EXTRACTION OF HIGH RESOLUTION POSITION INFORMATION FROM SINUSOIDAL ENCODERS

J. Burke, J. F. Moynihan, K. Unterkofler

Abstract - Precision motion control in industrial automation systems is placing increasing demands on the position feedback mechanisms used in the constituent servo drives of these systems. Traditional position transducer solutions based on resolvers or incremental encoders are increasingly being replaced by sinusoidal encoders that offer much higher position and speed resolution. The hardware or software techniques used to interface to the sinusoidal encoders can have a measurable impact on the achieved position resolution and accuracy. This paper compares the achievable performance levels with different interface techniques. All techniques are implemented on a common hardware platform. Each method is fully described and comparisons and trade-offs are made in terms of computational burden, achievable position bandwidth, the resolution of both the speed and position achievable as well as the sensitivity of each method to common transducer offsets, gain, and phase errors.

Fundamental Operation of Sinusoidal Encoders

Sinusoidal encoders encode position information by providing a pair of quadrature sine and cosine signals as the shaft is rotated. The signals may be generated by optical or magnetic means and typically produce 512 or 1024 cycles per mechanical revolution. For noise immunity the signals are typically transmitted differentially from the encoder to the sensor interface electronics. A typical interface configuration is shown in Figure 1.

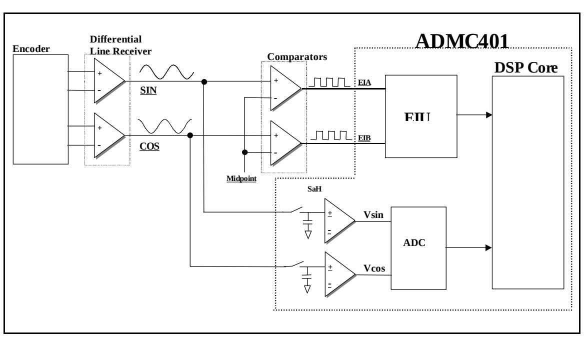

Figure 1: Typical Sinusoidal Encoder Interface

In order to extract reliable position and speed information from the sinusoidal encoder signals, a certain amount of pre-conditioning of the analog signals must be implemented. As a first stage, the differential SIN and COS signals (typically $1V_{pp}$ input signal range) from the sinusoidal encoder must be applied to input differential amplifiers. This ensures maximum noise immunity and may also be used to appropriately amplify and level shift the resultant single-ended SIN and COS signals for later input to the Analog to Digital Converter (ADC) stage. Next, the SIN and COS signals of Figure 1 are applied to comparators that generate square-wave TTL-level signals (EIA & EIB) that are synchronized to the SIN and COS signals respectively. The relationship between the SIN, COS, EIA and EIB signals is illustrated in Figure 2.

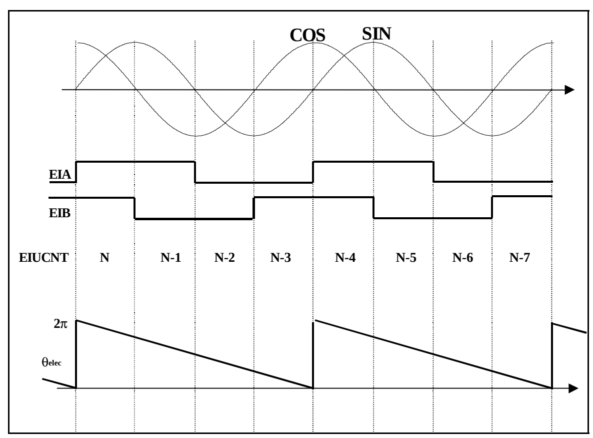

Figure 2: Signals Produced by the Sinusoidal Interface

The analog SIN and COS signals are also fed to dual sample and hold amplifiers (SHA) for subsequent conversion to digital and post-processing in the digital signal processor (DSP). The signals EIA and EIB are typically applied to the internal quadrature counter of a dedicated digital Encoder Interface Unit (EIU) as illustrated in Figure 1. The resultant parallel word in the quadrature counter provides the crude position estimate from the sinusoidal encoder. There

are four counts per cycle of the sinusoidal waveforms (SIN and COS) as seen in Figure 2, so that for a 512 line encoder the maximum count value is 2047 (4*512-1) which provides 11 bits of crude position information. In the case of a 1024 line encoder the maximum count 4095 which provides 12 bits of resolution. In the case where the EIB signal leads the EIA signal, the encoder is determined to be rotating in the reverse direction and the quadrature count value (EIUCNT) is decremented at each edge of the EIA and EIB signals, as seen in Figure 2. Naturally, when EIA leads EIB, the quadrature count value is incremented at each event.

Fine position resolution is obtained by further processing the digitized SIN and COS signals to provide much finer granularity between the EIA and EIB events, as illustrated by the lower waveform(θ_{elec}) of Figure 2. The particular techniques used to extract this fine position resolution are the main subject of this paper.

In some applications, it is necessary to know the initial position of the rotor following power up. There are different techniques used to obtain this information depending on the particular encoder design. Some encoder designs provide an alternative pair of sine and cosine signals that provide one cycle per mechanical revolution, from which it is possible to derive an initial position estimate. Alternatively, some modern sinusoidal encoders now provide a dedicated serial interface that can be used to extract the initial position following power up [1]. The particular method used to extract the initial position and the alignment of this value to the computed position value during normal operation is beyond the scope of this paper.

During normal operation, the complete position information must be constructed from both the crude and fine position information. The crude information contains the cycle identification information, which, depending on the encoder used, can be 11 or 12 bits. The fine information is the result of the calculation from the sinusoidal signals. With high resolution analogue to digital conversion of 12 bits on the SIN and COS, position information to 23/24 bit can be achieved. Since the crude position information is a quadrature value the 2 least significant bits (LSB) of the first segment should be the same as the 2 most significant bits (MSB) of the fine data and so are not needed and so only the 9 MSBs, in the case of a 512 line encoder are used. In the case of a 1024 line encoder the first 10 bits are used. The construction of the final position data is illustrated in Figure 3 for a 512-line encoder.

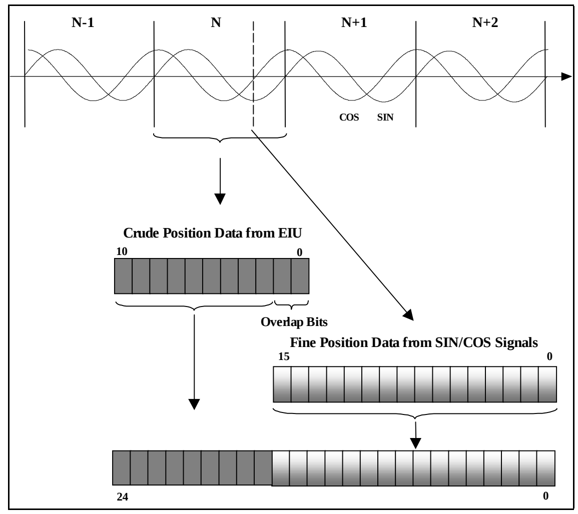

Figure 3: Construction of Total Position Information from Sinusoidal Encoder.

DSP Requirements

The minimum requirements for a hardware platform to implement the various methods below are:

1) A fast, high performance DSP core.
2) Simultaneous sampling multi-channel Analog to Digital Converter.
3) Encoder Interface Unit.
4) Serial Communication Port.

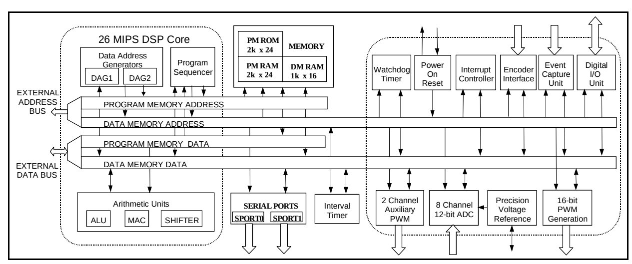

Figure 4: Block diagram for the ADMC401

The DSP chosen to implement both algorithms is the ADMC401 from Analog Devices, which includes a 26 MIPS, 16-bit fixed point DSP core[2]. The DSP provides a combination of efficient processing units such as hardware multiply and accumulator (MAC) and arithmetic logic unit (ALU) and shifter. The DSP core uses a modified Harvard architecture so that program and data memory have separate address and data buses permitting highly efficient and parallel memory accesses. A block diagram of the ADMC401 is shown in Figure 4.

The core of the ADC system is a single high-performance 12-bit analog to digital converter that uses a pipeline flash conversion technique to achieve very fast conversions. The ADC system provides 8 analog inputs arranged in two banks of four channels, each of which is provided with a dedicated sample and hold amplifier. This structure permits four pairs of input signals to be sampled simultaneously. The conversion time for all 8 analog inputs is under 2µs at a DSP clock rate of 26 MHz and conversion may be initiated for either an internal source synchronized to the PWM generation or from an external source.

The core of the encoder interface unit is a 16-bit quadrature up down counter that converts the A and B quadrature signals from the encoder into a parallel word. This word can be read by the DSP as part of the motor control algorithm for motor current commutation, speed estimation and/or position control. The encoder interface unit of the ADMC401 also provides a programmable input filter stage that permits encoder pulses less than some minimum value to be filtered and rejected. With this feature, spurious noise pulses do not adversely affect the operation of the quadrature counter. The encoder interface unit permits input A and B quadrature signals with rates of up to 4.3 MHz corresponding to a maximum quadrature rate of 17.2 MHz.

The ADMC401 provides two independent serial communication ports that can be used to interface with sophisticated sinusoidal encoders such as the Stegmann Hiperface[1]. The additional serial port can be used to transmit the computed position information to another processor or host. Alternatively the ADMC401 may address shared external memory so that data can be accessed by another processor or host.

Methods for Extracting Fine Position Information

Method 1: Direct Arctangent Computation

In the first method the fine position data is computed directly from the digitized SIN and COS signals. The DSP implements the calculation as:

$$\theta_{fine} = \tan^{-1}\left(\frac{V_{\sin}}{V_{\cos}}\right) \tag{1}$$

where V_{sin} and V_{cos} are the digital representations of the simultaneously sampled SIN and COS waveforms from the encoder at the particular sampling instant. The arctangent is approximated in the DSP calculation by a fifth order Taylor series expansion below:

$$\tan^{-1}(\alpha) = 0.318253\alpha + 0.003314\alpha^2 - 0.130908\alpha^3 + 0.068542\alpha^4 - 0.009159\alpha^5 \tag{2}$$

Using a Matlab simulation, values for V_{sin} and V_{cos} (quantized at the 12-bit level) were generated and transmitted to the DSP. These values were then used as the source to compute the divide and arctangent functions of (1), using standard, 16-bit, fixed-point arithmetic. The result of the computation was then fed back to Matlab. The difference between the high precision Matlab generated position and the estimate computed by the ADMC401 using (1) is plotted over 1 electrical cycle in Figure 5.

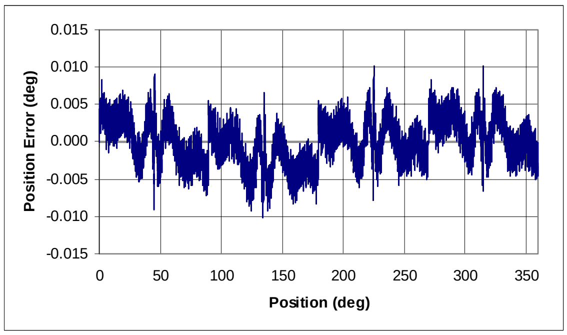

Figure 5: Quantization Error in Position over Electrical Cycle.

As seen from Figure 5, the resultant error in the computation is always less than or equal to 0.01 degrees per electrical revolution. In addition, for an encoder that offers 512 electrical cycles per mechanical revolution, the 9 MSBs of position information come from the cycle information as $2^9=512$, the number of lines in the encoder. This means that the error occurring in Figure 5 due to the finite quantization levels and calculation inaccuracies of the DSP is at the 15th bit position of the computed 16-bit fine position value. This implies that the fundamental limit on the 16-bit fixed point DSP calculation used in this method is located at the 24th bit in position (for a 512-line encoder). In other words, the maximum achievable resolution from such a solution is given by $2^{-24} * 360 * 60 * 60 = 0.077$ arc-sec .

This analysis does not take into account position accuracy errors due to the expected imperfections in the physical system due to offset or gain errors in the sine and cosine channels from the encoder or in the analog signal processing circuits prior to the ADC. Additional errors are also introduced by phase-imbalance in the encoder such that the sine and cosine waveforms are no longer in perfect quadrature.

The advantage of the arctangent method is that it allows position information to be requested at any time, as long as the time for consecutive requests is greater than the time taken to run the necessary DSP calculations. This means that if requests are time stamped, which is possible with the ADMC401, the speed of the motor can than be accurately calculated.

Using a test rig, a Sinusoidal Encoder was mounted on a very high-resolution turntable. The table was rotated slowly such that the position change was 1 electrical degree (of the sine and cosine waveforms) in 100µs. This corresponds to a shaft speed of approximately 3.25rpm for a 512-line encoder. The position was read from both the ADMC401 and the table counter and the difference is plotted in Figure 6.

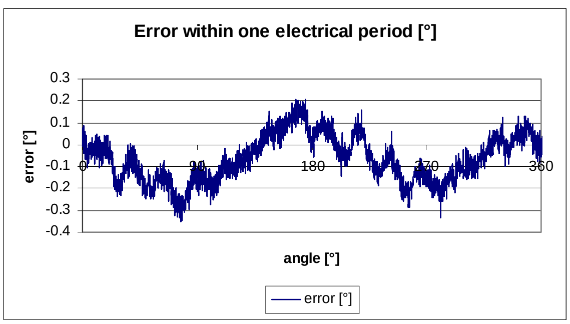

Figure 6: A graph of the error over one electrical revolution.

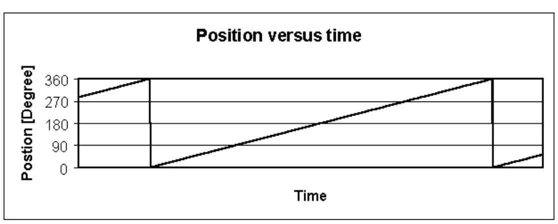

Figure 7: Plot of position for an encoder rotated at constant speed.

Figure 7 shows the derived rotor position information from a 512-line encoder over 1 complete mechanical revolution. The complete position data is the combination of the crude position information (11-bits) from the EIU and the fine position information derived using (1). Only the 16 MSB's of the resultant position value are plotted for clarity.

Presently, the execution time for this arctangent method is 6.57µs using the ADMC401 DSP. At a motor speed of 6000rpm, a 512-line sinusoidal encoder will produce sinusoidal outputs at 51.2kHz. At this speed the encoder signals have a period of 19.5µs which allows for just 3 samples per period. The length of time between consecutive samples compared to the period of one electrical cycle would possibly make feedback digital filtering of this position information less accurate because aliasing could occur. It is likely that motor noise could be significant, and this could present a problem with using this method. A possible solution is to shield the motor, which could reduce the noise. Alternatively, finding software solutions that could incorporate digital filtering as an inherent part of the process of extracting position information is one reason for considering another method.

Method 2 – Tracking and Filtering Loop

The second method of extracting precise position information from the sinusoidal encoder is similar to that used in existing hardware solutions for extracting position from resolvers using standard Resolver to Digital Converters (RDC) [3]. The solution here involves forming a closed loop tracking system that forces the estimate of the rotor position (φ) to converge to the actual position (θ). The error is formed by a heterodyning, cross-multiplication process to form the signal:

$$\begin{aligned}\Delta E &= \sin(\theta)\cos(\varphi) - \cos(\theta)\sin(\varphi) \\ &= \sin(\theta - \varphi) \\ &\approx \theta - \varphi, \quad \textit{for small } \theta - \varphi\end{aligned} \tag{3}$$

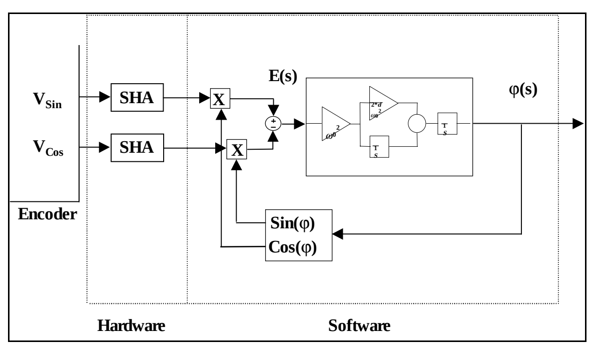

Figure 8 : Block diagram of the Tracking and Filtering Loop

This process is illustrated graphically in Figure 8. As with any closed loop system the aim is to force the error to zero. The benefit of this method is that digital filtering, which consists of a PI controller in series with an integrator, is built into the feedback loop. This method tracks the position and periodically updates a position counter.

In the continuous domain, the forward path of the system may be expressed as:

$$G(s) = \frac{\varphi(s)}{E(s)} = \left[\omega_0^{\,2}\right] * \left[\frac{1}{s} + \frac{2*d}{\omega_0}\right] * \left[\frac{1}{s}\right] \quad (4)$$

Using the Zero Order Hold (ZOH) approximation to translate the transfer function (4) to the discrete domain, G(z) becomes:

$$G(z) \approx \left[\frac{\omega_0^{\,2} * T_S^{\,2}}{2}\right] * \left[\frac{(z+1)}{(z-1)^2}\right] + \left[\frac{(2*d*\omega_0 * T_S)}{(Z-1)}\right] \quad (5)$$

As can be seen in equation (5) the three main parameters are ω_0, d and T_s, where ω_0 is the natural frequency of the filter, d the Damping Ratio and T_s is the Sample Time. Due to the ZOH approximation, equation (4) becomes:

$$G(z) = \frac{\varphi(z)}{Z^{-1}E(z)} \quad (6)$$

Combining equations (5) and (6) results in:

$$\varphi_{K+1} = A*E_K + B*E_{K-1} + 2*\varphi_K - \varphi_{K-1} \quad (7)$$

where

$$A = \frac{\omega_0^{\,2} * T^2}{2} + 2*d*\omega_0 * T_S$$

$$B = \frac{\omega_0^{\,2} * T_S^{\,2}}{2} - 2*d*\omega_0 * T_S \quad (8)$$

To improve the resolution and speed of convergence the algorithm was implemented in 32-bit arithmetic. These 32 bits of fine position information combined with the 9 bits of cycles information from a 512 line encoder results in 41 bits of position information over one complete mechanical cycle of which only the 32 MSB's are stored for use by subsequent algorithms.

The following is a summary of some of the results of the tests using the ADMC401 to implement the closed loop tracking of (7) and the cross multiplication of (3). Figure 9 shows the transient response in tracking a step change of 45 electrical degrees. As in the previous method Matlab was used to generate two values V_{Sin} and V_{Cos} which were used as a starting point for the algorithm. The result of the calculation was then sent back to Matlab. [Step] is the step change, [DSP Pos.] is the DSP estimated position.

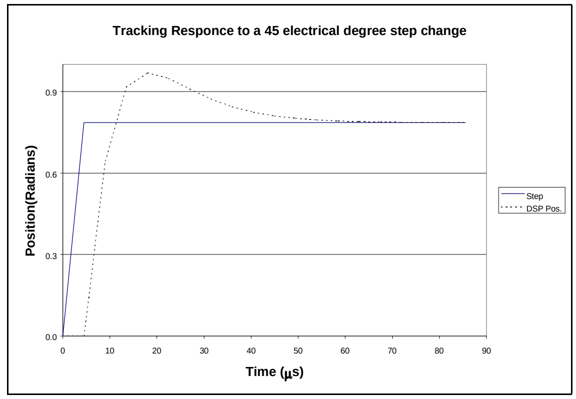

Figure 9: Plot of DSP and Simulation position against a step change.

The position error is also plotted in Figure 10, so that the tacking error can be seen more clearly.

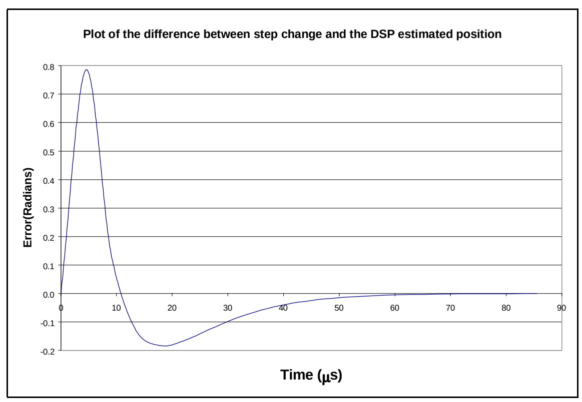

Figure 10: Plot of the tracking error for a 45° step change.

Figure 11 illustrates the transient response of the position error, E, when the speed of rotation is 1000 rpm. In this case the rotor position is ramping linearly and Figure 11 shows the initial error in the rotor position estimate due to the finite tracking performance of the proposed method. It can also be seen that the tracking loop reduces the steady state estimation error to zero in about 70µs.

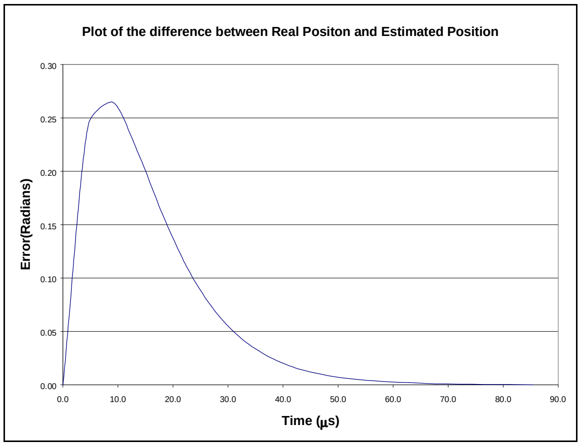

Figure 11: Error in tracking a ramp at 1000 rpm.

The maximum number of DSP cycles executed in implementing this tracking is 111. This is equivalent to 4.269µs and so T_s was chosen to be 4.5µs. Using the sampling theorem as a starting point, ω_{0Max} is approximately 700krad/s. For a 512-line encoder, this gives a theoretical limit of the mechanical speed of approximately 12000rpm. For the test showed above, the bandwidth ω_0 of the filter was set to 100krad/s. The Damping Ratio (d) was set to 0.9. With these values, there is no need of scaling of the coefficients of equation (8).

This method can produce very high position resolutions once the motor speed is kept inside the bandwidth of the filter. As with all such closed-loop tracking systems, it is possible to trade-off the mutually exclusive characteristics of resolution and position bandwidth by adjusting the filter coefficients in the feedback loop. Using the above parameters the motor could be tracked to a speed of 11000rpm, which is in accordance with the theoretical limit stated above.

Unlike the arctangent method, this method requires a settling time, during which the position estimation may not be exact and incur some error as in Figure 11. However the obvious benefit for a second order tracking loop is zero steady state error (at least to within the quantization noise levels in the DSP implementation).

Sensitivity to System Offsets and Signal Gains

Test were also carried out in regards to the effects of system offsets and mismatched gains on the final result. Due to the cross multiplication approach the effect of mismatched gains and offsets is greatly reduced. The final algorithm will incorporate amplitude scaling and offset compensation to fully eliminate these effects.

Conclusions

At a speed of 120rpm, a 512-line encoder will have swept out an angle of 1.65888 electrical degrees in the 4.5µs computation time, making the 32 bits of position information relatively useless for position estimation for an outside controller. Thus, it makes sense to say that such high resolution is really only required at low speeds, and for the most part the encoder will be at rest in order to determine exact position information. Given this, both methods have their advantages. The tracking loop requires less DSP cycles and results in higher resolution than the arctangent method. However the arctangent method has zero settling time. The greatest considerations in choosing a method would be the maximum speed the encoder is allowed to travel at and the required resolution of the position estimation.

References:

[1] SRS50, Motor Feedback System for Servomotors SinCos SRS 50 with HIPERFACE by STEGMANN.

[2] ADMC401, Single-Chip, DSP-Based, High Performance, Motor Controller, Analog Devices Inc., 1999.

[3] AD2S81A/AD2S82A, Variable Resolution Monolithic Resolver to Digital Converter, Analog Devices Inc., 1998.

Acknowledgements:

The authors of this paper would like to thank Ulrich Armbruster, Andrew Monnin and Bernd Appel of Stegmann Inc. for all their input.