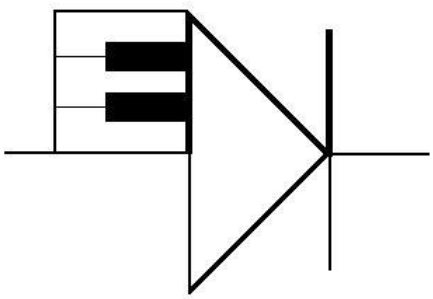


# ELECTRONOTES

WEBNOTE 42 9/1/2016

ENWN-42

## THE 4-POLE LOOKED AT A BIT MORE

## Back-Story and New Finding

## Back-Story

People send me stuff. Often saying that they are sure I have probably seen it and know about it. Mostly it is completely new to me. So never hesitate to send me a note. Not infrequently, the hint leads to a new posting published on this site, as is the current case.

The topic responsible for the current Webnote was an item posted on the Synth-DIY site which I seldom get to these days. Because the reader who sent the link said it had to do with the 4-pole low-pass, a popular topic here [1a, 1b, 2-5], and because they asked me a question about using feedback, I looked at it. A number of fun things and interesting things emerged in just an hour's playing around.

The story begins with a posting by Neil Johnson on the Synth-DIY site titled "Analysing 4-stage RC filters without the hot air" dated 21 Mar 2016. You can get the full post by going to our Synth-DIY link and searching for electronotes:

http://search.retrosynth.com/synth-diy/

The take-away line from Neil's post is:

> "…As a worked example I used a 4-stage RC filter from Electronotes where I show that actually it's not that difficult at all to analyse the unbuffered 4-stage RC filter contrary to Bernie's protestations."

Indeed, Neil has provided a solution to a problem which we understood but never finished. A pdf of Neil's presentation is found here:

http://www.milton.arachsys.com/nj71/index.php?menu=2&submenu=2&subsubmenu=14

This describes a valuable finding (the poles of the unbuffered 4-pole which I previously declined to fight with). Neil's paper is basically commendable. But I did wonder, upon reading the Synth-DIY text, what my protestation was going to be. It's in the pdf. Neil correctly borrowed a figure from EN#210 (posted free) and properly acknowledged the source. So I find my protest:

> "I don't know how many times I have started out to actually solve the unbuffered case (A). I always gave up, even after dropping back to the case where all the R's and all the C's were equal. It clearly was not impossible - just excessively tedious."

Sure – I wrote than in 2012. Note the word "tedious". Neil did solve the problem for which he gets pretty good marks. The marks would have been higher if he had actually done the algebra I had declined to do. But he did get the answer. How? Apparently he used an online program called "maxima" that chews up the algebra. What was his excuse for running to a program? He wrote:

> "Expanding out the multiplication would be rather tedious by hand, so we use maxima to do the algebraic gymnastics for us. Setting all resistors equal and all capacitors equal simplifies the transfer function to the following expression:"

The excuse in red (mine) and Neil's in blue say essentially the same thing ("tedious", and let's set the R's and C's equal)! The difference is the algebra crunching program Neil used.

My protestation above was not the only reason why I deserted the algebra problem. Immediately following the red quote from EN#210 I also said:

> "More importantly, there was no "pot of gold" lurking at the end of the tedious path. We already understood the general nature of the anemic results."

I abandoned the attempt in the late 1970's.

I should mention that the setup of the impossible hand-algebra in my case (simultaneous network equations) was different from Neil's (matrix 2-port cascade). Likely a program exists today to do the algebra with the network equations. Apparently "maxima" does matrix multiplies with symbolic elements.

Although Neil does not say so, the method of cascading 2-ports is not new. Perhaps he should have referenced any of a vast number of network theory books, but the fact that such material is taught (more than 50 years) at the engineering sophomore level is probably reason enough to omit it, or perhaps it was referenced in previous Synth-DIY comments. It does seem to turn out to be simpler than the simultaneous equations. In fact, it can lead to the correct solution in about 1/3 page of hand algebra. In fact, if all R's are equal and all C's are equal, the cascade of four-ports (Neil's equation just below my Fig. 4) becomes:

$$\begin{bmatrix} V_{in} \\ I_{in} \end{bmatrix} = \begin{bmatrix} 1 + sCR & R \\ sC & 1 \end{bmatrix}^4 \begin{bmatrix} V_{out} \\ I_{out} \end{bmatrix}$$

Squaring the 2x2 matrix and squaring the result is really not hard, and we only need the top left element. For additional simplicity, let RC=1, and the denominator of $V_{out}/V_{in}$ becomes:

$$D(s) = s^4 + 7s^3 + 15s^2 + 10s + 1$$

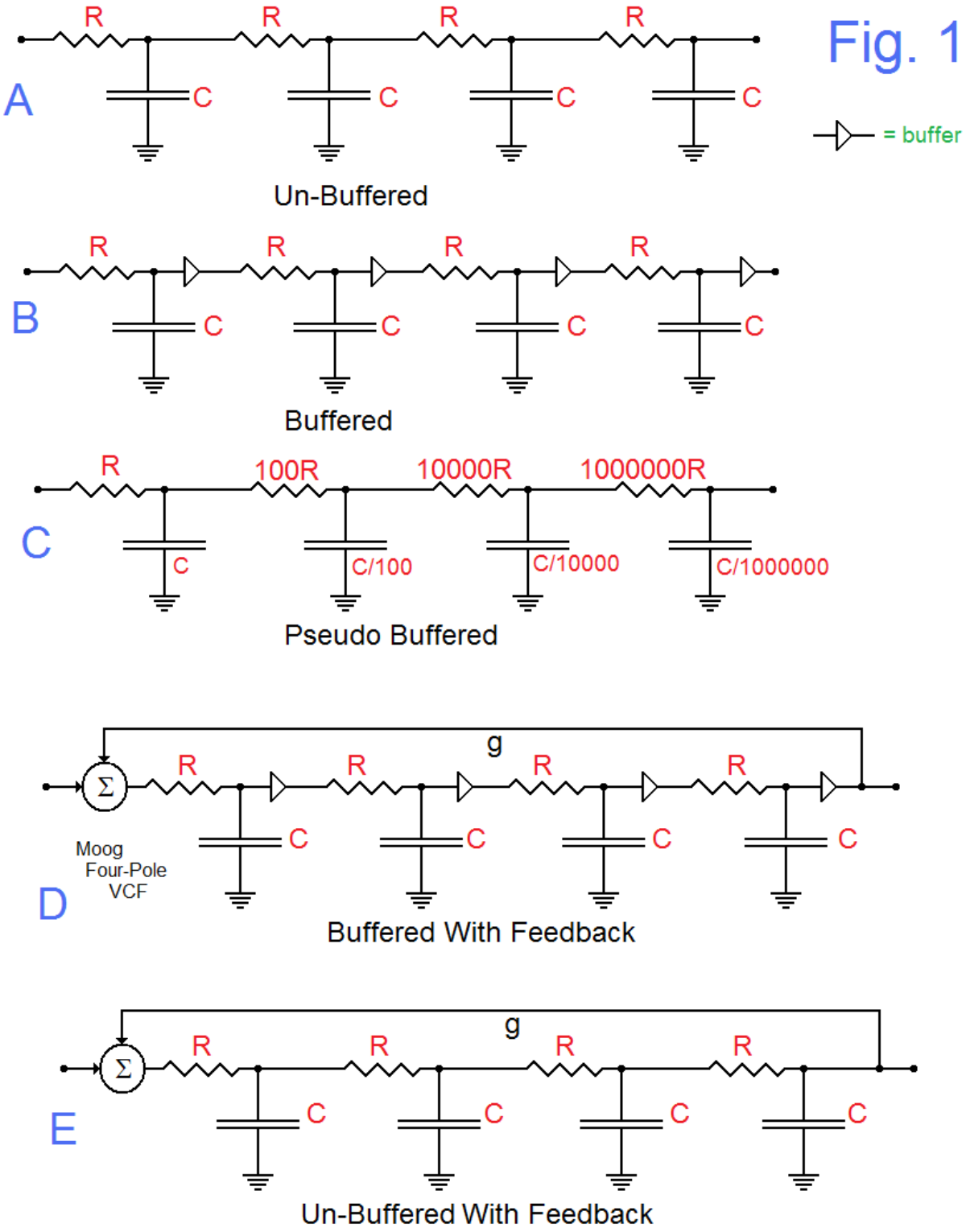
This is the same result Neil got. So it seems to be the right answer. Neil solves for the poles by a strange procedure of factoring out an obvious pole at s=-1 and then solving the remaining cubic with an online cubic solver. More directly in Matlab I just did:

```
roots([1 7 15 10 1])
ans =   -3.5321   -2.3473   -1.0000   -0.1206
```

which is the same answer Neil got. That is, four negative real poles. In comparison, in the buffered case we have all four poles at s=-1, so without buffering, the poles spread along the negative real axis.

## New Findings

Fig. 1 (A, B, C, D, and E) show the variations on the basic fourth-order cascade under consideration here. A is the original passive, unbuffered case that Neil (and I) end up with. [If you prefer, consider the start to actually be the same circuit with a series of potentially different R values ($R_1$, $R_2$, $R_3$, and $R_4$, left to right) and potentially different C values ($C_1$, $C_2$, $C_3$, and $C_4$, left to right) instead of jumping to this special case.] B is the case where buffers are used to simplify the cascaded interconnections to a simple multiply. The case in D is this buffered arrangement with the positive feedback loop that made Moog's four-pole low-pass so useful and popular. Two points are perhaps worth mentioning: (1) the original ladder version of the low-pass Moog first used was not quite like D, although the response was the same, and (2) the buffering between stages is largely automatic, part of the voltage-controlled implementation.




Ultimately we will address the reader's question to me about what happens when we use feedback, as in Moog's filter on the unbuffered cascade (E). But first note that what we have done here is not <u>just</u> to make the sections have the same RC product, but to have all the R's the same and all the C's the same. This is different. If we had allowed the R's and C's of the sections to vary, as long as the RC products were the same, we

could have arranged that each successive section minimally loads the previous one as a sort of pseudo-buffering (C of Fig. 1). { A version of this was recently posted [6].} For example, we might have the R of the first stage (Fig. 1 here) be 100 ohms while R of the second stage is 100 times larger (10k) and on to 1M and finally to 100M. This would stress the range of possible resistors. Each C would be adjusted to the corresponding R so that RC is the same for all. In this case, the output impedance of one state would be no larger than R while the input impedance to the following stage would be no smaller than 100R. This would look a lot like isolation, pretty much as with the buffers.

This leaves us with E to examine, and it should show any essential results. With the feedback gain g, the denominator of the transfer function would become:

$$D(s) = s^4 + 7s^3 + 15s^2 + 10s + (1 - g)$$

and this is all we need to solve for the poles and assess stability. Clearly when g=1, we have a pole at zero (a pole at DC). In addition, note that if g is equal to or greater than 1, the system is unstable as the lasts term becomes negative, and the Routh stability criterion does not allow any sign changes in the denominator. Any negative values of g would not violate simple Routh, but we then need to calculate the poles to determine stability or not.

[ As a sort of aside, note that we are calling the negative values of g POSITIVE feedback. No feedback is either positive or negative until the phase around the loop is determined. For example, in the Moog four-pole (Fig. 1, D) we obtained poles which are, as a function of g, moving from s=-1 (four of them) outward from s=-1 at the corners of a square. Each stage of the length-4 cascade, at ω=1 thus has a gain of $1/\sqrt{2}$ and a phase of 45º, for each of the four poles. This is a total gain of 1/4 and total phase of 180º. By setting g=-4, we have a feedback gain of +1. ]

Here we will first see what happens with positive values of g (Fig. 2) and then with negative values of g (Fig. 3) and finally make a combined display in Fig. 4. Here we are simply setting up a scan of values of g to be tested and calculating and plotting the poles using Matlab's ***roots***. In Fig. 2, we have the four g=0 poles (all real and negative) plotted in green. As soon as g goes to +1, we have the pole that was closest to s=0 moving exactly to s=0 (plotted in blue, with three other poles that are still stable). We are kind of finished at this point. FOR example, with g=+2, we have the light blue poles, one of which is well inside the right half plane. Here in fact note that two poles have met on the real axis and split into a complex conjugate pair. Fig. 3 shows the case of g running from 0 to +20. This reminds us more of the Moog pole locus. In this case the feedback draws the real poles together in two pairs which then split to complex conjugate pairs. The pole pair that are least negative will dominate (as with the Moog

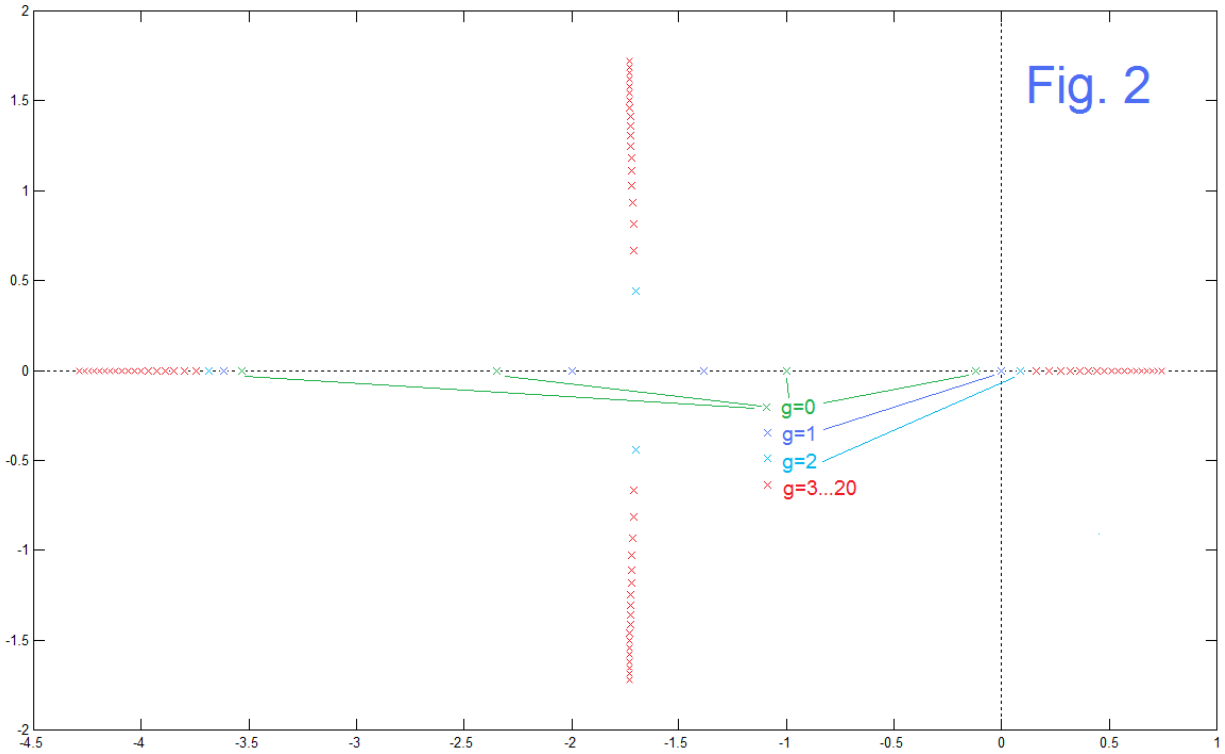


Fig. 2
g=0
g=1
g=2
g=3...20

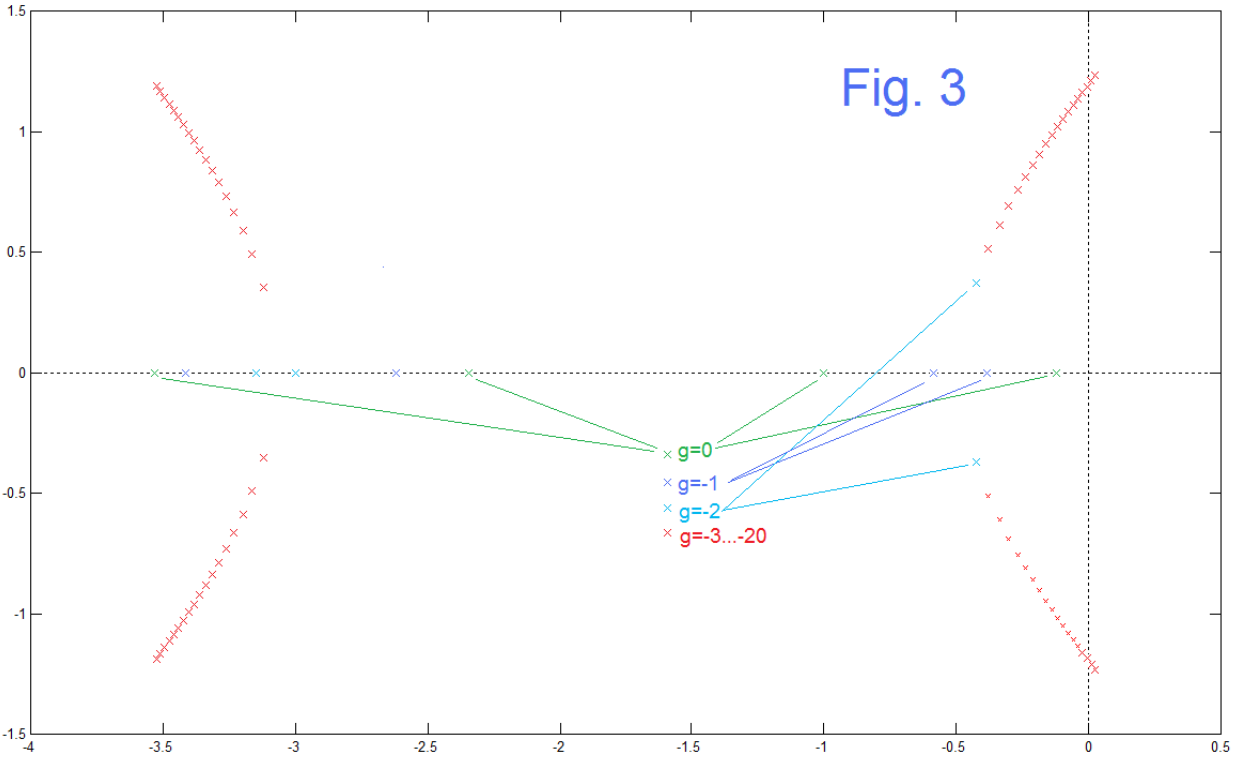
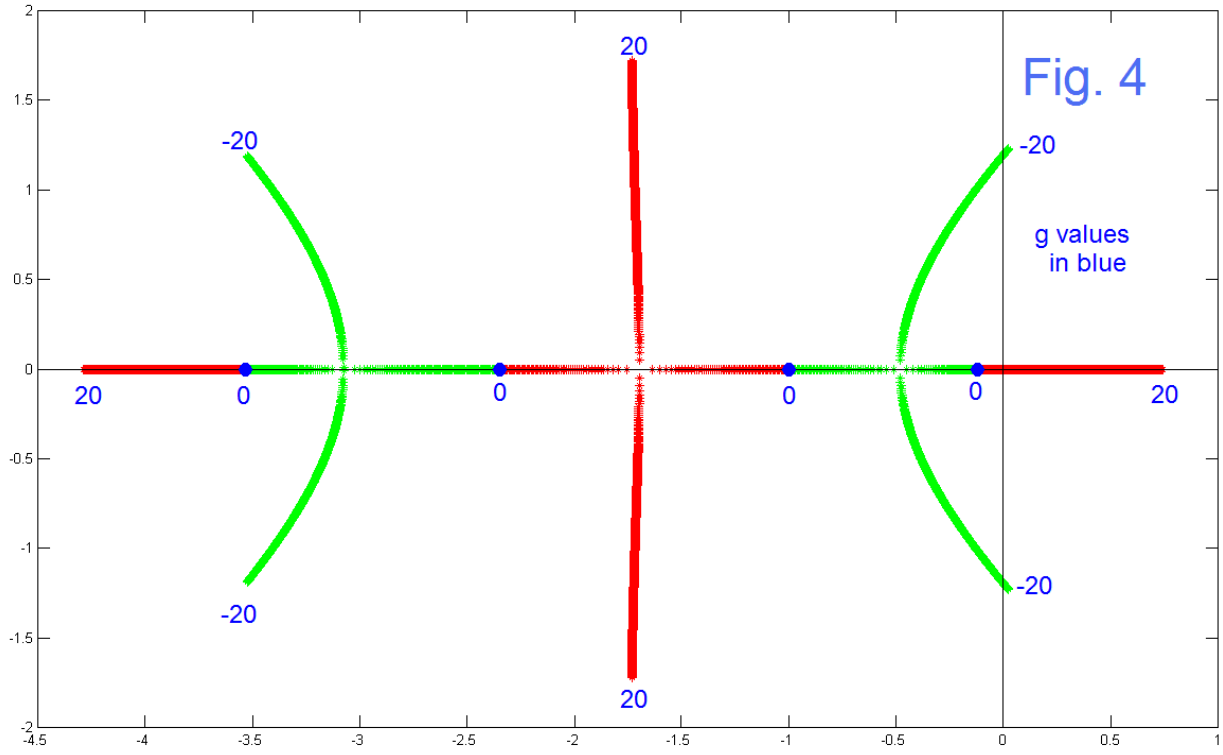


Fig. 3
g=0
g=-1
g=-2
g=-3...-20

circuit). This pair are joined by the time we arrive at g=-2, and swing around, approaching the jω axis (plotted in red). In fact, we rely on the root-finder program to show that the poles are still stable by g=-18 but have crossed to the right half-plane by g=-19.

Although we do not anticipate using this arrangement, basically because any voltage-controlled case would certainly offer free buffering, the plot showing the full sweep, with a finer increment, is attractive (Fig. 4). This is a combination of Fig. 2 and Fig. 3. It is one of those cases where we see pairs of real poles joining and splitting into complex pairs. Which real pole (from the right of the left) goes up or down! A meaningless question apparently. In Fig. 4 we can enjoy tracing any pole starting at g=20, following a red path to g=0, having it turn green and proceed to g=-20. Note that you can only use a path once. And, there are exactly the set of paths needed. Fun.



## References


[1a] B. Hutchins, “Looking Again at the RC Low-Pass”, ***Electronotes***, Vol. 22, No 210, May 2012 http://electronotes.netfirms.com/EN210.pdf

[1b] B. Hutchins, “Revisiting Some VCF Ideas – and a Few New Ideas”, ***Electronotes*** Vol. 23, No. 215 May 2013 http://electronotes.netfirms.com/EN215.pdf

[2] http://electronotes.netfirms.com/EN97VCF.PDF

[3] B. Hutchins, “Additional Design Ideas for Voltage-Controlled Filters,” ***Electronotes***, Vol. 10, No. 85, January 1978

[4] R. Bjorkman, “A Brief Note on Polygon Filters***,” Electronotes***, Vol. 11, No. 97, January 1979

[5] B. Hutchins, “The Migration of Poles as a Function of Feedback in a Class of Voltage-Controlled Filters,” ***Electronotes***, vol. 10, No. 95, Nov. 1978.

[6] B. Hutchins, “Cascading Voltage Dividers – and Friends”, Electronotes Application Note No. 429, Aug. 1, 2016 http://electronotes.netfirms.com/AN429.pdf