FREQUENTLY ASKED QUESTIONS

February 14, 2017

Content Questions

Why does the high pass filter have a zero but the low pass doesn't?

The low-pass filter transfer function is $\hat{H}(\hat{s}) = \frac{1}{1+j\omega RC}$, which is never zero for any value of ω. If you write it as a quotient of factorized polynomials, the denominator is just 1, which has no roots. In contrast, the high-pass filter transfer function is $\hat{H}(\hat{s}) = \frac{j\omega RC}{1+j\omega RC}$. The numerator is zero when $\omega = 0$ (only), so it has one zero.

What do zeroes do in the distance method?

The numerator of $|\hat{H}(\hat{s})|$ is the product of distances to the zeroes; the denominator is the product of distances to the poles.

Can a pole be reached for a low or high pass filter? How can $\hat{s} = -\frac{1}{RC}$?

Do you mean, "Can the complex frequency actually take on the pole value?". In our examples today, we were dealing with only pure sinusoids, which have purely imaginary $\hat{s}$, i.e., $\hat{s}$ values only along the imaginary ω axis in the complex frequency plane. Under these assumptions, we never have a signal with a frequency *at* the pole of a low-pass or high-pass filter.

(Complex frequencies with non-zero real parts describe transients, and are useful for understanding of "impulse response" of a circuit. However we will not be covering that in this course.)

Why do we use ω/ω_c instead of ωRC?

Either is OK– it can be useful sometimes to express transfer functions in terms of the corner frequency, since it then makes it easy to see what the response is for $\omega >> \omega_c$ or $\omega << \omega_c$.

What constitutes the two filters "seeing" each other?

When we say that circuit 1 "sees" circuit 2's impedance, this means that circuit 2 behaves like a box with some equivalent impedance across its input terminals. If you were to probe it at its imput terminals with some current or voltage, it would behave as if it has input impedance $\hat{Z}_{\text{in}}$. Similarly, circuit 2 "sees" circuit 1 as a box with equivalent impedance $\hat{Z}_{\text{out}}$.

How does a buffer make the second segment of sequential filters not draw any current?

An ideal buffer is a device that has infinite input impedance and zero output impedance (if it has unity gain, it doesn't change any amplitudes). If you insert a buffer between the two sections of a sequential filter, from the point of view of section 1 of the sequential filters on the left: the buffer device looks like it has infinite Z_{in} and doesn't draw any current from section 1. From the point of view of section 2 on the right: section 2 sees zero impedance from section 1 (section 1's output impedance looks like zero) and it can draw as much current as it likes (as if it were seeing the AC voltage source directly), without affecting what goes on in section 1. So if section 1 and section 2 are separated by a buffer, their transfer functions can be applied sequentially, i.e., as a product $H_{\text{tot}} = H_1 \cdot H_2$.

Do buffers contribute any other internal impedance?

In practice, *real* buffers, unlike ideal ones, do not have infinite input impedance and zero output impedance, but rather just large and small input and output impedances, respectively. We'll see examples later in the course.

What are sequential low-pass filters used for?

You would use them to create a particular frequency response, with different slopes in different regions. In practice though, it might not be so much that you would want to *design* a set of low-pass filters for some particular use... rather, the common situation is that you have some circuit that's made up of different components, to do whatever thing you're doing, but the components act like filters and you have to figure out what the frequency response is (and whether or not it's okay for your application).

If you have multiple sequential filters, can you just keep multiplying $\hat{H}_n$ as long as you have a buffer between each one?

Basically, yes, so long as there's a buffer, or the impedance of each subsequent stage is much higher than the prior one.

How do sequential filters behave if the second section does draw non-negligible current?

You can treat such networks with the usual Kirchoff's kind of analysis, but you can't just multiply the transfer functions to get the total transfer function. You would get an extra voltage drop at the output of the first network that would need to be taken into account. (This could be fixed by placing an active buffer between the sections; this would provide extra current to the second section, so that the voltage at the output of the first is maintained, regardless of the second section's impedance.)

There seem to be a lot of approximations with filters. How sharp can the actual "passing" zones be made?

In principle, you can make the passing zones arbitrarily sharp by adding enough reactive components with appropriate impedances to make steep slopes. If you want a sharp cutoff so that zero signal gets through for a range of ω's, you need a lot of zeros (in fact, an infinite number in the limit of perfect suppression) covering a whole region along the ω axis in the $\mathbf{s}$ plane. Since you need at least one reactive element per zero, this could be difficult in practice.

Is the sole purpose of a buffer enabling us to use the multiplication rule? Will it affect the circuit in other ways?

While a buffer between sequential filters does allow us to multiply their transfer functions, in fact a buffer's use is broader— the idea is to ensure that circuit elements don't affect each other.

If you don't have a buffer, you will always get a voltage drop over a load, because the circuit acts as a voltage divider when including the source impedance. You might not want that if you want the output voltage to follow the input voltage. A unity-gain buffer "protects" the load from the source impedance. A perfect such buffer will have infinite input resistance and draw no current from the source. In general buffers “isolate” input circuits from output circuits, so they don’t affect each others’ impedances (and they are very common in circuit design... we’ll see examples).