

A Ten-Minute Introduction to Digital Filters

Don Labriola, QuickSilver Controls

Filtering is needed in various control and communication applications. These can be both continuous time – analog filters, or sampled data – called discrete or digital filters. The basics of the filters can be understood in a few minutes, a “complete” understanding could be a thesis and would not cover everything. We will dive into the basics of digital filters in this article, including how to use a spread sheet to implement a simple low pass filter.

Filters may be viewed by how they alter a input signal, both in time and when looking at a range of frequencies. For this article we will start with the basic low pass filter, meaning the filter allows low frequency signals to pass without being diminished or attenuated, while reducing or attenuating higher frequencies.

Figure 1 shows a simple RC (resistor-capacitor) low pass filter. The same transfer function is observed in mechanical systems where one element varies the flow based on differences while another element integrates the flow. For example, for hydraulics this would be a flow restriction and an accumulator or a tank.

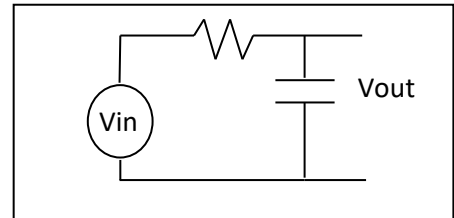


Figure 1: Low Pass Analog Filter

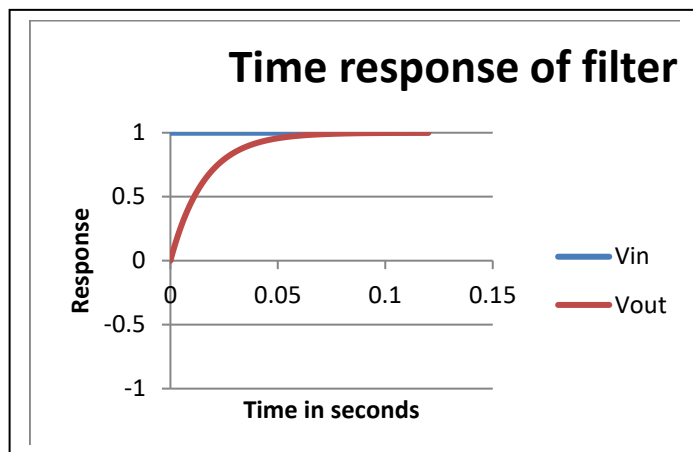


Figure 2: Time response of a low pass filter

Output vs. Time - ANALOG:

The low-pass RC filter in Figure 1 can be studied by looking at its time response. To simplify, we will assume the capacitor starts out discharged, and then Vin steps to 1v. This filter is designed for 10Hz low pass cutoff – that is at 10 HZ half of the power (70.71% of the voltage) will pass through the filter. As you can see, initially the capacitor will charge at a rate, as the voltage across the resistor is the full 1 volt input (when the capacitor voltage is zero). As the capacitor charges, less voltage will be across the resistor, so the current will drop. As the voltage across the capacitor reaches close to the full 1 v

input signal, the voltage across the resistor drops to almost zero, so the current will drop to almost zero, and the capacitor stops charging.

Now consider if the input reverses before the capacitor has fully charged; the output voltage will only change a fraction of the input voltage. The faster the signal reversals, the less the output will change. Thus, sufficiently low frequencies will “pass” through this filter, while higher frequencies are attenuated or “stopped”.

Digital Filter: The digital filter approximates this response by simulating the interaction of the resistor and capacitor with the input voltage. We look (sample) only at certain times, but if we sample at least 2x faster than the highest input frequencies present, we get a fairly good approximation.

Capacitor: The voltage across the capacitor proportional to its charge, which is the time summation (Integral) of the current into the capacitor. The Digital simulation of Figure 1 is shown in Figure 3. The box with the 1/Z is just a delay box: it holds the previous sampled output estimate until we update the simulated estimate, implemented as a memory location when implemented on a computer. The circles represent addition / subtraction functions, and the tapered box is a multiplication operation.

The simulated capacitor voltage is the sum of the simulated current, taken at discrete time steps. As long as the sampling rate is fast compared to the input variations, this is a good approximation. In the computer, this is done by saving the previous "charge" as a value in memory, and adding the new charge (current * time step) to the existing charge each update cycle. The "current" is calculated from the difference in the input voltage and the capacitor voltage. (There is also a scale factor of 1/C built into the calculation to get voltage from charge in amp-seconds.) The value stored is updated each sample period. For the QuickSilver Controllers, this is typically every 120 microseconds – about 8333 times per second.

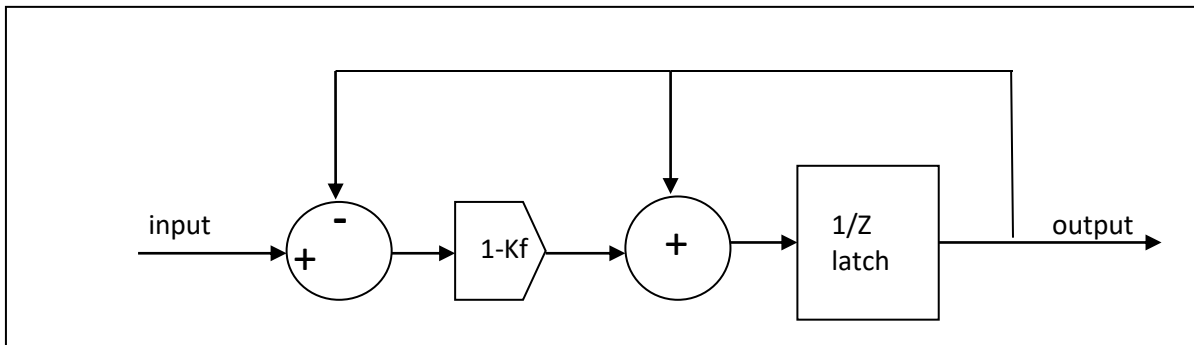


Figure 3: Digital filter Implementation

In the digital implementation, the $T/(R \cdot C)$ becomes $1-K_f$, where K_f is the feedback term, always less than 1; K_f represents the fraction of the previous charge remaining on the capacitor after one sample period if the input were zero. It may also be easily observed the new output value will equal the previous output value if the input is equal to the old output value. Said another way, the filter has a gain of one for DC inputs.

Many modern processors can multiply as fast as they can add. To speed up the calculations, the filter can be re-arranged as shown in Figure 4:

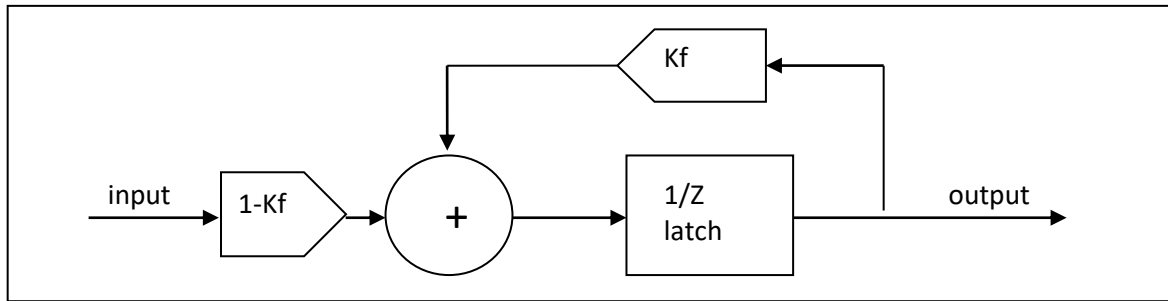


Figure 4: Simplified Low-Pass Digital Filter

Although the simulation is simplified, it actually produces good results as long as the sampling rate is sufficiently fast compared to the time response of the filter and at least twice as high as the highest frequency of the input waveform.

The spread sheet related to this article <https://qci.ms/10min.xlsx> shows the step response as the first sheet, and the frequency responses in the next several Figures 5, 6, and 7.

Figure 5 shows a 50 Hz low-pass filter with a 100 Hz input sine wave. There is a slight start up transient when the input is switched on. In the example, the first cycle of input started at peak value but was only on for half of a sine peak before reversing. The filter did not fully charge on the first cycle, so it takes several time constants for the resulting DC offset to settle). The steady state amplitude is $1/((F_{in}/F_c)^2 + 1)^{.5} = 1/(2^2 + 1)^{.5} = .447$

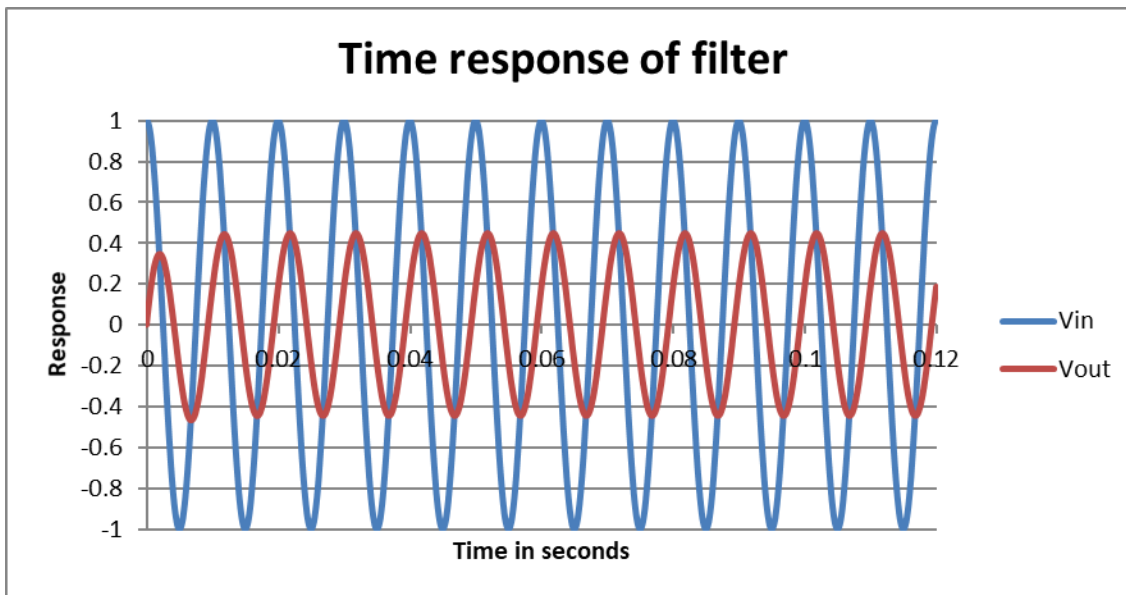


Figure 5: 100 Hz input, 50 Hz cutoff selected

When we repeat this with a 50 Hz input to a 50Hz filter, the steady state amplitude is

$$1/((50/50)^2+1)^{.5} = 1/\text{Sqrt}(2) = .707$$

This is the half power point called the cutoff frequency.

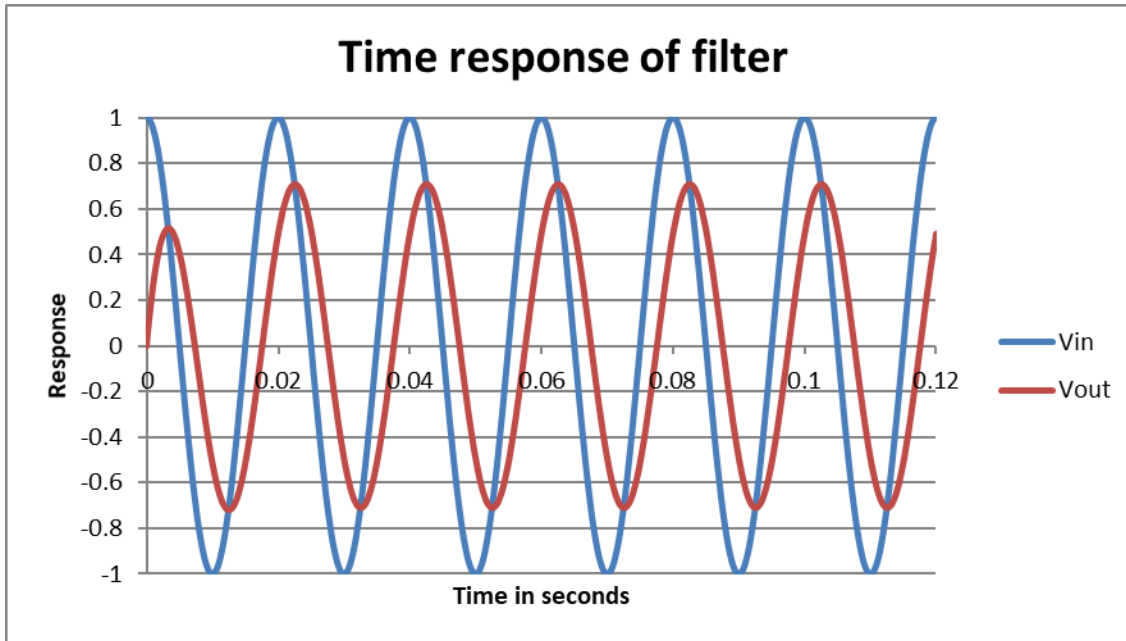


Figure 6: 50 Hz input, 50 Hz cutoff

We repeat with a 20 Hz input into a 50Hz filter and we get a steady state response amplitude of: $1/((20/50)^2+1)^{.5} = 0.928$, as see in figure 7.

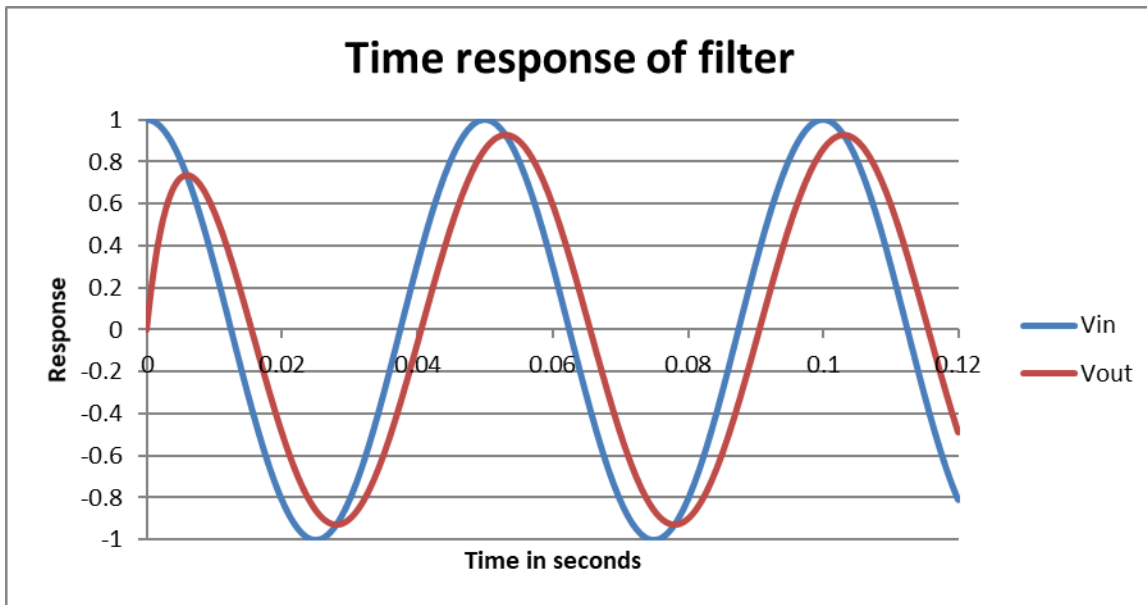


Figure 7: 20 Hz input, 50 Hz cutoff

Each of these filters is included as a sheet in the spread sheet referenced above. You can change the selected cutoff frequency and input frequency. The sampling time in the spread sheet is 120uS, which corresponds to the filters used in our QuickSilver products. This corresponds to 8333 Hz. The input frequency for this implementation should remain below 4.166 KHz for proper operation.

Calculating Kf for the desired cutoff frequency

For the analog version, consider the capacitor is charged to 1v, and at time=0 we attach the resistor.

The natural (decay) response for the RC circuit will be $V(t)=V_o*e^{-(t/RC)}$, a simple exponential decay.

We can do the same for the digital circuit by loading the register holding the estimate to 1 while keeping the simulated input voltage at zero. The output would start at 1. At the next sample period the output would be $1 * Kf = Kf$. The 2nd sample period it would become $Kf*Kf = Kf^2$ at the second sample period. In general, it would drop to Kf^n at sample interval n. Time(n) = sample period * n = $120e-6 * n$ for this particular filter.

Going back to the analog filter we are simulating: this filter has a natural response (for an initial charge voltage =1) of

$$v(t)=e^{-(t/RC)} = e^{-(2*\pi*F_c*t)} \quad \text{where the cutoff frequency } F_c = 1/(2*\pi*RC) \Rightarrow RC = 1/(2*\pi*F_c)$$

If we set the transient output for the time series equal to the transient output for the analog filter (looking only at the sampled times),

$$e^{-(2*\pi*F_c*t)} = Kf^{(t/T_s)}$$

Where T_s is the sample time and F_c is as defined above, we can take the natural log of both sides, isolate Kf, and then take the natural antilog of both sides and we have:

$$Kf = e^{(-2*\pi*F_c*T_s)}$$

This calculation is shown in cell F4 of the respective worksheets.

Limits of Simulation: This simple model of the low pass filter works well as long as there are enough samples compared to the frequency components in the input. For inputs containing frequencies of more than half the sample frequency, an effect known as "aliasing" will occur, which will cause frequencies in the output that are different from those that were applied to the input (down mixing for those familiar). These are normally not desired and should normally be avoided. (Aliasing= one frequency taking on an alias as a different frequency due to the sampling process.) Carefully controlled aliasing can be used to mix RF frequencies to an IF by proper sampling, so this effect can be used effectively, but the total bandwidth of the input signal still needs to be limited to prevent unwanted mixing!

This aliasing effect may be seen in movies showing the propeller of an airplane or the blades of a helicopter that appear to stop spinning or to reverse direction. In the case of the movie, the camera is commonly at 24 frames per second for film (often other speeds for electronically captured videos). If a propeller is spinning right at 24 revolutions per second (1440 RPM) the camera catches successive rotations of the propeller on each subsequent

frame, making the propeller appear to be stationary. A little faster and the propeller will appear to be slowly turning forward, a little slower and the blade will appear to reverse direction. A serious issue may occur if older fluorescent lighting (non-electronic ballast) is powered from a single phase: and tool fed from a synchronous motor may appear stationary when it is actually spinning; an induction motor with a small slip will appear to be moving slowly, which may appear to make the rotating device safe to touch! This is one of the reasons that the lighting was wired 3 phase in machine shops so that some of the lights we always lit throughout the waveform to prevent acting like a strobe light stopping motion.

Sampling a fast moving waveform can cause the same effects for the sampled data. In Figure 8 the input waveform is 8.033kHz and is sampled at 8.333 kHz. The sampling is less than twice the frequency component of the waveform, so the sampling causes mixing between the actual waveform and various harmonics of the sampling rate. In this case, the first harmonic of 8333Hz sampling rate is mixed with the 8033 Hz signal, resulting in a 300 Hz mixing, which then is filtered and shows up in the data.

The green “true input” shows this data if it was sampled 20x faster. The Blue “Vin” diamond that overlaps the green arrowhead shows those points of the actual data that gets collected by the sampling at a rate that is less than twice the input data frequency.

Although no 300 Hz signal was present in the input waveform, this 300Hz aliasing artifact shows up. Although the 50Hz filter should have almost completely filtered out an 8kHz signal, a significant 300 Hz signal appears at the output of the filter. If this is a control system, significant mechanical problems can result! If the input noise is near the sampling frequency, the alias can show up as a DC baseline that is varying around, which is essentially impossible to filter out.

This issue can be addressed by filtering the analog input with a continuous filter or by sampling at a much higher frequency, filtering sufficiently to remove the high frequency components, and then down-sampling (decimating) the data prior to subsequent processing at a lower processing rate.

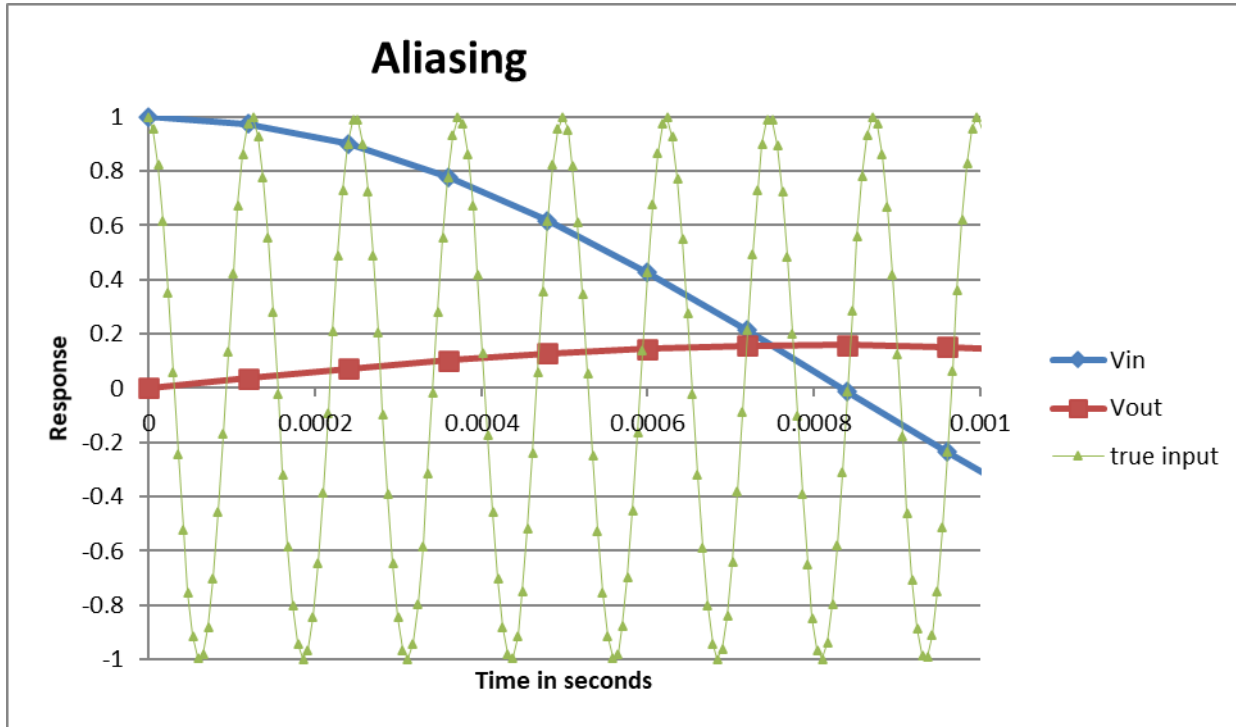


Figure 8: Aliasing: sampling a 8.033 kHz waveform at 8333 Hz

This aliasing can appear in surprising places in sampled data. On one optical sampling system in which I was involved, a flash lamp was used to take data from multiple sample containers as they spun between the flash lamp and a polychromatic (multi color) detector. The data was taken over periods as long as several minutes to watch chemistry reactions occurring. The flash lamp had a certain amount of flash to flash variation, which was expected, and which the math tried to handle as random. In looking at the data we would occasionally see a large saw-tooth shaped artifact that was quite significant in amplitude and sometimes has positive ramp, sometimes negative ramp and had a wide variety of frequencies. A little more analysis showed that the high voltage power supply powering the flash lamp had a simple hysteretic voltage regulator: when the voltage got below a certain threshold, the switch mode power supply would add one pulse of energy to the storage capacitor. The voltage waveform was thus a slow decay with an occasional step up. The actual pulsing to keep the voltage stable was on the order of a few kHz once the voltage had been reached. This variation happening at a couple thousand times per second is what got played out over minutes when the temperature of the system caused the frequency of the oscillator in the power supply to slowly drift to exactly the right (wrong!) frequency compared to the approximately 1 sample point per second per sample when we were taking data. We changed the power supply design to be proportional rather than hysteretic – this greatly reduced the amplitude of the ramp in power supply voltage (and the resulting flash intensity), and the resulting ramp in the measured data disappeared!

I encourage you to download the spread sheet and experiment with the results!

Donald Labriola P.E., President, QuickSilver Controls, Inc.