Chapter 1

Basic FIR Filter Design

Overview

In this chapter we discuss the basic principles of FIR filter design. We concentrate mostly on lowpass filters, but most of the results apply to other response types as well. We discuss the basic trade offs and the degrees of freedom available for FIR filter design. We motivate the use of optimal designs and introduce both optimal equiripple and optimal least-squares designs. We then discuss optimal minimum-phase designs as a way of surpassing in some sense comparable optimal linear-phase designs. We introduce sloped equiripple designs as a compromise to obtain equiripple passband yet non-equiripple stopband. We also mention a few caveats with equiripple filter design. We end with an introductory discussion of different filter structures that can be used to implement an FIR filter in hardware.

The material presented here is based on [1] and [2]. However, it has been expanded and includes newer syntax and features from the Filter Design Toolbox.

1.1 Why FIR filters?

There are many reasons why FIR filters are very attractive for digital filter design. Some of them are:

- Simple robust way of obtaining digital filters

- Inherently stable when implemented non recursively
- Free of limit cycles when implemented non recursively
- Easy to attain linear phase
- Simple extensions to multirate and adaptive filters
- Relatively straight-forward to obtain designs to match custom magnitude responses
- Some vendors and specialized hardware only support FIR
- Low sensitivity to quantization effects compared to many IIR filters

FIR filters have some drawbacks however. The most important is that they can be computationally expensive to implement. Another is that they have a long transient response. It is commonly thought that IIR filters must be used when computational power is at a premium. This is certainly true in some cases. However, in many cases, the use of multistage/multirate techniques can yield FIR implementations that can compete (and even surpass) IIR implementations while retaining the nice characteristics of FIR filters such as linear-phase, stability, and robustness to quantization effects.* However, these efficient multistage/multirate designs tend to have very large transient responses, so depending on the requirements of the filter, IIR designs may still be the way to go.

In terms of the long transient response, we will show in Chapter 2 that minimum-phase FIR filters can have a shorter transient response than comparable IIR filters.

1.2 Lowpass filters

The ideal lowpass filter is one that allows through all frequency components of a signal below a designated cutoff frequency ω_c, and rejects all frequency components of a signal above ω_c.

* That being said, there are also modern IIR design techniques that lend themselves to efficient multirate implementations and are extremely computationally efficient. We are referring here to more traditional IIR designs implemented using direct-form I or II structures, possibly in cascaded second-order section form.

Its frequency response satisfies

$$H_{\mathrm{LP}}(e^{j\omega}) = \begin{cases} 1, & 0 \le \omega \le \omega_c \\ 0, & \omega_c < \omega \le \pi \end{cases} \tag{1.1}$$

The impulse response of the ideal lowpass filter (1.1) can easily be found to be [3]

$$h_{\mathrm{LP}}[n] = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty < n < \infty. \tag{1.2}$$

1.2.1 FIR lowpass filters

Because the impulse response required to implement the ideal lowpass filter is infinitely long, it is impossible to design an ideal FIR lowpass filter.

Finite length approximations to the ideal impulse response lead to the presence of ripples in both the passband ($\omega < \omega_c$) and the stopband ($\omega > \omega_c$) of the filter, as well as to a nonzero transition width between the passband and stopband of the filter (see Figure 1.1).

1.2.2 FIR filter design specifications

Both the passband/stopband ripples and the transition width are undesirable but unavoidable deviations from the response of an ideal lowpass filter when approximating with a finite impulse response. Practical FIR designs typically consist of filters that meet certain design specifications, i.e., that have a transition width and maximum passband/stopband ripples that do not exceed allowable values.

In addition, one must select the filter order, or equivalently, the length of the truncated impulse response.

A useful metaphor for the design specifications in FIR design is to think of each specification as one of the angles in a triangle as in Figure 1.2 *.

The metaphor is used to understand the degrees of freedom available when designating design specifications. Because the sum of the angles is fixed, one can at most select the values of two of the specifications. The third specification will be determined by the design algorithm utilized.

* For the ripples we should more generally speak of some measure or norm of them. The peak ripple corresponding to the $\mathcal{L}_\infty$-norm is the most commonly used measure, but other norms are possible.

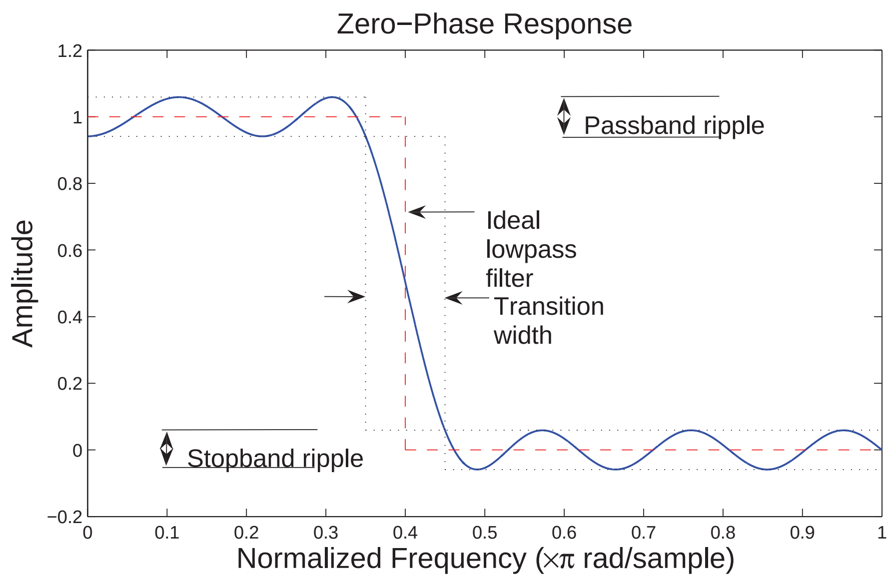

Figure 1.1: *Illustration of the typical deviations from the ideal lowpass filter when approximating with an FIR filter,* $\omega_c = 0.4\pi$.

Moreover, as with the angles in a triangle, if we make one of the specifications larger/smaller, it will impact one or both of the other specifications.

Example 1 *As an example, consider the design of an FIR filter that meets the following specifications:*

Specifications Set 1

1. *Cutoff frequency:* 0.4π *rad/sample*
2. *Transition width:* 0.06π *rad/sample*
3. *Maximum passband/stopband ripple: 0.05*

*The filter can easily be designed with the truncated-and-windowed impulse response algorithm (a.k.a. the "window method") if we use a Kaiser window**:

* Notice that when specifying frequency values in MATLAB, the factor of π should be omitted.

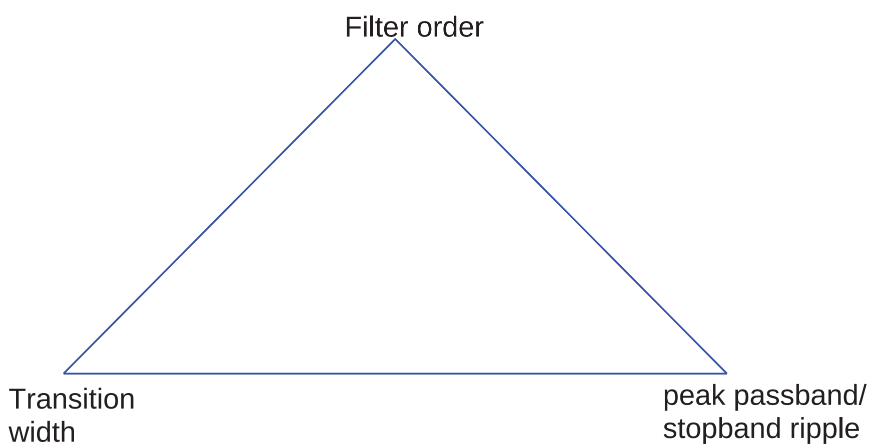

Figure 1.2: *FIR design specifications represented as a triangle.*

```
Fp = 0.4 - 0.06/2; Fst = 0.4 + 0.06/2;
Hf = fdesign.lowpass('Fp,Fst,Ap,Ast',Fp,Fst,0.05,0.05,'linear');
design(Hf,'kaiserwin');
```

The zero-phase response of the filter is shown in Figure 1.3. Note that since we have fixed the allowable transition width and peak ripples, the order is determined for us.

*Close examination at the passband-edge frequency**, $\omega_p = 0.37\pi$, *and at the stopband-edge frequency,* $\omega_s = 0.43\pi$, *shows that the peak passband/stopband ripples are indeed within the allowable specifications. Usually the specifications are exceeded because the order is rounded to the next integer greater than the actual value required.*

* The passband-edge frequency is the boundary between the passband and the transition band. If the transition width is T_w, the passband-edge frequency ω_p is given in terms of the cutoff frequency ω_c by $\omega_p = \omega_c - T_w/2$. Similarly, the stopband-edge frequency is given by $\omega_s = \omega_c + T_w/2$.

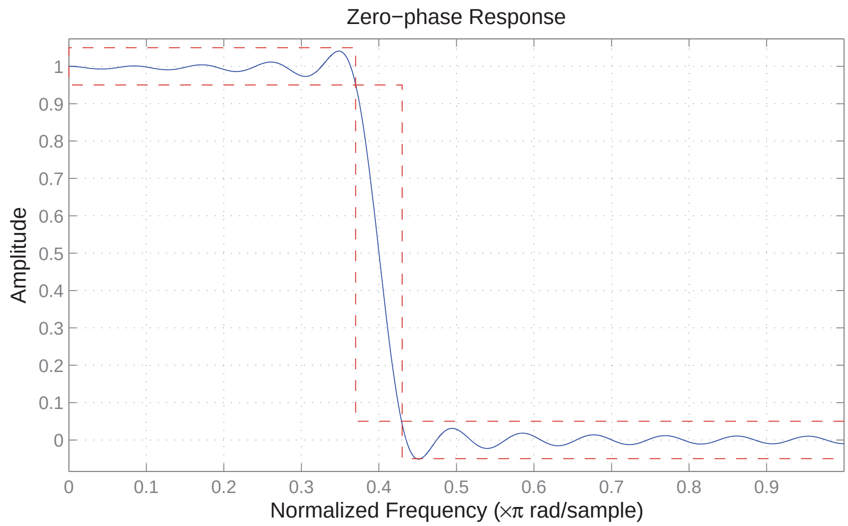

Figure 1.3: *Kaiser window design meeting prescribed specifications.*

1.2.3 Working with Hertz rather than normalized frequency

In many applications the specifications are given in terms of absolute frequency in Hertz rather than in terms of normalized frequency. Conversion between one and the other is straightforward. Recall that normalized frequency is related to absolute frequency by

$$\omega = \frac{2\pi f}{f_s}$$

where f is absolute frequency in cycles/second, f_s is the sampling frequency in samples/second, and ω is normalized frequency in radians/sample.

Suppose the specifications for a filter design problem include a passband frequency of 250 Hz, a stopband frequency of 300 Hz, and a sampling frequency of 1 kHz. The corresponding normalized passband and stopband frequencies are 0.5π and 0.6π respectively.

However, it is not necessary to perform such conversion manually. It is possible to specify design parameters directly in Hertz.

For example, the following two filters H1 and H2 are identical.

```
Hf  = fdesign.lowpass('Fp,Fst,Ap,Ast',.5,.6,1,80);
```

```
H1  = design(Hf);
Hf2 = fdesign.lowpass('Fp,Fst,Ap,Ast',250,300,1,80,1000);
H2  = design(Hf2);
```

Notice that we don't add `'Fs'` to the string `'Fp,Fst,Ap,Ast'` (or any other specification string) when we specify parameters in Hertz. Simply appending the sampling frequency to the other design parameters indicates that all frequencies specified are given in Hertz.

1.3 Optimal FIR filter design

While the truncated-and-windowed impulse response design algorithm is very simple and reliable, it is not optimal in any sense. The designs it produces are generally inferior to those produced by algorithms that employ some optimization criteria in that it will have greater order, greater transition width or greater passband/stopband ripples. Any of these is typically undesirable in practice, therefore more sophisticated algorithms come in handy.

1.3.1 Optimal FIR designs with fixed transition width and filter order

Optimal designs are computed by minimizing some measure of the deviation between the filter to be designed and the ideal filter. The most common optimal FIR design algorithms are based on fixing the transition width and the order of the filter. The deviation from the ideal response is measured only by the passband/stopband ripples. This deviation or error can be expressed mathematically as [4]

$$E(\omega) = H_a(\omega) - H_{\mathrm{LP}}(e^{j\omega}), \quad \omega \in \Omega$$

where $H_a(\omega)$ is the zero-phase response of the designed filter and $\Omega = [0,\omega_p] \cup [\omega_s,\pi]$. It is still necessary to define a measure to determine "the size" of $E(\omega)$ - the quantity we want to minimize as a result of the optimization. The most often used measures are the $\mathcal{L}_\infty$-norm ($\|E(\omega)\|_\infty$ - minimax designs) and the $\mathcal{L}_2$-norm ($\|E(\omega)\|_2$ - least-squares designs).

In order to allow for different peak ripples in the passband and stopband, a weighting function, $W(\omega)$ is usually introduced,

$$E_W(\omega) = W(\omega)[H_a(\omega) - H_{\mathrm{LP}}(e^{j\omega})], \quad \omega \in \Omega$$

Linear-phase designs

A filter with linear-phase response is desirable in many applications, notably image processing and data transmission. One of the desirable characteristics of FIR filters is that they can be designed very easily to have linear phase. It is well known [5] that linear-phase FIR filters will have impulse responses that are either symmetric or antisymmetric. For these types of filters, the zero-phase response can be determined analytically [5], and the filter design problem becomes a well behaved mathematical approximation problem [6]: Determine the best approximation to a given function (the ideal lowpass filter's frequency response) by means of a polynomial (the FIR filter) of a given order. By "best" it is meant the one which minimizes the difference between them - $E_W(\omega)$ - according to a given measure.

The `equiripple` design implements an algorithm developed in [7] that computes a solution to the design problem for linear-phase FIR filters in the $\mathcal{L}_\infty$-norm case. The design problem is essentially to find a filter that minimizes the *maximum* error between the ideal and actual filters. This type of design leads to so-called equiripple filters, i.e. filters in which the peak deviations from the ideal response are all equal.

The `firls` design implements an algorithm to compute solution for linear-phase FIR filters in the $\mathcal{L}_2$-norm case. The design problem is to find a filter that minimizes the energy of the error between ideal and actual filters.

Equiripple filters

Linear-phase equiripple filters are desirable because they have the smallest maximum deviation from the ideal filter when compared to all other linear-phase FIR filters of the same order. Equiripple filters are ideally suited for applications in which a specific tolerance must be met. For example, if it is necessary to design a filter with a given minimum stopband attenuation or a given maximum passband ripple.

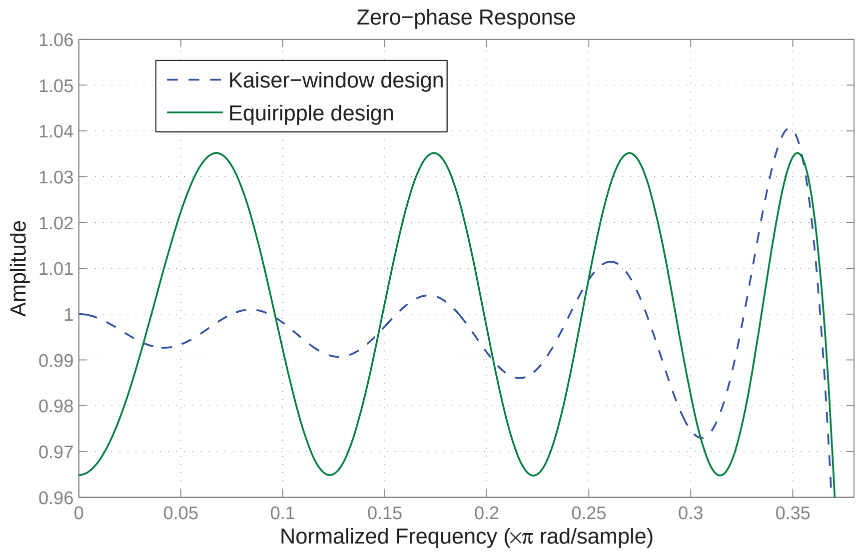

Figure 1.4: *Passband ripple for of both the Kaiser-window-designed FIR filter and the equiripple-designed FIR filter.*

Example 2 *The Kaiser-window design of Example 1 was of 42nd order. With this same order, an equiripple filter (with fixed transition width) can be designed that is superior to the Kaiser-window design:*

```
Fp  = 0.4 - 0.06/2; Fst = 0.4 + 0.06/2;
Hf  = fdesign.lowpass('N,Fp,Fst',42,Fp,Fst);
Heq = design(Hf,'equiripple');
```

Figure 1.4 shows the superposition of the passband details for the filters designed with the Kaiser window and with the `equiripple` *design. Clearly the maximum deviation is smaller for the* `equiripple` *design. In fact, since the filter is designed to minimize the maximum ripple (minimax design), we are guaranteed that no other linear-phase FIR filter of 42nd order will have a smaller peak ripple for the same transition width.*

We can measure the passband ripple and stopband attenuation in dB units using the `measure` *command,*

```
Meq = measure(Heq);
```

If we compare the measurements of the equiripple design to those of the Kaiser-window design, we can verify for instance that the equiripple design provides a minimum stopband attenuation of 29.0495 dB compared to 25.8084 dB for the Kaiser-window design.

Least-squares filters

Equiripple designs may not be desirable if we want to minimize the energy of the error (between ideal and actual filter) in the passband/stopband. Consequently, if we want to reduce the energy of a signal as much as possible in a certain frequency band, least-squares designs are preferable.

Example 3 *For the same specifications,* `Hf`*, as the equiripple design of Example 2, a least-squares FIR design can be computed from*

```
Hls = design(Hf,'firls');
```

The stopband energy for this case is given by

$$E_{sb} = \frac{2}{2\pi} \int_{0.43\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

where $H(e^{j\omega})$ is the frequency response of the filter.

In this case, the stopband energy for the equiripple filter is approximately 3.5214-004 while the stopband energy for the least-squares filter is 6.6213e-005. (As a reference, the stopband energy for the Kaiser-window design for this order and transition width is 1.2329e-004).

The stopband details for both equiripple design and the least-squares design are shown in Figure 1.5.

So while the equiripple design has less peak error, it has more "total" error, measured in terms of its energy. However, although the least-squares design minimizes the energy in the ripples in both the passband and stopband, the resulting peak passband ripple is always larger than that of a comparable equiripple design. Therefore there is a larger disturbance on the signal to be filtered for a portion of the frequencies that the filter should allow to pass (ideally undisturbed). This is a drawback of least-squares designs. We will see in Section 1.3.5 that a possible compromise is to design equiripple filters in such a way that the maximum ripple in the passband is minimized, but with a sloped stopband that can reduce the stopband energy in a manner comparable to a least-squares design.

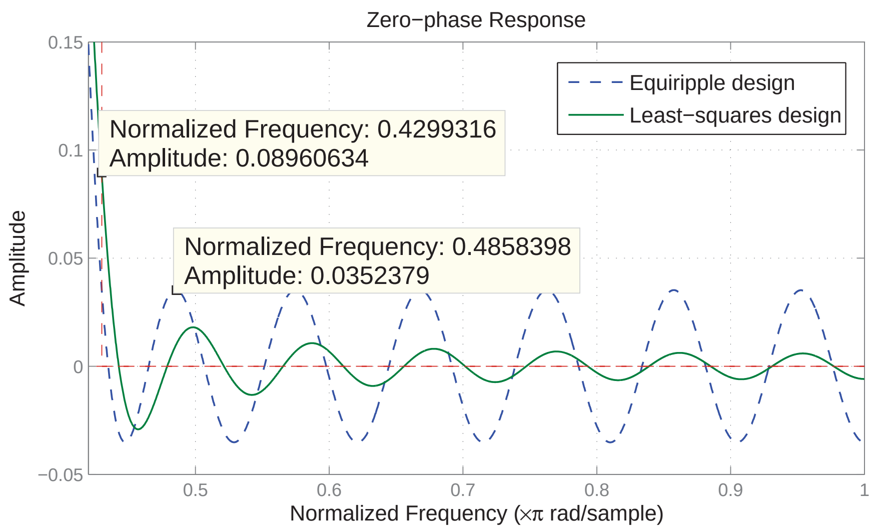

Figure 1.5: *Comparison of an optimal equiripple FIR design and an optimal least-squares FIR design. The equiripple filter has a smaller peak error, but larger overall error.*

Using weights

Both equiripple and least-squares designs can be further controlled by using weights to instruct the algorithm to provide a better approximation to the ideal filter in certain bands. This is useful if it is desired to have less ripple in one band than in another.

Example 4 *In Example 2 above, the filter that was designed had the same ripples in the passband and in the stopband. This is because we implicitly were using a weight of one for each band. If it is desired to have a stopband ripple that is say ten times* smaller *than the passband ripple, we must give a weight that is ten times* larger:

```
Heq2 = design(Hf,'equiripple','Wpass',1,'Wstop',10);
```

The result is plotted in Figure 1.6.

It would be desirable to have an analytic relation between the maximum ripples in a band and the weight in such band. Unfortunately no such relation exists. If the design specifications require a specific maximum ripple amount, say δ_p in the passband and δ_s in the stopband (both in linear units, not decibels), for a lowpass filter we can proceed as follows:

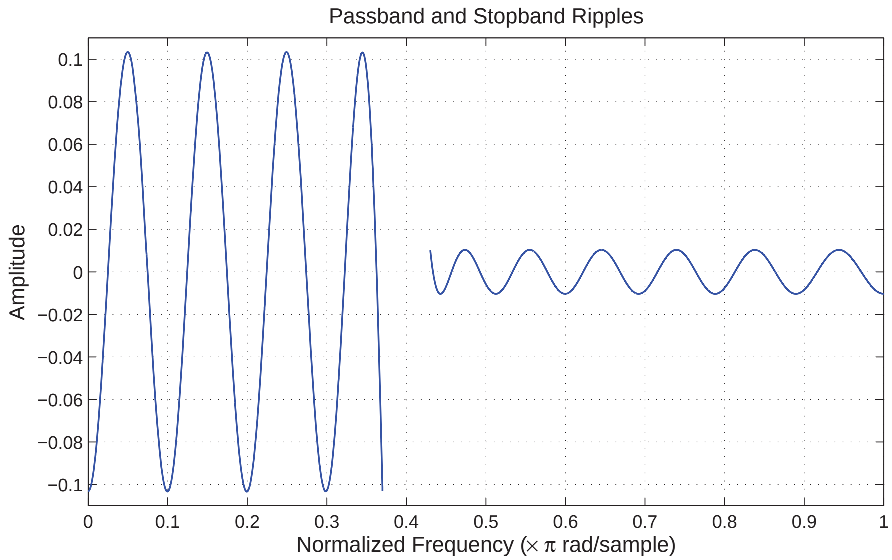

Figure 1.6: *Passband and stopband ripples obtained from weighing the stopband 10 times higher than the passband.*

1. Set the passband weight to one.

2. Set the stopband weight to $\frac{\delta_p}{\delta_s}$.

since both the filter order and the transition width are assumed to be fixed, this will *not* result in the desired ripples unless we are very lucky. However, the relative amplitude of the passband ripple relative to the stopband ripple will be correct. In order to obtain a ripple of δ_p in the passband and δ_s in the stopband we need to vary either the filter order or the transition width.

The procedure we have just described requires trial-and-error since either the filter order or the transition width may need to be adjusted many times until the desired ripples are obtained. Instead of proceeding in such manner, later we will describe ways of designing filters for given passband/stopband ripples and either a fixed transition width or a fixed filter order.

For least-squares designs, the relative weights control not the amplitude of the ripple but its energy relative to the bandwidth it occupies. This means that if we weigh the stopband ten times higher than the passband, the energy in the stopband relative to the stopband bandwidth will be 10 times smaller than the energy of the ripples in the passband relative to the passband bandwidth. For the case of lowpass filters these means that

$$\frac{E_{\mathrm{sb}}}{\pi - \omega_s} = \frac{2}{2\pi(\pi - \omega_s)} \int_{\omega_s}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

will be ten times smaller than

$$\frac{E_{\mathrm{pb}}}{\omega_p} = \frac{2}{2\pi(\omega_p)} \int_{0}^{\omega_p} \left| H(e^{j\omega}) \right|^2 d\omega.$$

Minimum-phase designs

One of the advantages of FIR filters, when compared to IIR filters, is the ability to attain exact linear phase in a straightforward manner. As we have already mentioned, the linear phase characteristic implies a symmetry or antisymmetry property for the filter coefficients. Nevertheless, this symmetry of the coefficients constraints the possible designs that are attainable. This should be obvious since for a filter with $N+1$ coefficients, only $N/2+1$ of these coefficients are freely assignable (assuming N is even). The remaining $N/2$ coefficients are immediately determined by the linear phase constraint.

If one is able to relax the linear phase constraint (i.e. if the application at hand does not require a linear phase characteristic), it is possible to design minimum-phase equiripple filters that are superior to optimal equiripple linear-phase designs based on a technique described in [8].

Example 5 *For the same specification set of Example 2 the following minimum-phase design has both smaller peak passband ripple and smaller peak stopband ripple* than the linear-phase equiripple design of that example:*

```
Hmin = design(Hf,'equiripple','Wpass',1,'Wstop',10,...
  'minphase',true);
```

It is important to note that this is not a totally unconstrained design. The minimum-phase requirement restricts the resulting filter to have all

* This can easily be verified using the measure command.

its zeros on or inside the unit circle.* However, the design is optimal in the sense that it satisfies the minimum-phase alternation theorem [9].

Having smaller ripples for the same filter order and transition width is not the only reason to use a minimum-phase design. The minimum-phase characteristic means that the filter introduces the lowest possible phase offset (that is, the smallest possible transient delay) to a signal being filtered.

Example 6 *Compare the delay introduced by the linear-phase filter of Example 2 to that introduced by the minimum-phase filter designed above. The signal to be filtered is a sinusoid with frequency* 0.1π *rad/sample.*

```
n    = 0:500;
x    = sin(0.1*pi*n');
yeq  = filter(Heq,x);
ymin = filter(Hmin,x);
```

The output from both filters are plotted overlaid in Figure 1.7. The delay introduced is equal to the group delay of the filter at that frequency. Since group-delay is the negative of the derivative of phase with respect to frequency, the group-delay of a linear-phase filter is a constant equal to half the filter order. This means that all frequencies are delayed by the same amount. On the other hand, minimum-phase filters do not have constant group-delay since their phase response is not linear. The group-delays of both filters can be visualized using `fvtool(Heq,Hmin,'Analysis','Grpdelay');`*. The plot of the group-delays is shown in Figure 1.8.*

1.3.2 Optimal equiripple designs with fixed transition width and peak passband/stopband ripple

We have seen that the optimal equiripple designs outperform Kaiser-window designs for the same order and transition width. The differences are even more dramatic when the passband ripple and stopband ripple specifications are different. The reason is that the truncated-and-windowed impulse response methods always give a result with approximately the same passband and stopband peak ripple. Therefore, always the more stringent

* Given any linear-phase FIR filter with non negative zero-phase characteristic, it is possible to extract the minimum-phase spectral factor using the firminphase function.

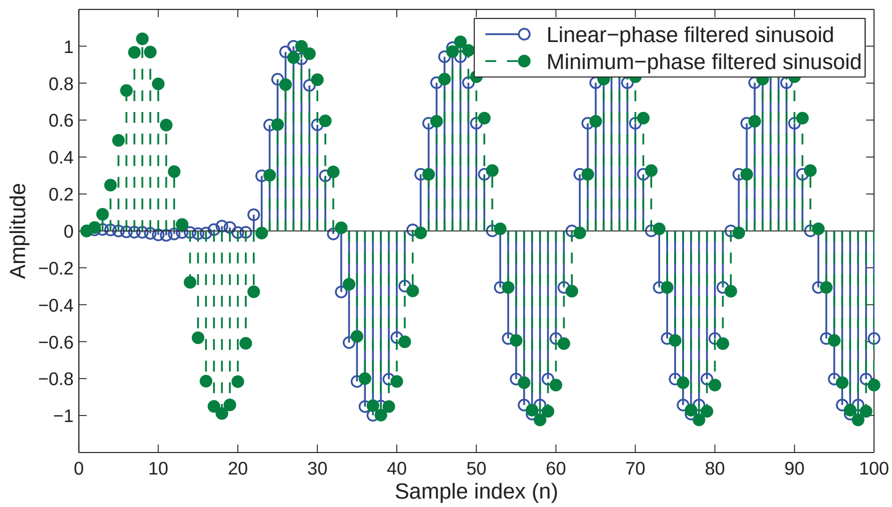

Figure 1.7: *Sinusoid filtered with a linear-phase filter and a minimum-phase filter of the same order.*

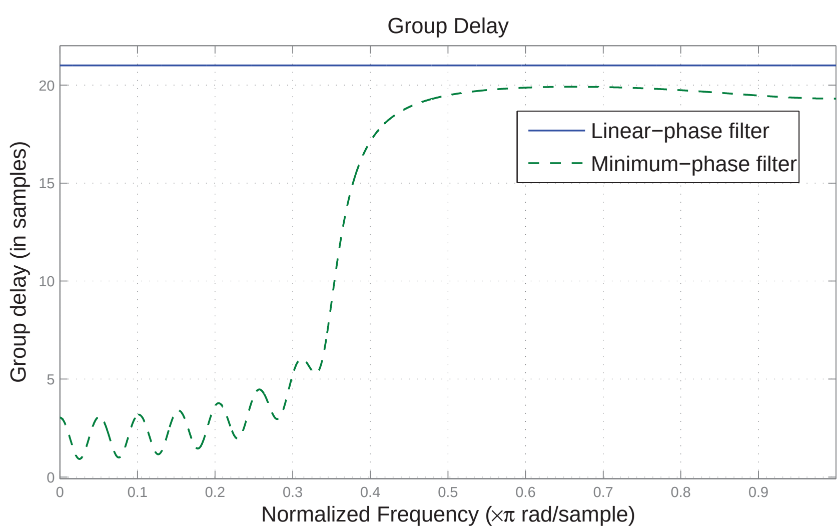

Figure 1.8: *Group-delay of a linear-phase filter and a minimum-phase filter of the same order.*

peak ripple constraint is satisfied, resulting in exceeding (possibly significantly) all other ripple constraints at the expense of unnecessarily large filter order.

To illustrate this, we turn to a different equiripple design in which both the peak ripples and the transition width are fixed. Referring back to the triangle in Figure 1.2, this means the resulting filter order will come from the design algorithm.

Example 7 *Consider the following specifications:*

Specifications Set 2

1. *Cutoff frequency:* 0.375π *rad/sample*
2. *Transition width:* 0.15π *rad/sample*
3. *Maximum passband ripple: 0.13 dB*
4. *Minimum stopband attenuation: 60 dB*

An equiripple design of this filter,

```
Fp  = 0.375 - 0.15/2; Fst = 0.375 + 0.15/2;
Hf  = fdesign.lowpass('Fp,Fst,Ap,Ast',Fp,Fst,0.13,60);
Heq = design(Hf,'equiripple');
cost(Heq)
```

results in a filter of 37th order (38 taps) as indicated by the `cost` *command. By comparison, a Kaiser-window design requires a 49th order filter (50 taps) to meet the same specifications. The passband details can be seen in Figure 1.9. It is evident that the Kaiser-window design over-satisfies the requirements significantly.*

Minimum-phase designs with fixed transition width and peak passband/stopband ripple

The same procedure to design minimum-phase filters with fixed filter order and fixed transition width can be used to design minimum-phase filters with fixed transition width and peak passband/stopband ripple. In this case, rather than obtaining smaller ripples, the benefit is meeting the same transition width and peak passband/stopband ripples with a reduced filter order.

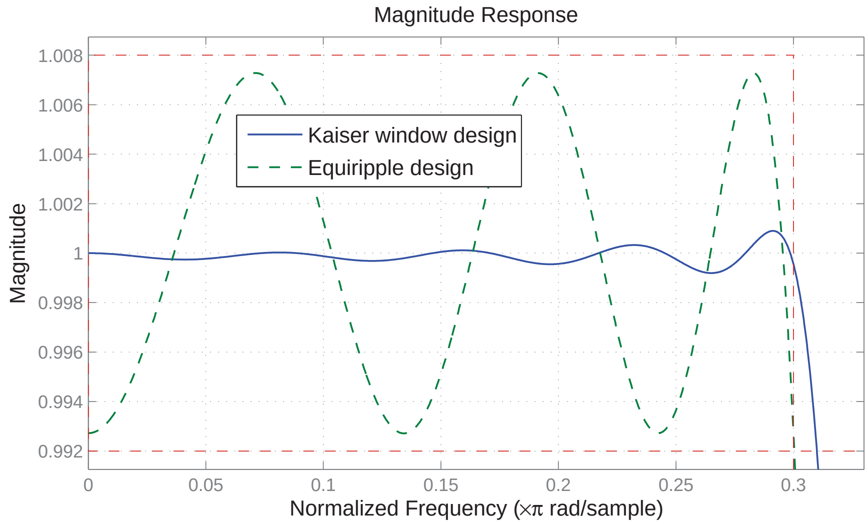

Figure 1.9: *Passband ripple details for both the Kaiser-window-designed FIR filter and the equiripple-designed FIR filter. The Kaiser-window design over-satisfies the requirement at the expense of increase number of taps.*

Example 8 *Consider the following specifications:*

Specifications Set 3

1. *Cutoff frequency:* 0.13π *rad/sample*
2. *Transition width:* 0.02π *rad/sample*
3. *Maximum passband ripple: 0.175 dB*
4. *Minimum stopband attenuation: 60 dB*

The minimum order needed to meet such specifications with a linear-phase FIR filter is 262. This filter must be the result of an optimal equiripple design. If we relax the linear-phase constraint however, the `equiripple` *design (based on the algorithm proposed in [8]) results in a minimum-phase FIR filter of 216th order that meets the specifications:*

```
Hf   = fdesign.lowpass('Fp,Fst,Ap,Ast',.12,.14,.175,60);
Hmin = design(Hf,'equiripple','minphase',true);
cost(Hmin)
```

Note that for these designs minimum-phase filters will have a much smaller transient delay not only because of their minimum-phase property but also because their filter order is lower than that of a comparable linear-phase filter. In fact this is true in general we had previously seen that a filter with the same transition width and filter order with minimum-phase has a smaller delay than a corresponding linear-phase filter. Moreover, the minimum-phase filter has smaller ripples so that the two filters are not really comparable. In order to compare apples to apples, the order of the linear-phase filter would have to increased until the ripples are the same as those of the minimum-phase design. This increase in filter order would of course further increase the delay of the filter.

1.3.3 Optimal equiripple designs with fixed peak ripple and filter order

So far we have illustrated equiripple designs with fixed transition width and fixed order and designs with fixed transition width and fixed peak ripple values. The Filter Design Toolbox also provides algorithms for designs with fixed peak ripple values and fixed filter order [10]. This gives maximum flexibility in utilizing the degrees of freedom available to design an FIR filter.

We have seen that, when compared to Kaiser-window designs, fixing the transition width and filter order results in an optimal equiripple design with smaller peak ripple values, while fixing the transition width and peak ripple values results in a filter with less number of taps. Naturally, fixing the filter order and the peak ripple values should result in a smaller transition width.

Example 9 *Consider the following design of an equiripple with the same cutoff frequency as in Example 7. The filter order is set to be the same as that needed for a Kaiser-window design to meet the ripple specifications*

```
Hf  = fdesign.lowpass('N,Fc,Ap,Ast',49,0.375,0.13,60);
Heq = design(Hf,'equiripple');
```

The comparison of this new design with the Kaiser-window design is shown in Figure 1.10. The transition width has been reduced from 0.15π *to approximately* 0.11π.

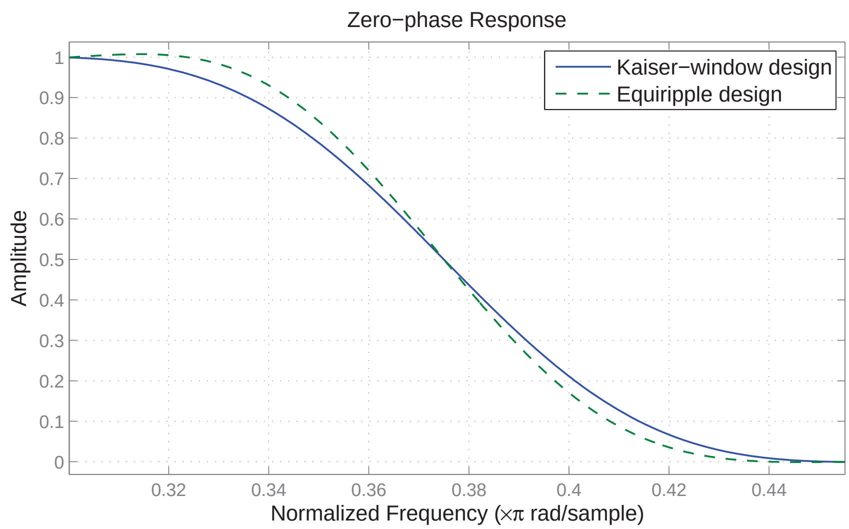

Figure 1.10: *Comparison of a Kaiser-window-designed FIR filter and an optimal equiripple FIR filter of the same order and peak ripple values. The equiripple design results in a reduced transition-width.*

Minimum-phase designs with fixed peak ripple and filter order

Once again, if linear-phase is not a requirement, a minimum-phase filter can be designed that is a superior in some sense to a comparable linear-phase filter. In this case, for the same filter order and peak ripple value, a minimum-phase design results in a smaller transition width than a linear-phase design.

Example 10 *Compared to the 50th order linear-phase design* `Heq`*, the following design has a noticeably smaller transition width:*

```
Hmin = design(Hf,'equiripple','minphase',true);
```

1.3.4 Constrained-band equiripple designs

Sometimes when designing lowpass filters (for instance for decimation purposes) it is necessary to guarantee that the stopband of the filter begins at a specific frequency value and that the filter provides a given minimum stopband attenuation.

If the filter order is fixed - for instance when using specialized hardware - there are two alternatives available in the Filter Design Toolbox for optimal equiripple designs. One possibility is to fix the transition width, the other is to fix the passband ripple.

Example 11 *For example, the design specifications of Example 7 call for a stopband that extends from* 0.45π *to* π *and provide a minimum stopband attenuation of 60 dB. Instead of a minimum-order design, suppose the filter order available is 40 (41 taps). One way to design this filter is to provide the same maximum passband ripple of 0.13 dB but to give up control of the transition width. The result will be a filter with the smallest possible transition width for any linear-phase FIR filter of that order that meets the given specifications.*

```
Hf  = fdesign.lowpass('N,Fst,Ap,Ast',40,.45,0.13,60);
Heq = design(Hf,'equiripple');
```

If instead we want to fix the transition width but not constrain the passband ripple, an equiripple design will result in a filter with the smallest possible passband ripple for any linear-phase FIR filter of that order that meets the given specifications.

```
Hf  = fdesign.lowpass('N,Fp,Fst,Ast',40,.3,.45,60);
Heq = design(Hf,'equiripple');
```

The passband details of the two filters are shown in Figure 1.11. Note that both filters meet the Specifications Set 2 because the order used (40) is larger than the minimum order required (37) by an equiripple linear phase filter to meet such specifications. The filters differ in how they "use" the extra number of taps to better approximate the ideal lowpass filter.

1.3.5 Sloped equiripple filters

In many cases it is desirable to minimize the energy in the stopband of a signal being filtered. One common case is in the design of decimation filters. In this case, the energy in the stopband of a signal after being filtered aliases into the passband region. To minimize the amount of aliasing, we want to minimize the stopband energy. Least-squares filters can be used for this, however, the drawback is that the passband presents larger fluctuations than may be desirable. An alternative is to design optimal

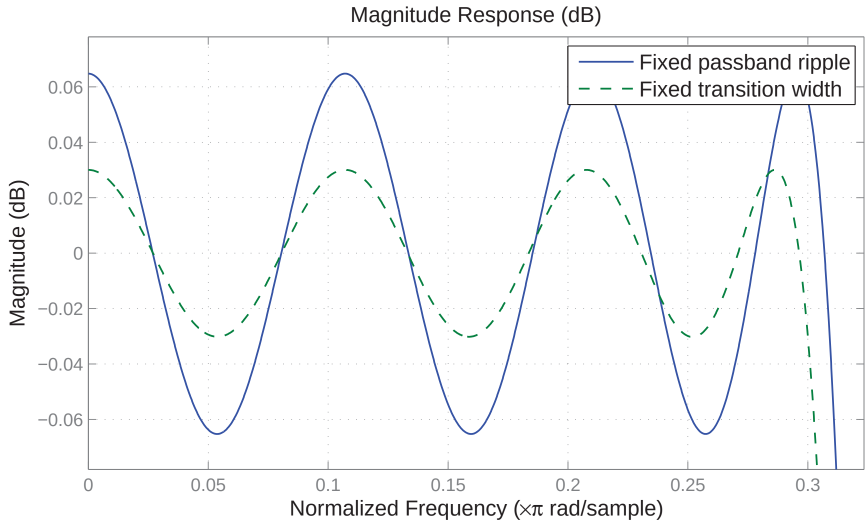

Figure 1.11: *Comparison of a two optimal equiripple FIR filters of 40th order. Both filters have the same stopband-edge frequency and minimum stopband attenuation. One is optimized to minimize the transition width while the other is optimized to minimize the passband ripple.*

equiripple filters but allowing for a slope in the stopband of the filter. The passband remains equiripple thus minimizing the distortion of the input signal in that region.

There are many ways of shaping the slope of the stopband. One way [11] is to allow the stopband to decay as $(1/f)^k$, that is as a power of the inverse of frequency. This corresponds to a decay of $6k$ dB per octave. Another way of shaping the slope is to allow it to decay in logarithmic fashion so that the decay appears linear in dB scale.

Of course there is a price to pay for the sloped stopband. Since the design provides smaller stopband energy than a regular equiripple design, the passband ripple, although equiripple, is larger than that of a regular equiripple design. Also, the minimum stopband attenuation measured in dB is smaller than that of a regular equiripple design.

Example 12 *Consider an equiripple design similar to that of Example 2 but with a stopband decaying as* $(1/f)^2$*:*

```
Fp  = 0.4 - 0.06/2; Fst = 0.4 + 0.06/2;
```

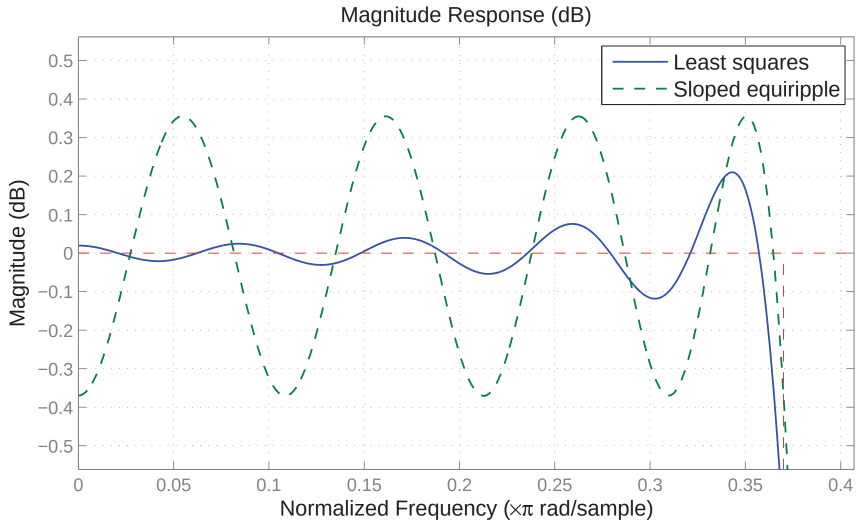

Figure 1.12: *Passband details of a sloped optimal equiripple FIR design and an optimal least-squares FIR design. The equiripple filter has a smaller peak error or smaller transition width depending on the interpretation.*

```
Hf  = fdesign.lowpass('N,Fp,Fst',42,Fp,Fst);
Hsloped = design(Hf,'equiripple','StopbandShape','1/f',...
   'StopbandDecay',2);
```

results in a stopband energy of approximately 8.4095e-05, not much larger that the least-squares design (6.6213e-005), while having a smaller transition width (or peak passband ripple - depending on the interpretation). The passband details of both the least-squares design and the sloped equiripple design are shown in Figure 1.12 (in dB). The stopband details are shown in Figure 1.13 (also in dB).

If we constrain the filter order, the passband ripple, and the minimum stopband attenuation, it is easy to see the trade-off between a steeper slope and the minimum stopband attenuation that can be achieved. Something has to give and since everything else is constrained, the transition width increases as the slope increases as well.

Example 13 *Design two filters with the same filter order, same passband ripple, and same stopband attenuation. The slope of the stopband decay is zero for the first filter and 40 for the second.*

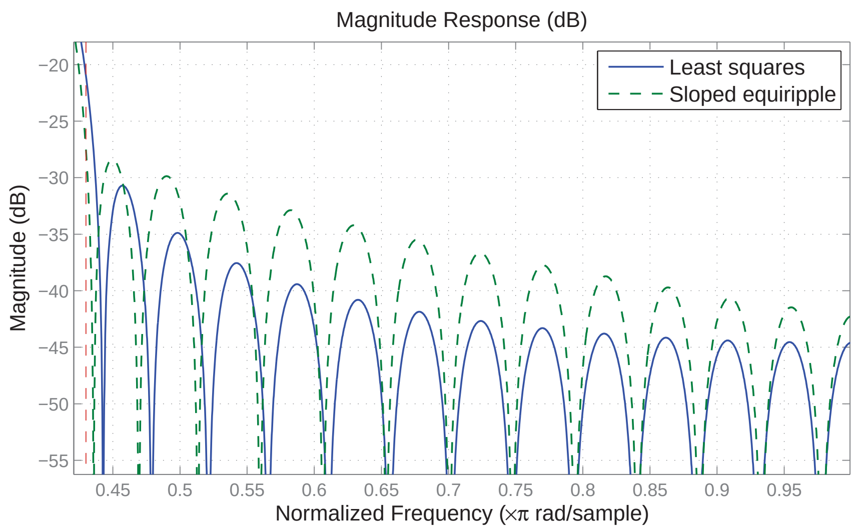

Figure 1.13: *Stopband details of a sloped optimal equiripple FIR design and an optimal least-squares FIR design. The overall error of the equiripple filter approaches that of the least-squares design.*

```
Hf   = fdesign.lowpass('N,Fst,Ap,Ast',30,.3,0.4,40);
Heq  = design(Hf,'equiripple','StopbandShape','linear',...
   'StopbandDecay',0);
Heq2 = design(Hf,'equiripple','StopbandShape','linear',...
   'StopbandDecay',40);
```

The second filter provides better total attenuation throughout the stopband. Since everything else is constrained, the transition width is larger as a consequence. This is easy to see with `fvtool(Heq,Heq2)`.

It is also possible to design minimum-phase sloped equiripple filters. These designs possess similar advantages over linear-phase designs as those described for other equiripple designs when linearity of phase is not a design requirement.

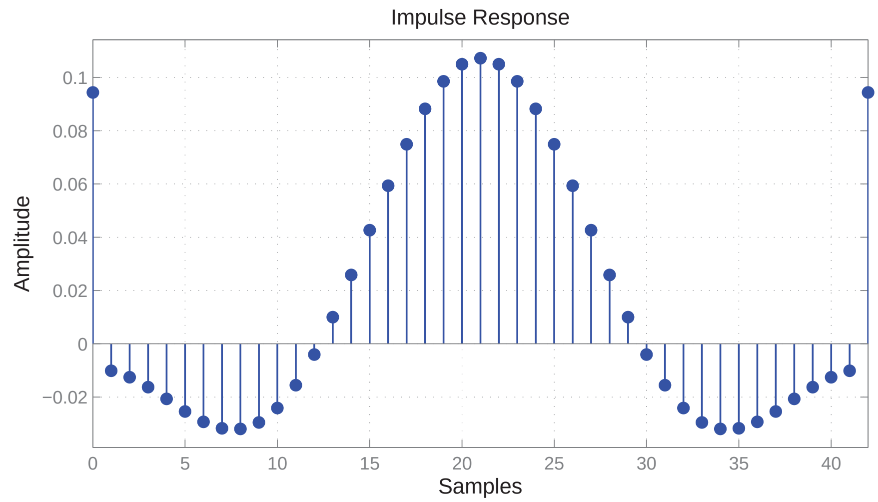

Figure 1.14: *Impulse response of equiripple filter showing anomalous end-points. These end points are the result of an equiripple response.*

1.4 Further notes on equiripple designs

1.4.1 Unusual end-points in the impulse response

In some cases the impulse response of an equiripple design appears to be anomalous. In particular, the end points seem to be incorrect. However, these end points are in fact a consequence of an equiripple stopband. If we remove the anomaly by say repeating the value in the sample closest to the end point, the stopband is no longer equiripple.

Example 14 *Consider the design of this lowpass filter with band edges that are quite close to DC:*

```
Hf  = fdesign.lowpass('N,Fp,Fst',42,0.1,0.12);
Heq = design(Hf,'equiripple');
fvtool(Heq,'Analysis','impulse')
```

The impulse response is show in Figure 1.14. Notice that the two end points seem completely out of place.

If the anomalous end-points are a problem*, it is not always feasible to modify them by replacing their value with that of their nearest neighbor because sometimes it is more than just the two end points that are anomalous. Moreover, simply using the nearest neighbor removes the equiripple property of the stopband in an uncontrolled way. It is preferable to use a sloped equiripple design to control the shape of the stopband and at the same time remove the anomalies.

Example 15 *Compare the magnitude response and the impulse response of these two designs using* `fvtool`*.*

```
Hf   = fdesign.lowpass('N,Fp,Fst',800,.064,.066);
Heq  = design(Hf,'equiripple','Wpass',1 ,'Wstop',10);
Heq2 = design(Hf,'equiripple','Wpass',1 ,'Wstop',10,...
  'StopbandShape','linear','StopbandDecay',80);
fvtool(Heq,Heq2)
```

A look at the impulse response reveals that adding a sloped stopband has altered several of the end points of the impulse response removing the discontinuity.

1.4.2 Transition region anomalies

In equiripple design, the transition band between passband and stopband or vice versa is treated as a don't care region for optimization purposes. That is, the frequency response is not optimized between bands with the hope that the response will smoothly transition between them. This is the case for lowpass and highpass filters, but may not be the case for filters with more than two bands when the width of the transition bands differ [4].

A possible solution is to reduce the larger transition bands to make them all the same width. This of course incurs in some degradation in the performance of the filter that will depend on what constraints are imposed. If this is not feasible, another solution would be to include the problematic transition band in the optimization by means of a arbitrary magnitude design.

* For instance, when the impulse response is very long, sometimes only a subset of it is stored in memory and the rest of the values are computed by curve fitting or some other interpolation scheme. In such cases the anomalous points pose a problem because the curve fitting is unlikely to reproduce them accurately.

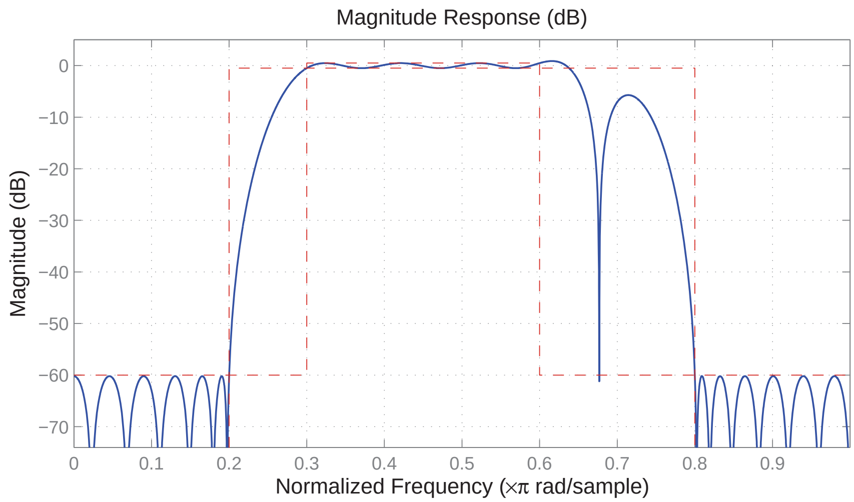

Figure 1.15: *Bandpass filter with transition-band anomaly due to having different transition-band widths.*

Example 16 *Consider the design of a bandpass filter with a first transition band* 0.1π *rad/sample wide and a second transition band* 0.2π *rad/sample wide:*

```
Hf  = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',...
   .2,.3,.6,.8,60,1,60);
Heq = design(Hf,'equiripple');
```

The magnitude response is shown in Figure 1.15. The anomalies in the second "don't-care" band are obvious. The design can be fixed by making the second transition band 0.1π *rad/sample wide. Since this is a minimum-order design, the price to pay for this is an increase in the filter order required to meet the modified specifications:*

```
Hf2  = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',...
   .2,.3,.6,.7,60,1,60);
Heq2 = design(Hf,'equiripple');
cost(Heq)
cost(Heq2)
```

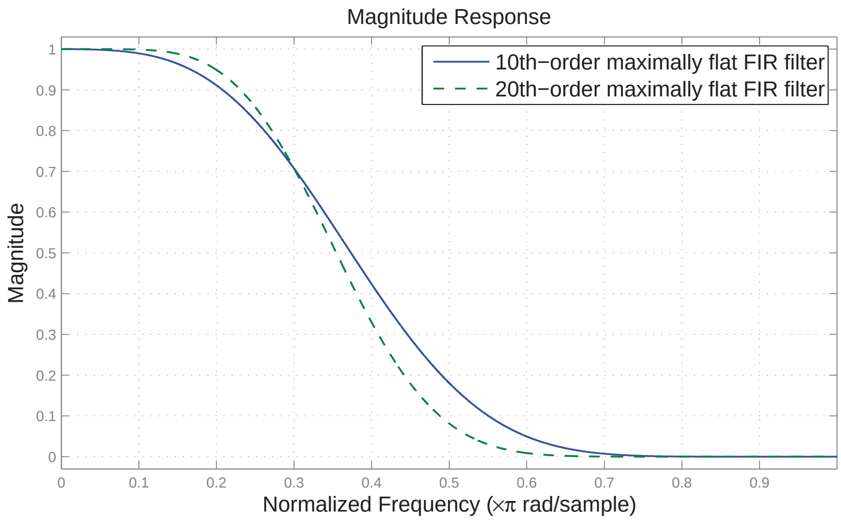

Figure 1.16: *Maximally-flat FIR filters. The smaller transition width in one of the filters is achieved by increasing the filter order.*

1.5 Maximally-flat FIR filters

Maximally-flat filters are filters in which both the passband and stopband are as flat as possible given a specific filter order.

As we should come to expect, the flatness in the bands comes at the expense of a large transition band (which will also be maximum). There is one less degree of freedom with these filters than with those we have look at so far. The only way to decrease the transition band is to increase the filter order.

Example 17 *Figure 1.16 shows two maximally-flat FIR filters. Both filters have a cutoff frequency of* 0.3π*. The filter with smaller transition width has twice the filter order as the other.*

The Maximally-flat stopband of the filter means that its stopband attenuation is very large. However, this comes at the price of a very large transition width. These filters are seldom used in practice (in particular with fixed-point implementations) because when the filter's coefficients are quantized to a finite number of bits, the large stopband cannot be

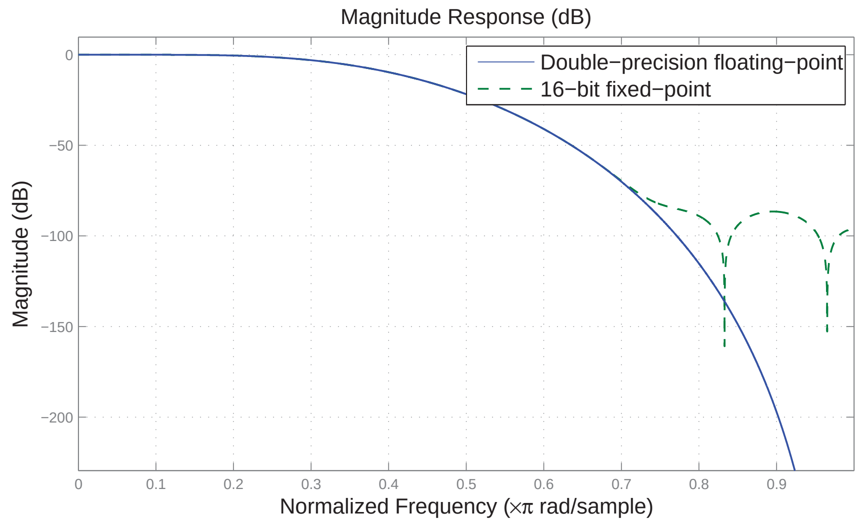

Figure 1.17: *A maximally-flat FIR filter implemented with double-precision floating-point arithmetic and the same filter implemented with 16-bit fixed-point coefficients.*

achieved (and often is not required anyway) but the large transition band is still a liability.

Example 18 *Compare the stopband attenuation of a maximally-flat FIR filter implemented using double-precision floating-point arithmetic with that of the same filter implemented with 16-bit fixed-point coefficients. The comparison of the two implementations is shown in Figure 1.17. The fixed-point implementation starts to differ significantly from the floating-point implementation at about 75-80 dB. Nevertheless, both filters have the same large transition width.*

The maximally-flat passband may be desirable because it causes minimal distortion of the signal to be filtered in the frequency band that we wish to conserve after filtering. So it may seem that a maximally-flat passband and an equiripple or perhaps sloped stopband could be a thought-after combination. However, if a small amount of ripple is allowed in the passband it is always possible to get a smaller transition band and most applications can sustain a small amount of passband ripple.

Example 19 *We can approach a maximally-flat passband by making the passband ripple of an equiripple design progressively smaller. However, for the same*

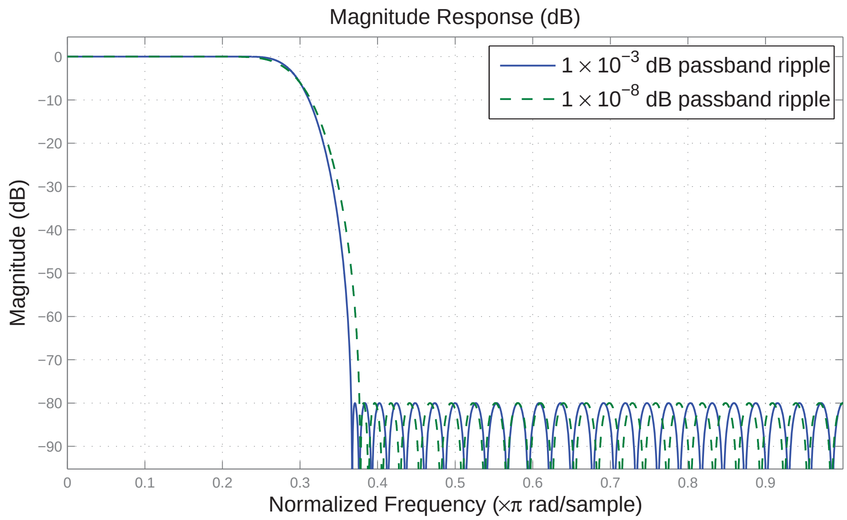

Figure 1.18: *Equiripple filters with passband approximating maximal flatness. The better the approximation, the larger the transition band.*

filter order and stopband attenuation, the transition width increases as a result. Consider the following two filter designs:

```
Hf   = fdesign.lowpass('N,Fc,Ap,Ast',70,.3,1e-3,80);
Heq  = design(Hf,'equiripple');
Hf2  = fdesign.lowpass('N,Fc,Ap,Ast',70,.3,1e-8,80);
Heq2 = design(Hf2,'equiripple');
```

The two filters are shown in Figure 1.18. It is generally best to allow some passband ripple as long as the application at hand supports it given that a smaller transition band results. The passband details are shown in Figure 1.19.

1.6 Summary and look ahead

Understanding FIR filter design is a matter of understanding the trade offs involved and the degrees of freedom available. A drawback of FIR filters is that the tend to require a large filter order and therefore a high computational cost to achieve the specifications desired. There are many ways of addressing this. One is to use IIR filters. Another is to use multistage

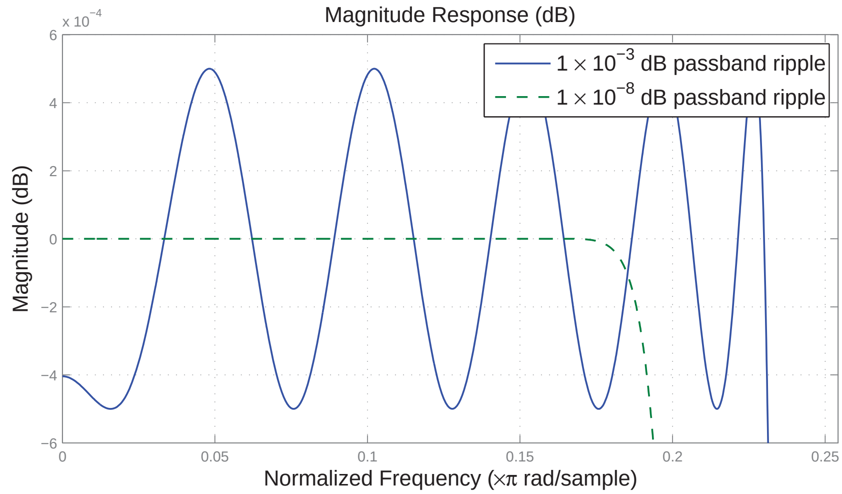

Figure 1.19: *Passband details of equiripple filters with very small passband ripple. The flatter passband is obtained at the expense of a larger transition band, i.e. a smaller usable passband.*

and/or multirate techniques that use various FIR filters connected in cascade (in series) in such a way that each filter shares part of the filtering duties while having reduced complexity when compared to a single-stage design. The idea is that for certain specifications to combined complexity of the filters used in multistage design is lower than the complexity of a comparable single-stage design.

We will be looking at all these approaches in the following chapters. We will then look into implementation of filters and discuss issues that arise when implementing a filter using fixed-point arithmetic.

Chapter 2

Basic IIR Filter Design

Overview

One of the drawbacks of FIR filters is that they require a large filter order to meet some design specifications. If the ripples are kept constant, the filter order grows inversely proportional to the transition width. By using feedback, it is possible to meet a set of design specifications with a far smaller filter order than a comparable FIR filter*. This is the idea behind IIR filter design. The feedback in the filter has the effect that an impulse applied to the filter results in a response that never decays to zero, hence the term infinite impulse response (IIR).

We will start this chapter by discussing classical IIR filter design. The design steps consist of designing the IIR filter in the analog-time domain where closed-form solutions are well known and then using the bilinear transformation to convert the design to the digital domain. We will see that the degrees of freedom available are directly liked to the design algorithm chosen. Butterworth filters provide very little control over the resulting design since it is basically a maximally-flat design. Chebyshev designs increase the degrees of freedom by allowing ripples in the passband (type I Chebyshev) or the stopband (type II Chebyshev). Elliptic filters allow for maximum degrees of freedom by allowing for ripples in both the passband and the stopband. In fact, Chebyshev and Butterworth designs can be seen as a special case of elliptic designs, so usually one should only concentrate on elliptic filter design, decreasing the passband

* However, see Chapter 5.