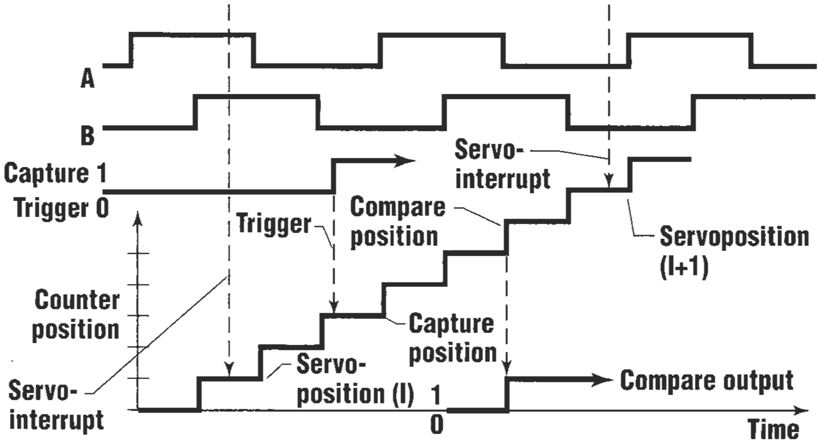
Synchronizing when speed counts

Special hardware can overcome software latency that may otherwise negate precise motion control.

Curtis S. Wilson
Vice President, Engineering and Research
Delta Tau Data Systems Inc.
Chatsworth, Calif.

Hardware capture and compare



Hardware capture-and-compare circuitry results in a more accurate position reading than that afforded by servointerrupts. Normal position capture records the last servointerrupt position, *i*. However, as shown, the true servoposition is two counts beyond the last capture point. Likewise, a compare position would not take place until the servointerrupt point that exceeds the compare value, shown as servoposition *i* + 1. That is almost two counts after the actual compare position of the servo.

Precision motion control is virtually never an end in itself. Rather, it ties position to other events or activities. As the need for application speed and precision rises, it becomes necessary to better synchronize motion to external events.

One common goal is to coordinate digital I/O as tightly as possible to physical tool position. In many cases the quest for maximum productivity dictates machine coordination take place "on the fly." Accuracy and repeatability are the key issues in this coordination that's based on the speed at which the I/O operates.

For example, say an encoder monitors the position of a moving slide, transmitting the slide's current location to the controller. The controller, however, needs a finite period to input and react to the position information. During this period the slide continues moving so it is no longer at the position the encoder indicates when the controller takes action. The error between the indicated and true position of the slide hinges on the slide velocity multiplied by the worst-case period of the I/O task.

Edited by Robert Repas

Either software or hardware can manage the synchronization of a task to a position. Software algorithms are, of course, far more flexible. But they operate much more slowly than a hardware trigger. Even interrupt-driven routines have large latencies compared to hardware schemes. Typically, interrupt-driven software tasks cannot access the instantaneous position at the moment of the interrupt. That still requires specialized hardware.

There are times when an output must take place at a specific position. The algorithm or circuit that performs this action is called a compare function. It compares the present position to the desired position and triggers the output when the two match. Compare circuits are particularly valuable for triggering measurement devices or firing lasers when the moving member reaches a specific point.

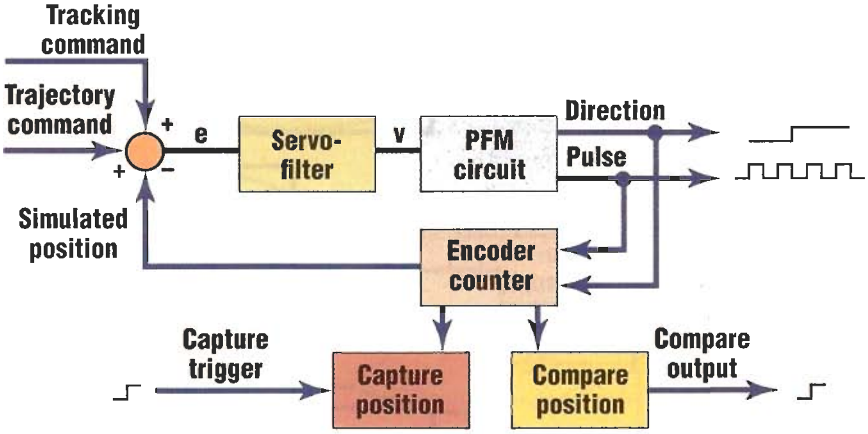
At other times, it might be necessary to note the position upon detection of an input signal. The algorithm or circuit that performs this action executes what's called a capture function because it captures the present position value on transition. Capture circuits are useful for homing-search moves in virtually all positioning applications, recording the location of probes like those found on coordinate-measurement machines, and for registration sensing in applications such as printing and cutting.

Controllers appeared around 1990 with dedicated hardware capture-and-compare circuits for digital quadrature encoders. These were usually implemented in application-specific integrated circuits (ASICs). The circuits exploited the fact that the encoder counter operated at several megahertz while the servocontroller cycled in the kilohertz range. Placing the circuits in the same ASIC that processed encoder data for feedback added virtually no extra cost to the system. The capture-and-compare circuits directly read the rapidly updating counter within the ASIC without software intervention.

A compare process starts when the controller writes the desired position value to the comparison register within the ASIC. When the encoder counter value matches the prewritten compare-position value, the capture-and-compare circuit toggles a digital output. The only delays in the circuit generating the output are a couple of clock cycles necessary to drive the decoding and counting circuitry.

For a capture process, a digital input latches the instantaneous value of the counter into a controller-readable register. Once latched, the capture circuit sets a special processor flag to signal the controller. Flags are digital inputs, outputs, or memory bits that indicate a process or action is complete and that the system is ready for the next step. The accuracy of the capture did not depend on either how long it took the processor to react to the trigger signal or values captured at an earlier time in the servo cycle.

Simulated encoder with capture and compare



A simulated or virtual encoder can produce similar results to a true encoder. The output of the virtual encoder feeds the hardware capture-and-compare circuitry as if the encoder actually exists.

The introduction of hardware capture-and-compare circuits galvanized the performance of many applications. Motion speed no longer limited the accuracy of captured positions or compare outputs. Machine productivity multiplied severalfold. But this initial success merely whetted the appetites of many users for still greater speed, flexibility, and accuracy.

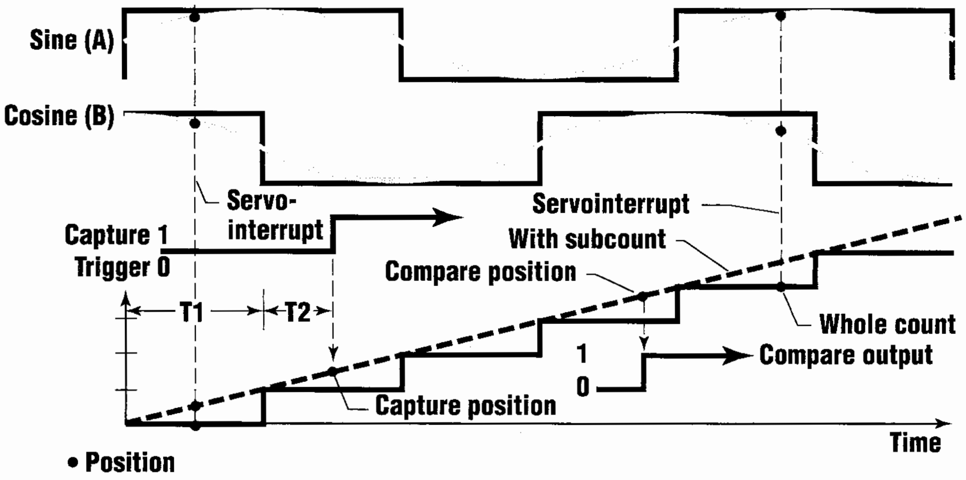
Early compare circuits needed separate software operations to write each compare position to the hardware. Although this did not affect the accuracy of any single output pulse, it could severely limit the frequency for generating multiple outputs — typically to only a few kilohertz. A hardware autoincrement circuit overcame this limitation.

Here, when the encoder value matches the compare position, the autoincrement hardware adds a fixed amount to the compare position. The processor writes only the starting compare position and the autoincrement value to the hardware registers. The hardware circuit then operates without any further software intervention to generate output pulses at frequencies into the megahertz range.

Logically combining multiple outputs from a single encoder counter with autoincrement opens an entirely new range of functions and capabilities. One important capability introduced by this arrangement was the "windowing" of an output pulse train to a particular area.

For example, an output pulse that triggers a marking laser must repeat every 10 counts between 4,000 and 5,000 counts. Two compare circuits are needed to generate this periodic output. The first establishes the window through which the pulses appear. It's set to turn on at 4,000 and off at 5,000 counts. The other compare circuit generates the actual pulses using the autoincrement feature. It is initially set to turn on at 3,900 counts and turn off at 3,905 counts with an autoincrement value of 10. Every time the second compare output cycles, autoincrement resets the on and off compare values 10 counts higher. So the second output turns on at 3,900, off at 3,905, on again at 3,910, then off at 3,915, back on at 3,920, and so forth. The two output signals combined through a logical AND gate function create the desired pattern of trigger pulses. The processor resets both compare circuits any-

Subcount capture and compare



A subcount capture-and-compare system uses the encoder sampling clock to determine an overall time for each whole step count, T_1. The fractional count, T_2, creates a percentage of the whole step to boost resolution. The fractional step position adds to or subtracts from the nearest whole step as needed for the more accurate position.

time after the counter exceeds 5,000 counts.

It is possible to use capture-and-compare circuits even when there is no physical encoder to represent the position along a path. For example, a stepper-motor-controlled *XY*-axis system may need a pulsed output that represents its movement along the vector path of the *XY* motion. To handle this application a computer-generated virtual axis is programmed to move in coordination with the true physical axes. If the *X* axis moves three units and the *Y* axis moves four units, the virtual axis "moves" five units and generates a simulated encoder output that represents the five units of motion.

The circuit generates output pulses as if they came from an actual physical encóder monitoring the vector path. The pulses created by the virtual encoder go to the capture-and-compare circuit as though the virtual axis were real.

Recent years have seen a significant trend toward the use of sinusoidal encoders in precision applications. For higher resolution and accuracy, adding more measured states to a single encoder line is usually more economical than boosting the number of lines per unit of movement. This has led to a variety of interpolation circuits for sinusoidal encoders.

One interpolation technique uses a multiplying tracking loop that resembles a phase-locked loop. The tracking loop outputs a digital quadrature waveform at higher frequencies than the analog sine waves that formed it. The digital signal feeds the capture-and-compare hardware as if a higher resolution encoder generated it.

There are several drawbacks to this technique. First, there is a limit to the frequency-multiplication factor and thus to the final resolution. Second, the design may require the axis speed be limited so the quadrature frequency does not overrun the receiving circuitry. Finally, the tracking loop has dynamics of its own that makes the digital count lag the true position. That affects both the stability and accuracy of the loop.

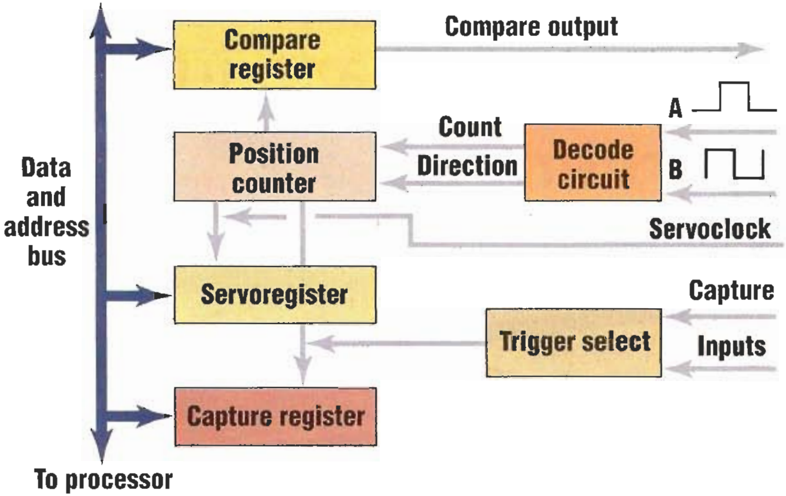
For these reasons, most high-resolution interpolators of sinusoidal encoders provide direct conversion rather than tracking conversion. Direct conversion requires two receiving circuits. Comparators in the first receiver convert the analog sine and cosine inputs to digital quadrature waveforms that match the input frequency. The waveforms then go to standard digital quadrature circuits.

The second receiver uses analog-to-digital converters to convert the sine and cosine signals into numerical values during each servo cycle. An arctangent calculation yields the fractional location within a single count of the encoder. This fractional position is combined with the count from the quadrature circuit to provide the full-resolution net position.

This approach works well for servo feedback by providing high resolution and high frequencies. It's quite common to see 4,096 div/line count while 16,384 divisions are starting to appear. Line rates of several megahertz are possible with effective count rates reaching the gigahertz range. However, the system only calculates the full-resolution position once every servocycle. Only the four counts/line of the quadrature signal is usable with traditional capture-and-compare hardware circuits during high-speed motions.

Attempts to solve this problem create mixed results. One of the best, in terms of both cost and performance, improves an old resolution-enhancement tech-

ASIC position capture and compare block diagram



Application- specific ICs have provided a low-cost method of creating hardware capture-and-compare circuits for position control. For basic position monitoring, quadrature encoder pulses directly control the up/down count of a position counter. A servoregister records the current position with each tick of the servoclock. To perform a compare function, the processor writes the desired position into the compare register. When the position counter matches the compare register, the compare output triggers. For the capture function, an external trigger transfers the current value in the servoregister to a capture register, which the processor then reads. Both capture and compare functions take place independent of the processor, so they are not affected by the processing time of the software.

nique. For many years, various controllers employed high-frequency timers with their quadrature-count circuits to enhance measurements. These are commonly called $1/T$ methods because velocity is inversely proportional to the time between counts. Typically these timers were driven by the same clock signal used to sample the incoming encoder signal, at a minimum frequency of several megahertz.

A more elaborate $1/T$ technique employed two timers. The first measures the time between the last two counts, while the second measures the time between the receipt of the last count and the present. The ratio of the second timer value to the first provides a good estimate of the fractional count value. The fractional count value either adds to or subtracts from the counter value, depending on the direction of motion.

Originally, the present time was the servo interrupt. The counter and two timer values were latched and the processor calculated the ratio between the two timer values. The servoloop used the combined whole-count and fractional-count value to generate the actual position value with enhanced resolution. But this software-based approach did nothing to help improve capture or compare resolution. The enhanced-resolution position was only calculated once per servocycle.

Transferring the fractional-count calculation to hardware and updating it every encoder-sample clock cycle keeps the estimate always available to the capture-and-compare circuit for the encoder. An input trigger latches the fractional value calculated within the last sample cycle along with the latest counter value. The compare output for the channel toggles when the counter matches a preset whole-count value and the latest fractional value matches or passes a preset fractional compare value.

This technique economically boosts resolution for the capture-and-compare positions. It simply adds more logic gates to the ASIC already present to process the encoder inputs. Many believe that analog sinusoidal encoders need additional analog circuitry to use these functions properly. However, close analysis indicates this purely digital approach is not only less expensive, it meets or exceeds the precision of a more expensive analog approach.

Use of hardware to enhance timer-based resolution matches the resolution and accuracy of the capture-and-compare for sinusoidal or quadrature encoders to that provided by the ADCs for servoloop-feedback positions. In many cases the fundamental limitation for capture-and-compare resolution comes from the encoder-sampling clock period — usually measured in tens of nanoseconds. This improvement produces the greater accuracy needed in the many high-precision processes used for semiconductors, optical and magnetic disks, and flat-panel displays. **MD**

MAKE CONTACT

Delta Tau Systems Inc., (818) 998-2095, *deltatau.com* Circle 624