

July 30, 2014

NOTCH FILTERING – SOME NOTES

THE CLASSIC R-L-C NOTCH – Adding LP+HP

Traditionally, engineering students (of all flavors) usually have a required course, called something like “electrical science”, at the sophomore level. Often such a course has a major effect on a student’s choice of the particular field of engineering – sometimes toward, and sometime away from EE. Typically the material looks more like physics and mathematics and less like “hobby” electronics. But likely there was at least one circuit that tied the transform math and physics together with such apparently useful notions as “filtering” and this was even fun, particularly as this circuit often was compared to simple mechanical networks. Even differential equations made some sense in this presentation. I am talking about the R-L-C circuits: the simplest possible combinations of a resistor (R), an inductor (L), and a capacitor (C).

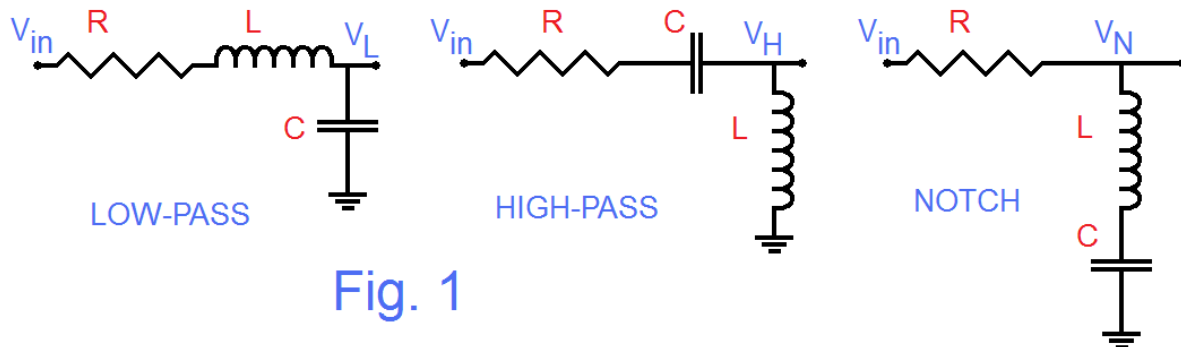


Fig. 1

Fig. 1 shows three typical R-L-C series circuits. We do not propose to go over the solution which we assume is familiar or readily available. We remind the reader that the order of the three components (R, L, and C) make no difference to the current that flows through the series. The only difference is the component (or components) in the lower output leg. Traditionally, the explanation of the Low-Pass (LP) is that the inductor tends to block high frequencies while the capacitor tends to short any residual high frequencies to ground. The High-Pass (HP) has the capacitor passing high frequencies (blocking low ones) while the inductor shorts any residual low-frequencies to ground. The notch, where the output leg has both the L and the C was more mysterious – something to do with resonance. It shunted the resonant frequency to ground.

Here our preferred analysis mode would be to use the complex impedances and the notion of a voltage-divider (to be mentioned below). Keeping this in mind, we have one of our main themes – that the notch is the sum of a LP and a HP since both the L and the C

are in the output leg. We note here that our choice of the three filters in Fig. 1 was to make just that point. In a more traditional presentation, we would have likely had the R-L-C series with the resistor R in the output leg as our third example: a Band-Pass (BP). Or an introduction might have used a parallel L-C.

We might well have considered the idea that a notch filter or at least a band-reject filter (blocking a range of frequencies, not just one) would be accomplished by using a LP and a HP. Or, we might have said that you could get a notch by subtracting a BP from unity. It is more subtle than this though, largely because we have to consider phase and we can't just add responses. In particular, a sharp notch can be achieved not due to a frequency space between a LP and a HP but because the LP and HP responses cross each other (same magnitude) but are perfectly OUT of phase. This was probably explained first as having a "complex impedance" going to zero.

In any event, it is easy to write down the transfer functions of Fig. 1 as simple voltage-divider observations:

$$T_L(s) = \frac{V_L}{V_{in}} = \frac{\frac{1}{LC}}{s^2 + s\left(\frac{R}{L}\right) + \frac{1}{LC}} \quad (1a)$$

$$T_H(s) = \frac{V_H}{V_{in}} = \frac{s^2}{s^2 + s\left(\frac{R}{L}\right) + \frac{1}{LC}} \quad (1b)$$

$$T_N(s) = \frac{V_N}{V_{in}} = \frac{s^2 + \frac{1}{LC}}{s^2 + s\left(\frac{R}{L}\right) + \frac{1}{LC}} \quad (1c)$$

and although not shown, if R is used in the output leg (BP) the transfer function would be:

$$T_B(s) = \frac{V_B}{V_{in}} = \frac{s\left(\frac{R}{L}\right)}{s^2 + s\left(\frac{R}{L}\right) + \frac{1}{LC}} \quad (1d)$$

These are perfectly good transfer functions, although we often see them modified to a standard form so that the denominator becomes:

$$d(s) = s^2 + Ds\omega_0 + \omega_0^2 \quad (2a)$$

where D is the "damping (1/Q) and

$$D = R \sqrt{\frac{C}{L}} \quad (2b)$$

and

$$\omega_0^2 = \frac{1}{LC} \quad (2c)$$

such that they have poles (roots of equation (2a) at (see also Appendix A):

$$p_{1,2} = \frac{-D\omega_0}{2} \pm \frac{j\omega_0}{2} \sqrt{4 - D^2} \quad (2d)$$

Note that when $L=1$ and $C=1$ (a common normalization) we have simply $D = R$. This is convenient, so while we could proceed here and convert everything to the standard form, it is perfectly good to just use the original transfer functions, and this we will do.

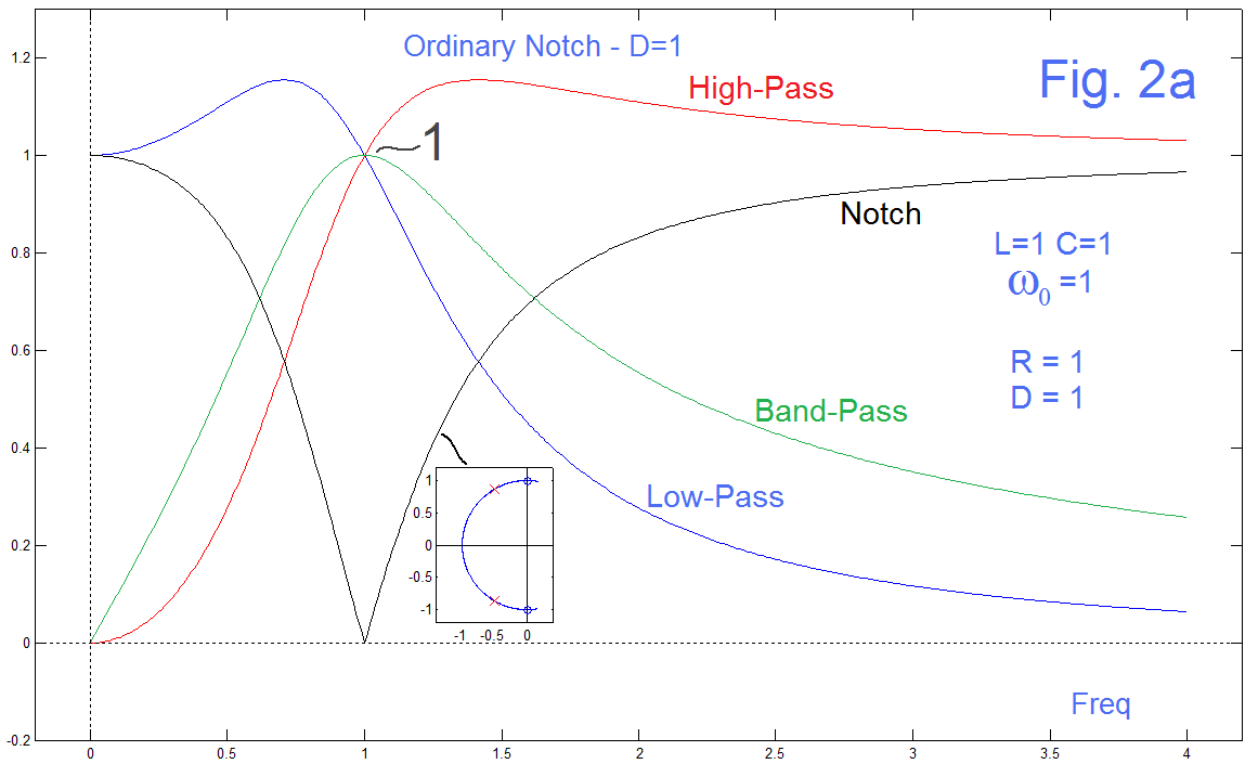


Fig. 2a shows frequency response magnitudes for the LP, HP, Notch, and BP according to equations (1a, 1b, 1c, and 1d) respectively with $L=1$, $C=1$ ($\omega_0=1$), and $R=1$ ($D=1$), all done with Matlab's *freqs* function. The value $D=1$ is somewhat underdamped ($D=2$ is critically damped, $D=\sqrt{2}$ is Butterworth) so there is a bit of peaking in the LP and HP. Another advantage of this choice is that the magnitude response of the LP and HP (and indeed the BP) is 1 at $\omega=1$. Since we have said that the notch is the summation of the LP and the HP, and the response of the notch at $\omega=1$ is 0 (not 2!) it might appear that we should have said the subtraction. The addition is correct however because it is the magnitudes that are equal –but the two response are exactly out of phase. Another way to see this is that the numerator of equation (1c) has zeros at $s = \pm j$. These zeros at $\omega = \pm 1$ are independent of D . The LP and HP magnitudes cross there (are equal) and are out of phase regardless of their particular common magnitude. That's what a notch is about.

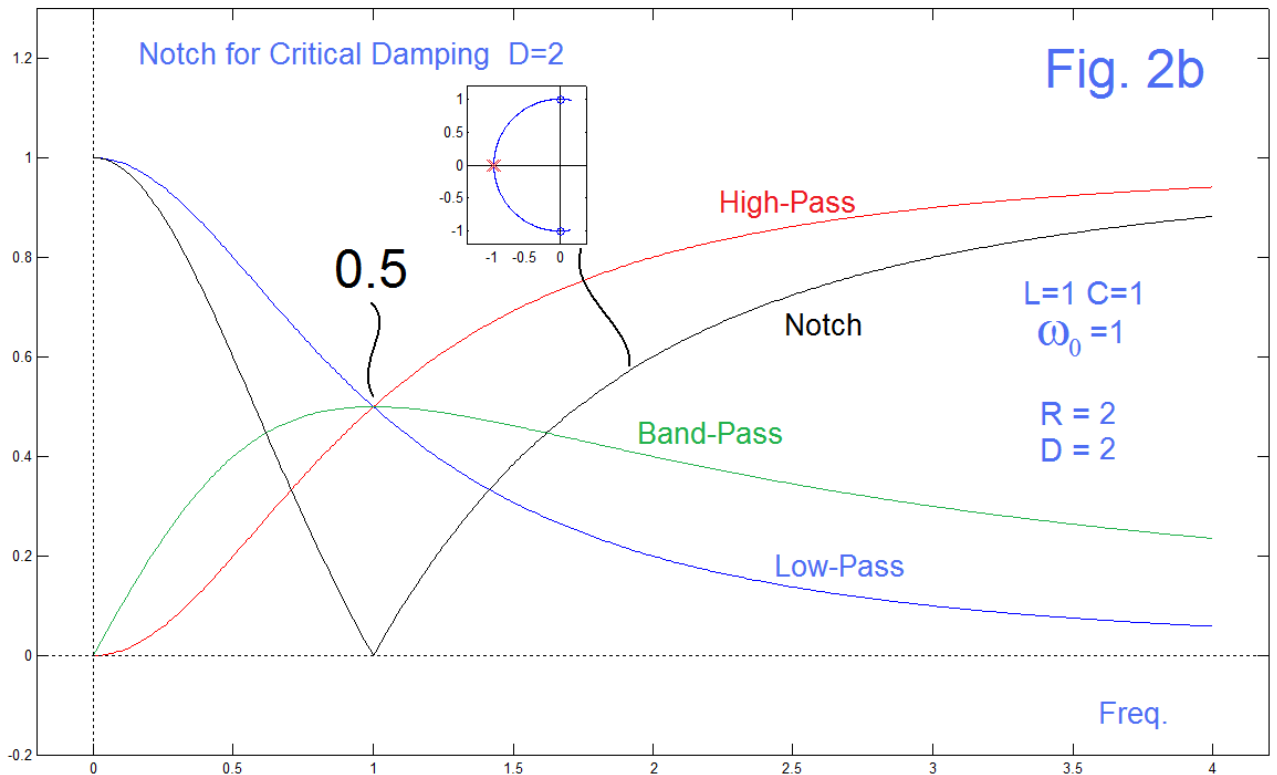


Fig. 2b is similar to Fig. 2a except we have changed R to 2, and thus $D=2$, and now the LP and HP overlap at a magnitude of 0.5, but cancel in the notch sum just the same. This value of $D=2$ ($Q=1/2$) is the so-called critically damped case. [Keep in mind that reaching critical damping is the point where time-overshoot ends. The frequency domain “ripple” disappeared at $D=\sqrt{2}$ (Butterworth).] As we are about to suggest an alternative path to a notch by subtracting a BP from a constant, we have also put the second example ($D=2$) here as a way of reminding ourselves that the BP peak is not in general unity.

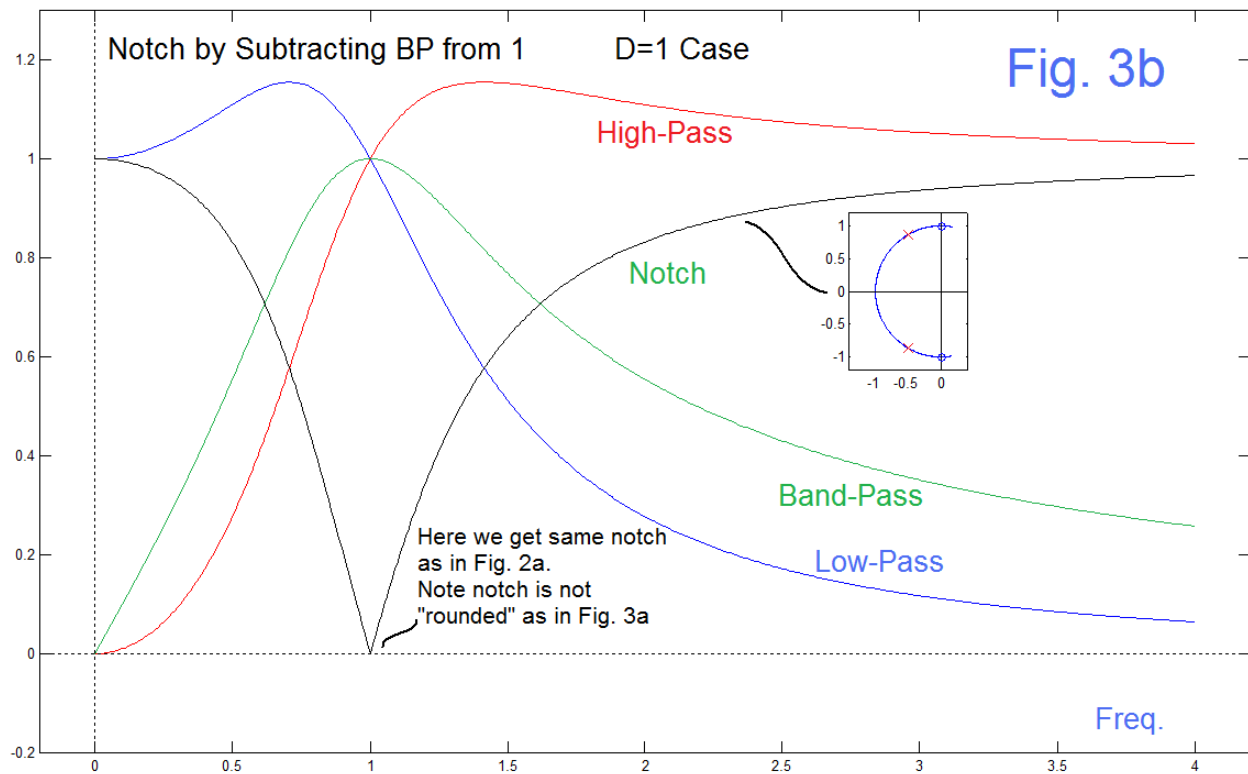
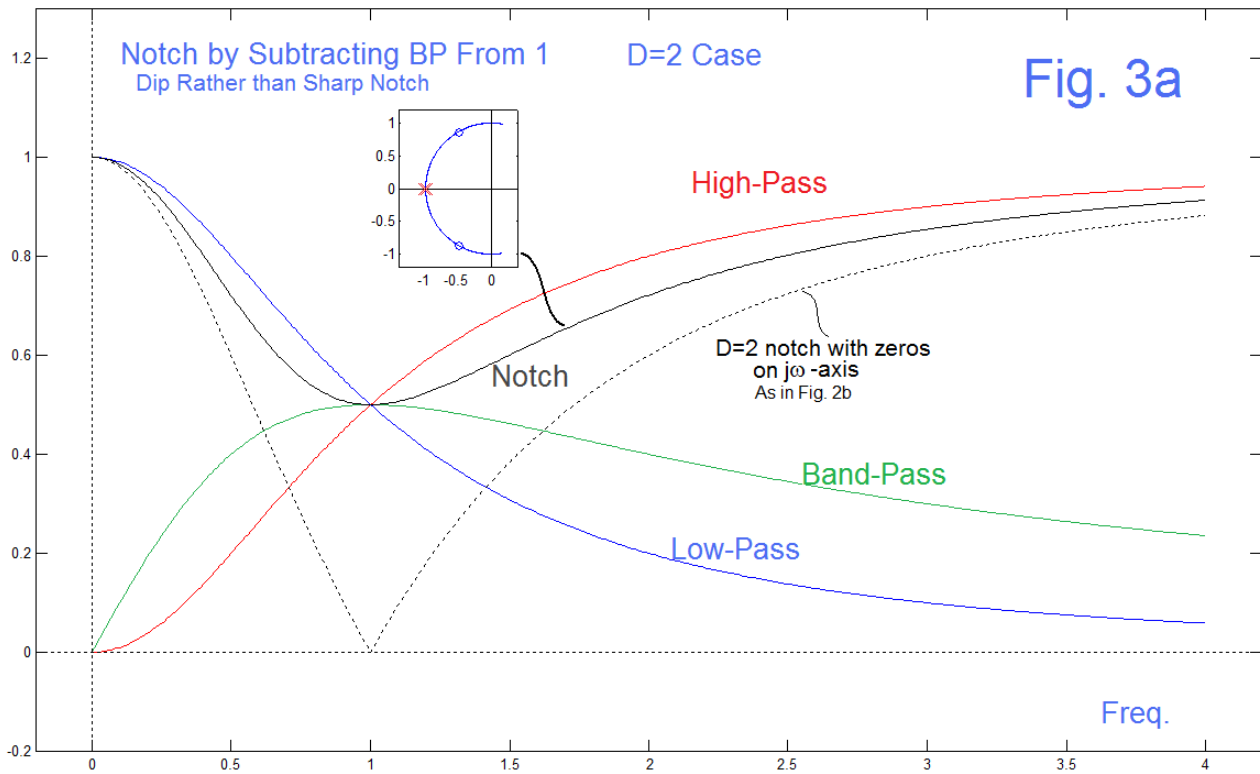
Subtracting a Band-Pass to form a Notch

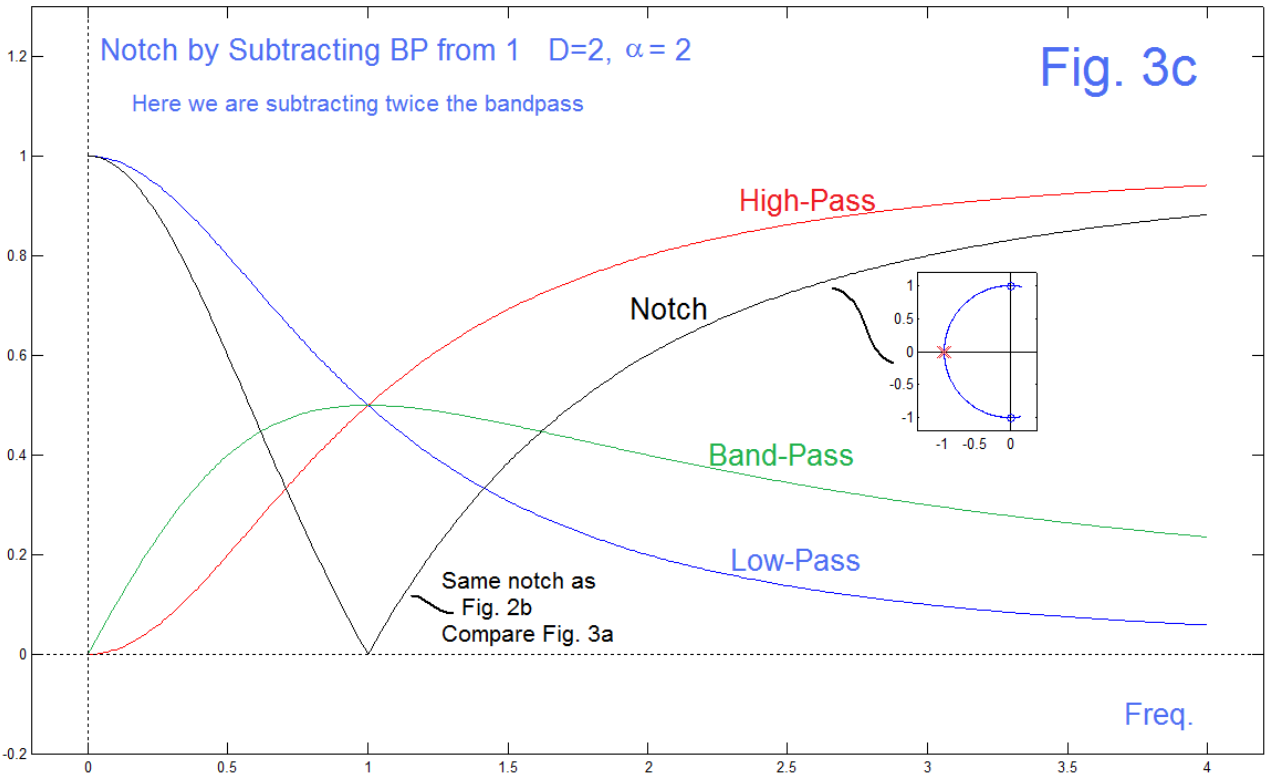
In many ways we tend to think of a notch as being the opposite of a BP filter. The BP favors one frequency while the notch rejects one frequency. We can almost see a subtraction in the works here. However, we always need to keep in mind that in adding or subtracting AC signals we have to worry about phase. First, what is the peak of the band-pass $T_B(s)$ where:

$$T_B(s) = \frac{s\omega_0}{s^2 + Ds\omega_0 + \omega_0^2} \quad (3a)$$

or:

$$T_B(s = j\omega_0) = \frac{j\omega_0\omega_0}{-\omega_0^2 + Dj\omega_0\omega_0 + \omega_0^2} = \frac{1}{D} \quad (3b)$$





which we see agrees with what we see in Fig. 2a (D=1) and Fig. 2b (D=2). We are also proposing to form a notch $T_N(s)$ as:

$$T_N(s) = 1 - \alpha T_B(s) = \frac{s^2 + Ds\omega_0 + \omega_0^2 - \alpha s\omega_0}{s^2 + Ds\omega_0 + \omega_0^2} = \frac{s^2 + (D-\alpha)s\omega_0 + \omega_0^2}{s^2 + Ds\omega_0 + \omega_0^2} \quad (4)$$

which is a standard notch when $\alpha=D$ so that the coefficient of s in the numerator goes to 0.

In Fig. 3a, $D=2$ and $\alpha=1$, so we do not have a proper notch. Another way to say this is that we have not subtracted enough BP. We see that we have instead a shallow dip to $1/2$ at $\omega=1$. Looking at the BP (green) it does look a bit like the proposed notch (black) is unity with the BP subtracted, but a closer look says it isn't that exactly. In fact, by solving for the poles and zeros of equation (4) we have both the poles at $s=-1$ (as we expect for $D=2$, critical damping), but the zeros are at $s=-0.5 \pm 0.866j$, not at $s=\pm j$ as we expect for a proper notch. Note that the zeros are at the correct radius of 1.

Fig. 3b is a surprising result. We make $D=1$ again as in Fig. 2a. Looking at the BP from Fig. 2a (replotted in Fig. 3b) we might expect a flat-bottomed notch response like Fig. 3a, except reaching zero. [The first time I tried this, I thought I plotted something wrong!] What happened? Well making $D=1$ and leaving $\alpha=1$, we have thereby set $D=\alpha$ and have the required condition that the zeros are at $s=\pm j$, the proper notch. In fact Fig. 3b is not

flat-bottomed at all, but sharp, exactly the same as Fig. 2a. Fig. 3c checks this idea by making $D=2$ again (critical damping) but now setting $\alpha=2$, and we again get zeros at $s=\pm j$. The lesson is, that we get the same final result whether we use the notch=LP+HP or the notch=1-BP (with proper weighting). We probably should have expected this.

The notches we have found and perhaps prefer have the zeros at $s=\pm j$, and thus have nulls at $\omega=\omega_0$, while the poles are on a circle of radius ω_0 . This means that the response levels, on either side of the notch, are equal (for DC and for high frequencies). Nothing prevents us from moving the poles so that the sides are unequal, or moving the zeros off the $j\omega$ -axis, if this results in a response that is more suitable in our application than what we have called here the “proper” notch.

The “State-Variable” Notch – Step Response

We started out with a classic R-L-C network, and then more or less abandoned it when we tried (successfully) to form a notch from a BP. Still we want to get rid of the inductor (the L) because in many cases (like audio frequencies), inductors are bulky, heavy, and non-ideal. Traditionally we use active elements (op-amps) to get rid of the inductors – so-called “active filters”. Many active notch networks have been described and used [1 and Appendix D], and we often find we only need a single op-amp for a second-order notch equivalent to the R-L-C. An exception is the popular “state-variable” (SV) circuit that uses three (or more) op-amps which is of interest (like in our music synthesizers) because it offers three or more filter type simultaneously (like LP, BP, HP, and Notch). Here this SV structure will be very convenient for our continuing analysis. Fig. 4 shows the structure.

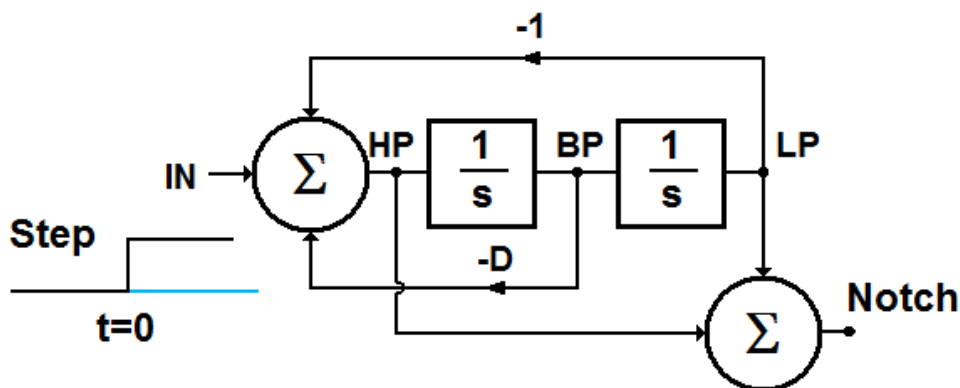


Fig. 4

The SV structure is just a summer and two integrators (the $1/s$ boxes) in a loop, and the Notch is formed by summing the LP and HP. [In music synthesizers this second summer was sometime done with two resistors of $2k$ each (averaging), providing the correct weighting and the desired $1k$ output impedance.] Here we want to just have this structure in mind as a basis for examining the step response of the notch by simulation. The step response of a notch filter has recently become an issue [2] in as much as it had mistakenly been claimed to be non-causal, while the step response should be and is of course causal (like all physical filters).

For our study here, we will simulate the network of Fig. 4. We start with all the voltages V_H , V_B , V_L , and $V_N=0$ with a step arriving. We then increment the simulation as in the Matlab code snippet here, for critical damping $D=2$:

```
VH=zeros(1,5000);
VB=zeros(1,5000);
VL=zeros(1,5000);
dt=0.002
D=2
for n=2:5000
    VL(n)=VL(n-1)+VB(n-1)*dt;
    VB(n)=VB(n-1)+VH(n-1)*dt;
    VH(n)=-VL(n-1)-D*VB(n-1)+1;
end
VN=VH+VL;
figure(1)
plot([0:4998],VH(2:5000),'r')
hold on
plot([-1000 0],[0 0],'c')
plot([0 6000],[1 1],'c')
plot([0:4998],VL(2:5000),'b')
plot([0:4998],VN(2:5000),'k')
plot([0,5000],[0 0],'k')
plot([0 0],[-1 2],'k')
plot([0 0],[0 1],'c')
axis([-1000 5100 -0.2 1.2])
hold off
figure(1)
```

Fig. 5 shows the result of this simulation with this code. Note that the Notch step response is indeed causal and is the sum of the HP and LP step responses, as we have said. A built-in Matlab function **step** can also be used to verify the calculation, and this is shown in Fig. 6, where we have shown an over-damped case and an under-damped case as well as the critical damping. A very nice calculator for this problem is available online [3] and gives the same result ($L=1$, $C=1$. and $R=2$).

An Experiment

Above we have shown some equations and some plots. To paraphrase Feynman, none of the theory matters if the experiment disagrees. Fig. 7 shows an experimental circuit and a photo of the scope trace showing a step response.

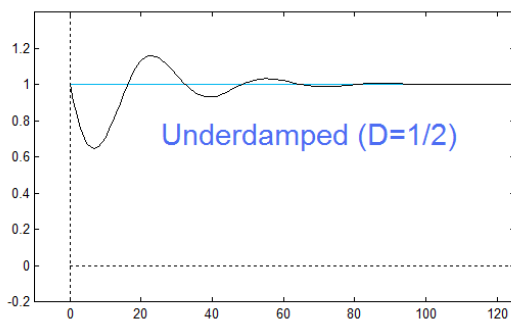
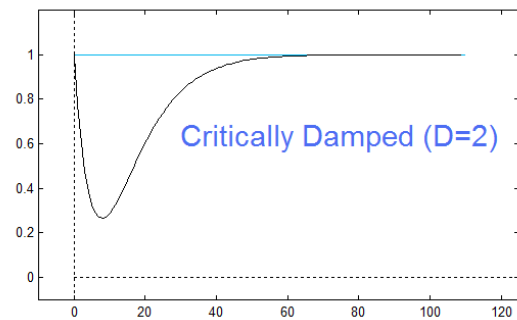
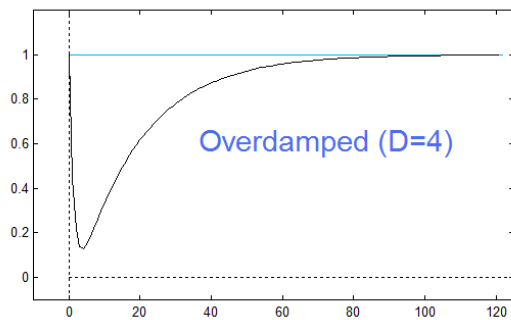
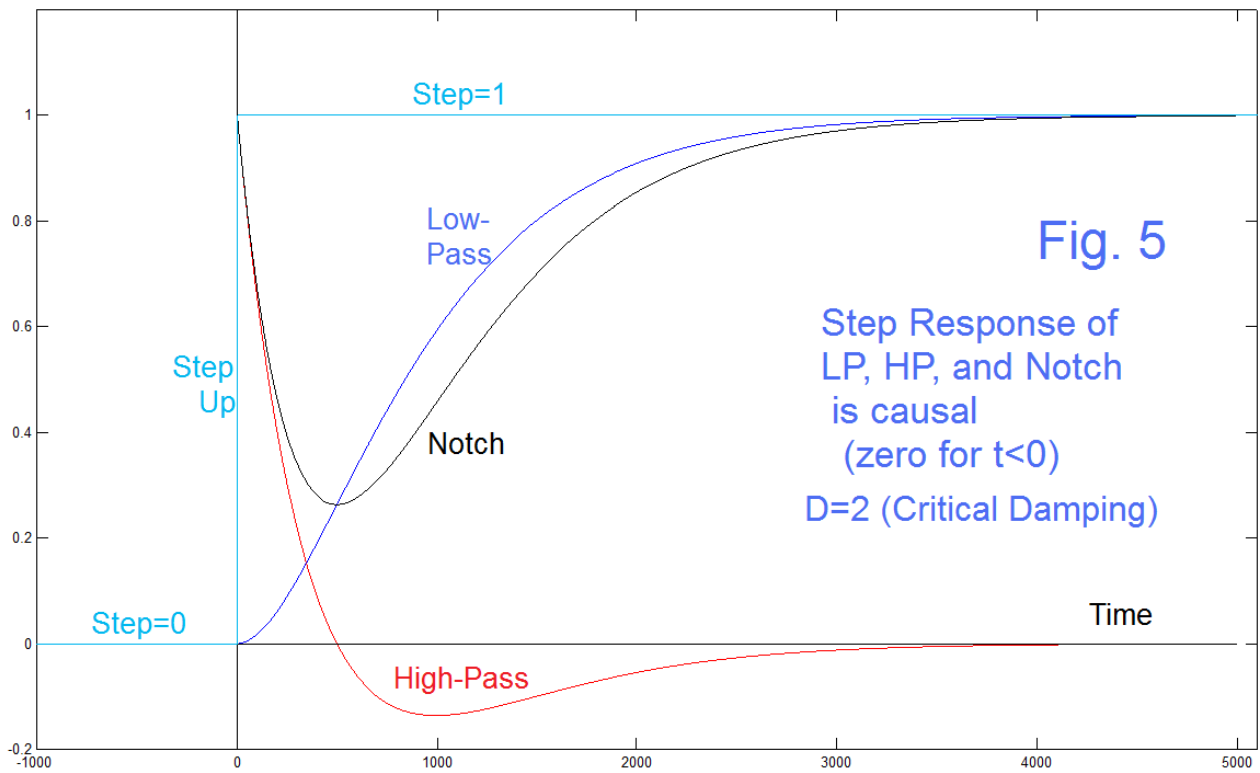


Fig. 6
 Step Responses of Notch Filter

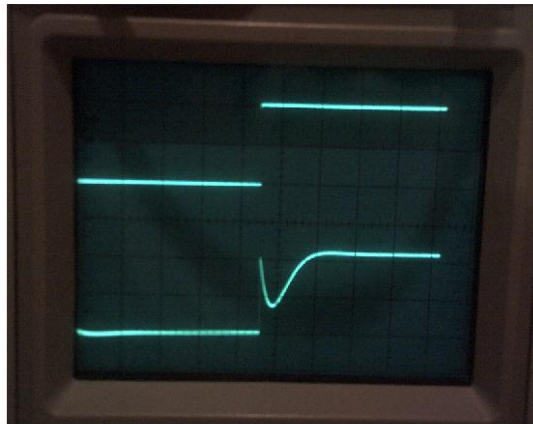
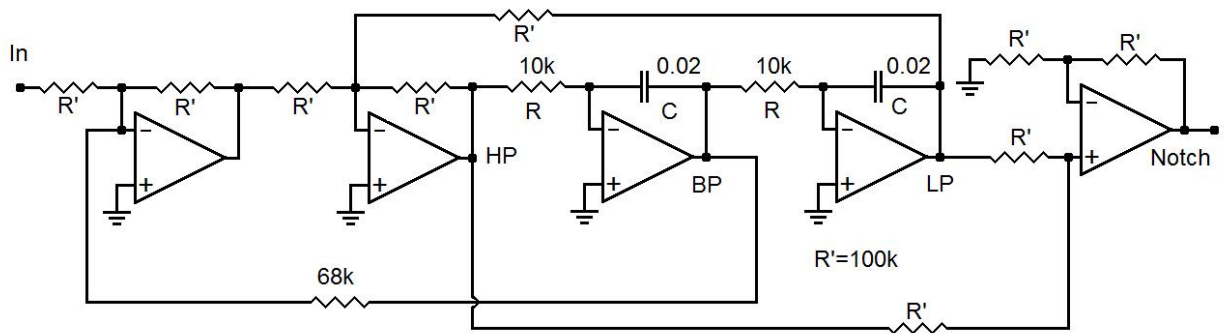


Fig. 7

Experiment
Test Circuit
 $D=100/68 = 1.47$

The circuit I used is likely very familiar to readers of these notes, as it is the SV configuration similar to our electronic music modules. I did use a total of 5 op-amps. The first one is to simplify the feedback from the BP and the last one was to achieve a unity-gain notch summer. The three in the middle are the conventional SV. In order to make a better demo, I used a damping of 1.47 so that there would be ever so slight overshoot. Note that the step at the top arrives, and then the response begins, very much like the calculated plots. The experiment was not a surprise of course, but you have to show the actual experimental data so that there is no doubt the experiment was done.

The Mechanical Analog

The R-L-C electrical circuit has a mechanical analog that involves a mass, a spring, and a rough surface for friction (damper). Fig. 8 shows one such mechanical network, and is shown here to point out the similarity to our findings of the step response. In particular, we will become interested in the critical damping cases.

If the damping were slight, we can easily associate the LP mode with the notion of the mass as driven by the red-ball-driver, thus by displacement. The HP is associated with the elongation of the spring relative to an equilibrium position (hence by Hook's law with the force and thus with the acceleration). The BP is associated with the velocity of the mass. Hence Fig. 4 shows that acceleration, velocity, and displacement are correctly arranged by the integrators (HP, BP, and LP). Notch, being the sum of HP and LP is less easy to visualize. One of the huge advantages of a mechanical analog is that we can in general visualize it. In Fig. 8, we have eliminated gravity as an issue, but all of us have

played with a mass on a spring vertically (perhaps a rubber band standing in for a spring). So we can appreciate that if the red-ball-driver moves left \longleftrightarrow right rapidly, the mass M stands virtually still due to inertia (LP) while the spring's elongation changes with the drive (HP). If the red-ball-driver moves very slowly, the mass moves obediently and patiently in parallel (LP) while the spring pretty much maintains its equilibrium elongation (HP). We all know that if we move the driver at a frequency near resonance, the mass can not only move a large distance, but with considerable velocity (resonance endangering nearby flower vases!). That's a response to periodic drive.

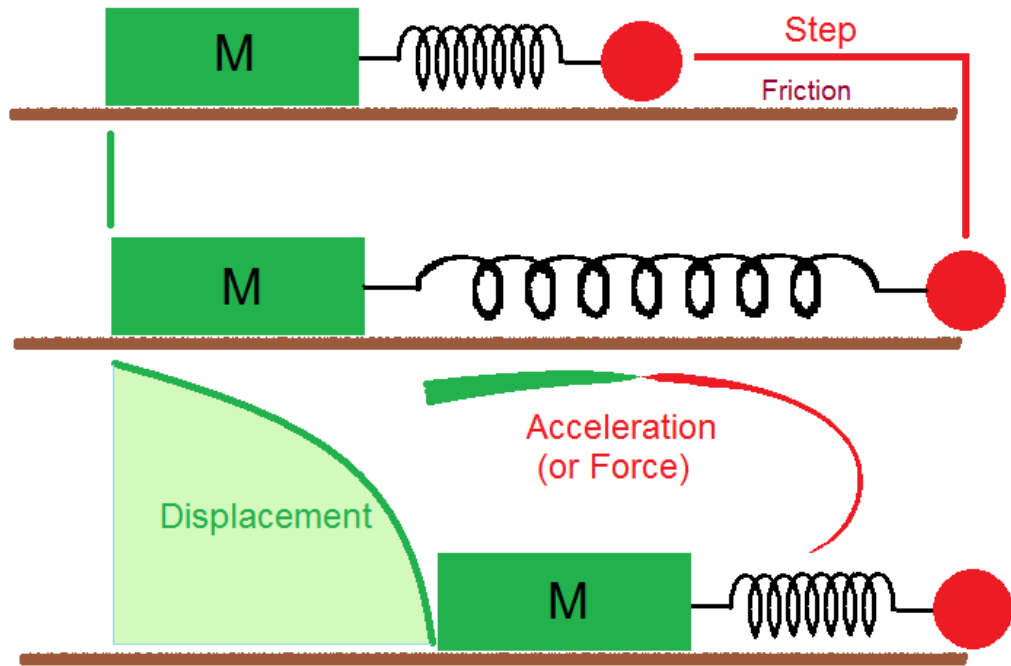


Fig. 8

Now, suppose we just move the red-ball-driver once in a step (Fig. 8). We envision the mass starting to move following the driver, and when the driver stops, we probably envision the spring compressing and then pushing the mass back, and so on. BUT we are here postulating critical damping (as in Fig. 5).

Things become confusing until we remember that the voltages (or numbers) in the SV system are representing actual physical quantities that are not necessarily voltages. (They might be electrical currents, for example.) In the mechanical system, the variables are not all displacements as is the LP (the mass), but may be velocity (or momentum) at the BP, and acceleration (or force) at the HP. This makes a drawing such as we are attempting in Fig. 8 very difficult, as we readily show only displacement in the diagram.

What does happen for critical damping is that the mass approaches but never reaches the actual level of the step. The force that is drawing the mass to the displacement of the step is being resisted by the friction that is drawing out the energy, and in turn, resisting the push-back as the mass tries to compress the spring. By far, critical damping (because of a double order pole) is more difficult to understand than under- or over-damping. For the step response of the notch, it begins with the full elongation of the spring (acceleration = HP) and then the displacement (LP) adds, as the spring contracts, as in Fig. 5.

We apparently have a good understanding of notch filtering the frequency domain, and know the basics of the time response to a step input. Some important additional details of computing the step response are found in Appendix B and Appendix C.

REFERENCES

[1] Here is a guide to finding some of the many presentations we have done on notch filtering over the years.

[a] Analog Signal Processing, Section 4-2 (notch filtering), *Electronotes*, Vol. 19, No. 193, March 2000, pp 6-10.

<http://electronotes.netfirms.com/EN193.pdf>

[b] Analog Signal Processing , Chapter 5, pp 9-10 of *Electronotes*, Vol. 20, No 194, April 2000 (a bit more on a general single op-amp M.F.I.G. approach).

<http://electronotes.netfirms.com/EN194.pdf>

[c] A whole series of Electronotes Application Notes were published in 1980

AN-183 “Elements of Notch Filter Design”, July 25, 1980

AN-184 “Selection of Notch and Bandstop Responses”, August 5, 1980

AN-185 “Notch Filter Using Simulated Inductor”, August 15, 1980

AN-186 “State-Variable Input-Notch Structure”, August 25, 1980

AN-187 “State-Variable Output-Notch Structure”, September 4, 1980

AN-188 “The Bootstrapped Twin-T Notch”, September 11, 1980

AN-189 “Notch Filter by Bandpass Subtraction,” September 18, 1980

and a follow-up App Note was:

AN-209 “Limited Depth Notch Filtering”, March 11, 1981

[2] See the link here:

<http://joannenova.com.au/2014/06/big-news-part-iii-the-notch-means-a-delay/>

as well as the many associated postings and updates.

[3] Online calculator (Try R=2, L=1, C=1)

<http://sim.okawa-denshi.jp/en/RLCbekeisan.htm>

APPENDIX A: Pole/Zero Plot (A Popular Case)

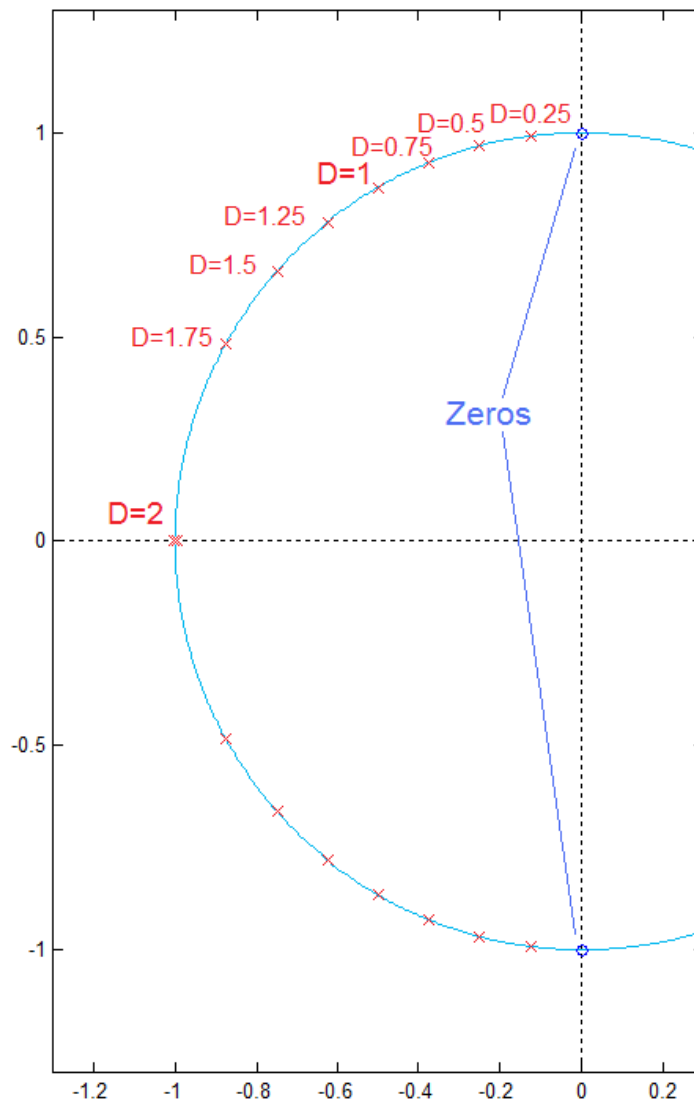


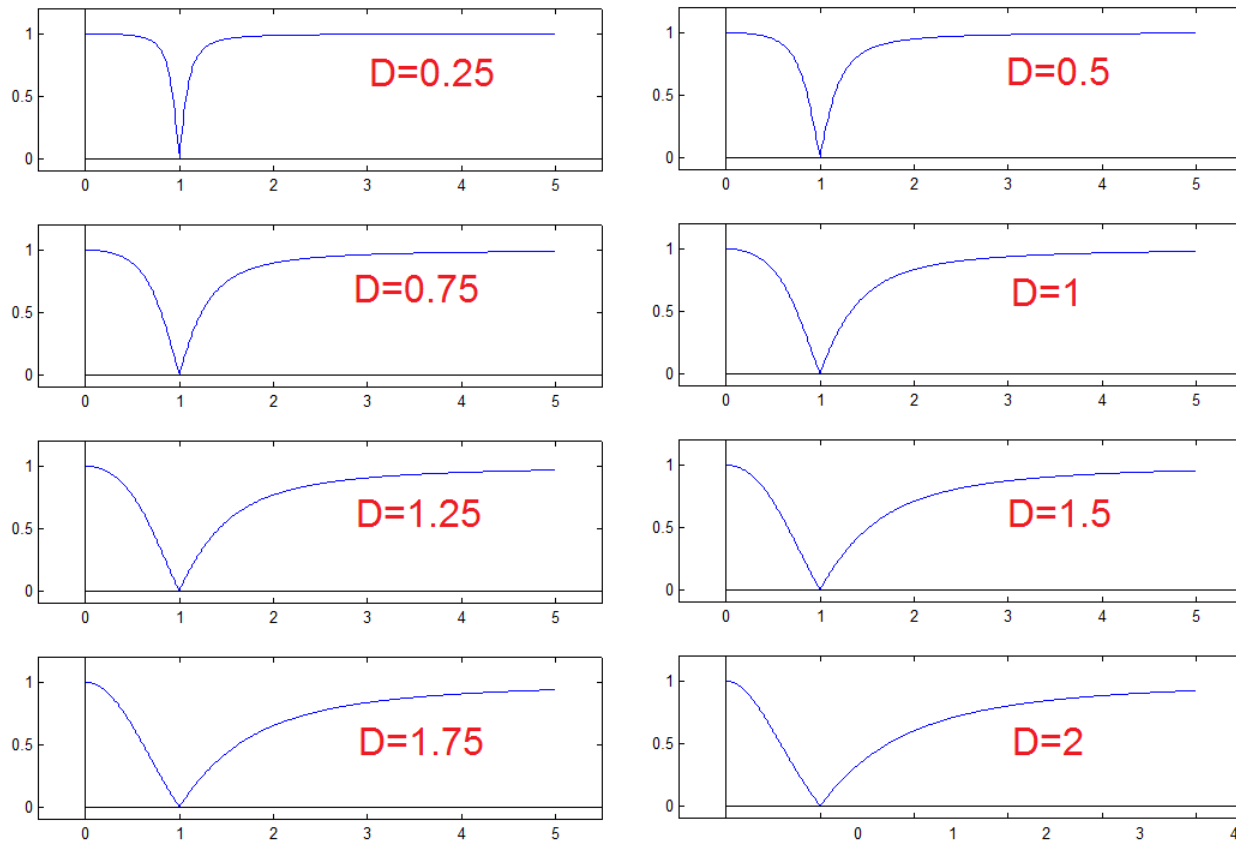
Fig. A1

Zeros of standard notch are at $s=+j$ and $s=-j$ and poles are on a circle of radius 1 if the frequency response levels are to be the same above and below the notch.

Here (Fig. A1) we remind the reader that many notch filters have a response that has exact nulls determined by $j\omega$ -axis zeros at $\pm j$ and a pair of poles at the radius of 1 in the negative half of the s -plane. The case $D=1$ is used in several examples above, as is the $D=2$ critical damping case which has a pair of real poles at $s=-1$. The popularity of the choice of zeros at $\pm j$ is in large part due to the fact that the “shelves” on either side of the notch tend to equal values, so that the notch is largely removing one frequency completely while tending to leave the remainder of the frequency response flat. As can be seen in Fig. A2, the poles have a sharpening effect on the notch as they approach the zeros. This is understood in terms of moving up the $j\omega$ -axis from zero, looking ahead to the pole and zero, which while being nearby to each other, still being relatively far away. The pole partially hides the zero until we get close to the notch frequency. Very far away (high frequencies) looking back two poles and two zeros cancel.

Frequency Responses

Fig. A2

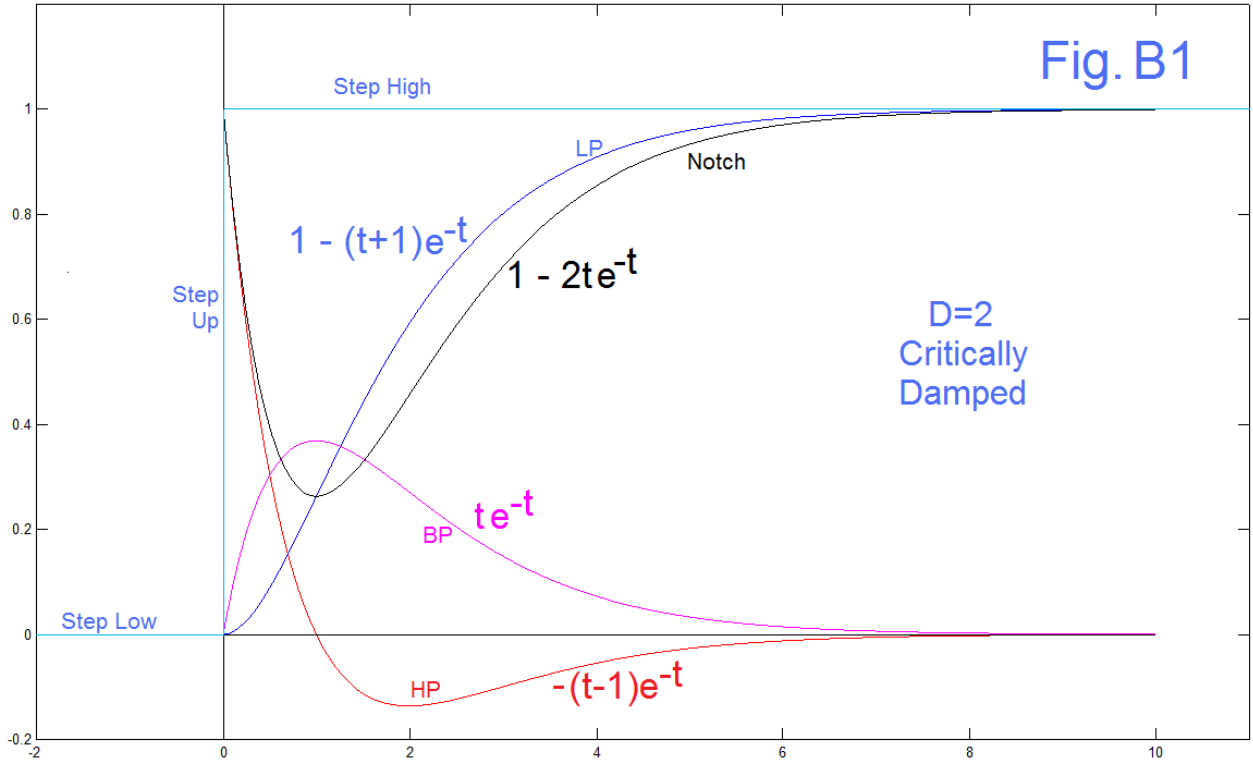


APPENDIX B: A Few “Closed-Form” Results

The photo in Fig. 7 is clearly an experiment, while Fig. 5 is a calculation, but not from closed-form, so is more closely related to an experiment. It is a simulation. One way we check a simulation is to see if it matches closed-forms where we have them. Fig. B1 shows the simulated curves of Fig. 5 with the closed form expressions added. This graph shows only the case of $D=2$, critical damping. Where did the closed-form expressions come from. From inverse Laplace transforms and various manipulations (and guesses). Ultimately the expressions were programmed and overplot the data of Fig. 5 (exactly). Portions of the Matlab code for Fig. B1 and for Fig. B2 and Fig. B3, is included below.

One thing probably occurs to the reader as to whether the closed-form expressions in Fig. B1 represent any sort of basis set to find a more general step response. This is true, at least for the critically damped case here. One example of the success is that the notch response is an equal sum of the LP and the HP. That is

$$T_N(s) = \frac{s^2 + \omega_0^2}{s^2 + Ds\omega_0 + \omega_0^2} \quad (B1)$$



This is a special case since we have used $D=2$ (critical damping), the zeros are exactly on the $j\omega$ -axis, and the poles and zeros have the same radii, ω_0 . We can however freely move ω_0 in or out – this is how we design a desired filter from a normalized prototype. We can make it less special (still keeping $D=2$ however) by putting the poles and zeros on different radii:

$$T_G(s) = \frac{s^2 + \omega_z^2}{s^2 + Ds\omega_p + \omega_p^2} \quad (B2)$$

and then we can allow the zeros to move off the $j\omega$ -axis with a more general form with choices of a , b , and c (b not required to be 0):

$$T_G(s) = \frac{as^2 + bs\omega_z + c\omega_z^2}{s^2 + Ds\omega_p + \omega_p^2} \quad (B3)$$

Note that if we were to divide ω_z and ω_p by ω_p in equation (B3) we would have a normalized and (with $D=2$) critically damped case, for some value of zero position other than the restricted case of $\pm j$.

$$T_G(s) = \frac{as^2 + bs(\omega_z/\omega_p) + c(\omega_z/\omega_p)^2}{s^2 + Ds + 1} \quad (B4)$$

In this case, the basis set of Fig. B1 should work. Fig. B2 shows the preliminary calculations from the blog entry mentioned [2] which has two real poles and zeros slightly in the left half-plane, the poles being further out than the zeros as well. We show the

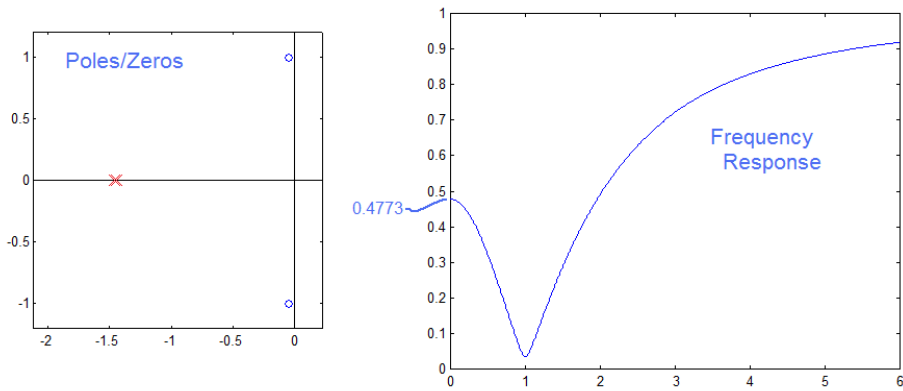


Fig. B2

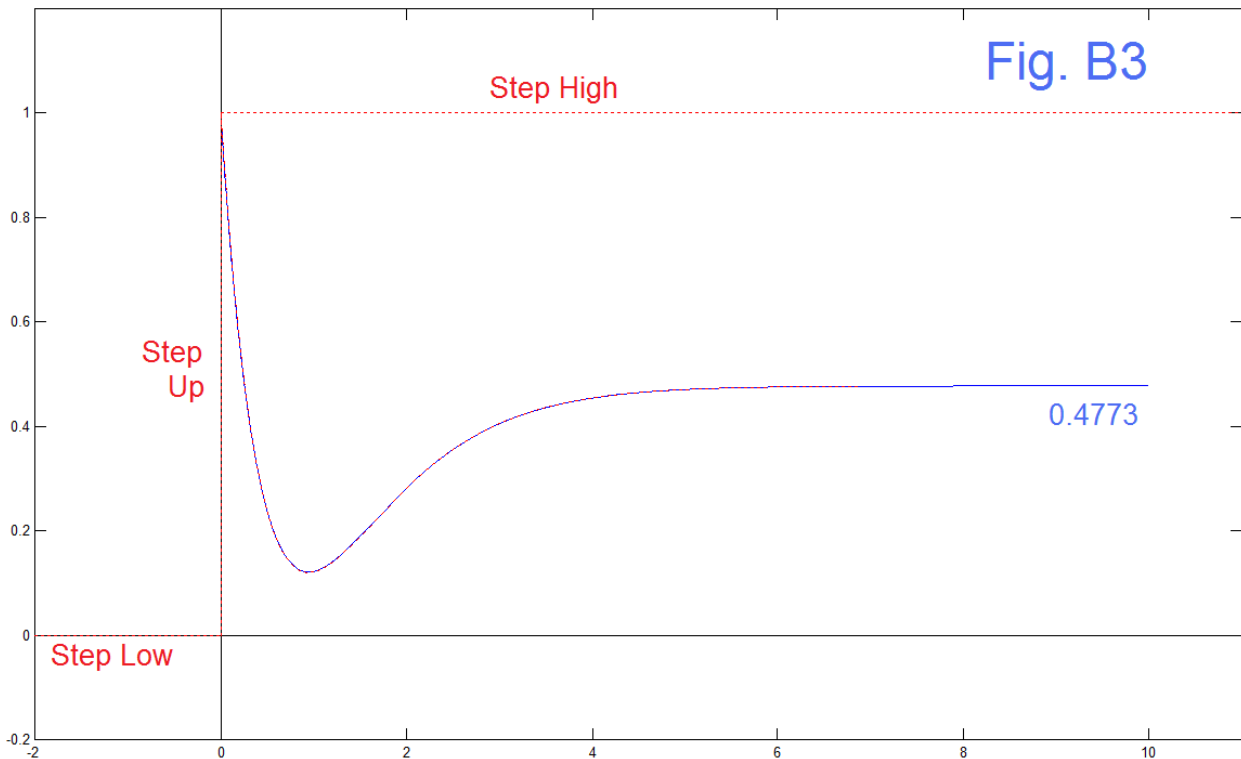


Fig. B3

pole/zero plot on the left with the frequency response (note unequal shelves) on the right. The dip in the response is very close to the radius of the zeros (1 in this case) so the time will be $1/2\pi$, so if the period corresponding to the notch frequency is 11 here, a time value of 1 will be 1.75. The blue step response in Fig. B3 is from Matlab's **step** function while the curve is also overplotted with dotted red representing the weighted sum of the closed-form components. The time axis is compressed (see program code) such that the red overplotting stops about time 7. Note that the step response converges on a value of 0.4773, which is appropriately, the DC response of the filter (Fig. B2, right figure).

Because the method is not general as presented here, but only used to verify that the Matlab **step** function agrees with a known closed-form, we give Matlab code here for completeness and to avoid any ambiguity.

The Matlab Code for Appendix B is here:

```
% stepbystep

% Step by Numerical Integration, D=2
VH=zeros(1,5000);
VB=zeros(1,5000);
VL=zeros(1,5000);
dt=0.002
D=2
for n=2:5000
    VL(n)=VL(n-1)+VB(n-1)*dt;
    VB(n)=VB(n-1)+VH(n-1)*dt;
    VH(n)=-VL(n-1)-D*VB(n-1)+ 1;
end
VN=VH+VL;
figure(1)
plot([0:4998]/500,VH(2:5000),'r')
hold on
plot([-2 0],[0 0],'c')
plot([0 12],[1 1],'c')
plot([0:4998]/500,VL(2:5000),'b')
plot([0:4998]/500,VN(2:5000),'k')
plot([0:4998]/500,VB(2:5000),'m')
plot([0,5000]/500,[0 0],'k')
plot([0 0],[-1 2],'k')
plot([0 0],[0 1],'c')
axis([-2 11 -0.2 1.2])
hold off

% Check Closed Forms
figure(2)
plot([0:4998]/500,VH(2:5000),'r')
hold on
plot([-2 0],[0 0],'c')
plot([0 12],[1 1],'c')
plot([0:4998]/500,VL(2:5000),'b')
plot([0:4998]/500,VN(2:5000),'k')
plot([0:4998]/500,VB(2:5000),'m')
plot([0,5000]/500,[0 0],'k')
plot([0 0],[-1 2],'k')
plot([0 0],[0 1],'c')
axis([-2 11 -0.2 1.2])
% check Kuo for Notch - dotted green
t=0:.01:10;
VNCheck = 1 - 2*t.*exp(-t);
plot(t,VNCheck,'g:')
VBCheck= t.*exp(-t);
plot(t,VBCheck,'g:')
VLCheck= 1 - (t+1).*exp(-t);
plot(t,VLCheck,'c:')
VHCheck= -(t-1).*exp(-t);
plot(t,VHCheck,'c:')
hold off
figure(2)
```

```

% Now use David's Poles/Zeros, both poles real and same, zeros not on j-omega
z=[-0.0047578+.0907845*j -0.0047578-.0907845*j]
p=[-.131579 -.131579]/0.0909091
n=poly([-0.0047578+.0907845*j -0.0047578-.0907845*j]/0.0909091)
no=n;
d=poly([-0.131579 -.131579]/0.0909091)
do=d;
z=roots(n)
p=roots(d)
% Matlab's step function
figure(3)
t=0:0.002:10;
y=step(n,d,t);
ymax=max(y)
[ymin,ix]=min(y)
tmin=(ix-1)*0.002
plot(t,y,'b')
yend=y(5000)
hold on
plot([-2 11],[0 0],'k')
plot([0 0],[-5 15],'k')
plot([-2 0],[0 0],'r:')
plot([0 0],[0 1],'r:')
plot([0 11],[1 1],'r:')
if ymin>0; axis([-2 11 -0.2 ymax+0.2]);end
if ymin<0; axis([-2 11 ymin-0.2 ymax+0.2]);end
% divide omegas by omega pole
div=d(3)
n(2)=n(2)/sqrt(div);
n(3)=n(3)/div;
d(2)=d(2)/sqrt(div);
d(3)=d(3)/div;
n
d
VNCheck=VH + VB*n(2) + VL*n(3);
% Note: d not actually used here
plot((0.0909091/0.131579)*[0:4999]/500,VNCheck,'r:')
hold off
figure(3)

% Supplemental - Check Freq Resp
T=abs(freqls(n,d,[0:.01:6]));
figure(4)
plot([0:.01:6],T,'b')
figure(4)
T(1)

T=abs(freqls(no,do,[0:.01:6]));
figure(4)
plot([0:.01:6],T,'b')
figure(4)
T(1)

```

Below is a portion of the printout of the Matlab code for Appendix B

```

dt = 0.0020
D = 2
z = -0.0048+ 0.0908i, -0.0048- 0.0908i
p = -1.4474, -1.4474
n = 1.0000 0.1047 1.0000
d = 1.0000 2.8947 2.0949
z = -0.0523+ 0.9986i, -0.0523- 0.9986i
p = -1.4474, -1.4474
yend = 0.4773

div = 2.0949
n = 1.0000 0.0723 0.4774
d = 1 2 1
ans = 0.4774
ans = 0.4774

```

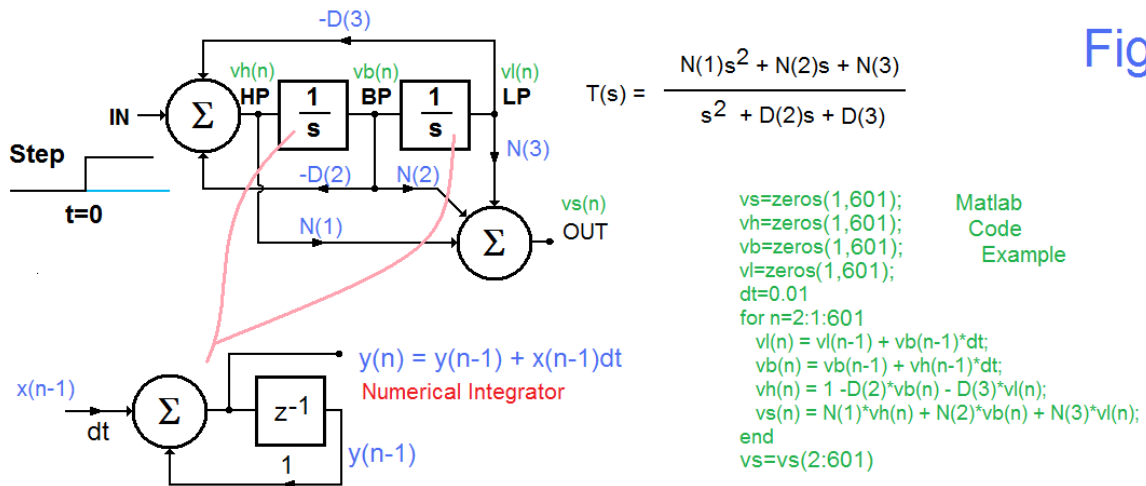
APPENDIX C: Matlab's *step* and Numerical Integration Compared

At this point we have established likely agreement between the Matlab *step* function and a known closed-form solution. Further we find the general use of “basis functions” for closed form is not immediately promising. When we look to the Matlab code we find that the script code for *step* is not available (as is the case for many built-in functions). What we do find is a suggestion that *step* simply uses numerical integration, the way we have done in our own code here. This suggestion is that when a user specifies the time vector, a parameter they call “dt” is the difference between consecutive times. Accordingly a goal here next is to write a general script similar to that used above to accommodate a general set of poles/zeros. That is, a kind of “reverse engineering” of the *step* function. This turns out to be simple and effective, and the whole procedure is summarized in Fig. C. The full code for this Appendix C is reproduced below, but note well that the core code is nothing more than the 12 lines of Fig. C (and more to the point, the 6 lines of the “for loop”).

The upper left of Fig. C is just the usual state-variable structure that solves transfer function in the Laplace domain. It modifies the earlier notch example of Fig. 4. Here we have replaced the continuous integrators with discrete integrators, which should not be a problem as long as we keep “dt” small enough. Setting dt smaller should not be a speed issue with most processors.

The result is virtually a complete success. This is evidenced by the near-perfect overprint of the results of our *step2* with Matlab's *step*. The agreement is better than the plotting error. We have computed and printed out the maximum error (see example printout below the code listing) and in the example the max error was 0.0032 for a step response on the order of 1 (or 0.6 at the end). It should be noted that it was necessary to shift the result of *step2* to the left by 1 step. This was because the iterative integration needs to reach back one step. This is just “bookkeeping” and does not matter overall.

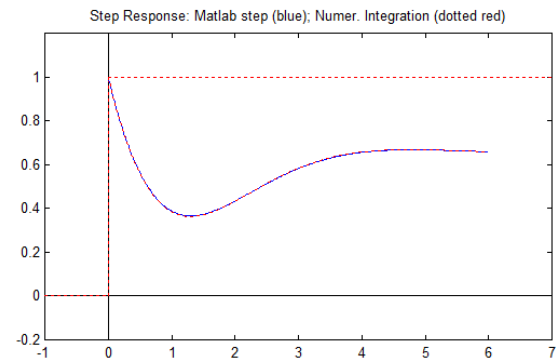
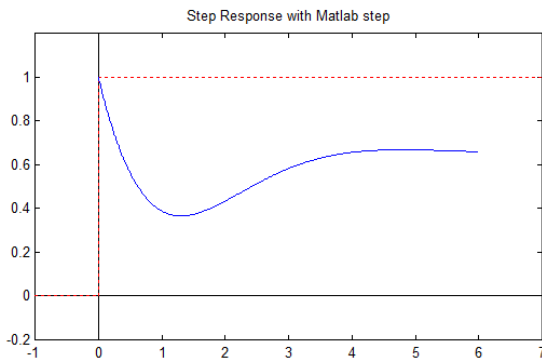
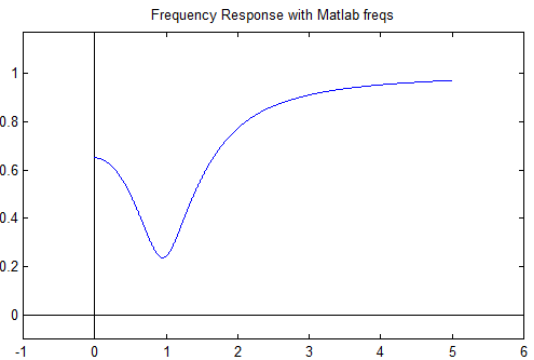
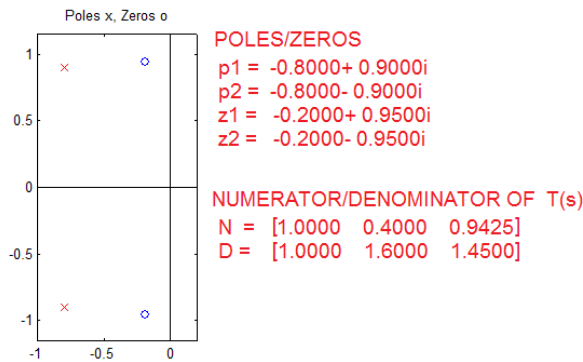
Fig. C



```

vs=zeros(1,601);
vh=zeros(1,601);
vb=zeros(1,601);
vl=zeros(1,601);
dt=0.01
for n=2:1:601
    vl(n) = vl(n-1) + vb(n-1)*dt;
    vb(n) = vb(n-1) + vh(n-1)*dt;
    vh(n) = 1 -D(2)*vb(n) - D(3)*vl(n);
    vs(n) = N(1)*vh(n) + N(2)*vb(n) + N(3)*vl(n);
end
vs=vs(2:601)
    
```

Matlab Code Example



The example used here just set zeros back away from the $j\omega$ -axis a bit ($-0.2 \pm 0.95j$) with sharpening poles at $-0.8 \pm 0.9j$, an arbitrary choice, with results as shown.

As with Appendix B, the code here is offered with many features that were for development and testing. For example, the comparison plot of figure(3) of the code (not shown) is not supposed to work unless both poles are real and the zeros are on the $j\omega$ -axis, or the accommodation as in Appendix B is made. We emphasize the simplicity of the code in Fig. C, and expect that the procedure could be easily written in just about any other language beside Matlab.

Here is the code relating to Appendix C:

```
% step2
%p1 = -1
p1=-0.8 + 0.9*j
p2 = p1'
z1 = -.2 +0.95*j
z2 = z1'

figure(1)
subplot(221)
plot([-10 10],[0 0],'k')
hold on
plot([0 0],[-10 10],'k')
plot(real(p1),imag(p1),'rx')
plot(real(p2),imag(p2),'rx')
plot(real(z1),imag(z1),'bo')
plot(real(z2),imag(z2),'bo')
maxrpz=max(real([p1 p2 z1 z2]))
minrpz=min(real([p1 p2 z1 z2]))
maxipz=max(imag([p1 p2 z1 z2]))
minipz=min(imag([p1 p2 z1 z2]))
if maxrpz<0; maxrpz=0; end
axis('equal')
axis([minrpz-0.2 maxrpz+0.2 minipz-0.2 maxipz+.2])
title('Poles x, Zeros o')
hold off

N=poly([z1 z2])
D=poly([p1 p2])
T=abs(freqs(N,D,0:.01:5));
subplot(222)
plot([-1 6],[0 0],'k')
hold on
plot([0 0],[-1 10],'k')
plot([0:.01:5],T,'b')
axis([-1 6 -.1 max(T)+.2])
hold off
title('Frequency Response with Matlab freqs')

s=step(N,D,[0:.01:6]);
subplot(223)
plot([-1 7],[0 0],'k')
hold on
plot([0 0],[-10 10],'k')
plot([-1 0],[0 0],'r:')
plot([0 0],[0 1],'r:')
plot([0 7],[1 1],'r:')
plot([0:.01:6],s,'b')
mins=min(s)
if mins>0; mins=0; end
axis([-1 7 mins-0.2 max(s)+.2])
hold off
title('Step Response with Matlab step')
```

```

%
% Main Code
vs=zeros(1,601);
vh=zeros(1,601);
vb=zeros(1,601);
vl=zeros(1,601);
dt=0.01
for n=2:1:601
    vl(n)=vl(n-1)+vb(n-1)*dt;
    vb(n)=vb(n-1)+vh(n-1)*dt;
    vh(n)=1-D(2)*vb(n)-D(3)*vl(n);
    vs(n)=N(1)*vh(n) + N(2)*vb(n) + N(3)*vl(n);
end
vs=vs(2:601);
%
%
subplot(224)
plot([-1 7],[0 0],'k')
hold on
plot([0 0],[-10 10],'k')
plot([-1 0],[0 0],'r:')
plot([0 0],[0 1],'r:')
plot([0 7],[1 1],'r:')
plot([0:.01:6],s,'b')
plot([0:.01:5.99],vs,'r')

mins=min(s)
if mins>0; mins=0; end
axis([-1 7 mins-0.2 max(s)+.2])
hold off
title('Step Response: Matlab step (blue); Numer. Integration (dotted
red)')

figure(2)
% check Kuo for Notch - dotted green
t=0:.01:6;
VKuo= 1 - 2*t.*exp(-t);
plot(t,VKuo,'k')
hold on
VBCheck= t.*exp(-t);
plot(t,VBCheck,'m')
VLCheck= 1 - (t+1).*exp(-t);
plot(t,VLCheck,'b')
VHCheck= -(t-1).*exp(-t);
plot(t,VHCheck,'r')
hold off
figure(2)

```

```

figure(3)
% This is not supposed to work except for two real poles
scheck=N(1)*VHCheck + N(2)*VBCheck + N(3)*VLCheck;
plot([-1 7],[0 0],'k')
hold on
plot([0 0],[-10 10],'k')
plot([0:.01:6],s,'b')
plot(t,scheck,'r:')
minscheck=min(scheck);
if minscheck>0;minscheck=0;end
axis([-1 7 minscheck-0.2 max(scheck)+0.2])
hold off
title('Matlab step (blue) and Closed Form (dotted red)')
figure(3)

```

And here is some of the printout of this program:

```

step2

p1 = -0.8000+ 0.9000i
p2 = -0.8000- 0.9000i

z1 = -0.2000+ 0.9500i
z2 = -0.2000- 0.9500i

N =    1.0000    0.4000    0.9425

D =    1.0000    1.6000    1.4500

dt =    0.0100

mins =    0.3656

maxerror =    0.0032

```

APPENDIX D: Analog Notch Examples

In as much as most of what we have done above does not involve actual circuitry, we here show some of the actual circuits extracted from the App Notes and other publications from as far back as 1980 [1].

