INFINITE-IMPULSE RESPONSE DIGITAL FILTERS

Classical analog filters and their conversion to digital filters

---

A recursive IIR digital filter is an LTI system based on a difference equation of the form:

$$y(n) = -\sum_{k=1}^{N} a(k)\, y(n-k) + \sum_{k=0}^{M} b(k)\, x(n-k) \tag{1}$$

1. The impulse response $h(n)$ is infinite in length.

2. A system described by this type of difference equation is called an IIR (Infinite Impulse Response) filter, a recursive filter, or an autoregressive moving-average (ARMA) filter.

The output $y(n)$ of the filter can be written as

$$y(n) = \sum_{k=0}^{\infty} h(k)\, x(n-k). \tag{2}$$

As the impulse response is infinite, the convolutional sum is an infinite sum. The transfer function of the system $H(z)$ can be written in term of the impulse response as

$$H(z) = \sum_{k=0}^{\infty} h(k)\, z^{-k}. \tag{3}$$

The frequency response is therefore given by

$$H(e^{j\omega}) = \sum_{n=0}^{\infty} h(n)\, e^{-j\omega}. \tag{4}$$

The transfer function can also be written in terms of the difference equation as

$$H(z) = \frac{b_0 + b_1\, z^{-1} + b_2\, z^{-2} + \cdots + b_M\, z^{-M}}{1 + a_1\, z^{-1} + a_2\, z^{-2} + \cdots + a_N\, z^{-N}} \tag{5}$$

or

$$H(z) = \frac{B(z)}{A(z)} \tag{6}$$

where

$$B(z) = \sum_{n=0}^{M} b(n)\, z^{-n} \tag{7}$$

$$A(z) = 1 + \sum_{n=1}^{N} a(n)\, z^{-n}. \tag{8}$$

$H(z)$ can also be written as

$$H(z) = \frac{z^{-M}}{z^{-N}} \cdot \frac{b_0\, z^M + b_1\, z^{M-1} + b_2\, z^{M-2} + \cdots + b_M}{z^N + a_1\, z^{N-1} + a_2\, z^{N-2} + \cdots + a_N}. \tag{9}$$

The the zeros of $H(z)$ are the roots of the polynomial

$$b_0\, z^M + b_1\, z^{M-1} + b_2\, z^{M-2} + \cdots + b_M \tag{10}$$

and the poles of $H(z)$ are the roots of the polynomial

$$z^N + a_1\, z^{N-1} + a_2\, z^{N-2} + \cdots + a_N. \tag{11}$$

In addition, if $N > M$, then $z = 0$ is a zero of multiplicity $N - M$; and if $M > N$, then $z = 0$ is a pole of multiplicity $M - N$.

Computing $h(n)$. Given the difference equation coefficients $a(k)$ and $b(k)$, the impulse response $h(n)$ can be obtained by taking the inverse Z-transform of $H(z)$, but it is usually simpler to calculate $h(n)$ numerically by running the difference equation with the input $x(n) = \delta(n)$. The Matlab function `filter` implements a recursive difference equation. The following code fragment computes the first 30 values of $h(n)$ from $a(k)$ and $b(k)$.

```
x = [1, zeros(1,29)];
h = filter(b,a,x);
```

where the vectors `b` and `a` contain the filter parameters

$$\texttt{b} = [b(0),\ b(1),\ \ldots,\ b(M)] \tag{12}$$

$$\texttt{a} = [a(0),\ a(1),\ \ldots,\ a(N)] \tag{13}$$

Computing $H(e^{j\omega})$. How can one compute the frequency response $H(e^{j\omega})$ from $h(n)$ over a grid of equally spaced samples? One approach is to write out the definition of the DTFT and truncate the infinite sum to a finite sum. Let

$$\omega_k = \frac{2\pi}{L}\, k, \qquad 0 \leq k \leq L-1, \tag{14}$$

be a set of equally spaced frequencies. Then

$$H(e^{j\omega_k}) = \sum_{n=0}^{\infty} h(n)\, e^{-j\omega_k n} \tag{15}$$

$$\approx \sum_{n=0}^{L-1} h(n)\, e^{-j\omega_k n} \tag{16}$$

$$\approx \mathsf{DFT}_L\{h(n) \;:\; 0 \leq n \leq L-1\} \tag{17}$$

However, this method has several disadvantages.

1. The resulting values are only approximate because part of the infinitely long impulse response is truncated.

2. The number of terms L must be large for the result to be accurate.

3. One needs to compute the impulse response $h(n)$.

A second, better approach is to write $H(z)$ as a rational function.

$$H(e^{j\omega_k}) = \left.\frac{B(z)}{A(z)}\right|_{z=e^{j\omega_k}} \tag{18}$$

$$= \frac{B(e^{j\omega_k})}{A(e^{j\omega_k})} \tag{19}$$

$$= \frac{\displaystyle\sum_{n=0}^{M} b(n)\, e^{-j\omega_k n}}{1 + \displaystyle\sum_{n=1}^{N} a(n)\, e^{-j\omega_k n}} \tag{20}$$

$$= \frac{\mathsf{DFT}_L\{b(n)\}}{\mathsf{DFT}_L\{a(n)\}} \tag{21}$$

This method has the advantages that

1. Exact values of $H(e^{j\omega})$ are obtained at the values ω_k.

2. It is simple to implement — the L-point DFT of the numerator coefficients is simply divided, point by point, by the L-point DFT of the denominator coefficients.

The following Matlab function, for calculating the frequency response $H(e^{j\omega})$ on equally spaced frequencies, is based on the second approach.

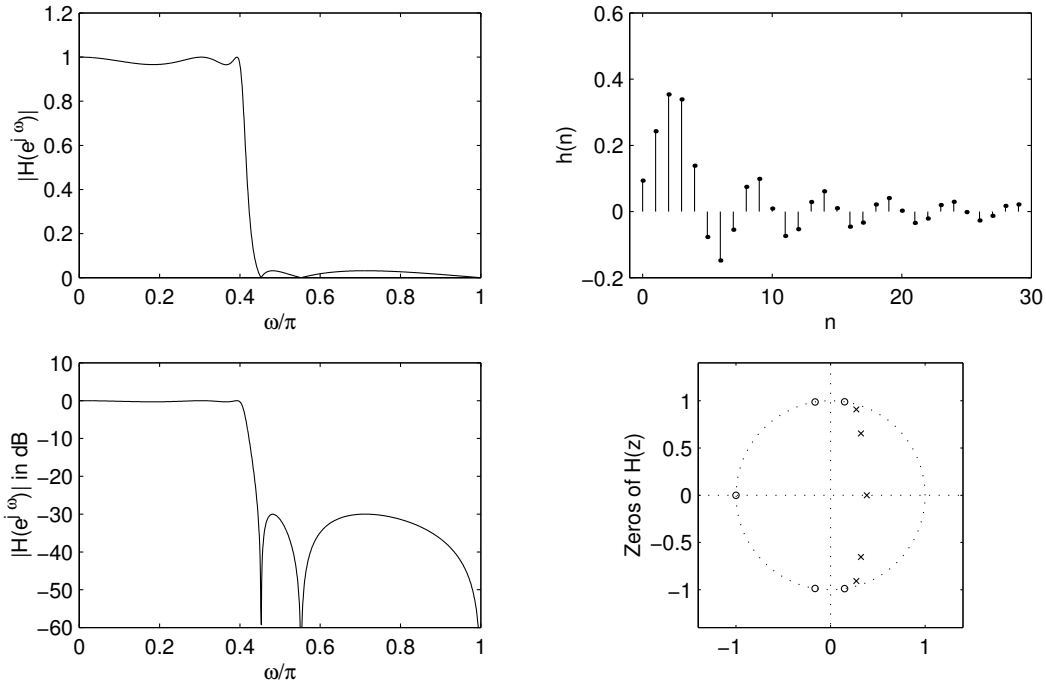
```
function [H,w] = iirfr(b,a,L)
% [H,w] = iirfr(b,a,L)
% -- input ----
% b : numerator  coefficients
% a : denominator coefficients
% L : number of points from [0:pi]
% -- output ---
% H : frequency response of an IIR digital filter
% w : [1:L]*pi/L

H = fft(b,2*L)./fft(a,2*L);   % frequency response from 0 to 2*pi.
H = H([0:L]+1);               % frequency response from 0 to pi.
w = [0:L]*pi/L;               % frequency grid.
```

EXAMPLE

The following values are the coefficients of a difference equation of an IIR digital elliptic filter.

k	$b(k)$	$a(k)$
0	0.0931	1.0000
1	0.0960	−1.5757
2	0.1801	2.2408
3	0.1801	−1.5554
4	0.0960	0.8123
5	0.0931	−0.1837

A plot of the frequency response, impulse response, and pole-zero diagram can be found with Matlab.



This figure was produces with the following code.

```
b = [0.0931    0.0960    0.1801    0.1801    0.0960    0.0931];
a = [1.0000   -1.5757    2.2408   -1.5554    0.8123   -0.1837];

% ------ Plot frequency response magnitude ---
subplot(4,2,1)
[H,w]=iirfr(b,a,2^10);
plot(w/pi,abs(H))
axis([0 1 0 1.2])
xlabel('\omega/\pi')
ylabel('|H(e^{j \omega})|')

% ------ Plot frequency response in dB -------
subplot(4,2,3)
plot(w/pi,20*log10(abs(H)))
axis([0 1 -60 10])
xlabel('\omega/\pi')
ylabel('|H(e^{j \omega})| in dB')

% ------ Plot impulse response h(n) ----------
subplot(4,2,2)
N  = 30;
im = [1 zeros(1,N-1)];
h  = filter(b,a,im);
stem(0:N-1,h,'.')
xlabel('n')
ylabel('h(n)')
axis([-1 30 -0.2 0.6])

% ------ Plot pole-zero diagram --------------
subplot(4,2,4)
z = zplane(b,a);
set(z,'markersize',3);
ylabel('Zeros of H(z)')
xlabel('')
axis([-1 1 -1 1]*1.4)
axis square

orient tall
print -deps ex1
```

REMARKS ON RECURSIVE IIR DIGITAL FILTERS

1. They are recursive.
2. They have poles as well as zeros. They will be unstable if not all the poles are inside $|z| = 1$.
3. They can not have linear-phase.
4. Methods for IIR filter design are either more complicated or less flexible than FIR design methods.
5. The implementation of IIR filters is more sensitive to finite precision effects than FIR filters are.
6. The advantage of IIR filters over FIR ones is that for the same filter complexity (number of filter parameters) the magnitude response of an IIR filter can be significantly better than that of an FIR one.

A causal IIR filter implemented with a rational transfer function can not have linear-phase. Two general approaches are:

1. Ignore the phase response and design a filter so that $|H(e^{j\omega})|$ matches a desired function $D(\omega)$.

2. If the phase response is important, then the design problem becomes more complicated.

The most common method for designing standard IIR digital filters is to convert a classical analog filter in a digital one.

- Highly developed methods for the design of classical analog filters can be used for the design of digital filters. Formulas exist for the classic analog filters.

- This method is of limited flexibility because not all IIR digital filters can be obtained by converting classical analog filters. The conversion of an analog filter into a digital filter works very well for the design of standard types: low-pass, high-pass, band-pass, and band-reject filters. However, if one wants to place constraints on the filters, or design a non-standard type of response, then the classical analog filters can usually not be used.

- A problem with the classical analog filters is that there is no control over the phase of the frequency response. They are developed so as to have good magnitude response but the phase-response can be very nonlinear.

- When an analog filter is converted into a digital filter, you generally get an IIR digital filter (not an FIR one). That is why the conversion of an analog filter to a digital one is not used for FIR filter design.

- The conversion of an analog filter into a digital one generally yields a digital filter for which the numerator and denominator of $H(z)$ have the same degree. $M = N$.

The transfer function of an analog filter is given by the Laplace transform of its impulse response,

$$H_a(s) = \mathcal{L}\{h(t)\} := \int_{-\infty}^{\infty} h(t)\, e^{-st} dt. \tag{22}$$

The subscript a is used to denote an analog transfer function. For realizable analog filters, the transfer function can be written as a rational function,

$$H_a(s) = \frac{P(s)}{Q(s)} \tag{23}$$

where $P(s)$ and $Q(s)$ are polynomials in s. The frequency response of an analog filter is obtained by evaluating $H_a(s)$ on the imaginary axis. The frequency response is given by

$$H_a(j\omega) = H_a(s)|_{s=j\omega}\,. \tag{24}$$

For a stable, causal filter, all the poles of $H_a(s)$ must lie in the left half plane (LHP)

$$\text{Re}\{p_k\} < 0 \tag{25}$$

for all p_k where $Q(p_k) = 0$.

The Square Magnitude. Note:

$$|H_a(j\omega)|^2 = H_a(j\omega) \cdot \overline{H_a(j\omega)} \tag{26}$$

$$= H_a(j\omega) \cdot H_a(-j\omega) \tag{27}$$

$$= H_a(s) \cdot H_a(-s)|_{s=j\omega} \tag{28}$$

Define

$$R(s) := H_a(s) \cdot H_a(-s). \tag{29}$$

Then

$$R(j\omega) = |H_a(j\omega)|^2. \tag{30}$$

For convenience, define

$$F(\omega) := R(j\omega), \tag{31}$$

so that we do not need to carry j along. Then

$$\boxed{R(s) = F(s/j)} \tag{32}$$

and

$$\boxed{F(\omega) = |H_a(j\omega)|^2.} \tag{33}$$

Note that:

$$R(-s) = R(s) \tag{34}$$

so $R(s)$ is an *even* function of s. Also note that,

$$F(-\omega) = R(-j\omega) = R(j\omega) = F(\omega) \tag{35}$$

so $F(\omega)$ is an *even* function of ω.

Example: If

$$H_a(s) = \frac{0.2\,s^2 + 0.02\,s + 1}{s^3 + s^2 + 2\,s + 1} \tag{36}$$

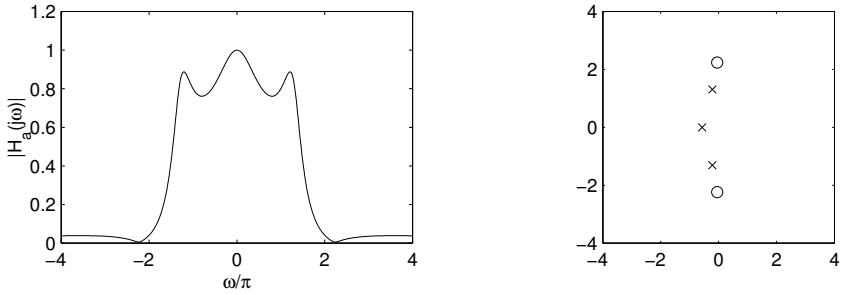
then

$$R(s) = H_a(s) \cdot H_a(-s) = \frac{0.04\,s^4 + 0.3996\,s^2 + 1}{-s^6 - 3\,s^4 - 2\,s^2 + 1} \tag{37}$$

and

$$F(\omega) = R(j\omega) = \frac{0.04\,\omega^4 - 0.3996\,\omega^2 + 1}{\omega^6 - 3\,\omega^4 + 2\,\omega^2 + 1} \tag{38}$$

Note that the function $F(\omega)$ contains only even powers of ω. $F(\omega)$ is an even function of ω. The magnitude of the frequency response and the pole-zero diagram of $H_a(s)$ is shown in the figure.



The classical analog filters are developed by the following approach. Design the function $F(\omega)$ so that $R(s) = F(s/j)$ can be spectrally factored, then compute a spectral factor to obtain $H_a(s)$. That means, given $R(s)$, find $H_a(s)$ so that

$$R(s) = H_a(s) \cdot H_a(-s).$$

This is the same approach taken in the design of minimum-phase FIR filters where the first step was to design a linear-phase filter with a non-negative frequency response and the second step entails a spectral factorization. This leads to the following problem.
Design a rational function $F(\omega) = |H_a(j\omega)|^2$ such that

1. $F(\omega) \geq 0$
2. $F(\omega)$ is an *even* function of ω.

The classical analog lowpass filters are developed by writing $F(\omega)$ as

$$\boxed{F(\omega) = \frac{1}{1 + \epsilon^2 V(\omega)^2}.} \tag{39}$$

Note that whatever $V(\omega)$ is, $F(\omega)$ will be non-negative. The non-negativity of $F(\omega)$ is built into this expression. $V(\omega)$ does not need to be a non-negative function. Note that if $V(\omega)$ is a rational function, then $F(\omega)$ will also be a rational function. The approach is to design a rational function $V(\omega)$. Note that

1. when $V(\omega)$ is very large, $F(\omega)$ is close to 0.

2. when $V(\omega)$ is very small, $F(\omega)$ is close to 1.

So in the pass-band, $V(\omega)$ should be small, and in the stop-band, $V(\omega)$ should be large.

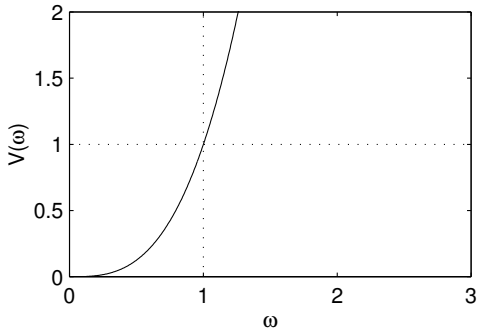
The simplest case is the Butterworth filter. For the Butterworth filter,

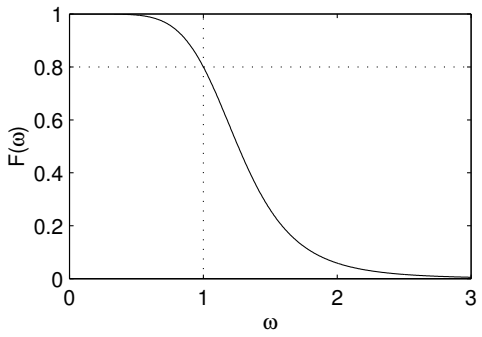
$$V(\omega) = \omega^N. \tag{40}$$

Then

$$\boxed{F(\omega) = \frac{1}{1+\epsilon^2\,\omega^{2N}}.} \tag{41}$$

For example, for $N = 3$ and $\epsilon = 0.5$, the functions $V(\omega)$ and $F(\omega)$ are as shown.





Note that the value of $F(\omega)$ at the frequency $\omega = 1$ is

$$F(1) = \frac{1}{1+\epsilon^2}. \tag{42}$$

For example, when $\epsilon = 0.5$, then

$$F(1) = 1/(1+0.5^2) = 1/(1+1/4) = 1/(5/4) = 0.8. \tag{43}$$

This is indicated by the dotted line in the figure above. Note that $F(\omega)$ is *monotonic* in both the pass-band and the stop-band.

To obtain the transfer function $H_a(s)$, we can first obtain $R(s)$ by,

$$R(s) = F(s/j) = \frac{1}{1+\epsilon^2(s/j)^{2N}} \tag{44}$$

$$= \frac{1}{1+\epsilon^2(s^2/j^2)^N} \tag{45}$$

$$= \frac{1}{1+\epsilon^2(-s^2)^N} \tag{46}$$

To find the poles, set the denominator of $R(s)$ to zero. This will give the roots of $Q(s)\cdot Q(-s)$. Once we find these roots, we can identify the roots of $Q(s)$ by taking those in LHP.

$$1+\epsilon^2(-1)^N s^{2N} = 0 \tag{47}$$

$$s^{2N} = \frac{-1}{\epsilon^2(-1)^N} \tag{48}$$

$$s^{2N} = \frac{-1(-1)^N}{\epsilon^2} \tag{49}$$

When N is **even** this becomes

$$s^{2N} = \frac{-1}{\epsilon^2}. \tag{50}$$

To find an expression for the poles, we can write

$$-1 = e^{j\pi} \tag{51}$$

or

$$-1 = e^{j(\pi+2\pi k)}. \tag{52}$$

This leads to the following chain

$$s^{2N} = \frac{-1}{\epsilon^2} \tag{53}$$

$$= \frac{e^{j(\pi+2\pi k)}}{\epsilon^2} \tag{54}$$

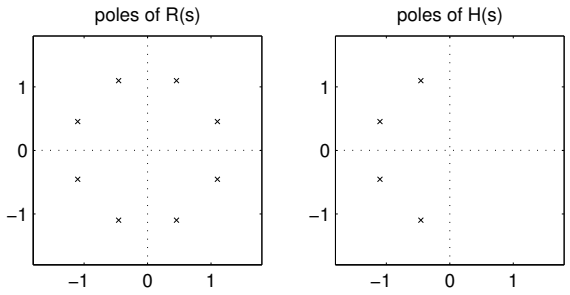
$$s = \left(\frac{e^{j(\pi+2\pi k)}}{\epsilon^2}\right)^{\frac{1}{2N}} \tag{55}$$

$$s = \frac{e^{j(\pi+2\pi k)/(2N)}}{\epsilon^{1/N}} \tag{56}$$

$$s = \frac{e^{j(1+2k)\frac{\pi}{2N}}}{\epsilon^{1/N}} \tag{57}$$

for $0 \leq k \leq 2N-1$.

For example, when $N = 4$, the poles are indicated by the marks in the figure, they are equally spaced on a circle of radius $\frac{1}{\epsilon^{1/N}}$.



When N is **odd**, setting the denominator of $R(s)$ to zero gives

$$s^{2N} = \frac{1}{\epsilon^2}. \tag{58}$$

To find an expression for the poles, we can write

$$1 = e^{j2\pi} \tag{59}$$

or

$$1 = e^{j2\pi k}. \tag{60}$$

Adding an integer multiple of 2π in the exponent leads to the following chain

$$s^{2N} = \frac{1}{\epsilon^2} \tag{61}$$

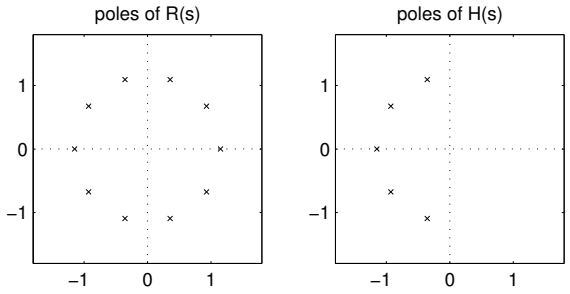
$$= \frac{e^{j2\pi k}}{\epsilon^2} \tag{62}$$

$$s = \left(\frac{e^{j2\pi k}}{\epsilon^2} \right)^{\frac{1}{2N}} \tag{63}$$

$$s = \frac{e^{j\pi k/N}}{\epsilon^{1/N}} \tag{64}$$

for $0 \leq k \leq 2N-1$.

For example, when $N = 5$, the poles are indicated by the marks in the figure, they are equally spaced on a circle of radius $\frac{1}{\epsilon^{1/N}}$.



The roots in the LHP, are given by the following formula, valid when N is either even or odd.

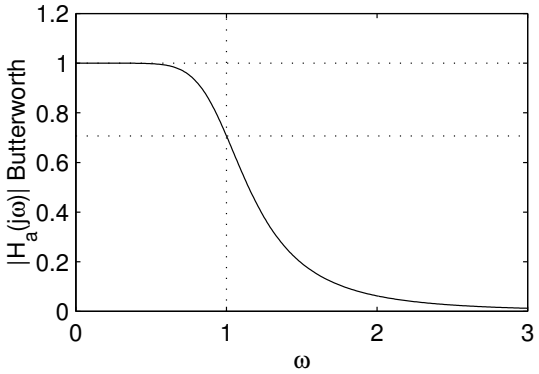
$$s_k = \frac{1}{\epsilon^{1/N}} \cdot e^{j(2k+1+N)\pi/(2N)} \tag{65}$$

for $0 \leq k \leq N-1$. These are the poles of the analog Butterworth filter of order N. The numerator is simply 1 so there are no finite zeros.

The Matlab command `buttap` can be used to obtain the poles of the Butterworth filter of degree N. (This command is part of the Matlab *Signal Processing Toolbox*.) The only input to this command is the filter order N. The result of `buttap` is normalized so that at $\omega = 1$, $|H_a(j\omega)| = 1/\sqrt{2}$.

The following code uses `buttap` to obtain the transfer function of a fourth order Butterworth analog filter and plots $|H_a(j\omega)|$.

```
N = 4;
[z,p,k] = buttap(N);
P = k*poly(z);
Q = poly(p);
w = 0:0.01:3;
H = polyval(P,j*w)./polyval(Q,j*w);
figure(1)
clf
subplot(4,2,1)
plot(w,abs(H),[0 3],[1 1]/sqrt(2),':',[1 1],[0 1.2],':',[0 3],[1 1],':');
axis([0 3 0 1.2])
xlabel('\omega')
ylabel('|H_a(j\omega)| Butterworth')

orient tall
print -deps butex2
```



The Chebyshev-I analog filter is based on the Chebyshev polynomials $C_N(\omega)$. For the Chebyshev-I filter,

$$V(\omega) = C_N(\omega). \tag{66}$$

Then

$$\boxed{F(\omega) = \frac{1}{1 + \epsilon^2\, C_N(\omega)^2}.} \tag{67}$$

The remarkable Chebyshev polynomials can be generated by the following recursive formula.

$$C_0(\omega) := 1 \tag{68}$$

$$C_1(\omega) := \omega \tag{69}$$

$$C_{k+1}(\omega) := 2\,\omega\, C_k(\omega) - C_{k-1}(\omega). \tag{70}$$

The next few $C_k(\omega)$ are

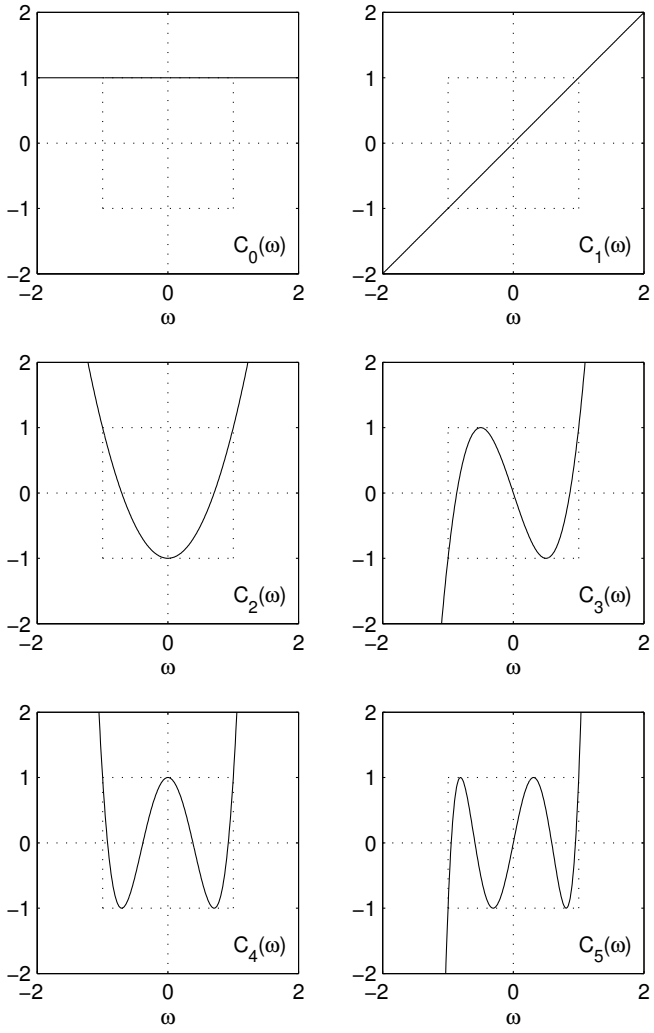
$$C_2(\omega) = 2\,\omega^2 - 1 \tag{71}$$

$$C_3(\omega) = 4\,\omega^3 - 3\,\omega \tag{72}$$

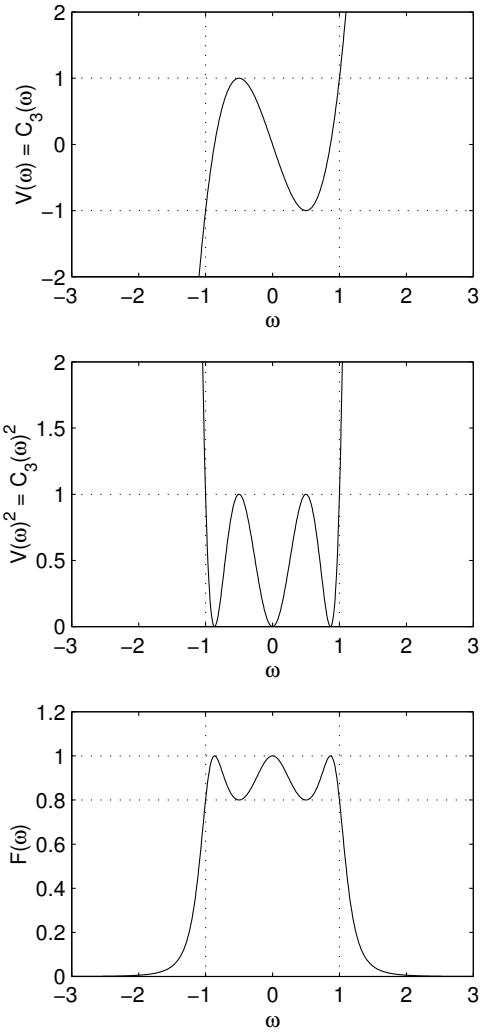
$$C_4(\omega) = 8\,\omega^4 - 8\,\omega^2 + 1 \tag{73}$$

$$C_5(\omega) = 16\,\omega^5 - 20\,\omega^3 + 5\,\omega. \tag{74}$$

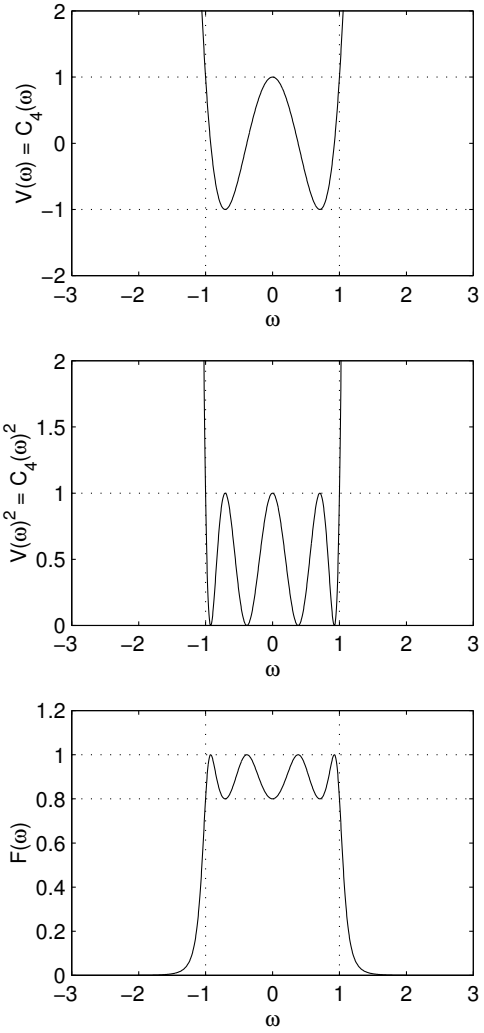
As can be seen in the following figures, in the interval $-1 \leq \omega \leq 1$, the Chebyshev polynomial oscillates between -1 and 1. This will create a equiripple behavior in the pass-band of the resulting analog filter.


C0(ω)
C1(ω)
C2(ω)
C3(ω)
C4(ω)
C5(ω)
ω

For example, when $N = 3$ and $\epsilon = 0.5$, then the functions $V(\omega)$ and $F(\omega)$ are as shown.



When $N = 4$ and $\epsilon = 0.5$, then the functions $V(\omega)$ and $F(\omega)$ are as shown.



The Chebyshev-I analog filter has no finite zeros.

Skipping the derivation, the poles of the Chebyshev-I filter lie at

$$s_k = -\sinh(v)\,\sin\left(\frac{(2k+1)\pi}{2N}\right) + j\,\cosh(v)\,\cos\left(\frac{(2k+1)\pi}{2N}\right) \tag{75}$$

for $0 \leq k \leq N-1$ where

$$v = \frac{\sinh^{-1}(1/\epsilon)}{N}. \tag{76}$$

The frequency response of the Chebyshev-I analog filter is equiripple in the pass-band, and monotonic in the stop-band.

The Matlab command `cheb1ap` can be used to obtain the poles of the Chebyshev-I filter of degree N. The input arguments of this command are the filter degree N and the pass-band ripple size R_p in dB. For this filter $|H_a(j\omega)|$ lies between the bounds 1 and $1-\delta_p$ in the pass-band. The value R_p is related to δ_p by

$$1 - \delta_p = 10^{-R_p/20} \tag{77}$$

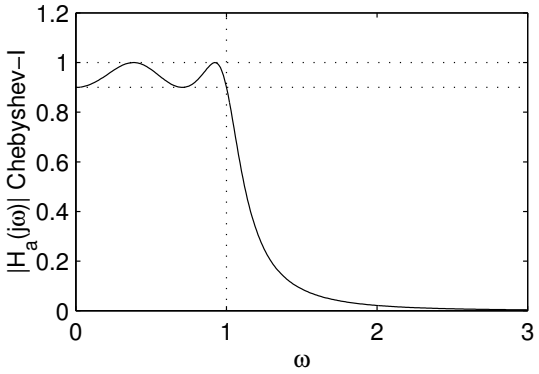
or

$$R_p = -20 \log_{10}(1 - \delta_p). \tag{78}$$

For example, to obtain a Chebyshev-I filter of degree N such that $|H_a(j\omega)|$ lies between 0.9 and 1 in the pass-band, we can use `cheb1ap` with

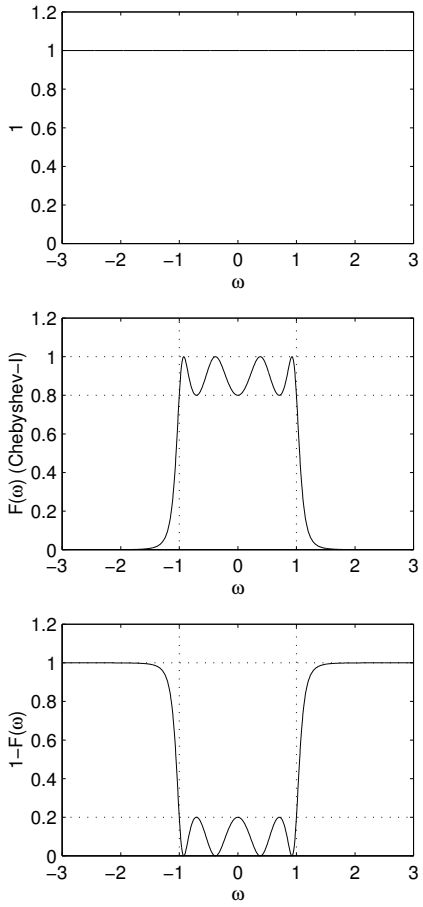
$$R_p = -20 \log_{10}(0.9). \tag{79}$$

The code on the next page uses `cheb1ap` to obtain the transfer function of a fourth order Chebyshev-I analog filter and plots $|H_a(j\omega)|$.

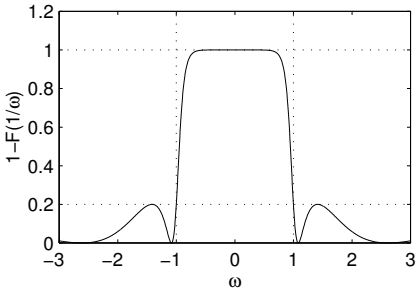


```
N  = 4;
dp = 0.1;
Rp = -20*log10(1-dp);
[z,p,k] = cheb1ap(N,Rp);
P = k*poly(z);
Q = poly(p);
w = 0:0.01:3;
H = polyval(P,j*w)./polyval(Q,j*w);
figure(1)
clf
subplot(4,2,1)
plot(w,abs(H),[0 3],[1 1]*(1-dp),':',[1 1],[0 1.2],':',[0 3],[1 1],':');
axis([0 3 0 1.2])
xlabel('\omega')
ylabel('|H_a(j\omega)| Chebyshev-I')

orient tall
print -deps chebex2
```

The Chebyshev-II analog filter (also called the inverse-Chebyshev filter) is designed so as to have a monotonic pass-band and an equiripple stop-band. This type of frequency response can be obtained by a two step procedure. First, subtract the Chebyshev-I $F(\omega)$ from 1. This is illustrated in the following figure.



Second, perform the change of variables $\omega \leftarrow 1/\omega$. This results in the Chebyshev-II frequency response.



As the figure shows, the frequency response has a monotonic pass-band and an equiripple stop-band. According to the two step procedure, the formula for $F(\omega)$ for the Chebyshev-II filter is given by

$$F(\omega) = 1 - \frac{1}{1 + \epsilon^2\, C_N(1/\omega)^2} \tag{80}$$

or

$$\boxed{F(\omega) = \frac{\epsilon^2\, C_N(1/\omega)^2}{1 + \epsilon^2\, C_N(1/\omega)^2}.} \tag{81}$$

The Chebyshev-II filter has zeros as the numerator is not just 1. It turns out that all of its zeros lie on the imaginary axis. Skipping the details, the zeros are given by

$$z_k = \frac{j}{\cos((2k+1)\pi/(2N))} \tag{82}$$

for $0 \leq k \leq N-1$. Surprisingly, the poles of the Chebyshev-II analog filter are the reciprocals of the poles of the Chebyshev-I analog filter.

The Matlab command `cheb2ap` can be used to obtain the poles and zeros of the Chebyshev-II filter of degree N. The input arguments of this command is the filter degree N and the stop-band ripple size R_s in dB. For this filter $|H_a(j\omega)|$ lies between the bounds 0 and δ_s in the stop-band. The value R_s is related to δ_s by

$$\delta_s = 10^{-R_s/20} \tag{83}$$

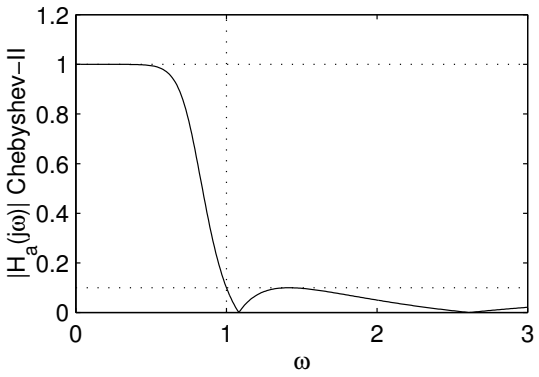
or

$$R_s = -20\ \log_{10}(\delta_s). \tag{84}$$

For example, to obtain a Chebyshev-II filter of degree N such that $|H_a(j\omega)|$ lies between 0 and 0.1 in the stop-band, we can use `cheb2ap` with

$$R_s = -20\ \log_{10}(0.1). \tag{85}$$

The code on the next pages uses `cheb2ap` to obtain the transfer function of a fourth order Chebyshev-II analog filter and plots $|H_a(j\omega)|$.

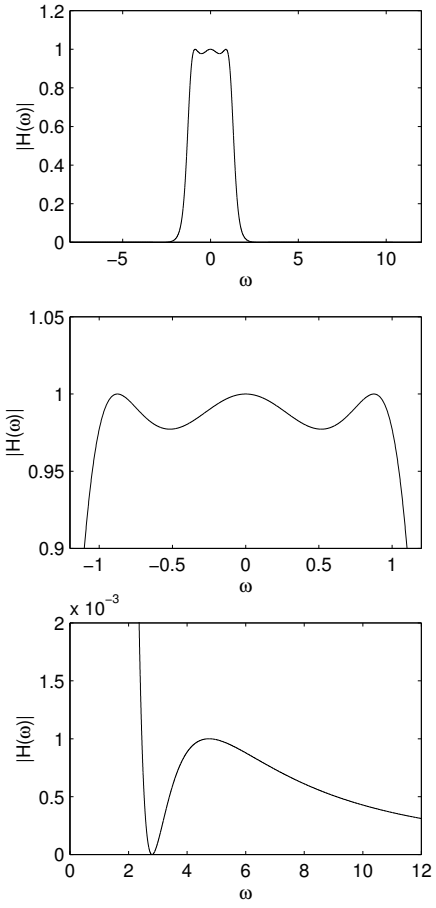


```
N  = 4;
ds = 0.1;
Rs = -20*log10(ds);
[z,p,k] = cheb2ap(N,Rs);
P = k*poly(z);
Q = poly(p);
w = 0:0.01:3;
H = polyval(P,j*w)./polyval(Q,j*w);
figure(1)
clf
subplot(4,2,1)
plot(w,abs(H),[0 3],[1 1]*ds,':',[1 1],[0 1.2],':',[0 3],[1 1],':');
axis([0 3 0 1.2])
xlabel('\omega')
ylabel('|H_a(j\omega)| Chebyshev-II')

orient tall
print -deps ichebex2
```

The elliptic analog filter (also called the Cauer filter) is equi-ripple in both the pass-band and the stop-band. For a given set of error tolerances, the elliptic filter gives the minimal-degree filter. For the elliptic filter, $V(\omega)$ is the *Chebyshev rational function* $R_N(\omega, \alpha)$ which depends on a parameter α, so

$$\boxed{F(\omega) = \frac{1}{1 + \epsilon^2 \, R_N(\omega, \alpha)^2}.} \tag{86}$$

The functional form for $V(\omega)$ is quite complicated as it is based on elliptic functions. As elliptic functions are difficult to apply to other design problems, we will skip the details here. An elliptic analog filter is illustrated in the following figure.



The Matlab command `ellipap` can be used to obtain the poles and zeros of the elliptic filter of degree N. The input arguments of this command is the filter degree N, the pass-band ripple size R_p in dB, and the stop-band ripple size R_s in dB. The magnitude of the frequency response of the elliptic filter $|H_a(j\omega)|$ will lie within the bounds $1-\delta_p$ and 1 in the pass-band, and it will lie within the bounds 0 and δ_s in the stop-band. The values R_p and R_s is related to the pass-band ripple by

$$\delta_p = 1 - 10^{-R_p/20} \tag{87}$$

$$\delta_s = 10^{-R_s/20} \tag{88}$$

or

$$R_p = -20\ \log_{10}(1-\delta_p) \tag{89}$$

$$R_s = -20\ \log_{10}(\delta_s). \tag{90}$$

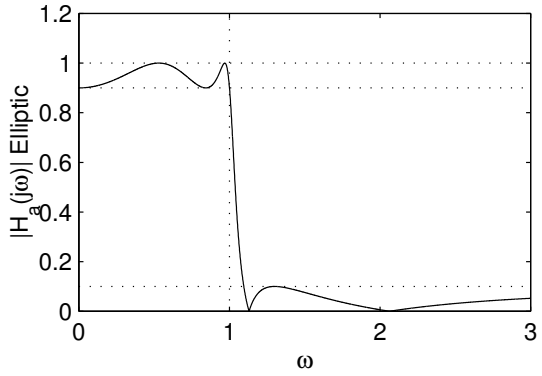
For example, to obtain an elliptic filter of degree N such that $|H_a(j\omega)|$ lies between 0.9 and 1 in the pass-band, and 0 and 0.1 in the stop-band, then we can use `ellipap` with

$$R_p = -20\ \log_{10}(0.9) \tag{91}$$

$$R_s = -20\ \log_{10}(0.1). \tag{92}$$

The following code uses `ellipap` to obtain the transfer function of a fourth order elliptic analog filter and plots $|H_a(j\omega)|$.

```
N  = 4;
ds = 0.1;
dp = 0.1;
Rs = -20*log10(ds);
Rp = -20*log10(1-dp);
[z,p,k] = ellipap(N,Rp,Rs);
P = k*poly(z);
Q = poly(p);
w = 0:0.001:3;
H = polyval(P,j*w)./polyval(Q,j*w);
figure(1)
clf
subplot(4,2,1)
plot(w,abs(H),[0 3],[1 1]*ds,':',[0 3],[1 1],':',[1 1],[0 1.2],':',[0 3],[1 1]*(1-
axis([0 3 0 1.2])
xlabel('\omega')
ylabel('|H_a(j\omega)| Elliptic')

orient tall
print -deps ellipex2
```



The four classical analog filters described in the previous sections were presented in *normalized* form. That means a band-edge is at the frequency $\omega = 1$. For the Chebyshev-I and elliptic filters, as presented, the *pass*-band edge was at $\omega = 1$. For the Chebyshev-II filter as it was presented, the *stop*-band edge was at $\omega = 1$. On the other hand, the Butterworth filter was normalized so that $|H_a(j\omega)| = 1/\sqrt{2}$.

To change the band-edges, one can simply scale the variable ω by a constant. If $H_a(j\omega)$ has its pass-band edge at $\omega = 1$, then $G_a(j\omega) = H_a(j\omega/\omega_c)$ has its pass-band edge at $\omega = \omega_c$ for example. The transfer function $G_a(s)$ is likewise given by $G_a(s) = H_a(s/\omega_c)$.

This scaling can be done using the poles, zeros, and gain factor. If one writes $H_a(s)$ as

$$H_a(s) = k \cdot \frac{\prod_{i=1}^{M}(s - z_i)}{\prod_{i=1}^{N}(s - p_i)} \tag{93}$$

then

$$G_a(s) = H_a(C\,s) \tag{94}$$

$$= k \cdot \frac{\prod_{i=1}^{M}(C\,s - z_i)}{\prod_{i=1}^{N}(C\,s - p_i)} \tag{95}$$

$$= k \cdot \frac{\prod_{i=1}^{M} C\,(s - z_i/C)}{\prod_{i=1}^{N} C\,(s - p_i/C)} \tag{96}$$

$$= k \cdot C^{M-N} \cdot \frac{\prod_{i=1}^{M}(s - z_i/C)}{\prod_{i=1}^{N}(s - p_i/C)} \tag{97}$$

So, given the zeros z_i, poles p_i, and gain factor k of a prototype transfer function $H_a(s)$, the transfer function $G_a(s) = H(C\,s)$ has the zeros, poles and gain factor given by

$$z_i' = z_i/C \tag{98}$$

$$p_i' = p_i/C \tag{99}$$

$$k' = k \cdot C^{M-N} \tag{100}$$

Filter type	Characteristic	Matlab function
Butterworth	$\|H(j\omega)\|\|_{\omega=1} = 1/\sqrt{2}$	`buttap(N)`
Chebyshev-I	$1-\delta_p \leq \|H(j\omega)\| \leq 1$ for $\|\omega\| \leq 1$	`cheb1ap(N,Rp)`
Chebyshev-II	$\|H(j\omega)\| \leq \delta_s$ for $\|\omega\| \geq 1$	`cheb2ap(N,Rs)`
Elliptic	$1-\delta_p \leq \|H(j\omega)\| \leq 1$ for $\|\omega\| \leq 1$; $\|H(j\omega)\| \leq \delta_s$ for $\|\omega\| \geq \omega_s$	`ellipap(N,Rp,Rs)`

$$\delta_p = 1 - 10^{-R_p/20}$$
$$\delta_s = 10^{-R_s/20}$$

$$R_p = -20\log_{10}(1-\delta_p)$$
$$R_s = -20\log_{10}(\delta_s)$$

For the elliptic filter, the stop-band edge ω_s is determined by N and the specified values of δ_p and δ_s.

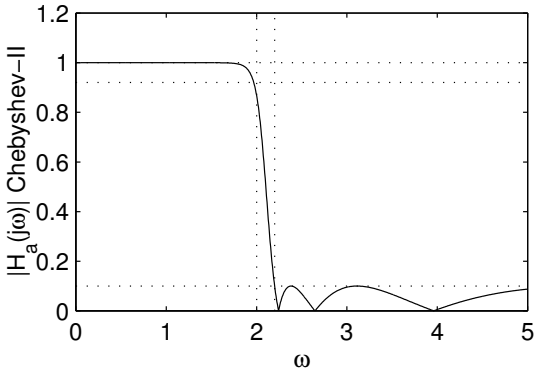
Problem: Design an analog Chebyshev-II of minimal degree meeting the specifications:

$$0.92 \leq |H_a(j\omega)| \leq 1 \qquad \text{for} \quad |\omega| \leq 2 \tag{101}$$

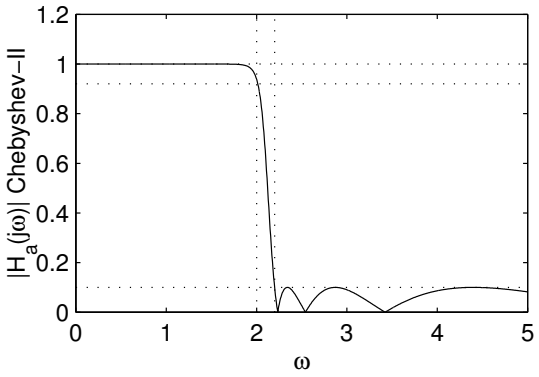
$$|H_a(j\omega)| \leq 0.1 \qquad \text{for} \quad |\omega| \geq 2.2 \tag{102}$$

To find the solution to this problem, note that the analog Chebyshev-II prototype has a stop-band edge at $\omega = 1$. Therefore, we will need to scale the prototype with the change of variables $s \leftarrow s/2.2$. That is, we define $G_a(s) = H_a(C\,s)$ where $H_a(s)$ is the Chebyshev-II prototype and the constant C is $1/(2.2)$.

Then the Chebyshev-II filter of degree 8, after rescaling, is shown in the following figure. It can be seen from the figure that it does not meet the pass-band edge.



However, the Chebyshev-II filter of degree 9, after rescaling, is shown in the following figure. It can be seen from the figure that it does meet the specifications.



This filter was obtained and the plot was generated with the following Matlab code.

```
N  = 9;                          % number of poles
dp = 0.08;                       % desired pass-band error margin
ds = 0.1;                        % desired stop-band error margin
wp = 2.0;                        % pass-band edge
ws = 2.2;                        % stop-band edge
Rs = -20*log10(ds);
[z,p,k] = cheb2ap(N,Rs);         % analog filter prototype
M  = length(z);                  % number of zeros
C  = 1/ws;                       % constant for scaling poles and zeros
k  = k*C^(M-N);                  % modified gain factor
P  = k*poly(z/C);                % modified numerator
Q  = poly(p/C);                  % modified denominator
w  = 0:0.005:5;
H  = polyval(P,j*w)./polyval(Q,j*w);
figure(1)
subplot(4,2,1)
plot(w,abs(H),[0 5],[1 1]*ds,':',ws*[1 1],[0 1.2],':',[0 5],[1 1],':',...
              wp*[1 1],[0 1.2],':',[0 5],[1 1]*(1-dp),':');
axis([0 5 0 1.2])
xlabel('\omega')
ylabel('|H_a(j\omega)| Chebyshev-II')
```

The bilinear transform (BLT) can be used to convert an analog filter into a digital filter. The BLT consists of the change of variables

$$\boxed{s = C \cdot \frac{z-1}{z+1}.} \tag{103}$$

We will use $C = 1$ in the following.

Let $H_a(s)$ denote the transfer function of an analog filter. Then define

$$H(z) = H_a(s)\big|_{s=\frac{z-1}{z+1}} ,$$

that is:

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

If $H_a(s)$ is a rational function of s, the $H(z)$ will be a rational function of z. Therefore it represents a finite order system (a finite order difference equation).

Question: What is the frequency response of $H(z)$?

$$H(e^{j\omega}) = H_a\left(\frac{e^{j\omega}-1}{e^{j\omega}+1}\right)$$

Let us simplify the term $\frac{e^{j\omega}-1}{e^{j\omega}+1}$.

To simplify it, we use the same 'trick' used when we dealt with linear-phase FIR filter — extract the 'center' frequency.

That means we write it as

$$\frac{e^{j\omega}-1}{e^{j\omega}+1} = \frac{e^{-j\frac{\omega}{2}}}{e^{-j\frac{\omega}{2}}} \cdot \frac{e^{j\omega}-1}{e^{j\omega}+1} \tag{104}$$

$$= \frac{e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}}{e^{j\frac{\omega}{2}} + e^{-j\frac{\omega}{2}}} \tag{105}$$

$$= \frac{2j\sin(\frac{\omega}{2})}{2\cos(\frac{\omega}{2})} \tag{106}$$

$$= j\tan\left(\frac{\omega}{2}\right) \tag{107}$$

Therefore

$$H(e^{j\omega}) = H_a\left(j\tan\left(\frac{\omega}{2}\right)\right) = H_a(j\Omega)$$

where $H_a(j\Omega)$ is the frequency response of the analog filter.

$H(e^{j\omega})$ is a *warped*, or *compressed*, version of $H_a(j\Omega)$.

$$\boxed{\Omega = \tan\left(\frac{\omega}{2}\right)} \tag{108}$$

Plot of Ω versus ω.

Fig 7.4 in Mitra

The analog filter has a frequency response which is defined for

$$0 \leq \Omega < \infty$$

(and $\Omega < 0$ too).

The digital filter frequency response only needs to be defined for

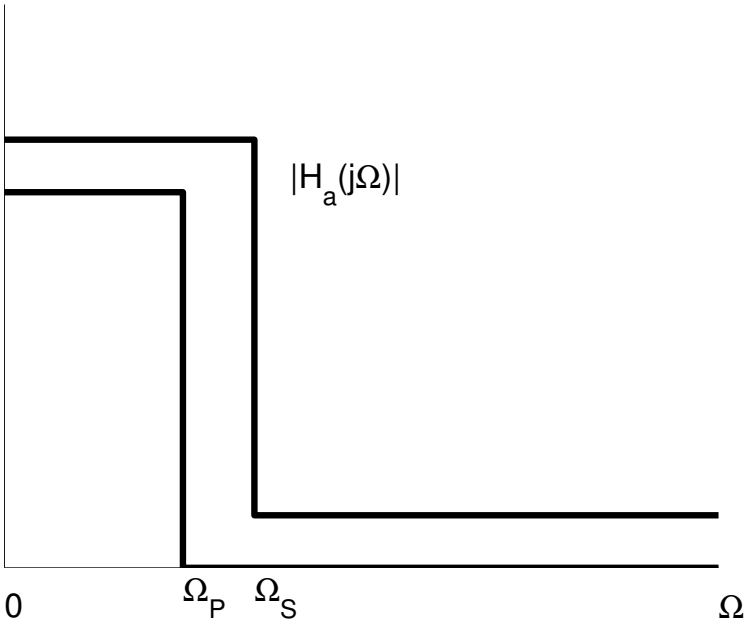
$$0 \leq \omega \leq \pi$$

(and $-\pi \leq \omega \leq 0$ too).

($H(e^{j\omega})$ is periodic beyond π).

So the infinite interval is compressed into the finite interval — some warping must occur. (Fig 7.5 in Mitra.)

If $|H_a(j\Omega)|$ fits into the template:



then what template does the digital filter fit into?

$$|H(e^{j\omega})| = |H_a\left(j\tan\left(\frac{\omega}{2}\right)\right)|$$

What is the pass-band edge and stop-band edge of the digital filter frequency response?

$$|H(e^{j\omega_p})| = |H_a\left(j\tan\left(\frac{\omega_p}{2}\right)\right)| =: |H_a(j\Omega_p)|$$

so

$$\boxed{\tan\left(\frac{\omega_p}{2}\right) = \Omega_p} \tag{109}$$

similarly,

$$\boxed{\tan\left(\frac{\omega_s}{2}\right) = \Omega_s} \tag{110}$$

If you want a digital filter with passband and stopband edges ω_p and ω_s, then

1. Design an analog filter with pass-band edge

   $$\Omega_p = \tan(\frac{\omega_p}{2})$$

   and stop-band edge

   $$\Omega_s = \tan(\frac{\omega_s}{2})$$

2. Convert the analog filter to a digital one using the BLT

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

Question: How to find $H(z)$ conveniently from $H_a(s)$?
Use poles and zeros!

$$H(z_o) = 0 \qquad H_a\left(\frac{z_o - 1}{z_o + 1}\right) = 0$$

Setting

$$s_o = \frac{z_o - 1}{z_o + 1}$$

gives

$$H_a(s_o) = 0.$$

So if s_o is a zero of $H_a(s)$, then a zero of $H(z)$ can be obtained by solving

$$s_o = \frac{z_o - 1}{z_o + 1}.$$

$$s_o = \frac{z_o - 1}{z_o + 1} \tag{111}$$

$$s_o\,(z_o + 1) = z_o - 1 \tag{112}$$

$$z_o\,s_o + s_o = z_o - 1 \tag{113}$$

$$z_o\,(s_o - 1) = -s_o - 1 \tag{114}$$

$$z_o = \frac{s_o + 1}{1 - s_o} \tag{115}$$

so

$$\boxed{z_o = \frac{s_o + 1}{1 - s_o}}$$

If s_o is a zero of $H_a(s)$, then $\frac{s_o+1}{1-s_o}$ is a zero of $H(z)$. Likewise, if s_o is a *pole* of $H_a(s)$, then $\frac{s_o+1}{1-s_o}$ is a *pole* of $H(z)$.

So to find $H(z)$ from $H_a(s)$, find the poles and zeros of $H_a(s)$ and convert to poles and zeros of $H(z)$, then get the numerator and denominator coefficicients of $H(z)$ from its poles and zeros.

Note: due to the warping of the frequency axis, the behavior of the frequency response can be affected.

However, when $|H_a(j\Omega)|$ is piecewise constant, then after the BLT, the piecewise constant behavior is preserved.

If $D(\omega)$ is not made of constant segments, then the BLT can destroy the shape of the frequency response.

REMARKS ON THE BLT

1. The order of $H(z)$ and $H_a(s)$ are the same. (The order is the number of poles, or the number of zeros, which ever is greater.)
2. The left half plane (LHP) is mapped to the unit disk $|z| \leq 1$
3. The imaginary axis is mapped to the unit circle $|z| = 1$.
4. Because the LHP is mapped to the disk, poles in the LHP are mapped to the inside of the unit circle. *Therefore, a causal stabel ananog system is converted by the BLT into a stable causal digital system.*
5. The optimality of the Chebyshev (minimax) approximation to piecewise constant $D(\omega)$ is preserved (in magnitude frequency response sense).

IMPULSE-INVARIANCE METHOD

Another well-known method for analog to digital filter conversion is the *impulse-invariance* method.

1. $h(n) = h_a(nT)$ for $0 \leq n \leq N-1$. The impulse response of the digital filter is matched to the samples of the impulse response of the analog filter.
2. It does not preserve the shape of $H_a(j\Omega)$.
3. It does not always preserve the stability of causal systems like the BLT does.

The impulse-invariance method is not usually as suitable for IIR digital filter design as the BLT is, but it is usually OK for *simulation*.