Quantifying Velocity Stability

Optimizing Sample Rate for Effective Measurement

By Vince Rolleri, Director of West Coast Business Development

There are two ways that one can quantify “velocity stability”, and almost everyone misses a key aspect on the more popular approach:

1. Instantaneous position error (preferred approach): Nearly all digital servo systems are based on only two quantities – time and position. There is rarely a direct velocity measurement, an exception being rotary motors with tachometers; and when was the last time you saw one of those? As such, the simplest way of quantifying and assessing velocity stability is to set a threshold on the instantaneous position error during a scan. This inherently provides a limit on velocity excursion based on simple $v=dx/dt$. Note that t is quantized by the motion controller’s servo-cycle period, and thus introduces a fundamental limitation on velocity resolution. As you’ll see below, this is why this approach is preferred.

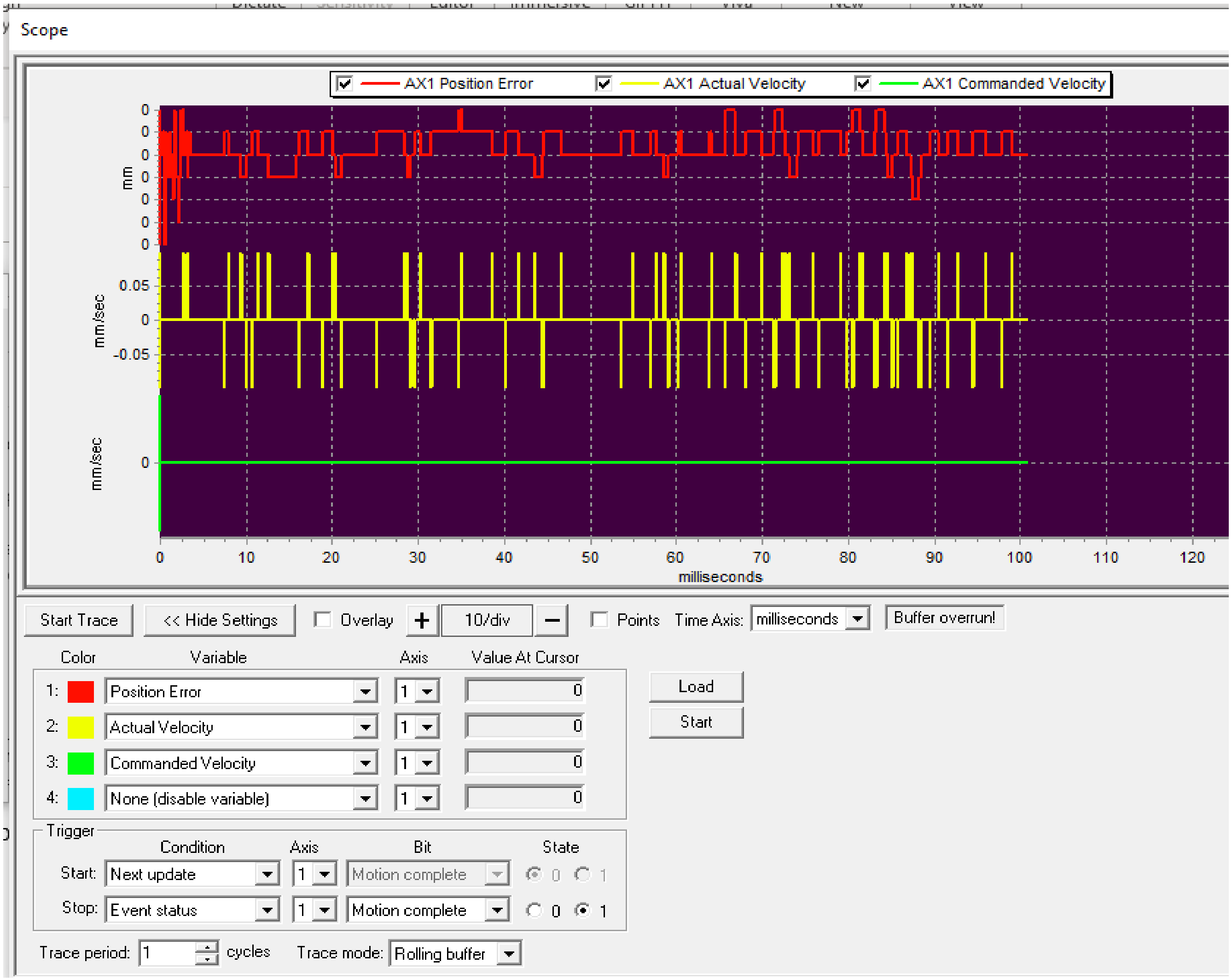
Example: Scan at x mm/sec; Perr must remain < ±n counts.

2. Nominal velocity, velocity error and sampling rate: This is the approach most often seen, but rarely stated completely. You’ll hear someone say, “I need to scan at x mm/sec ± y%. This is an incomplete definition. As above, the speed is quantized by virtue of finite encoder resolution and servo-cycle period. This often introduces appreciable noise, particularly as the nominal velocity decreases. So, to be complete, an associated sample rate must be defined for the data acquisition and resulting velocity calculation. You can think of the sample rate as a low-pass filter. By way of extreme example:
   a. Say a customer wants to scan a SmartStage™XY or DOF-5 Z at 100 nm/sec. When the data sampling period is minimized, it is equivalent to the servo-cycle period, i.e. 51.2 µsec. The resolution of these products is 5 nm. Therefore, the nominal velocity equates to 20 cts/sec. However, the high data sampling rate effectively maximizes our low-pass filter’s cutoff frequency. Now assume the stage moves perfectly: precisely 100 nm/sec. What you’ll observe is v = “0 mm/sec” for the first ~1000 data samples. At the next data sample, when an encoder count is recorded, you’ll have v = (5e-6 mm/51.2e-6 sec) = 0.097 mm/sec. And this pattern repeats. The result effectively provides no insight into the actual velocity!
   b. With the same scenario, let’s adjust the data sampling period to 1 sec. At a perfect 100 nm/sec actual speed, you’ll measure a perfect v = 100 nm/sec for every data sample. The only thing that changed was the sampling rate, but the results could not stand in starker contrast.

Example: Scan at x mm/sec ± y%, sampled at z Hz.

Actual data for a 100 nm/sec scan, from a Dover Motion DOF-5 Z focusing stage, based on above examples:

Sampling every 51.2 µsec (every servo cycle), we can see the Perr is nearly perfect, within 1 or 2 cts at every servo cycle, but the Actual Velocity is just useless noise, indicating the stage sporadically moving ~±100 µm/sec while otherwise not moving at all.



☐ Overlay + 10/div − ☐ Points Time Axis: milliseconds Buffer overrun!

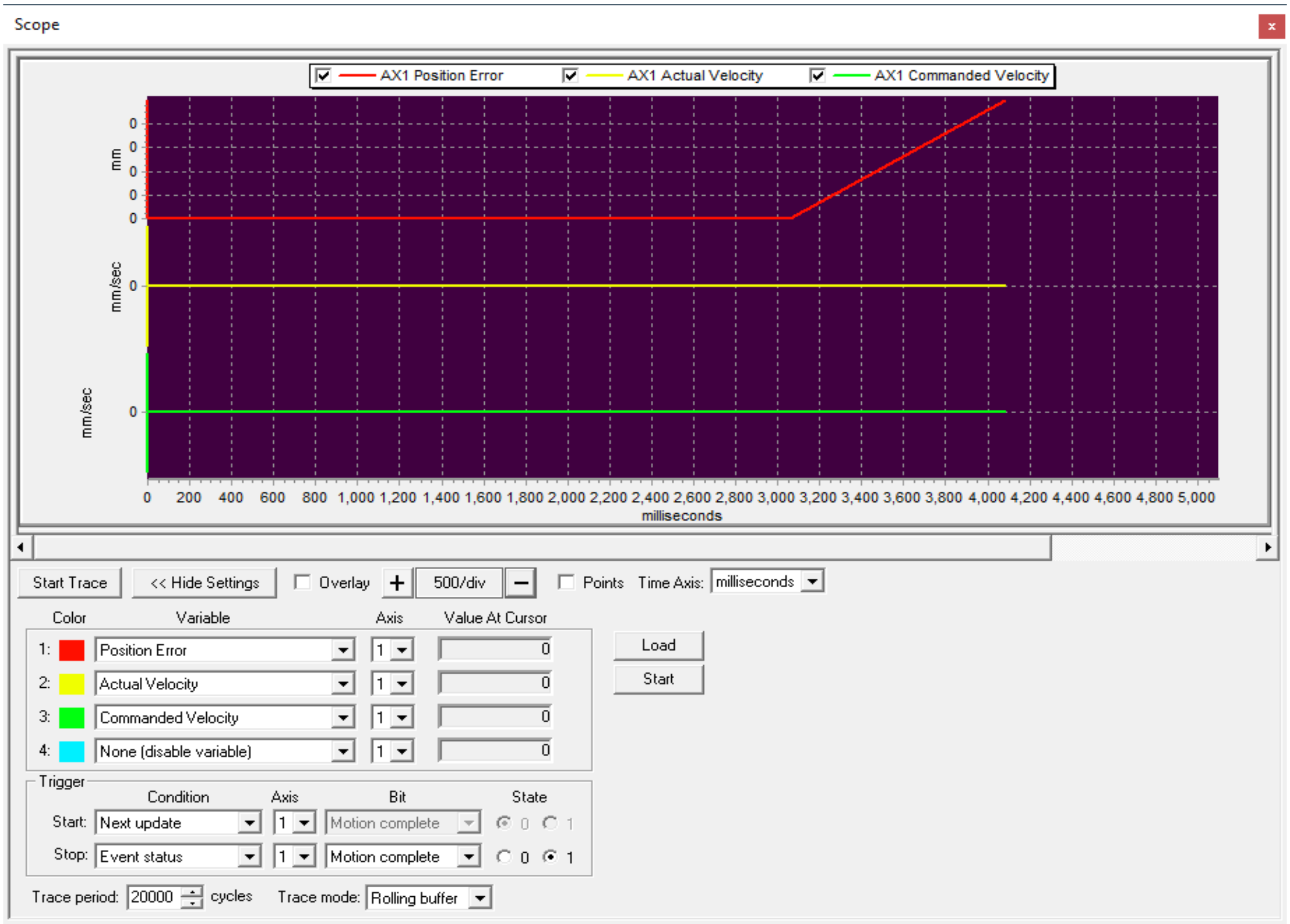
Color	Variable	Axis	Value At Cursor
1: 🟥	Position Error	1	0
2: 🟨	Actual Velocity	1	0
3: 🟩	Commanded Velocity	1	0
4: 🟦	None (disable variable)	1	0

Trigger

	Condition	Axis	Bit	State
Start:	Next update	1	Motion complete	● 0 ○ 1
Stop:	Event status	1	Motion complete	○ 0 ● 1

Trace period: 1 cycles Trace mode: Rolling buffer

Same as above, sampled at ~1 sec (every 20000 servo cycles), note the ideal response - looks pretty righteous:



☐ Overlay 500/div ☐ Points Time Axis: milliseconds

	Color	Variable	Axis	Value At Cursor
1:	🟥	Position Error	1	0
2:	🟨	Actual Velocity	1	0
3:	🟩	Commanded Velocity	1	0
4:	🟦	None (disable variable)	1	0

Trigger

	Condition	Axis	Bit	State
Start:	Next update	1	Motion complete	● 0 ○ 1
Stop:	Event status	1	Motion complete	○ 0 ● 1

Trace period: 20000 cycles Trace mode: Rolling buffer

Last bit: the appropriate sample period is generally related to the application. In the case of optical imaging, it is related to the rate of image acquisition (e.g. camera frame rate or clocking rate of the vertical transfers in a TDI scheme). But often it's just so much easier and more intuitive to think about instantaneous error.