Technical Article

Low-Pass Filter a PWM Signal into an Analog Voltage

April 11, 2016 by Robert Keim

In this article we will take a closer look at how to effectively low-pass filter a PWM signal into an analog voltage.

In this article we will take a closer look at how to effectively low-pass filter a PWM signal into an analog voltage.

Supporting Information

- Introduction to Digital–Analog Conversion
- Pulse Width Modulation
- What Is a Low Pass Filter? A Tutorial on the Basics of Passive RC Filters

Previous Article

- Turn Your PWM into a DAC

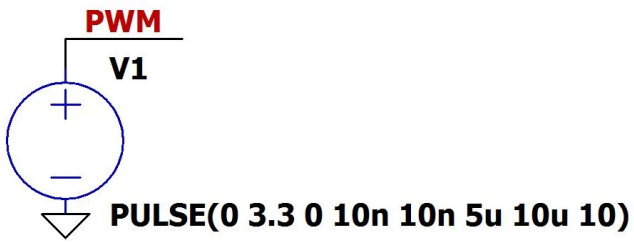
PWM in the Frequency Domain

In the previous article we saw that a pulse-width-modulated signal can be "smoothed" into a fairly stable voltage ranging from ground to logic high (e.g., 3.3 V); the smoothing is accomplished by a simple low-pass filter. Thus, we can achieve digital-to-analog conversion by using firmware or hardware to vary the PWM duty cycle according to the following relationship:

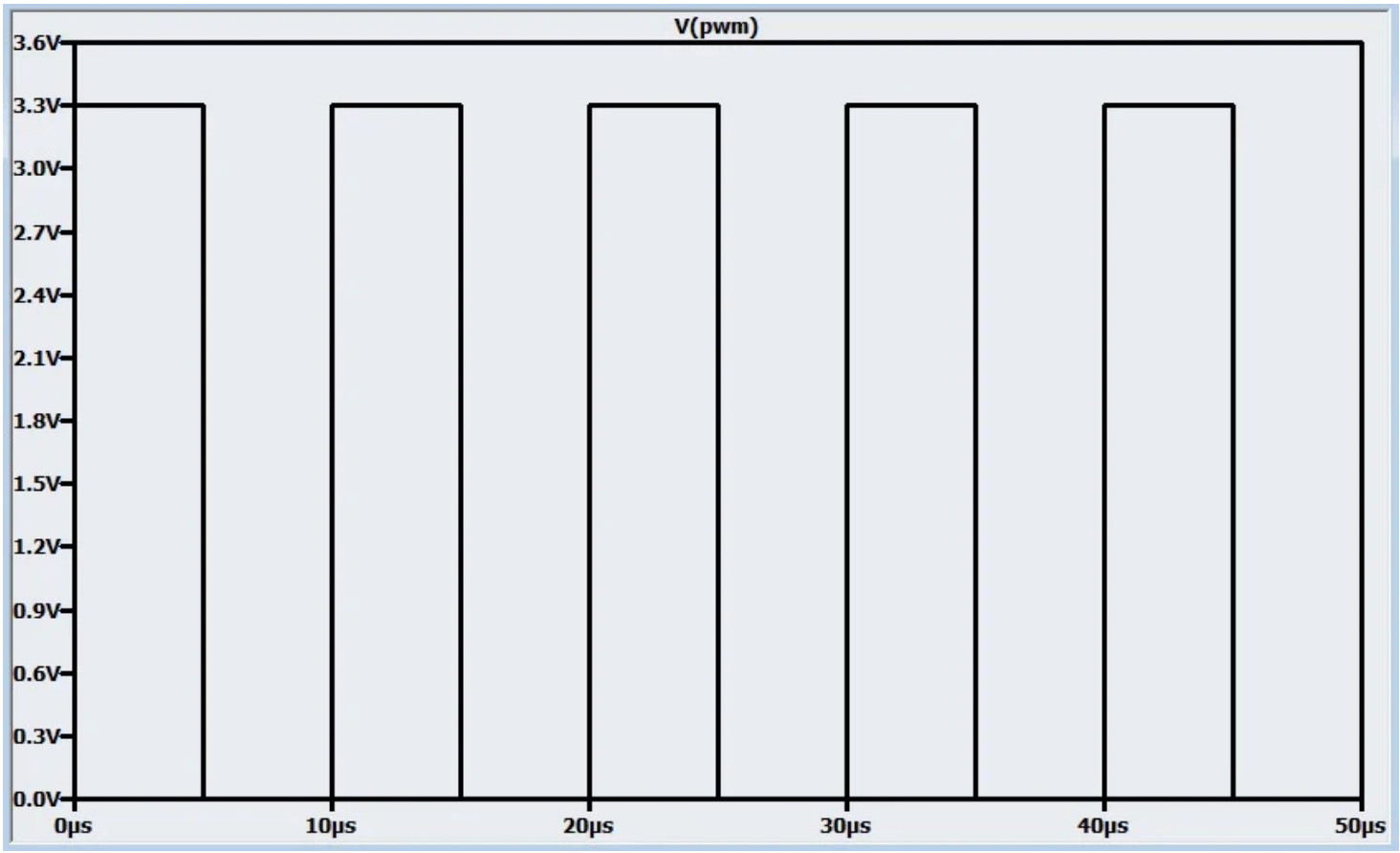
$$desired\ DAC\ voltage = A \times duty\ cycle$$

where A (for "amplitude") is the logic-high voltage.

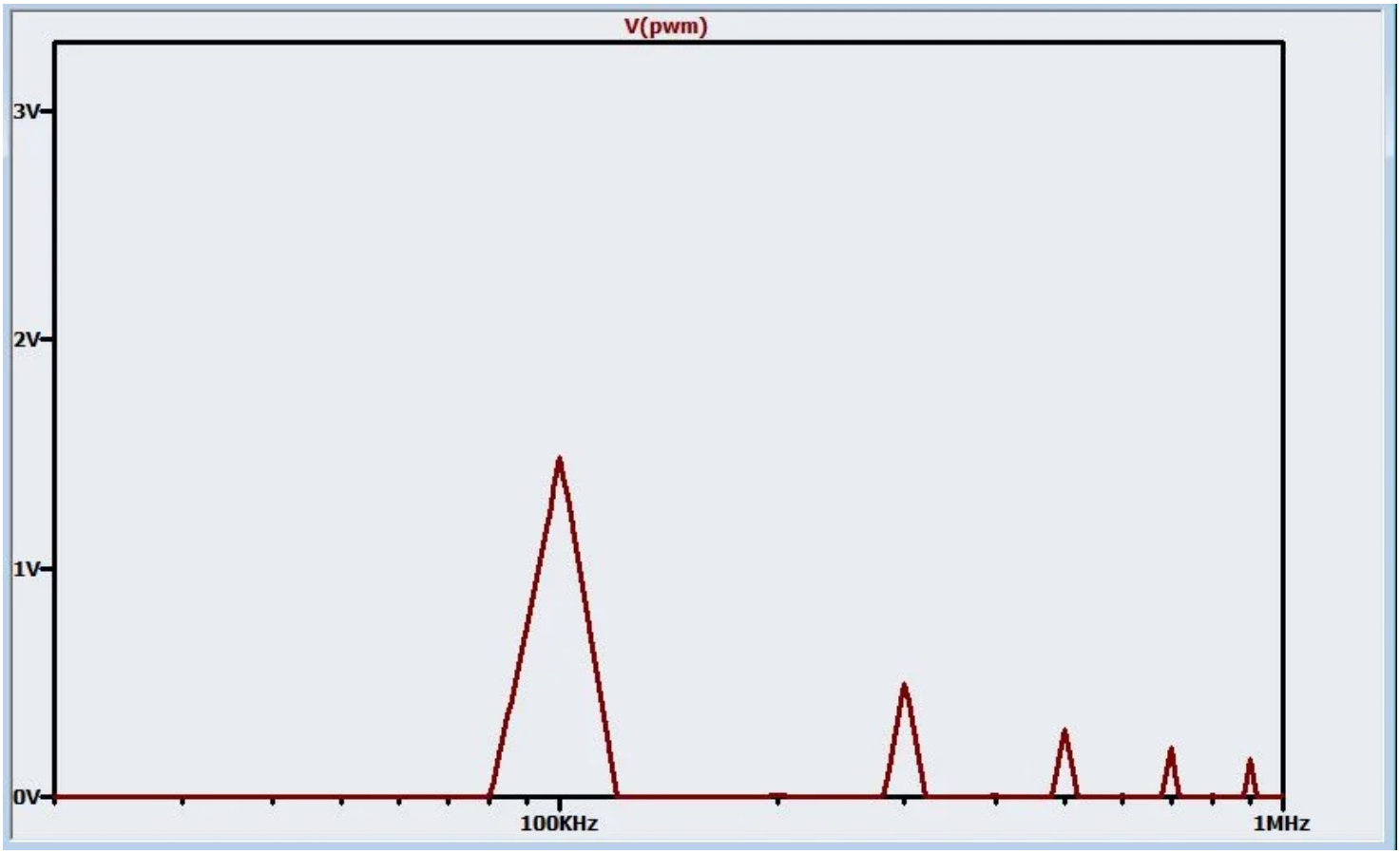
Let's begin our more-thorough exploration of the PWM DAC by looking at the frequency-domain representation of a PWM signal. Here is the LTspice schematic:



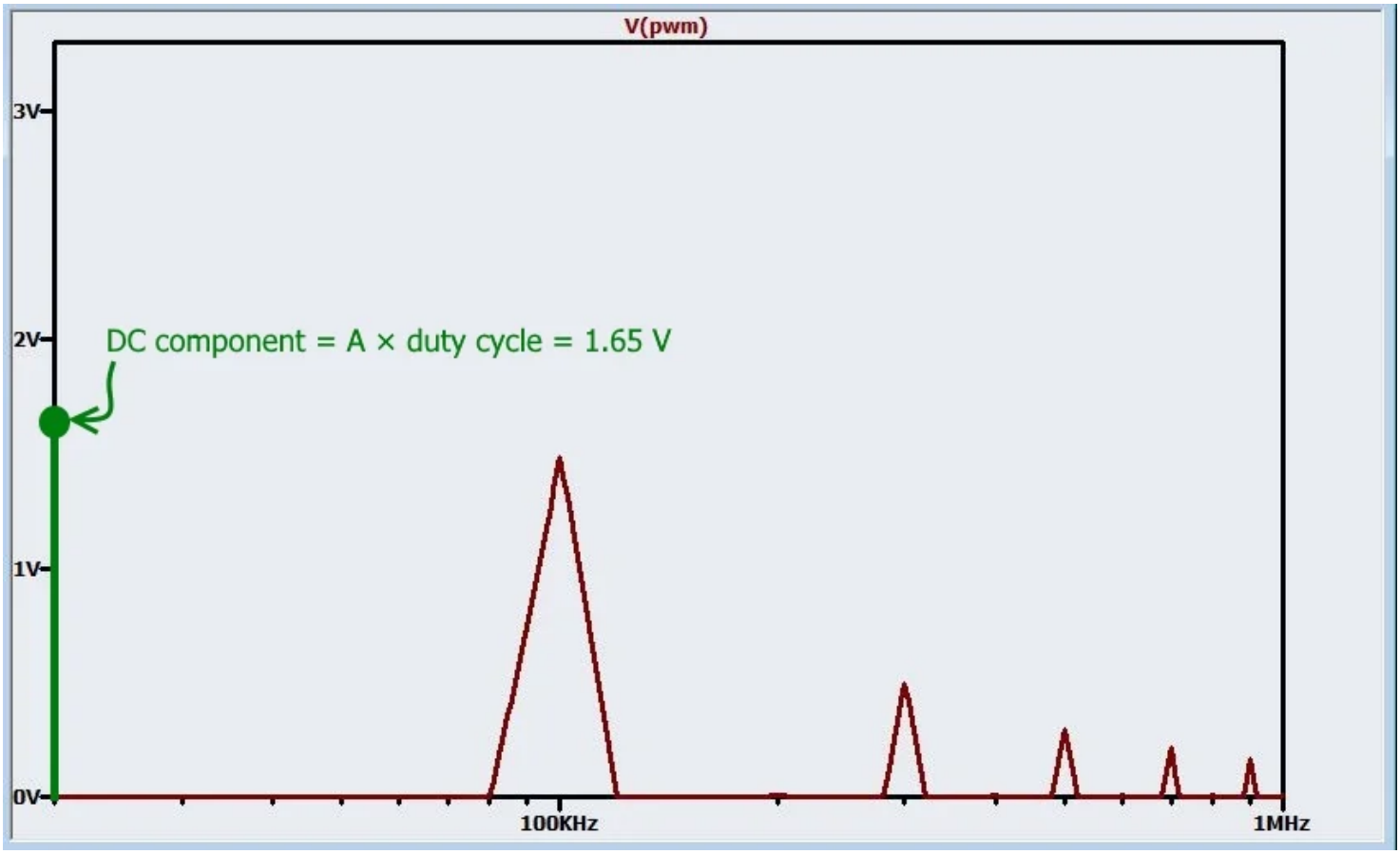
As you can see in the PULSE characteristics, the pulse width is 5 µs and the period is 10 µs. Thus, the duty cycle is 50% and the PWM carrier frequency is 100 kHz. Note also that A = 3.3 V and the rise and fall times are both 10 ns. Here is the time-domain signal:



And here is the FFT:

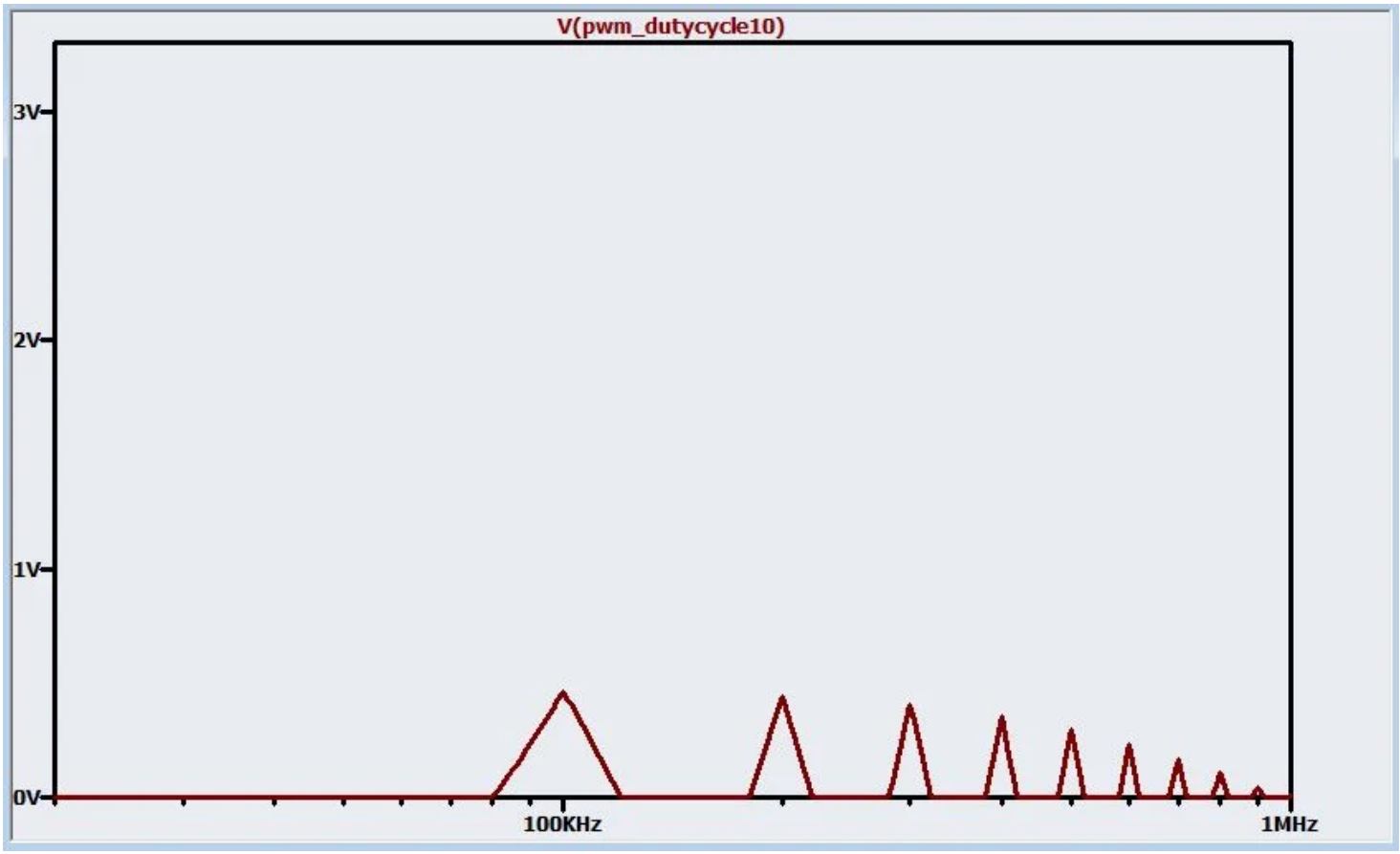


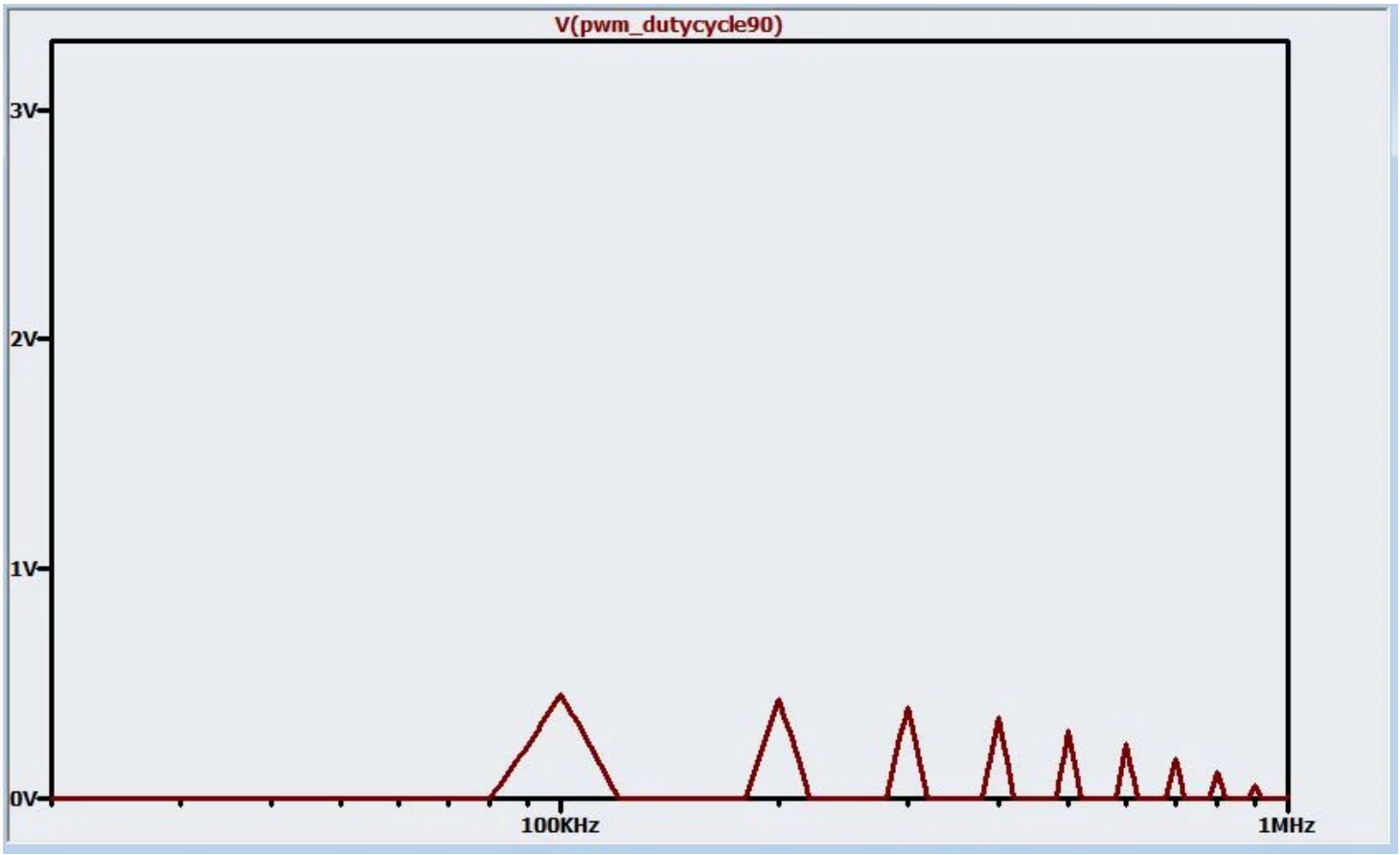
You might recognize this spectrum as the general pattern we expect from a square wave, i.e., a spike at the carrier frequency and then harmonics of decreasing amplitude at the carrier frequency times 3, the carrier frequency times 5, and so forth. However, the LTspice FFT doesn't show us the DC component, which is nonzero because this square wave is not symmetrical about the x-axis. I modified the next plot to include the DC component:



So what we want is that stable 1.65 V over on the far left, and what we don't want is that troublesome spike at 100 kHz (as well as all the higher-frequency spikes). At this point you can probably see why we use a low-pass filter in a PWM DAC: the filter retains the DC component while suppressing everything else. If we had a perfect filter, we would have a perfectly stable DAC voltage—just look back at the previous plot and imagine a "brick-wall" filter that transitions from no attenuation to complete attenuation at 50 kHz. All the non-DC components of the signal would be eliminated, and we would be left with a DC voltage at 1.65 V.

At this point, you might be wondering about how the spectrum changes as the pulse width is varied. What if the frequency components move around such that the low-pass filter becomes less effective? Consider the following two FFTs for 10% and 90% duty cycles:


V(pwm_dutycycle10)
3V
2V
1V
0V
100KHz
1MHz

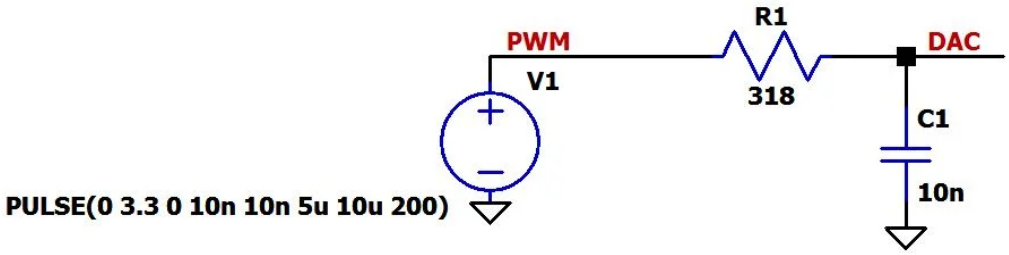

V(pwm_dutycycle90)
3V
2V
1V
0V
100KHz
1MHz

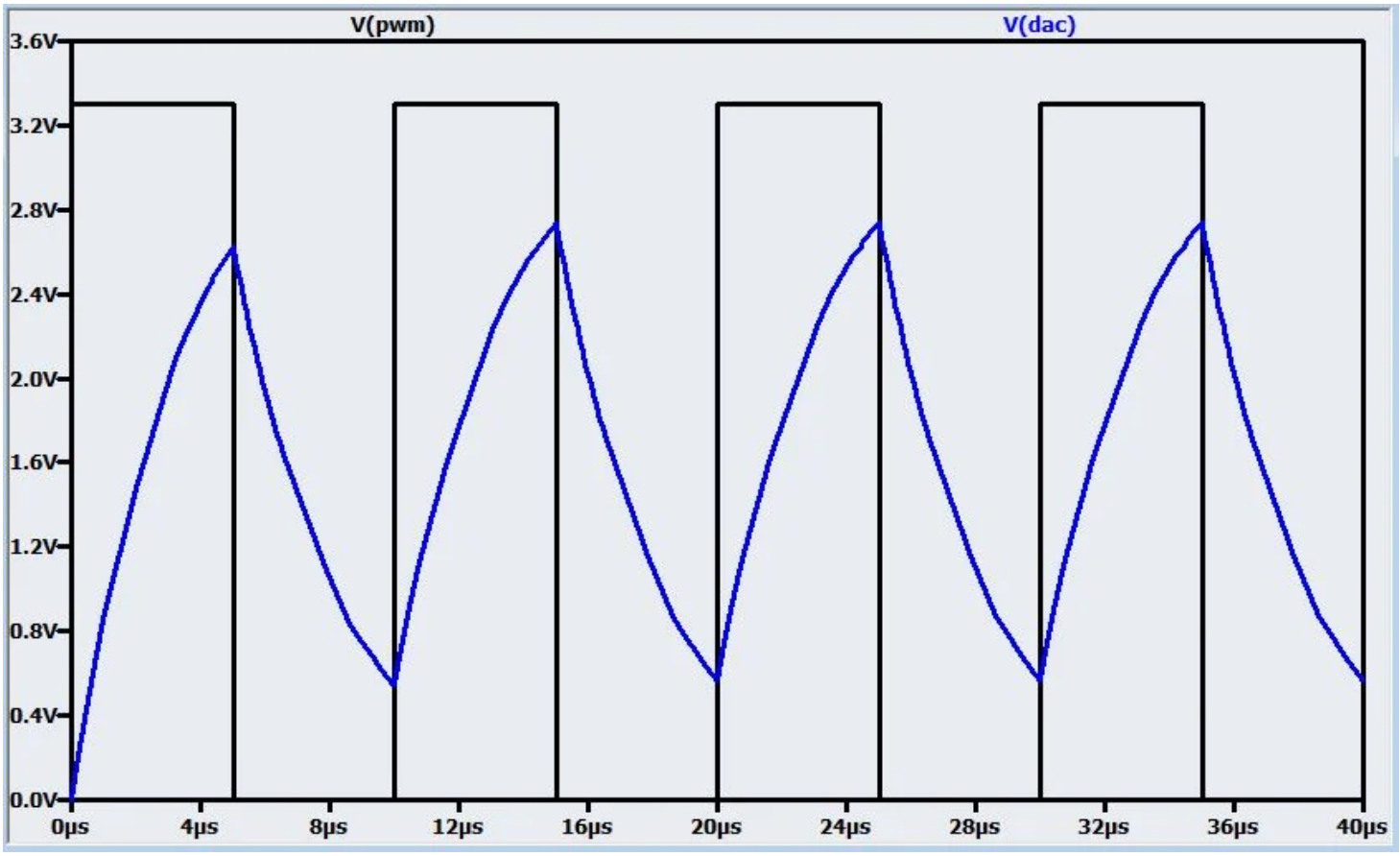
The spectrum certainly changes relative to 50% duty cycle, but one thing doesn't change: the first spike is at the carrier frequency. So regardless of the duty cycle, we have a fairly large frequency band—in this case, from DC to 100 kHz—in which the low-pass filter can transition from no attenuation to significant attenuation.

Ripple and Response with One Pole

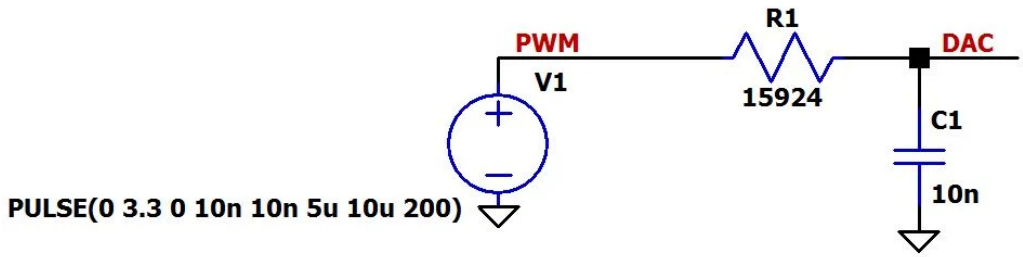
Let's see what kind of DAC quality we can get with a basic RC filter. We'll start with a cutoff frequency (denoted by f_c) in the middle of the DC-to-carrier band:

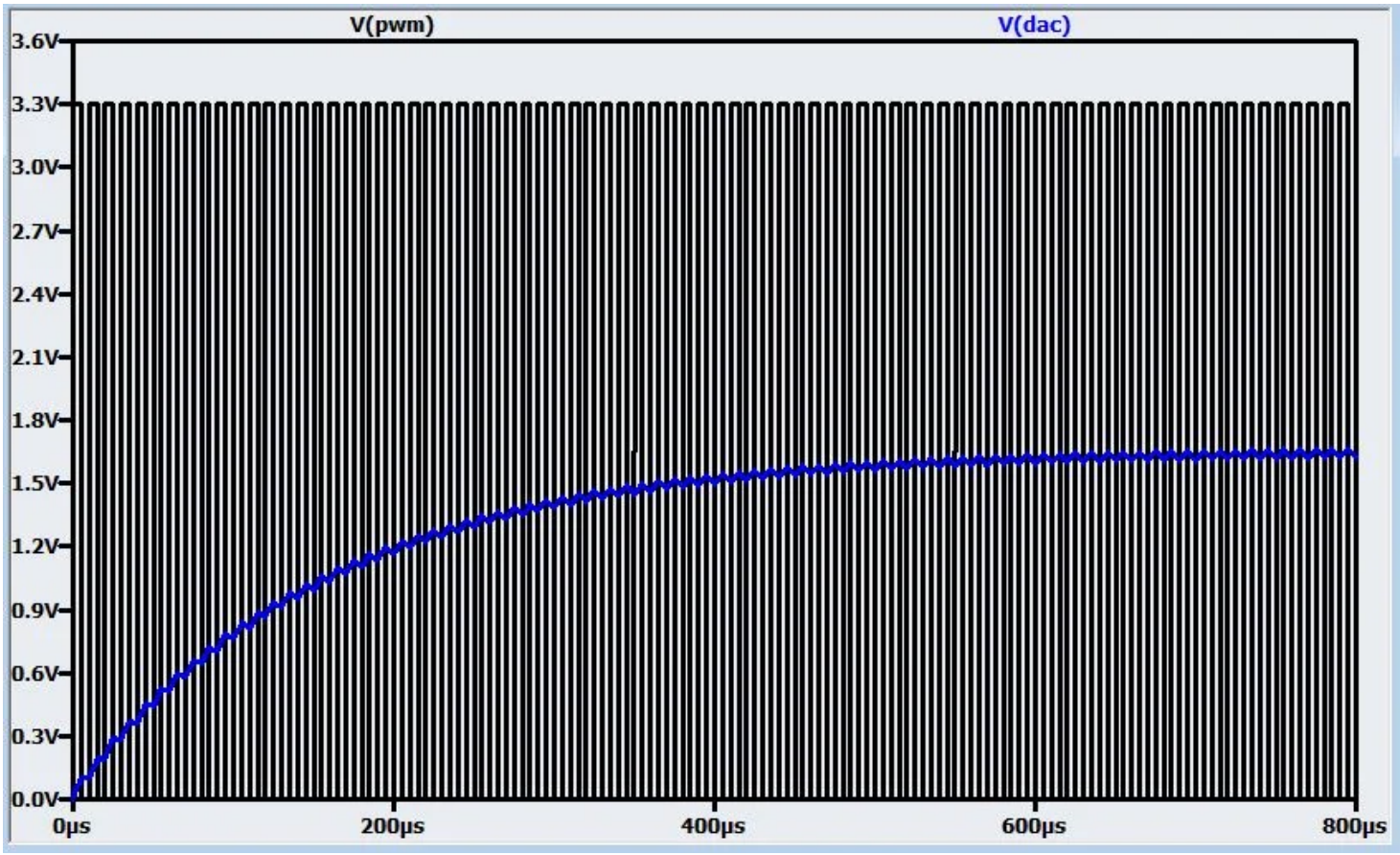
$$f_c = 50\ kHz = \frac{1}{2\pi RC};\ choose\ C = 10\ nF \quad \Rightarrow \quad R \approx 318\ \Omega$$



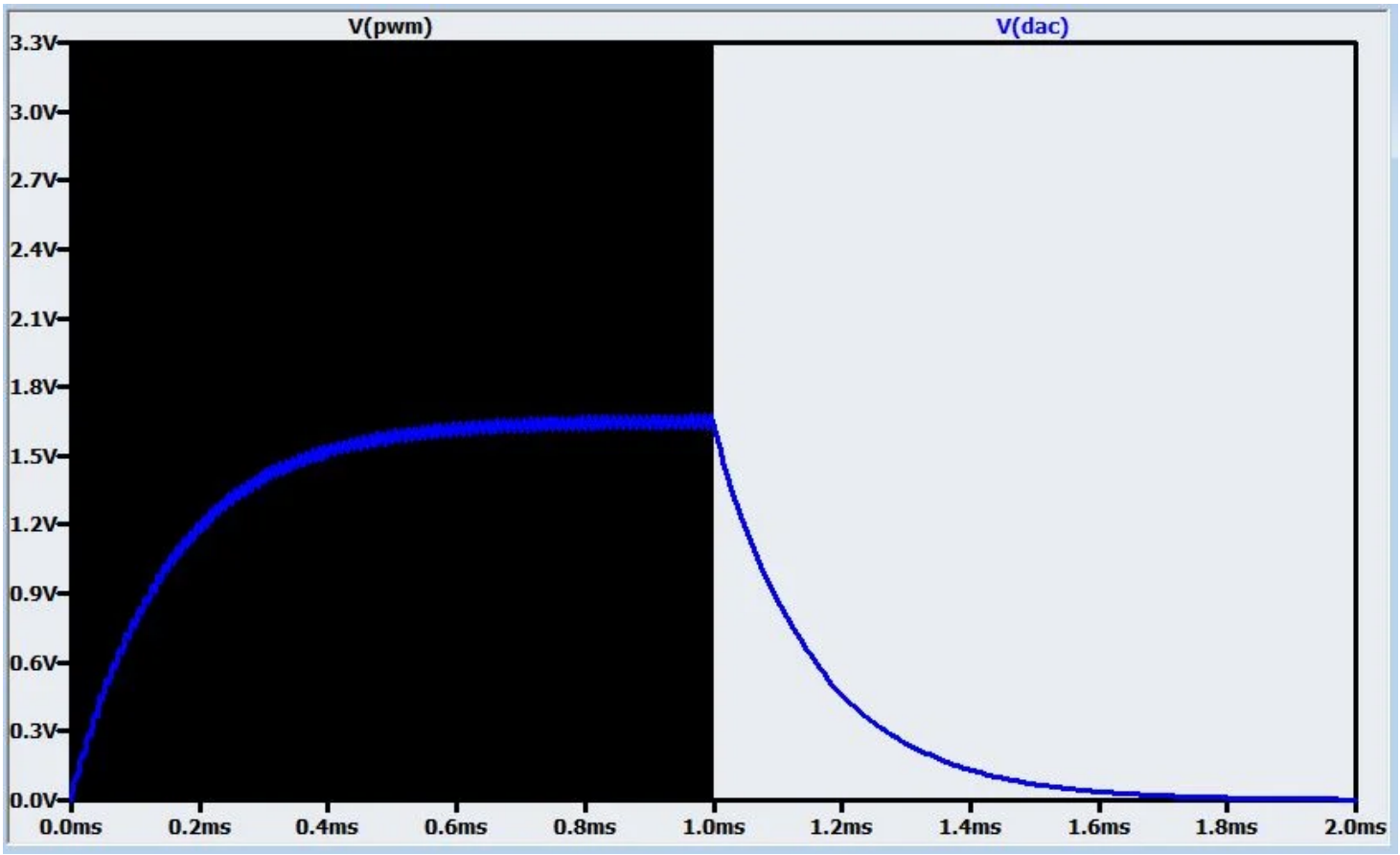


Not so great . . . obviously we need more attenuation than this. Let's move the cutoff frequency to 1 kHz:





The ripple is now much improved, but you probably noticed that we have a new problem: it takes quite a long time for the output to reach the desired DAC voltage. This occurs because the higher resistance in the RC filter not only lowers the cutoff frequency but also increases the time constant—more resistance means less current flowing to the capacitor, and thus the capacitor charges up more slowly. The next plot helps to convey the limitation that this imposes on a DAC:



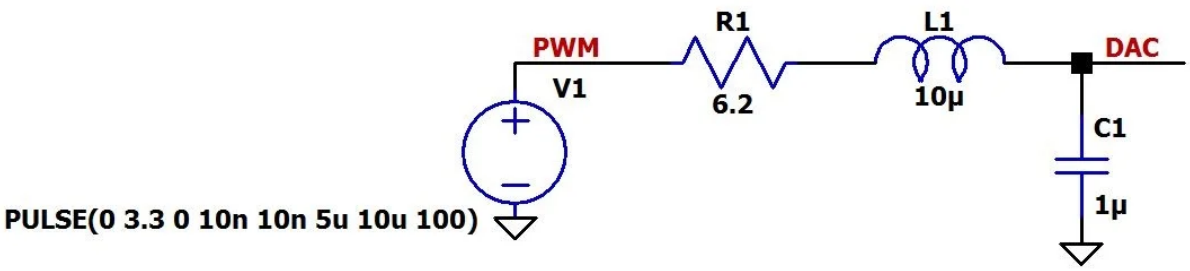
What you see in this plot is rather abysmal "settling time," which is a specification that conveys how quickly the DAC can adjust its output to a newly programmed voltage. The plot indicates that this particular circuit results in a settling time of almost 1 ms when the output increases or decreases by half of the full-scale range. Now don't get me wrong, in many applications 1 ms would be perfectly acceptable, but that doesn't change the fact that this settling performance is extremely unimpressive compared to what you would expect from a typical DAC.

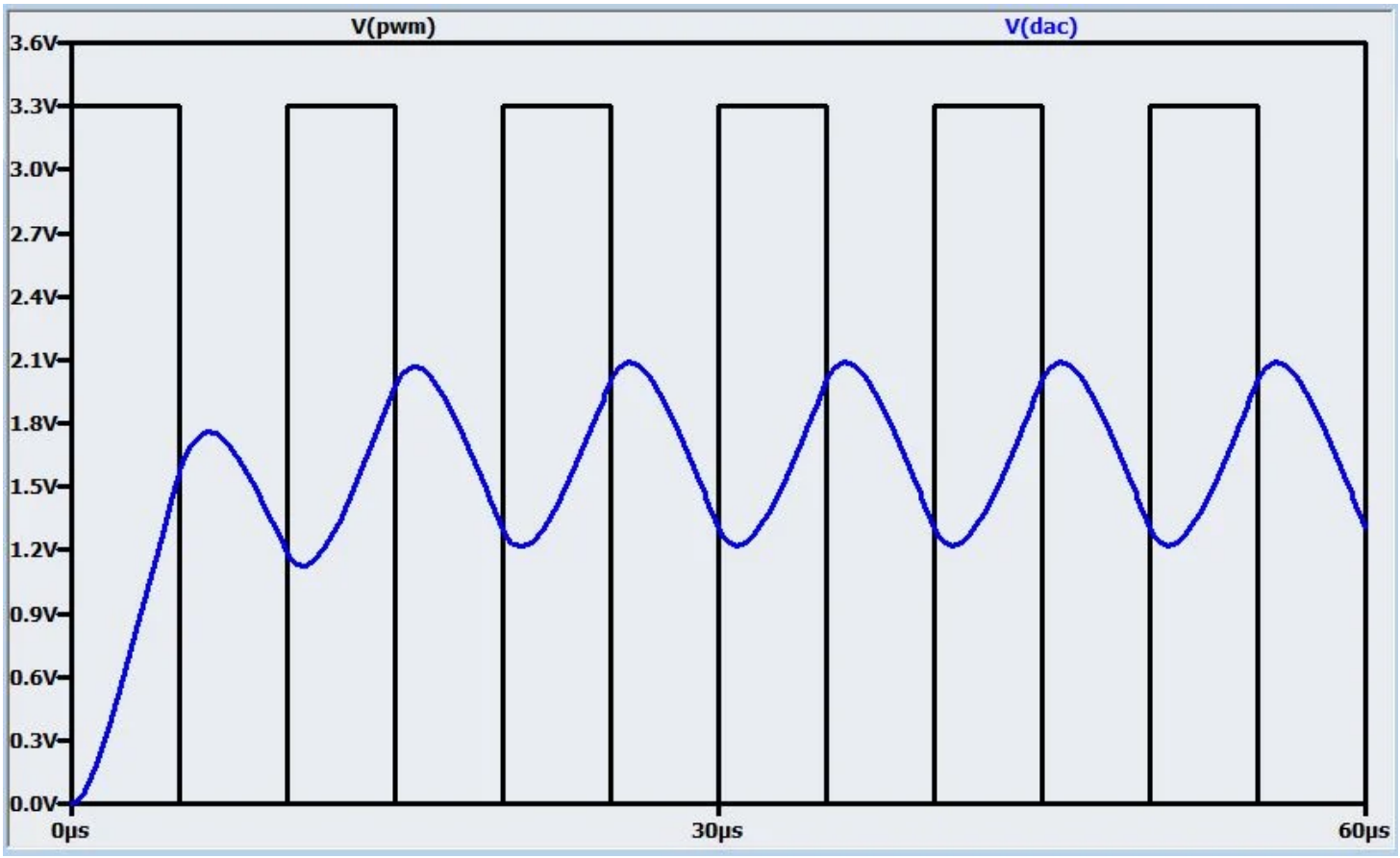
The above results bring us to the first of two dominant trade-offs involved in the design of PWM DACs.

- **Trade-off #1: A lower cutoff frequency means less ripple and longer settling time; a higher cutoff frequency means more ripple and shorter settling time.** So you have to think about your application and decide if you want a DAC that is more responsive or less subject to output ripple.

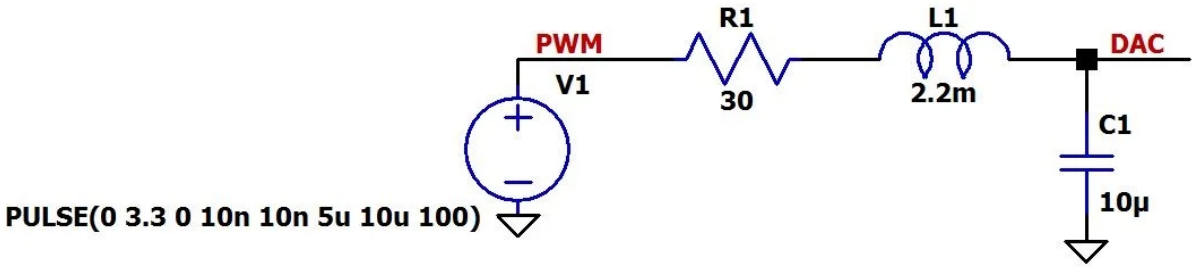
Are Two Poles Better Than One?

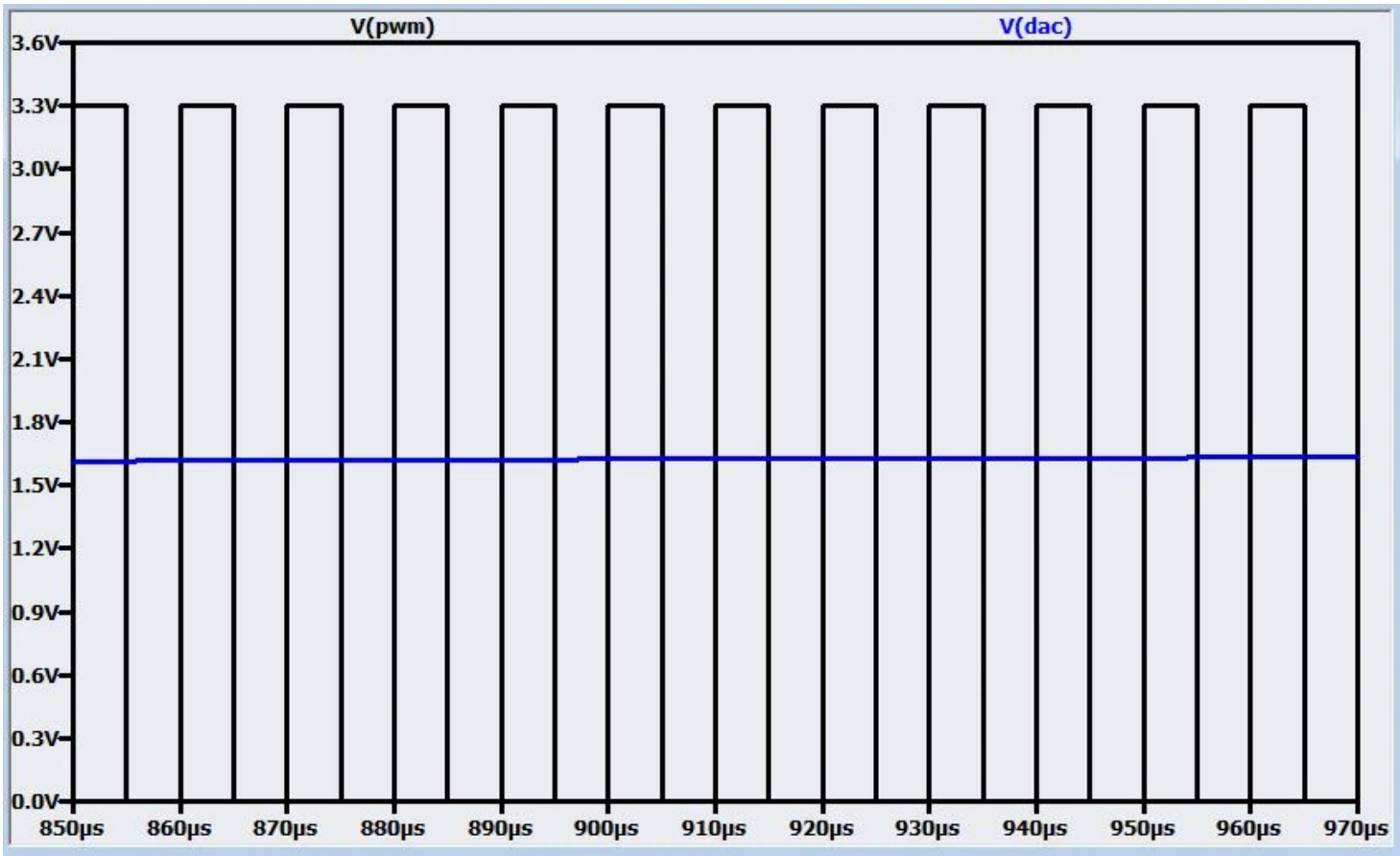
Let's look at two-pole filter results for the same two cutoff frequencies. The following circuit is a critically damped RLC filter with $f_c \approx 50$ kHz (I used this online filter calculator tool to determine the component values):



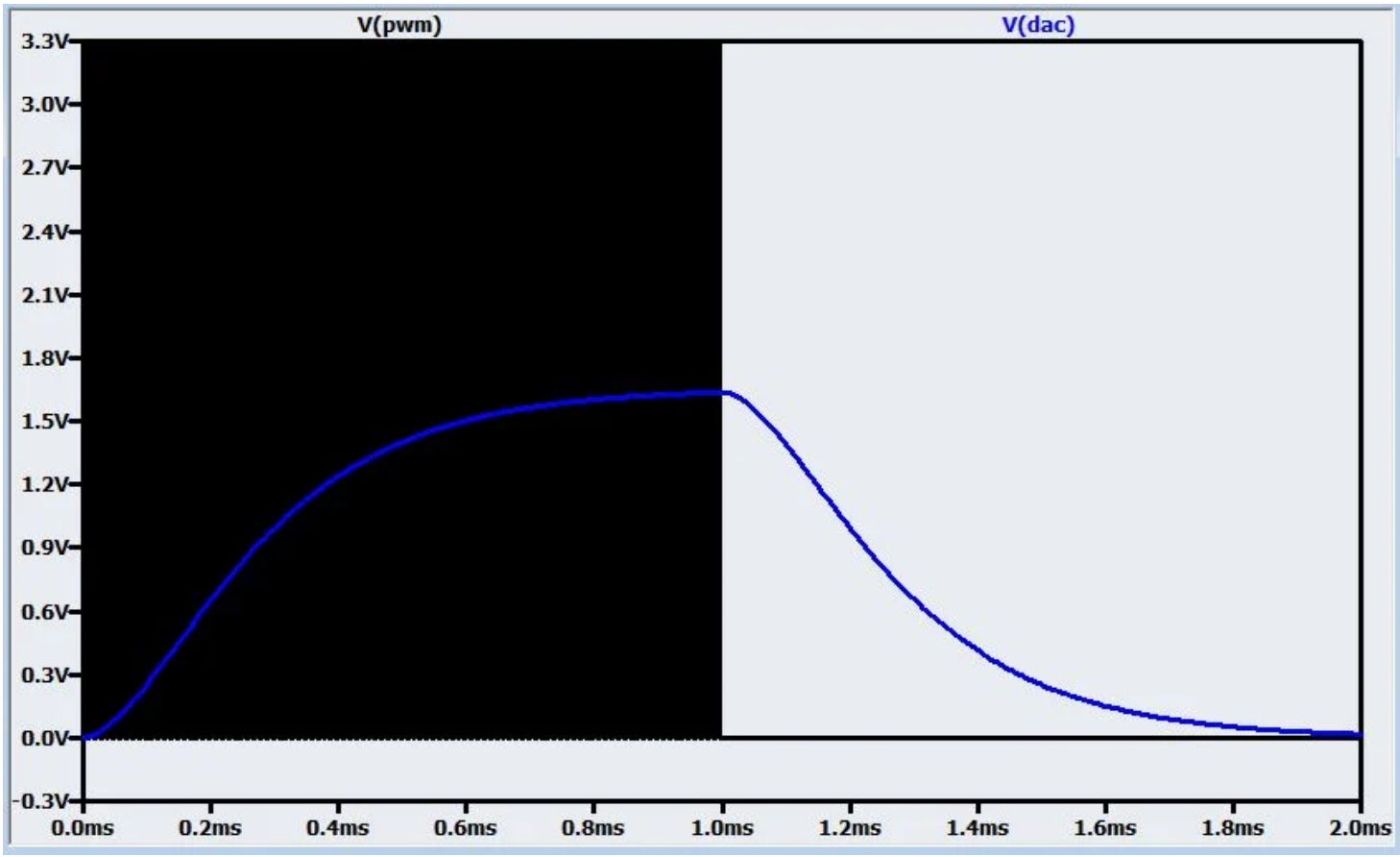


As expected, this is a significant improvement over the single-pole 50 kHz filter; the peak-to-peak ripple has decreased from about 2.15 V to less than 900 mV. Here is the circuit for a critically damped RLC filter with $f_c \approx 1$ kHz:





Here we have almost eliminated the ripple; if you were to zoom in, you would find that the peak-to-peak ripple is only about 500 µV. But now we have the settling-time problem again (remember trade-off #1):

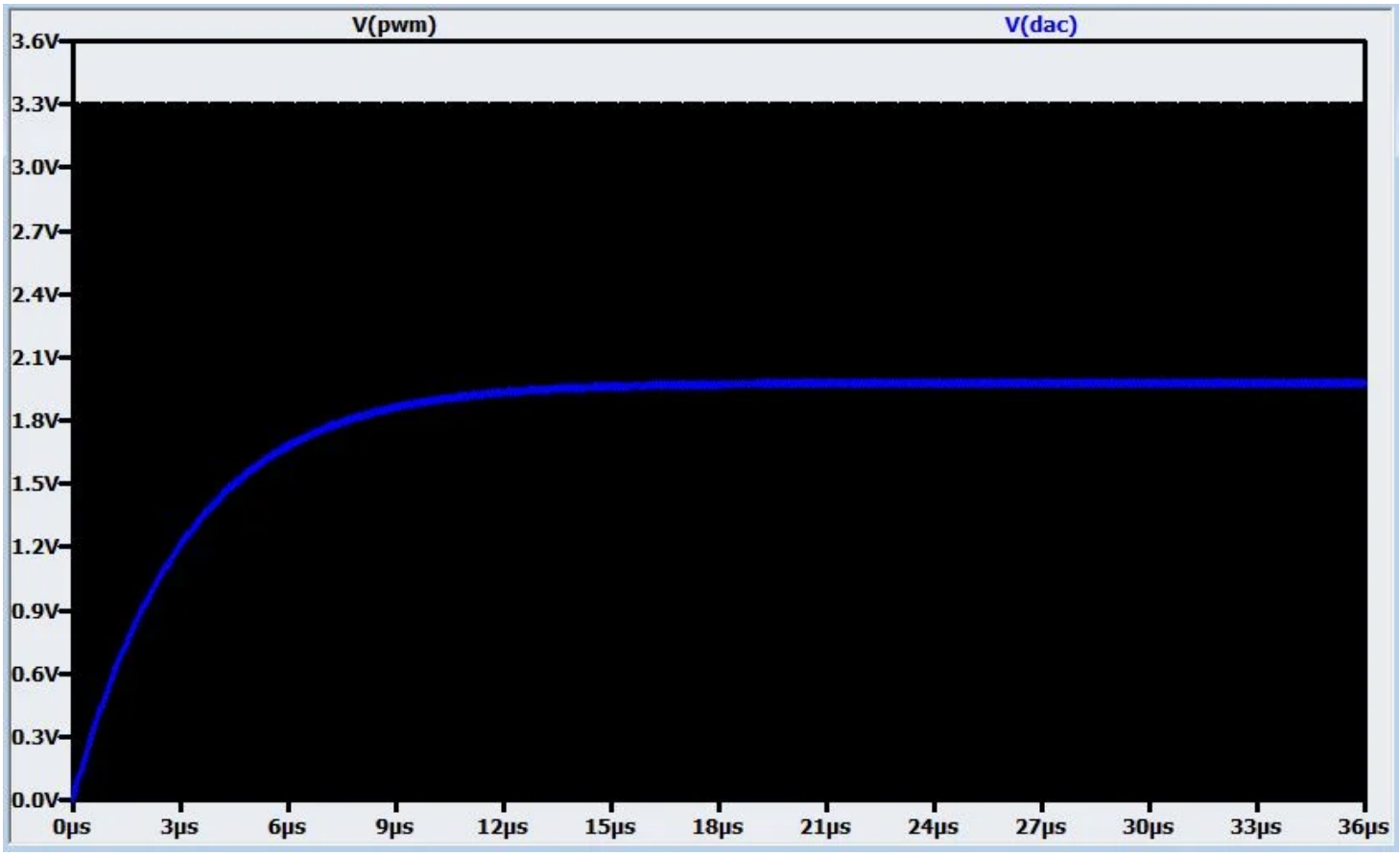


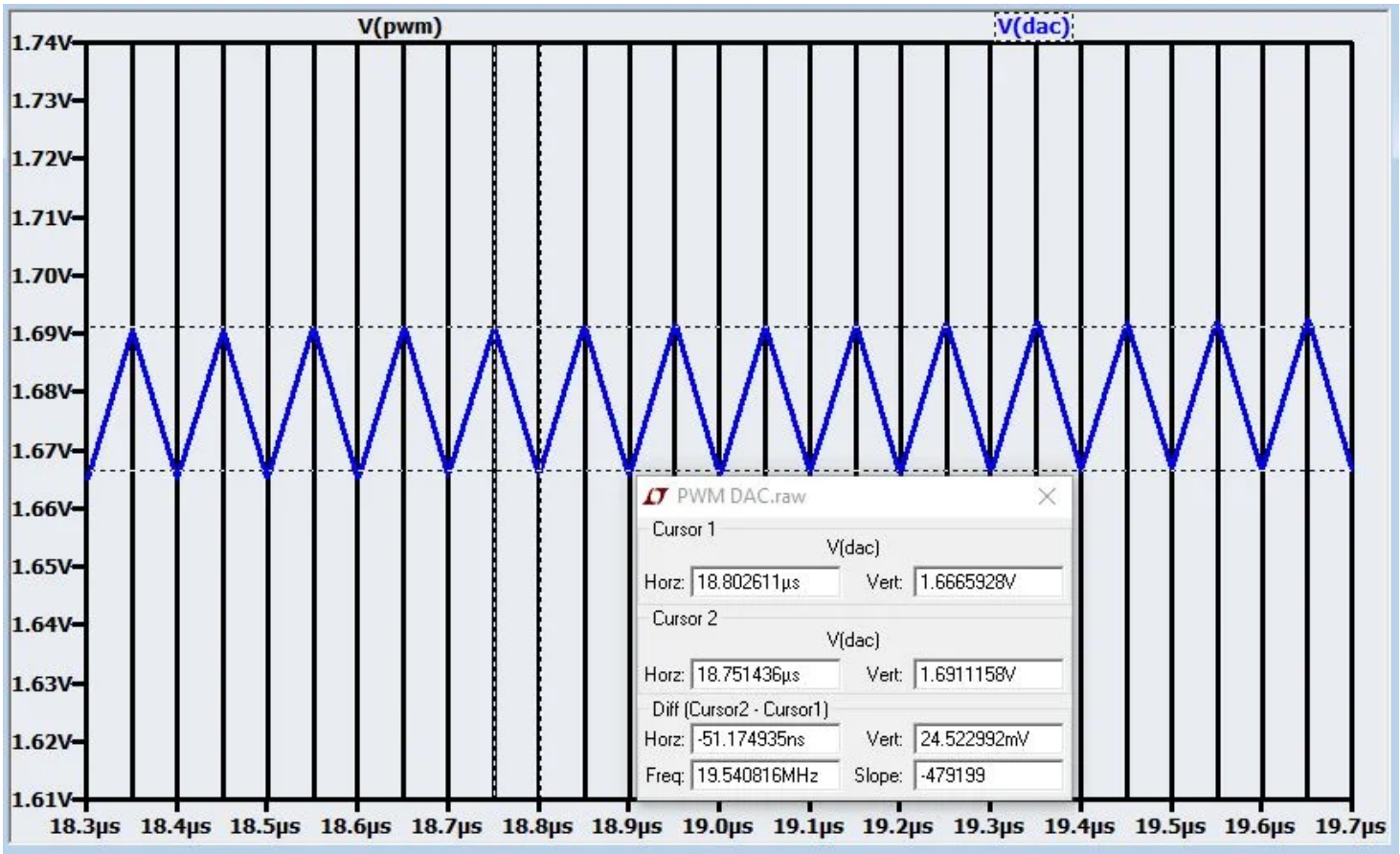
At this point you might be thinking about how you could improve this filter to achieve fast response combined with low ripple. Perhaps you noticed that the previous circuit needed 2.2 **milli**henries—that's a hefty inductor. So what about an active filter instead? Sallen–Key? Maybe a Sallen–Key followed by an RC filter? Wait, why not just use a switched-capacitor filter? Four poles, or even five or seven. . . . This brings us to the second trade-off:

- **Trade-off #2: Higher-order filters improve performance, but they also increase cost and complexity. Instead of spending time and money implementing a fancy filter for a mediocre PWM DAC, we should just use an external DAC!** In my opinion, you shouldn't go beyond one pole. External DACs (and microcontrollers with integrated DACs) are so affordable and widely available that the PWM DAC loses its appeal if you can't meet your performance requirements with an RC filter.

An Easy Way to Improve Your PWM DAC

We need not be discouraged by trade-off #2, because there is a straightforward way to get more performance out of an RC filter: just increase the frequency of the PWM signal! Remember, the PWM spectrum is empty from DC to the carrier frequency. Thus, a higher carrier frequency means a wider band in which the filter's response can roll off—same filter, same settling time, more attenuation. Let's go back to our RC filter with $f_c \approx 50$ kHz, and let's increase the carrier frequency to 10 MHz. Here are the results:


V(pwm)
V(dac)
3.6V
3.3V
3.0V
2.7V
2.4V
2.1V
1.8V
1.5V
1.2V
0.9V
0.6V
0.3V
0.0V
0µs
3µs
6µs
9µs
12µs
15µs
18µs
21µs
24µs
27µs
30µs
33µs
36µs


V(pwm)
V(dac)
1.74V
1.73V
1.72V
1.71V
1.70V
1.69V
1.68V
1.67V
1.66V
1.65V
1.64V
1.63V
1.62V
1.61V
18.3µs
18.4µs
18.5µs
18.6µs
18.7µs
18.8µs
18.9µs
19.0µs
19.1µs
19.2µs
19.3µs
19.4µs
19.5µs
19.6µs
19.7µs
PWM DAC.raw
Cursor 1
V(dac)
Horz: 18.802611µs
Vert: 1.6665928V
Cursor 2
V(dac)
Horz: 18.751436µs
Vert: 1.6911158V
Diff (Cursor2 - Cursor1)
Horz: -51.174935ns
Vert: 24.522992mV
Freq: 19.540816MHz
Slope: -479199

The settling time is only about 15 µs, and the ripple is only 25 mV (compared to 2.15 V when we used a 50 kHz filter with a carrier frequency of 100 kHz).

Practical Limitations

Before we finish up, I should point out that these idealized simulations don't reveal a major source of *non*ideal PWM DAC performance—namely, unreliable and hence unpredictable logic-high and logic-low voltages. The analog output voltage is directly proportional to the digital PWM amplitude, and thus variations in the PWM signal's actual logic-high and logic-low voltages will lead to corresponding variations in the DAC voltage. This problem is particularly relevant to battery-powered applications; if a microcontroller is powered directly by a battery, the logic-high voltage will be gradually decreasing as the battery discharges. Even with a regulated supply, though, you may not know the exact supply voltage—a regulator with ±2% accuracy means a DAC voltage with (at best) ±2% accuracy. And even if you have an extremely accurate voltage regulator and no significant supply variations caused by battery discharge or changes in environmental conditions, still, the actual logic-high and logic-low voltages can be affected by the operational state of the device generating the PWM signal (usually a microcontroller). One way to mitigate this problem is to use an external buffer IC that will help the PWM signal to maintain predictable voltage levels, but at that point you are in trade-off territory again—should you spend 40-some cents on a buffer IC, or 71 cents on a tiny 8-bit DAC?

Conclusion

We have covered the two major trade-offs that govern a PWM DAC design, and we have seen that a higher carrier frequency is a great way to improve performance. With a high-speed microcontroller that provides 16 bits of PWM resolution, you could make a pretty decent DAC with nothing more than an RC filter. In the next article we will employ the SAM4S Xplained Pro starter kit to explore the more practical side of PWM digital-to-analog conversion.
