7th INTERNATIONAL MULTIDISCIPLINARY CONFERENCE
Baia Mare, Romania, May 17-18, 2007
ISSN-1224-3264

IMPLEMENTATION OF A HIGH RESOLUTION VELOCITY SENSOR USING A QUADRATURE SHAFT ENCODER FOR ACTUATORS

BUCHMAN Attila, NASUI Vasile, UNGUREANU Miorita, ALEXANDRESCU Marius
North University of Baia Mare, Str.: V. Babes No. 62A, Ro-430083, E-mail: inginerie@ubm.ro,

Abstract. *This paper presents a straightforward way to decode a quadrature shaft encoder output in order to obtain an accurate estimate of velocity linear actuators. The resolution of the implemented velocity sensor wil be theoretically estimated and experimentally measure. A digital signal processing method to overcome the inherent error of the method at low speed will be discussed.*
Keywords: *actuator, shaft encoder, quadrature, resolution, full scale range, slew rate.*

1. MOTION CONTROL CONSIDERATIONS

The actuator have a complex structure, the mechanical part being composed of certain elements which ensure a high kinematic and dynamic precision, and the commanding part being represented by a computer-led system, based on appropriate software. The translation modular system is used to substitute hidraulics and pneumatics cylinders. This system can be used to advance mechanisms, elevate and manipulate devices, rotation devices, etc**.** The linear actuator is composed of a motor turning a screw in which the nut on screw is not allowed to rotate. This allows linear motion o the nut for the length of the screw. (fig. 1) A position at some point along the screw is commanded by the user and the motor turns the screw until the nut reaches that position. The electro mechanic liniar actuators, which are mecahatronic products and they are part of the intelligent flexible systems of manufacture and they have to fulfill special special requirements which impose new concepts of design (fig. 1).

The motion control system's purpose is to control any one, or combination, of the following parameters: position, velocity, acceleration, torque. Many motion control systems are integrated into a larger system. Various computer-based devices, such as programmable controllers, stand-alone industrial computers, or mainframe computers serve to link and coordinate the motion control function with other functions.

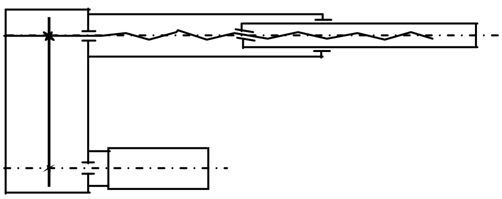

Fig. 1. Actuator linear with parallel acting

Feedback device for position actuator system are often integrally with the motor, but sometimes they are mounted on another system component such as an output gear shaft or a leadscrew. Many varietes of feedback decives have been designed: optical encoder, resolver and additional feedback devices. Optical encoder are two types of optical encoder: incremental and absolute. They combine high accuracy with low cost (incremental prototype), and are, therefore, in widest use for position feedback. They are available for integral mounting with the motor, or for mounting on another shaft. Optical encoders are also used in linear configuration for machine beds. Among their disadvantages, they are less robust than resolvers or magnetic encoders. If used to commutate a brushless DC motor, an optical encoder must also be enhanced with supplementary commutation tracks. The recent introduction of higher resolution magnetic encoders make this technology a viable feedback alternative.

The general form of the block diagram of a feedback control system is shown in the figure 2 depicting the basic elements of the two most common electric motion control systems: stepping motor-based and servo motor–based .

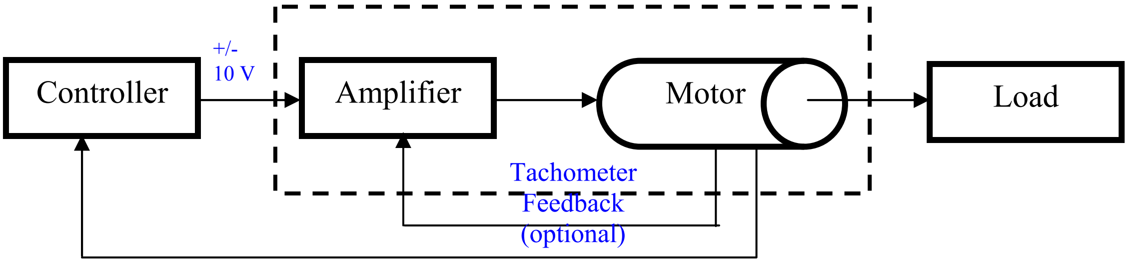

Fig. 2. Controller System Block Diagram

The scheme of a programmable servomechanism with speed and positioning reaction the most often used within the tracing systems presented in figure no. 3. Shaft encoders are widely used digital position sensors. Robotics, computer pointing devices, rotating radar platforms uses this kind of sensors.

Figure 4 presents the operating principle of an optical quadrature shaft encoder, used for linear position sensing, as in computer mice. Figura 4 illustration of the x quadrature shaft encoder operating principle Estimating velocity from a shaft encoder is a cost effective strategy in motor control. With the arrangement presented in figure 4 linear velocity along x axis can be estimated.

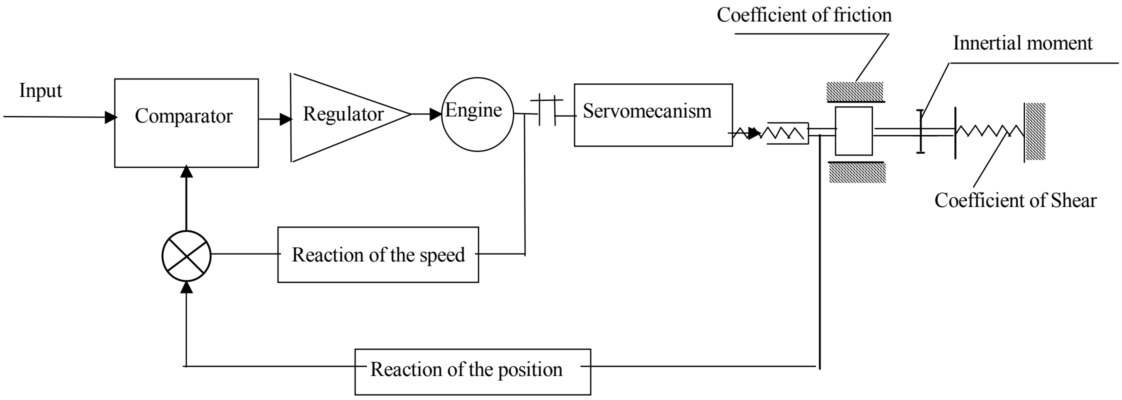

Fig. 3. The dynamic model of programmable servomechanism

2. VELOCITY ESTIMATION WITH POSITION SENSORS

Velocity estimation implies the addition of a time base. Let *T* be the period of that clock, which is the fixed unit time. Let *X* be a fixed unit position (i.e. the displacement corresponding to two consecutive slots on the disc). Two first order estimates for velocity may be written [1]: Equation 1 states, that measuring displacement, at equally spaced time intervals, gives an estimate of the velocity. This, according to [1], is the conventional approach to velocity estimation.

$$v(kT) = \frac{\Delta x}{T} \qquad (1)$$

$$v(kT) = \frac{X}{\Delta t} \qquad (2)$$

where:

$$\Delta x = x(kT) - x(k-1)T \qquad (3)$$

$$\Delta t = kT$$

v is velocity, *x* is position, *t* is time and *k* is the discrete-time index.

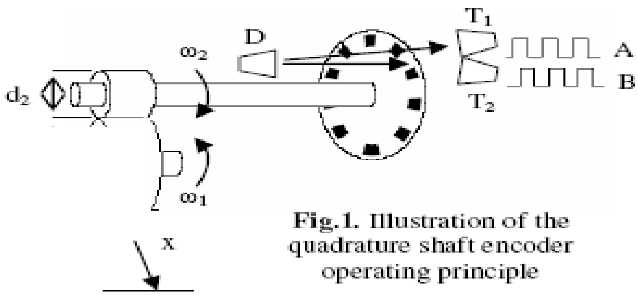

Fig. 4. Ilustration of the quadralature shaft encoder operating principale

Equation 2 states, that measuring the time it takes the encoder shaft to move a fixed angle, also gives an estimate for the velocity. Estimation based on equation 1 (method 1), has an inherent accuracy limit at low speed. The resolution of the encoder and the magnitude of the sampling interval (T) set a lower limit to the velocity estimate (*vMIN*). All velocities lower than *vMIN* will be erroneously reported as 0.

Estimation based on equation 2 (method 2), has an inherent accuracy limit at high speed. The resolution of the encoder and the magnitude of the sampling interval, (T), set an upper limit to the velocity estimate (*vMAX*). Any velocity higher than *vMAX* will produce no estimate (divide by 0 error). For systems whit very large speed range one can overcome these limitations by using method 2 for low speed and switch to method 1 when the speed exceeds a specified threshold. Another approach will be tested in this paper. A velocity sensor based on method 1 will be implemented then the *vMIN* limitation will be addressed by digitally filtering the results.

3. PRACTICAL IMPLEMENTATION OF A VELOCITY SENSOR

A velocity sensor was implemented based on method 1. The block diagram of the system is presented in figure 5. The shaft encoder is an optical quadrature encoder, as the one presented in figure 6, with the following specifications:

$d_1 = 21.4$ mm $d_2 = 3.8$mm $n = 34$

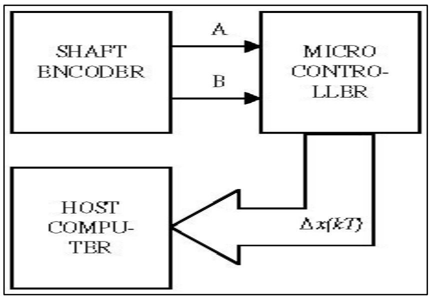

Fig. 5. Velocity sensor block diagram

where.

n is the number of windows on the slotted wheel.

One can find the expected resolution (r) of the encoder (see [3], for a more detailed. discussion). For a displacement $x_1 = \pi \cdot d_1 = 67.23$ *mm* the *d* slotted wheel makes $d_1 / d_2 = 5.63$ *turns,* corresponding $5{,}63 \cdot n = 191.47$ *impulses* on lines A and /or B.

In consequence the shaft encoder resolution is

$$r = 67.23 / 191.47 = 0.351 \text{ mm / pulse} \tag{4}$$

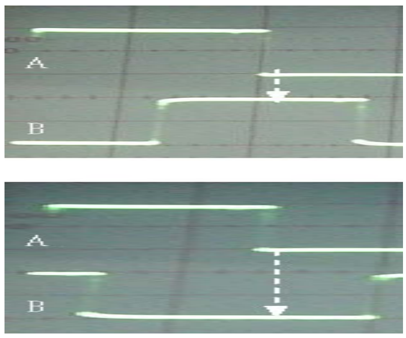

Fig. 6. Quadrature Encoder Output for Backward (**a.**) and Forward (**b**) displacement

Signals A and B are approximately ±90out of phase, depending on the direction of displacement, as presented in figure 6. These two signals has to be decoded in order to compute Δx. In [2] an ASIC is implemented to perform the task while in [4] a DSP is used. In this paper a very straightforward method for implementing the quadrature decoder with a microcontroller is presented. The microcontroller not only computes Δx but send this result at fixed time intervals (T) to a computer. Since T=const., from equation 5:

$$\Delta x\,(kT) = v.(kT) \tag{5}$$

In fact, what the computer gets is a scaled estimate of the velocity.

4. DECODER ALGORITHM AND EXPECTED ERROR

The decoding algorithm relay on two facts:

1 The number of pulses on lines A and/or B is a scaled estimate of relative displacement.

2 On the falling edge of pulse A, pulse B is LOW or HIGH, depending on the direction of displacement.

Consequently, the algorithm presented in figure 7 was implemented. The endless decoding process is interrupted every T seconds by one of the internal timers. On interrupt, Δx is sent to the host computer, via USART, then Δx is set to 0 and the decoding process restarts. Δx is an eight bit word, and since it must contain information regarding the direction of displacement, is treated as a signed integer.

Taking into account the resolution given by (4) the full scale range of decoder output is:

$$FSR\ (\Delta x)=[-128\ r\ +127\cdot\ \ r] = [-44.928+ 44.577]mm \qquad (6)$$

or, in terms of velocity:

$$\mathrm{FRS}(v) = \mathrm{FRS}(\Delta x)\ /\ T = [-5.99,\ + 5.94]\ m/s \qquad (7)$$

Relation (7) implies an estimated value of *7.5 m/s* for *T*.

For Δx counter to count 01, according to (4) a minimum displacement of ±0.351mm must occur in *T=7.5ms*.

That implies that the minimum detectible speed is:

$$V_{min} = r/T = 0.0468\ m/\ s \qquad (8)$$

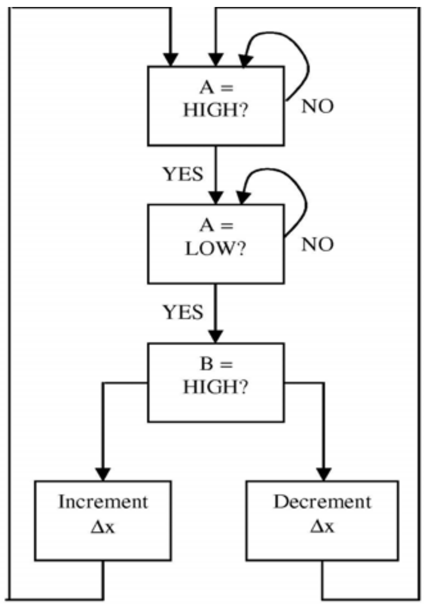

Fig.7. Decoding Algorithm Flowchart

From (7) and (8) the correct value of the estimated FSR is:

$$FSR(v) = [-5.99\ ,\ -0.0468] \cup\ [+0.0468\ ,\ 5.94] \qquad (9)$$

As stated in (9) there is a gap, *E*, in the middle of the velocity range.

$$E = (-0.047\ ,+0.047)\ m\ /\ s \tag{10}$$

Values of velocity in the interval E will be erroneously reported as 0. This is the inherent error due to method 1.

4. EXPERIMENTAL METHOD

Δx and T were estimated so far but a better estimate can be obtained experimentally. The following experiments were performed:

1. Measuring with an oscilloscope the *Rx* line of the computer's COM1 interface one can see the dat transmitted by the velocity sensor. The screen of $FSR(v) - T$ the scope looks like in figure 8. The actual value for the sampling interval is T = 7,2 m/s

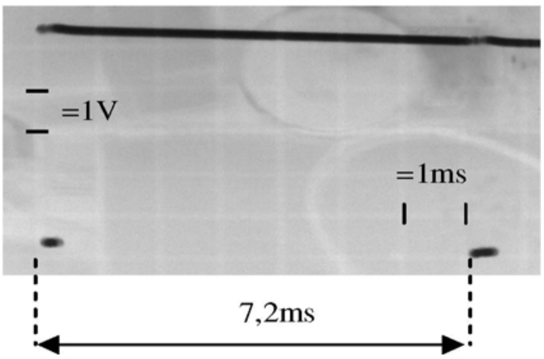

Fig. 8. Oscilogram of transmitted signal at 1V/div vertical and 1ms/div horizontal scale

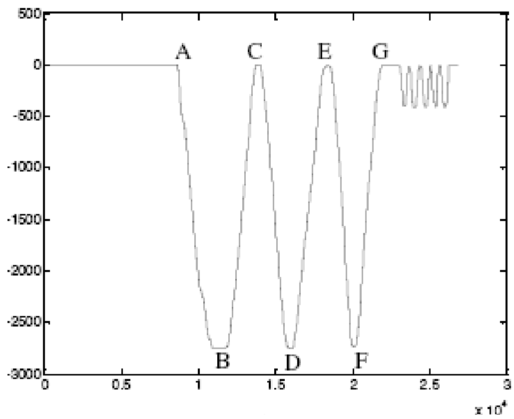

Fig. 9. Scaled plot of displacement obtained by integrating velocity data

2. In a time interval of 1 minute a number of 8357 samples were transmitted to the host computer. That leads to a sampling period:

$$T = 60\ /8357 = 0.0072\ s = 7.2ms \tag{11}$$

3. The sensor was moved back and forth between two markers 1m apart. Integrating sensor data the displacement was plotted using Matlab. The plot is represented in region A-G on figure 9. Table 1. synthesize the determination of the average resolution $r_{MED} = 0.364\ mm$.

Table 1.

y(A)	0	abs(Δy)	$abs(\Delta y)_{MED}$
y(B)	-2753	2753	
y(C)	2	2755	2747
y(D)	-2758	2760	
y(E)	-10	2748	r_{MED}
y(F)	-2740	2730	0,364mm
y(G)	-2	2738	

4. The same experiment was repeated for a displacement of 14.2cm. The plot of displacement is presented in the right side of figure 9. Table 2 shows the results. As one can see an average resolution of 0.360mm resulted.

With these new values for *T* and *r* the velocity resolution of the sensor may be computed:

$$V_{MIN} = r / T = 0.360 / 7.2 = 0.05 m / s \qquad (12)$$

Comparing with the previous estimate, (10),. one can observe a very good mach

Table 2

y(A)	-2	**abs(Δy)**	abs(Δy)$_{MED}$
y(B)	-392	390	
y(C)	-13	379	394,4
y(D)	-406	393	
y(E)	-8	398	
y(F)	-406	398	rMED
y(G)	-8	398	
y(H)	-401	393	
y(I)	-4	397	0,360mm
y(J)	-403	399	
y(K)	-4	399	

5. A very slow velocity (0.014m/s) motion of the sensor was observed. Figure 7 gives a plot of the velocity, as reported by the sensor (gray line) and after digitally filtering the data whit a 20 sample width moving average filter (MAF). One can see that since *v<vMIN* the sensor output is a pulse train with amplitude equal to *vMIN* and frequency proportional with the actual speed. In these circumstances a frequency to voltage converter would give a better estimate of the actual speed.

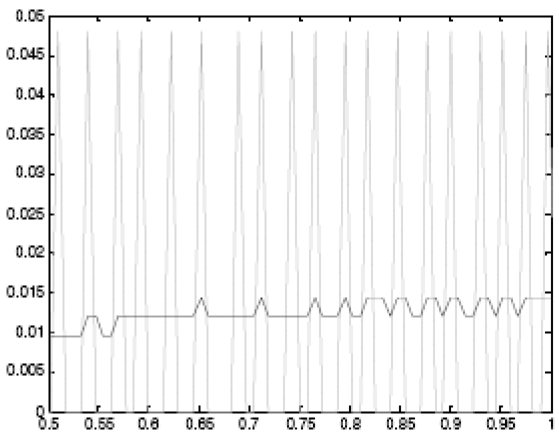

Fig. 11 Velocity plot for an impulse like excitation as reported by the sensor and after filtering with a 20 sample width filter

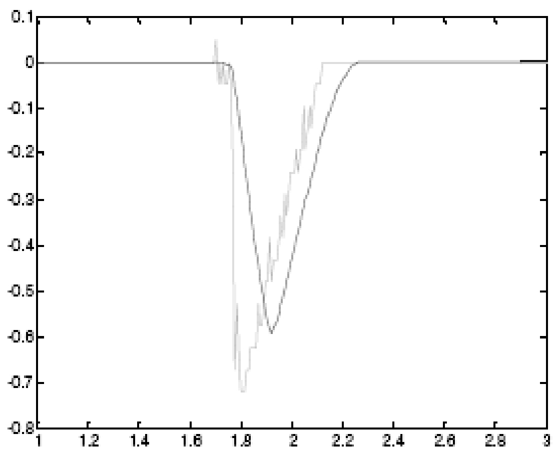

Fig.10. Velocity plot for v<vMIN as reported by the sensor (gray line) and after filtering with a 20 sample width filter

Figure 10 shows (black line) the effect of a 20 sample width MAF filter. It is obvious that the filtered version is a much better estimate of the actual (0.014m/s) speed then the original pulse train. Figure11 present the output of the sensor for an impulse like excitation

5. CONCLUSIONS

As stated in the previous section, the inherent error at very low speed can be reduced using a MAF filter. Theoretically, as the filter kernel gets wider a better estimate of the actual velocity will be obtained. Unfortunately, increasing the number of input samples used to compute one output sample would ultimately affect the slew rate of the output signal.

Beside the excellent smoothing effect, one can see that the filter output is an eroneus estimate of the actual speed, if the rate of change in velocity is too high.

Thus, using a *n* sample wide MAF filter has two oposite efects:

1. Improuves resolution, which extends the FSR of the sensor towards low speeds. If *n* increases the resolution increases.

2. Affects slew rate, which limits the ability of the sensor to folow fast changes of speed. Slew rate decrease as *n* increase. If the fastest rate of change in velocity takes *k* samples to produce, then, *n* must be less then *k* .

In conclusion, the velocity sensor described in this paper gives excellent results when used ith a MAF filter, if the monitored displacement takes place at low acceleration levels.

REFERENCES

[1]. Montgomery, D.C., Design of Analysis of Experiments, 4th Edition, John Wiley & Sons, New–York, 1996

[2]. David Alter, *TMS320 DSP Designer's Notebook: Using the Capture Units for Low Speed Velocity Estimation on a TMS320C240*, Texas Instruments, July, 1997.

[3]. Eduardo Galvh, Antonio Torralba, Leopoldo G. Franquelo, "ASIC Implementation of a Digital Tachometer with High Precision in a Wide Speed Range*", IEEE Transactions On Industrial Electronics,* Vol. 43, no. 6, December 1996,

[4]. Năsui, V.*, (*2006). *Actuatori liniari electromecanici*. Editura Risoprint Cluj - Napoca.

[5]****"Hardware and Software Resolution for a Pointing Device", *Application Notes,* Microchip Inc.,1993, pp.6-33 to 6-38.

[6]**** *TMS320x280x Enhanced Quadrature Encoder Pulse (eQEP) Module-Reference Guide,* Texas Instruments,
November 2004–Revised January 2006.

[7]*** AEROTECH. Motion Control Product Guide. Aerotech, Inc. 1990. USA