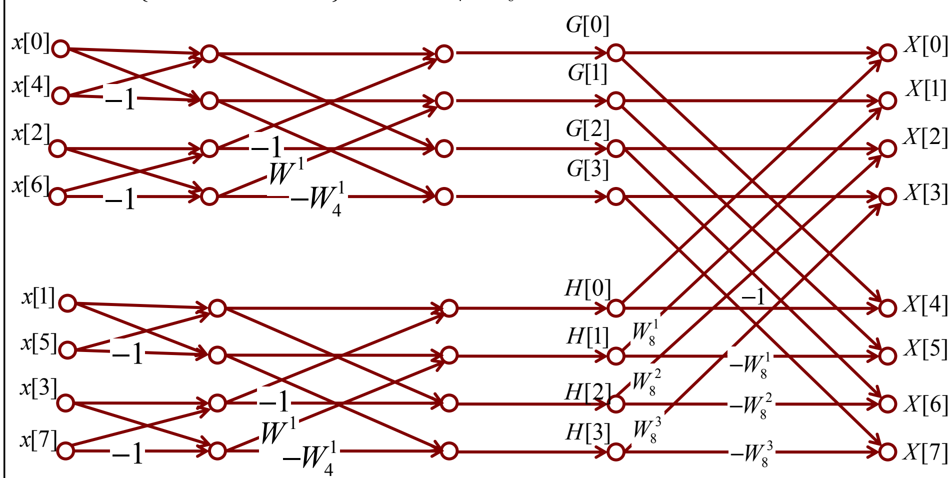
Work the 8-pt DIT algorithm

$$x[n] = \left\{ \overset{\Downarrow}{1}, 1, -1, -1, 1, 1, -1, -1 \right\}$$

$$W_8^1 = e^{-j2\pi/8} = e^{-j\pi/4} = 0.707 - j0.707$$

$$W_4^1 = W_8^2 = e^{-j2\pi/4} = e^{-j\pi/2} = -j$$



Bit reversed order

- **From this we have that the DIT FFT algorithm must reorder the inputs…**
- **…so that the outputs show up in the correct order.**
- **The decimation in frequency (DIF) FFT algorithm keeps the inputs in their natural (index) order…**
- **…and reorders the outputs *in exactly the same way!***
- **We call this reindexing *bit reversed order***

Binary	LSB on rt	LSB on LF
000	0	0
001	1	4
010	2	2
011	3	6
100	4	1
101	5	5
110	6	3
111	7	7

The other radix 2 algorithm

- **The other "famous" radix 2 algorithm is "decimation in frequency"**

$$X_{DFT}[k] = \underbrace{\sum_{n=0}^{N/2-1} x[n]W_N^{nk}}_{\text{first half of input data}} + \underbrace{\sum_{n=0}^{N/2-1} x[n+N/2]W_N^{(n+N/2)k}}_{\text{second half of input data}}$$

$$= \sum_{n=-}^{N/2-1} \left(x[n] + W_N^{(N/2)k}x[n+N/2]\right)W_N^{nk}$$

$$= \sum_{n=0}^{N/2-1} \left(x[n] + (-1)^k x[n+N/2]\right)W_N^{nk}$$

$$k \text{ even: } X[k] = \sum_{n=0}^{N/2-1} \left(x[n] + x[n+N/2]\right)W_{N/2}^{nk}$$

$$k \text{ odd: } X[k] = \sum_{n=0}^{N/2-1} \left(x[n] - x[n+N/2]\right)W_N^n W_{N/2}^{nk}$$

Now the dependence is on the *output index* being even or odd

...and we still have a nice

$$A \pm W_N^p B$$

form

$$X[2k] = \sum_{n=0}^{N/2-1} \left(x[n] + x[n+N/2]\right)W_{N/2}^{nk}$$

$$= DFT\left(x[n] + x[n+N/2]\right), k = 0,1,2,\ldots N/2-1$$

$$X[2k+1] = \sum_{n=0}^{N/2-1} \left(x[n] - x[n+N/2]W_N^k\right)W_{N/2}^{nk}$$

$$= DFT\left(\left(x[n] - x[n+N/2]\right)W_N^k\right), k = 0,1,2,\ldots N/2-1$$

$$\text{Let } g[n] = x[n] + x[n+N/2]$$

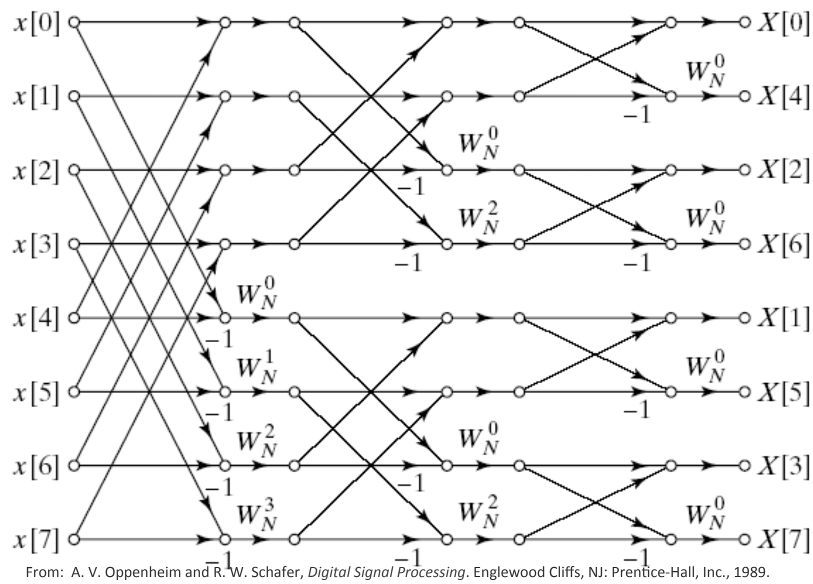
$$\text{Let } h[n] = \left(x[n] - x[n+N/2]\right)W_N^k$$

$$X[2k] = G[k]$$

$$X[2k+1] = H[k]$$

- **...and this, too has a butterfly structure**

DIF Butterfly structure



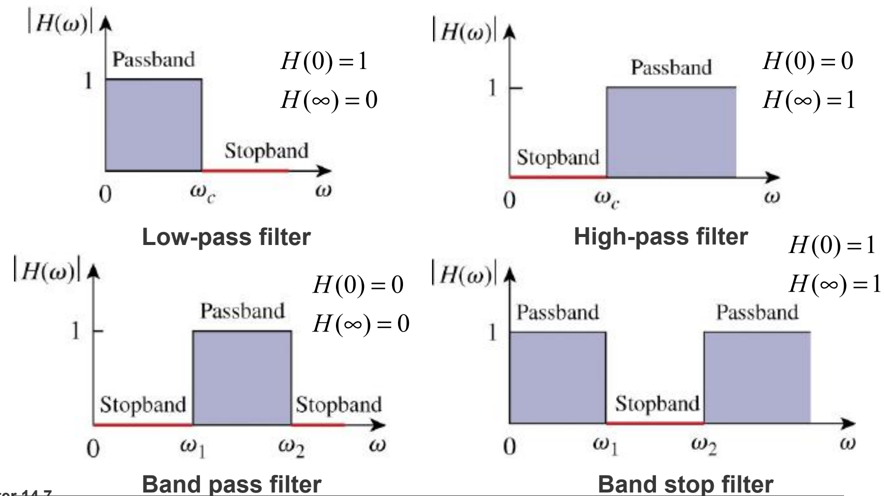
From: A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989.

CMPE323: Communication Systems

Lecture 22 Analog Filter Concepts

Filters

A filter is a circuit that is designed to pass signals with desired frequencies and reject or attenuate others



Chapter ~~14.7~~

Where we want to go?

- There are a variety of analog filter design techniques that are well-understood…
- …and permit easy design of s-domain transfer functions with specific amplitude and/or phase characteristics.
- We want to use these techniques to design a prototype analog filter…
- Eventually CMPE422 will then apply one of several techniques to transform the analog filter into a digital implementation.
- So we need to start with the analog filter designs

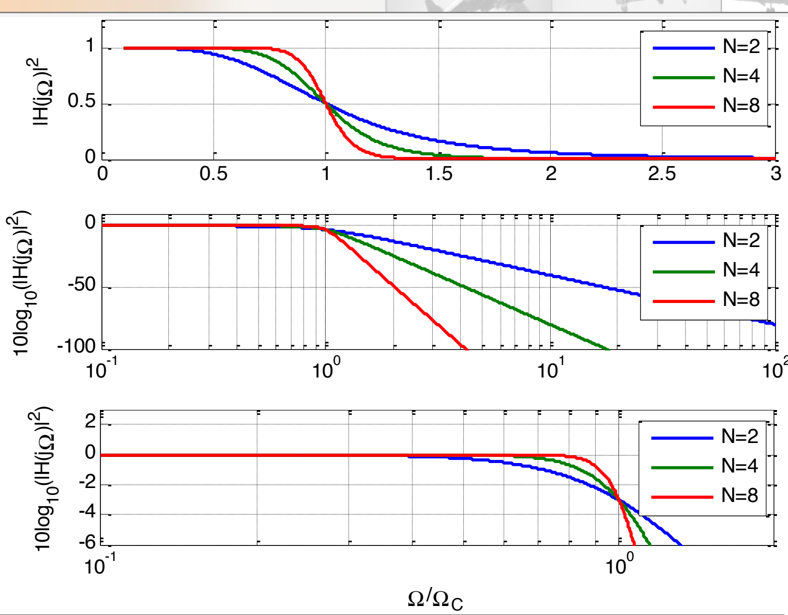
A Butterworth Filter

- The **Butterworth** filter is an analog design has a frequency amplitude response that is **maximally flat** in the passband
- First described by S. Butterworth "On the Theory of Filter Amplifiers", *Wireless Engineer,* vol. 7, 1930 (!)
- "Maximally flat" means that, for an N pole lowpass filter, the first $2N-1$ derivatives of $|H(j\omega)|^2$ are zero at $\omega = 0$ rad/sec.
- Butterworth lowpass filters are **monotonically decreasing** in both the passband and the stopband.
- The filter response satisfies

$$|H(j\omega)|^2 = \frac{1}{1+\left(j\omega / j\omega_C\right)^{2N}} = \frac{1}{1+\left(\omega / \omega_C\right)^{2N}}$$

- The filter is described by 2 parameters

$N =$ number of poles, ω_C cutoff frequency



Look at the squared magnitude

$$\left|H(s)\right|^2 = \frac{1}{1+\left(s/j\omega_C\right)^{2N}} = H(s)H(-s) = H(j\omega)H(-j\omega)$$

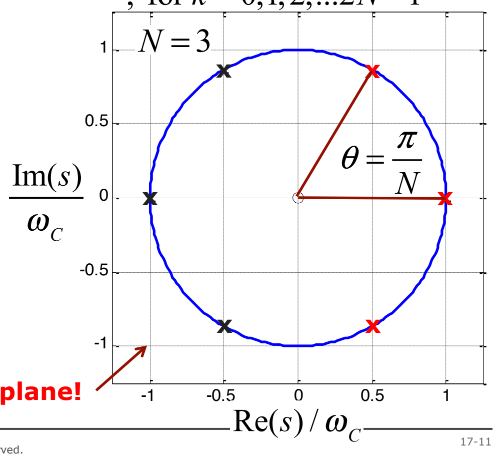
The poles of $\left|H(s)\right|^2$ are the roots of $1+\left(s/j\omega_C\right)^{2N} = 0$

$s_k = (-1)^{1/2N} j\omega_C = \omega_C e^{(j\pi/2N)(2k+N-1)}$, for $k = 0,1,2,\ldots 2N-1$

A pole of $\left|H(s)\right|^2$ at s_k

$\Leftrightarrow$ a pole at $-s_k$

The poles of $H(s)$ are the N poles with $\mathrm{Re}(s_k) < 0$ for stability.



s-plane, not the z-plane!

Example

- **3 pole Butterworth low pass with** $\omega_C = 2\pi\times 10^3$ rad/sec

k	$2k+N-1$	$e^{j\pi\frac{2k+N-1}{2N}}$	Real Part
0	2	$e^{j\pi/3}$	>0
1	4	$e^{j2\pi/3}$	<0
2	6	$e^{j\pi} = -1$	<0
3	8	$e^{j4\pi/3} = e^{-j2\pi/3}$	<0
4	10	$e^{j5\pi/3} = e^{-j\pi/3}$	>0
5	12	$e^{j2\pi} = 1$	>0

$$H(s) = \frac{\omega_C^3}{\left(s+\omega_C\right)\left(s-\omega_C e^{-j2\pi/3}\right)\left(s-\omega_C e^{j2\pi/3}\right)}$$

Estimating the number of poles

- Typically we have a desired 3-dB bandwidth Ω_C
- ...and a desired stop band rejection R at a stop band frequency Ω_S
- The Butterworth filter has out-of-band slope of

  $6N$ dB per octave, or $20N$ dB/decade

- If $[R]_{dB}$ is given in dB (>0), then

$$6N\frac{\log_{10}\left(\frac{\Omega}{\Omega_C}\right)}{\log_{10}(2)} < [R]_{dB} \text{ or } N = \left\lceil \frac{0.05[R]_{dB}}{\log_{10}\left(\frac{\Omega}{\Omega_C}\right)} \right\rceil$$

Analog Butterworth Design Method (Low Pass)

- Given the pass-band amplitude $A_P > 0$ (in dB) and pass-band corner frequency f_P (in Hz or rad/sec)...
- ...and the stop-band amplitude $A_S > 0$ (in dB) and stop band corner frequency f_S
- Process
  - 1) Compute $v_S = f_S / f_P$ (for LP), f_P / f_S (for HP)
  - 2) Compute $\varepsilon^2 = 10^{A_P/10} - 1$
  - 3) Compute

Natural log (ln)

$$N = \left\lceil \frac{0.5\log\left((10^{A_S/10} - 1)/\varepsilon^2\right)}{\log(v_S)} \right\rceil = \text{number of poles}$$

Or use Previous if $A_P = 3$ dB

  - 4) Compute $v_3 = (1/\varepsilon)^{1/N} = f_3 / f_p$
  - 4) Compute $\theta_k = (2k + N - 1)\pi / (2N),\ s_k = v_3 \exp(-j\theta_k)$
  - 5) Compute $\omega_C^N = \prod_{k=0}^{N-1} s_k$

You try

- **Design an Butterworth filter lowpass filter with a -3 dB passband of 75 kHz and rejection of at least 60 dB at 825 kHz. Define the passband, the transition band, the stop band, the number of poles and the pole locations. Write the form of the squared magnitude of the transfer function.**

Short way:

$$N = \left\lceil \frac{0.05\left[R\right]_{dB}}{\log_{10}\left(\frac{\Omega}{\Omega_C}\right)} \right\rceil = \left\lceil \frac{0.05\left[60\right]_{dB}}{\log_{10}\left(\frac{825}{75}\right)} \right\rceil = 3 \text{ Poles}$$

Long Way

```
CMPE422_S11 Lecture 20 modified 4/13/2011 EFCL
For fp of 75000 and fs of 825000 nu_s = 11
with Ap = 3 dB and stopband amplitude As = 60 dB
eps^2 = 0.99526
Number of Butterworth poles is: 3
Theory normalized poles:
Pole 1 = -0.5-0.86603i = 1 with angle -0.66667*pi
Pole 2 = -1-1.2246e-016i = 1 with angle -1*pi
Pole 3 = -0.5+0.86603i = 1 with angle 0.66667*pi
Theory unnormalized poles:
Omega_c: 471238.898 rad/sec = 75000 Hz
Squared_magnitude of normalized poles is: 1 sqrt is 1
The poles are :
Pole 1 = -471238.898 = 471238.898 with angle 1*pi
Pole 2 = -235619.449+408104.857i = 471238.898 with angle 0.66667*pi
Pole 3 = -235619.449-408104.857i = 471238.898 with angle -0.66667*pi
omega_c = 471238.898 rad/sec  = 75 kHz
-3 dB corner of Butterworth is 471400.0155 rad/sec = 75.0256 kHz
-60 dB corner of Butterworth is 4715018.0236 rad/sec = 750.4184 kHz
```

Chebyshev Filters

- Chebyshev filters relax the shape requirements
  - Do not require monotonic in passband, or,
  - Do not require monotonic in the stop band
- …and impose a new requirement
  - Type I: Maximum ripple (i.e. variation) in the pass band, or,
  - Type II: Maximum ripple (i.e. variation) in the stop band

  - Type I filters are far more common!

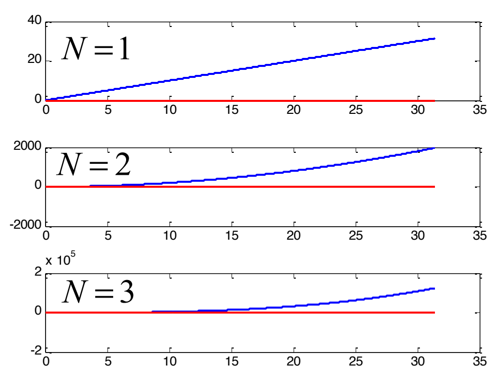
17-17

Type I Chebyshev filters

- Chebyshev filters depend on Chebyshev polynomials

$$V_N(x) = \cos\left(N\cos^{-1}(x)\right)$$

- For $x \le 2\pi,\ V_N(x)^2 \le 1$
- For $x > 2\pi,\ V_N(x)$ behaves like the cosh, and $\lim\limits_{x\to\infty}\cosh(x) = \infty$



17-18

Type I Chebyshev Filters

- **The Chebyshev polynomials satisfy a simple recursion**

$$V_0(x)=1,\ V_1(x)=x,\ V_{N+1}(x)=2xV_N(x)-V_{N-1}(x)$$

$$V_2(x)=2xV_1(x)-V_0(x)=2xx-1=2x^2-1$$

- **The squared magnitude depends on** $V_N(x)$

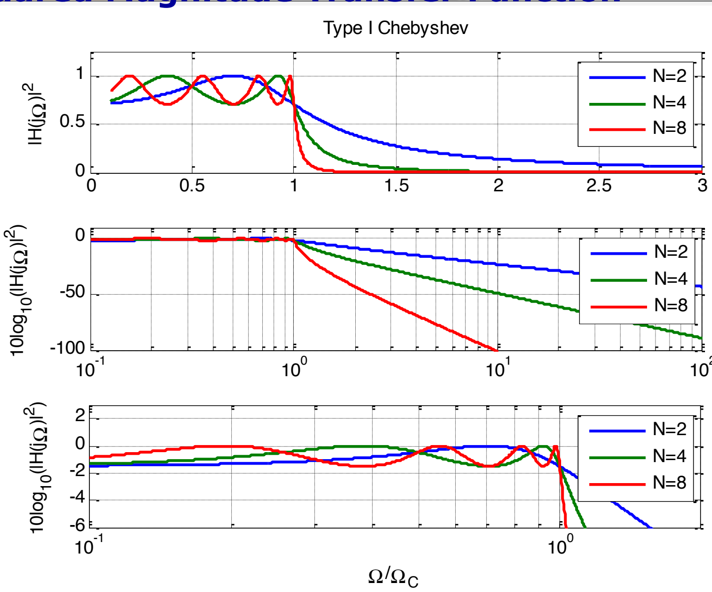
$$|H(s)|^2=\frac{1}{1+\varepsilon^2V_N^2(s/\omega_C)}$$

- **...where** ε is the in band voltage ripple $=10^{[\varepsilon]_{dB}/20}$
- **...and**

$|H(\Omega)|^2$ ripples between 1 and $1/(1+\varepsilon^2)\sim 1-\varepsilon^2$ for $0\le\omega\le\omega_C$

$|H(\Omega)|^2$ monotonically decreasing for $\omega>\omega_C$

Squared Magnitude Transfer Function

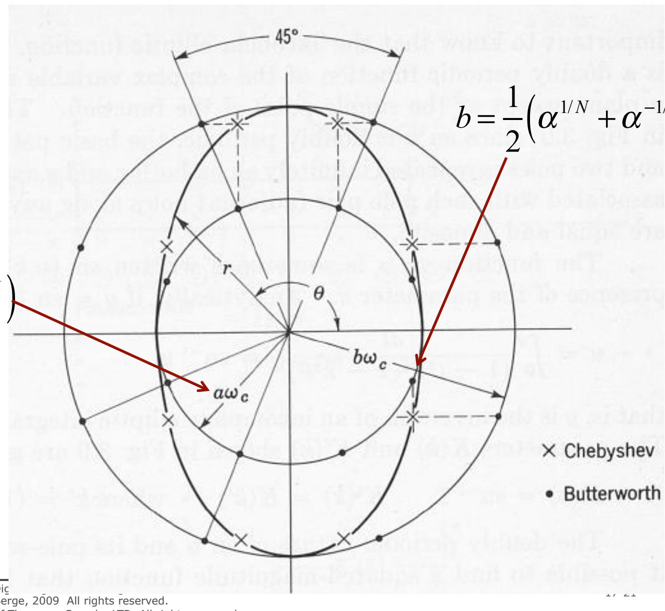


Type I Chebyshev poles (no zeros!)

$$\alpha = \varepsilon^{-1} + \sqrt{1+\varepsilon^{-2}}$$

$$a = \frac{1}{2}\left(\alpha^{1/N} - \alpha^{-1/N}\right)$$

$$b = \frac{1}{2}\left(\alpha^{1/N} + \alpha^{-1/N}\right)$$



Type I Chebyshev: What is N?

Let $[r_p]_{dB}$ be the pass band ripple in dB >0

Let $[r_s]_{dB}$ be the stop band rejection in dB >0

Let ω_C be the "r_p"-dB bandwidth

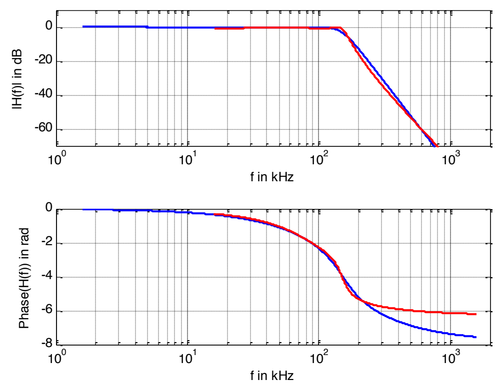
Let ω_S be the stop band bandwidth where a minimum of "r_S"-dB rejection.

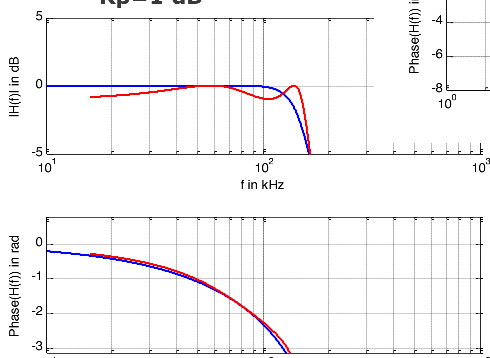
Then

$$N \approx \left\lceil \frac{\cosh^{-1}\sqrt{\dfrac{10^{r_S/10}-1}{10^{r_p/10}-1}}}{\cosh^{-1}(\omega_S/\omega_C)} \right\rceil$$

Chebyshev/ Butterworth Comparison

- **Butterworth**
  - **5 poles**
- **Chebyshev Type I**
  - **4 poles**
  - **Rp=1 dB**





Chebyshev Type II

- **Type II is monotonic in the pass band, with maximum ripple in the stop band.**
- **We obtain Type II from a baseline Type I design**

$$\left|H(\Omega)\right|^2_{\text{Type II}} = \frac{1}{1+\left[\varepsilon^2 V_N^2\left(\Omega_C/\Omega\right)\right]^{-1}}$$

Bandpass analog filters

- We can shift the passband by moving the poles along the $j\omega$ axis
- Let a Butterworth low-pass prototype be given by

$$|H(p)|^2 = \frac{1}{1+(p/j\omega_C)^{2N}}$$

- Now make the transformation

$$p = \frac{s^2+\omega_0^2}{s}, \text{ which maps } p=0 \text{ to } s=\pm j\omega_0$$

- And maps $p=(-\infty,\infty) \rightarrow s=[0,\infty)$ and $s=[0,-\infty)$

For a bandpass design

Let $\omega_{p1} < \omega_{p2}$ be the edges of the passband

and $\Omega_{s1} < \Omega_{s2}$ be the edges of the stopband

$$\omega_0 = \sqrt{\omega_{p1}\omega_{p2}}$$

$$\omega_{pp} = \frac{\omega_{p2}^2-\omega_0^2}{\omega_{p2}}, \omega_{ps} = \min\left(\frac{\omega_{s2}^2-\omega_0^2}{\omega_{s2}}, \frac{\omega_{s1}^2-\omega_0^2}{\omega_{s1}}\right)$$

See http://cnx.org/content/m16913/latest/

```
%Analog filter design in MATLAB
% Design Criteria
Ap=3; % dB always given positive
As=60;% dB always given positive
fs=825e3; %in Hz
fp=75e3; %in Hz
eps2=10^(Ap/10)-1;
nu_s=fs/fp; % fs/fp for low-pass
N=ceil(0.5*log10(((10^(As/10)-1)/eps2))/log10(nu_s));
[bs,as]=butter(N,2*pi*fp,'s');


% Do it again with Cheby Type I
rp=1; % 1 dB in-band ripple
rs=As;% 60 dB out of band rejection
N=ceil( acosh(sqrt((10^(rs/10)-1)/(10^(rp/10)-1)))/acosh(fs/
fp));
[bsc,asc]=cheby1(N,rp,2*pi*fp,'s');
```