Evolution of Sinusoidal Encoder Interpolators

Introduction

Sinusoidal encoders have become increasingly common in precision motion control systems over the last 25 years. As the precision requirements for these systems have steadily increased over that period, the demands on these encoders and the circuitry that processes their signals have increased as well.

The processing circuits for these encoders are usually called “interpolators” because they can yield many possible position values per signal cycle of the encoder. The design of these interpolators has evolved considerably over the years to meet the increasingly stringent requirements of users for speed, resolution, and accuracy.

Note that the high resolution produced by interpolators can be valuable even in applications that do not require very high position accuracy. The high position resolution reduces “quantization noise” in the feedback, especially in the estimated velocity, permitting higher servo gains and performance.

Tracking-Loop Interpolators

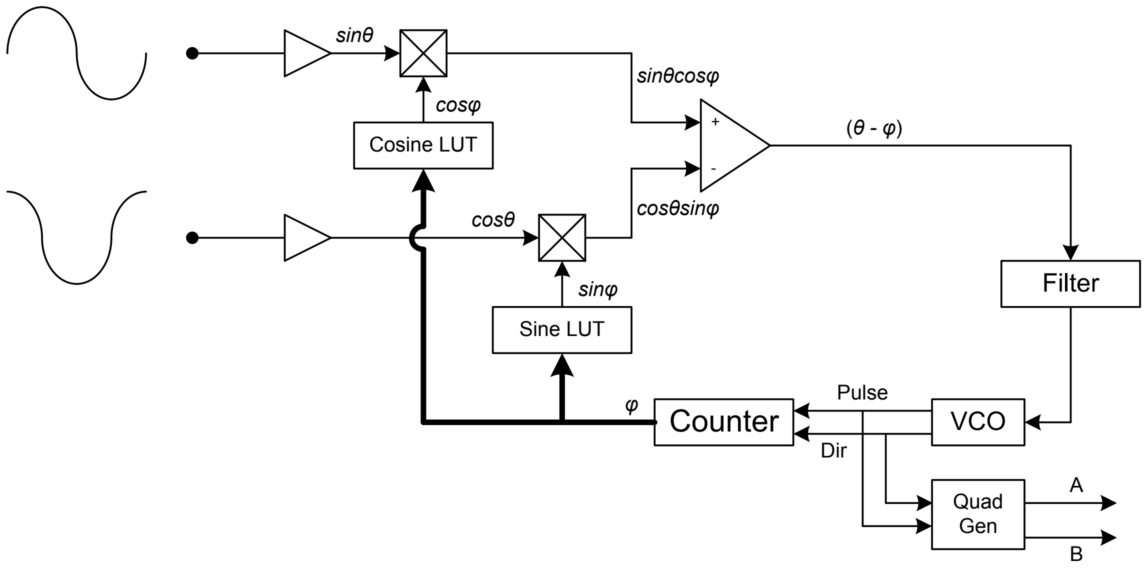
The earliest devices to produce many position states per cycle of the sinusoidal encoder signal employed “tracking loops”, in which hybrid analog/digital circuitry “tracked” the input signals, and intermediate signals in this loop were converted to digital quadrature format for output to the controller. These were similar to the tracking converters used for resolvers, although there was no carrier frequency to deal with in the case of the sinusoidal encoders.

One of the key advantages of this style of interpolator was that it could be used with existing controllers that could only accept digital quadrature signals for position feedback. These tracking interpolators were often produced by the manufacturers of the sinusoidal encoders themselves, so the encoders could be used with standard controllers.

The basic principle of operation of a typical tracking interpolator is shown in the figure below. The voltages from the encoder representing the sine and cosine of the present physical angle θ are multiplied by voltages representing cosine and sine of the tracking angle φ. The difference of these two analog products is the sine of the difference between θ and φ, which for small angle differences is effectively the difference itself.

The voltage representing this difference is input to a voltage controlled oscillator (VCO), which outputs a pulse frequency proportional to the voltage. This pulse train drives a counter whose value represents the tracking angle φ. The digital value of the counter selects sine and cosine values from lookup tables (LUTs), which generate voltages representing the sine and cosine of the estimated angle to close the loop. The number of entries in the lookup tables determines the interpolation factors.

Meanwhile, the pulse and direction signals out of the VCO are transformed to A and B digital quadrature signals for output to the controller. The controller can then treat this as a very high line-count digital quadrature encoder.



Tracking Interpolator Block Diagram

One of the key issues with tracking interpolators is the required signal frequency multiplication. The frequency of the output digital quadrature signals is much higher than the frequency of the input sinusoidal signals. For example, if the sine and cosine signal inputs are at 100 kHz and the interpolator produces 256 states per line of the encoder ("times 256" interpolation), the output quadrature signals would need to create over 25 million quadrature edges per second. With 4 quadrature edges per signal cycle, this requires signal outputs of over 6.25 MHz, which is faster than most controllers can handle.

Another issue is that the tracking loop is error driven, so in general the position represented by the simulated quadrature output signals does not match the measured position. The dynamics of the tracking loop are effectively part of the dynamics of any servo loop it is used in. In high-precision systems, these dynamics can have a noticeable effect on performance. In addition, some tracking interpolators exhibit significant non-linearities, including saturation and quantization irregularities, that can impact performance.

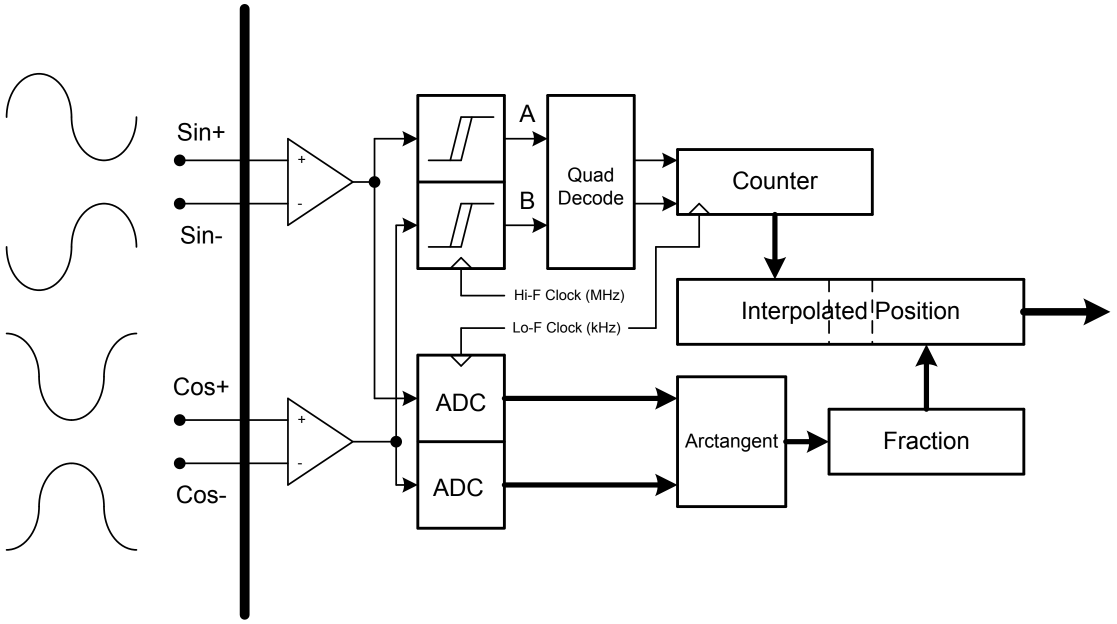
Direct Dual-Sample-Rate Interpolators

In response to the shortcomings of tracking-loop interpolators, controller manufacturers started producing "direct" interpolators, which would sample the sine and cosine signals directly and calculate the position from these signals. These direct interpolators had the advantage of not requiring the generation of any higher frequency signals. Also, without a tracking loop, they did not add any more dynamics or lag to the servo loop.

When these direct interpolators were first introduced, A/D converters of the resolution required to provide the needed interpolation factor could not sample the sine and cosine input fast enough to track all of the changes in the inputs. In terms of digital sampling theory, they could not sample at the required Nyquist rate (over 2 times the signal frequency) for any reasonable speed of the encoder.

So these designs employ a dual sample-rate strategy. The sine and cosine signals are sampled at two different frequencies. Simple comparators – effectively 1-bit ADCs – sample them at very high frequencies, typically measured in MHz. The output of the comparators is digital quadrature at the same frequency as the input sinusoidal signals. This quadrature is decoded and counted just as it would be for a digital quadrature encoder. Since this path is sampled above Nyquist rates for even high encoder frequencies, it can reliably track the signal, although with low resolution. Effectively, this path is used to determine which cycle of the encoder is used for the reported position.

The high-resolution ADCs are sampled only once per servo cycle, typically a few kHz, or a few tens of kHz, synchronously with latching of the quadrature counter. The resulting sine and cosine numbers are used to compute the angle within one cycle with an arctangent calculation. This “fractional line” value is then mathematically combined with the “whole line” value from the other path to yield a complete high-resolution value.



Direct Dual-Sample-Rate Interpolator Block Diagram

This style of interpolator has probably been the most popular over the past 10 to 15 years for very high-precision systems, providing interpolation factors of 1,024, 4,096, or 16,384 states per line of the encoder. It does a very good job processing the “perfect” encoder signal. However, it cannot cope well with distorted and/or noisy encoder signals.

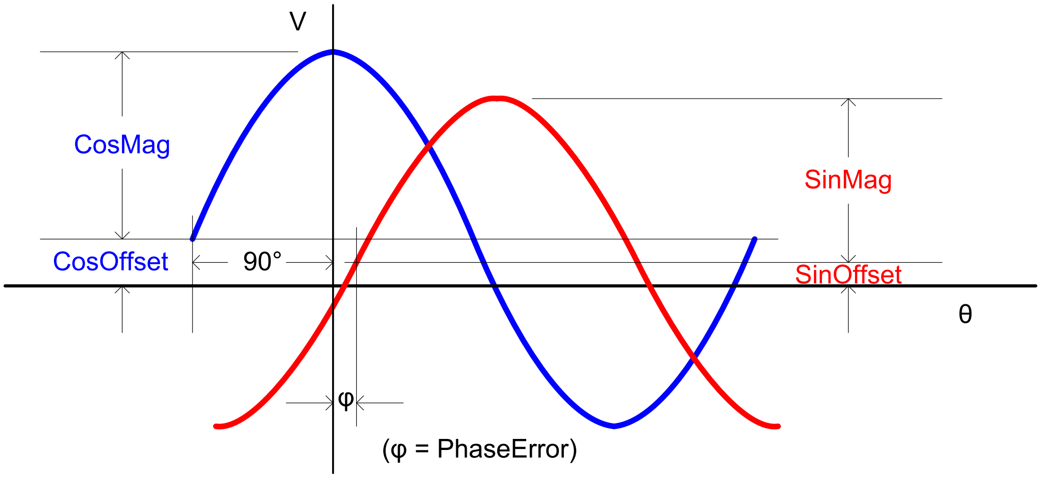
Noise mitigation with these interpolators is typically done either before or after the interpolation. Often, there are low-pass filters on the analog inputs to the interpolator circuits. With many interpolators, these limit the maximum signal frequency due to reduced magnitude and increased phase lag at higher frequencies.

It is also possible to employ a digital low-pass filter on the resulting position values. However, since these values are produced at a low frequency, it is difficult to provide significant filtering without introducing significant lag.

Common Sine Encoder Signal Errors

As machine accuracy requirements get increasingly stringent, machine builders care more and more about any imperfections in the signals from their sensors. In the highest-accuracy machines, it is no longer acceptable to consider the signals to be perfect sine and cosine signals, each centered about zero, of the same magnitude, and exactly 90° apart. Deviations from these ideal signals must be both identified and corrected.

The following diagram shows the most common and significant errors in sinusoidal encoder signals. The voltage of each signal is offset (vertically) from its ideal centering. The magnitudes of the two signals do not match, and they are not separated (horizontally) by the ideal 90°. Each of these imperfections causes a resulting systematic position error in an algorithm that assumes the signals are perfect.



Common Sinusoidal Encoder Signal Errors

It might seem that the resulting position errors amounting to relatively small fractions of the line pitch of the encoder would seldom be significant in most applications, but they can be important to a surprising extent. Many people assume that the frequency of the errors sampled at the servo rate would have a frequency too high for the physical system to respond to.

However, because the ADCs reading these imperfect signals in dual-sample-rate interpolators are sampling well below the Nyquist frequency for the signals, aliasing effects can occur. Like spinning wagon wheels sampled at a low rate by a movie camera appearing to spin much more slowly, the frequency of the resulting sampled errors can be much lower after sampling, brought down to within the bandwidth of the physical system's response.

It is possible – and quite common – to provide terms in the interpolator algorithm to compensate for these signal errors. However, before this can be done, the errors must be properly identified. This can be difficult and time consuming. And even if they are well identified at one location and one time, they can change over both location and time, often in ways that are hard to predict.

It would be wonderful to have circuitry that could both identify these errors on an ongoing basis and then correct the calculated interpolated positions automatically based on the most recently identified error characteristics. However, this capability requires new technology and a new approach.

Direct High-Sample-Rate Interpolators

For a long-time, you could obtain reasonably priced A/D converters that were either high sample rates with low resolutions (flash ADCs), or low sample rates with high resolutions (successive approximation register ADCs). Improvements in sinusoidal encoder interpolators require ADCs that provide both high sample rates and high resolutions. For common use, these ADCs must also be reasonably priced.

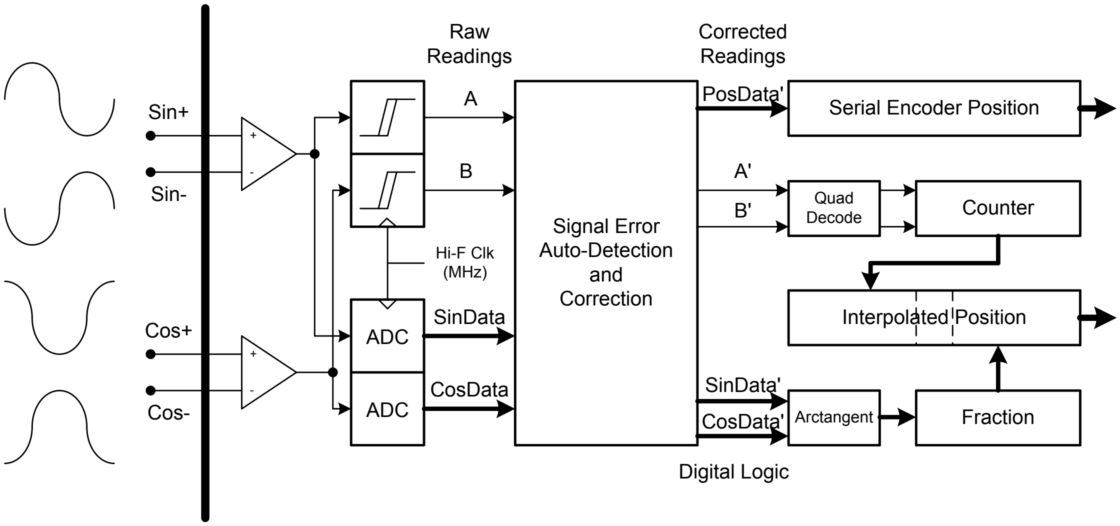
Enabling Technologies

Recently, ADCs meeting these requirements have become available. These ADCs are hybrids between flash and SAR technologies. An initial flash conversion provides most of the resolution (10 – 12 bits). Then a few stages of successive approximation provide the last 4 – 6 bits. All of this is done in a pipelined fashion, so a new full high-resolution value is available every clock cycle, at frequencies of 10 MHz and over, with minimal delay.

This capability permits the ADCs to sample the sinusoidal encoder signals at full resolution at greater than the Nyquist rate even for very high signal frequencies. This provides at least the potential for identifying the signal errors within a cycle, and then correcting for them.

Turning this potential into reality requires digital processing that can keep up with the data provided by these high-frequency ADCs. This processing can detect the sub-cycle patterns in the data can compensate for issues found. Steady advances in the capabilities and cost-effectiveness of customizable digital logic technologies, particularly in field-programmable gate arrays (FPGAs) have provided practical solutions to this part of the problem.

The following figure shows a block diagram of an interpolator using these technologies. In this implementation, the auto-correcting circuitry is interfaced to digital circuitry originally designed for an older dual-sample-rate interpolator (at lower right), providing it with simulated signals from a “perfect” encoder. This interface is not necessary for many purposes, as the correcting circuitry can provide the corrected interpolated position directly, as shown in the upper right.



Auto-Correcting Interpolator Block Diagram

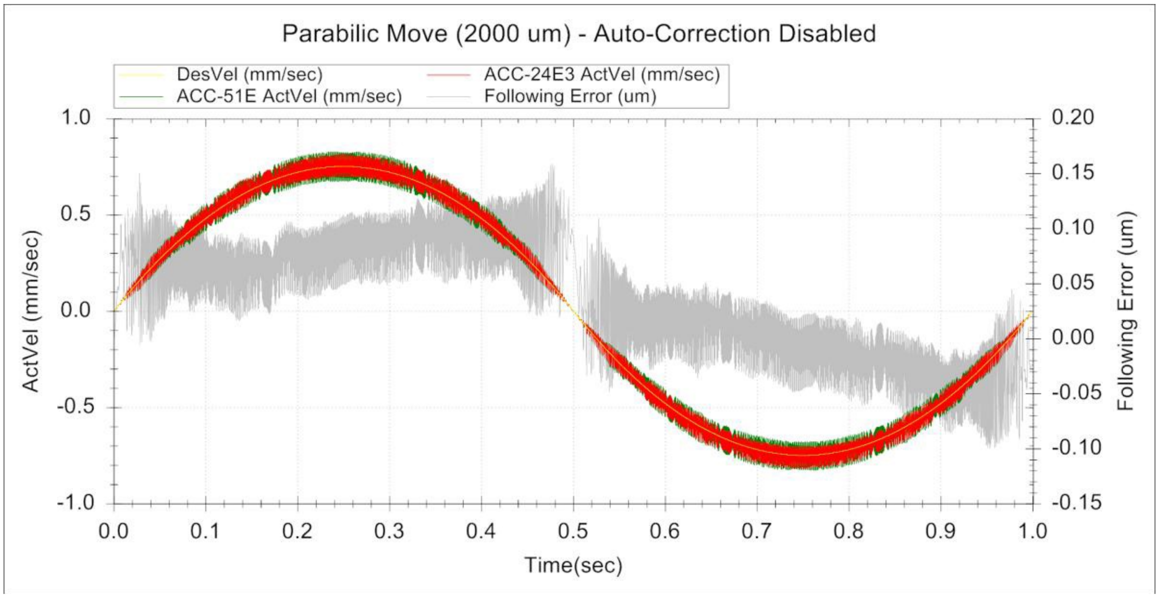
Error Identification

At most speeds, the position error patterns produced by the common signal imperfections are of far too high frequencies to be from actual physical motion, and so can be properly identified as measurement errors. Furthermore, the pattern of position error resulting from each of the four most significant imperfections – sine offset, cosine offset, magnitude mismatch, and phase error – is distinct from the others. (The patterns are sinθ, cosθ, sin2θ, and cos2θ.)

These facts permit the digital logic to automatically detect these errors on an ongoing basis and correct for them, even as these errors change at different locations on the encoder, and over time and temperature. Both automatically corrected position values and simulated digital signals equivalent to a "perfect" encoder can be transmitted to subsequent controller circuitry.

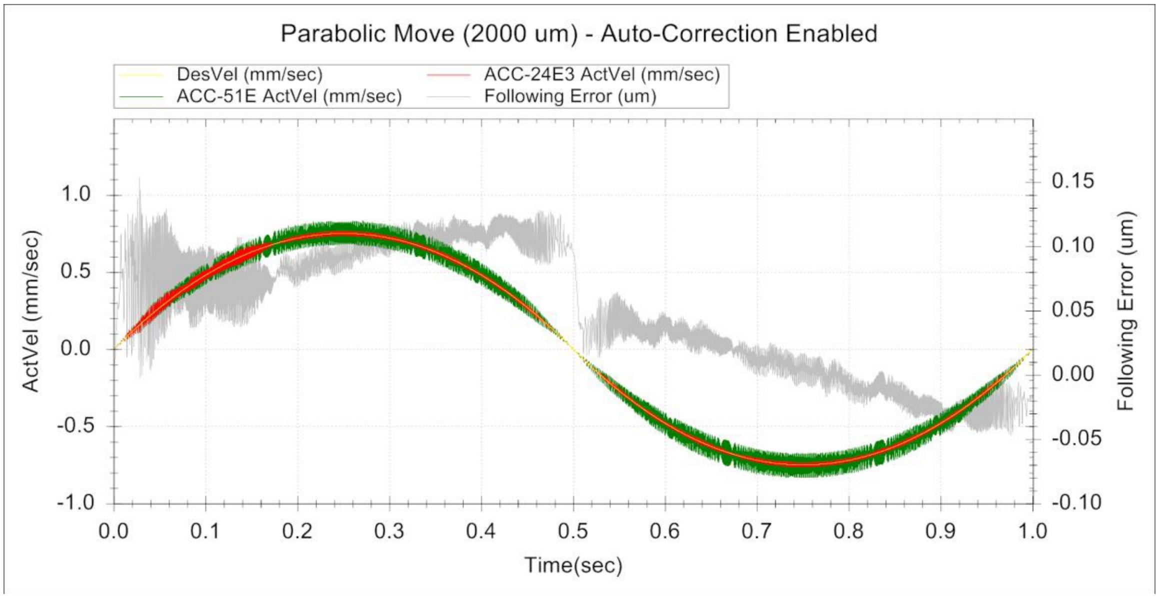
The effect of the corrections can be seen easily with simple moves. Tests run on the same physical system using old and new Delta Tau interpolators illustrate the improvement clearly. The tests were done using a Parker 110-2 I-Force ironless linear brushless motor on a large granite block with feedback from a Heidenhain LIP-401R linear scale with a 2-micron line spacing. The motor is driven by a Varedan VL-4115-01-RA sine-input linearly modulated amplifier. The tests compare the older ACC-51E dual-sample-rate interpolator with the new ACC-24E3 auto-correcting interpolator.

This first plot shows the actual velocity and servo error for a parabolic velocity-profile move with the auto-correction function disabled in the new interpolator. The actual velocity from the new interpolator is only slightly improved from the older interpolator. The oscillation in following error is typically about 80 nanometers.



Parabolic Move Results with Correction Disabled

This next plot shows the actual velocity and servo error for this same move with the auto-correction function enabled in the new interpolator. A dramatic reduction in the oscillation of the actual velocity reported by the new interpolator can be seen. This test begins with no correction at all, and it is not until the move reaches a speed where the effect of the signal errors can reliably be separated from physical vibration (at about 0.15 seconds) that the corrections are first introduced. Note that when the velocity goes below this speed again, the corrections are not removed (but they are not updated). The variation in following error once the corrections have been introduced is reduced to about 20 nanometers. The velocity results from the older interpolator are shown for reference.



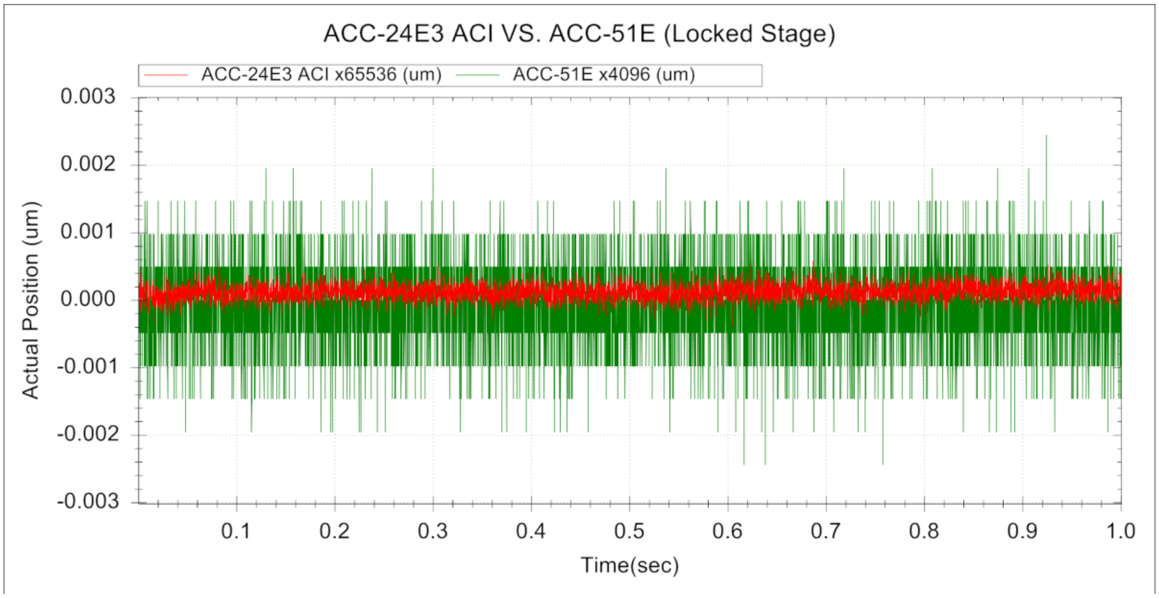
Parabolic Move Results with Correction Enabled

Noise Mitigation and Resolution Enhancement

In addition to providing the capability to detect these repeatable errors, the high-frequency sampling helps with random errors, of the type that result from electromagnetic noise. A simple running average of the position values calculated at this high sampling rate reduces the noise level dramatically while introducing only a minimal time delay. The running average of the oversampled readings can also yield a resolution higher than the raw resolution of the ADCs themselves.

This filtering is done on the position derived from the sinusoidal signals, not on the signals themselves, as is done on analog front-end filters. This means that the filter does not cause significant magnitude reduction and phase lag at high speeds.

The effect can be significant in the highest-precision systems. Tests run on the same physical system as in the above section show this clearly. The first test for noise mitigation is with the motor unpowered and the air bearing off, so the system is "locked" at rest by mechanical friction.

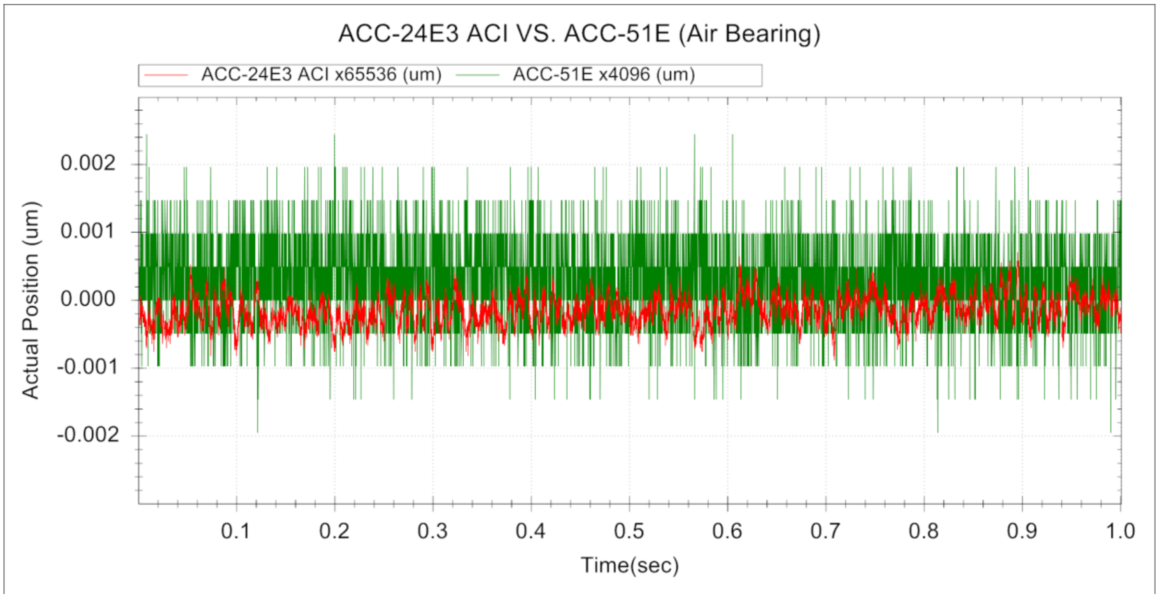


Unpowered Locked Stage Comparison of Interpolators

The red curve from the auto-correcting interpolator shows a reduction in noise magnitude of about an order of magnitude compared to the green curve of the older interpolator.

Both interpolators use 14-bit ADCs. Because the older interpolator cannot oversample to eliminate noise, it cannot attain a meaningful resolution of over 12 bits. The auto-correcting interpolator, with its high-frequency oversampling, can get a meaningful resolution of 16 bits from the averaged data.

The next test is also with the motor unpowered, but with the air bearing turned on so that mechanical friction is essentially eliminated, giving the stage more freedom to move.



Unpowered Floating Stage Comparison of Interpolators

This test also shows a dramatic reduction in the noise of the resulting measurement due to the oversampling and averaging of the new interpolator.

State Estimation

Further benefits are possible from this technique. If the servo algorithm receives only a position value each servo cycle, as in traditional systems, then its (implicit or explicit) calculation of velocity is done through digital differentiation at the servo rate. This has two significant problems. First, this differentiation introduces quantization noise. Second, the calculated velocity value has half a servo cycle time delay relative to the position value, adding destabilizing phase lag to the servo loop.

Increasingly, acceleration feedback is desired in high-performance systems, as it can provide some desirable effects such as creating an “electronic flywheel” that enhances stability and reduces ripple without the drawbacks of a physical flywheel. However, double differentiation of position feedback at the servo rate to obtain an acceleration value is even more problematic than the single differentiation for velocity, greatly magnifying the noise, and yielding a full servo-cycle delay. For this reason, this technique is virtually never used.

However, in the process of identifying and correcting the signal errors, the ACI first directly computes an estimated acceleration value, then numerically integrates this first into an estimated velocity value, and that into an estimated position value. These calculations are updated at the multi-MHz sampling rate.

Delta Tau’s auto-correcting interpolator makes it possible to use these “intermediate” values for velocity and even acceleration feedback in the Power PMAC’s servo loop. Using these quantities for feedback provides values with lower quantization noise and less time delay. Both of these advantages permit higher servo gains, yielding higher servo bandwidth, permitting better trajectory tracking and especially better disturbance rejection.

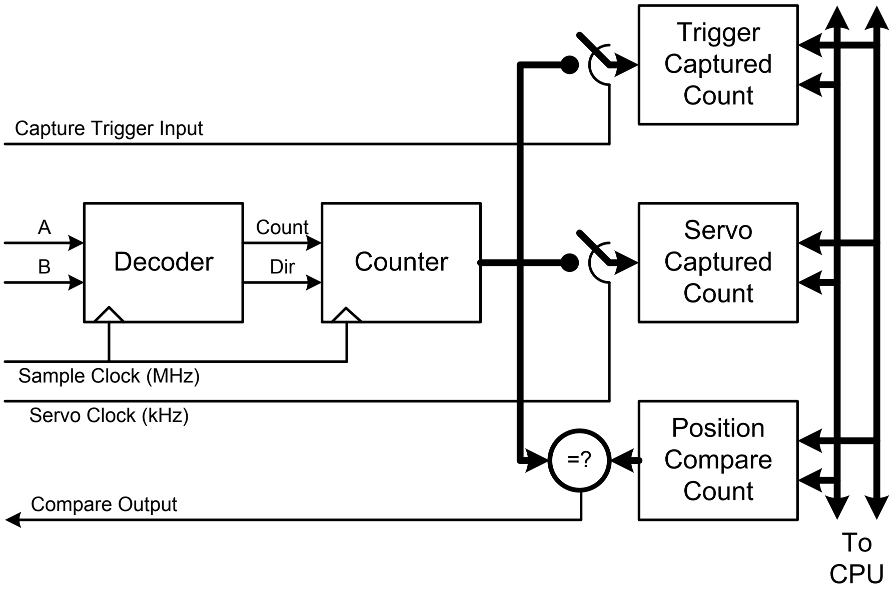
Asynchronous Position Capture and Compare

An important capability in many applications is the ability to perform very accurate position-capture and/or position-compare functions asynchronously to the servo-clock events. Position-capture capability requires the latching of the present position value at the moment of an input transition. Position-compare capability requires changing an output the moment the position reaches a pre-determined value.

In both of these cases, it is often not sufficient to use the position generated for the nearest servo cycle, as this does not produce the needed accuracy. Software tasks simply cannot execute at a high enough frequency to provide enough accuracy in high-precision applications, so dedicated hardware circuits are required.

With digital quadrature encoders, it is relatively easy to implement hardware capture and compare circuits with full-count resolution. The hardware counter is updated at the high (multi-MHz) hardware sampling frequencies. The same type of circuit that latches the count value at the instant of the servo clock interrupt can also be used for an asynchronous capture from an input signal. At this same high frequency, the count value can be compared to a preset user value, with an output toggled when the two values are equal.

The following figure shows the block diagram of this circuitry.



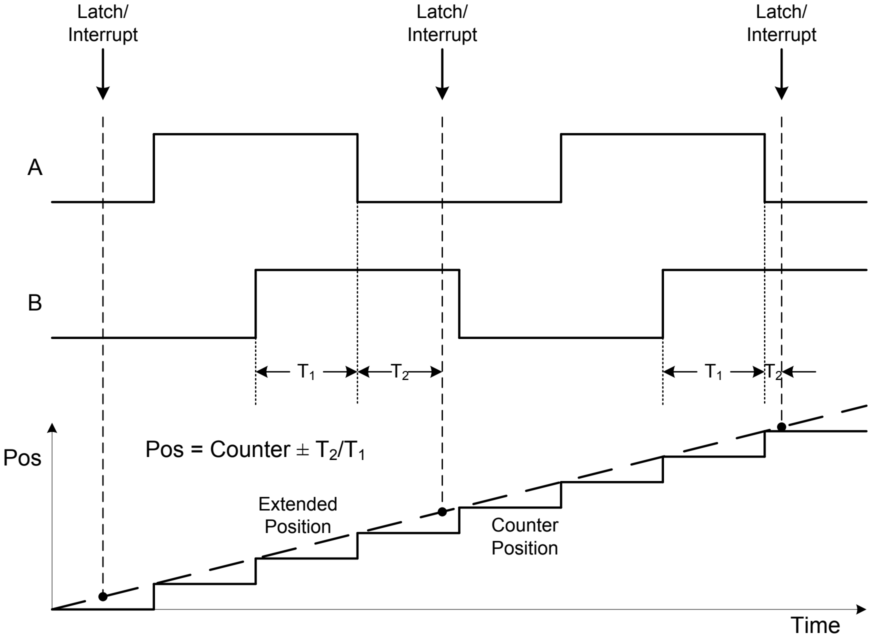
Hardware Capture and Compare Circuitry for Digital Quadrature Encoder

The key to the operation of these circuits is the fact that there is a position value – the hardware counter – that is updated at very high frequencies, even if it is not available to the software at these frequencies.

With the dual-sample-rate interpolators for sinusoidal encoders, the full-resolution position is only updated once per servo cycle, and the hardware counter value that produces four states per line does not have sufficient resolution for most applications. Different strategies are required in these applications.

Delta Tau provides sophisticated "1/T" timer-based sub-count interpolation techniques that yield a number of advantages. Originally introduced to reduce the quantization error from digital quadrature encoders in the servo feedback, particularly in the velocity loop, this technique uses high-frequency timers measuring the time between the last two count events and the time since the last count event. The ratio of these two timer events is an estimate of the fractional count value at an instant in time.

The following diagram shows the concept of this timer-based sub-count interpolation. Timer T_1 contains the time between the last two counts; timer T_2 contains the time since the last count. The ratio T_2/T_1 is a good estimate of the fractional count value. Depending on the direction of motion, it is added to or subtracted from the whole-count value.

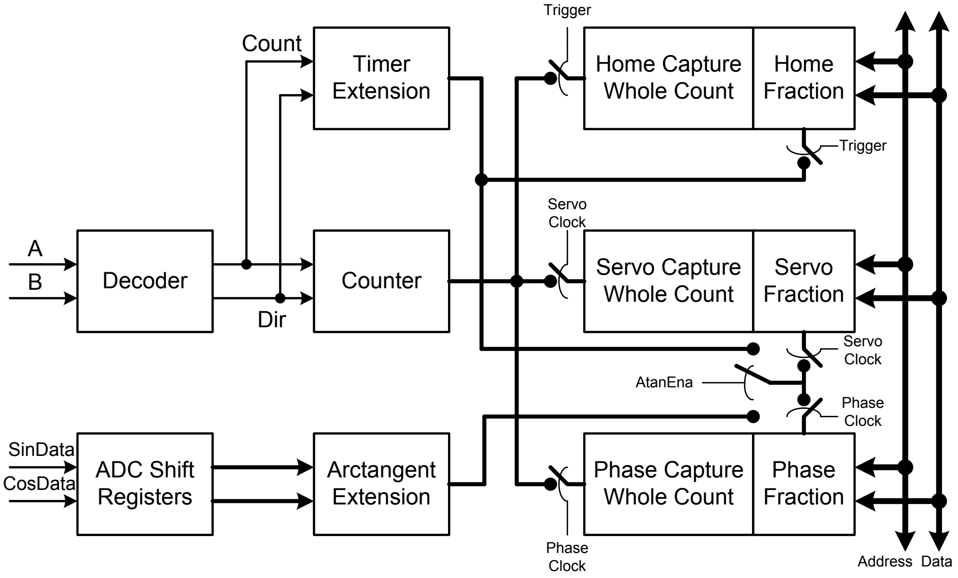


Timer-Based Sub-Count Interpolation

In its original implementation, these calculations were performed in software each servo cycle, improving the smoothness of servo operation, but not improving the resolution of asynchronous capture and compare positions. Subsequent implementations provided the capability to perform these calculations in hardware at the high hardware sampling frequencies, yielding enhanced resolution positions updated millions of times per second.

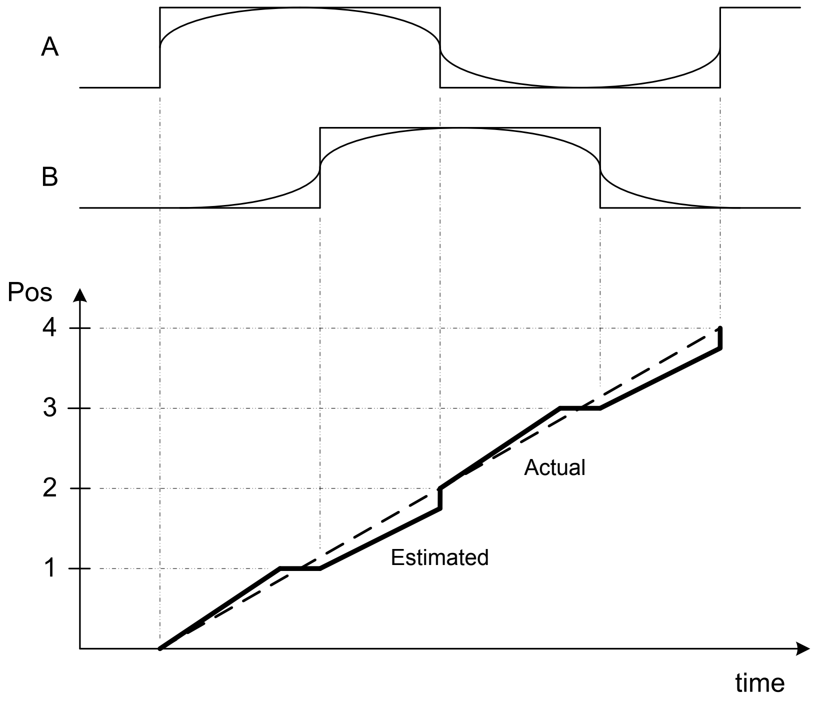
This timer-based sub-count interpolation in hardware has been very useful for both digital quadrature encoders and for sinusoidal encoders processed through a dual-sample-rate interpolator. In the second case, it uses the simulated quadrature generated by the high-frequency comparators.

The following diagram shows the synchronous servo (and phase) capture using either the timer-based interpolation or the arctangent interpolation and the asynchronous capture always using the timer-based interpolation. This circuitry is implemented in the logic of Delta Tau’s PMAC3-style “DSPGATE3” ASIC, used on the UMAC ACC-24E3 axis-interface board, the Power Brick controller/amplifier products, and the Power Clipper embedded controller.



Synchronous and Asynchronous Capture with Sub-Count Hardware Interpolation

For the timer-based sub-count interpolation to be accurate, the count events must be evenly spaced in position. All four of the common signal errors of sinusoidal encoders lead to uneven spacing of the "zero crossing" events that create counts from the generated digital quadrature. This in turn leads to errors in the sub-count estimation. The following diagram shows an example case of inaccurate interpolation from uneven whole-count spacing.



Sub-Count Interpolation Errors from Whole-Count Spacing Errors

These interpolation errors obviously lead to errors in capture and compare events. The discontinuities in interpolated position can be even worse, causing events to be missed outright.

While it is possible for identified signal errors to be compensated for in software at servo rates when using dual-sample-rate interpolators, this does not help for these asynchronous capture and compare events in hardware. However, because the auto-correcting interpolator compensates for these errors in hardware at high frequencies, it permits these capture and compare circuits to use corrected signals, virtually eliminating the resulting errors and discontinuities.

Comparison to Encoder-Based Interpolators

It is possible to purchase sinusoidal encoders with interpolation circuitry built into the encoder and some sort of digital interface to the controller. In evaluating such an encoder, it is important to understand how this interpolation works and how it can be integrated with the controller.

If the encoder outputs digital quadrature, it almost certainly has a tracking interpolator built in to the head. As far as the controller is concerned, the operation of this encoder interface is the same as an encoder with sinusoidal outputs and an intermediate tracking converter. An above section in this note explains the advantages and disadvantages of tracking interpolation.

More recently, this type of encoder provides some type of serial data interface, whether an open standard such as SSI, or a proprietary one such as EnDat. Encoders of this type are more likely to use some sort of direct interpolation. The quality, resolution, and sophistication of this interpolation vary widely from vendor to vendor.

The digital interface of such an encoder has the advantage of being more noise immune than the analog interface to a controller-based interpolator, but the time delay of the serial data transfer does introduce phase lag that can potentially compromise servo performance. In high-end systems, this effect must be evaluated carefully.

In addition, some encoders of this type have significant computational delays, sometimes of multiple sample cycles, in deriving the interpolated position even before transmission. It is important to understand if the encoder under consideration has delays of this kind.