

The Ten Commandments of Digital Control (Part 3)

Dave Wilson, Motion Products Evangelist, Texas Instruments

We are now halfway through our countdown of the “Ten Commandments of Digital Control”. Yesterday we discussed the Analog-to-Digital Converter, and more specifically, the timing associated with acquiring input samples. But there is even a bigger issue looming in the weeds that can bite you if you aren’t careful. Once the ADC conversion is complete, what do the results mean? How well do you really know what the input signal is, based on looking at these bits? How many bits do you really need, and how many bits can you really trust? So now that the stage is set, let’s continue our countdown with Commandment number 6:

6. **Don’t confuse accuracy with resolution.**

The ADC is a critical part of your digital control system, but it is often misunderstood. Unfortunately this situation isn’t helped much by misleading claims on some ADC data sheets, where the “real” number of ADC bits is padded with “marketing bits”. According to Dictionary.com, accuracy is defined as “the condition or quality of being true, correct, or exact; freedom from error or defect; precision or exactness; correctness”. However, resolution is defined as “the act or process of resolving or separating into constituent or elementary parts”. Clearly they are not the same thing! When applied to an ADC, accuracy is related to how closely the digital result can be correlated to the actual input voltage. However, resolution is related to identifying the constituent or elementary blocks that make up the digital answer. Ideally this is the LSB value. For example, a 12-bit ADC will ideally have better resolution than a 10-bit ADC, but the accuracy may or may not be better, depending on its gain and offset errors.

While both specs are important in a digital control system, I would argue that resolution is the more important of the two. I have three reasons for saying this:

1. Calibration procedures can be applied to significantly improve the accuracy of the ADC. However, no amount of calibration can affect the resolution of the ADC. In some cases, it is possible to improve resolution by averaging multiple samples, but this is not practical when dealing with rapidly changing signals which are common in many motor control applications.

2. Inaccuracies in some closed-loop applications can be “servoed out”. For example, in a cascaded velocity loop, the output of the velocity controller sets the commanded value for the current controller. If the converted value of current is off so that it really doesn’t represent the true current, the loop doesn’t care. The velocity loop will just command more or less current until the motor generates the required torque to fix the velocity error.

The Ten Commandments of Digital Control (Part 3)

Published on Electronic Component News (<http://www.ecnmag.com>)

3. Finally, it is uncommon for ADC accuracy to be the weak link in the signal chain anyway. Once you compensate for the overall gain and offset errors in an ADC, the INL spec is what determines accuracy. Using the ADC on TI's 2803x devices as an example, the INL error is only +/- 2 LSB! This corresponds to a 0.05% tolerance of a full-scale analog signal applied to the input of the ADC! In most cases, by the time you add up the worst-case inaccuracies of all the components in your amplifier circuit, you are probably way over this value to begin with.

But analog circuits do have one significant advantage...resolution. The resolution of an analog signal is theoretically one electron! Clearly the ADC can't compete with this. If the resolving power of your ADC isn't high enough, you end up with quantization noise in your waveforms which degrades performance. This can result in cycle-limit hunting, as the servo loop tries in vain to converge to a value which is in between two digital ADC values.

Also, I would argue that an ADC's true resolution is limited by the differential nonlinearity (DNL) spec. If you have missing codes in the specified number of bits for that ADC, then that ADC simply cannot resolve to that number of bits. Period. For example, a competitive microcontroller boasts of an integrated 16-bit converter. But in single-conversion mode, its worst case DNL spec can't even guarantee no missing codes in a 12-bit word length, let alone a 16 bit word! For further discussion about accuracy vs. resolution, read Jose Quinones blog on stepper motors [here](#).

5. Use IIR filters instead of FIR filters in closed-loop applications.

OK, let's shift gears for a moment and move into the digital control algorithm itself.

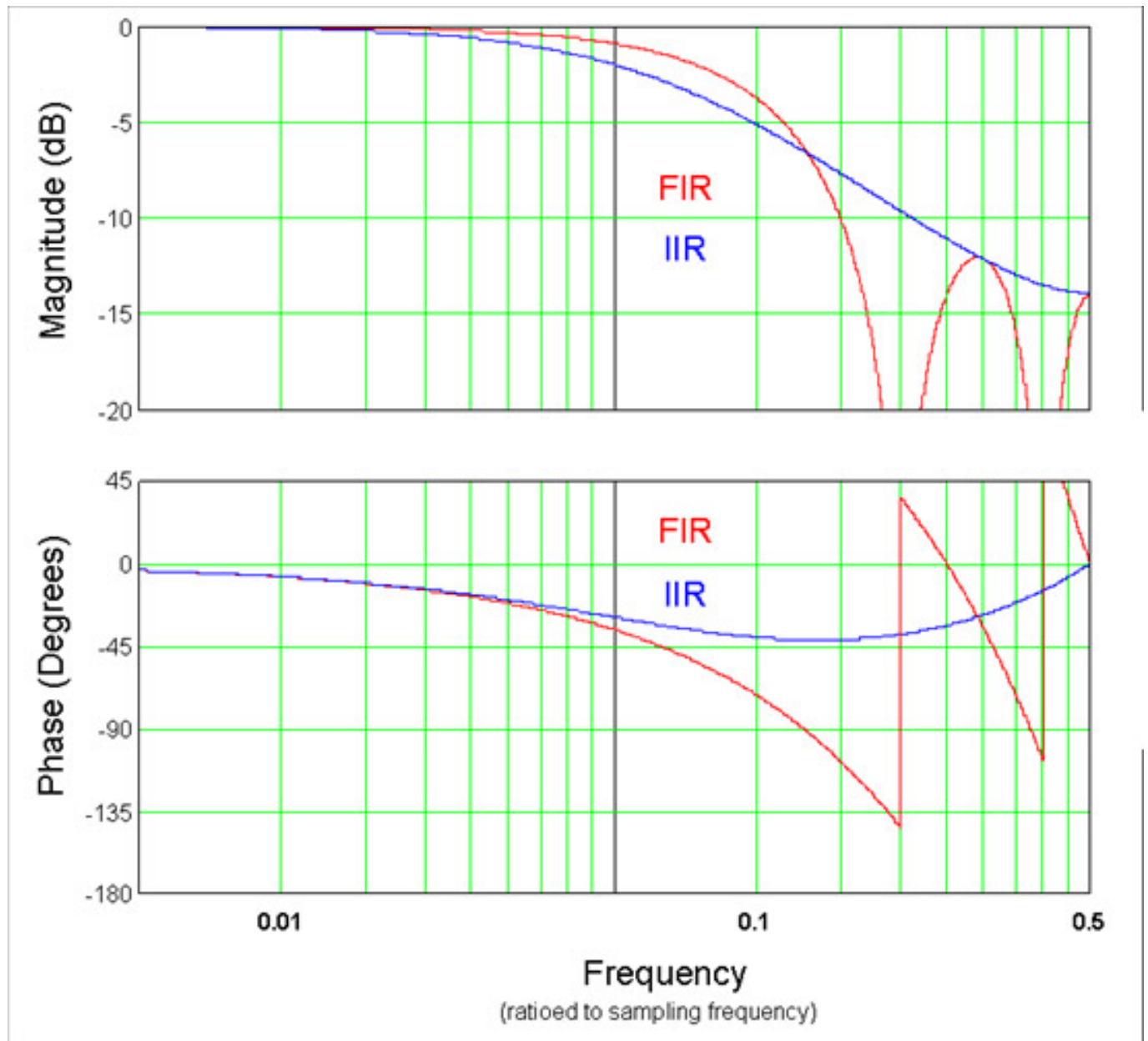
Sometimes it seems like the whole world is conspiring against you in an attempt to make your control loop burst into oscillation! But you can take control of the situation by doing a good stability analysis of your system. I like the bode-plot method best because it just seems more intuitive to me. Using this technique, it is easy to see why IIR filters are preferred in closed-loop structures to FIR filters. For example, below is a bode-plot of the frequency response of an "N-point-averager" (which is in fact an FIR filter), compared to a single-pole IIR filter designed to have roughly the same filtering capability. In particular, notice the phase lag generated by the FIR filter compared to the IIR filter. This is important because phase lag is your enemy when it comes to designing a stable system. Also notice that the filter changes the polarity of the signal each time its frequency passes through a zero in the magnitude plot, as indicated by the 180 degree discontinuities in the phase plot. If the open-loop unity-gain frequency of your control system happens to be close to one of these zeros, you may find that the operation of the loop changes radically with small changes in the power supply voltage or temperature.

Finally, FIR filters take more memory and also more MIPS (one MAC instruction for each tap in the filter) than their IIR counterparts. But they do possess one significant advantage...their transfer function contains only zeros. This is because an FIR filter doesn't have any feedback paths, which means it will always be stable. However, an IIR filter does contain feedback paths that create poles in its transfer function. You must be very careful where you place these poles, or the filter will

The Ten Commandments of Digital Control (Part 3)

Published on Electronic Component News (<http://www.ecnmag.com>)

become unstable. And once any component in your control loop is unstable, the whole loop will be unstable.



Tomorrow we will continue our stroll through the digital control loop, and explore another structure which can not only improve stability, but improve dynamic response as well! In the meantime...

Keep those motors spinning! :-)

Source URL (retrieved on 01/30/2015 - 8:37pm):

<http://www.ecnmag.com/blogs/2011/12/ten-commandments-digital-control-part-3>