IMPROVEMENT OF ALGORITHM FOR MEASURE ANGULAR SPEED

López-Chau A.[1], Medel-Juárez J. J.[2], Quezada Quezada J. C.[3]

internal@terra.com[1], jjmedelj@yahoo.com.mx[2], quezada@hotmail.com[3]

[1, 3] Universidad Autónoma del Estado de Hidalgo, Campus Tizayuca México, Hidalgo.

[2] Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada - IPN
CICATA-IPN Legaria 694 Col. Irrigación., Deleg. Miguel Hidalgo. CP. 11850.

[2] Centro de Investigación en Computación - IPN
Av. Juan de Dios Batíz s/n casi esq. Miguel Othón de Mendizabal, Unidad Profesional "Adolfo Lopez Mateos" Col. Nueva Industrial Vallejo, Mexico D.F. C.P. 07738

Abstract. In [1] we showed that the measure of a rotatory machine can be made using an incremental encoder –an electronic digital device that produce several electric pulses on each revolution [3] – and then we proposed an algorithm to correct the intrinsic error produced in these measures. In this paper, we improve the algorithm in [1] and proposing a scheme that allows getting a better response, combining two approaches [2]: First we count the number of pulses produced in time, taking into account that the remainder time to evaluate the angular speed is dynamically computed and changed respect to value, reducing the sample time error.
This combined algorithm considered the programming in a float point microprocessor and implanted in a reprogramable digital device.

Keywords: Angular speed, rotatory machines, state machine

© A. Argüelles, J. L. Oropeza, O. Camacho, O. Espinosa (Eds.)
Computer Engineering.
Research in Computing Science 30, 2007, pp. 57-69

MEASURING ANGULAR SPEED

In many cases, incremental encoders are used to measure the speed of rotatory machines. The square pulses on the outputs of these devices, are taken into account to calculate the angular speed and direction. The basic limits considered in this paper didn't include the jitter and noise in the output of incremental encoders.

One way of measuring angular speed using incremental encoders, counting the number of pulses produced in a pediod of time:

$$Angular\ Speed\,(rpm) = \frac{60 * M}{N * k * T_b} \tag{1}$$

Where:

rpm: Revolution per minute.
N: It is the number of pulses per revolution produced. This depends on the incremental encoder used, with $N \in Z_+$.
M : It is the number of the pulses produced in the time kT_b by the incremental encoder. $M \geq 0$
k : It is a positive integer number. $k \in Z_+$
kT_b : It is the time used to count the pulses given by the incremental encoder.

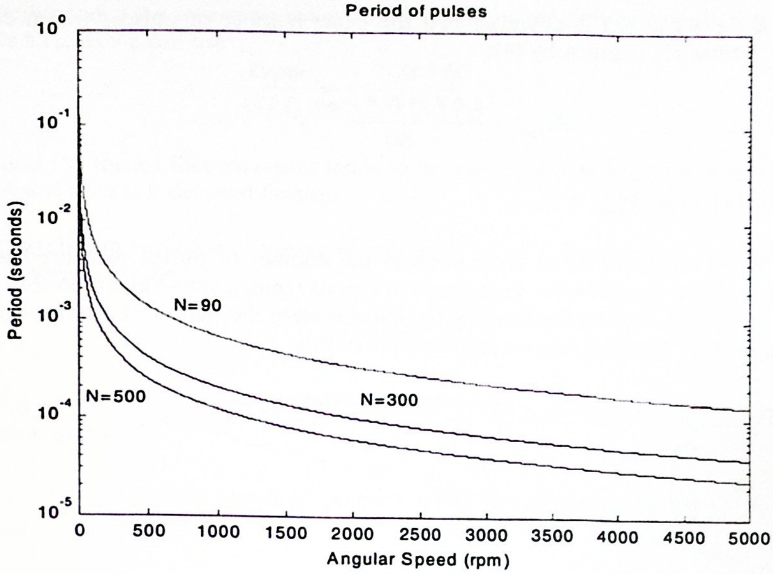
The period P of each pulse depends naturally, to the angular speed into rotatory machine, and is given by

$$P = \frac{60}{Angular\ Speed * N} \tag{2}$$

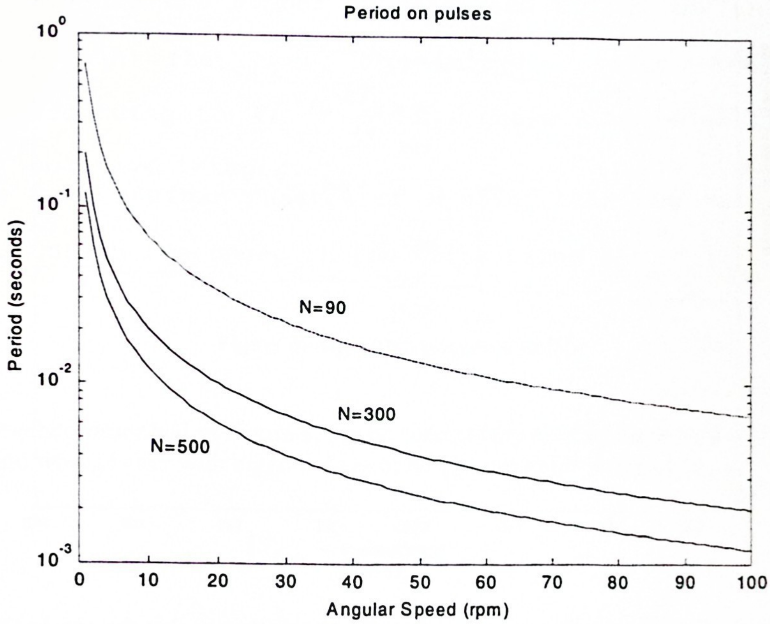
And we supposing that P is an invariant and consequently stationary for any kT_b time, accomplishing, that

$$\min(kT_b) \geq \min(P), \forall k \in Z_+ \tag{3}$$

The kT_b value tends to be greater when the angular speed tends to zero.
In Figure 1, we showing the several values of P for some incremental encoders [2]



(a)



(b)

Figure 1. Period of pulses for some values of N (a) 0<rpm≤500 (b) 0<rpm≤100.

Among the (1) and (2) is observed that the exact number of pulses, which can be counted in time kT_b is given by (4):

$$M_{real} = \frac{k * T_b * N * rpm}{60}, \tag{4}$$

but unlike (1) now $M_{real} \in Z_+$.

This situation conduces to an error because the number of pulses measured with a digital system are the same for an interval of speeds (rpm_1 to rpm_2) and therefore if the speed measured is used as feedback in a control system the output of the last one can be wrong[4] [5]. Figure 2, shows and example of this.

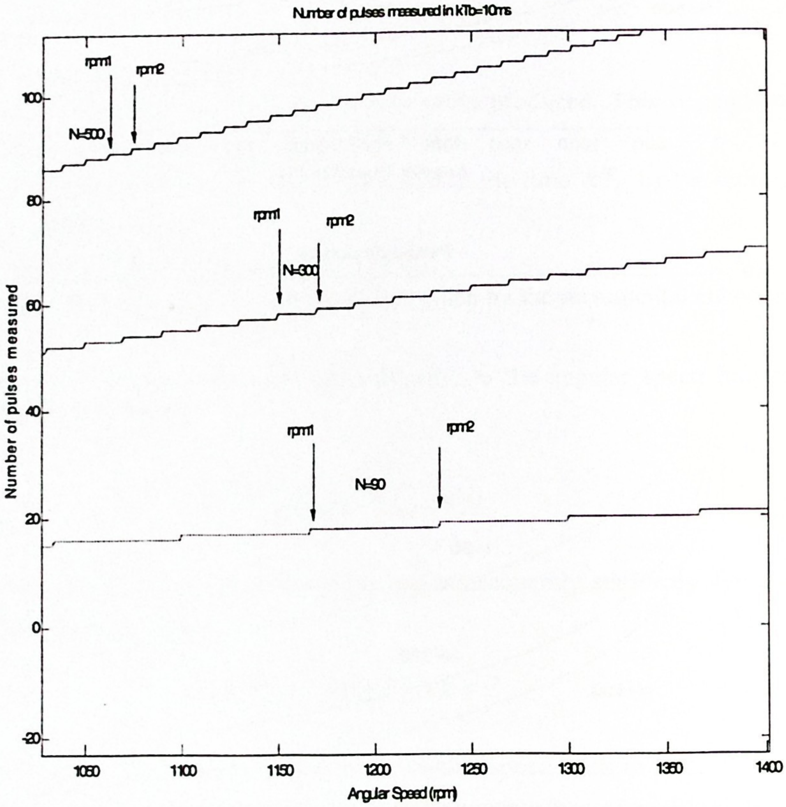


Figure 2. Number of pulses measured for kT_b= 10ms. Observe that the number of measured pulses is an integer.

In practice, once the measurement system is working, the only parameter that one can vary is kT_b, so we can use:

$$\frac{\partial rpm}{\partial (kT_b)} = -\frac{M * 60}{N * (k * T_b)^2}, \tag{5}$$

Equation (5) means that measures tends to be more stable if kT_b raise, and variations are toward zero in a descend fashion.

If the maximum variation of measures requiered for an aplication is V (rev per minute units where $V>0$) then:

$$|kT_b| \geq \left|\sqrt{\frac{M * 60}{N * V}}\right|, \tag{6}$$

Forcing $M=1$ in (6) the lowest time required to take a measure can be obtained with precision of $\pm V/2$.

kT_b in (6) can be calculated in practice using the algorithm proposed in [1] and showed again in Figure 3.

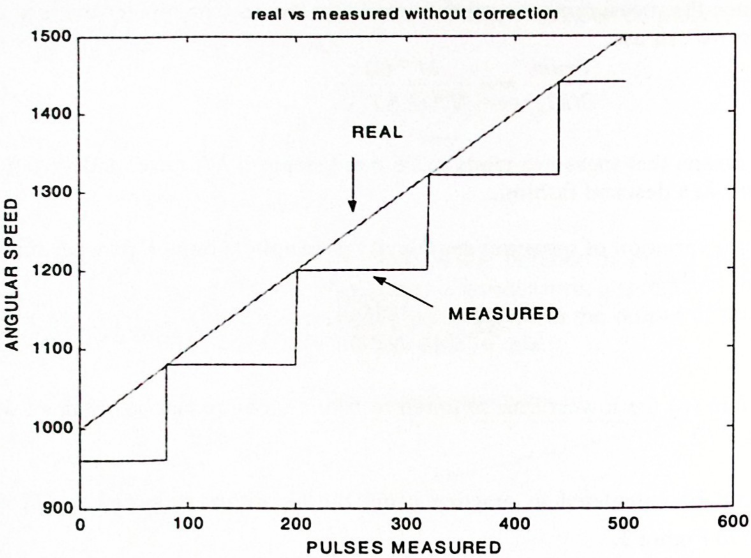
1. Begin with an arbitrary kT_b
2. Measure the number of generated pulses (M) in the kT_b time.
3. Obtain the next ("optimized") value of kT_b^+ according to $kT_b^+ = \frac{kT_b}{M} * K_1$, where $k_1>0$ is an fixed positive integer.
4. Take another measure of M using this new value of kT_b^+
5. Obtain the speed in rpm units using (1).

Figure 3. Algorithm proposed in [1].

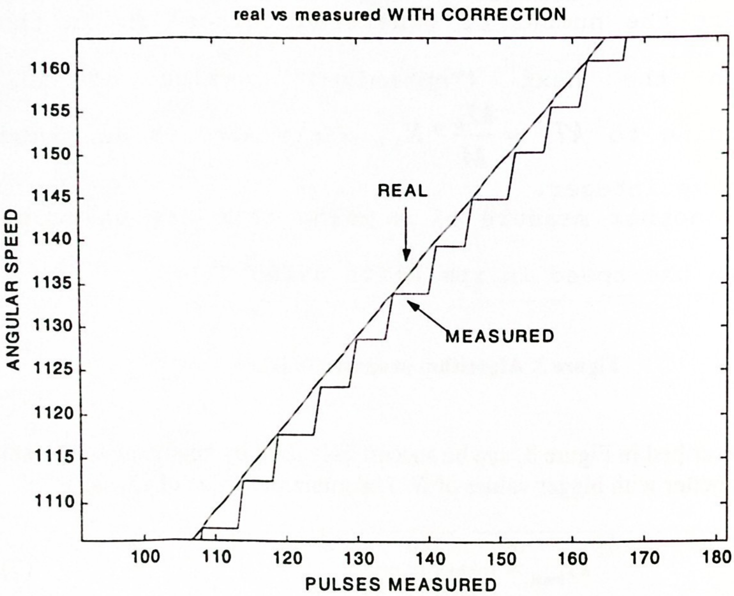
The method described in Figure 3, can be successfully used by beginning with a small kT_b and works better with bigger values of N. The minimum value of kT_b is:

$$kT_{b\min} = \frac{60}{(rpm)_{\max} * N}, \tag{7}$$

Implementing this algorithm an eight bits [1], RISC architecture microcontroller, we obtained good enough results. Simulations behavior about the algorithm [1] depicted in Figure 4 into the both sections.



(a)



(b)

Figure 4. Results of simulation of algorithm showed in Figure 3 (a) without using algorithm Figure 3 (a) Using algorithm Figure 3

IMPROVING THE ALGORITHM

When it is measuring the speed of an rotatory machine, is very important count the number of pulses produced by the incremental encoder in a period of time, another option is take a measure of the period of pulses, however both of them can leads to an error, even if the number of pulses missed is as small as one.

In this paper, we improve the algorithm showed in Figure 3. The basic idea combines the two strategies that frequently come apart: Count the number of pulses in a period of time, and measure the period of pulses. We use two subsystems, one of them for measure number of pulses and another one for measure the remained time, after kT_b finishes, in addition to this, we change dynamically the parameter k_1 in the system.
In Figure 5, simplifying the diagram designed to improve the algorithm, a brief pin description is showed in Table 1.

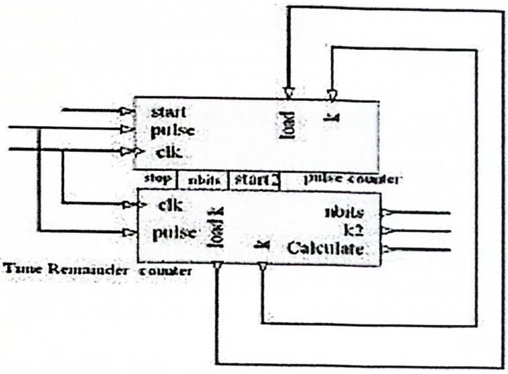


Figure 5. Basic diagram for the algorithm proposed.

PIN	DESCRIPTION
Start	Reset the entire system, begins execution of algorithm.
Clk	Clock signal for sincronizing all system.
Pulse	Pulses produced by incremental encoder.
stop	This singnal is generated by the *pulse counter* system.
n bits	The number of pulses measured in kT_b time.
nbits	Final measure
k2	Number of ticks after pulse counter systems finishes
calculate	Indicates the entire system has finished

Table 1. Pin description.

The *pulse counter* subsystem is basically a state machine that counts the pulses generated by an incremental encoder. T_b is the minimum "quantum" that corresponds with the resolution of system. A very simplified diagram is presented in Figure 6.

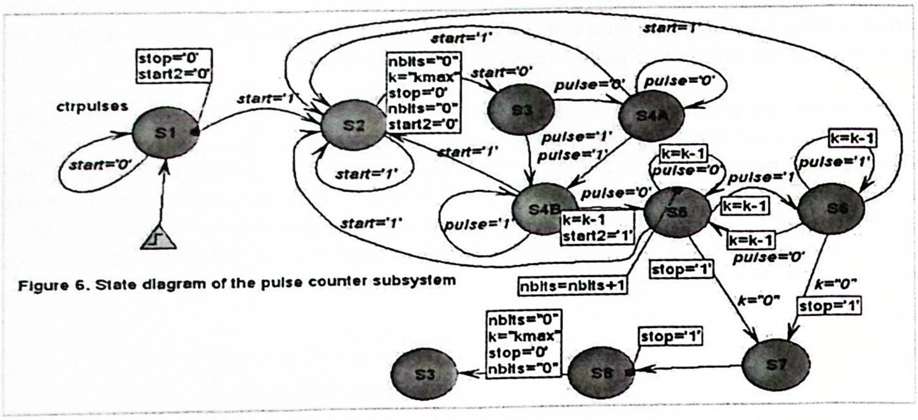


Figure 6. State diagram of the pulse counter subsystem

STATE	DESCRIPTION
S1	Initial state. Subsystem waits for a high level in *start* signal.
S2	A high level has arrived in *start* input. Wait in this state now for a low level in *start* signal. The state machine leaves this state after a falling edge in *start* signal. *nbits*, *k* and *stop* are all set to an apropiate value. This state machine returns to state S2 again if high level presented in *start* signal.
S3, S4A and S4B	These states allow waiting for the first falling edge of incremental encoder output.
S5 and s6	Begins (continues) counting the number of pulses produced in time kTb. The state machine is changing alternative between S5 and S6 states, depending on falling or risig edge of incremental encoder pulses. The kT_b time is since now (first time state machine gets into state 6) counting down.
S7	S3 and S4 states are leaved when the sample time kT_b has finished. Now, the number of pulses counted is passed to period counter subsystem, this one is notified by means of the stop signal.
S8	This state is used to permit that the state machine of second subsystem can read the output. *nbits*, *k* and *stop* are set to an appropriate value. Return to S3.

Can be observed that this subsystem is a simplified version of algorithm presented in Figure 3. Later, we'll show how to adapt the kT_b value. The task of the second subsystem consists in measure the interval between the last pulses counted and the time remained before kT_b expires. This is also a state machine, whose diagram can be seen in Figure 7.

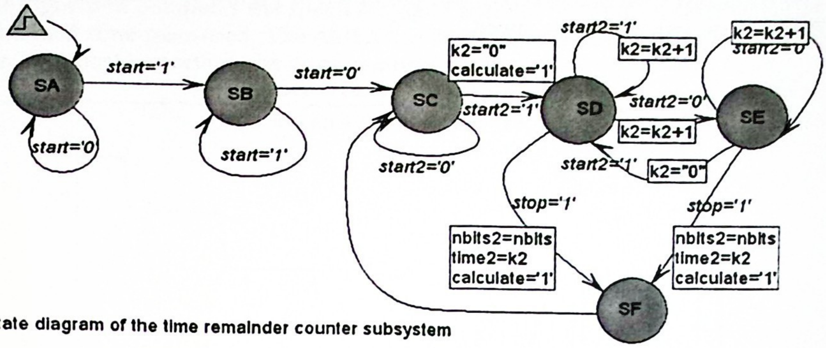


Figure 7. State diagram of the time remainder counter subsystem

STATE	DESCRIPTION
SA	Initial state. Subsystem waits for a high level in *start* signal.
SB and SC	These states allow to wait for the first pulse counted in the previous state machine. Begins (continues) counting the remainded time after a pulse has been counted.
SD and SE	The state machine is changing alternative between SD and SE states, depending on falling or risig edge of incremental encoder pulses. On each transition, k2 increments by one indicating that a tick has elapsed.
SF	Finally, when kT_b has finished stop incrementing k2. The values of k2, and nbits (number of pulses counted) are presented in outputs of this subsystem.

In order to compute the next value of k_1, we propose the use of an averager: so, when state machine arrives state SF, a recursive averager (not showed in simplified diagram of Figure 7) compute the average vaue of k_2. This is a fast way to compensate the next value of k_1 (kT_b^+) in the first state machine. Observe that for both subsystems, T_b is taken of `clk`, so we can obtain k_1 (average of k_2) mentioned in Figure 3.

SIMULATIONS

For all simulations, consider that the speed of a rotatory machine is constant at least while taking the measure. In practice acceleration must be also considered, [2][3]. The first simulation presented corresponds to running the algorithm with the following parameters: $k=1024$, $T_b=1x10^{-6}$ *sec* (state machines working at *1MHz*) and $M=500$. Figure 8 shows the number of pulses measured, observe that at this point, there is not any improvement with respect to the original algorithm, the number of pulses measured are always an integer, and this is a cause of error in calculating the speed.

The remained time is measured, and presented in Figure 9, observe a constant pattern that decays with respect to speed.

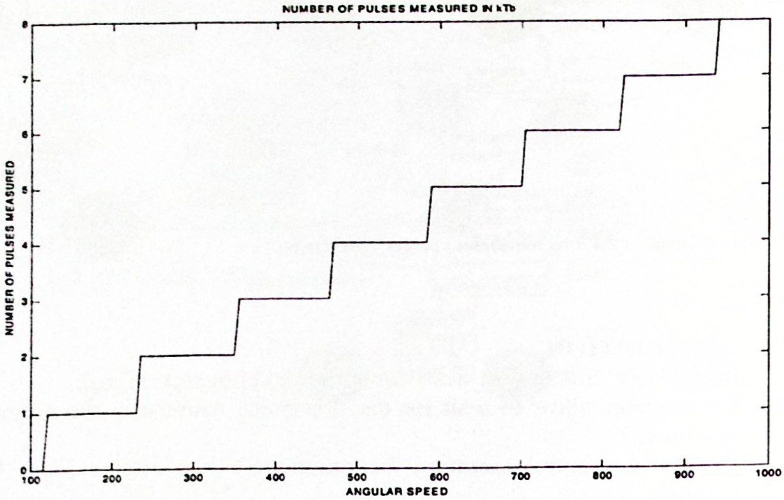


Figure 8. Number of pulses measured for T_b= 1us, k= 1024.

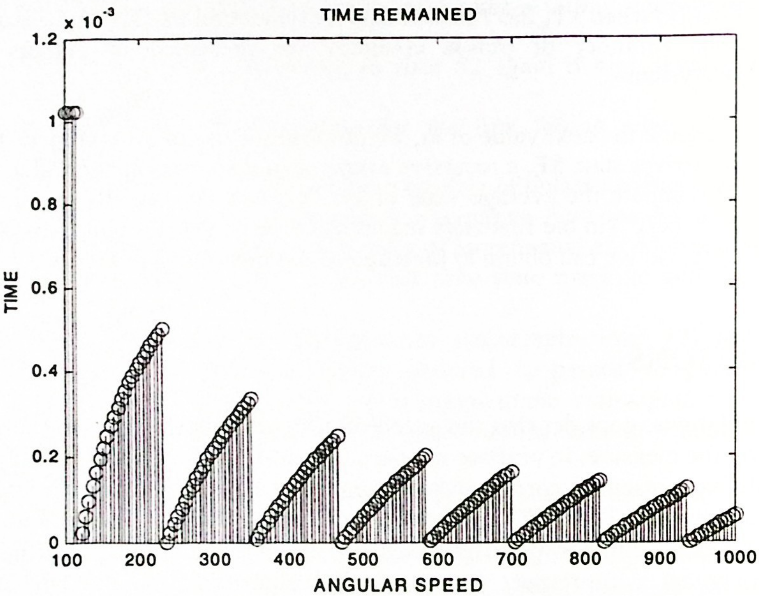


Figure 9. Remained time.

Finally, the algorithm computes the speed taking in account both the number of pulses and the remained time measured. The difference between real and measured speeds is near to one, but most important, the error can be a fraction._

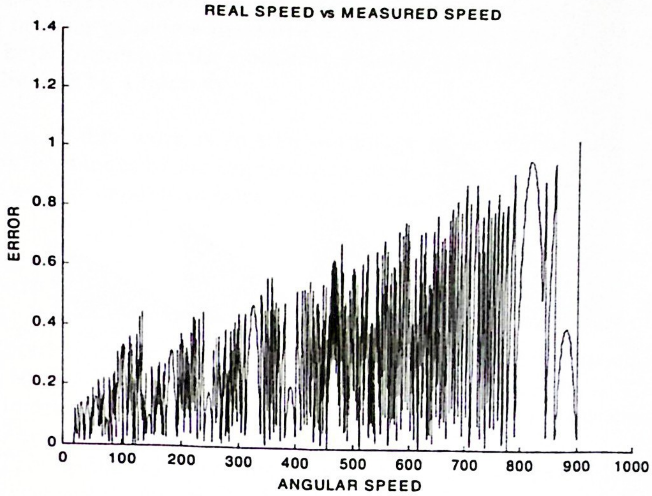
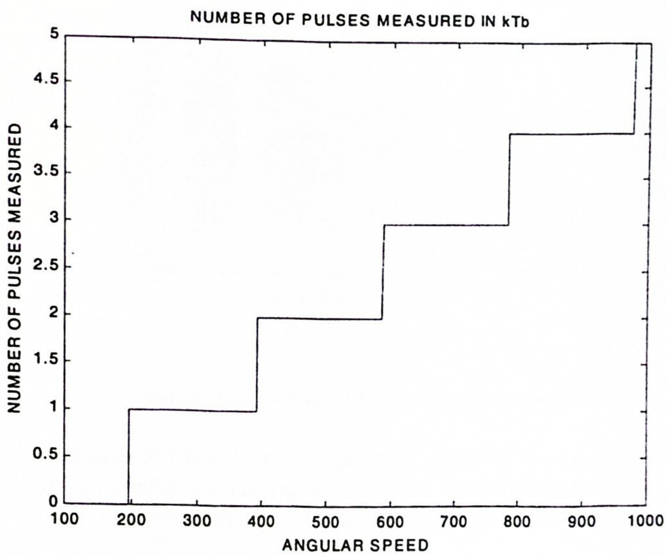
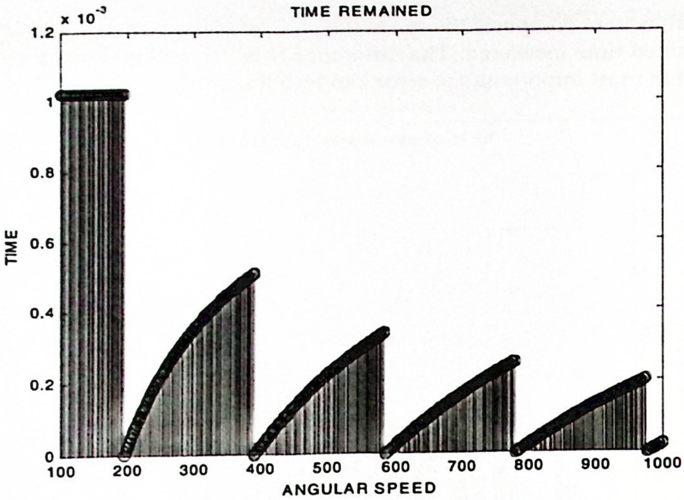


Figure 10. Error: difference between real and measured speed

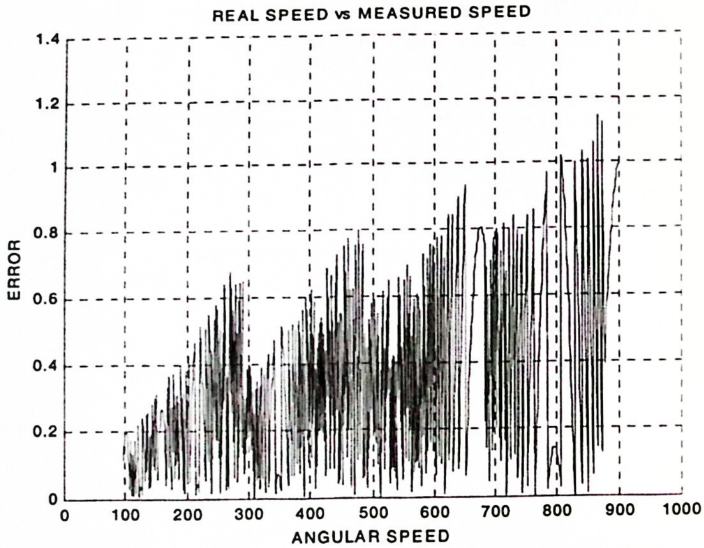
Figure 11 shows more results.



(a)



(b)



(c)

Figure 11. (a), (b) and (c) some other results.

The two main differences between this improvement and the original algorithm, is that the error is reduced and more over, is now a fraction, other important difference is that we can change parameters dynamically, using a recursive average.

CONCLUSIONS

In this paper, we have seen that when measuring angular speed with an incremental encoder, error presented is due to the number of pulses counted as integers. We propose a new improvement of an algorithm for measure of angular speed, in this new version the number of pulses measured and the remained time is taking into account to produce better results. In the simulations can be seen that the error is not necessary an integer, but can be a fraction.

The next step in this work is to take advantage of parallelism, we are working in create several instances of the improvement presented in this paper and introduce the outputs to a system capable to select the best measure.

Bibliography

[1] [1] "Reducción de error en la medición de velocidad de maquinas rotatorias", J. J. Medel-J., Guevara López Pedro, López Chau Asdrúbal, "CICINDI-2005. 6th International Conference on Control, Virtual Instrumentation, and Digital Systems

[2] [2] "An optimised algorithm for velocity estimation method for motor drives", Dapos,Arco, S.; Piegari, L.; Rizzo, R.

[3] 4th IEEE International Symposium on

[4] Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003

[5] [2]http://www.gpi-encoders.com/, http://www.servotek.com/

[6] [3]"Sensores y Acondicionadores de Señal" 3ª edición. Ramón Payas Areny.

[7] Alfa Omega Marcombo. pag.455-456.

[8] [4] "Sistemas Modernos de Control, Teoría y Práctica". 2ª edición. Richard C. dorf. Addison Wesley Iberoamerica. pag. 508

[9] [5] "Sistemas de Control en Tiempo Discreto" 2a Edición. Katsuhiko Ogata. pag 8.