

## **CHEBYSHEV AS A FILTER CHARACTERISTIC**

### INTRODUCTION

In the previous app note, AN-419 [11], we reviewed Chebyshev Polynomials as mathematical objects of interest, remarking that they were familiar in a practical sense from filter design. Well, the name at least is familiar, and the equiripple signature property are. Further, we know, more or less, exactly how to obtain (and realize) Chebyshev transfer functions [1,3,5,7,8,9]. What may be neglected and therefore less clear is the connection between the polynomials and the frequency response curves. For example, it is clear that a Chebyshev polynomial as seen in [11] could NOT be a frequency response curve, as the polynomial runs to infinity while (useful) frequency responses are finite everywhere and often run to (often asymptotically to) zero. So what is going on?

### DESIGNING THE TRANSFER FUNCTION

Here we will be looking at (reviewing) the design of an example Chebyshev low-pass filter. Once this design is done (essentially just finding the poles) we can exercise the mathematics in various ways.

Recall that we obtain Chebyshev poles by first obtaining Butterworth poles for the same order [4.5.7]. In fact, it is completely correct to say that a Butterworth filter IS just a Chebyshev filter designed for zero ripple. Once the Butterworth poles are obtained (as we will shortly review), the corresponding Chebyshev poles are obtained by simply multiplying the real parts of all these poles by some “reduction factor”  $r$  ( $0 < r < 1$ ) that is the same for all the poles [5,8,10]. (For Butterworth,  $r=1$ .) There is a formula for  $r$ , based on the order and ripple allowed, but we could just find a satisfactory value of  $r$  by trial and error. That is, we choose some trial value of  $r$  – perhaps we try 0.5. We then look at the resulting Chebyshev ripple and if it is too large, increase  $r$ , else we may want to try decreasing  $r$  for more ripple (which we didn’t really “want”) and a correspondingly sharper cutoff rate (which we often do want).

The “formula” for Butterworth (BW) [4,5,7] poles is not really mathematical in the sense that one writes it down, but in that one remembers how to derive the pole positions. The BW poles are all on a unit circle. For an order- $N$  BW pole set, place  $2N$  equally spaced poles on this unit circle. Now rotate this array until poles are complex conjugates, and so that there are no poles on the  $j\omega$ -axis. Finally, erase the  $N$  poles that are in the right half of the  $s$ -plane. Fig. 1 shows the results of this procedure of  $N=5$ .

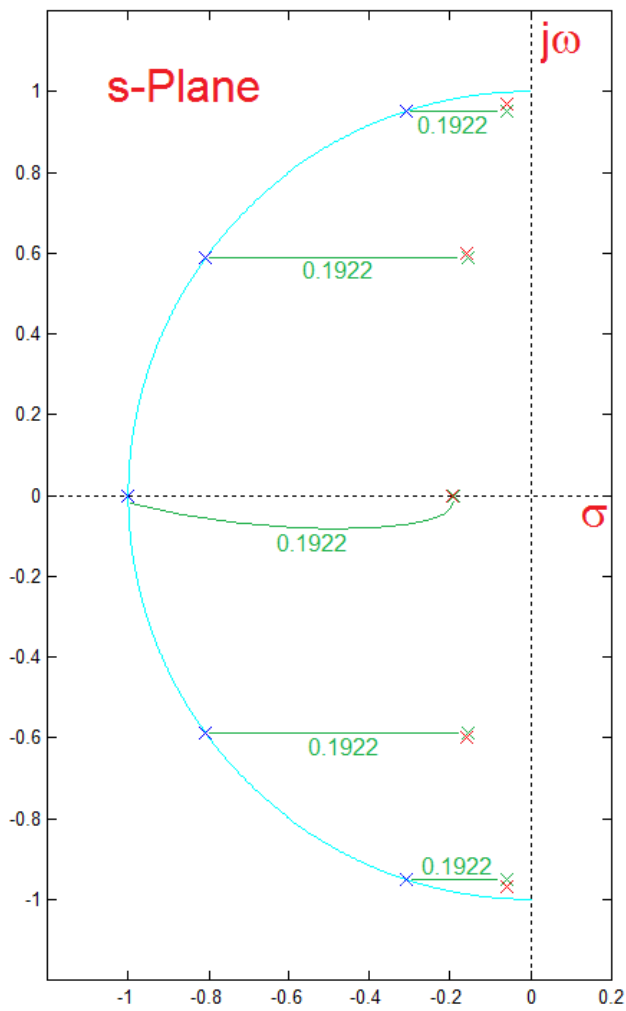


Fig. 1

N = 5  
 R=0.75  
 (r = 0.1922)

Butterworth poles (blue) are at radius 1 and are separated by  $36^\circ$

Chebyshev poles (green) have a reduced real part (multiplied by r) and subsequently are moved outward from  $s=0$  slightly (by 1.018) to the red positions to "tweak" the cutoff (see text)

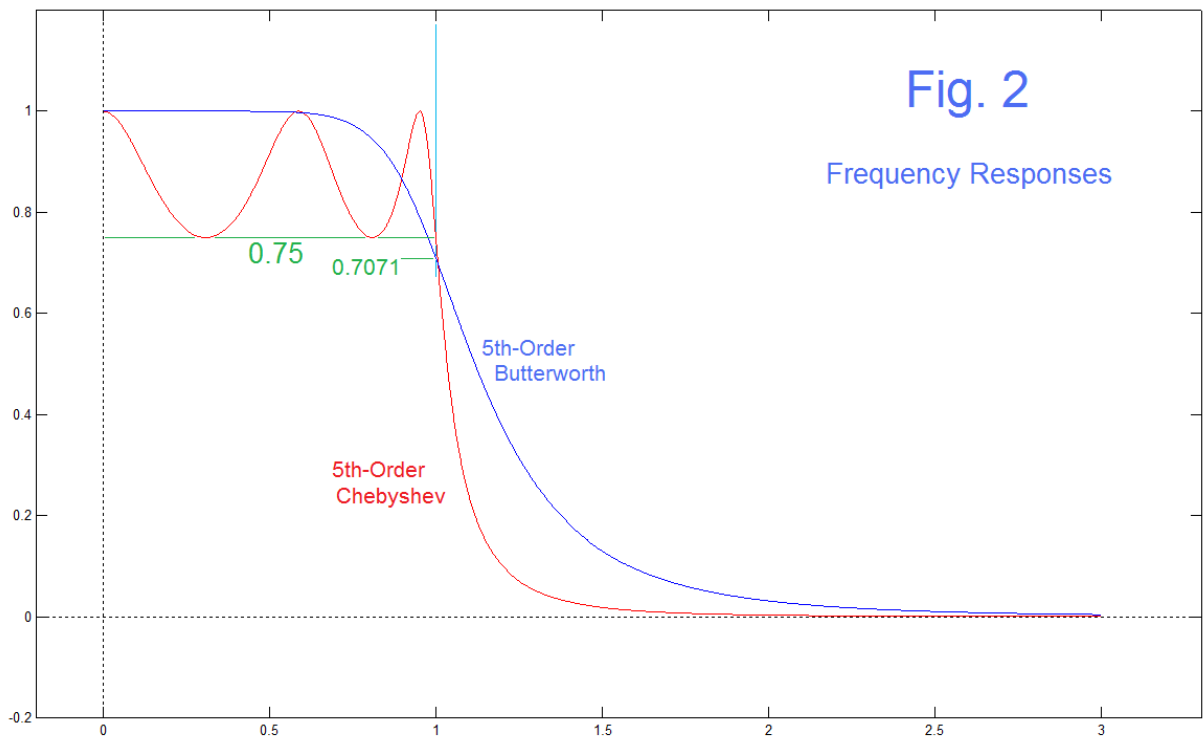


Fig. 2

Frequency Responses

We see from Fig. 1 the BW poles are separated by 36° with the poles closest to the  $j\omega$ -axis being 18° from the axis (36° from the “erased” right-half-plane pole). Then our remaining task is to reduce the real part of these poles by a factor  $r$ . In addition to choosing  $N=5$  we chose a ripple factor  $R=0.75$  (look ahead to Fig. 2) which means the response can ripple down to 0.75. This is about 2.5db of ripple. Often times Chebyshev ripples are chosen at 1/2db, 1db, 2db, and 3db, so the one we choose here is typical but on the higher side – making a better illustration perhaps. Nothing requires any particular value in terms of a fraction  $R$  or in db.

Now, having chosen  $R=0.75$ , there is a formula giving  $r$ :

$$r = \text{Tanh} \left[ \left( \frac{1}{N} \right) \text{sinh}^{-1} \frac{R}{\sqrt{1 + R^2}} \right] \quad (1)$$

which gives  $r = 0.1922$  for  $R=0.75$  and  $N=5$ . We note two things about this  $r$ . First the equation is not easy to remember, and secondly, the resulting value of  $r$  is quite small (so it seems). The green lines in Fig. 2 indicate how the BW poles move toward the  $j\omega$ -axis when their real parts are multiplied by  $r$ . These green poles are perfectly good in practice. However, we are interested in getting calculations to verify exactly (many decimal places) with the Chebyshev theory [1,2], so these poles were adjusted slightly outward (by about 2%) from  $s=0$  to the red positions. That is, the radii of the poles were all multiplied by 1.018 more or less by trial and error, but to correct for expected discrepancies as we will discuss just below.

Once we have the poles (BW, or the green or red Chebyshev poles) we can calculate the transfer functions (see below) and/or the frequency responses. These were actually done with Matlab’s **freqs** function (see code at end) but we will remind you how to do this in a moment [6,9]. Here we want to look at the frequency response curves of Fig. 2. We see the nice 5<sup>th</sup>-order BW which is nice and flat in the passband, and the 5<sup>th</sup>-order Chebyshev is remarkable for its equi-ripple and a much sharper cutoff region. This is the performance consideration of course – trading ripple for a sharp cutoff.

Now, BW low-pass filter of all orders have frequency responses that pass through  $1/\sqrt{2}$  at a frequency of 1. [ Here we have not found it necessary to specify frequency in exact units. Whatever units we choose for the radius of the unit circle is 1. ] In an actual design, this is denormalized as needed. So for BW, calling the cutoff 1 with corresponding magnitude response  $1/\sqrt{2}$  (which is close to, but not exactly -3db) is an obvious choice. But where is the corresponding cutoff for Chebyshev? Not so obvious. Perhaps 3db down from the peak? Or perhaps 3db down from dc, which would be the same here, but not for even  $N$ . Why not R down from the peak – and this is what we will use here (consistent with the Chebyshev theory[1,3]). Even for  $N=5$  we already see that there is a very sharp cutoff just above the pole that is closest to the  $j\omega$ -axis, and the radius of this pole approaches 1. Thus no matter how we exactly define the cutoff, it will be approximately 1. It is the slight differences which we have noted in our write-ups [5] that make the tweaks necessary, as the specific 1.018 expansion of the radii here. So that’s that.

So we have simply obtained poles and, based on them, jumped ahead to the frequency responses, showing that we likely are on the right track. Let's fill in some details. We will be continuing with the red poles.

The transfer function is obtained from the pole set  $p_1$  to  $p_N$  as:

$$T(s) = \frac{K}{(s - p_1)(s - p_2) \cdots (s - p_N)} \quad (2)$$

and the denominator can be multiplied out (mildly tedious algebra, or just something like Matlab's **poly** function if the poles are known numerically):

$$T(s) = \frac{K}{s^N + a_{N-1}s^{N-1} + a_{N-2}s^{N-2} + \cdots + a_0} \quad (3)$$

Here the value of K is an overall gain constant and we can freely adjust the gain for convenience. The denominator in equation (3) is a polynomial in s, but keep in mind that s is in general complex: ( $s = \sigma + j\omega$ ). Further, if we have used BW or Chebyshev poles, the denominator is a polynomial that corresponds to BW or Chebyshev transfer functions, but are not BW or Chebyshev polynomials in the more formal mathematical sense. (They may be all we need to complete a design and implement a working filter.) With the poles from a Chebyshev set, confusing the denominator or T(s) for a "Chebyshev polynomial" is the same as confusing the red curve of Fig. 2 for a Chebyshev polynomial of the type we saw before [11], and will revisit below.

Before going on to the Chebyshev polynomial study, we need to be clear about how we get the frequency response curve from equation (2) or equation (3). Since T(s) is a complex number, we find the frequency response by the usual procedures of finding the magnitude of a complex number, evaluated at  $s = j\omega$ . In fact, it will be convenient for what follows to find the square magnitude of T(s):

$$|T(s)|^2 = T(j\omega) \cdot T(-j\omega) \quad (4)$$

If we want "frequency response" we take the square root. There are several practical ways of obtaining this frequency response [6,9]. For the moment, note that we have the "recipe" here: we just plug equation (3) into equation (4) and chug it out. In the case of our 5<sup>th</sup>-order Chebyshev, this is quite tedious, but perhaps necessary to do once.

The red poles in Fig. 1, our example, are:

$$\begin{aligned} & -0.0605 + 0.9682i \\ & -0.1583 + 0.5984i \\ & -0.1956 + 0.0000i \\ & -0.1583 - 0.5984i \\ & -0.0605 - 0.9682i \end{aligned} \quad (5)$$

Plugging these into equation (2) and multiplying out, the denominator of equation (3) becomes:

$$D(s) = s^5 + 0.6331s^4 + 1.4480s^3 + 0.6107s^2 + 0.4278s + 0.0705 \quad (6)$$

and next we put in  $j\omega$  and  $-j\omega$  for  $s$

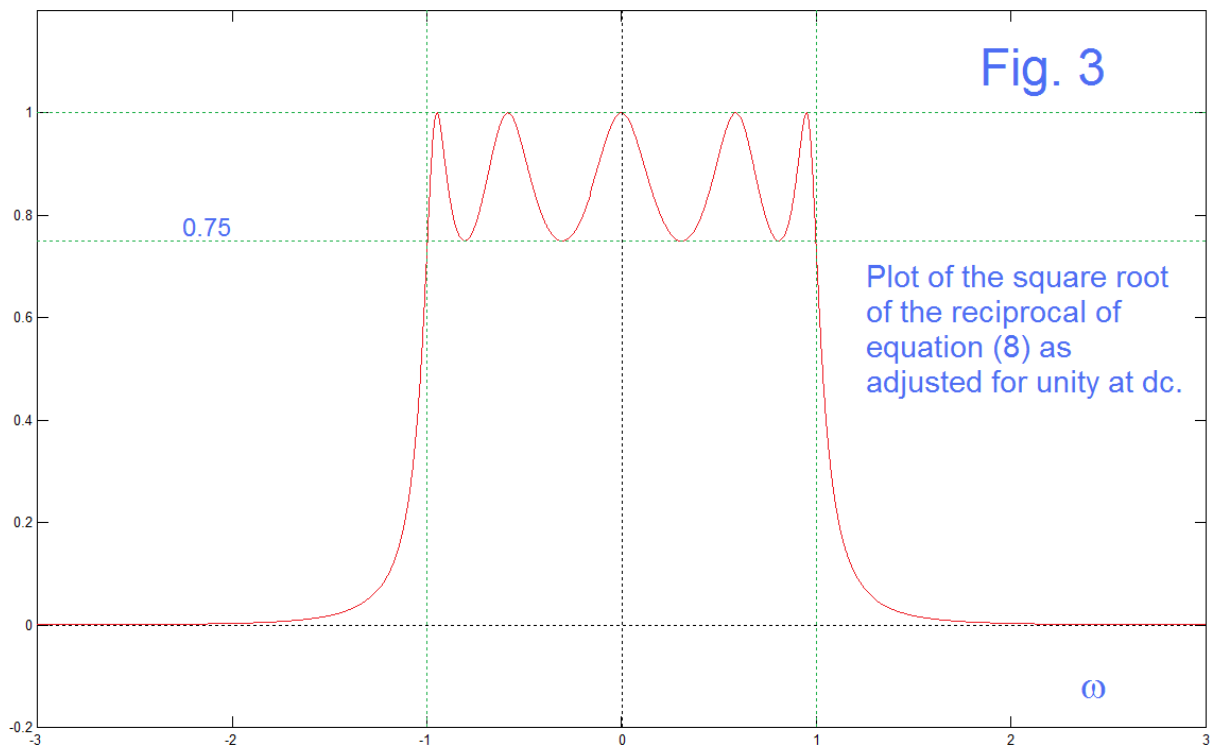
$$D(j\omega) = j\omega^5 + 0.6331\omega^4 - 1.4480j\omega^3 - 0.6107\omega^2 + 0.4278j\omega + 0.0705 \quad (7a)$$

$$D(-j\omega) = -j\omega^5 + 0.6331\omega^4 + 1.4480j\omega^3 - 0.6107\omega^2 - 0.4278j\omega + 0.0705 \quad (7b)$$

Where we have fought with powers of  $j$  and of  $-j$ . Multiplying these together looks very tedious, but we keep in mind that the odd powers of  $\omega$  (those multiplied by  $j$ ) will cancel in the cross-multiplication (since the magnitude must be real), we arrive at

$$D(j\omega)D(-j\omega) = \omega^{10} - 2.4951\omega^8 + 2.1790\omega^6 - 0.7767\omega^4 + 0.0969\omega^2 + 0.0050 \quad (8)$$

This is the denominator of  $|T(s)|^2$ , and is as far as we can go. We can plot  $|T(s)|$  however by evaluating equation (8) on some convenient interval (-3 to +3 here) and taking the square root of the reciprocal, adjusting for unity gain. This is plotted in Fig. 3. We see that this is the same as the frequency response of Fig. 2, which is the expected result. The response is symmetric about zero frequency because it uses only even powers of  $\omega$  – we could have plotted just the positive side. Note that the passband is nicely “boxed” inside  $\omega=\pm 1$  and magnitude  $R=0.75$  to 1.



## THE CHEBYSHEV APPROXIMATION

The Chebyshev polynomials as they are usually tabulated are:

$$C_N(\omega) = \cos(N\cos^{-1}\omega) \quad \text{for } |\omega| \leq 1 \quad (9a)$$

$$C_N(\omega) = \cosh(N\cosh^{-1}\omega) \quad \text{for } |\omega| > 1 \quad (9b)$$

which seem to be perfectly good functions. There is the restriction that these are not really polynomials yet, and that we have to switch the functional form as  $|\omega|$  goes through 1. Note that when  $N=0$  we get  $C_0(\omega)=\cos(0)=1$  and when  $N=1$  we get  $C_1(\omega) = \cos(\cos^{-1}(\omega))=\omega$ . There is also a recursion relationship [1]:

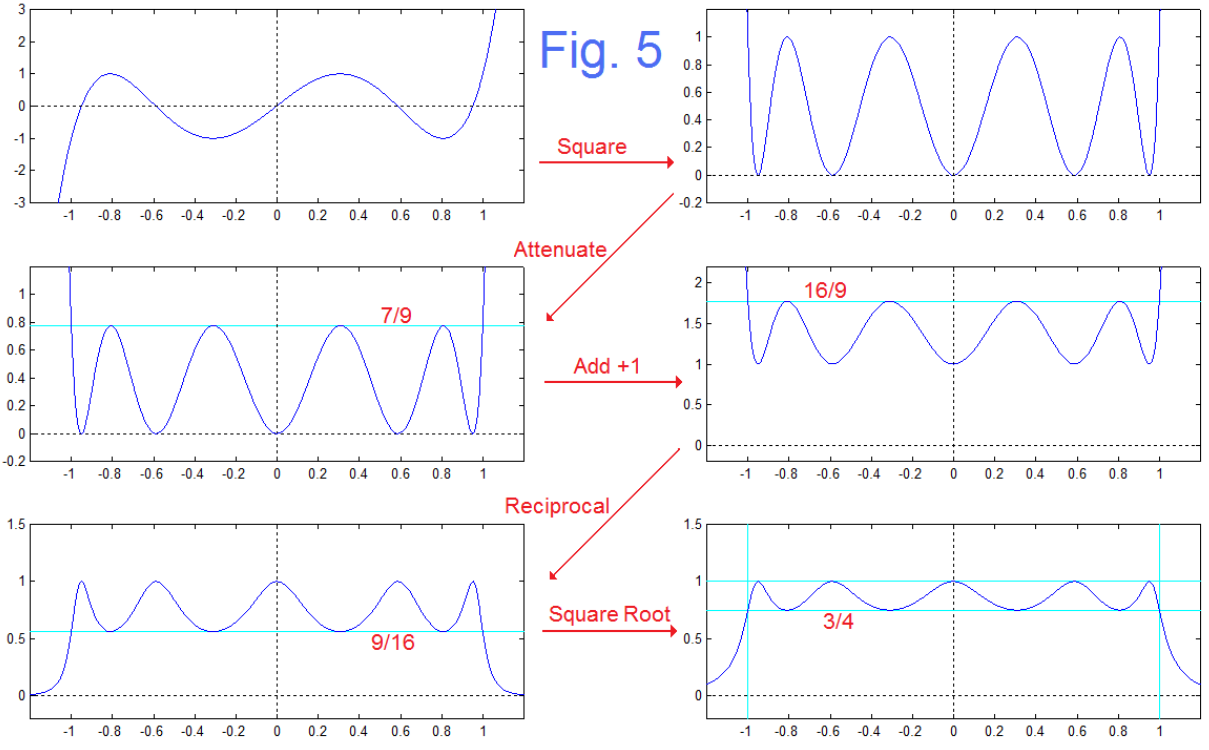
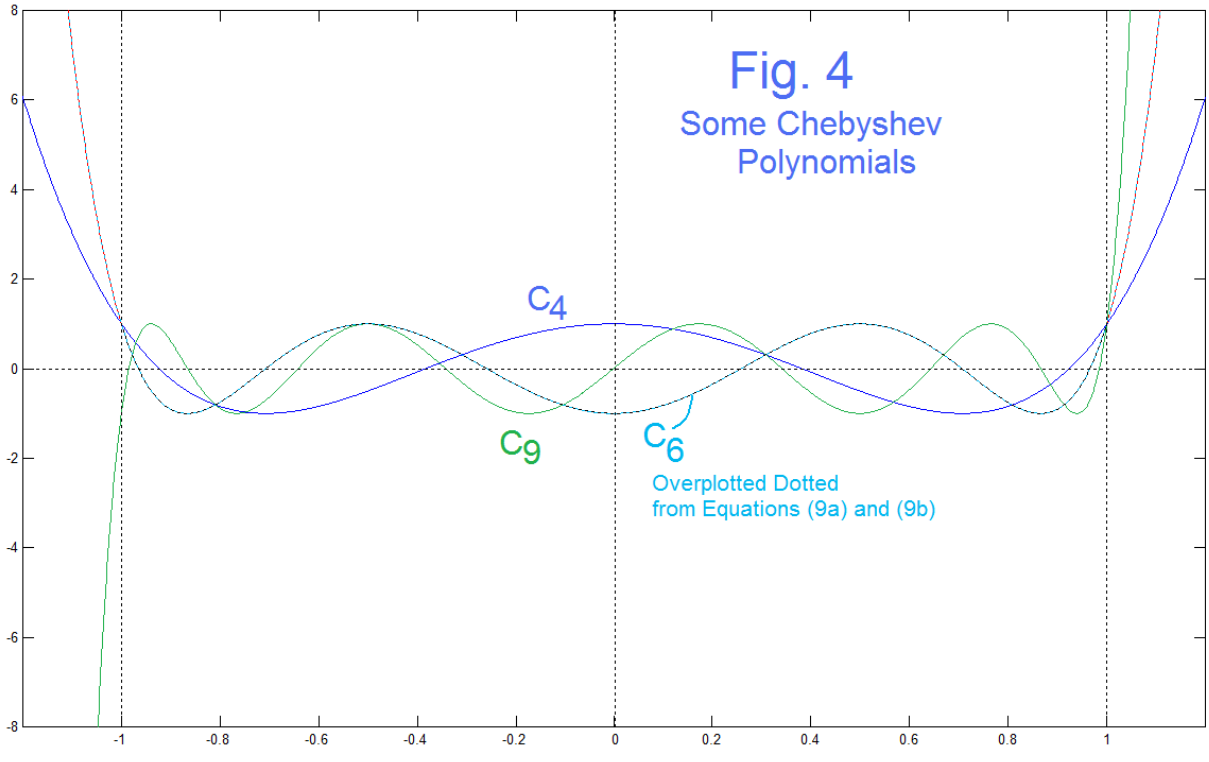
$$C_N(\omega) = 2\omega C_{N-1}(\omega) - C_{N-2}(\omega) \quad (10)$$

by which a new polynomial is obtained in terms of the previous two. This allows us to construct a table:

N	$C_N(\omega)$
0	1
1	$\omega$
2	$2\omega^2 - 1$
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$
5	$16\omega^5 - 20\omega^3 + 5\omega$
6	$32\omega^6 - 48\omega^4 + 18\omega^2 - 1$
7	$64\omega^7 - 112\omega^5 + 56\omega^3 - 7\omega$
8	$128\omega^8 - 256\omega^6 + 160\omega^4 - 32\omega^2 + 1$
9	$256\omega^9 - 576\omega^7 + 432\omega^5 - 120\omega^3 + 9\omega$
10	$512\omega^{10} - 1280\omega^8 + 1120\omega^6 - 400\omega^4 + 50\omega^2 - 1$
11	$1024\omega^{11} - 2816\omega^9 + 2816\omega^7 - 1232\omega^5 + 220\omega^3 - 11\omega$
12	$2048\omega^{12} - 6144\omega^{10} + 6912\omega^8 - 3584\omega^6 + 840\omega^4 - 72\omega^2 + 1$

Finally we see what the polynomials look like. We note that the coefficients are all integers and that every other power is skipped. These do not look at all like the numbers we have encountered working from the poles so far.

At least we have polynomials. Thus we can plot a few,  $C_4(\omega)$ ,  $C_6(\omega)$ , and  $C_9(\omega)$  in Fig. 4. These are plotted by choosing a range of  $\omega$  from -1.2 to +1.2 and directly calculating using the tabulated coefficients. As we expect, these stay nicely inside limits for values of  $\omega$  between -1 and +1, but then run away, headed for  $+\infty$  or  $-\infty$  outside that center range. To make sure we are calculating correctly, the  $C_6(\omega)$  polynomial is calculated using equations (9a) and (9b) and overplotted with dashed red (outside  $\pm 1$ ) or dashed black (inside  $\pm 1$ ). The agreement is perfect as it should be. Note that the difference is equations (9a) and (9b) is whether or not ordinary cosines or hyperbolic cosines (imaginary arguments) are required. In actual fact, Matlab is forgiving and chooses the right function automatically.



## COMPLETING THE PROCEDURE

Now, recognizing the fact that the connection between mathematical Chebyshev polynomials and the pole-determining polynomial of a Chebyshev transfer function is indirect, we can complete the loop and show exactly how to connect the two. This is given in the equation (11) below for the squared magnitude [corresponding to equation (8) of our pole procedure] and is also summarized in the example of Fig. 5. This equation is actually telling us what was going on all the time:

$$|T(s)|^2 = \frac{1}{[1 + \epsilon^2 C_N^2(\omega)]} \quad (11)$$

That is, you take the Chebyshev polynomial, square it (doubling the order and making the result strictly non-negative), multiply it by some attenuation factor  $\epsilon^2$ , add 1, and take the reciprocal (an arbitrary constant, which we will take to be 1, appears in the numerator). It is clear from equation (9a) that at the edge of the defined region, at  $\omega=1$ , the value of  $C_N(\omega=1) = 1$ . This is also clear that this is true from examining the table on page 6 (this is “cute!”), and is suggested in Fig. 4 for example. This means that:

$$|T(\omega = 1)| = \frac{1}{\sqrt{1 + \epsilon^2}} \equiv R \quad (12a)$$

or

$$\epsilon^2 = \frac{1}{R^2} - 1 \quad (12b)$$

So for our example of  $R=3/4$  we have  $\epsilon^2 = 7/9$ , which we can plug into equation (11), the denominator of which is now:

$$199.1111\omega^{10} - 497.7778\omega^8 + 435.5556\omega^6 - 155.5556\omega^4 + 19.4444\omega^2 + 1 \quad (13)$$

which is the same as equation (8) except for a scale factor of 199.1111.

That’s about it. Oh, this approach to setting a filter characteristic is classic [1] and quite general, and we can generalize equation (11) as:

$$|T(s)|^2 = \frac{1}{[1 + f(\omega^2)]} \quad (14)$$

In this context, Butterworth is just:

$$|T(s)|^2 = \frac{1}{[1 + \omega^{2N}]} \quad (15)$$

and we often see equation (15) freely presented without derivation/elaboration as the “way” to calculate the frequency response of a Butterworth filter.



## REFERENCES

### [General References on the Topic Here](#)

- [1] Kuo, Franklin F., **Network Analysis and Synthesis**, Wiley (1962), Chapter 12
- [2] Abramowitz, M., and I. A. Stegun, **Handbook of Mathematical Functions**, Dover (1965) pp 795 for tables
- [3] Schaumann, R., and M. E. Van Valkenburg, Design of Analog Filters, Oxford U. Press (2001) Section 7.2

### [General Electronotes References](#)

- [4] “Laboratory Problems and Examples in Active, Voltage-Controlled, and Delay Line Networks” Electronotes Supplement S-016 (1978)  
<http://electronotes.netfirms.com/S016.pdf>
- [5] “Placement of Butterworth and Chebyshev Low-Pass Poles: Cascade Form”, Chapter 3, Section 3-3 of Analog Signal Processing, **Electronotes**, Vol. 19, No. 192, Feb 2000. See page 22 for discussion of “tweaking” the cutoff.  
<http://electronotes.netfirms.com/EN192.pdf>

### [Specific Application Notes – Numerical Order](#)

- [6] “Graphical Determination of Frequency Response – s-Plane Case” Electronotes Application Note AN-45, July 4, 1977
- [7] “Design of Butterworth Filters Without Data,” Electronotes Application Note AN-75, Feb. 18, 1978
- [8] “Graphical Methods for Chebyshev Pole Placement,” Electronotes Application Note AN-76, Feb. 25, 1978
- [9] “TI-59 Program for T(s) Without factoring,” Electronotes Application Note AN-256, July 20, 1982 (program obsolete but theory short and simple there)  
<http://electronotes.netfirms.com/AN256.PDF>
- [10] “Design of Chebyshev Filters -1” and “Design of Chebyshev Filters – 2” Electronotes Application Notes AN-275 and AN-276, Mar. 25, 1983 and April 5, 1983
- [11] “Finding Chebyshev Polynomials”, Electronotes Application Note 419, Dec. 26, 2014  
<http://electronotes.netfirms.com/AN419.pdf>

## PROGRAM (Matlab)

```
% AN420.m
% Here figures 1-5 of program are exactly FIG. 1-5 pf this note

% Fig. 1 Poles
p=[]
N=5
R=0.75
r=tanh((1/N)*asinh(R/sqrt(1-R^2)))
sa=(2*pi)/(2*N)
for k=0:N-1
    pbutt(k+1)=j*cos(sa/2+k*sa) - sin(sa/2+k*sa);
    p(k+1)=j*cos(sa/2+k*sa) - r*sin(sa/2+k*sa);
end
porig=p
p=1.018*p % adjust poles slightly for cutoff
figure(1)
plot(real(porig),imag(porig),'gx')
hold on
plot(real(p),imag(p),'rx')
a=-pi/2:pi/1000:pi/2;
cx=cos(a);
cy=sin(a);
plot(-cx,cy,'c')
plot(real(pbutt),imag(pbutt),'xb')
plot([0 0],[-2 2],'k:')
plot([-2 1],[0 0],'k:')
hold off
axis('equal')
axis([-1.2 0.2 -1.2 1.2])
figure(1)
%
%
% Supp calculations
pn=poly(p);
pn=real(pn)
pn1=real(poly([p(1) p(5)]))
pn2=real(poly([p(2) p(4)]))
pn3=real(poly([p(3)]))
%
%
% Fig. 2 - Plot Frequency Responses
figure(2)
f=0:.001:3;
TC=abs(freqloc(1,pn,f));
TC=TC/max(TC);
plot(f,TC,'r')
pnbutt=real(poly(pbutt));
TB=abs(freqloc(1,pnbutt,f));
hold on
plot(f,TB,'b')
plot([-1 4],[0 0],'k:')
plot([0 0],[-1 2],'k:')
hold off
axis([-0.2 3.3 -0.2 1.2])
figure(2)
```

```

% Fig. 3 find Cheb Poly from Filter T(s), no freqs
D1=[pn(1)*j pn(2) -pn(3)*j -pn(4) pn(5)*j pn(6)]
D2=[-pn(1)*j pn(2) pn(3)*j -pn(4) -pn(5)*j pn(6)]
D=conv(D1,D2) % convolving = multiplication of terms
Dfrompoles=D
w=-3:.001:3;
DD=D(1)*w.^10 + D(3)*w.^8 + D(5)*w.^6+D(7)*w.^4 +D(9)*w.^2 + D(11);
SDD=1./sqrt(DD);
SDD=SDD/max(SDD);
figure(3)
plot(w,SDD)
hold on
plot([-3 3],[0 0],'k:')
plot([0 0],[-1 2],'k:')
plot([-3 3],[0.75 0.75],'c:')
plot([-3 3],[1 1],'c:')
plot([-1 -1],[-1 2],'c:')
plot([1 1],[-1 2],'c:')
hold off
axis([-3 3 -.2 1.2])

```

```

% Fig. 4 Plot some Cheby Polys
w=-1.2:.001:1.2;
C4=8*w.^4 - 8*w.^2 + 1;
C6=32*w.^6 - 48*w.^4 + 18*w.^2 -1;
C9=256*w.^9 -576*w.^7 +432*w.^5 - 120*w.^3 +9*w;
figure(4)
plot(w,C4,'b')
hold on
plot(w,C9,'g')
plot(w,C6,'c')
ww=-2:.001:-1;
C6form=cosh(6*acosh(ww));
plot(ww,C6form,'r:')
ww=1:.001:2;
C6form=cosh(6*acosh(ww));
plot(ww,C6form,'r:')
ww=-1:.001:1;
C6form=cos(6*acos(ww));
plot(ww,C6form,'k:')
plot([-2 2],[0 0],'k:')
plot([-1 -1],[-8 8],'k:')
plot([1 1],[-8 8],'k:')
plot([0 0],[-8 8],'k:')
hold off
axis([-1.2 1.2 -8 8])

```

```

% Fig. 5 procedure to manipulate polynomial
w=-2:.001:2;
C5=16*w.^5 -20*w.^3 + 5*w;
figure(5)
subplot(321)
plot(w,C5,'b')
hold on
plot([-2 2],[0 0],'k:')
plot([0 0],[-8 8],'k:')
axis([-1.2 1.2 -3 3])
hold off
% square
C5S=C5.^2;
subplot(322)
plot(w,C5S,'b')
hold on
plot([-2 2],[0 0],'k:')
plot([0 0],[-8 8],'k:')
axis([-1.2 1.2 -0.2 1.2])
hold off
%
C5SA=C5S*(7/9) ;
subplot(323)
plot([-2 2],[7/9 7/9],'c')
hold on
plot([-2 2],[0 0],'k:')
plot([0 0],[-8 8],'k:')
plot(w,C5SA,'b')
axis([-1.2 1.2 -0.2 1.2])
hold off
%
C5SAP1=C5SA + 1;
subplot(324)
plot([-2 2],[16/9 16/9],'c')
hold on
plot([-2 2],[0 0],'k:')
plot([0 0],[-8 8],'k:')
plot([-2 2],[16/9 16/9],'c')
plot(w,C5SAP1,'b')
axis([-1.2 1.2 -0.2 2.2])
hold off
%
C5SAP1R=1./C5SAP1;
subplot(325)
plot([-2 2],[0 0],'k:')
hold on
plot([0 0],[-8 8],'k:')
plot([-2 2],[9/16 9/16],'c')
plot(w,C5SAP1R,'b')
axis([-1.2 1.2 -0.2 1.5])
hold off

```

```

% square root
C5SAP1RSR = sqrt(C5SAP1R);
subplot(326)
plot([-2 2],[0 0],'k:')
hold on
plot([0 0],[-8 8],'k:')
plot([-2 2],[3/4 3/4],'c')
plot([-2 2],[1 1],'c')
plot([-1 -1],[-2 2],'c')
plot([1 1],[-2 2],'c')
plot(w,C5SAP1RSR,'b')
hold off
axis([-1.2 1.2 -0.2 1.5])

% find Cheb Poly from Textbook Data
w=-3:.001:3;
%D=[199.11109 0 -497.77778 0 435.5555 0 -155.55554 0 19.44444 0 1]
% numbers from hand-calculation - overwritten just below
ep2=1/R^2 -1
D=[ep2*conv([16 0 -20 0 5], [16 0 -20 0 5]) 0 1]
Dfromtext=D
DD=D(1)*w.^10 + D(3)*w.^8 + D(5)*w.^6+D(7)*w.^4 +D(9)*w.^2 + D(11);
SDD=1./sqrt(DD);
SDD=SDD/max(SDD);
figure(6) % NOT IN THIS NOTE - IDENTICAL TO Fig. 3 of Note
plot(w,SDD)
hold on
plot([-3 3],[0 0],'k:')
plot([0 0],[-1 2],'k:')
hold off
axis([-3 3 -.2 1.2])

%
% Check
RAT=Dfrompoles./Dfromtext %ignore divide by 0

```