MINIMUM-PHASE FIR FILTER DESIGN
The Lifting Procedure

1. INTRODUCTION
2. PROBLEM FORMULATION
3. THE SQUARE MAGNITUDE
4. A TRANSFORMATION
5. MEETING SPECIFICATIONS
6. SUMMARY OF THE LIFTING PROCEDURE
7. MEETING SPECIFICATIONS — EXAMPLE
8. PLOTTING THE PHASE
9. PHASE UNWRAPPING
10. GROUP DELAY

INTRODUCTION

FIR filters that have symmetric or antisymmetric impulse responses have no phase distortion (their phase is linear). However, the problem with linear-phase filters is that the delay can be too large. The delay of a linear-phase filter is equal to $(N-1)/2$, where N is the length of the filter. To obtain a linear-phase FIR filter with a narrow transition-band and high stop-band attenuation requires making the filter long. Therefore, linear-phase FIR filters satisfying demanding specifications will have a large delay. This large delay could be a major drawback. (1) If the filter is used inside a feedback loop in a control system it could cause instability. (2) If the filter is used in a communication system it could cause delays that are longer than is acceptable.

For applications where it is important to minimize the delay caused by a filter, a minimum-phase filter can be a good choice. Minimum-phase filters have all their zeros inside or on the unit circle. A minimum-phase filter can be obtained from a linear-phase filter by reflecting all of the zeros that are outside the unit circle to inside the unit circle. In other words, those zeros located at $z = r\, e^{j\theta}$ are moved to $z = \frac{1}{r}\, e^{j\theta}$. This modification results in a minimum-phase filter that has the same frequency response magnitude $|H(e^{j\omega})|$ as the linear-phase filter. However, the minimum-phase filter obtained in this way will not be optimal in general. There will generally be a minimum-phase filter that is superior than the one obtained by simply reflecting the zeros in this way.

PROBLEM FORMULATION

In this section we consider the design of low-pass filters. The design problem can be stated as follows.

Problem 1: Find the FIR filter $h(n)$ of minimal length (not necessarily with linear-phase) such that

$$||H(e^{j\omega})| - 1| \le \Delta_p \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{1}$$

$$|H(e^{j\omega})| \le \Delta_s \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{2}$$

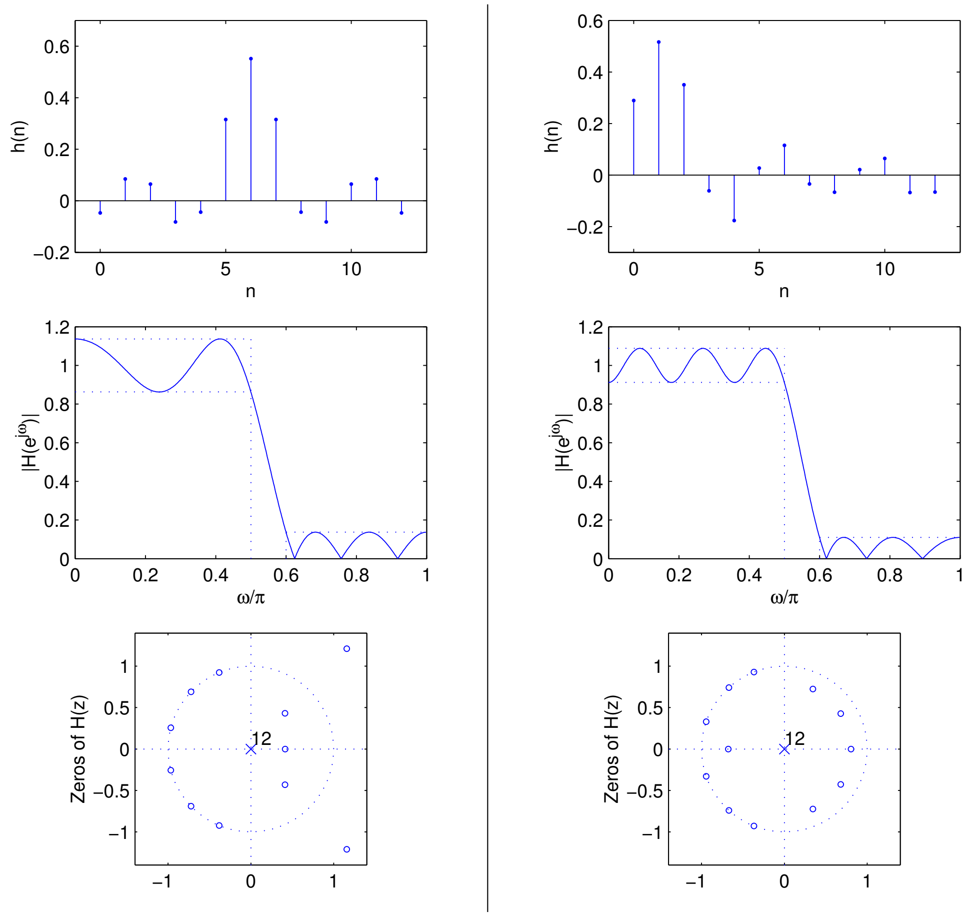
The given design parameters are the band-edges ω_p, ω_s and the maximum error in the pass-band and stop-band Δ_p, Δ_s.

This design problem is more difficult than the design of a linear-phase FIR filter, because the frequency response magnitude $|H(e^{j\omega})|$ is a nonlinear function of the filter coefficients $h(n)$.

$$|H(e^{j\omega})| = \left| \sum_{n=0}^{N-1} h(n)\, e^{j\omega n} \right|. \tag{3}$$

This section describes how to design minimum-phase FIR filters so that the error between $|H(e^{j\omega})|$ and a desired $D(\omega)$ is minimized.

EXAMPLE



The filter on the left is a linear-phase FIR filter of length 13 designed so that the Chebyshev error is minimized. The maximum value of the error in the pass-band and stop-band is 0.137. The filter on the right is a minimum-phase FIR filter of the same length, however, the maximum value of the error in the pass-band is 0.0882, and in the stop-band it is 0.1099. By not restricting the impulse response to be symmetric, the frequency response magnitude $|H(e^{j\omega})|$ can be improved. The remainder of this section describes how to design a non-symmetric low-pass filter $h(n)$ so that $|H(e^{j\omega})|$ has the smallest error.

THE SQUARE MAGNITUDE

Because $|H(e^{j\omega})|$ is not a linear function of the coefficients $h(n)$, it turns out to be more convenient to work with the square magnitude $|H(e^{j\omega})|^2$. Assuming the filter coefficients $h(n)$ are real, we can write:

$$|H(e^{j\omega})|^2 = H(e^{j\omega}) \cdot \overline{H(e^{j\omega})} \tag{4}$$

$$= H(e^{j\omega}) \cdot H(e^{-j\omega}) \tag{5}$$

$$= H(z) \cdot H(1/z)\big|_{z=e^{j\omega}} . \tag{6}$$

Suppose $h(n)$ is a length N FIR impulse response. If we define

$$R(z) := H(z) \cdot H(1/z) \tag{7}$$

then

$$r(n) = h(n) * h(-n) \tag{8}$$

and

$$R(e^{j\omega}) = |H(e^{j\omega})|^2. \tag{9}$$

Note that $r(n)$ will be symmetric with $r(n) = r(-n)$ and the length of $r(n)$ will be $2N - 1$. It turns out to be more straight-forward to design the filter $R(z)$, and then to obtain $H(z)$ from $R(z)$.

Herrmann and Schüssler proposed a method for designing minimum-phase filters for which the Chebyshev error is minimized [1]. Their method is based on the transformation of a linear-phase FIR Chebyshev filter in to a minimum-phase FIR filter.

Transformation of Type I FIR equi-ripple linear-phase filter into an equi-ripple minimum-phase filter

The method consists of 4 steps.

1. Design an equi-ripple Type I linear-phase filter $h_1(n)$ of length $2N-1$.

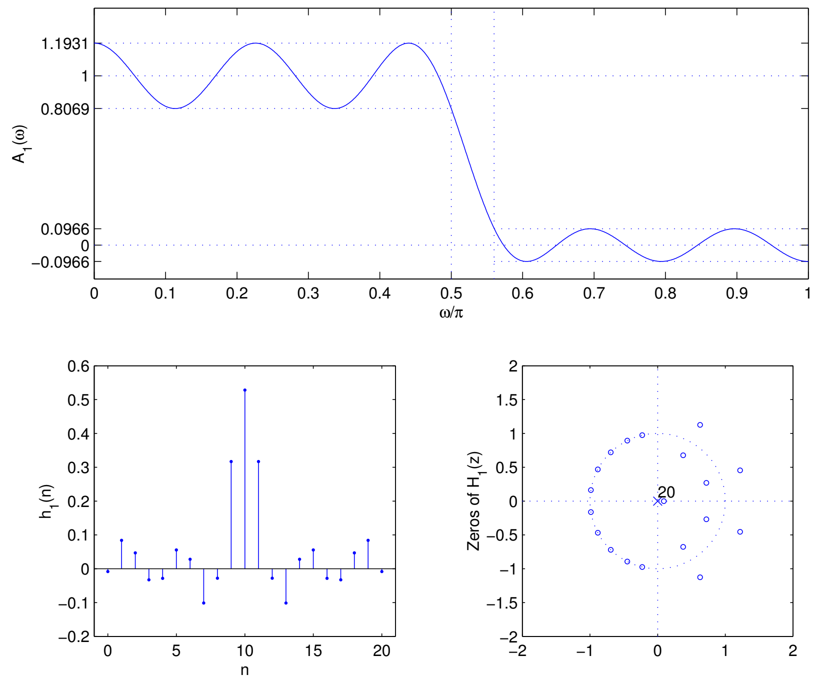
2. Obtain a second Type I linear-phase filter $h_2(n)$ by adjusting $h_1(n)$ so that $A_2(\omega) \geq 0$.

3. Spectrally factor $h_2(n)$ to obtain a minimum-phase FIR filter $h_3(n)$ of length N.

4. Scale $h_3(n)$ by an appropriate constant to obtain a minimum-phase FIR filter $h(n)$ that is equi-ripple.

This procedure is best illustrated by an example.

Step One: Design an equi-ripple Type I FIR filter of length $2N-1$. For example, to design a length 11 minimum-phase FIR filter, we will first design a length 21 Type I FIR filter.



Define δ_1 and δ_2 to be the maximum value of the error in the pass-band and stop-band.

$$\delta_1 := \max |A_1(\omega) - 1| \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{10}$$

$$\delta_2 := \max |A_1(\omega)| \qquad \text{for} \quad \omega_s \le \omega \le \pi \tag{11}$$

We can then write:

$$1 - \delta_1 \le A_1(\omega) \le 1 + \delta_1 \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{12}$$

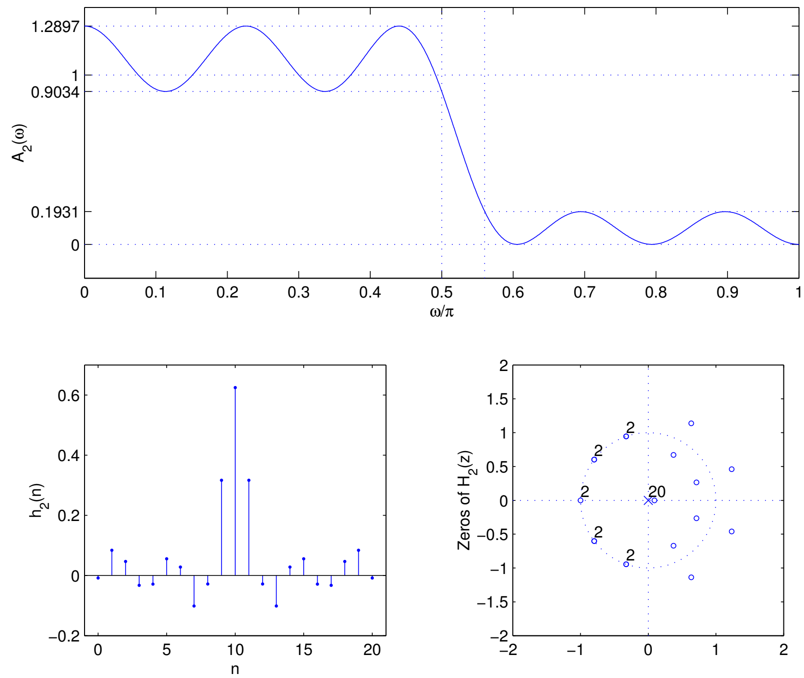
$$-\delta_2 \le A_1(\omega) \le \delta_2 \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{13}$$

```
N  = 11;
Kp = 1;
Ks = 2;
wp = 0.50*pi;
ws = 0.56*pi;
wo = (wp+ws)/2;
L  = 1000;
w  = [0:L]*pi/L;
D  = (w<=wo);
W  = Kp*(w<=wp) + Ks*(w>=ws);
[h1,del] = fircheb(2*N-1,D,W);
d1 = del/Kp;
d2 = del/Ks;
```

In this example, $\delta_1 = 0.1931$, $\delta_2 = 0.0966$.

STEP TWO

Step Two: Step two is to adjust the filter $h_1(n)$ so that the filter can be spectrally factored. This can be done by adding δ_2 to the middle coefficient of $h_1(n)$.



$$\boxed{h_2(n) = h_1(n) + \delta_2 \cdot \delta(n - N + 1)} \tag{14}$$

In Matlab this can be done with the command

```
h2 = h1;
h2(N) = h2(N) + d2;
```

This has the effect of 'lifting' the amplitude response

$$\boxed{A_2(\omega) = A_1(\omega) + \delta_2.} \tag{15}$$

The frequency response amplitude of $h_2(n)$ then satisfies the following bounds,

$$1-\delta_1+\delta_2 \le A_2(\omega) \le 1+\delta_1+\delta_2 \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{16}$$

$$0 \le A_2(\omega) \le 2\,\delta_2 \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{17}$$

Notice that the zeros of $H_2(z)$ on the unit circle are now of even multiplicity. Then the filter $h_2(n)$ can be spectrally factored because

$$\boxed{A_2(\omega) \ge 0} \tag{18}$$

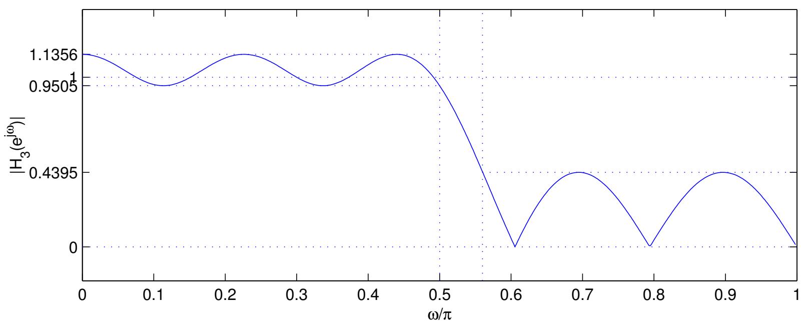
and because $h_2(n)$ is symmetric and of odd length.

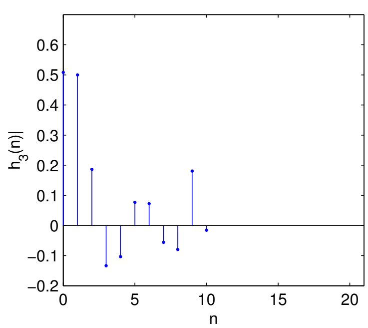
Step Three: We can obtain a minimum-phase filter by spectrally factoring the filter $h_2(n)$. If $h_3(n)$ is a spectral factor of $h_2(n)$, then we have

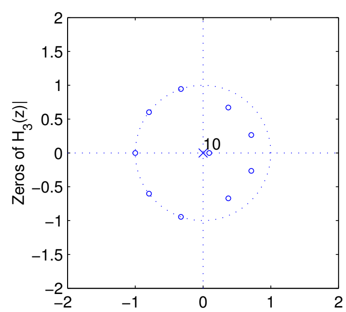
$$|H_3(e^{j\omega})|^2 = A_2(\omega). \tag{19}$$

or

$$\boxed{|H_3(e^{j\omega})| = \sqrt{A_2(\omega)}.} \tag{20}$$







This can be done with the Matlab command,

```
h3 = sfact(h2);
```

where the command `sfact` is described in the notes about spectral factorization. Then $|H_3(e^{j\omega})|$ satisfies the following bounds

$$\sqrt{1-\delta_1+\delta_2} \le |H_3(e^{j\omega})| \le \sqrt{1+\delta_1+\delta_2} \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{21}$$

$$|H_3(e^{j\omega})| \le \sqrt{2\,\delta_2} \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{22}$$

Note that $|H_3(e^{j\omega})|$ does not have an equi-ripple behavior in the pass-band. That is because the lower bound $\sqrt{1-\delta_1+\delta_2}$ and the upper bound $\sqrt{1+\delta_1+\delta_2}$ in the pass-band are not of the form $1+\delta_p$ and $1-\delta_p$.

Step Four: Finally, we need only scale $h_3(n)$ so that it has an equi-ripple behavior in the pass-band. The scaling

$$h(n) = C \cdot h_3(n), \qquad |H(e^{j\omega})| = C \cdot |H_3(e^{j\omega})| \tag{23}$$

gives the bounds

$$C\,\sqrt{1 - \delta_1 + \delta_2} \le |H(e^{j\omega})| \le C\,\sqrt{1 + \delta_1 + \delta_2} \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{24}$$

$$|H(e^{j\omega})| \le C\,\sqrt{2\,\delta_2} \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{25}$$

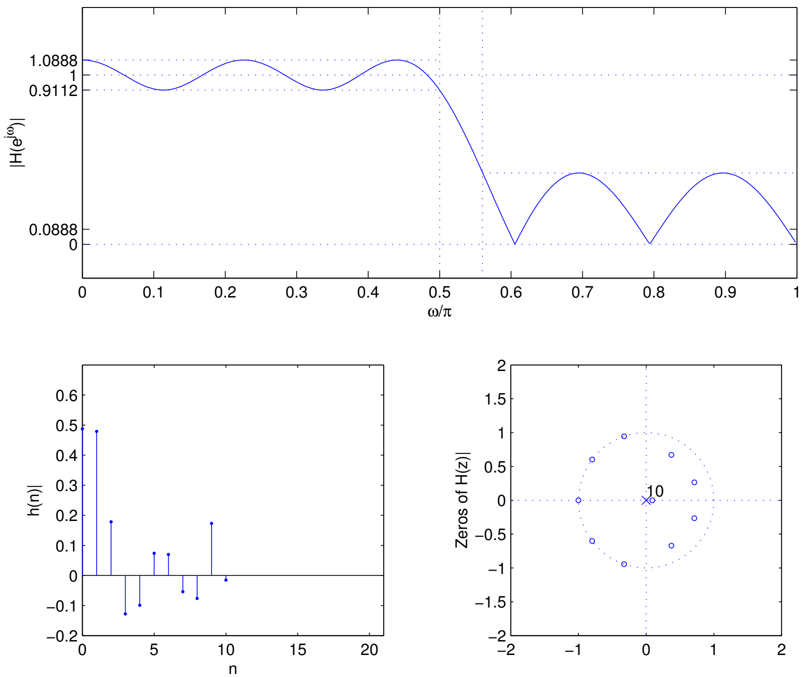
Setting the lower and upper bounds equal to $1 - \delta_p$ and $1 + \delta_p$ respectively, we get

$$1 - \delta_p = C\,\sqrt{1 - \delta_1 + \delta_2} \tag{26}$$

$$1 + \delta_p = C\,\sqrt{1 + \delta_1 + \delta_2}. \tag{27}$$

Solving for C, we obtain

$$\boxed{C = \frac{2}{\sqrt{1 + \delta_1 + \delta_2} + \sqrt{1 - \delta_1 + \delta_2}}.} \tag{28}$$



In Matlab we have

```
h  = h3*2/(sqrt(1+d1+d2)+sqrt(1-d1+d2));
```

With this value of C, we get the bounds

$$1 - \delta_p \le |H(e^{j\omega})| \le 1 + \delta_p \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{29}$$

$$|H(e^{j\omega})| \le \delta_s \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{30}$$

where

$$\delta_p = \frac{\sqrt{1+\delta_1+\delta_2} - \sqrt{1-\delta_1+\delta_2}}{\sqrt{1+\delta_1+\delta_2} + \sqrt{1-\delta_1+\delta_2}} \tag{31}$$

$$\delta_s = \frac{2\sqrt{2\,\delta_2}}{\sqrt{1+\delta_1+\delta_2} + \sqrt{1-\delta_1+\delta_2}}. \tag{32}$$

To ensure that $H(e^{j\omega})$ satisfies the given error bound we can invert the expressions for δ_p and δ_s to determine the corresponding target values δ_1, δ_2. Solving for δ_1 and δ_2 gives

$$\delta_1 = \frac{4\,\delta_p}{2 - \delta_s^2 + 2\,\delta_p^2} \tag{33}$$

$$\delta_2 = \frac{\delta_s^2}{2 - \delta_s^2 + 2\,\delta_p^2}. \tag{34}$$

Then, provided the linear-phase Type I filter $h_1(n)$ is designed according to the tolerances δ_1 and δ_2, the minimum-phase filter $h(n)$ will satisfy the constraints

$$||H(e^{j\omega})| - 1| \le \delta_p \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{35}$$

$$|H(e^{j\omega})| \le \delta_s \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{36}$$

Note that a Type I FIR filter specified to have a maximum error of Δ_1 and Δ_2 in the pass-band and stop-band respectively, can be designed using the Parks-McClellan (PM) algorithm with weights $K_p = 1/\Delta_1$, $K_s = 1/\Delta_2$. However, the resulting Type I FIR produced by the PM algorithm may have errors that are smaller than the specified tolerances. Therefore, the formulas (33,34) are to be used only to select the weighting parameters K_p and K_s.
The Matlab code for this example is combined into the following program.

```
N  = 11;
Kp = 1;
Ks = 2;
wp = 0.50*pi;
ws = 0.56*pi;
wo = (wp+ws)/2;
L  = 3000;
```

```
w  = [0:L]*pi/L;
D  = (w<=wo);
W  = Kp*(w<=wp) + Ks*(w>=ws);
[h1,del] = fircheb(2*N-1,D,W);
d1 = del/Kp;
d2 = del/Ks;

h2 = h1;
h2(N) = h2(N) + d2;

h3 = sfact(h2);

A  = sqrt(1+d1+d2);
B  = sqrt(1-d1+d2);
h  = h3*2/(A+B);

dp = (A-B)/(A+B);
ds = 2*sqrt(2*d2)/(A+B);
```

SUMMARY OF THE LIFTING PROCEDURE

To design a minimum-phase filter satisfying

$$||H(e^{j\omega})| - 1| \le \Delta_p \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{37}$$

$$|H(e^{j\omega})| \le \Delta_s \qquad \text{for} \quad \omega_s \le \omega \le \pi \tag{38}$$

the following algorithm can be used. This algorithm is sometimes referred to as the *lifting* procedure as it involves lifting the amplitude response $A_1(\omega)$ so that it can be spectrally factored.

1. Design a minimal-length Type I FIR filter $h_1(n)$ satisfying the constraints

   $$|A_1(\omega) - 1| \le \Delta_1 \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{39}$$

   $$|A_1(\omega)| \le \Delta_2 \qquad \text{for} \quad \omega_s \le \omega \le \pi. \tag{40}$$

   where

   $$\Delta_1 = \frac{4\,\Delta_p}{2 - \Delta_s^2 + 2\,\Delta_p^2} \tag{41}$$

   $$\Delta_2 = \frac{\Delta_s^2}{2 - \Delta_s^2 + 2\,\Delta_p^2}. \tag{42}$$

   This filter can be obtained with the Parks-McClellan algorithm using the pass-band and stop-band weights

   $$K_p = 1/\Delta_1, \qquad K_s = 1/\Delta_2. \tag{43}$$

2. Define

   $$\delta_1 := \max |A_1(\omega) - 1| \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{44}$$

   $$\delta_2 := \max |A_1(\omega)| \qquad \text{for} \quad \omega_s \le \omega \le \pi \tag{45}$$

and set

$$h_2(n) = h_1(n) + \delta_2 \cdot \delta(n - N + 1) \tag{46}$$

3. Compute a minimum-phase spectral factor $h_3(n)$ of $h_2(n)$ such that

$$|H_3(e^{j\omega})| = \sqrt{|H_2(e^{j\omega})|}. \tag{47}$$

4. Scale $h_3(n)$ to obtain the equi-ripple filter $h(n)$.

$$h(n) = C \cdot h_3(n) \tag{48}$$

where

$$C = \frac{2}{\sqrt{1 + \delta_1 + \delta_2} + \sqrt{1 - \delta_1 + \delta_2}}. \tag{49}$$

Note that when the spectral factorization is performed,

$$|H_3(e^{j\omega})| = \sqrt{|H_2(e^{j\omega})|} \tag{50}$$

the stop-band ripples become larger. If the spectral factor $h_3(n)$ is to have small ripples in the stop-band, the filter $h_2(n)$ must have ripples in the stop-band that are especially small.

In this example, we use the lifting procedure to design a minimum-phase FIR filter satisfying the specifications

$$||H(e^{j\omega})| - 1| \le \Delta_p \qquad \text{for} \quad 0 \le \omega \le \omega_p \tag{51}$$

$$|H(e^{j\omega})| \le \Delta_s \qquad \text{for} \quad \omega_s \le \omega \le \pi \tag{52}$$

where

$$\Delta_p = 0.03, \quad \Delta_s = 0.02, \quad \text{and} \quad \omega_p = 0.50\pi, \quad \omega_s = 0.56\pi. \tag{53}$$

Step one. Step one of the design processes calls for the design of a Type I linear-phase FIR filter meeting the transformed specifications. The transformed specifications for this example are

$$\Delta_1 = 0.0600, \quad \Delta_2 = 1.9986 \cdot 10^{-4}. \tag{54}$$

Therefore the weighting parameters K_p and K_s can be chosen to be

$$K_p = 16.6783, \qquad K_s = 5.0035 \cdot 10^3. \tag{55}$$

Notice that the stop-band is weighted much more than the pass-band. If we design a length 79 Type I FIR filter with these weighting parameters and the specified pass-band and stop-band edges, we find that the resulting pass-band and stop-band errors are

$$\delta_1 = 0.0605, \qquad \delta_2 = 2.0177 \cdot 10^{-4} \tag{56}$$

which violated the specified tolerance margins Δ_1 and Δ_2. But if we design a length 81 Type I FIR filter with the same weighting parameters, then we find that

$$\delta_1 = 0.0571, \qquad \delta_2 = 1.9032 \cdot 10^{-4} \tag{57}$$

which satisfy the transformed specifications. We call this filter $h_1(n)$.

Step two. In step two, we just need to add δ_2 to the center term of $h_1(n)$ and call the result $h_2(n)$. Then the zeros of $H_2(z)$ that lie on the unit circle will be of even multiplicity.

Step three. In step three we find the minimum-phase spectral factor of $h_2(n)$ and call the result $h_3(n)$. The length of the $h_3(n)$ will be 41.

Step four. In step three we scale the filter $h_3(n)$ by C, which for this example is $C = 1.0003127316$.

This solution found using the lifting procedure yields a minimum-phase FIR filter of length 41 with

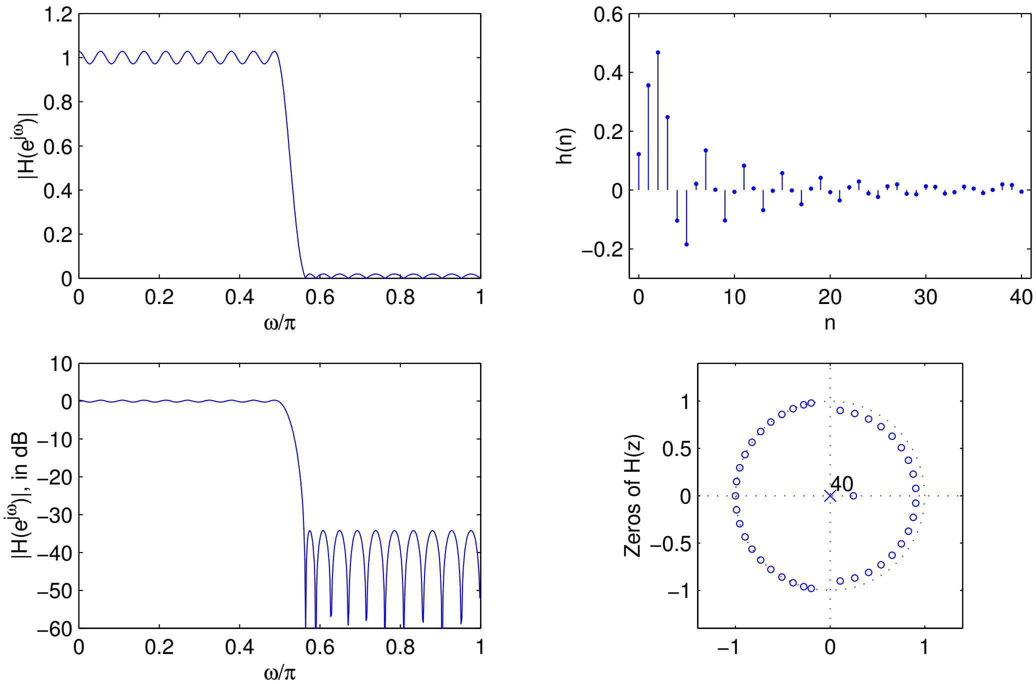
$$||H(e^{j\omega})| - 1| \leq \delta_p \qquad \text{for} \quad 0 \leq \omega \leq \omega_p \tag{58}$$

$$|H(e^{j\omega})| \leq \delta_s \qquad \text{for} \quad \omega_s \leq \omega \leq \pi \tag{59}$$

where

$$\delta_p = 0.0286, \qquad \delta_s = 0.0195. \tag{60}$$

The minimum-phase filter is illustrated in the figure.

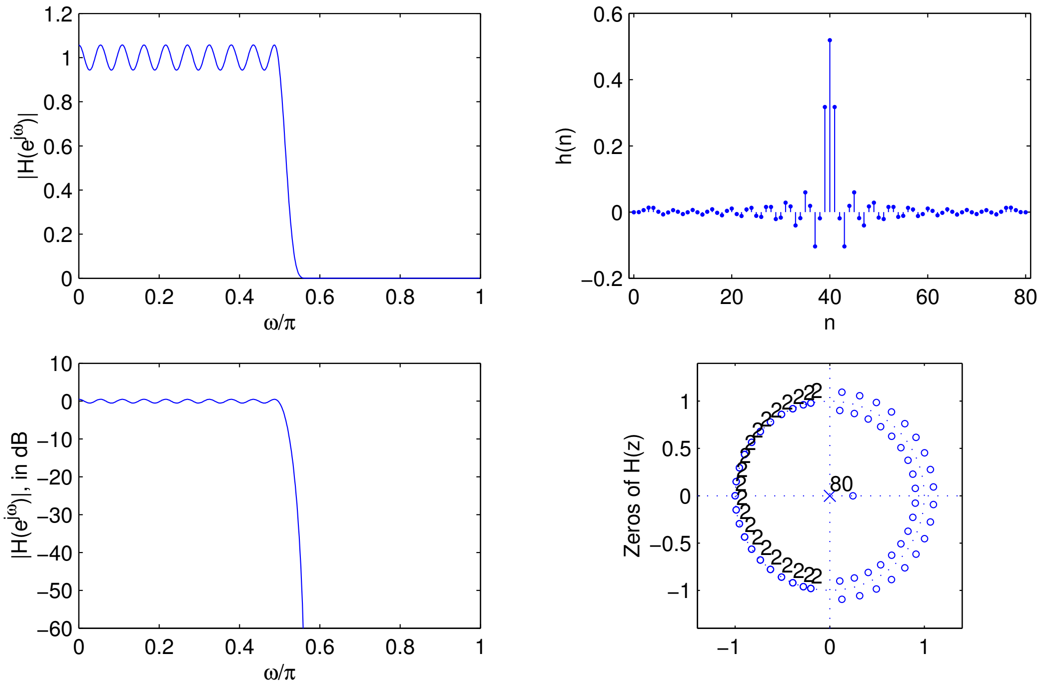


The filter was obtained with the following Matlab code.

```
% Set up specifications

Dp = 0.03;                            % Dp: pass-band spec for h
Ds = 0.02;                            % Ds: stop-band spec for h
D1 = (4*Dp)/(2-Ds^2 + 2*Dp^2);        % D1: pass-band spec for h1
D2 = (Ds^2)/(2-Ds^2 + 2*Dp^2);        % D2: stop-band spec for h1
Kp = 1/D1;                            % Kp: pass-band weight for h1
Ks = 1/D2;                            % Ks: stop-band weight for h1
wp = 0.50*pi;
ws = 0.56*pi;
wo = (wp+ws)/2;
L  = 3000;
w  = [0:L]*pi/L;
D  = (w<=wo);
W  = Kp*(w<=wp) + Ks*(w>=ws);

% -----------------------------------------------------------

% STEP 1 - Design a Type I FIR filter of minimal length that
%          meets the transformed specifications.

N  = 40;
[h1,del] = fircheb(2*N-1,D,W);
d1 = del/Kp;                         % d1: actual pass-band error for h1
d2 = del/Ks;                         % d2: actual stop-band error for h1

[d1 d2] % ---> A length 79 filter can not satisfy the specifications.
```

```
% Try a filter of length 81

N  = 41;
[h1,del] = fircheb(2*N-1,D,W);
d1 = del/Kp;                    % d1: actual pass-band error for h1
d2 = del/Ks;                    % d2: actual stop-band error for h1

[d1 d2] % ---> This length 81 filter does satisfy the specifications.

figure(1), mfigNS(h1), orient tall, print -depsc lift2A

% -----------------------------------------------------------

% STEP 2 - "lift" the frequency response

h2 = h1;
h2(N) = h2(N) + d2;             % "lift" h1

figure(2), mfigNS(h2), orient tall, print -depsc lift2B

% -----------------------------------------------------------

% STEP 3 - Compute a minimum-phase spectral factor

h3 = sfact(h2);                 % spectral factorization of h2

figure(3), mfigNS(h3), orient tall, print -depsc lift2C

% -----------------------------------------------------------

% STEP 4 - Scale h(n)

A  = sqrt(1+d1+d2);
B  = sqrt(1-d1+d2);
h  = h3*2/(A+B);                % scaling of h3

figure(4), mfigNS(h), orient tall, print -depsc lift2D

% -----------------------------------------------------------

dp = (A-B)/(A+B);               % actual pass-band error for h
ds = 2*sqrt(2*d2)/(A+B);        % actual stop-band error for h
```

To compare the minimum-phase to a linear-phase filter with the same length, we obtain the linear-phase filter shown in the figure.



The pass-band and stop-band errors for the linear-phase filter are:

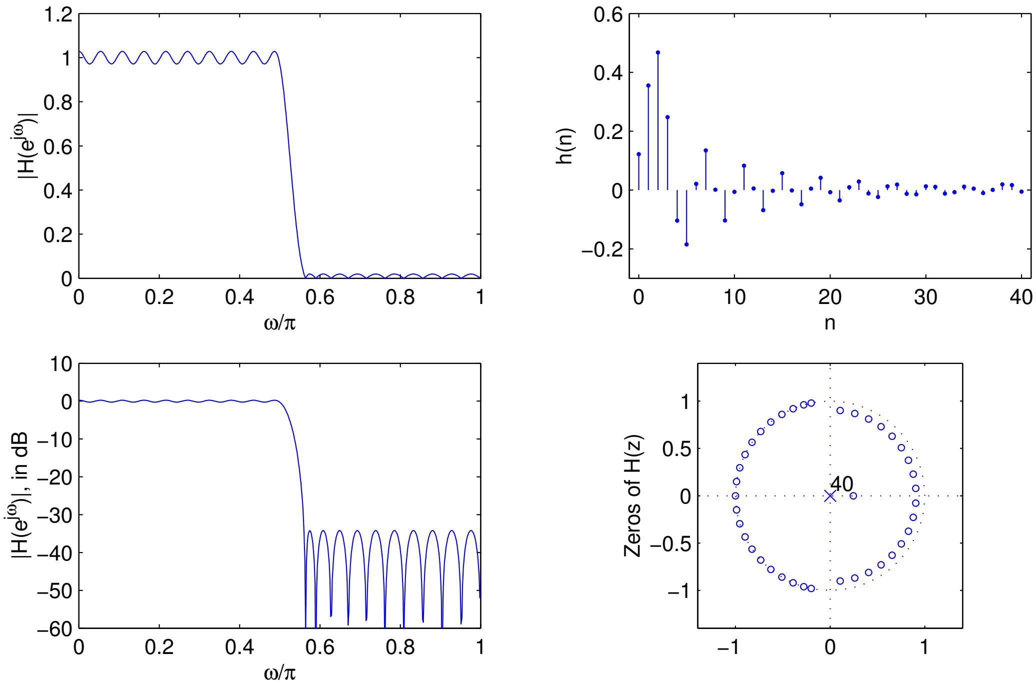
$$\delta_p = 0.0461, \qquad \delta_s = 0.0307. \tag{61}$$

which are greater than the those of the minimum-phase solution. This linear-phase filter was obtained with the following Matlab commands.

```
N  = 41;
Dp = 0.03;
Ds = 0.02;
Kp = 1/Dp;
Ks = 1/Ds;
wp = 0.50*pi;
ws = 0.56*pi;
wo = (wp+ws)/2;
L  = 3000;
w  = [0:L]*pi/L;
D  = (w<=wo);
W  = Kp*(w<=wp) + Ks*(w>=ws);
[h,del] = fircheb(N,D,W);
dp = del/Kp;
ds = del/Ks;
```

Therefore, we observe that by using a minimum-phase filter obtained using the lifting procedure, we get a filter that has (1) a better frequency response magnitude and (2) a shorter delay. Notice that the impulse response of the minimum-phase filter has most of its energy in the first few samples. The trade-off is that the minimum-phase filter introduces phase-distortion, while the linear-phase filter does not. The relative importance of delay versus phase-distortion depends on the application. In image processing for example, it is usually important to use filters that have symmetric impulse responses. When a filter with a non-symmetric impulse response is used in image processing it can causes distortion that is visible (around edges in particular). However, in other applications, the phase-distortion is less important. For example, in audio processing and compression, minimum-phase filters are often used because the ear is less sensitive to phase distortion.

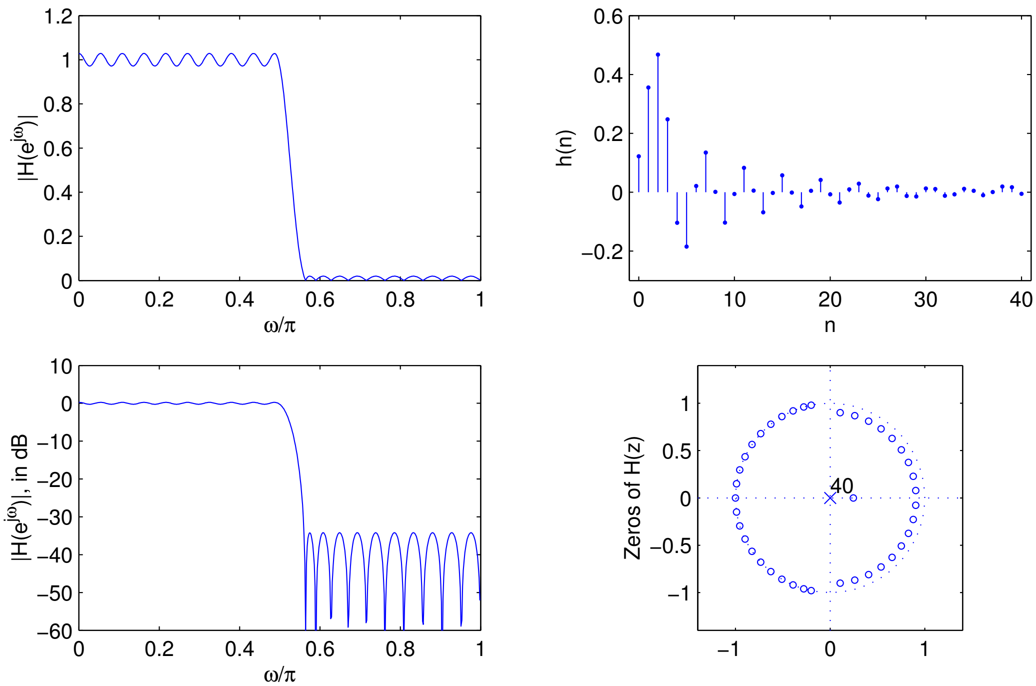
It was mentioned in the introduction that a minimum-phase filter can always be obtained by reflecting zeros z_k that lie outside the unit circle to $1/z_k$. If this is done with the length 41 linear-phase in the previous figure, one obtains the following filter.



The delay is reduced, but the frequency response magnitude $|H_1(e^{j\omega})|$ is not changed, and is therefore not optimal. This minimum-phase filter was obtained with the following Matlab commands.

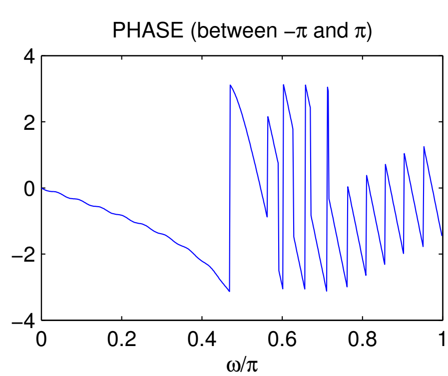
```
r = roots(h);
k = abs(r)>1.01;
r1 = r(k);          % roots outside |z|=1
r2 = r(~k);         % roots on or inside |z|=1
s = [1./r1; r2];
h2 = poly(leja(s));
h2 = h2*norm(h)/norm(h2);
```

Let us examine the phase of the minimum-phase filter we obtained earlier (illustrated here again for convenience).



We can compute the complex-valued frequency response with the the `freqz` command. Then the phase can be obtained with the `angle` command.

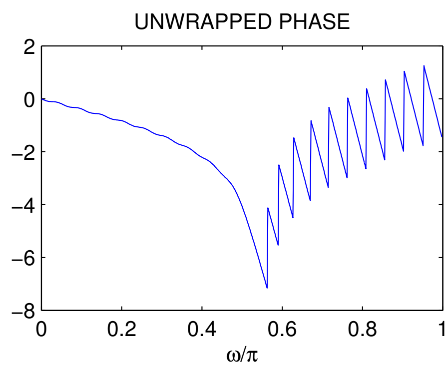
```
[H,w] = freqz(h);
plot(w/pi,angle(H))
```



PHASE UNWRAPPING

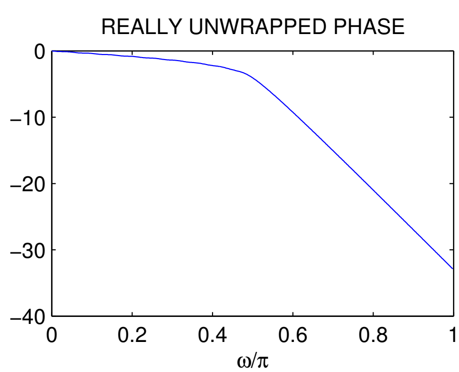
The plot of the phase has many discontinuities because the angle of a complex number provided by the `angle` command is between $-\pi$ and π. That causes jumps of size 2π. The processes of removing those jumps is called *phase unwrapping* and is implemented with the command `unwrap`.

```
plot(w/pi,unwrap(angle(H)))
```



There are still some jumps of size π corresponding to zeros on the unit circle. These can be removed by multiplying the phase by 2, using the `unwrap` command, and then dividing the result by 2.

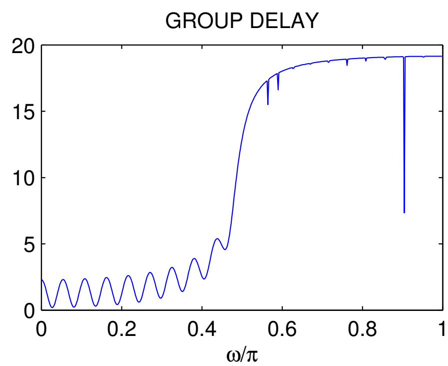
```
plot(w/pi,unwrap(angle(H)*2)/2)
```



GROUP DELAY

The *group delay* of a filter is the negative derivative of the phase. It indicates the delay caused by the filter measured in samples. It can be computed with the Matlab command `grpdelay`.

```
[G,w] = grpdelay(h);
```



Because the value of the magnitude is very small in the stop-band (ideally, it is zero), the phase and group delay in the stop-band is actually not so important.

References

[1] O. Herrmann and H. W. Schuessler. Design of nonrecursive filters with minimum phase. *Electron. Lett.*, 6(11):329–330, 28th May 1970. Also in [2].

[2] L. R. Rabiner and C. M. Rader, editors. *Digital Signal Processing*. IEEE Press, 1972.