

ELECTRONOTES 210

Newsletter of the Musical Engineering Group

1016 Hanshaw Road, Ithaca, New York 14850

Volume 22, Number 210

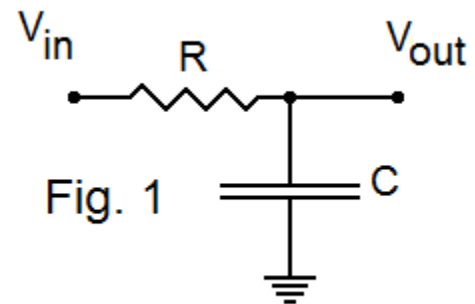
May 2012

Looking Again at the RC Low-Pass

-by Bernie Hutchins

The Basic RC Low-Pass

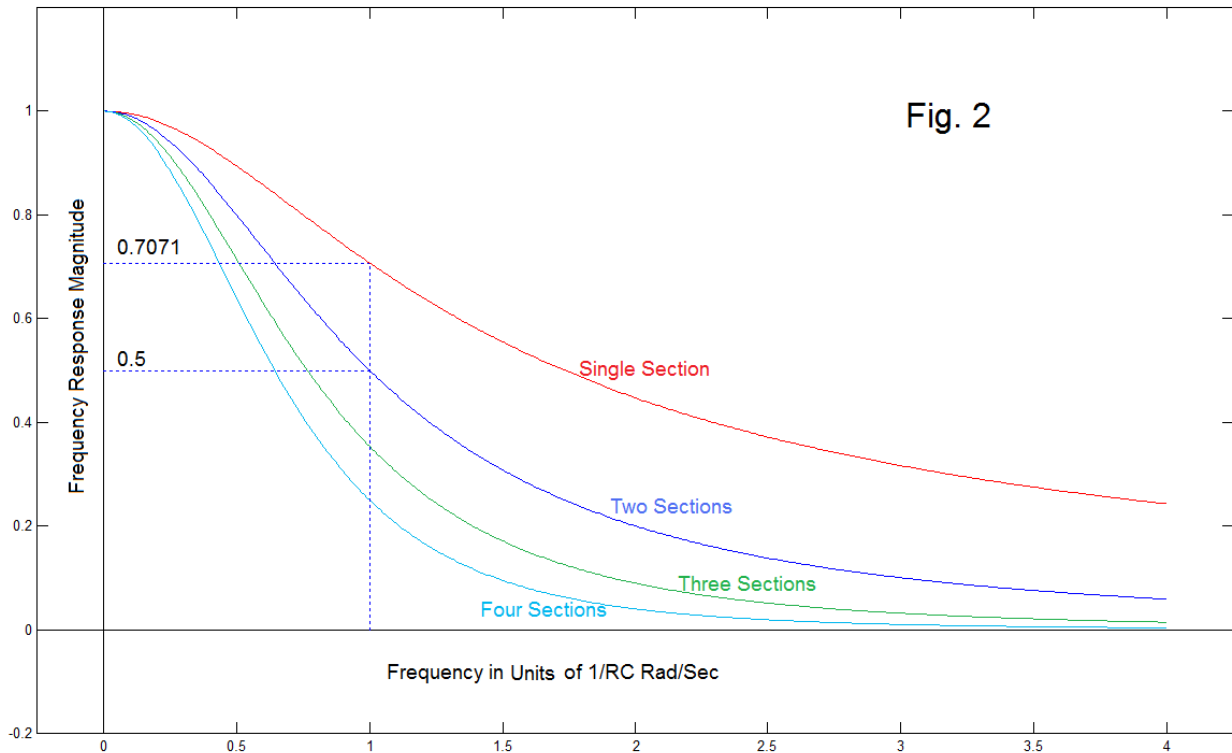
It is impossible to imagine a frequency-sensitive circuit (a filter) that is simpler than the RC low-pass (Fig. 1). Possibly most books on linear circuit theory use it as an early example. In electronic music, it is the basis of the “Moog 4-Pole Low-Pass” VCF and it is the heart of most analog envelope generators. In addition, we can describe it both in terms of its physics and its mathematics rather completely.



The filter has a transfer function: $T(s) = 1 / (1 + sRC)$ (1)

So it has a real pole at $-1/RC$, and its frequency response magnitude is just the expected first-order roll-off (red curve of Fig. 2). You can't do any better, or even anything different, with first-order low-pass.

In Fig. 2, the frequency axis is in units of $1/RC$. When this normalized frequency is 1, the magnitude of the response is down $1 / \sqrt{2} = 0.7071$. Jumping ahead a bit (ahead in our review) we also show here what we would have if we used two of these first-order sections in



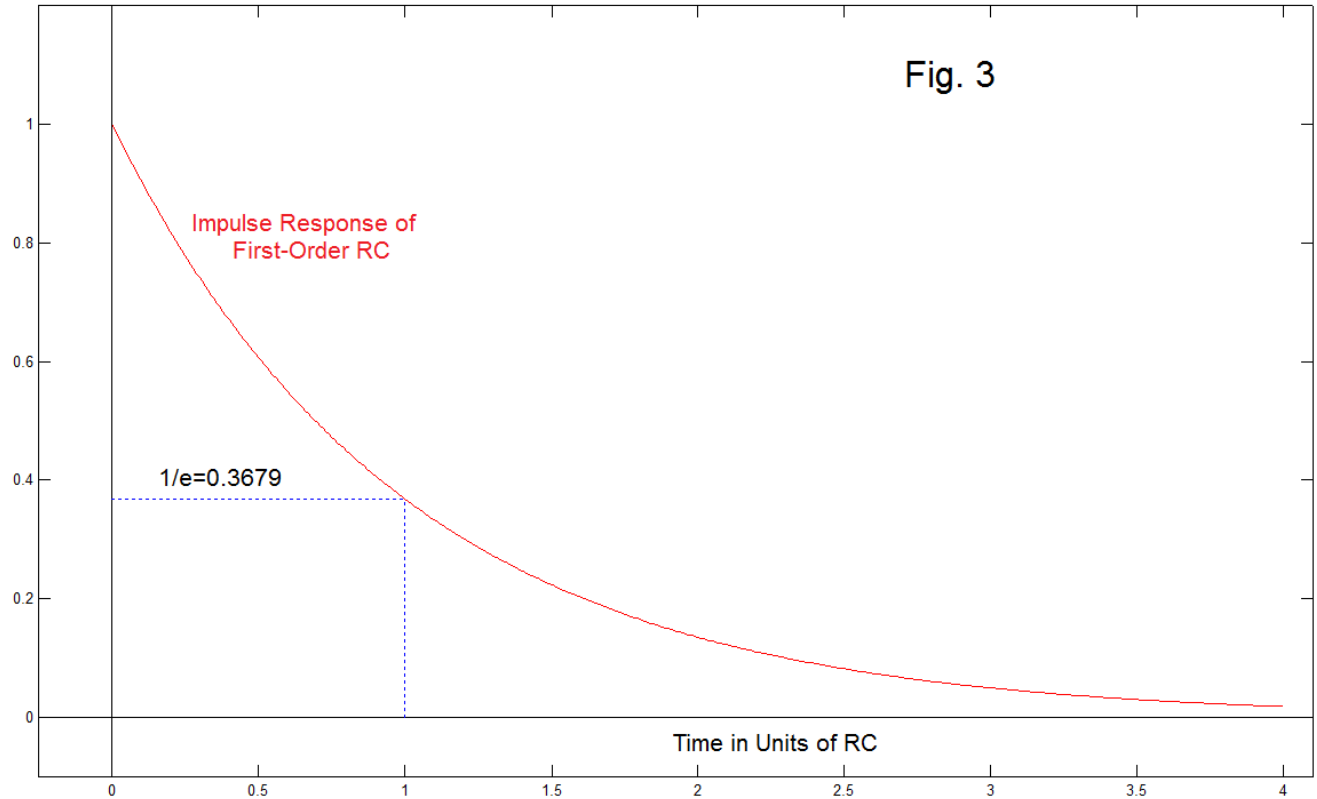
series (dark blue), three of them (green) and four of them (light blue). Note that the cutoff at frequency 1 for second order is $1/2$ (the blue curve is the square of the red one). The other curves are the third and fourth powers. We will remind the reader a bit later (in Fig. 4) what we need to consider when we actually try to “cascade” these sections (use them in series).

The impulse response of the RC is well known as well:

$$g(t) = e^{-t/RC} u(t) \quad (2)$$

– simply a decaying exponential in time (Fig. 3). This is just the decay of a charged capacitor C through a resistor R (with RC normalized to 1 second). The “impulse” deposits a charge on the capacitor at exactly time zero. But then the impulse ends, so it is the same as though the input was grounded immediately after time 0. So the capacitor is connected to ground through what we were thinking of as an input resistor. As the capacitor discharges back through what was its input, the decay reaches $1/e = 0.3679$ at time = 1.

Here we do not intend to discuss anything about phase, so this more or less summarizes what the text books say about the RC low-pass.

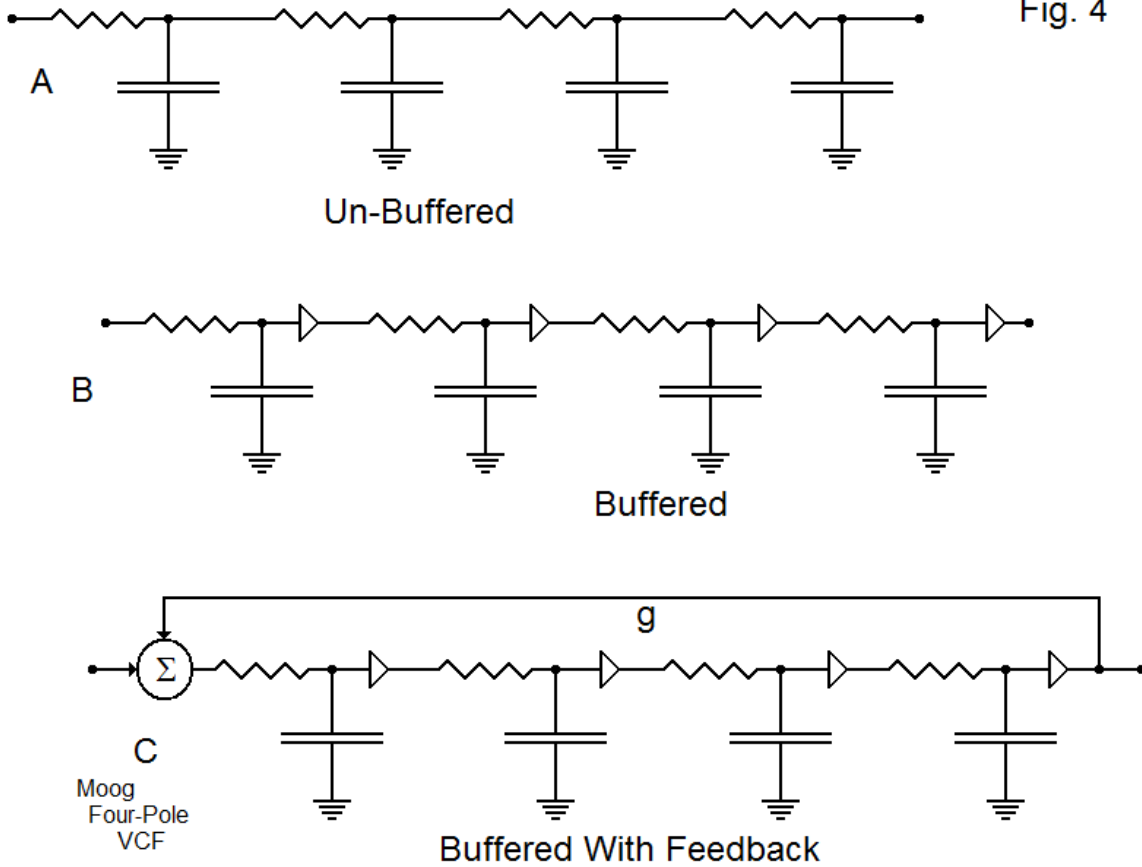


Electronic Music Extensions

We basically know how to accomplish designs with the first-order low-pass - even how to make voltage-controlled versions. Sometimes it does just what we need, is cheap, and (when necessary) does not require power. But it is just not a very good filter (Fig.1 – red curve). In an attempt to get a sharper filter, we generally think in terms of achieving a higher order. This means we need more poles (the circuit will have additional capacitors and resistors). Further, we know that real poles will not get us very far – we need complex poles. We know exactly how to do such designs, the art of “active filtering” [1] but doing them and at the same time making them voltage-controlled is more of a problem.

In Fig. 4 we will complete our review by considering the first-order in its basic cascade forms. Here we show three ways of using four (instead of just one) RC section.

Fig. 4



The following are to be understood:

- (1) We could be using any number of sections, a total different from the four shown.
- (2) We have not shown values for the resistors and capacitors, or even suggested values for them by symbols. All could have different values, although the case where all four R's and all four C's are the same is common.
- (3) We have not indicated any point as input (almost always the leftmost point) although we might insert an impulse anywhere – by depositing a charge instantaneously on any of the capacitors. The output may be the rightmost point, but other points may be outputs as well. For example, we might use an output from the first section.

Next, the following observations should be made:

(1) The difference between the first cascade, A = unbuffered, and B = buffered is that in A, the sections to the right “load” the sections to the left. The triangular symbols in B are voltage buffers. The output provides the same voltage as the one at the input, but no current is drawn by the input of the buffer. For example, they might be emitter followers or op-amp unity-gain followers. This greatly simplifies analysis.

(2) One way of achieving a measure of buffering, without using buffers, would be to maintain the RC product of all stages but to substantially increase the impedance of stages as we move to the right. The load impedance of a stage to the right is $(R + 1/sC)$, or less if there are more stages. If we increase R, while decreasing C in proportion, we can increase the impedance (reduce the loading) while maintaining the same RC product. This is of limited use.

(3) Neither A nor B is very useful. However, see more below. By adding the feedback loop g around the four stages (Bob Moog’s intuitive notion of how peaking could be achieved) a practical and popular circuit configuration was achieved (Fig 4C).

This completes the review.

WHY THE ANALYSIS IS TOUGH

Well, actually, the analysis is not that tough, as long as we work with the buffered implementations (B or C). B was never a problem, with all resistors the same, or with them different. It was just an increasingly wimpy filter. These are the dark blue, green, and light blue curves of Fig. 1. The Moog configuration was not a problem [2] particularly as Richard Bjorkman simplified the calculation [3] or as it would have eventually yielded to numerical root-finding [4].

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. 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. Further, it was only a matter of 10 minutes breadboarding it to see that there was no wonderful filter awaiting us.

Of course, we did know, exactly, the transfer function in the buffered case. Further, although we may not have attempted to invert this Laplace transform (invert the fourth-order transfer function to an impulse response), it was not difficult to understand that the result could merely be obtained with convolution (convolving Fig. 2 with itself) as necessary. Numerical convolution was easy enough.

THE UNBUFFERED CASE

The unbuffered case remains. We have every reason to understand that, as a filter, this unbuffered case would be even wimpier than the already disparaged buffered case. And it promised pages and pages of algebra, and little means of verifying a correct result except by repeating the calculation. That is, getting someone else to do it – if you try to check your own calculations, you almost always make the same mistakes.

We can however, without much effort at all, use numerical simulation to calculate such things as the impulse response, the step response, a constant-current drive response, etc. You just don't get a formula. Well – today's computers are plenty fast that you don't even get to blink. Write some code, initialize the state, and push the enter key. We can see what's out there.

In trying this, our first goal is (obviously) to get results that agree with the ones we know (or suspect) to be true. Then you can play with the numbers. One side result, which I did not see coming, was that this forces you to notice other things going on. (After all, you had to calculate all the states inside the network to do the simulation, so why would you not plot them?) In particular, in the case of Fig. 4A, there are four possible outputs – not just the one that is rightmost. We really weren't even thinking about this. Now, in the buffered case, the impulse response of the first stage is just the trivial first-order case (Fig. 2). And the impulse response out of the second stage would be this convolved with itself. And so on. But what happens when the second stage starts stealing charge from (loading down) the first stage, as happens in the unbuffered case? Now that's something to puzzle over. I don't think we ever did this.

In Fig. 5 we show two cases for our simulation study. Since we always prefer to test our procedures on simple cases, just to make sure they work, in Fig. 5A we see what can be considered a single stage, or the first of multiple stages with buffering. In other words, let's check Fig. 1. Here an impulsive charge has arrived at time zero and instantaneously

charged the capacitor. When the impulse disappears subsequent to zero, the input is effectively grounded, and current flows out of the capacitor to ground, the usual exponential decay (Fig. 2).

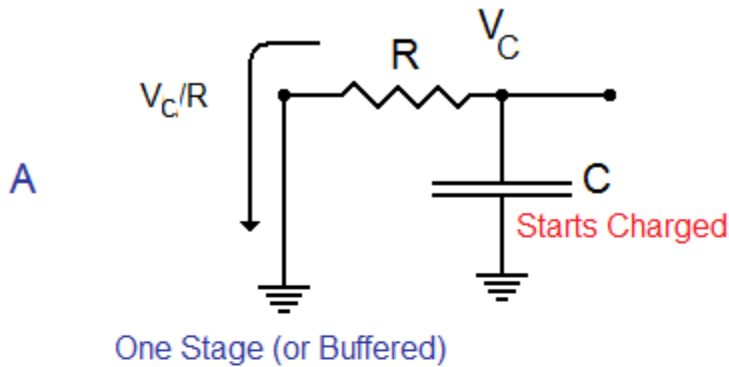
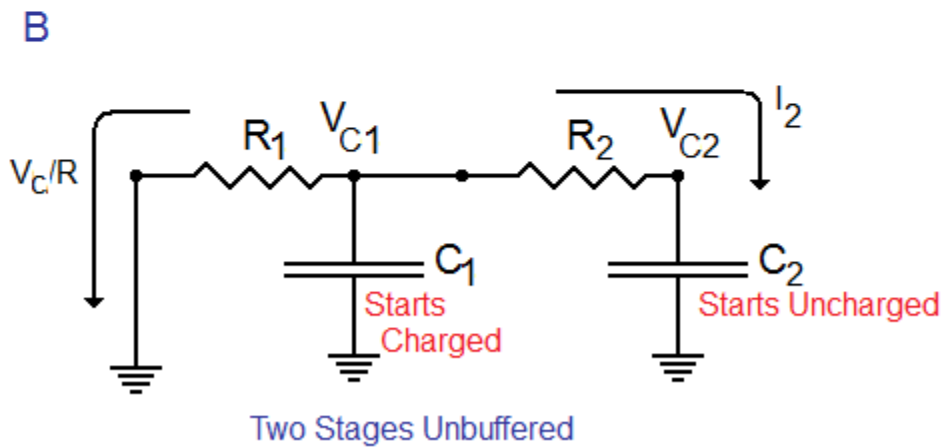


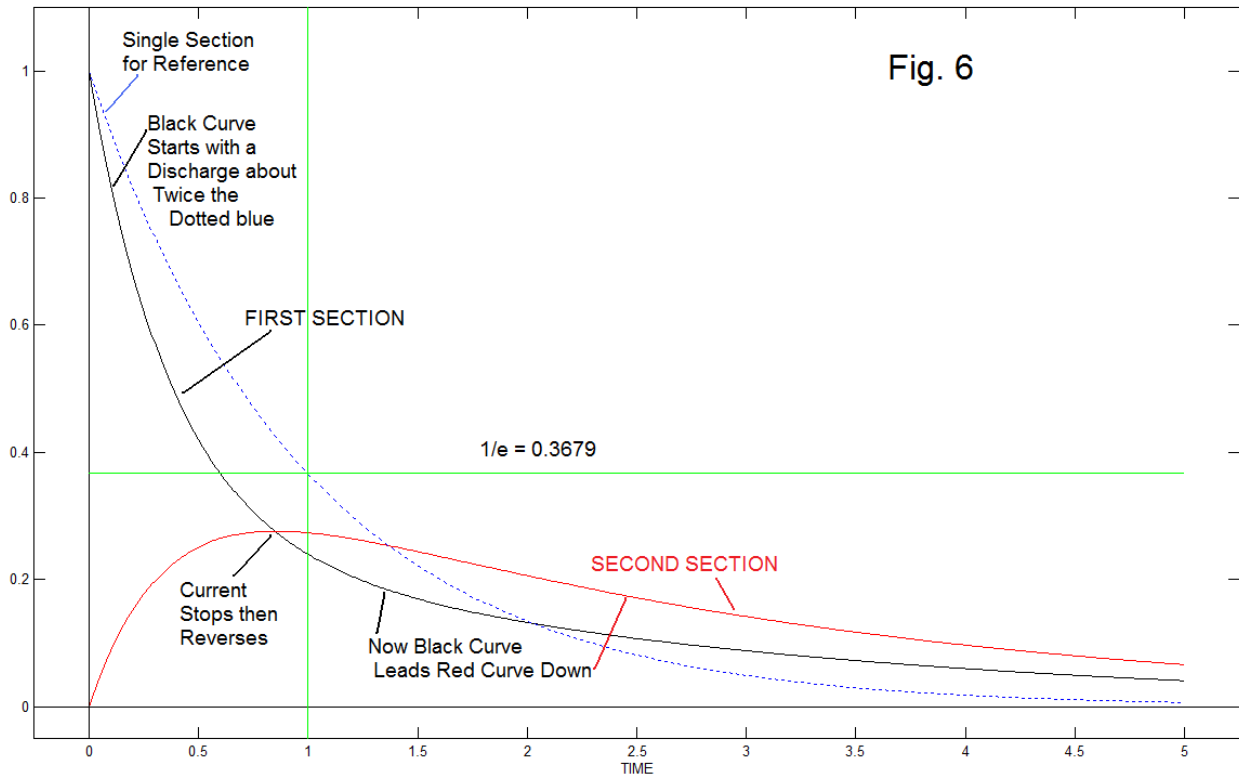
Fig. 5



In comparison, in Fig. 5B we have the case of two stages. Stage 2 on the right is loading stage 1 on the left (no buffering here). So in addition to the current flowing out the former input, a second loading current, I_2 , flows to the right. Again, the capacitor on the left receives an impulsive charge. The capacitor in the second (right) stage was uncharged. Thus, immediately following time zero, C_2 looks just like a ground, and the current I_2 that flows is initially V_C/R , the same as the left-going current. This means that the impulse response will show an initial decay that is at twice the rate of case A.

Of course, the capacitor on the right is now getting charged, and as its voltage goes up while the voltage on the left capacitor is dropping, the current I_2 decreases, becomes zero, and then reverses, both capacitors now discharging to the left, both capacitors now headed to ground. So the voltage in the second stage has a bump, up then down. Perhaps more interesting, the voltage on the first capacitor seems to be discharging at a changing rate.

We can see that this should happen. It would be interesting to see if we remember how to solve this analytically. Fig 6 below shows this happening by simulation (see below).



SOLVING THE TWO-STAGE NETWORK

TALKING IT THROUGH – AND CHEATING (AN EXPERIMENT!)

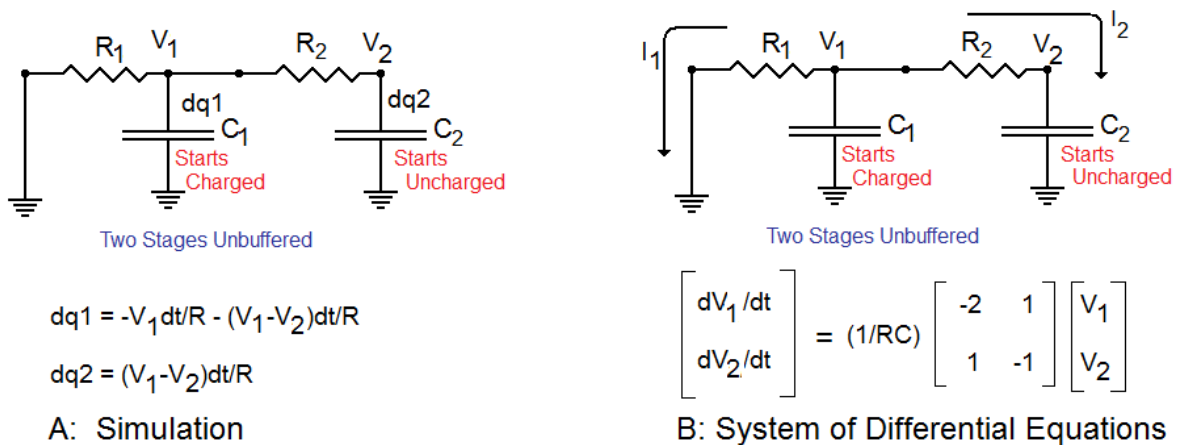
So – where did Fig. 6 come from? We will look at three methods that show how the exact curves in Fig. 6 can be calculated. However, initially we describe two ways of convincing ourselves that we really have the correct answer. First, Fig. 6 does look like the general notion we have discussed as to how the network should behave physically. Secondly – we

do an experiment - it is not at all difficult to breadboard the circuit. We used two 10 microfarad capacitors and two 1 megohm resistors. A dual channel scope was connected (using buffers) to V_{C1} and V_{C2} and very briefly V_{C1} is connected to positive supply (the charging impulse). We observe V_{C1} discharging with approximately a 5 second time constant initially, and V_{C2} starts to charge. As we approach 10 seconds, the curves do cross over, and V_{C2} follows behind V_{C1} toward ground. Of course this had to work – but I can't imagine NOT doing the experiment. Because something always seems to go strange in the analysis, we did the experiment because we want no lingering doubt as to what really happens. At this point, we have discussed what should happen physically, and we have observed the general result to have happened. Now for the math.

SIMULATION

Fig. 6 was actually produced by simulation (Fig 7A), numerical solutions of differential equations. Fig. 6 also shows the corresponding first-order case (the dotted blue curve), which offers an important comparison. Note right here that the voltage V_1 begins decaying faster than the single stage case, and then evolves to a rate slower than the single stage. So the result is different from a single stage despite the fact that we seem to be obtaining an output from the first state. The second stage, in effect, “pushes back”.

Fig. 7



In Fig. 7A, we are not “solving” any equations but rather simulating the way charge flows in the network. Program 1 shows the Matlab code we used. We have chosen time intervals equal to 0.001 (relative to an RC time constant of 1). During this time interval, denoted dt, we have currents (I) flowing, and corresponding amounts of charges (dq=I dt) transferred through resistors, ending up modifying (ever so slightly each iteration) the charge on attached capacitors, and thereby the voltages to which the capacitors are charged. This method works, and can be extended in a straightforward manner to other networks. As mentioned, this method produced Fig. 6.

SOLVING THE DIFFERENTIAL EQUATIONS

Fig. 7B is based on the same physical principles ($I = dQ/dt = CdV/dt$) and here we are writing the differential equations with the intention of solving them. This is the classic case of a system of linear constant coefficient differential equations. We write these equations in matrix form because we intend to use the usual eigenvector solution method.

We can briefly review the analysis. Summing currents at the V_1 node tells us that the current into C_1 is the negative of the sum of I_1 and I_2 . The current into C_2 is just I_2 .

$$\text{Thus } dV_1/dt = (-1/C) (I_1 + I_2) \quad (3)$$

$$\text{where } I_1 = V_1/R \quad (4)$$

$$\text{and } I_2 = (V_1 - V_2)/R \quad (5)$$

$$\text{and also } dV_2/dt = (1/C) I_2 \quad (6)$$

Which results in:

$$\begin{bmatrix} dV_1/dt \\ dV_2/dt \end{bmatrix} = (1/RC) \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (7)$$

so the system is governed by the matrix A:

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} \quad (8)$$

Using Matlab, the **eig** function, we find the eigenvectors of A as the columns of a matrix E, with corresponding eigenvalues L.

$$E = \begin{bmatrix} E(1,1) & E(1,2) \\ E(2,1) & E(2,2) \end{bmatrix} = \begin{bmatrix} 0.8507 & 0.5257 \\ -0.5257 & 0.8507 \end{bmatrix} \quad (9)$$

$$L = [-2.6180 \quad -0.3820] \quad (10)$$

The complete solution for the voltages V_1 and V_2 , involving two constants C_1 and C_2 to be determined by initial conditions is:

$$V_1 = C_1 E(1,1) e^{-2.618t} + C_2 E(1,2) e^{-0.382t} \quad (11a)$$

$$V_2 = C_1 E(2,1) e^{-2.618t} + C_2 E(2,2) e^{-0.382t} \quad (11b)$$

Plugging in the actual eigenvectors from equation (9) we have:

$$V_1 = C_1 0.8507 e^{-2.618t} + C_2 0.5257 e^{-0.382t} \quad (12a)$$

$$V_2 = -C_1 0.5257 e^{-2.618t} + C_2 0.8507 e^{-0.382t} \quad (12b)$$

Evaluating these at $t=0$ (the initial conditions) gives:

$$V_1(0) = C_1 0.8507 + C_2 0.5257 = 1 \quad (13a)$$

$$V_2(0) = -C_1 0.5257 + C_2 0.8507 = 0 \quad (13b)$$

which can be solved for:

$$C_1 = 0.8507 \tag{14a}$$

$$C_2 = 0.5257 \tag{14b}$$

so that our final result is:

$$V_1 = 0.7236 e^{-2.618t} + 0.2764 e^{-0.382t} \tag{15a}$$

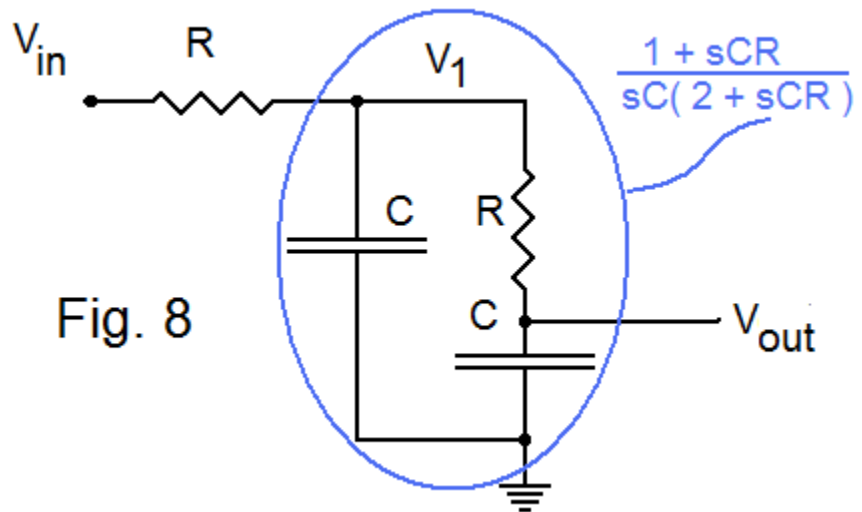
$$V_2 = -0.4472 e^{-2.618t} + 0.4472 e^{-0.382t} \tag{15b}$$

These equations when plotted are the same as in Fig. 6, V_1 being the first section (black) and V_2 being the second section (red).

POLES – LAPLACE TRANSFORM:

We have expressed a desire not to actually solve out the network for a transfer function and then try to invert it. This is not all that bad for just two stages and it can be used to illustrate that the Laplace approach is equivalent to the eigenvector approach. Fig. 8 is equivalent to Fig. 7 with the input shown as an actual input.

Here we will be concerned with the output of the second stage. It is simple to see that the three components in the oval in Fig. 8 have a combined impedance as shown there, and with the R in the input leg, V_1 is just determined by a voltage divider. V_{out} is then just a first-order low-passed version of V_1 .



Thus the transfer function $V_{out}(s)/V_{in}(s)$ is just:

$$T(s) = V_{out}(s)/V_{in}(s) = (1/R^2C^2) / (s^2 + 3s/RC + 1/R^2C^2) \quad (16)$$

We can set $RC=1$ and get

$$T(s) = V_{out}(s)/V_{in}(s) = 1 / (s^2 + 3s + 1) \quad (17)$$

so the poles are at:

$$p_1, p_2 = (-3 \pm \sqrt{5})/2 = -0.382, -2.618 \quad (18)$$

and these number are familiar to us, the eigenvalues found above (as we probably knew). Having the poles, we can expand $T(s)$ in partial fractions:

$$T(s) = a / (s-p_1) + b / (s-p_2) \quad (19)$$

Cross multiplying and equating powers of s in the numerator to equation (16), or whatever method you prefer to do partial fractions, we arrive at:

$$T(s) = (1/\sqrt{5}) / [s + (-3 + \sqrt{5})/2] - (1/\sqrt{5}) / [s + (-3 - \sqrt{5})/2] \quad (20)$$

The constants of the terms of the partial fraction expansion are 0.4472 and -0.4472, and are thus seen to be the coefficients of the eigenfunction solution. All that remains is to invert the two terms of the partial fraction expansion, noting the Laplace transform pair:

$$1 / (s+\alpha) \leftrightarrow e^{-\alpha t} \quad (21)$$

and we obtain the same answer as V_2 of equation (15b):

$$V_{out} = 0.4472 e^{-0.382t} - 0.4472 e^{-2.618t}$$

We can find the corresponding equation for V_1 . Curiously we have found first the equation for V_{out} (V_2 previously) because it was only a first-order low-pass of V_1 . Thus we just back up the process, and $T(s)$ for V_1 becomes the transfer function for V_{out} now multiplied by the reciprocal of the first-order low-pass.:

$$T(s) = V_1(s)/V_{in}(s) = (1 + s) / (s^2 + 3s + 1) \quad (22)$$

This has partial fraction coefficients [corresponding to equation (19)] as $b = 0.2764$ and $a = 1-b = 0.7236$, so

$$V_1 = 0.2764 e^{-0.382t} - 0.7236 e^{-2.618t} \quad (23)$$

which is equation (15a) back again.

A PREFERENCE FOR SIMULATION

Three things should be noted about our math: the eigenfunction analysis and the Laplace analysis. First, they were a lot of work. Second, we only did the second-order case. Thirdly, we did learn some general things, like the solution having poles shifted from the $1/RC$ positions. But the equations did not tell us much else directly – we needed the plots. Getting more plots would mean solving more equations and then plotting the results. Very tedious at best.

Accordingly, we might say that in general, we prefer to do simulation, getting the correct curves without the equations. We write some easy code, and then plug in numbers and press ENTER.

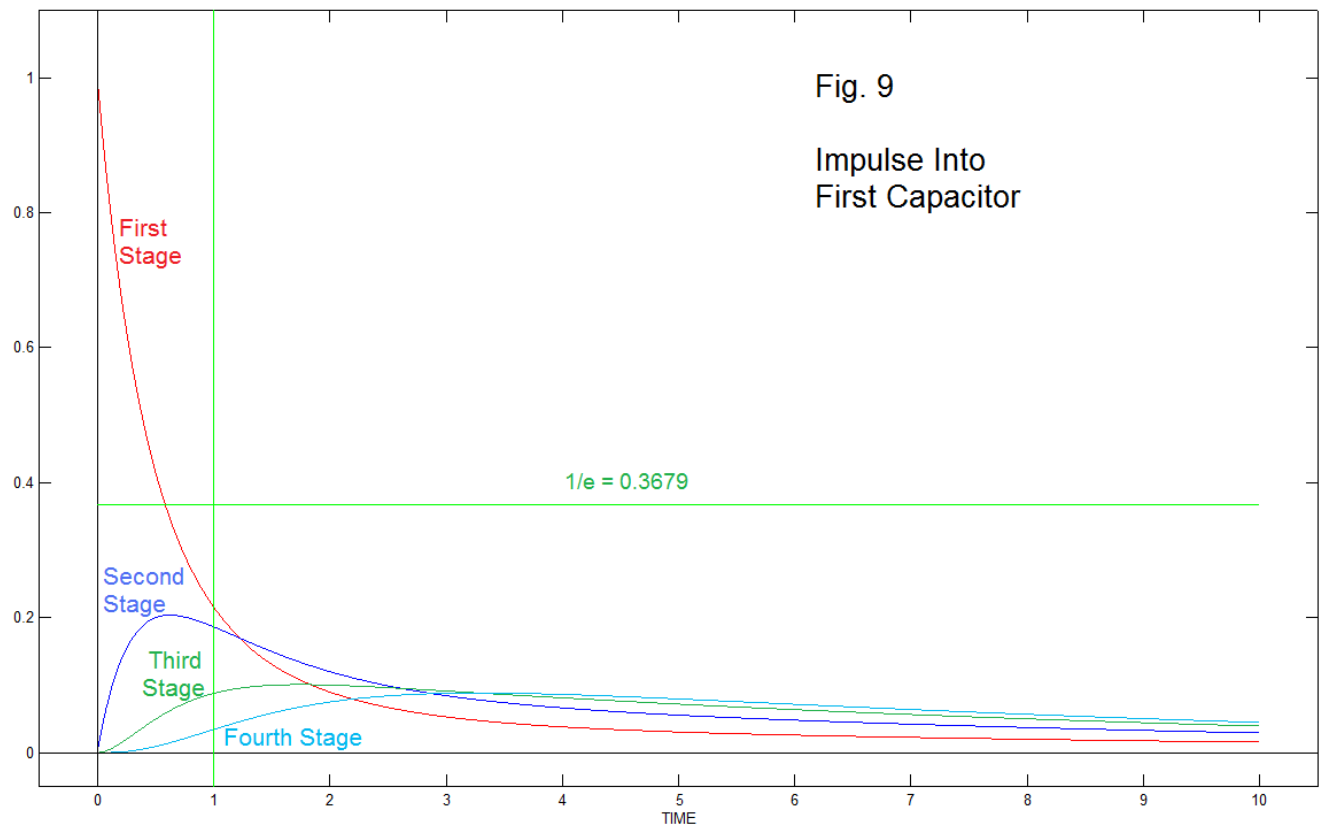
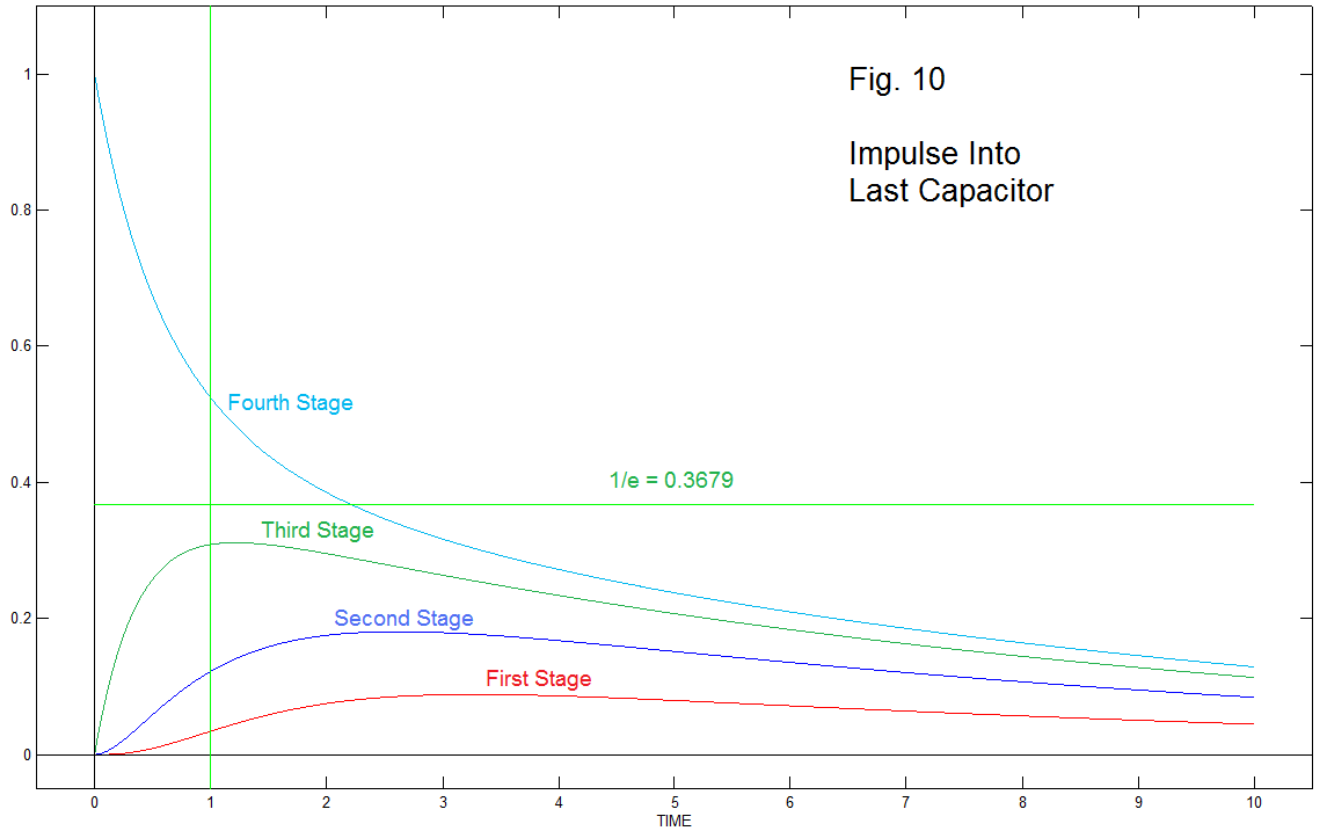


Fig. 9 shows the graph (using Program 2 to be listed later) that corresponds to four stages, as in Fig. 4A. For this program we can set any component values (the values of each R and C), the initial voltages, and so on. For Fig. 9, we have just added two additional stages to the result of Fig. 6. The additional stages change the details of the result, but things are qualitatively very much the same. We do see the later stages down the line charging and discharging much as the second stage did in Fig. 6.

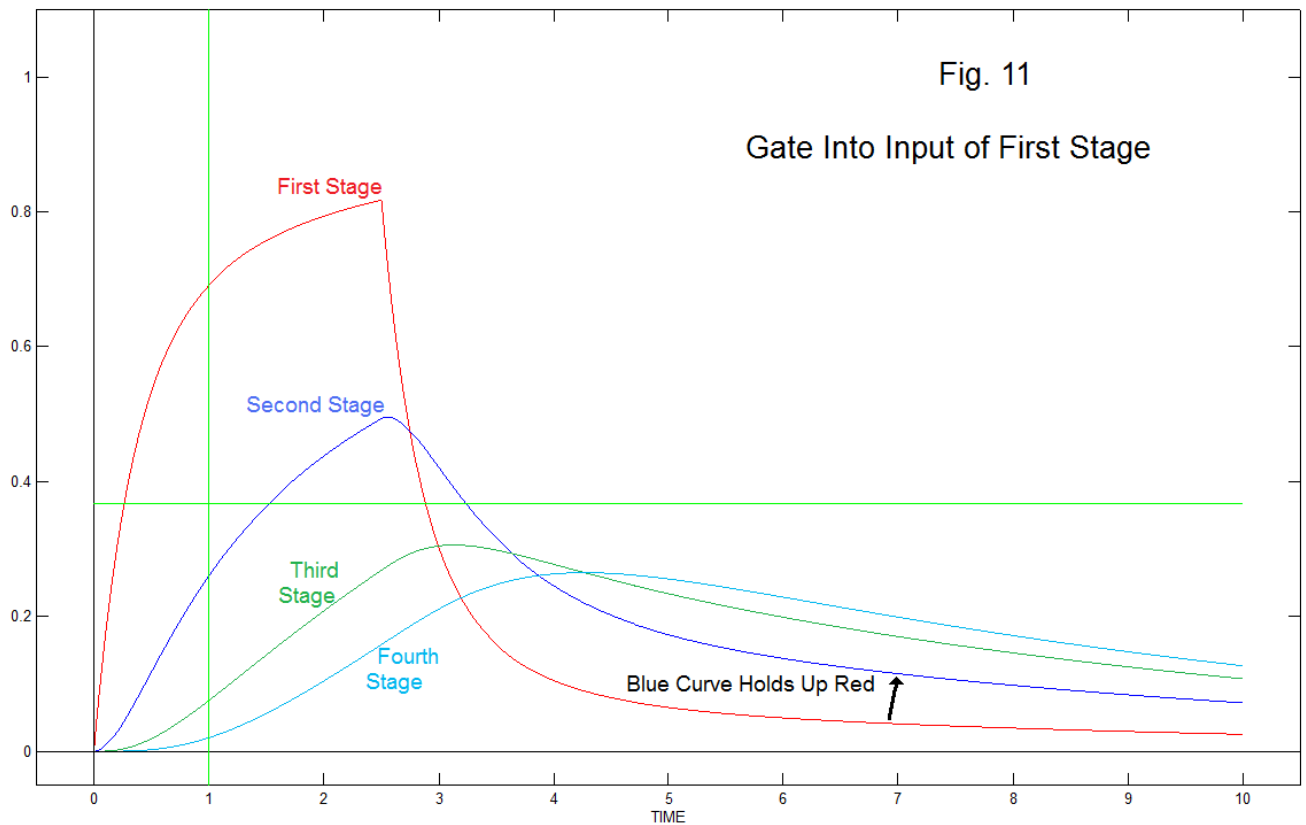


By changing two numbers, making $V1 = 0$ and $V4=1$, we change the impulse to the fourth stage. This results in Fig. 10. Our first guess might be (mine was) that this would just reorder the curves, but there otherwise seemed to be symmetry. It was different – this is why we DO experiments. The difference is because the last stage does discharge backward into the earlier stages, but there is no other path to ground, as there was with stage 1. In consequence, the discharge is much slower (compare red curve of Fig. 9 to light blue curve of Fig. 10). Note that as the program is written, the input stage is grounded.

NATURAL NETWORKS

In our work with music synthesis, we have frequently found more success by (intentionally or by chance) mimicking natural sound producers. Two examples come to mind. We found exponential decaying envelopes to be preferred over linear decays. Also, we found a preference for low-pass spectra, with lower cutoffs as time progressed. Both these seemed to be tied to the fact that natural, physical, acoustic devices worked this way. Other schemes might well be fun to try, but when it comes down to designing synthesizers that produced sounds that musicians could use to produce music to sell to the public – well practical considerations usually ruled – even with regard to what we considered successful ourselves.

With regard to how energy is input to a network and how it dissipates away, we have gotten a lot of mileage out of RC decays. Yet at times we know the exponential decay does not seem quite right. Accordingly in some cases, we have made provisions for an early truncation. After all, a true asymptotic approach to zero may be quite fanciful. This was done either by adding a small negative component to an exponentially decaying envelope, or by offsetting a VCA (usually a two-quadrant multiplier). Thus a tone can end, not too suddenly, but not seeming to hang on too long.

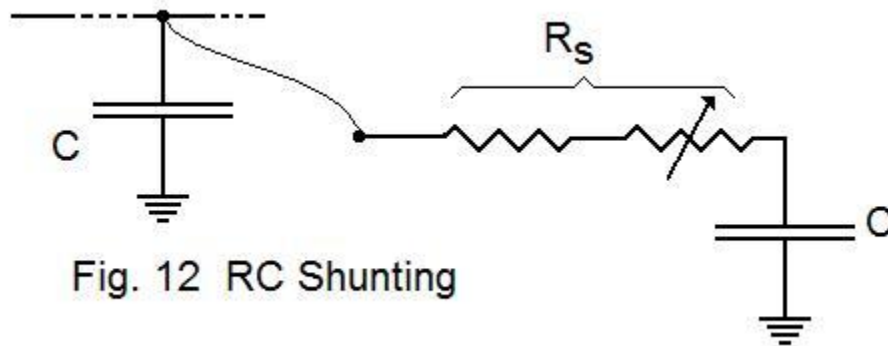


On the other hand, we may want a tone to hang on longer and “sing” such as occurs naturally in coupled piano strings [5]. Yet we really can’t do this by adding a positive offset to an exponential envelope. It would never stop, while we really would like to just go to a slower rate of decay as time progresses. Here is where we might want a dual rate of decay as we have seen above, and with a recent examination of Kautz functions [6].

Fig. 11 shows that the dual rate seen in the exponential decay from an impulse also carries over to the exponential decay on the falling edge of a gate. We have four RC stages total here – probably two is enough in practice. The first stage is the red curve. Note that when the gate disappears at time 2.5, the red curve first decays rapidly toward the blue curve, crosses it, and is then slowed up as charge comes back from the second capacitor.

The RC networks we have been discussing are natural enough, provide a variety of potential envelope waveforms (available with buffers) and can be fed impulses (short pulses from monostables) and/or gates, likely through diodes. The key point is that we don’t have to take steps for a natural response – the curves are natural already. Potentially, this fact might be of interest in generating natural musical tones.

In its simplest trial, we might just take any RC envelope generator we have, and couple a second capacitor to it through a shunting RC section, as suggested in Fig. 12. We have in mind here a capacitor equal to the one in the original envelope generator (indicated as the left capacitor), with a resistor R_s composed of a series of a pot and a limiting (minimum) resistor. The resistance R_s should probably be capable of being adjusted so as to be large enough that the shunting capacitor is essentially isolated.



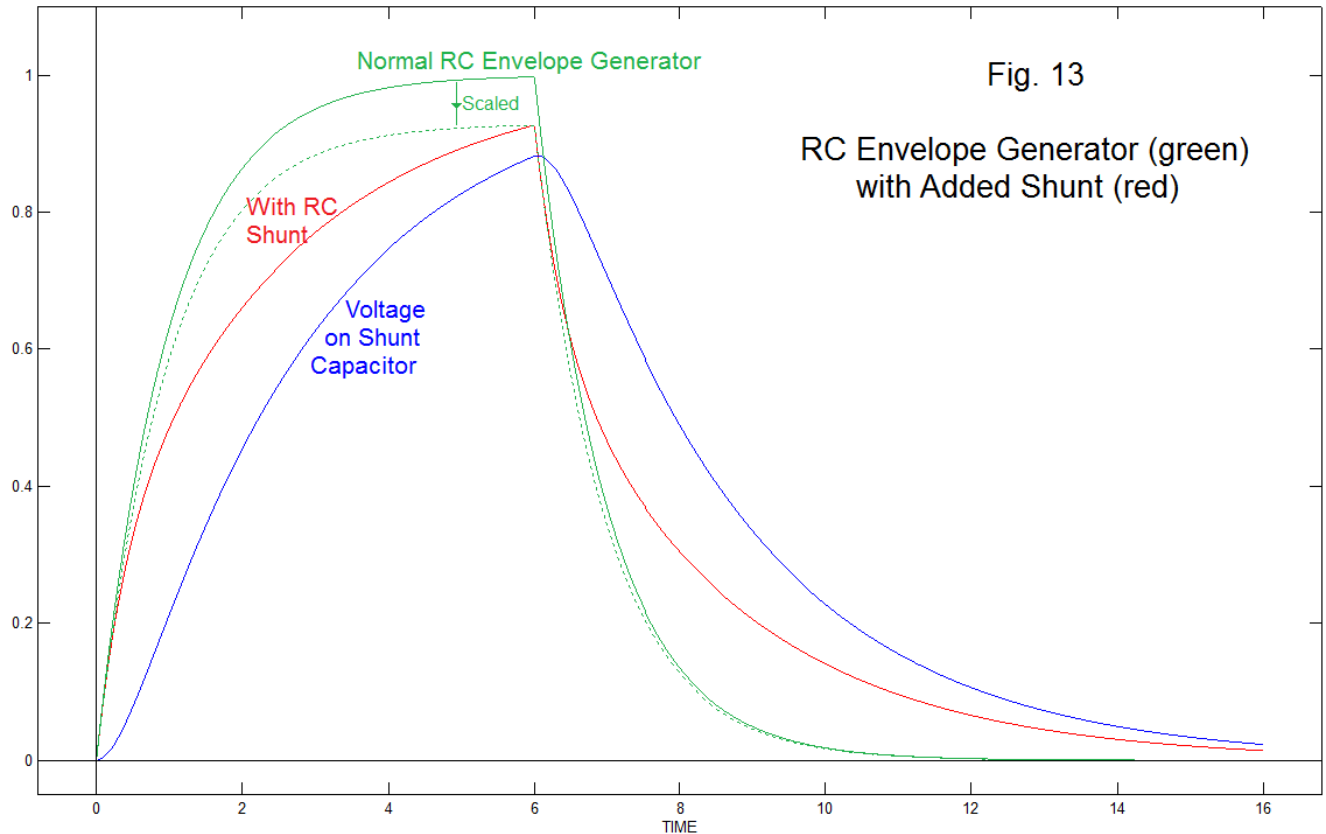


Fig. 13 shows a result of a simulation (similar to Program 2) that allows for a “gate” input. The point is to see how the shunting works on a rising edge as well as a decaying one. In this case, we have chosen the shunting R and C to be the same as the original R and C. In Fig. 13, the red curve is the output of the envelope generator with the shunt connected, as compared to the green curve without the shunt, a normal “AR envelope generator”. The dotted green curve is simply a scaled version of the green one, scaled smaller so that it exactly matches the peak of the red curve. The blue curve is the voltage on the shunting capacitor.

We can easily see what is happening. The shunt slows down both the attack and the decay, as we would expect. That is, it pulls off current during the charge, and supplies current during the decay. Yet we see that at the beginning of the decay (at time = 6) both the green and the red curves have a very similar decay rate. Very soon however the red curve slows, and we see an extended possibility of a “singing” response.

PROGRAM 1

% PROGRAM RC2.m TWO STAGES – This produced Fig. 6

C1=1

C2=1

R1=1

R2=1

V1=[];

V2=[];

V11=[]; % For single stage

V1(1)=1 % Impulse to C1

V2(1)=0 % C2 starts at 0

V11(1)=1 % Impulse for single stage comparison

dQa=0

%dQa=.0001 % non-zero = constant current

dt=0.001 %Time interval

trials=5000 T trials for five seconds

for k=1:trials

% Here currents $i = dq/dt$, so...

dQ1= -V1(k)*dt/R1 - (V1(k)-V2(k))*dt/R2; % Change of charge C1

dQ2= (V1(k)-V2(k))*dt/R2; % Change of charge C2

V1(k+1) = V1(k) + dQ1/C1; % Change of Voltage C1

V2(k+1) = V2(k) + dQ2/C2; % Change of Voltage C2

dQ11 =-V11(k)*dt/R1; % single-stage comparison

V11(k+1)=V11(k)+ dQ11/C1;

end

figure(1)

t=0:trials;

t=t*dt;

plot(t,V1,'k')

hold on

plot(t,V2,'r')

plot(t,V11,'b:')

plot([-10 10000],[0 0],'k')

```
plot([0 0],[-1 2],'k')
plot([1 1],[-1 2],'g')
plot([0 trials*dt],[0.3679 0.3679],'g')
hold off
axis([-0.05*trials*dt 1.05*trials*dt -0.05 1.1])
xlabel('TIME')
figure(1)
```

PROGRAM 2

% RC4.m for Fig. 9 and Fig. 10 and other experiments

```
C1=1
C2=1
C3=1
C4=1
```

```
R1=1
R2=1
R3=1
R4=1
```

```
V1=[];
V2=[];
V3=[];
V4=[];
```

```
V1(1)=0
V2(1)=0
V3(1)=0
V4(1)=1
```

```
dQa=0
%dQa=.001 % non-zero = constant current
```

```
dt=0.001
trials=10000
```

```

for k=1:trials

    dQ1= -V1(k)*dt/R1 - (V1(k) - V2(k))*dt/R2;

    dQ2= (V1(k)-V2(k))*dt/R2 - (V2(k)-V3(k))*dt/R3;

    dQ3= (V2(k)-V3(k))*dt/R3 - (V3(k)-V4(k))*dt/R4;

    dQ4 =(V3(k)-V4(k))*dt/R4;

    V1(k+1) = V1(k) + dQ1/C1;
    V2(k+1) = V2(k) + dQ2/C2;
    V3(k+1) = V3(k) + dQ3/C3;
    V4(k+1) = V4(k) + dQ4/C4;

end

figure(1)
t=0:trials;
t=t*dt;
plot(t,V1,'r')
hold on
plot(t,V2,'b')
plot(t,V3,'g')
plot(t,V4,'c')
plot([-10 10000],[0 0],'k')
plot([0 0],[-1 2],'k')

plot([1 1],[-1 2],'g')
plot([0 trials*dt],[0.3679 0.3679],'g')

hold off
axis([-0.05*trials*dt 1.05*trials*dt -0.05 1.1])

xlabel('TIME')
figure(1)

```

PROGRAM 3

```
% RCEig  this is just the Eigenanalysis using Matlab
```

```
[E,Lam]=eig([-2 1;1 -1])
L1=Lam(1,1)
L2=Lam(2,2)

C2=0.5252
C1=0.8834

t=0:.001:5;
v1=C1*E(1,1)*exp(L1*t) + C2*E(1,2)*exp(L2*t);
v2=-C1*E(1,2)*exp(L1*t) + C2*E(2,2)*exp(L2*t);
figure(1)
plot(t,v1,'k')
hold on
plot(t,v2,'r')
plot([-10 10000],[0 0],'k')
plot([0 0],[-1 2],'k')

plot([1 1],[-1 2],'g')
plot([0 trials*dt],[0.3679 0.3679],'g')

hold off
axis([-0.05*trials*dt 1.05*trials*dt -.05 1.1])

hold off

figure(1)
```

Summary

The offering here is possibly first one of reminding ourselves how we can use a variety of network analysis tools for electrical circuits, and indeed, for models that involve electrical networks. We are also reminded that purely mathematical approaches are always interesting, but tedious when there are a lot of cases to look at. Hence, simulation.

For our music synthesis work, we have suggested the use of coupled unbuffered RC networks for shaping envelopes and generating new ones. The use of a simple single RC network "shunting" the main capacitor in a standard RC envelope generator is suggested. We have called this resistor R_s . In our thinking, the "s" has variously been thought of as shunt, sustain, sharpen, and sing.

References

[1] We have, in our Electronotes literature, extensive coverage of active filtering. A summary, in the form of a serialized (10 chapter) manuscript, written somewhere around 1988, was published in Electronotes issues 191-196 in 1999-2000, and is available online:

<http://electronotes.netfirms.com/free.htm>

This offering includes a good number of valuable approaches that never made it into textbooks.

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

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

see also: <http://electronotes.netfirms.com/EN97VCF.PDF>

[4] 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.

[5] G. Weinreich, "Coupled Motion of Piano Strings", *Scientific American*, January 1979; and in "Coupled Piano Strings", *J. Acoustical Soc. Amer.*, Vol. 62, No. 6, pp 1474-1484 (1977)

[6] B. Hutchins, "Tools for Investigating Kautz Functions." *Electronotes*, Vol. 22, Number 207, December 2011

Electronotes Vol. 22, No. 210, May 2012