DESIGN OF IIR FILTERS

Samir V. Ginde & Joseph A.N. Noronha

DSP and Filter Design
(ECE 4624)

“I have neither given nor received unauthorized aid on this project”

__________________________  _______________________
Samir V. Ginde                              Joseph A.N. Noronha

Bradley Department of Electrical & Computer Engineering
Virginia Polytechnic and State University
Blacksburg, VA
U.S.A
# CONTENTS

1. Introduction .......................................................... 1

2. Analog to Digital Domain Mapping Techniques .......... 2
   2.1 Impulse Invariance ........................................... 2
   2.2 Bilinear Transformation ..................................... 2

3. Filter Types .......................................................... 5
   3.1 Butterworth Filters .......................................... 5
   3.2 Chebyshev Filters ............................................ 6
   3.3 Elliptic Filters .............................................. 7

4. Design ................................................................. 9
   4.1 Section I ......................................................... 10
      4.1.1 Butterworth Filter ...................................... 11
      4.1.2 Chebyshev Type I Filter ............................... 14
      4.1.3 Chebyshev Type II Filter .............................. 17
      4.1.4 Elliptic Filter .......................................... 20
      4.1.5 FIR Filter .............................................. 23
         4.1.5.1 Kaiser Window .................................... 23
         4.1.5.2 Optimum Equiripple Filter (Parks McClellan Algorithm) ........................................ 26
      4.1.6 Comparison of the Designed Filters .................. 29
   4.2 Section II ........................................................ 30
      4.2.1 Part I ...................................................... 30
         4.2.1.1 Direct Design of Elliptic High-Pass Filter .......... 31
         4.2.1.2 Design of Elliptic High Pass Filter by Transformation from LPF 35
      4.2.2 Part II ...................................................... 35
         4.2.2.1 Direct Design of Chebyshev Type I Band Pass filter ........ 36
         4.2.2.2 Design of Chebyshev I Band Pass Filter by Transformation from LPF ........ 38

5. Conclusion ............................................................ 41

References ............................................................ 41

Appendix 1  MATLAB Code ........................................... 42
Appendix 2  MATLAB Functions ..................................... 52
1. Introduction

The Digital Filter Design problem involves the determination of a set of filter coefficients to meet a set of design specifications. These specifications typically consist of the width of the passband and the corresponding gain, the width of the stopband(s) and the attenuation therein; the band edge frequencies (which give an indication of the transition band) and the peak ripple tolerable in the passband and stopband(s).

Two types of digital filters exist – the IIR (Infinite Impulse Response) and the FIR (Finite Impulse Response). FIR filters were studied in an earlier project. They possess certain properties, which make them the preferred design choices in numerous situations over IIR filters. Most notably, FIR filters (all zero system function) are always stable, with a realization existing for each FIR filter. Another feature exclusive to FIR filters is that of a linear phase response.

The content of this project is the design of IIR filters using MATLAB. The design of IIR filters proceeds through a vastly different set of steps than those followed by FIR filter design algorithms. The design of IIR filters is closely related to the design of analog filters, which is a widely studied topic. An analog filter is usually designed and a transformation is carried out into the digital domain. Two transformations exist – the impulse invariant transformation and the bilinear transformation. In this project, the focus is on designing minimum order IIR filters to meet a set of specifications using MATLAB functions. The bilinear transformation has been used.

The designed IIR filters are characterized by significantly lower order than the corresponding FIR filters. This comparison is brought about in Section I of the design where FIR filters are designed along with the IIR filters. There it is shown that the best IIR filter is less complex than the optimum FIR filter. The price to pay is the non-linear phase of the IIR filters, which is unavoidable.

The rest of this report is organized as follows. The next section describes the IIR digital filter design process. Then the design is presented in two sections. In the first, a low pass filter is designed for four IIR and two FIR realizations and various comparisons are drawn from the results. The second section deals with the design of a high pass Elliptic filter and a band pass Chebyshev Type I filter. Each design is accompanied by a plot of its frequency response, impulse response and pole-zero diagram. The MATLAB code is presented as appendix with an accompanying appendix with a glossary of the key functions for filter design.
2. Analog to Digital Domain Mapping Techniques

Digital Filters are designed by using the values of both the past outputs and the present input, an operation brought about by convolution. If such a filter is subjected to an impulse then its output need not necessarily become zero. The impulse response of such a filter can be infinite in duration. Such a filter is called an *Infinite Impulse Response* filter or **IIR** filter. The infinite impulse response of such a filter implies the ability of the filter to have an infinite impulse response. This indicates that the system is prone to feedback and instability.

The report studies several different types of IIR filters including the Butterworth Filter, Chebyshev I & II Filters and Elliptic Low, High and Bandpass filters. IIR filters are designed essentially by the Impulse Invariance or the Bilinear Transformation method.

2.1 Impulse Invariance

This procedure involves choosing the response of the digital filter as an equi-spaced sampled version of the analog filter.

1. Decide upon the desired frequency response  
2. Design an appropriate analogue filter  
3. Calculate the impulse response of this analogue filter  
4. Sample the analogue filter's impulse response  
5. Use the result as the filter coefficients

The impulse invariance method maps the left hand portion of the s-plane into the interior of the unit circle and the right hand portion of the s-plane to the exterior of the unit circle; hence each horizontal strip in the s-plane is overlayed onto the z-plane to form the digital system function from the analog system function. Since any practical analog filter can never be band-limited interference is a major consideration. Due to the aliasing that arises in the sampling process the digital frequency response is distinct from the analog filter frequency response. Hence distortion in the frequency response is one of the major limiting factors of this implementation while its advantage lies in the fact that there is a linear relationship between the analog and digital frequency response. Hence in order to prevent sever distortion due to the band limiting this method is restricted to the design of Low and Bandpass Filters.

2.2 Bilinear Transformation

The Bilinear Transformation method overcomes the effect of aliasing that is caused to due the analog frequency response containing components at or beyond the *Nyquist* Frequency. The bilinear transform is a method of compressing the infinite, straight analogue frequency axis to a finite one long enough to wrap around the unit circle once only. This is also sometimes called frequency warping. This introduces a distortion in the frequency. This is undone by pre-warping the critical frequencies of the analog filter (cutoff frequency, center frequency) such that when the analog filter is transformed into the digital filter, the designed digital filter will meet the desired specifications.
Bilinear Transformation:

Consider an analog filter

\[ H(s) = \frac{b}{s + a} \]

This system can be characterized by a differential equation

\[ \frac{d}{dt}y(t) + ay(t) = bx(t) \]

Suppose we approximate the integral rather than the derivative

\[ y(t) = \int_{t_0}^{t} y\tau d\tau + y(t_0) \]

We can approximate the integral by using the Trapezoidal formula

\[ y(nT) = \frac{T}{2}[y'(nT) + y'(nT - T)] + y(nT - T) \]

From the differential equation we can substitute for y(t)

\[ y(nT) = -ay(nT) + bx(nT) \]

We can substitute this in the trapezoidal rule and write

\[ \left(1 + \frac{aT}{2}\right)y(n) - \left(1 - \frac{aT}{2}\right)y(n - 1) = \frac{bT}{2}[x(n) + x(n - 1)] \]

The z-transform of this gives:

\[ \left(1 + \frac{aT}{2}\right)Y(z) - \left(1 - \frac{aT}{2}\right)z^{-1}Y(z) = \frac{bT}{2}(1 + z^{-1})X(z) \]

Which is simplified to

\[ H(z) = \frac{Y(z)}{X(z)} = \frac{b}{\frac{2}{T}\frac{1}{1 + z^{-1}}} + a \]

Clearly the mapping is as follows

\[ H(z) = H(s) \bigg|_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} \]

This mapping is known as the bilinear transformation

where \( u \) is the derivative of \( y \).

By solving this equation for \( y \) we obtain

\[ s \leftrightarrow \frac{2}{T}\left(\frac{z-1}{z+1}\right) \]

This transformation is known as the Bilinear or Tustin Transformation. The Laplace transforms in the filter expressions are replaced by the corresponding z-transforms.

Replacing \( s = \sigma + j\Omega \) and performing algebraic manipulations, substituting \( z = e^{j\omega} \) we get
\[ \omega = 2 \tan^{-1}(\Omega T/2) \]

It can be seen that analog dc \((s = 0)\) maps to digital dc \((z = 1)\) and the highest analog frequency \((s = \infty)\) maps to the highest digital frequency \((z = -1)\). It is easy to show that the entire \(jw\) axis in the \(s\) plane is mapped exactly once around the unit circle in the \(z\) plane. Therefore, it does not alias. With \((2/T)\) real and positive, the left-half \(s\) plane maps to the interior of the unit circle, and the right-half \(s\) plane maps outside the unit circle.

The constant provides one remaining degree of freedom that can be used to map any particular finite frequency the \(jw\) axis in the \(s\) plane to a particular desired location on the unit circle \(e^{j\omega}\) in the \(z\) plane. All other frequencies will be warped. In particular, approaching half the sampling rate, the frequency axis compresses more and more. Filters having a single transition frequency, such as lowpass or highpass filters, map beautifully under the bilinear transform; you simply map the cut-off frequency where it belongs, and the response looks great. In particular, ``equal ripple'' is preserved for optimal filters of the elliptic and Chebyshev types because the values taken on by the frequency response are identical in both cases; only the frequency axis is warped.
3. Filter Types

3.1 Butterworth Filters

Butterworth filters are causal in nature and of various orders, the lowest order being the best (shortest) in the time domain, and the higher orders being better in the frequency domain. Butterworth or maximally flat filters have a monotonic amplitude frequency response which is maximally flat at zero frequency response (Fig 3.1) and the amplitude frequency response decreases logarithmicly with increasing frequency. The butterworth filter has minimal phase shift over the filter’s band pass when compared to other conventional filters.

\[ \overline{B(\omega)B(\omega)} = \frac{1}{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}} \]

![Figure 3.1](image-url)
3.2 Chebyshev Filters

Chebyshev filters are of two types: Chebyshev I filters are all pole filters which are equi-ripple in the passband and are monotonic in the stopband. (Fig 3.2)

![Fig 3.2](image)

Chebyshev II filters contain both poles and zeros exhibiting a monotonic behavior in the passband and equi-ripple in the stopband. (Fig 3.3)

![Fig 3.3](image)

The frequency response of the filter is given by

$$|H(Ω)|^2 = \left(1 + \epsilon^2 T_N^2(Ω/Ω_p)\right)^{-1}$$

where $\epsilon$ is a parameter related to the ripple present in the passband

$$T_N = \begin{cases} \cos(N\cos^{-1}x) & |x| \leq 1 \\ \cos(N\cosh^{-1}x) & |x| \geq 1 \end{cases}$$
3.3 Elliptic Filters

Elliptic filters are characterized by equi-ripples in both their stop and their passbands. (Fig 3.4) They provide a realization with the lowest order for a particular set of conditions.

\[ |H(j\Omega)| = 10^{-R_{p/20}} \Omega = 1 \]

**Fig. 3.4**

**Frequency transformations:**

This is one of the more common techniques employed in the design of filters. A low pass analog or digital filter may be designed first and then transformed into digital high or bandpass filters.

**Analog frequency transformations:**

The frequency transformations that can be used to obtain a high pass, low pass, bandpass or band reject filter are indicated below in Table 1.

Here \( \Omega_0^2 = \Omega_1 \Omega_2 \) which is defined as the cutoff frequency for a low or highpass and the center frequency for the bandpass and band reject filter.

\[ Q = \frac{\Omega_0}{\Omega_2 - \Omega_1} \]

where \( \Omega_2 \) and \( \Omega_1 \) are the upper and lower cutoff frequencies respectively. \( \Omega_2 - \Omega_1 \) denotes the bandwidth.
<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Pass</td>
<td>$S = \frac{s}{\Omega_c}$</td>
</tr>
<tr>
<td>High Pass</td>
<td>$S = \frac{\Omega_c}{s}$</td>
</tr>
<tr>
<td>Bandpass</td>
<td>$S = Q\frac{(s^2 + \Omega_0^2)}{\Omega_0 s}$</td>
</tr>
</tbody>
</table>

Table 1
4. Design

*The Signal Processing Toolbox* in MATLAB includes several useful functions for designing both FIR and IIR digital filters as well as traditional analog filters. The MATLAB functions used are listed in the appendix. The design techniques for the digital filters were briefly described in the previous section. The basic characteristics of the common analog filter types were also summarized. The filters considered in this project are the Butterworth, Chebyshev (Type I and Type II), and Elliptic. (In addition, Bessel filters were also studied).

Three design problems are now considered. The first, a low pass IIR filter, is the subject of section I. As mentioned earlier, IIR filter design proceeds by conversion of digital specifications to analog specifications and an analog filter is then designed prior to conversion to the required digital filter using the appropriate frequency transformation. In this problem all the analog filter types are considered, making the problem a useful example for comparison of different design options. Then the same specifications are applied to the design of an FIR filter. This too involves the window method using, first a Kaiser Window, then the Park-McClellan algorithm. Ultimately, this design problem allows a close comparison of all filter design options.

In section II, two filters are designed. The first is a fifth-order high-pass elliptic filter and the second, a sixth order bandpass Chebyshev Type I filter. Each of these design problems is approached in two ways. The first approach uses the prescribed MATLAB command to design the filter. The second first involves the construction of a low-pass analog prototype, followed by a transformation to the respective high-pass or band-pass filter. The last step is the application of the bilinear transformation to the converted analog filter, which results in the desired digital filter.

All the examples are accompanied by the corresponding frequency response characteristics (both magnitude and phase plots), with close-ups given wherever required, impulse response diagrams (truncated to a significant number of terms – in our examples 60 when infinite), and pole-zero diagrams. **All the magnitude response plots are in dB and phase response plots in radians. The frequency axis is in terms of normalized frequency.**
4.1 SECTION I

This section involves the design of a low-pass filter with specifications as follows:

- Sampling Frequency, \( F_s = 10 \text{ Hz} \)
- Passband Edge Frequency, \( F_{c1} = 2 \text{ Hz} \)
- Stopband Edge Frequency, \( F_{c2} = 2.4 \text{ Hz} \)
- Passband Ripple 0.1 dB
- Minimum Stopband Attenuation 30dB

The following IIR filters were designed to meet these specifications:

- Butterworth
- Chebyshev type I
- Chebyshev type II
- Elliptic

Then the design of an FIR filter was considered. The following FIR filters were designed:

- Using the Kaiser Window
- Using the Park-McClellan algorithm

The results of each design are specified and then they are compared.
4.1.1 Butterworth Filter

Order = 21

Frequency Response of the Butterworth Filter

Fig. 4.1.1.1 Frequency response of Butterworth filter

Fig. 4.1.1.2 Passband of Butterworth filter
The following observations can be made about the frequency response of the Butterworth filter (Figs. 4.1.1.1-3):

1. The designed filter meets the specifications. (Fig 4.1.1.2 and Fig 4.1.1.3)

2. The magnitude response is flat in the passband. This is called a maximally flat response.

3. The magnitude response is monotonic in the stopband.

4. The phase response is approximately linear in the passband.
Impulse Response and Pole Zero Diagram of the Butterworth Filter

From the impulse response (Fig 4.1.1.4), which decays with time, we can infer that the designed filter is stable. This is affirmed by the pole-zero diagram (Fig 4.1.1.5), in which all the poles reside within the unit circle. The pole zero diagram also contains 21 zeros corresponding to the roots of \((1+z^{-21})\) term that appears in the numerator of the transfer function on account of the bilinear transformation used by the `butter` function in MATLAB.
4.1.2 Chebyshev Type I Filter

Order = 9

Frequency Response of the Chebyshev Type I Filter

Fig. 4.1.2.1 Frequency response of Chebyshev I filter

Fig. 4.1.2.2 Passband ripple of Chebyshev I filter

Fig. 4.1.2.3 Stopband of Chebyshev I filter
The following observations can be made about the frequency response (Figs. 4.1.2.1-3) of the Chebyshev I filter:

1. The designed filter meets the specifications. (Fig 4.1.2.2 and Fig 4.1.2.3)

2. The passband is characterized by the presence of ripples, each having amplitude 0.1 dB as per the specifications. Such a response is referred to as an equiripple characteristic.

3. The stopband is monotonically decreasing.

4. The phase response is non-linear for the major part of the passband.

5. It can be noticed that the order of the Chebyshev filter, nine, is substantially less than that of the Butterworth filter considered earlier. This was of order 21.

Impulse Response and Pole Zero Diagram of the Chebyshev Type I Filter

From the impulse response (Fig 4.1.2.4), which decays with time, we can infer that the designed filter is stable. This is affirmed by the pole-zero diagram, in which all the poles reside within the unit circle. The pole zero diagram (Fig 4.1.2.5) also contains 9 zeros corresponding to the roots of \((1+z^6)\) term that appears in the numerator of the transfer function on account of the bilinear transformation which is used by the cheby1 function in MATLAB.
Fig. 4.1.2.5 Pole-Zero diagram of the Chebyshev I filter

Fig. 4.1.2.6 Close-up of the zeros
4.1.3 Chebyshev Type II Filter

Order = 9

Frequency Response of the Chebyshev Type II Filter

Fig. 4.1.3.1 Frequency response of Chebyshev II filter

Fig. 4.1.3.2 Passband edge of Chebyshev II filter

Fig. 4.1.3.3 Complete Passband of Chebyshev II Filter
The following observations can be made from the frequency response (Figs. 4.1.3.1-4) of the Chebyshev II filter:

1. The designed filter meets the specifications. (Fig. 4.1.3.2, Fig. 4.1.3.3, and Fig. 4.1.3.4)

2. The passband is flat.

3. The stopband is characterized by the presence of ripples, each having the same amplitude. This is the converse of what was observed in the case of the Chebyshev I filter.

4. Interestingly, the order of the filter is 9, just as was the case for the Chebyshev I filter.

5. The phase response is approximately linear for lower passband frequencies and has value less than zero, indicating lagging phase.
Impulse Response and Pole Zero Diagram of the Chebyshev Type II Filter

From the impulse response (Fig. 4.1.3.5), which decays with time, we can infer that the designed filter is stable. This is affirmed by the pole-zero diagram (Fig. 4.1.3.6), in which all the poles reside within the unit circle. The pole zero diagram also contains 9 zeros which reside on the unit circle.
4.1.4 Elliptic Filter

Order = 5

Frequency Response of the Elliptic Filter

Fig. 4.1.4.1 Frequency Response of Elliptic filter

Fig. 4.1.4.2 Passband of Elliptic filter
The following observations can be made from the frequency response (Figs. 4.1.4.1-3) of the Elliptic filter:

1. The designed filter meets the specifications. (Fig. 4.1.4.2 and Fig 4.1.4.3)

2. The frequency response is characterized by ripples in both the passband and the stopband. These ripples are of equal amplitude.

3. The order of the filter, five, is the lowest of all types considered thus far. This is because this filter enables the sharpest transition of all types considered for a fixed order. Alternatively, the order of the filter designed to meet a particular transition specification will be the least. This explains the presence of ripples in the passband and stopband. It is these that make the sharp transitions possible.

4. The phase response resembles that of the Chebyshev II filter in the passband only, exhibiting linearity for lower passband frequencies.
Impulse Response and Pole Zero Diagram of the Elliptic Filter

From the impulse response, which decays with time, we can infer that the designed filter is stable. This is affirmed by the pole-zero diagram, in which all the poles reside within the unit circle. The pole zero diagram also contains 5 zeros which reside on the unit circle. This observation is similar to that made in the case of the Chebyshev II filter.
4.1.5 FIR Filters

4.1.5.1 Kaiser Window

Order = 55

Frequency Response of the FIR Filter

Fig. 4.1.5.1.1 Frequency response of FIR filter (using Kaiser Window)

Fig. 4.1.5.1.2 Passband edge ripple
The subject of FIR filter design using windows was studied in depth in an earlier project. The following characteristics can be observed from the frequency response (Figs 4.1.5.1-3):

1. The designed filter meets the specifications. (Fig. 4.1.5.2 and Fig. 4.1.5.3)

2. The passband contains ripples of increasing magnitude with the peak ripple appearing at the band-edge.

3. The stopband also consists of ripples of decreasing amplitude.

4. The ripples appear on account of the ripples in the frequency response of the Kaiser window [1]

5. The order of the FIR filter is 55, which is considerably higher than those obtained for any of the designed IIR filters. This is a general result.

6. The phase response is linear. This is a highly desirable property.
The impulse response (Fig. 4.1.5.1.4) and pole-zero diagram Fig. (4.1.5.1.5) exhibit some familiar properties – the impulse response is symmetric about its midpoint. The filter has only zeros, and every zero is accompanied by the inverse of its conjugate. Consequently this filter will always be stable.
4.1.5.2 Optimum Equiripple FIR filter (Parks-McClellan Algorithm)

Order = 41

Frequency Response of the FIR Filter

Fig. 4.1.5.2.1 Frequency response of optimum FIR filter

Fig. 4.1.5.2.2 Passband ripples in response of optimum FIR filter
The frequency response (Fig. 4.1.5.2.1 – 4) of the optimum FIR filter displays the following characteristics:

1. The designed filter meets the specifications. (Fig. 4.1.5.2.2 – 4)

2. The stopband and the passband are characterized by ripples of equal amplitude.

3. The phase response is linear in the passband. This is highly desirable.

4. The order of the filter, 41 renders it superior to the filter obtained using the Kaiser window. Yet, it is nearly twice the order of the Butterworth filter.
Impulse Response and Pole Zero Diagram of the FIR Filter

As in the case of the filter designed using the Kaiser window, the impulse response (Fig. 4.1.5.2.5) and pole-zero diagram (Fig. 4.1.5.2.6) exhibit some familiar properties – the impulse response is symmetric about its midpoint. The filter has only zeros, and every zero is accompanied by the inverse of its conjugate. Consequently this filter will always be stable.
### 4.1.6 Comparison of the Designed Filters

The problem of digital low-pass filter designed was approached from all possible angles. This offers a means for comparison of the different options available to a designer. The results of the previous section are summarized below (Table 2).

<table>
<thead>
<tr>
<th>Type of Filter</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIR Butterworth</td>
<td>21</td>
</tr>
<tr>
<td>IIR Chebyshev type I</td>
<td>9</td>
</tr>
<tr>
<td>IIR Chebyshev type II</td>
<td>9</td>
</tr>
<tr>
<td>IIR Elliptic</td>
<td>5</td>
</tr>
<tr>
<td>FIR Kaiser Window</td>
<td>55</td>
</tr>
<tr>
<td>FIR Optimum Filter</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 2.

All the designed filters meet a common specification. The obvious comparison is that between FIR and IIR filters. The IIR filters were found to have the following advantages:

- They require a significantly lower order for the same specifications
- This means that their hardware requirements should be less. This advantage is not so emphatic in practice. **It is found that FIR filters having an order 4 times that of the corresponding IIR filter possess the same complexity** [3]. This arises by calculating the number of multipliers required to implement the design. The importance of realization is not emphasized enough in this project, since only design is considered. Several references [1,2,3] treat this topic in detail.

On the other hand the FIR filters possess the following desirable properties:

- They are always stable
- They can be designed to have linear phase
- In the case of the optimum filter the passband ripple can be controlled. For the Kaiser window this is not true. In fact, in the designed filter, a tighter passband tolerance (0.09 dB) had to be specified so that the filter met the requirements.

The ultimate choice depends on the type of application and budget. If budget were a constraint, the Elliptic filter, having minimum complexity would be the obvious choice. If linear phase were a necessity then an FIR filter would have to be the design choice.
4.2 SECTION II

4.2.1 Part I

The goal of this section is to design an IIR filter with the following characteristics:

Type: High Pass IIR Elliptic filter  
Order = 5  
Sampling Frequency, $F_s = 10$ Hz  
Passband Edge Frequency, $F_{c1} = 2.4$ Hz  
Stopband Edge Frequency, $F_{c2} = 2$ Hz  
Passband Ripple = 1 dB  
Minimum Stopband Attenuation = 40 dB

The design process was approached in two different ways:

The design using the `ellip` function available in MATLAB, called direct design, was first undertaken.

The design was then attempted indirectly by performing the following steps:

1. First a low pass elliptic analog prototype filter was designed with cut-off of 1 rad/s and order 5 to meet the required passband and stopband ripple specifications.

2. Then this was converted to an analog high-pass filter with the given passband edge frequency.

3. Then, using the bilinear transformation this analog high-pass filter was converted to the required the digital high-pass filter.
4.2.1.1 Direct Design of Elliptic High-Pass Filter

Order = 5

Frequency Response of Elliptic Filter

Fig. 4.2.1.1.1 Frequency response of Elliptic high pass filter

Fig. 4.2.1.1.2 Passband ripple of elliptic high pass filter

Fig. 4.2.1.1.3 Stopband ripple of elliptic high pass filter

The observations made from the frequency response (Fig. 4.2.1.1.1 – 3) of the Elliptic filter are:

1. The frequency response meets the specifications. (Fig. 4.2.1.1.2 and Fig. 4.2.1.1.3)

2. The frequency response is characterized by ripples in both the passband and the stopband. These ripples are of equal amplitude.

3. The phase response exhibits linearity for higher passband frequencies
Impulse Response and Pole Zero Diagram of Elliptic Filter

The impulse response and pole-zero diagram display properties similar to those displayed by the filters designed under Section I. Only the positions of the poles and zeros are interchanged. The zeros are now in the right-hand z-plane, reflecting the high pass behavior of this filter.

Fig. 4.2.1.1.4 Impulse response of Elliptic high-pass filter

Fig. 4.2.1.1.5 Pole-Zero diagram of Elliptic high pass filter
4.2.1.2 Design of Elliptic High Pass Filter by Transformation from LPF

Order = 5

Frequency Response of Elliptic Filter

The frequency response of the filter designed using the bilinear transformation bears a resemblance to that of the filter previously described. The function \texttt{bilinear} permits a frequency to be specified at which the two responses should match. This was taken to be the passband edge frequency. As a result the responses appear to be identical. The frequency response will not show
the effects of frequency warping produced by the bilinear transformation, which compresses the higher frequencies in the response of the analog filter when subjected to the bilinear transformation. (See 2.2)

Impulse Response and Pole Zero Diagram of Elliptic Filter

![Impulse response of elliptic high pass filter (bilinear)](image1)

Fig. 4.2.1.2.4 Impulse response of elliptic high pass filter (bilinear)

The pole-zero diagram and impulse response are identical to those obtained for the direct design.

![Pole-Zero diagram of Elliptic high pass filter (bilinear)](image2)

Fig. 4.2.1.2.5 Pole-Zero diagram of Elliptic high pass filter (bilinear)
4.2.2 Part II

The goal of this section is to design an IIR filter with the following characteristics:

Type: Band Pass IIR Chebyshev Type I filter  
Order = 6  
Sampling Frequency, $F_s = 10$ Hz  
Lower band Edge Frequency, $F_{c1} = 2$ Hz  
Upper band Edge Frequency, $F_{c2} = 4$ Hz  
Passband Ripple = 1 dB

The design process was approached in two different ways:

The design using the `cheby1` function available in MATLAB, called direct design, was first undertaken.

The design was then attempted indirectly by performing the following steps:

1. First a low pass Chebyshev I analog prototype filter was designed with cut-off of 1 rad/s and order 6 to meet the required passband ripple specification.

2. Then this was converted to an analog band-pass filter with the given specification of center frequency (geometric mean of band edges) and bandwidth (difference between edge frequencies).

3. Then, using the bilinear transformation this analog high-pass filter was converted to the required the digital band-pass filter.
4.2.2.1 Direct Design of Chebyshev Type I Band Pass filter

Order = 6

Frequency Response of Chebyshev I Filter

Fig. 4.2.2.1.1 Frequency response of Chebyshev I band pass filter

Fig. 4.2.2.1.2 Passband characteristics of Chebyshev I high pass filter
The passband (Fig. 4.2.2.1.2) of the designed filter is characterized by ripples of equal amplitude, while its stopbands (Fig. 4.2.2.1.1) are monotonic. This is as expected. The filter displays approximately linear phase through the mid-section of its passband (between frequencies of 0.225 and 0.375) with the non-linearity creeping in at the band edges.

**Impulse Response and Pole Zero Diagram of Chebyshev I Filter**

![Impulse Response of Chebyshev I Bandpass Filter](image.png)

Fig. 4.2.2.1.3 Impulse response of Chebyshev I band pass filter

The impulse response and pole-zero diagram of the designed filter show it to be stable. More will be said about these diagrams in the next section.

![Pole-Zero Diagram of Chebyshev I Bandpass Filter](image.png)

Fig. 4.2.2.1.4 Pole-Zero diagram of Chebyshev I band pass filter
4.2.2.2 Design of Chebyshev I Band Pass Filter by Transformation from LPF

**Order = 6**

**Frequency Response of Chebyshev I Filter**

![Chebyshev I BandPass Filter, Magnitude response, order=6](image1)

![Chebyshev I BandPass Filter, Phase response, order=6](image2)

Fig. 4.2.2.1 Frequency response of Chebyshev I band pass filter (Bilinear)

![Chebyshev I BandPass Filter, Magnitude response, order=6](image3)

Fig. 4.2.2.2 Passband characteristics of Chebyshev I band pass filter (Bilinear)
The frequency response (Fig. 4.2.2.2.1 – 3) of the final filter obtained following the bilinear transformation is drastically different from that obtained from the use of the cheby1 function, having bandwidth reduced to about 0.11. The two responses are quite similar for the low range of frequencies. However, for higher frequencies the effect of the warping, or frequency compression induced by the bilinear transformation becomes more apparent. The function bilinear, as mentioned earlier permits the specification of only one frequency at which the response of the resulting filter should match that of the transformed analog filter. This worked well in the case of the high pass filter designed in Part I of this section. Here, there are two frequencies at which the responses should match – the lower and upper passband edge frequencies. Obviously, the function bilinear does not permit this. Consequently we have to exercise one of the following three options:

- Match at lower band edge frequency
- Match at the center frequency
- Match at the upper band edge frequency

In exploring these possibilities it was noticed that matching at the center frequency resulted in the lower band edge of 0.2 being around 15 dB below the passband gain and the upper band edge frequency being nearly 40 dB below. Attempting to match the responses at the upper band edge frequency results in the lower band edge being nearly 40 dB below. The upper band edge frequency was found to be attenuated by the same amount, when at attempt to match the lower frequency was made. Therefore, at first glance, matching at the lower or upper edge frequency appears to be equally viable options. However, a closer comparison revealed that the bandwidth obtained when matching at the lower frequency was slightly greater than that obtained by matching at the upper frequency. As a result the match for the final design was made at the lower band edge frequency of 0.2. This is evident from Fig 4.2.2.2.2. It must finally be mentioned that the bilinear transformation does not in any way alter the basic nature of the frequency response, such as its shape. The only effect of this transformation, it being a non-linear transformation
between analog and digital frequencies, is to compress higher frequencies. These frequencies fall in the region of non-linearity.

**Impulse Response and Pole Zero Diagram of Chebyshev I Filter**

Fig. 4.2.2.4 Impulse response of Chebyshev I band pass filter (Bilinear)

An interesting question that arises when comparing (4.2.2.1 and 4.2.2.2) the designed Chebyshev I band pass filters is whether the two filters are the same. One is lead to suspect from the frequency responses (Fig. 4.2.2.1.1 and Fig. 4.2.2.2.1) that they are not so. This is confirmed by observing the impulse responses (Fig. 4.2.2.1.3 and Fig. 4.2.2.2.4) and the pole-zero plots. (Fig. 4.2.2.1.4 and Fig. 4.2.2.2.5) The impulse response of the filter obtained using the bilinear transformation takes a longer time to decay than that obtained from direct application of the `cheby1` function. A possible explanation for this could be the frequency warping effect that results in the higher frequencies being attenuated. As a result the impulse response of the latter filter cannot make the fast change required for it to decay quickly.

Fig. 4.2.2.2.5 Pole-Zero diagram of Chebyshev I filter (Bilinear)
5. Conclusion

In this project, the design of IIR filters was considered. Several results from theory were verified in the design. The bilinear transformation was studied in some depth through its application to the design of two filters in Section II. The characteristics of a number of important approximations – Butterworth, Chebyshev, and Elliptic – were affirmed from the results obtained. The design of the low-pass filter in Section I was particularly insightful in comparing the relative merits and demerits of FIR and IIR filters in general as well as the individual IIR filter approximations.

The significant observations made in the design process were:

- IIR filters result in a lower order than the corresponding (designed to meet the same specification) FIR filter
- IIR filters exhibit non-linear phase.

The bilinear transformation results in a frequency warping of the higher frequencies. In general, the resulting filter will not be the same as the original analog filter. In using the bilinear function in MATLAB, the pre-warping conducted by MATLAB, is effective in maintaining the original response for the design of a high pass filter, but not for a band pass filter.

References:

Appendix 1

Matlab Code

% DSP Project 2

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

%Part 1 (a) Design of IIR Filter

%Filter Specifications
Fs = 10; fs = Fs/2; %Sampling Frequency
Fpb = 2; fpb = Fpb/fs; %Passband edge frequency
Fsb = 2.4; fsb = Fsb/fs; %Stopband edge frequency
Rpb = 0.1; %Passband Ripple
Rsb = 30; %Stopband Attenuation

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

%Butterworth Filter

[n,fn] = buttord(fpb,fsb,Rpb,Rsb);
[b,a] = butter(n,fn);
[H,w] = freqz(b,a,512,1);
figure;
% Plotting the magnitude Response
subplot(2,1,1);
plot(w, 20*log10(abs(H)));
grid on;
title(['Butterworth Lowpass Filter, Magnitude response, order=',num2str(n)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w,angle(H));
title(['Butterworth Lowpass Filter, Phase response, order=',num2str(n)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b,a,60);
figure;
stem(t,y);
title(['Butterworth Lowpass Filter, Impulse response, order=',num2str(n)]);

% Pole-Zero Plot

z = roots(b); %zeros
p = roots(a); %poles
figure;
zplane(z,p);
title('Pole-Zero plot for Butterworth Low Pass Filter');

%---------------------------------------------------------------------
%Chebyshev I Filter

[n1,fn1] = cheb1ord(fpb,fsb,Rpb,Rsb);
[b1,a1] = cheby1(n1,.1,fn1);
[H1,w1] = freqz(b1,a1,512,1);
figure;

% Plotting the magnitude Response
subplot(2,1,1);
plot(w1, 20*log10(abs(H1)));
grid on;
title(['Chebyshev I Lowpass Filter, Magnitude response, order=',num2str(n1)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w1,angle(H1));
title(['Chebyshev I Lowpass Filter, Phase response, order=',num2str(n1)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b1,a1,60);
figure;
stem(t,y);
title(['Chebyshev I Lowpass Filter, Impulse response, order=',num2str(n1)]);

% Pole-Zero Plot
z = roots(b1);%zeros
p = roots(a1);%poles
figure;
zplane(z,p);
title('Pole-Zero plot for Chebyshev I Low Pass Filter');

%---------------------------------------------------------------------
%Chebyshev II Filter

[n2,fn2] = cheb2ord(fpb,fsb,Rpb,Rsb);
[b2,a2] = cheby2(n2,30,fn2);
[H2,w2] = freqz(b2,a2,512,1);
figure;

% Plotting the magnitude Response
subplot(2,1,1);
plot(w2, 20*log10(abs(H2)));
grid on;
title(['Chebyshev II Lowpass Filter, Magnitude response, 
order=',num2str(n2)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w2,angle(H2));
title(['Chebyshev II Lowpass Filter, Phase response, 
order=',num2str(n2)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b2,a2,60);
figure;
stem(t,y);
title(['Chebyshev II Lowpass Filter, Impulse response, 
order=',num2str(n2)]);

% Pole-Zero Plot
z = roots(b2);%zeros
p = roots(a2);%poles
figure;
zplane(z,p);
title('Pole-Zero plot for Chebyshev II Low Pass Filter');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Elliptical Filter
[n3,fn3] = ellipord(fpb,fsb,Rpb,Rsb);
[b3,a3] = ellip(n3,.1,30,fn3);
[H3,w3] = freqz(b3,a3,512,1);
figure;

% Plotting the magnitude Response
subplot(2,1,1);
plot(w3, 20*log10(abs(H3))); grid on;
title(['Elliptical Lowpass Filter, Magnitude response, 
order=',num2str(n3)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w3,angle(H3));
title(['Elliptical Lowpass Filter, Phase response, 
order=',num2str(n3)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b3,a3,60);
figure;
stem(t,y);
title(['Elliptical Lowpass Filter, Impulse response,
order=',num2str(n3)]);

% Pole-Zero Plot
z = roots(b3);  %zeros
p = roots(a3);  %poles
figure;
zplane(z,p);
title('Pole-Zero plot for Elliptical Low Pass Filter');
% Part 1(b) Design of FIR filter to meet specifications given in Part 1(a) above

% Kaiser Window

% Filter Specifications
Fs = 10; fs = Fs/2;   % Sampling Frequency
Fpb = 2; fpb = Fpb/fs; % Passband edge frequency
Fsb = 2.4; fsb = Fsb/fs; % Stopband edge frequency
Rpb = 0.09;           % Passband Ripple
Rsb = 30;             % Stopband Attenuation

% Design
f = [Fpb Fsb];
a = [1 0];
Gpb = 1 - 10^(-Rpb/20); % Deviation from passband magnitude of 1
Gsb = 10^(-Rsb/20); % Min. stopband atten.
dev = [Gpb Gsb];
[N,Wn,beta,ftype] = kaiserord(f,a,dev,Fs);
h = fir1(N,Wn,kaiser(N+1,beta));

% Computing the frequency response of the practical filter.
figure;
[H,w] = freqz(h,1,512,1);

% Plotting the magnitude response
subplot(2,1,1);
plot(w, 20*log10(abs(H)));
gird on;
title(['Kaiser Window: Magnitude response, N=',num2str(N)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w,angle(H));
title(['Kaiser Window: Phase response, N=',num2str(N)]);
xlabel('f');
ylabel('angle H(f) rad');
gird on;

% Impulse Response
n = [0:N]; % order = N and length = N+1
figure;
stem(n,h);
title(['Impulse Response of FIR Filter using Kaiser window, order = ',num2str(N-1)]);

% Computing the pole zero plot
figure
z = roots(h); % zeros
den = [zeros(1,N-1) 1];
p = roots(den); % poles
zplane(z,p);
title('Pole Zero Plot for Kaiser Window');
Part 1(b) Design of optimum equiripple FIR filter

rp = 0.09; % Passband ripple
rs = 30; % Stopband ripple
fs = 10; % Sampling frequency
f = [2 2.4]; % Cutoff frequencies
a = [1 0]; % Desired amplitudes

Gpb1 = (10^(rp/20)-1)/(10^(rp/20)+1);
Gpb2 = 1 - 10^(-rp/20);
Gsb = 10^(-rs/20);

% Compute deviations
dev = [Gpb2 Gsb];
[n,fo,ao,w] = remezord(f,a,dev,fs);
b = remez(n,fo,ao,w);

%title('Lowpass Filter Designed to Specifications');

% Plotting the magnitude Response
figure;
subplot(2,1,1);
plot(w, 20*log10(abs(H)));
grid on;
title(['Equiripple Lowpass Filter, Magnitude response, order=',num2str(n)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w,angle(H));
title(['Equiripple Lowpass Filter, Phase response, order=',num2str(n)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b,1,60);
figure;
stem(t,y);
title(['Equiripple Lowpass Filter, Impulse response, order=',num2str(n)]);

% Pole-Zero Plot
z = roots(b); %zeros
p = roots(1); %poles
figure;
zplane(z,p);
title('Pole-Zero plot for Equiripple Low Pass Filter');
%Part 2(a) Design of an Elliptic High Pass Filter

%Elliptical Filter

%Filter Specifications
Fs = 10; fs = Fs/2;  %Sampling Frequency
Fpb = 2.4; fpb = Fpb/fs;  %Passband edge frequency
Fsb = 2; fsb = Fsb/fs;  %Stopband edge frequency
Rpb = 1;  %Passband Ripple
Rsb = 40;  %Stopband Attenuation

[n3,fn3] = ellipord(fpb,fsb,Rpb,Rsb);
[b3,a3] = ellip(n3,1,40,fn3,'high');
[H3,w3] = freqz(b3,a3,512,1);
figure;

% Plotting the magnitude Response
subplot(2,1,1);
plot(w3, 20*log10(abs(H3)));
ggrid on;
ttitle(['Elliptical Highpass Filter, Magnitude response, order=',num2str(n3)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w3,angle(H3));
ttitle(['Elliptical Highpass Filter, Phase response, order=',num2str(n3)]);
xlabel('f');
ylabel('angle H(f) rad');
ggrid on;

% Impulse Response
[y,t] = impz(b3,a3,60);
figure;
stem(t,y);
ttitle(['Elliptical Highpass Filter, Impulse response, order=',num2str(n3)]);

% Pole-Zero Plot
z = roots(b3);%zeros
p = roots(a3);%poles
figure;
zplane(z,p);
ttitle('Pole-Zero plot for Elliptical HighPass Filter');
% Part 2(b) Design of an elliptic HPF by conversion from LPF specs and using bilinear transformation

% Filter Specifications
Fs = 10; % Sampling Frequency
f_pb = 2.4; % Passband edge frequency
f_sb = 2; % Stopband edge frequency
Rpb = 1; % Passband Ripple
Rsb = 40; % Stopband Attenuation

N = 5;
[z, p, k] = ellipap(N, Rpb, Rsb); % Designing an analog LP prototype with cut-off = 1 rad/s
b = k*poly(z); % Numerator polynomial
a = poly(p); % Denominator
[b1, a1] = lp2hp(b, a, 2*pi*f_pb); % Low Pass to High Pass transformation
[b2, a2] = bilinear(b1, a1, Fs, f_pb); % Bilinear Transformation, f_pb = Pre-warping freq.
[H3, w3] = freqz(b2, a2, 512, 1);
figure;

% Plotting the magnitude Response
subplot(2, 1, 1);
plot(w3, 20*log10(abs(H3)));
grid on;
title(['Elliptical Highpass Filter, Magnitude response, order=', num2str(N)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2, 1, 2);
plot(w3, angle(H3));
title(['Elliptical Highpass Filter, Phase response, order=', num2str(N)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y, t] = impz(b2, a2, 60);
figure;
stem(t, y);
title(['Elliptical Highpass Filter, Impulse response, order=', num2str(N)]);

% Pole-Zero Plot
z = roots(b2); % zeros
p = roots(a2); % poles
figure;
zplane(z, p);
title('Pole-Zero plot for Elliptical HighPass Filter');
%Part 2(c) Design of Chebyshev IIR Filter

% Filter Specifications
Fs = 10; fs = Fs/2; % Sampling Frequency
Fpb1 = 2; fpb1 = Fpb1/fs; % Lower Passband edge frequency
Fpb2 = 4; fpb2 = Fpb2/fs; % Upper Passband edge frequency
Rp = 1; % Passband Ripple
n = 3; % Since it returns a filter of order 2*n
Wn = [fpb1 fpb2];
[b, a] = cheby1(n, Rp, Wn);
[H, w] = freqz(b, a, 512, 1);

figure;

% Plotting the magnitude response
subplot(2,1,1);
plot(w, 20*log10(abs(H)));
grid on;
title(['Chebyshev I Bandpass Filter, Magnitude response, order=',num2str(2*n)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w, angle(H));
title(['Chebyshev I Bandpass Filter, Phase response, order=',num2str(2*n)]);
xlabel('f');
ylabel('angle H(f) rad');

% Impulse Response
[y, t] = impz(b, a, 60);
figure;
stem(t, y);
title(['Chebyshev I Bandpass Filter, Impulse response, order=',num2str(2*n)]);

% Pole-Zero Plot
z = roots(b); % zeros
p = roots(a); % poles
figure;
zplane(z, p);
title('Pole-Zero plot for Chebyshev I BandPass Filter');
% Part 2(d) Design of Chebyshev I BPF using Bilinear transformation

%Filter Specifications
Fs = 10; fs = Fs/2; %Sampling Frequency
Fpb1 = 2; fpb1 = Fpb1/fs; %Passband edge frequency
Fpb2 = 4; fpb2 = Fpb2/fs; %Stopband edge frequency
Rp = 1; %Passband Ripple
wo = 2*pi*sqrt(Fpb1*Fpb2); %Centre Frequency rad/s
bw = (Fpb2-Fpb1)*2*pi; %Bandwidth rad/s
N = 3;
[z,p,k] = cheb1ap(N,Rp);
b = k*poly(z); % Numerator polynomial
a = poly(p); % Denominator
[bt,at] = lp2bp(b,a,wo,bw); % Low Pass to Bandpass Transformation
[b2,a2] = bilinear(bt,at,Fs,Fpb1); % Bilinear Transformation matching lower pass frequency
%Chebyshev I Filter

[H,w] = freqz(b2,a2,512,1);
figure;
% Plotting the magnitude response
subplot(2,1,1);
plot(w, 20*log10(abs(H)));
grid on;
title(['Chebyshev I BandPass Filter, Magnitude response,' num2str(2*N)]);
xlabel('f');
ylabel('|H(f)| dB');

% Plotting the phase response
subplot(2,1,2);
plot(w,angle(H));
title(['Chebyshev I BandPass Filter, Phase response,' num2str(2*N)]);
xlabel('f');
ylabel('angle H(f) rad');
grid on;

% Impulse Response
[y,t] = impz(b2,a2,60);
figure;
stem(t,y);
title(['Chebyshev I BandPass Filter, Impulse response,' num2str(2*N)]);

% Pole-Zero Plot
z = roots(b2); %zeros
p = roots(a2); %poles
figure;
zplane(z,p);
title('Pole-Zero plot for Chebyshev I BandPass Filter');
APPENDIX 2

Matlab Functions:

The matlab functions used along with a brief explanation of each is included below:

cheby1 / cheby2

\[ [b,a] = \text{cheby1}(n,Rp,Wn) \]

\[ [b,a] = \text{cheby2}(n,Rp,Wn) \]

where \( Wn \) is the cutoff frequency where the magnitude response of the filter is equal to \(-Rp\) dB

\( Rp \) is the peak to peak ripple

Chebyshev Type I filters are equiripple in the passband and monotonic in the stopband. Chebyshev Type II filters are equiripple in the stopband and equiripple in the stopband. The function returns the filter coefficients in the length \( n+1 \) row vectors \( b \) and \( a \), with coefficients in descending powers of \( z \).

\[ H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \cdots + a(n+1)z^{-n}} \]

If \( Wn \) is a two-element vector, \( Wn = [w1 \ w2] \), \text{cheby1} returns an order \( 2n \) bandpass filter with \( w_1 < \omega < w_2 \).

Chebyshev Type I (Type II) filters are equiripple in the passband (stopband) and monotonic in the stopband (passband). The poles are evenly spaced about an ellipse in the left half plane. The Chebyshev Type I cutoff frequency \( \omega \) is set to 1.0 for a normalized result. This is the frequency at which the passband ends and the filter has magnitude response of \( 10^{ Rp/20} \).

Algorithm:

1. It finds the lowpass analog prototype poles, zeros, and gain using the \text{cheb1ap} function.
2. It converts the poles, zeros, and gain into state-space form.
3. It transforms the lowpass filter into a bandpass, highpass, or bandstop filter with desired cutoff frequencies, using a state-space transformation.
4. For digital filter design, \text{cheby1} uses \text{bilinear} to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment guarantees that the analog filters and the digital filters will have the same frequency response magnitude at \( Wn \) or \( w1 \) and \( w2 \).
5. It converts the state-space filter back to transfer function or zero-pole-gain form, as required.

\text{cheb1ap}

Chebyshev Type I analog lowpass filter prototype

\[ [z,p,k] = \text{cheb1ap}(n,Rp) \]

Description
It returns the poles and gain of an order $n$ Chebyshev Type I analog lowpass filter prototype with $R_p$ dB of ripple in the passband. The function returns the poles in the length $n$ column vector $p$ and the gain in scalar $k$. $z$ is an empty matrix, because there are no zeros. The transfer function is

$$H(s) = \frac{z(s)}{p(s)} = \frac{k}{(s - p(1))(s - p(2)) \cdots (s - p(n))}$$

**Butter**

Butterworth analog and digital filter design

$$[b,a] = \text{butter}(n,Wn)$$

where

- $n$ is the order of the filter
- $Wn$ is the cutoff frequency where the magnitude response is $1/\sqrt{2}$

Butterworth filters are characterized by a magnitude response that is maximally flat in the passband and monotonic overall. Butterworth filters sacrifice rolloff steepness for monotonicity in the pass- and stopbands.

It returns the filter coefficients in length $n+1$ row vectors $b$ and $a$, with coefficients in descending powers of $z$.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \cdots + a(n+1)z^{-n}}$$

If $Wn$ is a two-element vector, $Wn = [w1 \ w2]$, cheby2 returns an order $2*n$ bandpass filter with passband $w1 < \omega < w2$.

**Algorithm**

1. It finds the lowpass analog prototype poles, zeros, and gain using the buttap function.
2. It converts the poles, zeros, and gain into state-space form.
3. It transforms the lowpass filter into a bandpass, highpass, or bandstop filter with desired cutoff frequencies, using a state-space transformation.
4. For digital filter design, butter uses bilinear to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment guarantees that the analog filters and the digital filters will have the same frequency response magnitude at $Wn$ or $w1$ and $w2$.
5. It converts the state-space filter back to transfer function or zero-pole-gain form, as required.
**ellip**

Elliptic (Cauer) filter design

```
[b,a] = ellip(n,Rp,Rs,Wn)
```

where

- `n` is the order of the filter
- `Rp` is the ripple in dB in the Passband
- `Rs` is the ripple in dB in the Stopband

Elliptic filters offer steeper rolloff characteristics than Butterworth or Chebyshev filters, but are equiripple in both the pass- and stopbands. In general, elliptic filters meet given performance specifications with the lowest order of any filter type.

It returns the filter coefficients in the length `n+1` row vectors `b` and `a`, with coefficients in descending powers of `z`.

\[
H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \cdots + a(n+1)z^{-n}}
\]

If `Wn` is a two-element vector, `Wn = [w1 w2]`, `ellip` returns an order `2*n` bandpass filter with passband `w1 < \omega < w2`.

`ellip` sets the cutoff frequency `\omega` of the elliptic filter to 1 for a normalized result. The cutoff frequency is the frequency at which the passband ends and the filter has a magnitude response of \(10^{\frac{Rp}{20}}\).

**Algorithm**

The design of elliptic filters is the most difficult and computationally intensive of the Butterworth, Chebyshev Type I and II, and elliptic designs.

1. It finds the lowpass analog prototype poles, zeros, and gain using the `ellipap` function.
2. It converts the poles, zeros, and gain into state-space form.
3. It transforms the lowpass filter to a bandpass, highpass, or bandstop filter with the desired cutoff frequencies using a state-space transformation.
4. For digital filter design, `ellip` uses `bilinear` to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment guarantees that the analog filters and the digital filters will have the same frequency response magnitude at `Wn` or `w1` and `w2`.
5. It converts the state-space filter back to transfer function or zero-pole-gain form, as required.
ellipap

Elliptic analog lowpass filter prototype

\[ [z, p, k] = \text{ellipap}(n, Rp, Rs) \]

Description

It returns the zeros, poles, and gain of an order \( n \) elliptic analog lowpass filter prototype, with \( Rp \) dB of ripple in the passband, and a stopband \( Rs \) dB down from the peak value in the passband. The zeros and poles are returned in length \( n \) column vectors \( z \) and \( p \) and the gain in scalar \( k \). If \( n \) is odd, \( z \) is length \( n - 1 \). The transfer function is

\[
H(s) = \frac{z(s)}{p(s)} = k \frac{(s - z(1))(s - z(2)) \cdots (s - z(n))}{(s - p(1))(s - p(2)) \cdots (s - p(n))}
\]

Elliptic filters offer steeper rolloff characteristics than Butterworth and Chebyshev filters, but they are equiripple in both the passband and the stopband. Of the four classical filter types, elliptic filters usually meet a given set of filter performance specifications with the lowest filter order.

Transformations in filter design

Low pass to Bandpass

lp2bp

Transform lowpass analog filters to bandpass

\[ [bt, at] = \text{lp2bp}(b, a, Wo, Bw) \]

Transforms an analog lowpass filter prototype given by polynomial coefficients into a bandpass filter with center frequency \( Wo \) and bandwidth \( Bw \). Row vectors \( b \) and \( a \) specify the coefficients of the numerator and denominator of the prototype in descending powers of \( s \).

\[
\frac{b(s)}{a(s)} = \frac{b(1)s^n + \cdots + b(n)s + b(n + 1)}{a(1)s^m + \cdots + a(m)s + a(m + 1)}
\]

Scalars \( Wo \) and \( Bw \) specify the center frequency and bandwidth in units of rad/s. For a filter with lower band edge \( w1 \) and upper band edge \( w2 \), use \( Wo = \sqrt{w1 \cdot w2} \) and \( Bw = w2 - w1 \). lp2bp returns the frequency transformed filter in row vectors \( bt \) and \( at \).

lp2hp

Transform lowpass analog filters to highpass
$$[bt,at] = \text{lp2hp}(b,a,Wo)$$

lp2hp transforms analog lowpass filter prototypes with a cutoff frequency of 1 rad/s into highpass filters with desired cutoff frequency. The transformation is one step in the digital filter design process for the butter, cheby1, cheby2, and ellip functions.

Transforms an analog lowpass filter prototype given by polynomial coefficients into a highpass filter with cutoff frequency $Wo$. Row vectors $b$ and $a$ specify the coefficients of the numerator and denominator of the prototype in descending powers of $s$.

$$\frac{b(s)}{a(s)} = \frac{b(1)s^n + \cdots + b(n)s + b(n + 1)}{a(1)s^m + \cdots + a(m)s + a(m + 1)}$$

Scalar $Wo$ specifies the cutoff frequency in units of radians/second. The frequency transformed filter is returned in row vectors $bt$ and $at$.

**bilinear**

Bilinear transformation method for analog-to-digital filter conversion

$$[zd,pd,kd] = \text{bilinear}(z,p,k,fs)$$
$$[zd,pd,kd] = \text{bilinear}(z,p,k,fs,Fp)$$

**Description**

The bilinear transformation is a mathematical mapping of variables. In digital filtering, it is a standard method of mapping the $s$ or analog plane into the $z$ or digital plane. It transforms analog filters, designed using classical filter design techniques, into their discrete equivalents.

The function convert the $s$-domain transfer function specified by $z$, $p$, and $k$ to a discrete equivalent. Inputs $z$ and $p$ are column vectors containing the zeros and poles, $k$ is a scalar gain, and $fs$ is the sampling frequency in hertz. bilinear returns the discrete equivalent in column vectors $zd$ and $pd$ and scalar $kd$. $Fp$ is the optional match frequency, in hertz, for prewarping. It is important that the numerator order cannot be greater than that of the denominator.

**Zero-Pole-Gain Algorithm**

1. For a system in zero-pole-gain form, bilinear performs four steps:
2. If $Fp$ is present, $k = 2*\pi*Fp/\tan(\pi*Fp/fs)$; otherwise $k = 2*fs$.
3. It strips any zeros at $\pm \infty$ using $z = z(\text{finite}(z))$;
4. It transforms the zeros, poles, and gain using
   - $pd = (1+p/k)/(1-p/k)$;
   - $zd = (1+z/k)/(1-z/k)$;
   - $kd = \text{real}(k*\text{prod}(fs-z)/\text{prod}(fs-p))$;
5. It adds extra zeros at -1 so the resulting system has equivalent numerator and denominator order.
Determination of the order of different filters

cheb1ord / cheb2ord

Calculate the order for a Chebyshev Type I / II filter

Syntax

\[ [n, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs) \]
\[ [n, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs, 's') \]
\[ [n, Wn] = \text{cheb2ord}(Wp, Ws, Rp, Rs, 's') \]

Description

cheb1ord / cheb2ord calculates the minimum order of a digital or analog Chebyshev Type I / Type 2 filter required to meet a set of filter design specifications. If ‘s’ is included then it will calculate the order for an analog filter.

It returns the lowest order \( n \) of the Chebyshev Type I / II filter that loses no more than \( Rp \) dB in the passband and has at least \( Rs \) dB of attenuation in the stopband. The scalar (or vector) of corresponding cutoff frequencies \( Wn \), is also returned.

Wp  Passband corner frequency \( Wp \), the cutoff frequency, is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency, \( \pi \) radians per sample.
Ws  Stopband corner frequency \( Ws \), is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency.
Rp  Passband ripple, in decibels. This value is the maximum permissible passband loss in decibels.
Rs  Stopband attenuation, in decibels. This value is the number of decibels the stopband is down from the passband.

ellipord

Calculate the minimum order for elliptic filters

\[ [n, Wn] = \text{ellipord}(Wp, Ws, Rp, Rs) \]
\[ [n, Wn] = \text{ellipord}(Wp, Ws, Rp, Rs, 's') \]

Description

ellipord calculates the minimum order of a digital or analog elliptic filter required to meet a set of filter design specifications.

It returns the lowest order \( n \) of the elliptic filter that loses no more than \( Rp \) dB in the passband and has at least \( Rs \) dB of attenuation in the stopband. The scalar (or vector) of corresponding cutoff frequencies \( Wn \), is also returned. Use the output arguments \( n \) and \( Wn \) in ellip. If ‘s’ is specified in the equation then it will return the order of the corresponding analog filter.
Wp
Passband corner frequency Wp, the cutoff frequency, is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency, \( \pi \) radians per sample.

Ws
Stopband corner frequency Ws, is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency.

Rp
Passband ripple, in decibels. Twice this value specifies the maximum permissible passband width in decibels.

Rs
Stopband attenuation, in decibels. This value is the number of decibels the stopband is attenuated with respect to the passband response.

\textbf{buttord}

Calculate the order and cutoff frequency for a Butterworth filter

\[ [n,Wn] = \text{buttord}(Wp,Ws,Rp,Rs) \]
\[ [n,Wn] = \text{buttord}(Wp,Ws,Rp,Rs,'s') \]

\textbf{Description}

buttord calculates the minimum order of a digital or analog Butterworth filter required to meet a set of filter design specifications.

\( n \) returns the lowest order, of the digital Butterworth filter that loses no more than \( Rp \) dB in the passband and has at least \( Rs \) dB of attenuation in the stopband. The scalar (or vector) of corresponding cutoff frequencies, \( Wn \), is also returned. If ‘s’ is included it will return the order of the corresponding analog filter.

Wp  Passband corner frequency Wp, the cutoff frequency, is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency, \( \pi \) radians per sample.
Ws  Stopband corner frequency Ws, is a scalar or a two-element vector with values between 0 and 1, with 1 corresponding to the normalized Nyquist frequency.
Rp  Passband ripple, in decibels. This value is the maximum permissible passband loss in decibels.
Rs  Stopband attenuation, in decibels. This value is the number of decibels the stopband is down from the passband.
kaiserord

Estimates parameters for an FIR filter design with a Kaiser window

\[
[n,Wn,beta,ftype] = kaiserord(f,a,dev)
\]
\[
[n,Wn,beta,ftype] = kaiserord(f,a,dev,fs)
\]

**Description**

kaiserord returns a filter order \( n \) and beta parameter to specify a Kaiser window for use with the `fir1` function. Given a set of specifications in the frequency domain, kaiserord estimates the minimum FIR filter order that will approximately meet the specifications. kaiserord converts the given filter specifications into passband and stopband ripples and converts cutoff frequencies into the form needed for windowed FIR filter design.

It finds the approximate order \( n \), normalized frequency band edges \( W_n \), and weights that meet input specifications \( f \), \( a \), and \( \text{dev} \). \( f \) is a vector of band edges and \( a \) is a vector specifying the desired amplitude on the bands defined by \( f \). The length of \( f \) is twice the length of \( a \), minus 2. Together, \( f \) and \( a \) define a desired piecewise constant response function.

\( \text{dev} \) is a vector the same size as \( a \) that specifies the maximum allowable error or deviation between the frequency response of the output filter and its desired amplitude, for each band. The entries in \( \text{dev} \) specify the passband ripple and the stopband attenuation. \( \text{dev} \) is specified as a positive number, representing absolute filter gain.

remez

Compute the Parks-McClellan optimal FIR filter design

\[
b = remez(n,f,a)
\]
\[
b = remez(n,f,a,w)
\]

**Description**

remez designs a linear-phase FIR filter using the Parks-McClellan algorithm. The Parks-McClellan algorithm uses the Remez exchange algorithm and Chebyshev approximation theory to design filters with an optimal fit between the desired and actual frequency responses. The filters are optimal in the sense that the maximum error between the desired frequency response and the actual frequency response is minimized. Filters designed this way exhibit an equiripple behavior in their frequency responses and are sometimes called *equiripple* filters. remez exhibits discontinuities at the head and tail of its impulse response due to this equiripple nature.

It returns row vector \( b \) containing the \( n+1 \) coefficients of the order \( n \) FIR filter whose frequency-amplitude characteristics match those given by vectors \( f \) and \( a \).

The output filter coefficients (taps) in \( b \) obey the symmetry relation

\[
b(k) = b(n+2-k), \quad k = 1, \ldots, n+1
\]
Vectors \( f \) and \( a \) specify the frequency-magnitude characteristics of the filter:

1. \( f \) is a vector of pairs of normalized frequency points, specified in the range between 0 and 1, where 1 corresponds to the Nyquist frequency. The frequencies must be in increasing order.
2. \( a \) is a vector containing the desired amplitudes at the points specified in \( f \).
3. The desired amplitude at frequencies between pairs of points \((f(k), f(k+1))\) for \( k \) odd is the line segment connecting the points \((f(k), a(k))\) and \((f(k+1), a(k+1))\).
4. The desired amplitude at frequencies between pairs of points \((f(k), f(k+1))\) for \( k \) even is unspecified. The areas between such points are transition or "don't care" regions.
5. \( f \) and \( a \) must be the same length. The length must be an even number.