

LINEAR-PHASE FIR FILTER DESIGN BY LEAST SQUARES

This section describes how to design linear-phase FIR filters based on the square error criterion. Recall that the interpolation approach to filter design allows one to specify the frequency response only at a finite number of points. However, the square error criterion can include the entire frequency band.

We will see that FIR filters that minimize the square error can be found by solving a linear system of equations. This technique is straight-forward and is applicable to arbitrary desired frequency responses. In addition, linear constraints on the coefficients are easily included.

1. Derivation
 - (a) Structure of the matrix Q .
 - (b) Relation to the DTFT.
 - (c) Special case: $W(\omega) = 1$
 - (d) Low-pass: Weighted square error
2. Square Error with Constraints
 - (a) Low-pass with specified null
 - (b) Notch Filter
3. Discrete Square Error
4. Pros and Cons

DERIVATION

The weighted integral square error (or “ L_2 error”) is defined by

$$\mathcal{E}_2 = \int_0^\pi W(\omega) (A(\omega) - D(\omega))^2 d\omega \quad (1)$$

where

$A(\omega)$: the actual amplitude response

$D(\omega)$: the ideal amplitude response

$W(\omega)$: nonnegative weighting function

The weighting function can be used to assign more importance to specific parts of the frequency response. For example, it is common to weight the stop-band more heavily than the pass-band.

Once the length N and the Type of the filter (I, II, III, or IV) is chosen, the goal is to find the filter coefficients $h(n)$ that minimizes \mathcal{E}_2 . To develop this approach, we will consider the design of Type I FIR filters. The design of the other 3 FIR filter types can be developed similarly.

Recall that for a Type I FIR filter,

$$A(\omega) = \sum_{n=0}^M a(n) \cos(n\omega)$$

where

$$a(0) = h(M), \quad a(n) = 2h(M - n), \quad 1 \leq n \leq M.$$

DERIVATION

To obtain the coefficients $a(n)$ to minimize \mathcal{E}_2 , we can set the derivatives equal to zero,

$$\frac{d\mathcal{E}_2}{da(k)} = 0, \quad 0 \leq k \leq M.$$

The derivative of \mathcal{E}_2 with respect to $a(k)$ can be found as:

$$\frac{d\mathcal{E}_2}{da(k)} = \int_0^\pi \frac{d}{da(k)} [W(\omega) (A(\omega) - D(\omega))^2] d\omega \quad (2)$$

$$= 2 \int_0^\pi W(\omega) (A(\omega) - D(\omega)) \frac{dA(\omega)}{da(k)} d\omega \quad (3)$$

$$= 2 \int_0^\pi W(\omega) (A(\omega) - D(\omega)) \cos(k\omega) d\omega. \quad (4)$$

Therefore, $\frac{d\mathcal{E}_2}{da(k)} = 0$ becomes

$$\int_0^\pi W(\omega) A(\omega) \cos(k\omega) d\omega = \int_0^\pi W(\omega) D(\omega) \cos(k\omega) d\omega,$$

or

$$\int_0^\pi W(\omega) \left(\sum_{n=0}^M a(n) \cos(n\omega) \right) \cos(k\omega) d\omega = \int_0^\pi W(\omega) D(\omega) \cos(k\omega) d\omega$$

or

$$\sum_{n=0}^M a(n) \int_0^\pi W(\omega) \cos(n\omega) \cos(k\omega) d\omega = \int_0^\pi W(\omega) D(\omega) \cos(k\omega) d\omega$$

DERIVATION

If we define

$$Q(k, n) = \frac{1}{\pi} \int_0^\pi W(\omega) \cos(n\omega) \cos(k\omega) d\omega \quad (5)$$

and

$$b(k) = \frac{1}{\pi} \int_0^\pi W(\omega) D(\omega) \cos(k\omega) d\omega \quad (6)$$

then the derivative conditions can be written as

$$\sum_{n=0}^M Q(k, n) a(n) = b(k), \quad 0 \leq k \leq M. \quad (7)$$

This is a linear system of equations, and can be written in matrix form as

$$\begin{bmatrix} Q(0,0) & Q(0,1) & \cdots & Q(0,M) \\ Q(1,0) & Q(1,1) & \cdots & Q(1,M) \\ \vdots & & & \vdots \\ Q(M,0) & Q(M,1) & \cdots & Q(M,M) \end{bmatrix} \cdot \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(M) \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ \vdots \\ b(M) \end{bmatrix}$$

or

$$Q \cdot a = b.$$

Therefore, the Type I FIR filter that minimizes the square error can be obtained by solving this linear system of equations.

$$a = Q^{-1} b.$$

STRUCTURE OF THE MATRIX Q .

It turns out that the matrix Q has a special structure. Note that

$$\cos(n\omega) \cos(k\omega) = \frac{1}{2} \cos((k-n)\omega) + \frac{1}{2} \cos((k+n)\omega), \quad (8)$$

so that

$$Q(k, n) = \frac{1}{\pi} \int_0^\pi W(\omega) \cos(n\omega) \cos(k\omega) d\omega \quad (9)$$

$$= \frac{1}{2\pi} \int_0^\pi W(\omega) \cos((k-n)\omega) d\omega + \quad (10)$$

$$\frac{1}{2\pi} \int_0^\pi W(\omega) \cos((k+n)\omega) d\omega$$

or

$$Q(k, n) = \frac{1}{2} Q_1(k, n) + \frac{1}{2} Q_2(k, n) \quad (11)$$

where Q_1 and Q_2 are defined as

$$Q_1(k, n) = \frac{1}{\pi} \int_0^\pi W(\omega) \cos((k-n)\omega) d\omega \quad (12)$$

and

$$Q_2(k, n) = \frac{1}{\pi} \int_0^\pi W(\omega) \cos((k+n)\omega) d\omega. \quad (13)$$

Accordingly, we can write

$$Q_1(k, n) = q(k-n), \quad Q_2(k, n) = q(k+n) \quad (14)$$

where

$$q(n) = \frac{1}{\pi} \int_0^\pi W(\omega) \cos(n\omega) d\omega. \quad (15)$$

STRUCTURE OF THE MATRIX Q .

With this notation, the matrices Q_1 and Q_2 are written as

$$Q_1 = \begin{bmatrix} q(0) & q(1) & \cdots & q(M) \\ q(1) & q(0) & \cdots & q(M-1) \\ \vdots & & & \vdots \\ q(M) & q(M-1) & \cdots & q(0) \end{bmatrix} \quad (16)$$

and

$$Q_2 = \begin{bmatrix} q(0) & q(1) & \cdots & q(M) \\ q(1) & q(2) & \cdots & q(M+1) \\ \vdots & & & \vdots \\ q(M) & q(M+1) & \cdots & q(2M) \end{bmatrix}. \quad (17)$$

Note that we have used $q(-n) = q(n)$ here. The matrix Q_1 is a symmetric Toeplitz matrix (constant along its diagonals), the matrix Q_2 is a Hankel matrix (constant along its anti-diagonals). Consequently,

1. the matrices can be stored with less memory than arbitrary matrices ($2M + 1$ numbers instead of $(M + 1)^2$ numbers),
2. there are fast algorithms to compute the solution to 'Toeplitz plus Hankel' systems with computational complexity $O(M^2)$ instead of $O(M^3)$. (In fact, the complexity can be reduced further, but with higher overhead.)

RELATION TO THE DTFT

To express $q(k)$ and $b(k)$ using the inverse Fourier transform, extend $D(\omega)$ and $W(\omega)$ symmetrically, so that $D(-\omega) = D(\omega)$ and $W(-\omega) = W(\omega)$. Then we can write

$$q(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega) \cos(n\omega) d\omega. \quad (18)$$

As sin is an anti-symmetric function

$$\int_{-\pi}^{\pi} W(\omega) \sin(n\omega) d\omega = 0, \quad (19)$$

so we can write

$$q(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega) (\cos(n\omega) + j \sin(n\omega)) d\omega \quad (20)$$

or

$$q(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega) e^{jn\omega} d\omega \quad (21)$$

which we recognize as the inverse discrete-time Fourier transform

$$\boxed{q(n) = \text{DTFT}^{-1}\{W(\omega)\}.} \quad (22)$$

Similarly,

$$\boxed{b(n) = \text{DTFT}^{-1}\{W(\omega) D(\omega)\}.} \quad (23)$$

SPECIAL CASE: $W(\omega) = 1$

When the error function is equally weighted at all frequencies, meaning $W(\omega) = 1$, the square error is said to be *unweighted*. In this case the system of equations above becomes very simple. With $W(\omega) = 1$, the matrix Q becomes

$$Q(k, n) = \frac{1}{\pi} \int_0^\pi \cos(n\omega) \cos(k\omega) d\omega \quad (24)$$

$$= \begin{cases} 1 & k = n = 0 \\ \frac{1}{2} & k = n \neq 0 \\ 0 & k \neq n. \end{cases} \quad (25)$$

The inverse of Q is therefore simple,

$$Q = \begin{bmatrix} 1 & & & \\ & \frac{1}{2} & & \\ & & \ddots & \\ & & & \frac{1}{2} \end{bmatrix}, \quad Q^{-1} = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 2 \end{bmatrix}.$$

As

$$a = Q^{-1} b,$$

we have

$$a(0) = \frac{1}{\pi} \int_0^\pi D(\omega) d\omega,$$

and

$$a(n) = \frac{2}{\pi} \int_0^\pi D(\omega) \cos(n\omega) d\omega, \quad 1 \leq n \leq M.$$

SPECIAL CASE: $W(\omega) = 1$

which in turn gives

$$h(n) = \frac{1}{\pi} \int_0^\pi D(\omega) \cos((n - M)\omega) d\omega, \quad 0 \leq n \leq N.$$

To relate this to the inverse Fourier transform of $D(\omega)$, we can write

$$h(n + M) = \frac{1}{\pi} \int_0^\pi D(\omega) \cos(n\omega) d\omega. \quad (26)$$

As above, we recognize the inverse discrete-time Fourier transform,

$$h(n) = d(n - M), \quad 0 \leq n \leq N \quad (27)$$

where

$$d(n) = \text{DTFT}^{-1}\{D(\omega)\}. \quad (28)$$

The impulse response $h(n)$ is the truncated and shifted inverse DTFT of $D(\omega)$. Therefore, *the filter $h(n)$ obtained by truncating the inverse discrete-time Fourier transform of the desired amplitude response is optimal according to the **unweighted** square error criterion.* This method, sometimes called 'impulse response truncation' (IRT), simply consists of taking the inverse Fourier transform of the desired amplitude response, shifting it, and truncating it so that it is causal and of finite length.

IDEAL LOW-PASS

When $D(\omega)$ is the ideal low-pass filter with band-edge ω_o ,

$$D(\omega) = \begin{cases} 1 & 0 \leq \omega \leq \omega_o \\ 0 & \omega_o \leq \omega \leq \pi \end{cases} \quad (29)$$

then the formula above for $h(n)$ gives

$$h(n) = \frac{1}{\pi} \int_0^{\omega_o} D(\omega) \cos((n - M)\omega) d\omega \quad (30)$$

$$= \frac{1}{\pi} \int_0^{\omega_o} \cos((n - M)\omega) d\omega \quad (31)$$

$$= \frac{\sin((n - M)\omega_o)}{(n - M)\pi} \quad (32)$$

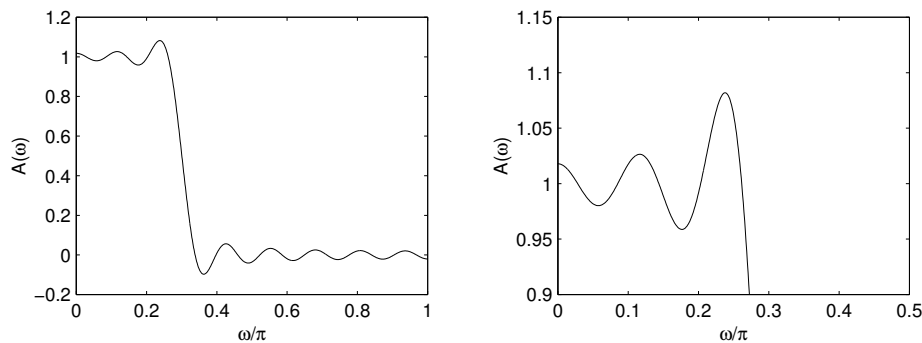
or

$$\boxed{h(n) = \frac{\omega_o}{\pi} \operatorname{sinc}\left(\frac{\omega_o}{\pi}(n - M)\right)} \quad (33)$$

where

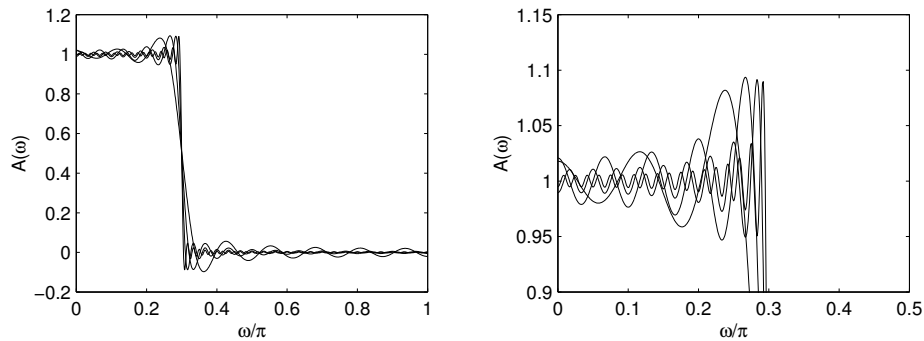
$$\operatorname{sinc}(\theta) = \frac{\sin(\pi\theta)}{\pi\theta}. \quad (34)$$

A length $N = 31$ Type I IRT low-pass:



IDEAL LOW-PASS

We illustrate the amplitude response $A(\omega)$ for Type I IRT low-pass filters with lengths 31, 61, 121, 241, and with $\omega_o = 0.3\pi$.



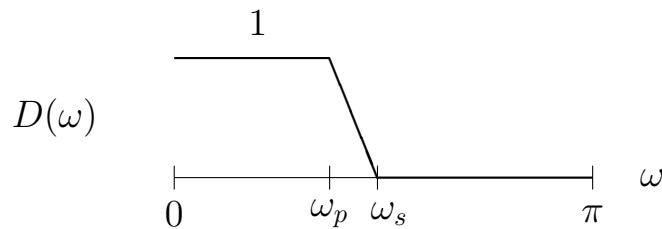
The second figure shows in detail the behavior at the pass-band edge. It is clear from this figure, that while the square error decreases for longer filters, the peak error does not diminish. This effect, known as *Gibbs phenomenon*, is a main weakness of the IRT method. There are several approaches to reduce the size of the peak error:

1. Multiply the impulse response $h(n)$ by a window that is tapered at its ends (*windowing*).
2. Modify $D(\omega)$ so that it has a smooth transition function.
3. Delete a neighborhood around the band edge from the square error criterion. This can be done with a weighting function $W(\omega)$ that is equal to zero in the chosen neighborhood. This is developed below.

LOW-PASS WITH TRANSITION FUNCTION

Sec 7.6.6
in Mitra

One way to reduce the peak error at the band edge is to introduce a transition band between the pass-band and stop-band. For example, the desired frequency response of the ideal low-pass filter can be modified so that it does not contain a discontinuity.



The pass-band is defined as $0 \leq \omega \leq \omega_p$, the transition region as $\omega_p \leq \omega \leq \omega_s$, and the stop-band as $\omega_s \leq \omega \leq \pi$. When the desired amplitude response $D(\omega)$ is modified in this way, then the FIR filter obtained by taking the inverse Fourier transform of $D(\omega)$ will have a smaller peak error. The transition function shown in the figure is a straight line (a first order spline), but it could be any smooth function.

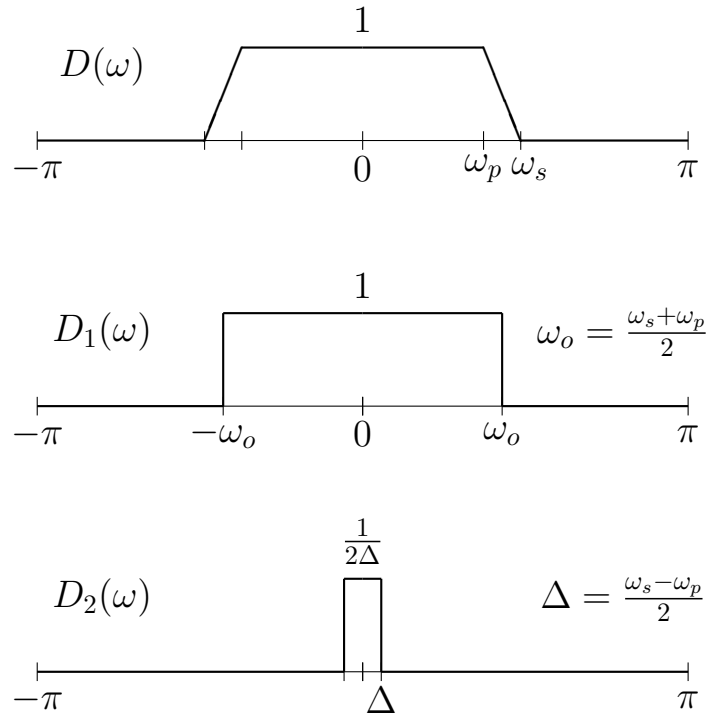
To find the unweighted square error solution we need to take inverse discrete-time Fourier transform of $D(\omega)$. An easy way to find the inverse DTFT of the function shown in the figure, is to note that it can be written as the convolution of two rectangular functions with appropriate widths.

LOW-PASS WITH TRANSITION FUNCTION

Namely,

$$D(\omega) = D_1(\omega) \circledast D_2(\omega) \quad (35)$$

where $D(\omega)$, $D_1(\omega)$, $D_2(\omega)$ as shown:



We can then use the convolution theorem to write the inverse DTFT as the product of $d_1(n)$ and $d_2(n)$. As both are sinc functions, we obtain the expression

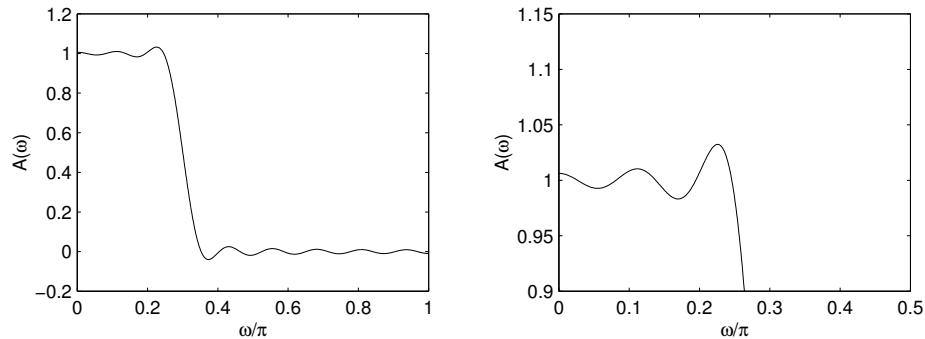
$$d(n) = 2\pi d_1(n) \cdot d_2(n) \quad (36)$$

$$= 2\pi \frac{\omega_o}{\pi} \operatorname{sinc}\left(\frac{\omega_o}{\pi} n\right) \cdot \frac{1}{2\Delta} \frac{\Delta}{\pi} \operatorname{sinc}\left(\frac{\Delta}{\pi} n\right) \quad (37)$$

$$= \frac{\omega_o}{\pi} \operatorname{sinc}\left(\frac{\omega_o}{\pi} n\right) \operatorname{sinc}\left(\frac{\Delta}{\pi} n\right). \quad (38)$$

TRANSITION FUNCTION EXAMPLE

The following figure shows the amplitude response of a length 31 Type I FIR filter designed using this method. The band edges used are $\omega_p = 0.26\pi$ and $\omega_s = 0.34\pi$. It is clear from the figure, that the peak error is significantly reduced.



Compare to the IRT filter on page 10.

The filter was designed using the following Matlab code.

```
% filter length
N = 31;
M = (N-1)/2;

% set parameters
wp = 0.26*pi;
ws = 0.34*pi;
wo = (ws+wp)/2;
Del = (ws-wp)/2;
fo = wo/pi;
Df = Del/pi;

% form impulse response h(n)
n = 0:N-1;
h = fo * sinc(fo*(M-n)) .* sinc(Df*(M-n));
```

LOW-PASS WITH TRANSITION FUNCTION

Instead of a straight line, a smoother transition function can be obtained by using, for example, a p^{th} order spline (a family of functions that are piece-wise polynomial). The inverse DTFT can then be obtained by writing $D(\omega)$ as a convolution of p rectangular functions, and $d(n)$ will therefore be of the form

$$d(n) = \frac{\omega_o}{\pi} \operatorname{sinc}\left(\frac{\omega_o}{\pi} n\right) \left[\operatorname{sinc}\left(\frac{\Delta}{\pi} n\right) \right]^p \quad (39)$$

where

$$\Delta = \frac{\omega_s - \omega_p}{p}. \quad (40)$$

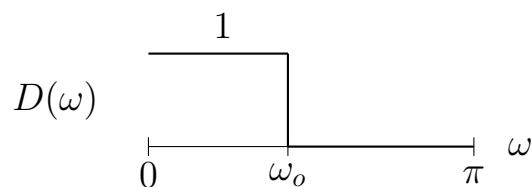
It is interesting to note that this works even when p is not an integer.

LOW-PASS: WEIGHTED SQUARE ERROR

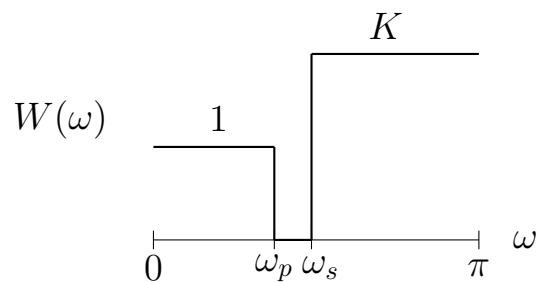
The weighting function $W(\omega)$ can be used to improve the FIR low-pass filter because

1. it allows you to eliminate Gibbs phenomenon by deleting a neighborhood around the band edge, and
2. it allows you to assign different weights to the pass-band and stop-band.

For example, if the ideal low-pass amplitude response is as shown



and if the weighting function is as shown



where $\omega_p < \omega_o < \omega_s$, then the square error criterion becomes

$$\mathcal{E}_2 = \int_0^{\omega_p} (A(\omega) - 1)^2 d\omega + K \int_{\omega_s}^{\pi} A^2(\omega) d\omega. \quad (41)$$

LOW-PASS: WEIGHTED SQUARE ERROR

To find the matrix Q and vector b corresponding to this weighting function it is useful to recall

$$\frac{1}{\pi} \int_{\omega_1}^{\omega_2} \cos(n\omega) d\omega = \frac{\omega_2}{\pi} \operatorname{sinc}\left(\frac{\omega_2}{\pi} n\right) - \frac{\omega_1}{\pi} \operatorname{sinc}\left(\frac{\omega_1}{\pi} n\right). \quad (42)$$

Then

$$q(k) = \frac{1}{\pi} \int_0^{\pi} W(\omega) \cos(k\omega) d\omega \quad (43)$$

$$= \frac{1}{\pi} \int_0^{\omega_p} \cos(k\omega) d\omega + \frac{K}{\pi} \int_{\omega_s}^{\pi} \cos(k\omega) d\omega \quad (44)$$

$$= \begin{cases} \frac{\omega_p}{\pi} + K \left(1 - \frac{\omega_s}{\pi}\right) & k = 0 \\ \frac{\omega_p}{\pi} \operatorname{sinc}\left(\frac{\omega_p}{\pi} k\right) - K \cdot \frac{\omega_s}{\pi} \operatorname{sinc}\left(\frac{\omega_s}{\pi} k\right) & k \neq 0. \end{cases} \quad (45)$$

Similarly,

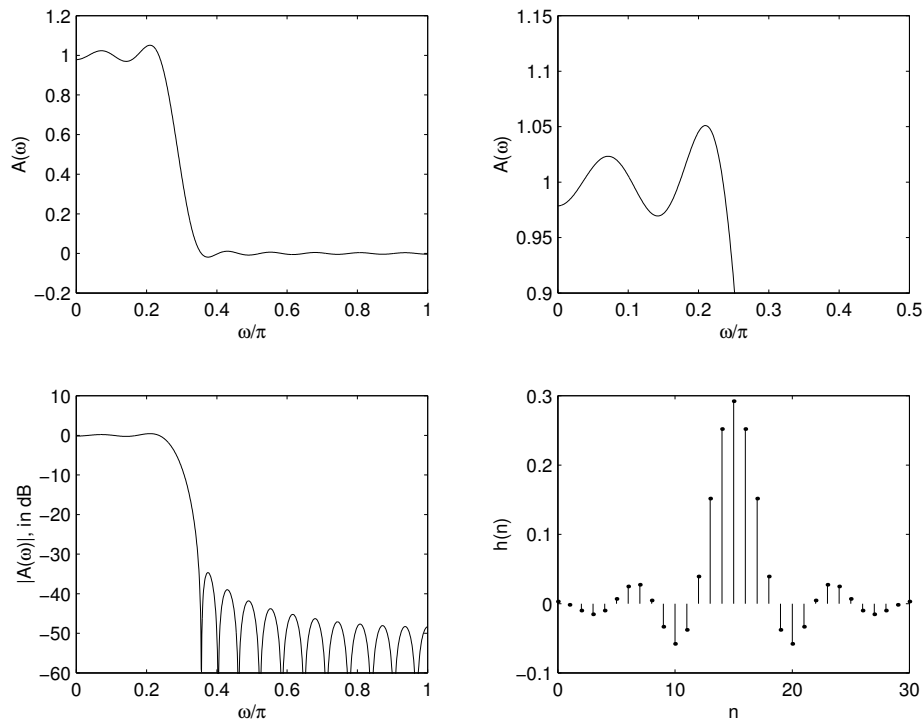
$$b(k) = \frac{1}{\pi} \int_0^{\pi} W(\omega) D(\omega) \cos(k\omega) d\omega \quad (46)$$

$$= \frac{1}{\pi} \int_0^{\omega_p} \cos(k\omega) d\omega \quad (47)$$

$$= \frac{\omega_p}{\pi} \operatorname{sinc}\left(\frac{\omega_p}{\pi} k\right). \quad (48)$$

WEIGHTED LOW-PASS EXAMPLE

In the following example, we design a Type I FIR low-pass filter of length 31, with band-edges $\omega_p = 0.26\pi$, $\omega_s = 0.34\pi$ and a stop-band weight of $K = 10$.



The second figure shows the pass-band edge detail.

Comparing to the figure on page 10 (IRT method), we see that the peak error is reduced.

Comparing to the figure on page 14 (transition function method), we see that the stopband error is smaller than the passband error.

WEIGHTED LOW-PASS EXAMPLE

The filter was designed using the following Matlab code.

```
% WEIGHTED LEAST SQUARE LOWPASS FILTER

% filter length
N = 31;
M = (N-1)/2;

% set band-edges and stop-band weighting
wp = 0.26*pi;
ws = 0.34*pi;
K = 10;

% normalize band-edges for convenience
fp = wp/pi;
fs = ws/pi;

% construct q(k)
q = [fp+K*(1-fs), fp*sinc(fp*[1:2*M])-K*fs*sinc(fs*[1:2*M])];

% construct Q1, Q2, Q
Q1 = toeplitz(q([0:M]+1));
Q2 = hankel(q([0:M]+1),q([M:2*M]+1));
Q = (Q1 + Q2)/2;

% construct b
b = fp*sinc(fp*[0:M]');

% solve linear system to get a(n)
a = Q\b;

% form impulse response h(n)
h = [a(M+1:-1:2); 2*a(1); a(2:M+1)]/2;
```

SQUARE ERROR WITH SIDE CONSTRAINTS

Sec 7.9 in
Mittra

In many circumstances, it is desired that the filter satisfy some specific constraints. For example, it might be required that the amplitude response have a null at a specified frequency ω_1 ,

$$A(\omega_1) = 0. \quad (49)$$

This can be written as a single linear equation

$$G a = d \quad (50)$$

where G is a row vector

$$G = [1 \quad \cos(\omega_1) \quad \cos(2\omega_1) \quad \cdots \quad \cos(M\omega_1)], \quad (51)$$

a is a column vector containing the filter coefficients

$$a = [a(0) \quad a(1) \quad a(2) \quad \cdots \quad a(M)]^t, \quad (52)$$

and

$$d = 0. \quad (53)$$

In general, one can have more than one constraint, in which case G is a matrix rather than a row vector, and d is a column vector rather than a scalar. Such a filter can be obtained by using a constrained square error approach. This method minimizes the square error \mathcal{E}_2 subject to the side constraints $G a = d$.

SQUARE ERROR WITH SIDE CONSTRAINTS

To minimize \mathcal{E}_2 subject to the side constraints, we can use the method of *Lagrange multipliers*. When a single constraint is imposed we form the Lagrangian as

$$\mathcal{L} = \mathcal{E}_2 + (G a - d) \cdot \mu. \quad (54)$$

When K constraints are imposed the Lagrangian becomes

$$\mathcal{L} = \mathcal{E}_2 + \sum_{k=1}^K (G a - d)_k \cdot \mu(k) \quad (55)$$

which can be written in matrix form as

$$\mathcal{L} = \mathcal{E}_2 + (G a - d)^t \cdot \mu \quad (56)$$

where μ is a column vector containing the K Lagrange multipliers,

$$\mu = [\mu(1) \quad \mu(2) \quad \cdots \quad \mu(K)]^t. \quad (57)$$

The necessary conditions for the minimization of \mathcal{E}_2 subject to the constraint $G a = d$ is obtained by setting the derivatives of the Lagrangian \mathcal{L} with respect to $a(n)$ and $\mu(k)$ equal to zero.

$$\frac{d\mathcal{L}}{da(n)} = 0, \quad 0 \leq n \leq M, \quad \frac{d\mathcal{L}}{d\mu(k)} = 0, \quad 1 \leq k \leq K. \quad (58)$$

When there is a single constraint, we find that

$$\frac{d\mathcal{L}}{da(n)} = \frac{d\mathcal{E}_2}{da(n)} + G_n \mu. \quad (59)$$

SQUARE ERROR WITH SIDE CONSTRAINTS

When there are K constraints,

$$\frac{d\mathcal{L}}{da(n)} = \frac{d\mathcal{E}_2}{da(n)} + \sum_{k=1}^K G_{k,n} \mu_k. \quad (60)$$

We also have

$$\frac{d\mathcal{L}}{d\mu(k)} = (G a - d)_k. \quad (61)$$

The derivative condition $\frac{d\mathcal{L}}{da(n)} = 0$ for $0 \leq n \leq M$ can be written in matrix form as

$$[Q \quad G^t] \cdot \begin{bmatrix} a \\ \mu \end{bmatrix} = b \quad (62)$$

where Q and b are the same matrix and vector as those arising in the unconstrained square error problem treated above. Combining this with the condition $\frac{d\mathcal{L}}{d\mu} = 0$ into a single matrix equation gives

$$\begin{bmatrix} Q & G^t \\ G & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ \mu \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix}. \quad (63)$$

Therefore, the filter that minimizes \mathcal{E}_2 subject to the side constraint $G a = d$ can be obtained by solving again a linear system of equations. In this case, we need to solve the equations for both the Lagrange multiplier μ and the filter coefficients a .

SQUARE ERROR WITH SIDE CONSTRAINTS

It can be shown that the solution to this system of equations can be written as

$$\mu = (G Q^{-1} G^t)^{-1} (G Q^{-1} b - d) \quad (64)$$

and

$$a = Q^{-1} (b - G^t \mu). \quad (65)$$

Notice that the expression for a depends of μ . Hence, we can find μ first, and then find a ; in each case by solving a linear system of equations. Also note that the expressions can be written as

$$\mu = (G Q^{-1} G^t)^{-1} (G a_u - d) \quad (66)$$

and

$$a = a_u - Q^{-1} G^t \mu \quad (67)$$

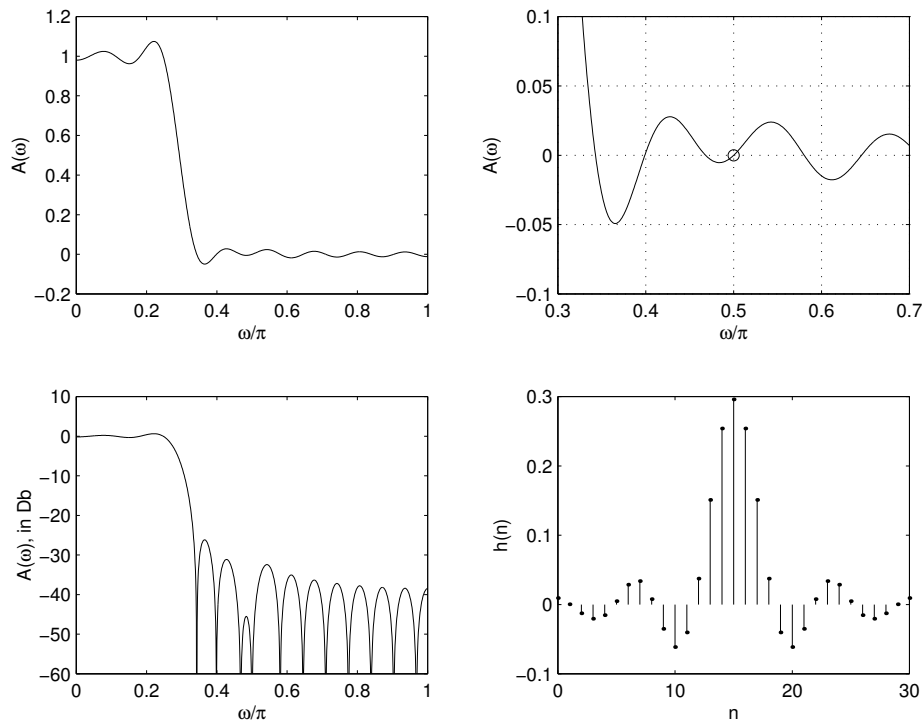
where

$$a_u = Q^{-1} b \quad (68)$$

is the solution that minimizes the *unconstrained* square error. The term $Q^{-1} G^t \mu$ can be viewed as the additive correction term that makes the constraints satisfied and maintains optimality.

LOW-PASS WITH SPECIFIED NULL

The constrained square error criterion can be used to design a filter with a null (zero) at a specified frequency. For example, the following figure shows a length 31 Type I FIR low-pass filter designed to minimize the weighted square error subject to the constraint that the amplitude response $A(\omega)$ has a null at $\omega_1 = 0.5\pi$. The pass-band and stop-band edges used in this example are $\omega_p = 0.28\pi$ and $\omega_s = 0.32\pi$. The error in the stop-band is weighted by a factor of $K = 4$.



It can be seen in the figure that the amplitude response has a null at the specified frequency. The filter was designed using the following Matlab code. In the Matlab code, some expressions of the form $R^{-1}b$ are written as $R \setminus b$ instead of $\text{inv}(R) * b$ because the first form is more efficient. The form $R \setminus b$ solves the linear system $Rx = b$ without computing the inverse of R explicitly.

LOW-PASS WITH SPECIFIED NULL

```
% LEAST SQUARE LOWPASS FILTER with SPECIFIED NULL

% filter length
N = 31;
M = (N-1)/2;

% set band-edges and stop-band weighting
wp = 0.28*pi;
ws = 0.32*pi;
K = 4;

% set null
w1 = 0.5*pi;

% normalize band-edges for convenience
fp = wp/pi;
fs = ws/pi;

% construct q(k)
q = [fp+K*(1-fs), fp*sinc(fp*[1:2*M])-K*fs*sinc(fs*[1:2*M])];

% construct Q1, Q2, Q
Q1 = toeplitz(q([0:M]+1));
Q2 = hankel(q([0:M]+1),q([M:2*M]+1));
Q = (Q1 + Q2)/2;

% construct b
b = fp*sinc(fp*[0:M]');

% construct G and d
G = cos([0:M]*w1);
d = [0];

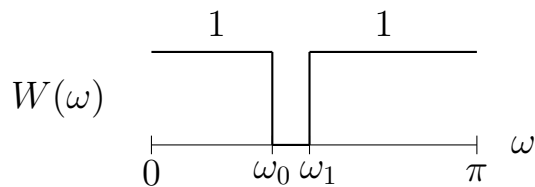
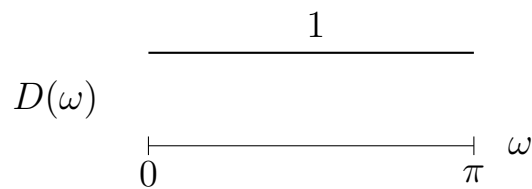
% solve linear system for Lagrange multipliers
c = Q\b;
mu = (G*inv(Q)*G')\ (G*c-d);
% solve linear system for filter coeffs
a = c-Q\ (G'*mu);

% form impulse response h(n)
h = [a(M+1:-1:2); 2*a(1); a(2:M+1)]/2;
```

NOTCH FILTER

As an example of a filter design problem based on constrained square error minimization, consider the design of a notch filter. A notch filter is used when one wishes to remove interference at a specific frequency and not affect the signal at other frequencies. That is, one would ideally like $A(\omega)$ to equal 1 for all frequencies ω except for a specific notch frequency ω_n at which $A(\omega_n) = 0$.

One way to pose the design problem is to let the desired amplitude response be a constant, but to delete an interval around the notch frequency ω_n from the square error by setting the weighting function $W(\omega)$ to 0 there. In this case, the desired amplitude $D(\omega)$ and weight function $W(\omega)$ will be as shown.



The notch of the filter can then be imposed as a side constraint,

$$A(\omega_n) = 0. \quad (69)$$

NOTCH FILTER

The side constraint $A(\omega_n) = 0$ will result in a filter that has a zero at $z = e^{j\omega_n}$ and $z = e^{-j\omega_n}$. (Because the zeros must occur in complex conjugate pairs when the impulse response is real, which we assume here.)

We can ask in addition that

$$\frac{dA(\omega_n)}{d\omega} = 0, \quad (70)$$

which will result in a filter that has a double zero at $z = e^{j\omega_n}$ and $z = e^{-j\omega_n}$. In this case, we have two constraints, so the constraint vector G becomes a matrix with 2 rows and $M + 1$ columns. Note that

$$\frac{dA(\omega_n)}{d\omega} = -a(1) \sin(\omega_n) - a(2) 2 \sin(2\omega_n) - \dots - a(M) M \sin(M\omega_n) \quad (71)$$

so that

$$G = \begin{bmatrix} 1 & \cos(\omega_n) & \cos(2\omega_n) & \cdots & \cos(M\omega_n) \\ 0 & -\sin(\omega_n) & -2 \sin(2\omega_n) & \cdots & -M \sin(M\omega_n) \end{bmatrix} \quad (72)$$

and

$$d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (73)$$

NOTCH FILTER

To compute Q and b for this problem, notice that

$$q(k) = \frac{1}{\pi} \int_0^{\pi} W(\omega) \cos(k\omega) d\omega \quad (74)$$

$$= \frac{1}{\pi} \int_0^{\omega_0} \cos(k\omega) d\omega + \frac{1}{\pi} \int_{\omega_1}^{\pi} \cos(k\omega) d\omega \quad (75)$$

$$= \begin{cases} \frac{\omega_0}{\pi} + 1 - \frac{\omega_1}{\pi} & k = 0 \\ \frac{\omega_0}{\pi} \operatorname{sinc}\left(\frac{\omega_0}{\pi}k\right) - \frac{\omega_1}{\pi} \operatorname{sinc}\left(\frac{\omega_1}{\pi}k\right) & k \neq 0. \end{cases} \quad (76)$$

Similarly,

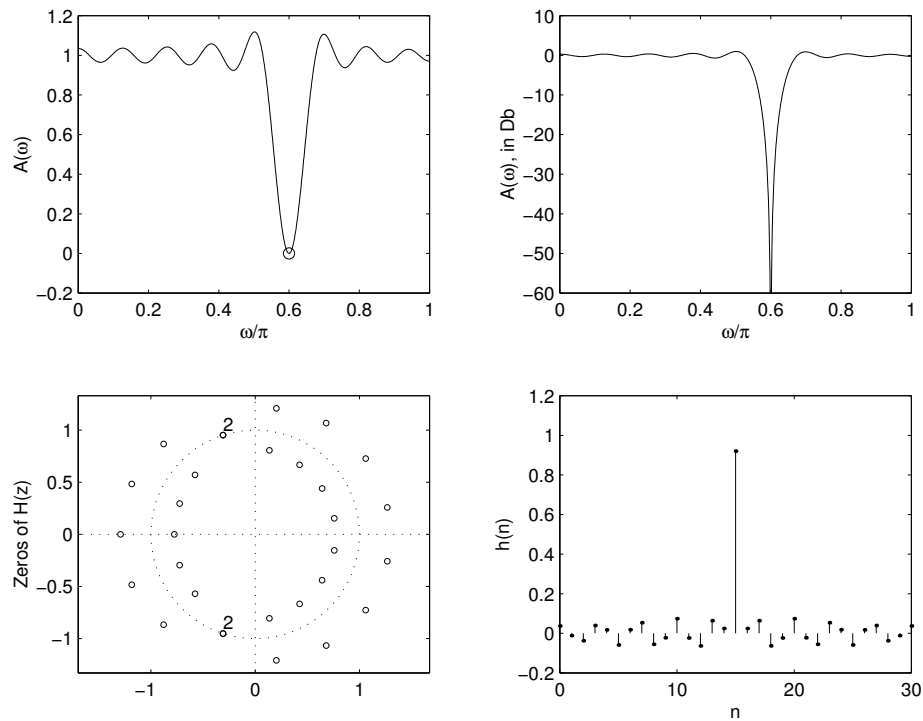
$$b(k) = \frac{1}{\pi} \int_0^{\pi} W(\omega) D(\omega) \cos(k\omega) d\omega \quad (77)$$

$$= \frac{1}{\pi} \int_0^{\omega_0} \cos(k\omega) d\omega + \frac{1}{\pi} \int_{\omega_1}^{\pi} \cos(k\omega) d\omega \quad (78)$$

$$= q(k). \quad (79)$$

NOTCH FILTER EXAMPLE

For example, a length 31 Type I FIR notch filter designed according to this criterion with notch frequency $\omega_n = 0.6\pi$, and $\omega_0 = 0.55\pi$, $\omega_1 = 0.65\pi$, is shown in the following figure.



NOTCH FILTER EXAMPLE

```
% LEAST SQUARE NOTCH FILTER

% filter length
N = 31;
M = (N-1)/2;

% set band-edges and stop-band weighting
w0 = 0.55*pi;
w1 = 0.65*pi;

% set null
wn = 0.6*pi;

% normalize band-edges for convenience
f0 = w0/pi;
f1 = w1/pi;

% construct q(k)
q = [f0+(1-f1), f0*sinc(f0*[1:2*M])-f1*sinc(f1*[1:2*M])];

% construct Q1, Q2, Q
Q1 = toeplitz(q([0:M]+1));
Q2 = hankel(q([0:M]+1),q([M:2*M]+1));
Q = (Q1 + Q2)/2;

% construct b
b = q([0:M]+1)';

% construct G and d
G = [cos([0:M]*wn); -[0:M].*sin([0:M]*wn)];
d = [0; 0];

% solve linear system for Lagrange multipliers
c = Q\b;
mu = (G*inv(Q)*G')\ (G*c-d);
% solve linear system for filter coeffs
a = c-Q\ (G'*mu);

% form impulse response h(n)
h = [a(M+1:-1:2); 2*a(1); a(2:M+1)]/2;
```

DISCRETE SQUARE ERROR

Sec 7.8 in
Mittra

When the desired amplitude response $D(\omega)$ and the weighting function $W(\omega)$ are simple, then $q(n)$ and $b(n)$ can be found analytically. However, the square error approach can be used even for more general functions as well when the integral square error is replaced by the discrete square error.

A discretized version of the square error criterion is useful in several cases.

1. If $D(\omega)$ and $W(\omega)$ are not simple enough to permit the integrals arising in $q(n)$ and $b(n)$ to be found analytically.
2. Closed form formulas for $D(\omega)$ and $W(\omega)$ are not always known. For example, if they are obtained through the result of measurements, they may be known only on a finite set of frequencies.

We will assume that $D(\omega)$ and $W(\omega)$ are known on a uniform grid of L frequencies from 0 to 2π :

$$\omega_k = \frac{2\pi}{L}k, \quad 0 \leq k \leq L - 1, \quad (80)$$

and we'll define

$$W_k = W\left(\frac{2\pi}{L}k\right), \quad D_k = D\left(\frac{2\pi}{L}k\right), \quad 0 \leq k \leq L - 1. \quad (81)$$

DISCRETE SQUARE ERROR

Then $q(n)$ can be approximated as

$$q(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega) e^{jn\omega} d\omega \quad (82)$$

$$\approx \frac{1}{L} \sum_{k=0}^{L-1} W(\omega_k) e^{jn\omega_k} \quad (83)$$

$$\approx \frac{1}{L} \sum_{k=0}^{L-1} W_k e^{jn\frac{2\pi}{L}k}. \quad (84)$$

Recognizing the inverse DFT, we have

$$\boxed{q(n) \approx \text{DFT}_L^{-1} \{W_k\}} \quad (85)$$

and similarly

$$\boxed{b(n) \approx \text{DFT}_L^{-1} \{W_k D_k\}}. \quad (86)$$

For the discretization to be accurate (for the integral to be well approximated by the sum) L should be significantly larger than the length of the filter N , say $L > 5N$.

Note that although the DFT gives $q(n)$ for $0 \leq n \leq L-1$, only the first $2M+1$ values will be used — the other values do not appear in the matrix Q .

DISCRETE SQUARE ERROR: EXAMPLE

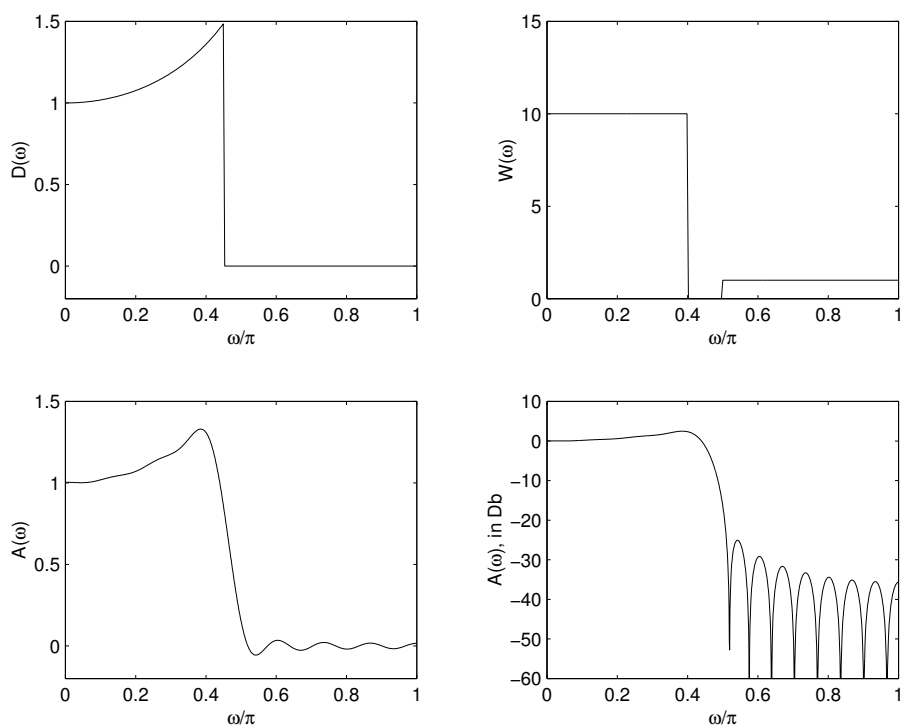
To illustrate this procedure, suppose the desired amplitude has the form

$$D(\omega) = \begin{cases} \frac{1}{\text{sinc}(\omega/3)} & |\omega| \leq 0.45\pi \\ 0 & 0.45\pi < |\omega| \leq \pi \end{cases} \quad (87)$$

and the weighting function is

$$W(\omega) = \begin{cases} 10 & |\omega| \leq 0.4\pi \\ 0 & 0.4\pi < |\omega| \leq 0.52\pi \\ 1 & 0.52\pi < |\omega| \leq \pi. \end{cases} \quad (88)$$

The amplitude response $A(\omega)$ for length-31 Type I filter designed to minimize the discrete weighted square error is shown in the following figures. The grid size $L = 2^9$ was used.



DISCRETE SQUARE ERROR: EXAMPLE

The filter was designed using the following Matlab code. The commands to build the vectors D_k and W_k require some comment — The DFT requires samples of the functions over the interval from 0 to 2π . Therefore, it is necessary to extend the vectors appropriately. Here this is done with the command $D = [DA, DA(L/2:-1:2)]$.

```
% DISCRETE SQUARE ERROR

% filter length
N = 31;
M = (N-1)/2;
L = 2^8;

w = [0:L]*pi/L;          % 0 <= omega <= pi
DA = (1./sinc(w/3)).*(w<=0.45*pi);
D = [DA, DA(L:-1:2)];   % periodize

w0 = 0.4*pi;
w1 = 0.5*pi;
WA = 10*(w<=w0)+(w>=w1);
W = [WA, WA(L:-1:2)];   % periodize

% construct q(k)
q = ifft(W);

% construct Q1, Q2, Q
Q1 = toeplitz(q([0:M]+1));
Q2 = hankel(q([0:M]+1),q([M:2*M]+1));
Q = (Q1 + Q2)/2;

% construct b
b = ifft(W.*D);
b = b([0:M]+1)';        % just need n=0:M coeffs

% solve linear system to get a(n)
a = Q\b;

% form impulse response h(n)
h = [a(M+1:-1:2); 2*a(1); a(2:M+1)]/2;
```

GENERAL DISCRETE SQUARE ERROR

When no analytic expressions are available for the inverse DTFT of $D(\omega)$ and $W(\omega)$, then one may consider the discrete square error measure, written as

$$\epsilon_2 = \sum_{k=1}^L W(\omega_k) (A(\omega_k) - D(\omega_k))^2 \quad (89)$$

where ω_k are a set of frequency points which are not necessarily uniformly spaced. To develop the second discrete square error approach, define

$$A_k = A(\omega_k), \quad D_k = D(\omega_k), \quad W_k = W(\omega_k), \quad 1 \leq k \leq L. \quad (90)$$

Then the error function,

$$e_k = A_k - D_k \quad (91)$$

or

$$e_k = \sum_{n=0}^M a(n) \cos(n\omega_k) - D_k, \quad (92)$$

can be written in matrix form as

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_L \end{bmatrix} = \begin{bmatrix} 1 & \cos(w_1) & \cdots & \cos(Mw_1) \\ 1 & \cos(w_2) & \cdots & \cos(Mw_2) \\ \vdots & & & \vdots \\ 1 & \cos(w_L) & \cdots & \cos(Mw_L) \end{bmatrix} \cdot \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(M) \end{bmatrix} - \begin{bmatrix} D_0 \\ D_1 \\ \vdots \\ D_L \end{bmatrix} \quad (93)$$

GENERAL DISCRETE SQUARE ERROR

or more briefly as

$$e = C \cdot a - D. \quad (94)$$

The matrix C has more rows than columns and we can see that $C a = D$ is an over-determined system of equations. The 'least square solution' to an over-determined system minimizes the (unweighted) sum of squares of the error vector, which can be compactly written as

$$\epsilon_2 = e^t e. \quad (95)$$

The least square solution is obtained by first multiplying both sides of

$$C a \approx D \quad (96)$$

on the left by C^t to obtain the *normal equations*

$$C^t C a = C^t D. \quad (97)$$

The least square solution (unweighted) is then given by

$$a = (C^t C)^{-1} C^t D. \quad (98)$$

The weighted square error solution is obtained by first multiplying on the left by $W = \text{diag}(\sqrt{W_k})$

$$W C a \approx W D \quad (99)$$

GENERAL DISCRETE SQUARE ERROR

and then by C^t

$$C^t W C a = C^t W D. \quad (100)$$

Solving gives

$$a = (C^t W C)^{-1} C^t W D. \quad (101)$$

In practice, the inverse of $C^t W C$ need not be computed explicitly. Instead, there are readily available efficient algorithms for solving over-determined linear equations in the least square sense. In Matlab, when used for an over-determined system, the backslash `\` gives the least square solution automatically.

This matrix approach to least square error FIR filter design is straightforward and simple to use.

GENERAL DISCRETE SQUARE ERROR

To illustrate the design of a low-pass filter by solving an over-determined system of equations in the least square sense, we consider the design of a low-pass filter with exactly the same parameters as the design example on page 18. Namely, $\omega_p = 0.26\pi$, $\omega_s = 0.34\pi$, and a stop-band weight of $K = 10$. The filter is obtained with the following Matlab code. The figures are not given, as they are the same as the figures on page 18.

```
% DISCRETE SQUARE ERROR - MATRIX METHOD

% filter length
N = 31;
M = (N-1)/2;

% parameters
wo = 0.30*pi;           % cut-off freq
wp = 0.26*pi;           % pass-band edge
ws = 0.34*pi;           % stop-band edge
K = 10;                 % stop-band weight

L = 2^9;                % grid size
w = [0:L]'*pi/L;        % frequency grid
D = w < wo;             % desired response
W = diag(sqrt((w <= wp)+K*(w >= ws))); % weight function
C = cos(w*[0:M]);        % rectangular C matrix

% solve linear system to get a(n)
a = (W*C)\(W*D);

% form impulse response h(n)
h = [a(M+1:-1:2); 2*a(1); a(2:M+1)]/2;
```

LEAST SQUARE FIR FILTER DESIGN: PROS AND CONS

1. Optimal with respect to square error criterion.
2. Simple, non-iterative method.
3. Analytic solutions sometimes possible, otherwise solution is obtained via solution to linear system of equations.
4. Allows the use of a frequency dependent weighting function.
5. Suitable for arbitrary $D(\omega)$ and $W(\omega)$.
6. Easy to include arbitrary linear constraints.
7. Does not allow direct control of maximum ripple size.