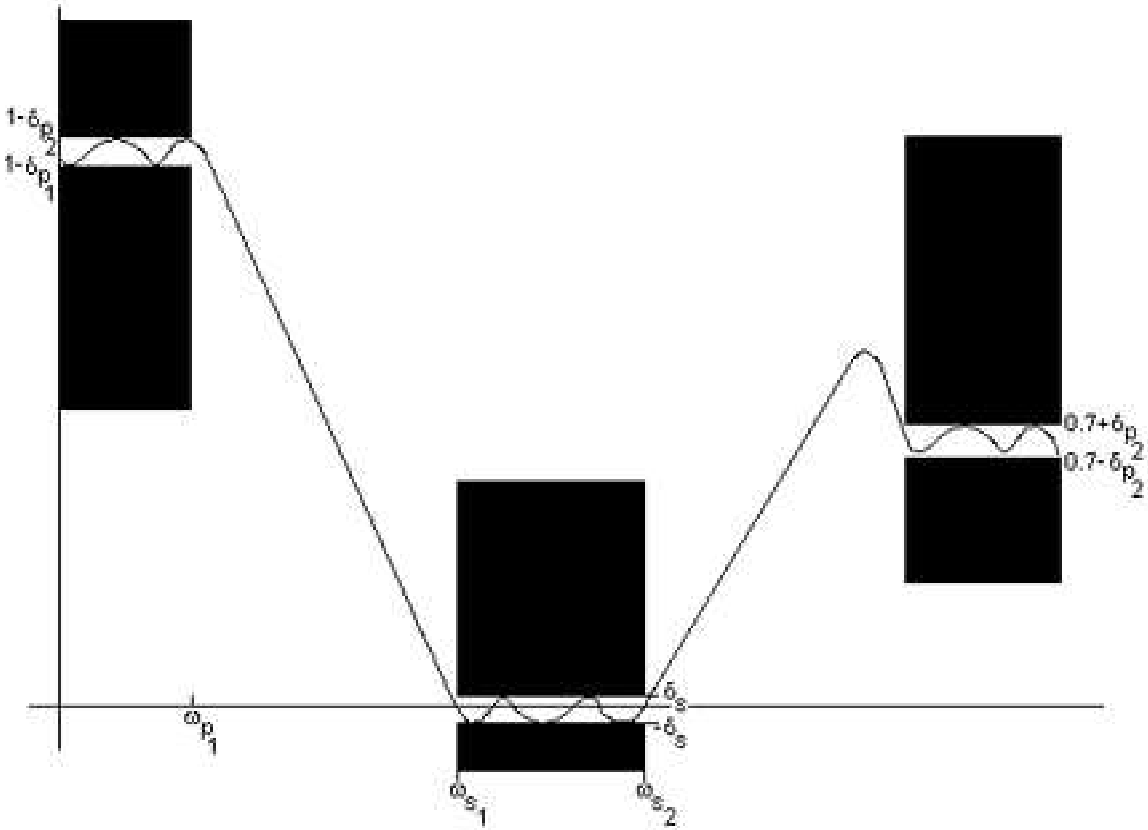


Figure 1.8: The black boxes on the left and right are the passbands, the black boxes in the middle represent the stop band, and the space between the boxes are the transition bands. Note that overshoots may be allowed in the transition bands.

Exercise 1.4 ***(Solution on p. 19.)***

Must there be a transition band?

1.4.1 Formal Statement of the L-∞ (Minimax) Design Problem

For a given filter length (M) and type (odd length, symmetric, linear phase, for example), and a relative error weighting function $W(\omega)$, find the filter coefficients minimizing the maximum error

$$\underset{h}{argmin}\underset{\omega\in F}{argmax}|E(\omega)| = \underset{h}{argmin}\| E(\omega) \|_\infty$$

where

$$E(\omega) = W(\omega)(H_d(\omega) - H(\omega))$$

and F is a compact subset of $\omega \in [0, \pi]$ (*i.e.*, all ω in the passbands and stop bands).

NOTE: Typically, we would often rather specify $\| E(\omega) \|_\infty \leq \delta$ and minimize over M and h; however, the design techniques minimize δ for a given M. One then repeats the design procedure for different M until the minimum M satisfying the requirements is found.

We will discuss in detail the design only of odd-length symmetric linear-phase FIR filters. Even-length and anti-symmetric linear phase FIR filters are essentially the same except for a slightly different implicit weighting function. For arbitrary phase, exactly optimal design procedures have only recently been developed (1990).

1.4.2 Outline of L-∞ Filter Design

The Parks-McClellan method adopts an indirect method for finding the minimax-optimal filter coefficients.

1. Using results from Approximation Theory, simple conditions for determining whether a given filter is L^∞ (minimax) optimal are found.
2. An iterative method for finding a filter which satisfies these conditions (and which is thus optimal) is developed.

That is, the L^∞ filter design problem is actually solved **indirectly**.

1.4.3 Conditions for L-∞ Optimality of a Linear-phase FIR Filter

All conditions are based on Chebyshev's "Alternation Theorem," a mathematical fact from polynomial approximation theory.

1.4.3.1 Alternation Theorem

Let F be a compact subset on the real axis x, and let $P(x)$ be and Lth-order polynomial

$$P(x) = \sum_{k=0}^{L} a_k x^k$$

Also, let $D(x)$ be a desired function of x that is continuous on F, and $W(x)$ a positive, continuous weighting function on F. Define the error $E(x)$ on F as

$$E(x) = W(x)(D(x) - P(x))$$

and

$$\| E(x) \|_\infty = \underset{x \in F}{argmax} |E(x)|$$

A necessary and sufficient condition that $P(x)$ is the unique Lth-order polynomial minimizing $\| E(x) \|_\infty$ is that $E(x)$ exhibits **at least** $L+2$ "alternations;" that is, there must exist at least $L+2$ values of x, $x_k \in F$, $k = [0, 1, \dots, L+1]$, such that $x_0 < x_1 < \dots < x_{L+2}$ and such that $E(x_k) = -E(x_{k+1}) = \pm(\| E \|_\infty)$

Exercise 1.5 ***(Solution on p. 19.)***

What does this have to do with linear-phase filter design?

1.4.4 Optimality Conditions for Even-length Symmetric Linear-phase Filters

For M even,

$$A(\omega) = \sum_{n=0}^{L} h(L-n) \cos\left(\omega\left(n+\frac{1}{2}\right)\right)$$

where $L = \frac{M}{2} - 1$ Using the trigonometric identity $\cos(\alpha+\beta) = \cos(\alpha-\beta) + 2\cos(\alpha)\cos(\beta)$ to pull out the $\frac{\omega}{2}$ term and then using the other trig identities (p. 19), it can be shown that $A(\omega)$ can be written as

$$A(\omega) = \cos\left(\frac{\omega}{2}\right) \sum_{k=0}^{L} \alpha_k \cos^k(\omega)$$

Again, this is a polynomial in $x = \cos(\omega)$, except for a weighting function out in front.

$$\begin{aligned} E(\omega) &= W(\omega)(A_d(\omega) - A(\omega)) \\ &= W(\omega)\left(A_d(\omega) - \cos\left(\frac{\omega}{2}\right)P(\omega)\right) \\ &= W(\omega)\cos\left(\frac{\omega}{2}\right)\left(\frac{A_d(\omega)}{\cos\left(\frac{\omega}{2}\right)} - P(\omega)\right) \end{aligned} \quad (1.10)$$

which implies

$$E(x) = W^{'}(x)\left(A_d^{'}(x) - P(x)\right) \quad (1.11)$$

where

$$W^{'}(x) = W\left((\cos(x))^{-1}\right)\cos\left(\frac{1}{2}(\cos(x))^{-1}\right)$$

and

$$A_d^{'}(x) = \frac{A_d\left((\cos(x))^{-1}\right)}{\cos\left(\frac{1}{2}(\cos(x))^{-1}\right)}$$

Again, this is a polynomial approximation problem, so the alternation theorem holds. If $E(\omega)$ has at least $L + 2 = \frac{M}{2} + 1$ alternations, the even-length symmetric filter is optimal in an L^{∞} sense.

The prototypical filter design problem:

$$W = \begin{cases} 1 & \text{if } |\omega| \leq \omega_p \\ \frac{\delta_s}{\delta_p} & \text{if } |\omega_s| \leq |\omega| \end{cases}$$

See Figure 1.9.

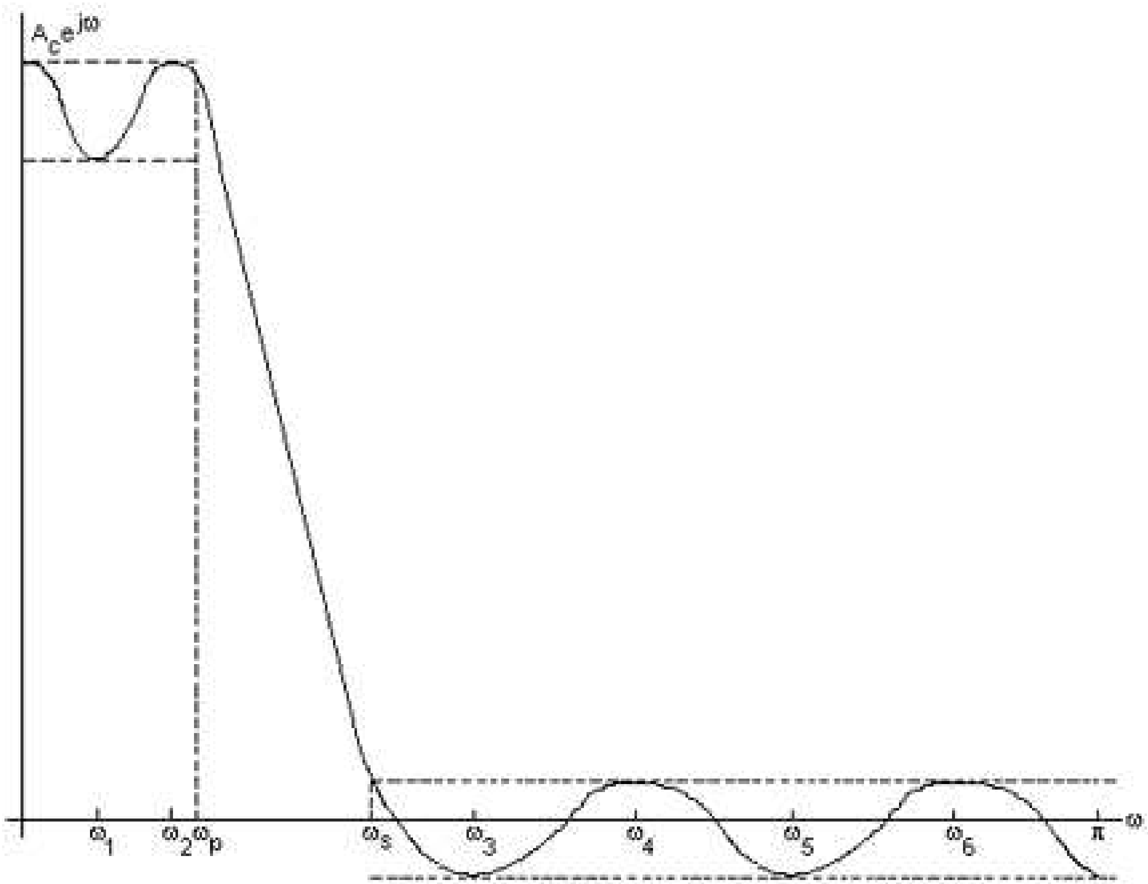


Figure 1.9

1.4.5 L-∞ Optimal Lowpass Filter Design Lemma

1. The maximum possible number of alternations for a lowpass filter is $L+3$: The proof is that the extrema of a polynomial occur only where the derivative is zero: $\frac{\partial P(x)}{\partial x} = 0$. Since $P'(x)$ is an $(L-1)$th-order polynomial, it can have at **most** $L-1$ zeros. **However**, the mapping $x = \cos(\omega)$ implies that $\frac{\partial A(\omega)}{\partial \omega} = 0$ at $\omega = 0$ and $\omega = \pi$, for two more possible alternation points. **Finally**, the band edges can also be alternations, for a total of $L - 1 + 2 + 2 = L + 3$ possible alternations.
2. There must be an alternation at either $\omega = 0$ or $\omega = \pi$.
3. Alternations must occur at ω_p and ω_s. See Figure 1.9.
4. The filter must be equiripple except at possibly $\omega = 0$ or $\omega = \pi$. Again see Figure 1.9.

NOTE: The alternation theorem doesn't directly suggest a method for computing the optimal filter. It simply tells us how to recognize that a filter **is** optimal, or **isn't** optimal. What we need is an intelligent way of guessing the optimal filter coefficients.

In matrix form, these $L+2$ simultaneous equations become

$$\begin{pmatrix} 1 & \cos(\omega_0) & \cos(2\omega_0) & ... & \cos(L\omega_0) & \frac{1}{W(\omega_0)} \\ 1 & \cos(\omega_1) & \cos(2\omega_1) & ... & \cos(L\omega_1) & \frac{-1}{W(\omega_1)} \\ \vdots & \vdots & \ddots & ... & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & ... & \ddots & \vdots \\ 1 & \cos(\omega_{L+1}) & \cos(2\omega_{L+1}) & ... & \cos(L\omega_{L+1}) & \frac{\pm(1)}{W(\omega_{L+1})} \end{pmatrix} \begin{pmatrix} h(L) \\ h(L-1) \\ \vdots \\ h(1) \\ h(0) \\ \delta \end{pmatrix} = \begin{pmatrix} A_d(\omega_0) \\ A_d(\omega_1) \\ \vdots \\ \vdots \\ \vdots \\ A_d(\omega_{L+1}) \end{pmatrix}$$

or

$$W \begin{pmatrix} h \\ \delta \end{pmatrix} = A_d$$

So, for the given set of $L+2$ extremal frequencies, we can solve for h and δ via $(h,\delta)^T = W^{-1}A_d$. Using the FFT, we can compute $A(\omega)$ of $h(n)$, on a dense set of frequencies. If the old ω_k are, in fact the extremal locations of $A(\omega)$, then the alternation theorem is satisfied and $h(n)$ is **optimal**. If not, repeat the process with the new extremal locations.

1.4.6 Computational Cost

$O(L^3)$ for the matrix inverse and $N\log_2 N$ for the FFT ($N \geq 32L$, typically), **per iteration**!

This method is expensive computationally due to the matrix inverse.

A more efficient variation of this method was developed by Parks and McClellan (1972), and is based on the Remez exchange algorithm. To understand the Remez exchange algorithm, we first need to understand Lagrange Interpoloation.

Now $A(\omega)$ is an Lth-order polynomial in $x = \cos(\omega)$, so Lagrange interpolation can be used to **exactly** compute $A(\omega)$ from $L+1$ samples of $A(\omega_k)$, $k = [0,1,2,...,L]$.

Thus, given a set of extremal frequencies and knowing δ, samples of the amplitude response $A(\omega)$ can be computed **directly** from the

$$A(\omega_k) = \frac{(-1)^{k(1)}}{W(\omega_k)}\delta + A_d(\omega_k) \tag{1.12}$$

without solving for the filter coefficients!

This leads to computational savings!

Note that (1.12) is a set of $L+2$ simultaneous equations, which can be solved for δ to obtain (Rabiner, 1975)

$$\delta = \frac{\sum_{k=0}^{L+1} \gamma_k A_d(\omega_k)}{\sum_{k=0}^{L+1} \frac{(-1)^{k(1)}\gamma_k}{W(\omega_k)}} \tag{1.13}$$

where

$$\gamma_k = \prod_{i=i\neq k,0}^{L+1} \frac{1}{\cos(\omega_k) - \cos(\omega_i)}$$

The result is the Parks-McClellan FIR filter design method, which is simply an application of the Remez exchange algorithm to the filter design problem. See Figure 1.10.

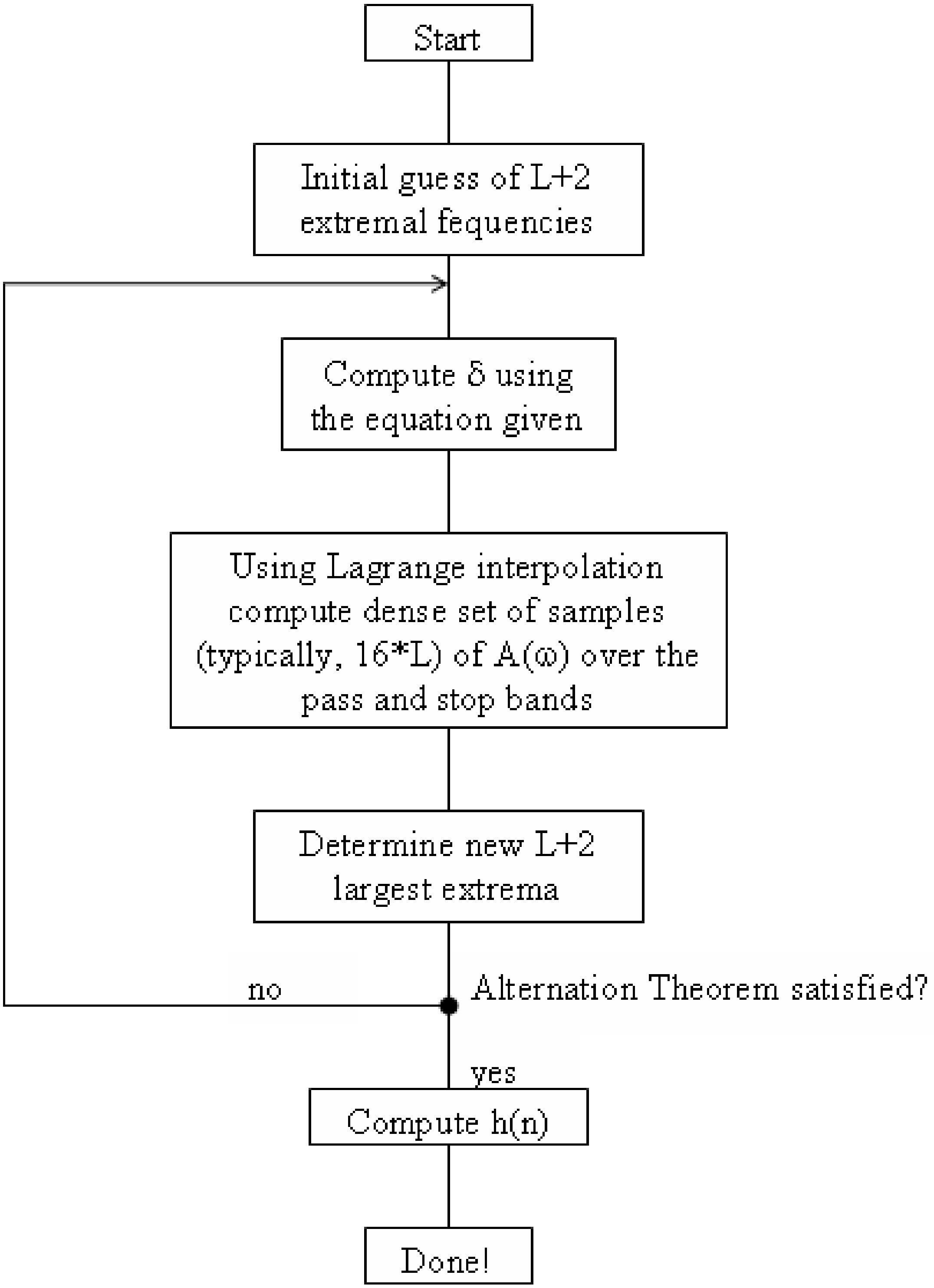


Figure 1.10: The initial guess of extremal frequencies is usually equally spaced in the band. Computing δ costs $O\left(L^2\right)$. Using Lagrange interpolation costs $O\left(16LL\right) \simeq O\left(16L^2\right)$. Computing $h\left(n\right)$ costs $O\left(L^3\right)$, but it is only done once!

The cost per iteration is $O\left(16L^2\right)$, as opposed to $O\left(L^3\right)$; much more efficient for large L. Can also interpolate to DFT sample frequencies, take inverse FFT to get corresponding filter coefficients, and zeropad and take longer FFT to efficiently interpolate.

1.5 Lagrange Interpolation[8]

Lagrange's interpolation method is a simple and clever way of finding the unique Lth-order polynomial that exactly passes through $L+1$ distinct samples of a signal. Once the polynomial is known, its value can easily be interpolated at any point using the polynomial equation. Lagrange interpolation is useful in many applications, including Parks-McClellan FIR Filter Design (Section 1.4).

1.5.1 Lagrange interpolation formula

Given an Lth-order polynomial

$$P(x) = a_0 + a_1 x + ... + a_L x^L = \sum_{k=0}^{L} a_k x^k$$

and $L+1$ values of $P(x_k)$ at different x_k, $k \in \{0, 1, ..., L\}$, $x_i \neq x_j$, $i \neq j$, the polynomial can be written as

$$P(x) = \sum_{k=0}^{L} P(x_k) \frac{(x - x_1)(x - x_2) ... (x - x_{k-1})(x - x_{k+1}) ... (x - x_L)}{(x_k - x_1)(x_k - x_2) ... (x_k - x_{k-1})(x_k - x_{k+1}) ... (x_k - x_L)}$$

The value of this polynomial at other x can be computed via substitution into this formula, or by expanding this formula to determine the polynomial coefficients a_k in standard form.

1.5.2 Proof

Note that for each term in the Lagrange interpolation formula above,

$$\prod_{i=0,,i\neq k}^{L} \frac{x - x_i}{x_k - x_i} = \begin{cases} 1 \text{ if } x = x_k \\ 0 \text{ if } (x = x_j) \wedge (j \neq k) \end{cases}$$

and that it is an Lth-order polynomial in x. The Lagrange interpolation formula is thus exactly equal to $P(x_k)$ at all x_k, and as a sum of Lth-order polynomials is itself an Lth-order polynomial.

It can be shown that the Vandermonde matrix[9]

$$\begin{pmatrix} 1 & x_0 & {x_0}^2 & ... & {x_0}^L \\ 1 & x_1 & {x_1}^2 & ... & {x_1}^L \\ 1 & x_2 & {x_2}^2 & ... & {x_2}^L \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_L & {x_L}^2 & ... & {x_L}^L \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_L \end{pmatrix} = \begin{pmatrix} P(x_0) \\ P(x_1) \\ P(x_2) \\ \vdots \\ P(x_L) \end{pmatrix}$$

has a non-zero determinant and is thus invertible, so the Lth-order polynomial passing through all $L+1$ sample points x_j is unique. Thus the Lagrange polynomial expressions, as an Lth-order polynomial passing through the $L+1$ sample points, must be the unique $P(x)$.

[8] This content is available online at <http://cnx.org/content/m12812/1.2/>.
[9] http://en.wikipedia.org/wiki/Vandermonde_matrix

Solutions to Exercises in Chapter 1

Solution to Exercise 1.1 (p. 7)
Yes; in fact it's optimal! (in a certain sense)
Solution to Exercise 1.2 (p. 8): Gibbs Phenomenon

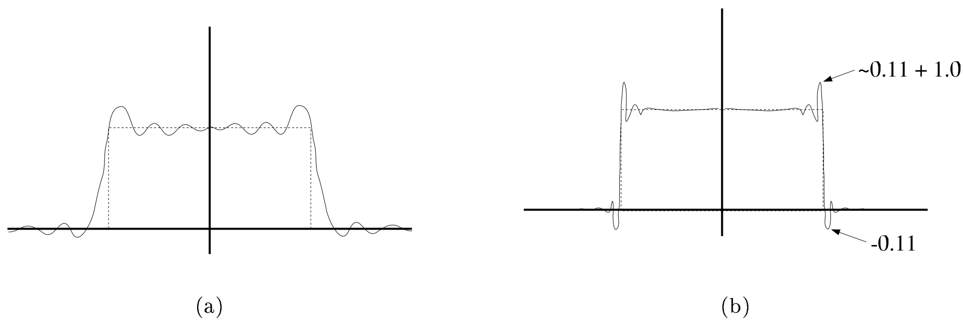


Figure 1.11: (a) $A(\omega)$, small M (b) $A(\omega)$, large M

Solution to Exercise 1.3 (p. 9)

$$\begin{cases} e^{-\left(i\omega\frac{M-1}{2}\right)} & \text{if} \quad -\omega_c \leq \omega \leq \omega_c \\ 0 \ \text{if} \ \ (-\pi \leq \omega < -\omega_c) \ \vee \ (\omega_c < \omega \leq \pi) \end{cases}$$

Solution to Exercise 1.4 (p. 12)
Yes, when the desired response is discontinuous. Since the frequency response of a finite-length filter must be continuous, without a transition band the worst-case error could be no less than half the discontinuity.
Solution to Exercise 1.5 (p. 13)
It's the same problem! To show that, consider an odd-length, symmetric linear phase filter.

$$\begin{aligned} H(\omega) &= \sum_{n=0}^{M-1} h(n)\, e^{-(i\omega n)} \\ &= e^{-\left(i\omega\frac{M-1}{2}\right)}\left(h\left(\frac{M-1}{2}\right) + 2\sum_{n=1}^{L} h\left(\frac{M-1}{2} - n\right)\cos(\omega n)\right) \end{aligned} \tag{1.14}$$

$$A(\omega) = h(L) + 2\sum_{n=1}^{L} h(L-n)\cos(\omega n) \tag{1.15}$$

Where $L \doteq \frac{M-1}{2}$.

Using trigonometric identities (such as $\cos(n\alpha) = 2\cos((n-1)\alpha)\cos(\alpha) - \cos((n-2)\alpha)$), we can rewrite $A(\omega)$ as

$$A(\omega) = h(L) + 2\sum_{n=1}^{L} h(L-n)\cos(\omega n) = \sum_{k=0}^{L} \alpha_k \cos^k(\omega)$$

where the α_k are related to the $h(n)$ by a linear transformation. Now, let $x = \cos(\omega)$. This is a one-to-one mapping from $x \in [-1, 1]$ onto $\omega \in [0, \pi]$. Thus $A(\omega)$ is an Lth-order polynomial in $x = \cos(\omega)$!

NOTE: The alternation theorem holds for the L^∞ filter design problem, too!

Therefore, to determine whether or not a length-M, odd-length, symmetric linear-phase filter is optimal in an L^∞ sense, simply count the alternations in $E(\omega) = W(\omega)(A_d(\omega) - A(\omega))$ in the pass and stop bands. If there are $L + 2 = \frac{M+3}{2}$ or more alternations, $h(n)$, $0 \leq n \leq M-1$ is the optimal filter!

Chapter 2

IIR Filter Design

2.1 Overview of IIR Filter Design[1]

2.1.1 IIR Filter

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{M-1} b_k x(n-k)$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_M z^{-M}}$$

2.1.2 IIR Filter Design Problem

Choose $\{a_i\}$, $\{b_i\}$ to **best** approximate some desired $|H_d(w)|$ or, (occasionally), $H_d(w)$.

As before, different design techniques will be developed for different approximation criteria.

2.1.3 Outline of IIR Filter Design Material

- **Bilinear Transform** - Maps $\| L \|_\infty$ optimal (and other) **analog** filter designs to $\| L \|_\infty$ optimal digital IIR filter designs.
- **Prony's Method** - Quasi-$\| L \|_2$ optimal method for time-domain fitting of a desired impulse response (*ad hoc*).
- **Lp Optimal Design** - $\| L \|_p$ optimal filter design ($1 < p < \infty$) using non-linear optimization techniques.

2.1.4 Comments on IIR Filter Design Methods

The bilinear transform method is used to design "typical" $\| L \|_\infty$ magnitude optimal filters. The $\| L \|_p$ optimization procedures are used to design filters for which classical analog prototype solutions don't exist. The program by Deczky (*DSP Programs Book*, IEEE Press) is widely used. Prony/Linear Prediction techniques are used often to obtain initial guesses, and are almost exclusively used in data modeling, system identification, and most applications involving the fitting of real data (for example, the impulse response of an unknown filter).

[1] This content is available online at <http://cnx.org/content/m12758/1.2/>.

2.2 Prototype Analog Filter Design[2]

2.2.1 Analog Filter Design

Laplace transform:

$$H(s) = \int_{-\infty}^{\infty} h_a(t) e^{-(st)} dt$$

Note that the continuous-time Fourier transform[3] is $H(i\lambda)$ (the Laplace transform evaluated on the imaginary axis).

Since the early 1900's, there has been a lot of research on designing analog filters of the form

$$H(s) = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_M s^M}{1 + a_1 s + a_2 s^2 + \ldots + a_M s^M}$$

A causal IIR filter **cannot** have linear phase (no possible symmetry point), and design work for analog filters has concentrated on designing filters with equiriplle ($\| L \|_\infty$) **magnitude** responses. These design problems have been solved. We will not concern ourselves here with the design of the analog prototype filters, only with how these designs are mapped to discrete-time while preserving optimality.

An analog filter with **real** coefficients must have a magnitude response of the form

$$(|H(\lambda)|)^2 = B(\lambda^2)$$

$$\begin{aligned} H(i\lambda)\overline{H(i\lambda)} &= \frac{b_0 + b_1 i\lambda + b_2 (i\lambda)^2 + b_3 (i\lambda)^3 + \ldots}{1 + a_1 i\lambda + a_2 (i\lambda)^2 + \ldots} \overline{H(i\lambda)} \\ &= \frac{b_0 - b_2\lambda^2 + b_4\lambda^4 + \ldots + i\lambda\left(b_1 - b_3\lambda^2 + b_5\lambda^4 + \ldots\right)}{1 - a_2\lambda^2 + a_4\lambda^4 + \ldots + i\lambda(a_1 - a_3\lambda^2 + a_5\lambda^4 + \ldots)} \overline{\frac{b_0 - b_2\lambda^2 + b_4\lambda^4 + \ldots + i\lambda(b_1 - b_3\lambda^2 + b_5\lambda^4 + \ldots)}{1 - a_2\lambda^2 + a_4\lambda^4 + \ldots + i\lambda(a_1 - a_3\lambda^2 + a_5\lambda^4 + \ldots)}} \\ &= \frac{\left(b_0 - b_2\lambda^2 + b_4\lambda^4 + \ldots\right)^2 + \lambda^2\left(b_1 - b_3\lambda^2 + b_5\lambda^4 + \ldots\right)^2}{(1 - a_2\lambda^2 + a_4\lambda^4 + \ldots)^2 + \lambda^2(a_1 - a_3\lambda^2 + a_5\lambda^4 + \ldots)^2} \\ &= B(\lambda^2) \end{aligned} \quad (2.1)$$

Let $s = i\lambda$, note that the poles and zeros of $B(-s^2)$ are symmetric around **both** the real and imaginary axes: that is, a pole at p_1 implies poles at p_1, $\overline{p_1}$, $-p_1$, and $-\overline{p_1}$, as seen in Figure 2.1 (s-plane).

[2] This content is available online at <http://cnx.org/content/m12763/1.2/>.

[3] "Continuous Time Fourier Transform (CTFT)" <http://cnx.org/content/m10098/latest/>

s-plane

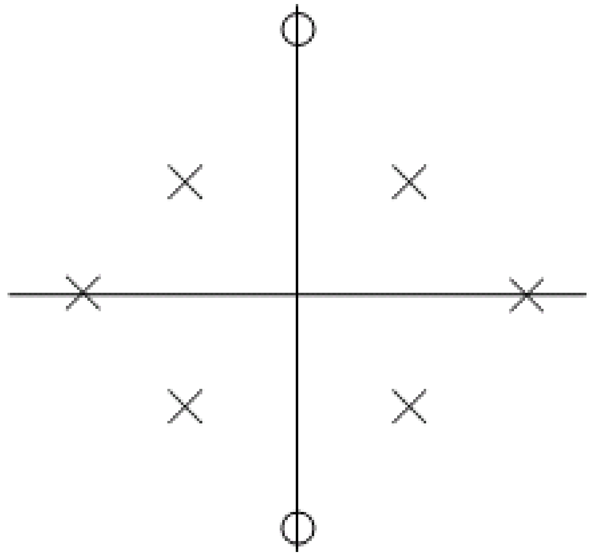


Figure 2.1

Recall that an analog filter is stable and causal if all the poles are in the left half-plane, LHP, and is **minimum phase** if all zeros and poles are in the LHP.

$s = i\lambda$: $B(\lambda^2) = B(-s^2) = H(s)H(-s) = H(i\lambda)H(-(i\lambda)) = H(i\lambda)\overline{H(i\lambda)}$ we can factor $B(-s^2)$ into $H(s)H(-s)$, where $H(s)$ has the left half plane poles and zeros, and $H(-s)$ has the RHP poles and zeros.

$(|H(s)|)^2 = H(s)H(-s)$ for $s = i\lambda$, so $H(s)$ has the magnitude response $B(\lambda^2)$. The trick to analog filter design is to design a good $B(\lambda^2)$, then factor this to obtain a filter with that **magnitude** response.

The traditional analog filter designs all take the form $B(\lambda^2) = (|H(\lambda)|)^2 = \frac{1}{1+F(\lambda^2)}$, where F is a rational function in λ^2.

Example 2.1

$$B(\lambda^2) = \frac{2+\lambda^2}{1+\lambda^4}$$

$$B(-s^2) = \frac{2-s^2}{1+s^4} = \frac{(\sqrt{2}-s)(\sqrt{2}+s)}{(s+\alpha)(s-\alpha)(s+\overline{\alpha})(s-\overline{\alpha})}$$

where $\alpha = \frac{1+i}{\sqrt{2}}$.

NOTE: Roots of $1+s^N$ are N points equally spaced around the unit circle (Figure 2.2).

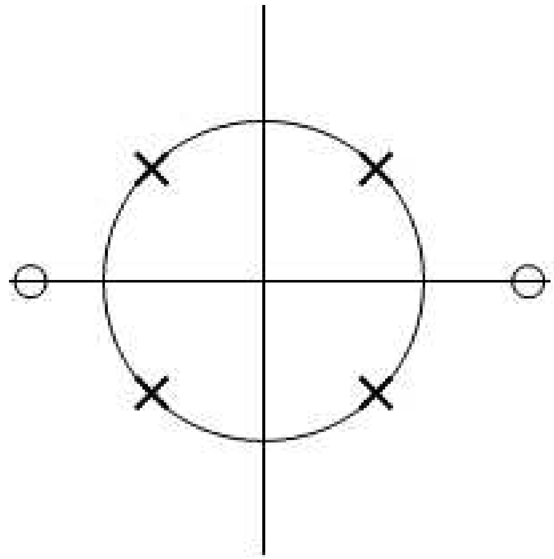


Figure 2.2

Take $H(s) =$ LHP factors:

$$H(s) = \frac{\sqrt{2}+s}{(s+\alpha)(s+\overline{\alpha})} = \frac{\sqrt{2}+s}{s^2+\sqrt{2}s+1}$$

2.2.2 Traditional Filter Designs

2.2.2.1 Butterworth

$$B\left(\lambda^2\right) = \frac{1}{1+\lambda^{2M}}$$

NOTE: Remember this for homework and rest problems!

"Maximally smooth" at $\lambda = 0$ and $\lambda = \infty$ (maximum possible number of zero derivatives). Figure 2.3.

$$B\left(\lambda^2\right) = \left(|H(\lambda)|\right)^2$$

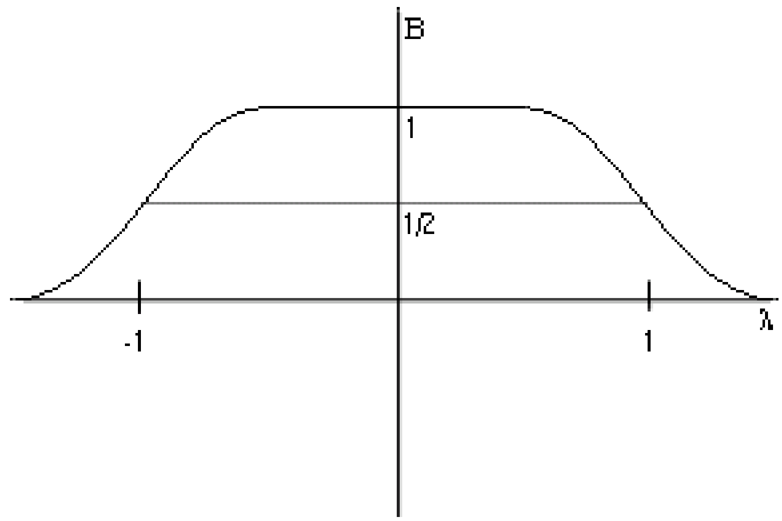
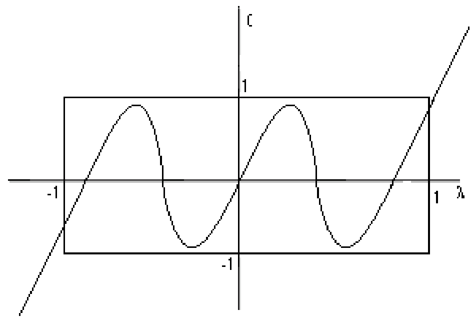


Figure 2.3

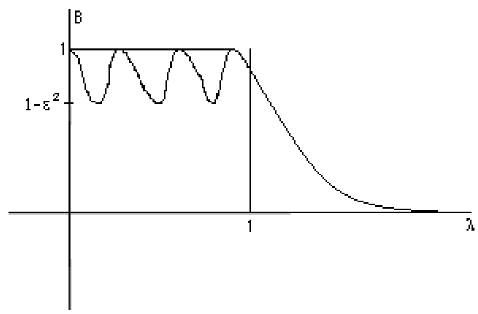
2.2.2.2 Chebyshev

$$B\left(\lambda^2\right) = \frac{1}{1+\epsilon^2 {C_M}^2\left(\lambda\right)}$$

where ${C_M}^2\left(\lambda\right)$ is an M^{th} order Chebyshev polynomial. Figure 2.4.



(a)



(b)

Figure 2.4

2.2.2.3 Inverse Chebyshev

Figure 2.5.

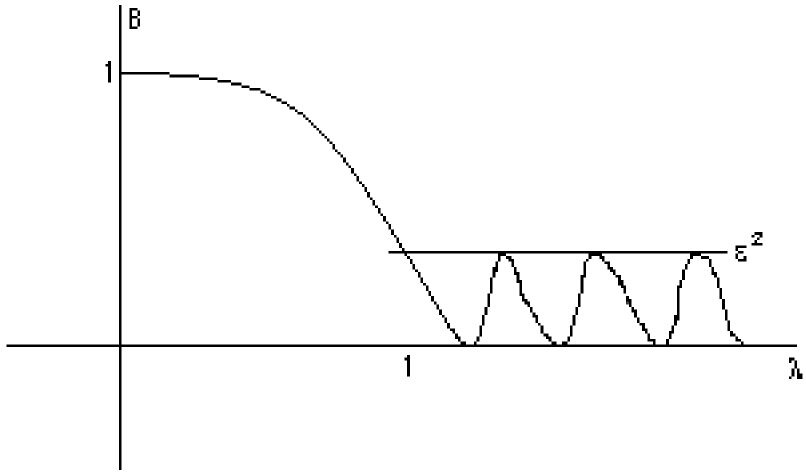


Figure 2.5

2.2.2.4 Elliptic Function Filter (Cauer Filter)

$$B\left(\lambda^2\right) = \frac{1}{1 + \epsilon^2 {J_M}^2\left(\lambda\right)}$$

where J_M is the "Jacobi Elliptic Function." Figure 2.6.

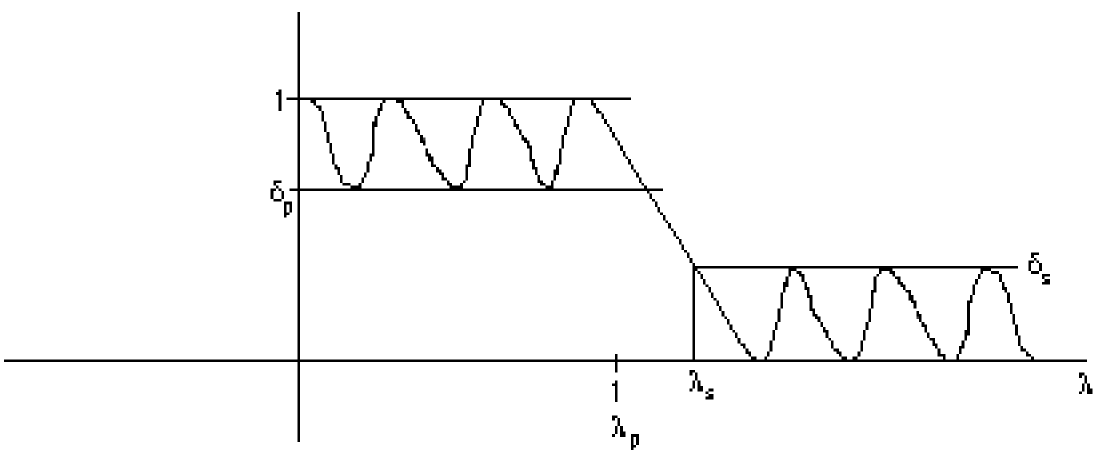


Figure 2.6

The Cauer filter is $\| L \|_\infty$ optimum in the sense that for a given M, δ_p, δ_s, and λ_p, the transition bandwidth is smallest.

That is, it is $\parallel L \parallel_\infty$ optimal.

2.3 IIR Digital Filter Design via the Bilinear Transform[4]

A **bilinear transform** maps an analog filter $H_a(s)$ to a discrete-time filter $H(z)$ of the same order.

If only we could somehow map these optimal analog filter designs to the digital world while preserving the magnitude response characteristics, we could make use of the already-existing body of knowledge concerning optimal analog filter design.

2.3.1 Bilinear Transformation

The Bilinear Transform is a nonlinear $\mathbb{C} \to \mathbb{C}$ mapping that maps a function of the complex variable s to a function of a complex variable z. This map has the property that the LHP in s ($\Re(s) < 0$) maps to the interior of the unit circle in z, and the $i\lambda = s$ axis maps to the unit circle $e^{i\omega}$ in z.

Bilinear transform:

$$s = \alpha \frac{z-1}{z+1}$$

$$H(z) = H_a\left(s = \alpha \frac{z-1}{z+1}\right)$$

NOTE: $i\lambda = \alpha \frac{e^{i\omega}-1}{e^{i\omega}+1} = \alpha \frac{\left(e^{i\omega}-1\right)\left(e^{-(i\omega)}+1\right)}{\left(e^{i\omega}+1\right)\left(e^{-(i\omega)}+1\right)} = \frac{2i\sin(\omega)}{2+2\cos(\omega)} = i\alpha\tan\left(\frac{\omega}{2}\right)$, so $\lambda \equiv \alpha\tan\left(\frac{\omega}{2}\right)$, $\omega \equiv 2\arctan\left(\frac{\lambda}{\alpha}\right)$. Figure 2.7.

[4] This content is available online at <http://cnx.org/content/m12757/1.2/>.

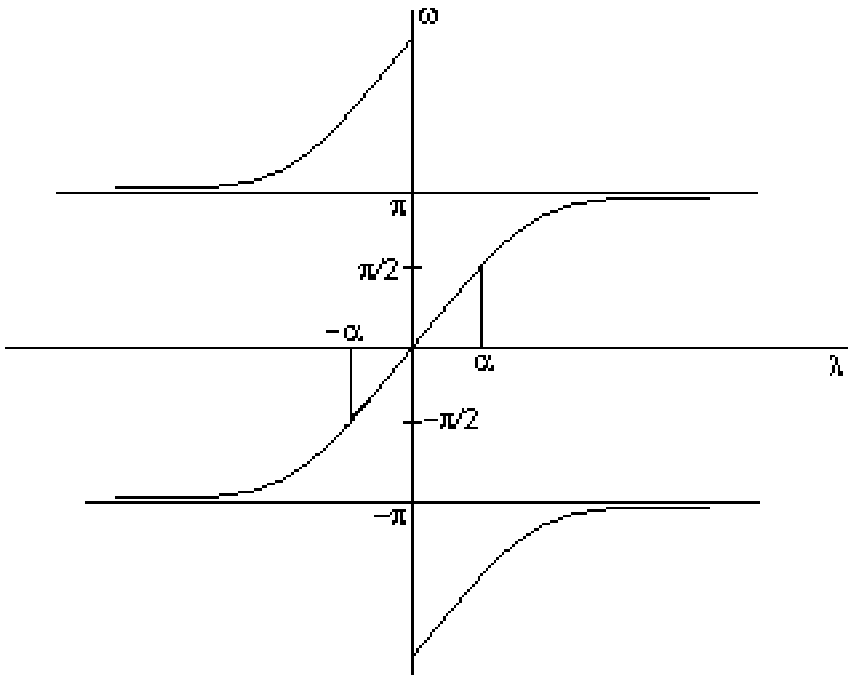
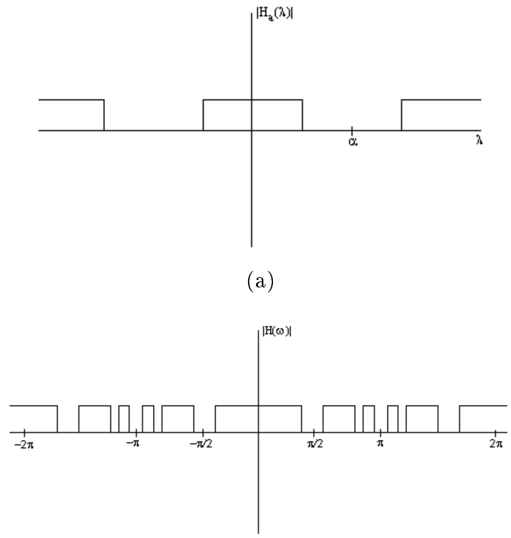


Figure 2.7

The magnitude response doesn't change in the mapping from λ to ω, it is simply warped nonlinearly according to $H(\omega) = H_a\left(\alpha \tan\left(\frac{\omega}{2}\right)\right)$, Figure 2.8.



(a)

(b)

Figure 2.8: The first image implies the second one.

NOTE: This mapping preserves $\| L \|_\infty$ errors in (warped) frequency bands. Thus optimal Cauer ($\| L \|_\infty$) filters in the analog realm can be mapped to $\| L \|_\infty$ optimal discrete-time IIR filters using the bilinear transform! This is how IIR filters with $\| L \|_\infty$ optimal magnitude responses are designed.

NOTE: The parameter α provides one degree of freedom which can be used to map a single λ_0 to any desired ω_0:

$$\lambda_0 = \alpha \tan\left(\frac{\omega_0}{2}\right)$$

or

$$\alpha = \frac{\lambda_0}{\tan\left(\frac{\omega_0}{2}\right)}$$

This can be used, for example, to map the pass-band edge of a lowpass analog prototype filter to any desired pass-band edge in ω. Often, analog prototype filters will be designed with $\lambda = 1$ as a band edge, and α will be used to locate the band edge in ω. Thus an M^{th} order optimal lowpass analog filter prototype can be used to design **any** M^{th} order discrete-time lowpass IIR filter with the same ripple specifications.

2.3.2 Prewarping

Given specifications on the frequency response of an IIR filter to be designed, map these to specifications in the analog frequency domain which are equivalent. Then a satisfactory analog prototype can be designed which, when transformed to discrete-time using the bilinear transformation, will meet the specifications.

Example 2.2
The goal is to design a high-pass filter, $\omega_s = \omega_s$, $\omega_p = \omega_p$, $\delta_s = \delta_s$, $\delta_p = \delta_p$; pick up some $\alpha = \alpha_0$. In Figure 2.9 the δ_i remain the same and the band edges are mapped by $\lambda_i = \alpha_0 \tan\left(\frac{\omega_i}{2}\right)$.

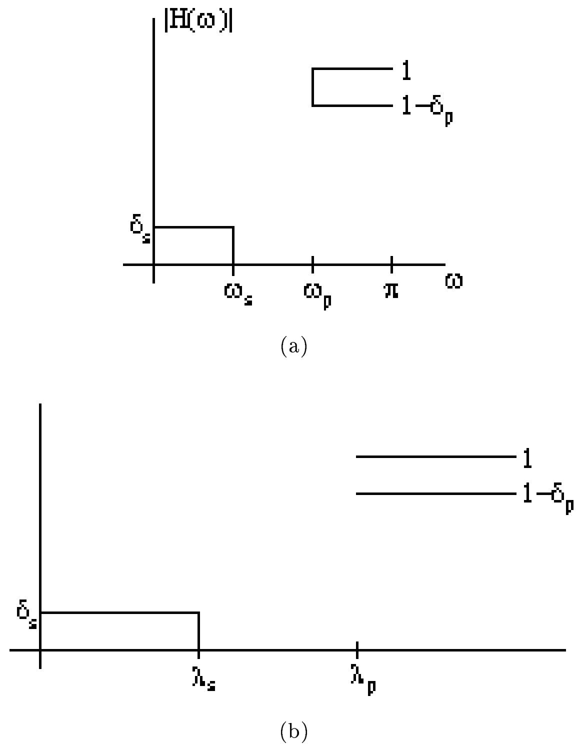


Figure 2.9: Where $\lambda_s = \alpha_0 \tan\left(\frac{\omega_s}{2}\right)$ and $\lambda_p = \alpha_0 \tan\left(\frac{\omega_p}{2}\right)$.

2.4 Impulse-Invariant Design[5]

Pre-classical, adhoc-but-easy method of converting an analog prototype filter to a digital IIR filter. Does not preserve any optimality.

Impulse invariance means that digital filter impulse response exactly equals samples of the analog prototype impulse response:

$$\forall n : (h(n) = h_a(nT))$$

How is this done?

The impulse response of a causal, stable analog filter is simply a sum of decaying exponentials:

$$H_a(s) = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_p s^p}{1 + a_1 s + a_2 s^2 + \ldots + a_p s^p} = \frac{A_1}{s - s_1} + \frac{A_2}{s - s_2} + \ldots + \frac{A_p}{s - s_p}$$

which implies

$$h_a(t) = \left(A_1 e^{s_1 t} + A_2 e^{s_2 t} + \ldots + A_p e^{s_p t}\right) u(t)$$

[5]This content is available online at <http://cnx.org/content/m12760/1.2/>.

For impulse invariance, we desire

$$h(n) = h_a(nT) = \left(A_1 e^{s_1 nT} + A_2 e^{s_2 nT} + \dots + A_p e^{s_p nT}\right) u(n)$$

Since

$$A_k e^{(s_k T)n} u(n) \equiv \frac{A_k z}{z - e^{s_k T}}$$

where $|z| > |e^{s_k T}|$, and

$$H(z) = \sum_{k=1}^{p} A_k \frac{z}{z - e^{s_k T}}$$

where $|z| > max_k \left\{k, |e^{s_k T}|\right\}$.

This technique is used occasionally in digital simulations of analog filters.

Exercise 2.1 ***(Solution on p. 38.)***

What is the main problem/drawback with this design technique?

2.5 Digital-to-Digital Frequency Transformations[6]

Given a prototype **digital** filter design, transformations similar to the bilinear transform can also be developed.

Requirements on such a mapping $z^{-1} = g\left(z^{-1}\right)$:

1. points inside the unit circle stay inside the unit circle (condition to preserve stability)
2. unit circle is mapped to itself (preserves frequency response)

This condition (list, item 2, p. 32) implies that $e^{-(i\omega_1)} = g\left(e^{-(i\omega)}\right) = |g(\omega)| e^{i\angle(g(\omega))}$ requires that $|g\left(e^{-(i\omega)}\right)| = 1$ on the unit circle!

Thus we require an **all-pass** transformation:

$$g\left(z^{-1}\right) = \prod_{k=1}^{p} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}}$$

where $|\alpha_K| < 1$, which is required to satisfy this condition (list, item 1, p. 32).

Example 2.3: Lowpass-to-Lowpass

$${z_1}^{-1} = \frac{z^{-1} - a}{1 - a z^{-1}}$$

which maps original filter with a cutoff at ω_c to a new filter with cutoff ω_c',

$$a = \frac{\sin\left(\frac{1}{2}\left(\omega_c - \omega_c'\right)\right)}{\sin\left(\frac{1}{2}\left(\omega_c + \omega_c'\right)\right)}$$

Example 2.4: Lowpass-to-Highpass

$${z_1}^{-1} = \frac{z^{-1} + a}{1 + a z^{-1}}$$

which maps original filter with a cutoff at ω_c to a frequency reversed filter with cutoff ω_c',

$$a = \frac{\cos\left(\frac{1}{2}\left(\omega_c - \omega_c'\right)\right)}{\cos\left(\frac{1}{2}\left(\omega_c + \omega_c'\right)\right)}$$

(Interesting and occasionally useful!)

[6] This content is available online at <http://cnx.org/content/m12759/1.2/>.

2.6 Prony's Method[7]

Prony's Method is a quasi-least-squares time-domain IIR filter design method.

First, assume $H(z)$ is an "all-pole" system:

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{M} a_k z^{-k}} \tag{2.2}$$

and

$$h(n) = -\sum_{k=1}^{M} a_k h(n-k) + b_0 \delta(n)$$

where $h(n) = 0$, $n < 0$ for a causal system.

NOTE: For $h = 0$, $h(0) = b_0$.

Let's attempt to fit a desired impulse response (let it be **causal**, although one can extend this technique when it isn't) $h_d(n)$.

A true least-squares solution would attempt to minimize

$$\epsilon^2 = \sum_{n=0}^{\infty} \left(|h_d(n) - h(n)|\right)^2$$

where $H(z)$ takes the form in (2.2). This is a difficult non-linear optimization problem which is known to be plagued by local minima in the error surface. So instead of solving this difficult non-linear problem, we solve the **deterministic linear prediction** problem, which is related to, **but not the same as**, the true least-squares optimization.

The deterministic linear prediction problem is a **linear** least-squares optimization, which is easy to solve, but it minimizes the **prediction** error, not the $(|\text{desired} - \text{actual}|)^2$ response error.

Notice that for $n > 0$, with the all-pole filter

$$h(n) = -\sum_{k=1}^{M} a_k h(n-k) \tag{2.3}$$

the right hand side of this equation (2.3) is a **linear predictor** of $h(n)$ in terms of the M previous samples of $h(n)$.

For the desired reponse $h_d(n)$, one can choose the recursive filter coefficients a_k to minimize the squared prediction error

$${\epsilon_p}^2 = \sum_{n=1}^{\infty} \left(|h_d(n) + \sum_{k=1}^{M} a_k h_d(n-k)|\right)^2$$

where, in practice, the ∞ is replaced by an N.

In matrix form, that's

$$\begin{pmatrix} h_d(0) & 0 & \dots & 0 \\ h_d(1) & h_d(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_d(N-1) & h_d(N-2) & \dots & h_d(N-M) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{pmatrix} \simeq - \begin{pmatrix} h_d(1) \\ h_d(2) \\ \vdots \\ h_d(N) \end{pmatrix}$$

or

$$H_d a \simeq -h_d$$

[7]This content is available online at <http://cnx.org/content/m12762/1.2/>.

The optimal solution is

$$a_{\mathrm{lp}} = -\left(\left({H_d}^H H_d \right)^{-1} {H_d}^H h_d \right)$$

Now suppose $H(z)$ is an M^{th}-order IIR (ARMA) system,

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}$$

or

$$\begin{aligned} h(n) &= -\sum_{k=1}^{M} a_k h(n-k) + \sum_{k=0}^{M} b_k \delta(n-k) \\ &= \begin{cases} -\sum_{k=1}^{M} a_k h(n-k) + b_n & \text{if } \ 0 \le n \le M \\ -\sum_{k=1}^{M} a_k h(n-k) & \text{if } \ n > M \end{cases} \end{aligned} \tag{2.4}$$

For $n > M$, this is just like the all-pole case, so we can solve for the best predictor coefficients as before:

$$\begin{pmatrix} h_d(M) & h_d(M-1) & \dots & h_d(1) \\ h_d(M+1) & h_d(M) & \dots & h_d(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_d(N-1) & h_d(N-2) & \dots & h_d(N-M) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{pmatrix} \simeq \begin{pmatrix} h_d(M+1) \\ h_d(M+2) \\ \vdots \\ h_d(N) \end{pmatrix}$$

or

$$H_d \, a \simeq h_d$$

and

$$a_{\mathrm{opt}} = \left(\left(H_d \right)^H H_d \right)^{-1} {H_d}^H h_d$$

Having determined the a's, we can use them in (2.4) to obtain the b_n's:

$$b_n = \sum_{k=1}^{M} a_k h_d(n-k)$$

where $h_d(n-k) = 0$ for $n - k < 0$.

For $N = 2M$, H_d is square, and we can solve **exactly** for the a_k's with no error. The b_k's are also chosen such that there is no error in the first $M+1$ samples of $h(n)$. Thus for $N = 2M$, the first $2M+1$ points of $h(n)$ **exactly equal** $h_d(n)$. This is called **Prony's Method**. Baron de Prony invented this in 1795.

For $N > 2M$, $h_d(n) = h(n)$ for $0 \le n \le M$, the prediction error is minimized for $M + 1 < n \le N$, and whatever for $n \ge N + 1$. This is called the **Extended Prony Method**.

One might prefer a method which tries to minimize an overall error with the numerator coefficients, rather than just using them to exactly fit $h_d(0)$ to $h_d(M)$.

2.6.1 Shank's Method

1. Assume an all-pole model and fit $h_d(n)$ by minimizing the prediction error $1 \le n \le N$.
2. Compute $v(n)$, the impulse response of this all-pole filter.
3. Design an all-zero (MA, FIR) filter which fits $v(n) * h_z(n) \simeq h_d(n)$ optimally in a least-squares sense (Figure 2.10).

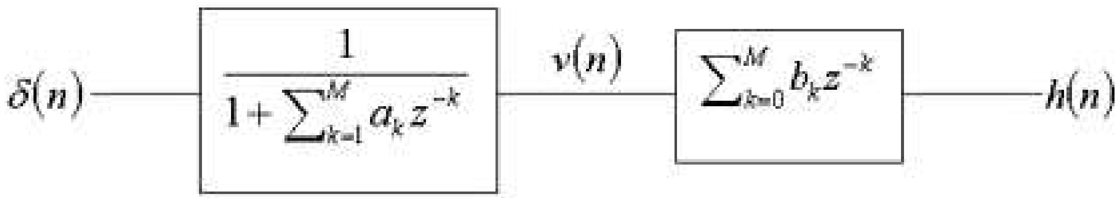


Figure 2.10: Here, $h(n) \simeq h_d(n)$.

The final IIR filter is the cascade of the all-pole and all-zero filter.

This (list, item 3, p. 34) is is solved by

$$min_{\mathrm{b\ k}} \left\{ \mathrm{b\ k}, \sum_{n=0}^{N} \left(|h_d(n) - \sum_{k=0}^{M} b_k v(n-k)| \right)^2 \right\}$$

or in matrix form

$$\begin{pmatrix} v(0) & 0 & 0 & ... & 0 \\ v(1) & v(0) & 0 & ... & 0 \\ v(2) & v(1) & v(0) & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v(N) & v(N-1) & v(N-2) & ... & v(N-M) \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_M \end{pmatrix} \simeq \begin{pmatrix} h_d(0) \\ h_d(1) \\ h_d(2) \\ \vdots \\ h_d(N) \end{pmatrix}$$

Which has solution:

$$b_{\mathrm{opt}} = \left(V^H V\right)^{-1} V^H h$$

Notice that none of these methods solve the true least-squares problem:

$$min_{a,b} \left\{ a, , , b, \sum_{n=0}^{\infty} \left(|h_d(n) - h(n)| \right)^2 \right\}$$

which is a difficult non-linear optimization problem. The true least-squares problem can be written as:

$$min_{\alpha,\beta} \left\{ \alpha, , , \beta, \sum_{n=0}^{\infty} \left(|h_d(n) - \sum_{i=1}^{M} \alpha_i e^{\beta_i n}| \right)^2 \right\}$$

since the impulse response of an IIR filter is a sum of exponentials, and non-linear optimization is then used to solve for the α_i and β_i.

2.7 Linear Prediction[8]

Recall that for the all-pole design problem, we had the overdetermined set of linear equations:

$$\begin{pmatrix} h_d(0) & 0 & \dots & 0 \\ h_d(1) & h_d(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_d(N-1) & h_d(N-2) & \dots & h_d(N-M) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{pmatrix} \simeq - \begin{pmatrix} h_d(1) \\ h_d(2) \\ \vdots \\ h_d(N) \end{pmatrix}$$

with solution $a = \left({H_d}^H H_d\right)^{-1} {H_d}^H h_d$

Let's look more closely at ${H_d}^H H_d = R$. r_{ij} is related to the **correlation** of h_d with itself:

$$r_{ij} = \sum_{k=0}^{N-max\{i,j\}} h_d(k)\, h_d(k + |i-j|)$$

Note also that:

$${H_d}^H h_d = \begin{pmatrix} r_d(1) \\ r_d(2) \\ r_d(3) \\ \vdots \\ r_d(M) \end{pmatrix}$$

where

$$r_d(i) = \sum_{n=0}^{N-i} h_d(n)\, h_d(n+i)$$

so this takes the form $a_{\text{opt}} = -\left(R^H r_d\right)$, or $Ra = -r$, where R is $M \times M$, a is $M \times 1$, and r is also $M \times 1$.

Except for the changing endpoints of the sum, $r_{ij} \simeq r(i-j) = r(j-i)$. If we tweak the problem slightly to make $r_{ij} = r(i-j)$, we get:

$$\begin{pmatrix} r(0) & r(1) & r(2) & \dots & r(M-1) \\ r(1) & r(0) & r(1) & \dots & \vdots \\ r(2) & r(1) & r(0) & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(M-1) & \dots & \dots & \dots & r(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_M \end{pmatrix} = - \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(M) \end{pmatrix}$$

The matrix R is **Toeplitz** (diagonal elements equal), and a can be solved for with $O\left(M^2\right)$ computations using Levinson's recursion.

2.7.1 Statistical Linear Prediction

Used very often for forecasting (e.g. stock market).

Given a time-series $y(n)$, assumed to be produced by an auto-regressive (AR) (all-pole) system:

$$y(n) = -\sum_{k=1}^{M} a_k y(n-k) + u(n)$$

[8] This content is available online at <http://cnx.org/content/m12761/1.2/>.

where $u(n)$ is a white Gaussian noise sequence which is stationary and has zero mean.

To determine the model parameters $\{a_k\}$ minimizing the variance of the prediction error, we seek

$$min_{\text{a k}} \left\{ \text{a k}, E\left[\left(y(n) + \sum_{k=1}^{M} a_k y(n-k)\right)^2\right]\right\} = min_{\text{a k}} \left\{ \text{a k}, E\left[y^2(n) + 2\sum_{k=1}^{M} a_k y(n) y(n-k) + \sum_{k=}^{M} \right.\right. \tag{2.5}$$

$$min_{\text{a k}} \left\{ \text{a k}, E\left[y^2(n)\right] + 2\sum_{k=1}^{M} a_k E\left[y(n) y(n-k)\right] + \sum_{k=1}^{M} \sum_{l=1}^{M} a_k a_l E\left[y(n-k) y(n-l)\right]\right\}$$

NOTE: The mean of $y(n)$ is zero.

$$\epsilon^2 = r(0) + 2\begin{pmatrix} r(1) & r(2) & r(3) & \ldots & r(M) \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_M \end{pmatrix} + \tag{2.6}$$

$$\begin{pmatrix} a_1 & a_2 & a_3 & \ldots & a_M \end{pmatrix}\begin{pmatrix} r(0) & r(1) & r(2) & \ldots & r(M-1) \\ r(1) & r(0) & r(1) & \ldots & \vdots \\ r(2) & r(1) & r(0) & \ldots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(M-1) & \ldots & \ldots & \ldots & r(0) \end{pmatrix}$$

$$\frac{\partial \epsilon^2}{\partial a} = 2r + 2Ra \tag{2.7}$$

Setting (2.7) equal to zero yields: $Ra = -r$ These are called the **Yule-Walker** equations. In practice, given samples of a sequence $y(n)$, we estimate $r(n)$ as

$$r(n) = \frac{1}{N}\sum_{k=0}^{N-n} y(n) y(n+k) \simeq E\left[y(k) y(n+k)\right]$$

which is extremely similar to the deterministic least-squares technique.

Solutions to Exercises in Chapter 2

Solution to Exercise 2.1 (p. 32)

Since it samples the non-bandlimited impulse response of the analog prototype filter, the frequency response **aliases**. This distorts the original analog frequency and destroys any optimal frequency properties in the resulting digital filter.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

A alias, § 2.4(31)
aliases, 38
all-pass, 32
amplitude of the frequency response, 6

B bilinear transform, 28

D design, § 1.4(11)
deterministic linear prediction, 33
digital, § 1.4(11)
DSP, § 1.2(7), § 1.3(9), § 1.4(11)

E Extended Prony Method, 34

F filter, § 1.3(9), § 1.4(11)
filters, § 1.2(7)
FIR, § 1.2(7), § 1.3(9), § 1.4(11)
FIR filter, § (1)
FIR filter design, § 1.1(3)

G Generalized Linear Phase, 6

I IIR Filter, § (1)
interpolation, § 1.5(18)

L lagrange, § 1.5(18)
linear predictor, 33

M minimum phase, 23

O optimal, 16

P polynomial, § 1.5(18)
pre-classical, § 2.4(31), 31
Prony's Method, 34

T Toeplitz, 36

Y Yule-Walker, 37

Attributions

Collection: *Digital Filter Design*
Edited by: Douglas L. Jones
URL: http://cnx.org/content/col10285/1.1/

Module: "Overview of Digital Filter Design"
By: Douglas L. Jones
URL: http://cnx.org/content/m12776/1.2/
Page: 1

Module: "Linear Phase Filters"
By: Douglas L. Jones
URL: http://cnx.org/content/m12802/1.2/
Pages: 3-7

Module: "Window Design Method"
By: Douglas L. Jones
URL: http://cnx.org/content/m12790/1.2/
Pages: 7-8

Module: "Frequency Sampling Design Method for FIR filters"
By: Douglas L. Jones
URL: http://cnx.org/content/m12789/1.2/
Pages: 9-11

Module: "Parks-McClellan FIR Filter Design"
By: Douglas L. Jones
URL: http://cnx.org/content/m12799/1.3/
Pages: 11-18

Module: "Lagrange Interpolation"
By: Douglas L. Jones
URL: http://cnx.org/content/m12812/1.2/
Page: 18

Module: "Overview of IIR Filter Design"
By: Douglas L. Jones
URL: http://cnx.org/content/m12758/1.2/
Page: 21
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "Prototype Analog Filter Design"
By: Douglas L. Jones
URL: http://cnx.org/content/m12763/1.2/
Pages: 22-28
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "IIR Digital Filter Design via the Bilinear Transform"
By: Douglas L. Jones
URL: http://cnx.org/content/m12757/1.2/
Pages: 28-31
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "Impulse-Invariant Design"
By: Douglas L. Jones
URL: http://cnx.org/content/m12760/1.2/
Pages: 31-32
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "Digital-to-Digital Frequency Transformations"
By: Douglas L. Jones
URL: http://cnx.org/content/m12759/1.2/
Page: 32
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "Prony's Method"
By: Douglas L. Jones
URL: http://cnx.org/content/m12762/1.2/
Pages: 33-35
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Module: "Linear Prediction"
By: Douglas L. Jones
URL: http://cnx.org/content/m12761/1.2/
Pages: 36-37
Copyright: Douglas L. Jones
License: http://creativecommons.org/licenses/by/2.0/

Digital Filter Design

An electrical engineering course on digital filter design.

About Connexions