

## Reducción del Error en la Medición de Velocidad en Máquinas Rotatorias

López Chau, A.,<sup>a</sup> Medel Juárez, J. J.,<sup>b</sup> Guevara López, P.<sup>c</sup>

<sup>a</sup> Universidad Autónoma del Estado de Hidalgo  
km 2,5 Carretera Federal México Pachuca, Jilotepec, México.

<sup>b</sup> Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Instituto Politécnico Nacional  
Legaria 694, Col. Irrigación, 11850, Distrito Federal, México.

<sup>c</sup> Centro de Investigación en Computación, Instituto Politécnico Nacional  
Av. Juan de Dios Batíz S/N, Col. Nueva Industrial Vallejo, 07738, Distrito Federal, México.  
pguevara@ipn.mx

### RESUMEN

El medir la velocidad angular real de una máquina rotatoria es un problema que tiene que valorarse, ya que si dicha medición retroalimenta a un sistema de control, la respuesta de este último pudiera no ser la adecuada. Para medir la velocidad angular, es común utilizar un codificador incremental - un dispositivo que produce pulsos eléctricos conforme gira la fecha de la máquina rotatoria,- junto con circuitos electrónicos digitales. La implementación de este tipo de soluciones digitales, en general introduce un error en las mediciones, ya que solamente pueden contarse número de enteros de pulsos en cada periodo de tiempo, o si se mide el intervalo de tiempo entre un pulso y otro, el error se introduce de acuerdo a la resolución del temporizador utilizado. En este trabajo se detallan dos métodos para minimizar el error en la medición de velocidad angular cuando se usan medios digitales.

### 1. INTRODUCCIÓN

En muchas aplicaciones, los codificadores incrementales son usados para medir la velocidad de máquinas rotatorias. Los pulsos rectangulares en las salidas de estos codificadores son utilizados para cálculo de velocidad y sentido de giro.

Una forma básica de medir velocidad angular usando estos dispositivos es contar el número de pulsos producidos por unidad de tiempo la relación:

$$\omega = \frac{60 * M}{N * k * T_b}, \quad (1)$$

Donde  $\omega$  es la velocidad angular, en revoluciones por minuto,  $N$  es el número de pulsos por revolución que produce el codificador incremental:  $N \in Z_+$ ,  $M$  es el número de pulsos producidos en el tiempo  $kT_b$ ,  $M \geq 0$ ,  $k$ : es un parámetro ajustable:  $k \in Z_+$ , y  $kT_b$  es el tiempo durante el cual se cuentan los pulsos producidos por el codificador incremental.

En la práctica, una vez que el sistema de medición digital está funcionando, el único parámetro que se puede variar (mediante la programación del firmware) es  $kT_b$ .

La medición de velocidad en máquinas rotatorias es un problema que todavía no está resuelto completamente [1] y que se ha enfrentado de diversas maneras. En este trabajo se presentan los resultados de un algoritmo diseñado para reducir el error de lectura de velocidad angular usando codificadores incrementales, tomando como base las ecuaciones presentadas anteriormente.

### 2. MEDICIÓN DE VELOCIDAD

Los codificadores incrementales proporcionan ondas rectangulares [2], que se considerarán estacionarias en el intervalo en que se realizan las mediciones.

De la Ec. (1), el periodo  $P$  de cada pulso es calculado mediante:

$$P = \frac{60}{\omega * N}, \quad (2)$$

de donde :

$$\min(kT_b) \geq \min(P), \forall k \in Z_+; \quad (3)$$

$kT_b$  se incrementa cuando la velocidad angular tiende a cero. La Fig. 1 muestra algunos valores de  $P$  para codificadores incrementales comerciales [3].

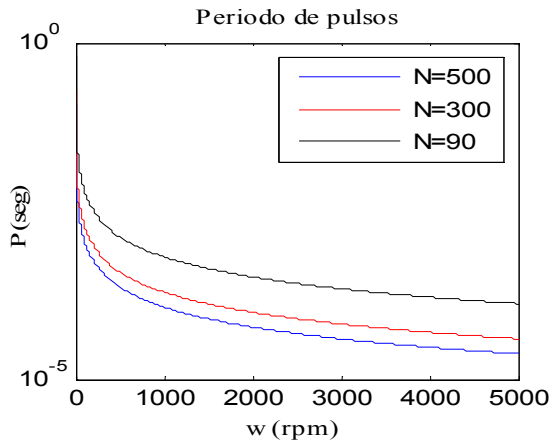


Fig. 1. Valores de P para diferentes codificadores incrementales en el rango (0 a 5 000 r.p.m.).

La cantidad de pulsos que son contados durante el tiempo  $kT_b$  es exactamente:

$$M_{real} = \frac{k * T_b * N * \omega}{60} \quad V: \quad (4)$$

Se observa que  $M_{real} \in Z_+$  a diferencia de  $M \in R$  en la Ec. (1). Esta situación conduce intrínsecamente a un error, y si esta lectura se usa para un sistema de control [4-5], la respuesta de este último no es en general correcta, ya que la cantidad de pulsos medidos es la misma para un intervalo de velocidades, y por lo tanto la velocidad obtenida es falsa, esta situación se observa en la Fig. 2, para N=500. .

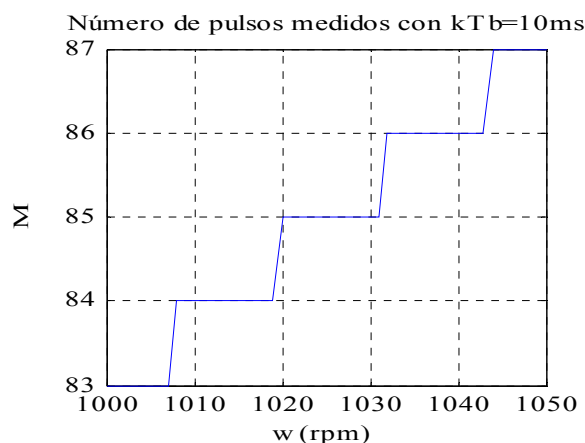


Fig. 2. La cantidad de pulsos es un número entero.

Se observa, que en el alcance de 1 010 r.m.p. a 1 019 r.m.p., se cuentan 84 pulsos, y la velocidad

calculada por el sistema es 1 008 r.p.m. en todo el intervalo. Este comportamiento (dar errores en las mediciones) es independiente del valor de  $kT_b$  elegido y del poder computacional del procesador utilizado.

Para reducir este error, se parte de la Ec. (1), de la cual se encuentra que:

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

A fin de reducir lo más posible el error en la medición de velocidad, se observa que si en (5) la máxima variación permitida se establece a un valor arbitrario y constante, al que llamaremos V, entonces:

$$|kT_b| \geq \left| \sqrt{\frac{M * 60}{N * V}} \right| \quad (6)$$

donde  $kT_b$ , M y N han sido definidos en la Ec. (1).

### Máxima Variación de Velocidad Angular en la Medición

Forzando a  $M=1$  (sólo un pulso producido durante el intervalo de observación), se obtiene el menor tiempo requerido para realizar una medición. La exactitud de la medición puede mejorarse en un factor de  $\sqrt{2}$  si se utilizan las dos señales de codificadores incrementales en cuadratura. En la práctica, los temporizadores utilizados para la generación del intervalo de medición no siempre pueden entregar exactamente el valor requerido de  $kT_b$ , especialmente si N es grande, lo que contribuye al error de medición.

Un algoritmo desarrollado por los autores de este trabajo, para la reducción del error es el siguiente:

1. Comenzar con un  $kT_b$  arbitrario.
2. Medir el número de pulsos generados (M) en el tiempo  $kT_b$ .
3. Obtener el siguiente valor de  $kT_b$  optimizado (llamado ahora  $kT_b^+$ ) de acuerdo a  $kT_b^+ = \frac{kT_b}{M} * K_1$ , donde  $k_1 > 0$  es un número entero positivo.
4. Tomar otra medición de M usando ahora el valor  $kT_b^+$ .
5. Obtener el valor de velocidad angular usando Ec. (1).

El método descrito anteriormente puede ser utilizado comenzando con un valor pequeño de  $kT_b$  y trabaja mejor con valores altos de N. Si se conoce

de antemano  $N$ , se puede proponer un valor inicial de  $kT_b$  de modo que converja  $kT_b^+$  más rápido. El mínimo valor de  $kT_b$  es descrito por:

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

Al implementar esto en un microcontrolador con arquitectura RISC, se obtuvieron resultados que se muestran en la sección 3.

**2.1 Reduciendo Aún Más el Error de Medición**

El algoritmo mostrado anteriormente produce mediciones de manera rápida y con menos error si se compara con una medición realizada usando un  $kT_b$  fijo. Sin embargo, se puede mejorar todavía este algoritmo tomando en cuenta dos estrategias que normalmente están separadas: Contar el número de pulsos por unidad de tiempo (similar al algoritmo propuesto) y medir el periodo de los pulsos. Para esto se usan dos subsistemas, uno para medir el número de pulsos producidos en el tiempo  $kT_b$  y otro subsistema para medir el tiempo que dura el último pulso después de que termina  $kT_b$ .

El primer subsistema, que cuenta los pulsos es una máquina de estados, cuyo diagrama simplificado se muestra en la Fig. 3; mientras que la descripción sintetizada de cada estado se explica a continuación.

Estado S1. Estado inicial. El sistema espera la señal de *start*, para iniciar. Pasa S2 cuando *start* está en alto.

Estado S2. La máquina de estados permanece en S2 hasta que se presente un flanco descendente en la señal *start*. Los registros *nbits*, *k* y *stop* se establecen con valores apropiados. Notar que se regresa a S2 si la señal *start* pasa a alto.

Estados S3, S4A y S4B. Estos estados permiten esperar al primer flanco descendente que se presente en la salida del codificador incremental.

Estados S5 y S6. Comienza o continúa el conteo de pulsos que produce el codificador en el tiempo  $kT_b$ . La máquina de estados cambia entre S5 y S6, de acuerdo a los flancos de subida o bajada que produce el codificador incremental.  $kT_b$  comienza su conteo descendente desde la primera vez que entra a S6.

Estado S7. S3 y S4 son abandonados cuando  $kT_b$  expira. Desde ahora, el número de pulsos contados se pasa al subsistema contador de periodos, el cual es notificado de esto mediante la señal *stop*.

Estado S8. Este estado es usado para permitir a la máquina de estados del segundo subsistema leer la salida del primero. Los registros *nbits*, *k* son establecidos al valor predeterminado.

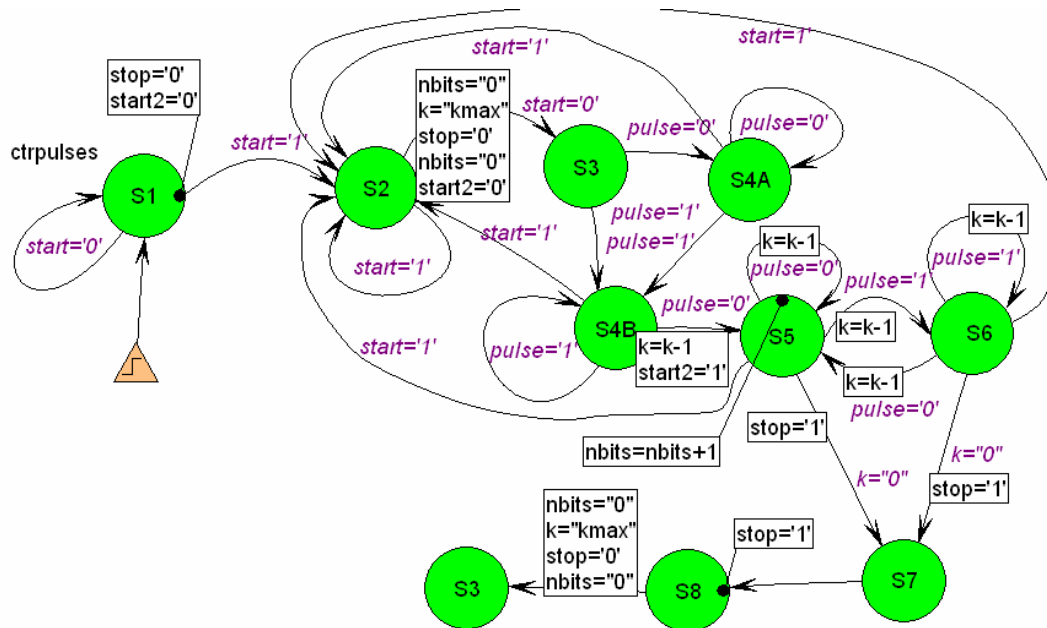


Fig. 3. Primer subsistema empleado para la mejora del algoritmo.

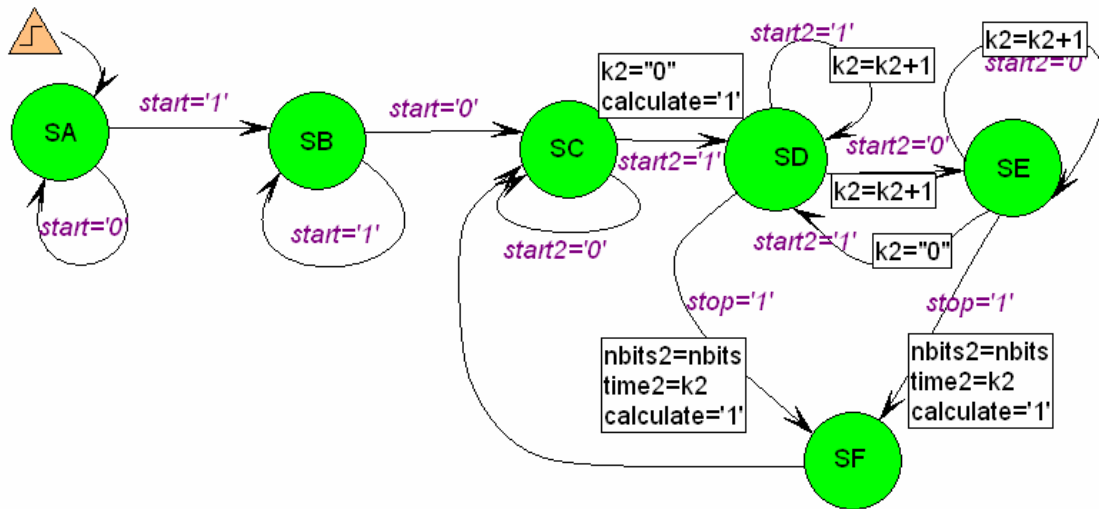


Fig. 4. Segundo subsistema empleado para la mejora del algoritmo.

La función del segundo subsistema consiste en medir el intervalo de tiempo entre el último pulso contado y el tiempo remanente antes de que  $kT_b$  termine. Para esto se diseñó la máquina de estados mostrada en la Fig. 4.

Estado SA. Estado inicial. El subsistema espera nivel alto de la señal *start*.

Estados SB y SC. Estos estados permiten esperar al primer pulso contado por el subsistema anterior.

Estados SD y SE. La máquina comienza o continúa el conteo del tiempo remanente después del último pulso contado. De manera similar a la máquina anterior, se alterna entre SD y SE en flancos positivos y negativos del codificador incremental, en cada transición el registro *k2* se incrementa, indicando que ha contado un *tick* del reloj que sincroniza a la máquina.

Estado SF. Cuando  $kT_b$  ha expirado, se detiene el incremento del registro *k2*. Los registros *k2* y *nbits* son presentados en la salida del subsistema.

$kT_b$  se calcula dinámicamente usando un promediador recursivo sobre *k2* (no mostrado en la Fig. 4) en el momento en que se alcanza SF, este cálculo permite obtener rápidamente  $kT_b^+$ .

**3. RESULTADOS**

Mediante la implementación del algoritmo propuesto, en un microcontrolador de 8 bits

arquitectura RISC (Familia AVR), se obtuvieron resultados que permiten comprobar la efectividad del algoritmo.

**3.1. Resultados de la Implementación con Microcontrolador**

La Fig. 5, muestra la medición de pulsos generados por un codificador incremental sin usar corrección alguna, y se observa claramente el problema mencionado en párrafos anteriores.

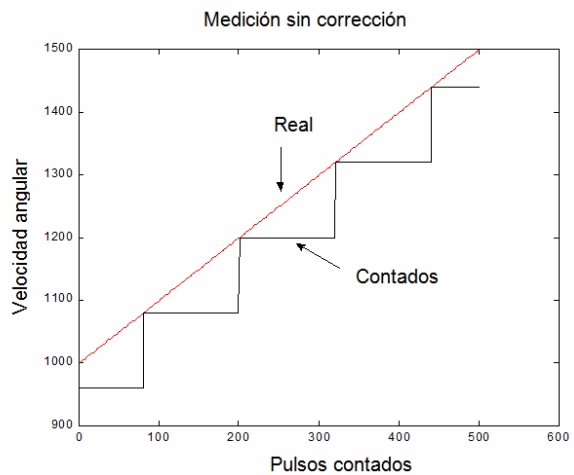


Fig. 5. Resultados de medición sin aplicar algoritmo de corrección de error.

Aplicando el algoritmo para corrección de error, se obtiene los resultados mostrados en la Fig. 6; donde se observa que ahora el error se redujo notablemente.

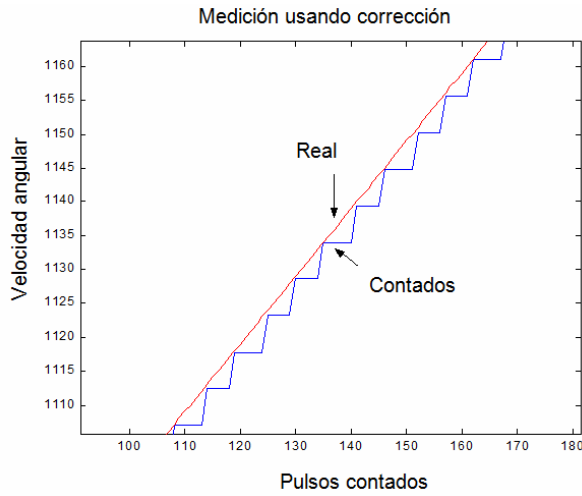


Fig. 6. Resultados de medición aplicando algoritmo de corrección de error.

**3.2. Resultados con el Algoritmo Mejorado**

Para la mejora del algoritmo, las máquinas de estado se implementaron en dispositivos de lógica programable, en vez de microcontroladores, debido a que la estructura del algoritmo no es completamente secuencial y presentan cierto paralelismo, al ejecutarse ambas máquinas al mismo tiempo ( $kT_b^+$ ).

La primera máquina de estados cuenta el número de pulsos, cuyo resultado se muestra en la Fig. 7, aparentemente no hay una mejora notable en las mediciones.

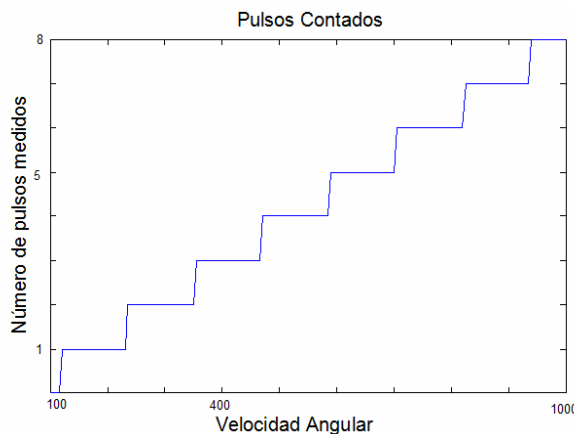


Fig. 7. Resultado del primer subsistema.

El segundo subsistema, implementado en la máquina de estados de la Fig. 4, calcula los tiempos remanentes, mismos que se muestran en la Fig. 8.

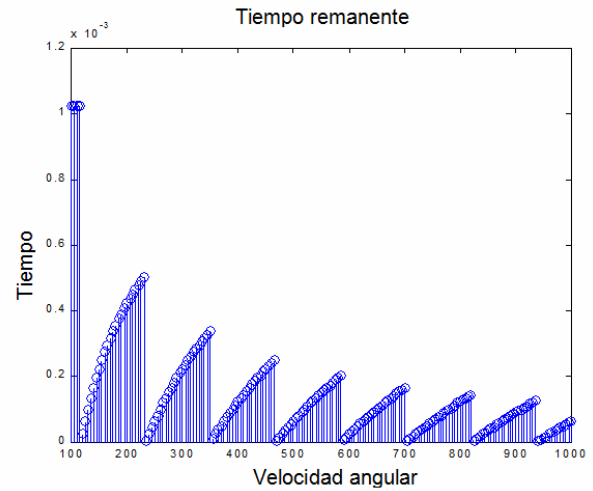


Fig. 8. Resultado del segundo subsistema

Considerando ambos resultados, se calcula finalmente el valor de velocidad, resultando en un error menor a 1 r.p.m. Este resultado se muestra en la Fig. 9.

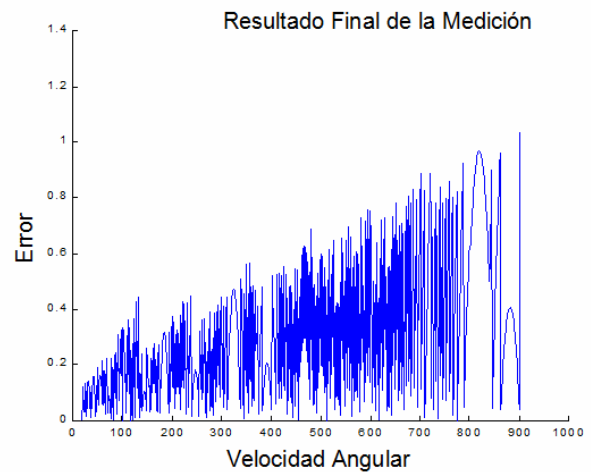


Fig. 9. Error producido con el algoritmo mejorado.

Un resultado interesante es que el error es fraccional y menor a uno, algo no común en instrumentos de medición de velocidad comerciales de propósito general.

**4. CONCLUSIONES**

En éste trabajo demostró que cuando se mide velocidad angular usando codificadores incrementales, el error de medición se debe a que los pulsos contados son tratados como enteros por

los sistemas electrónicos digitales. Se propuso un nuevo algoritmo y una mejora del mismo tomando en cuenta el número de pulsos y el tiempo remanente, para producir mejores resultados. La simulación del algoritmo proporciona información acerca del error, el cual en general es menor a la unidad.

La siguiente etapa en este trabajo consiste en realizar el análisis de optimalidad entre calidad de respuesta y tiempo de respuesta.

## REFERENCIAS

- [1] Dapos, Arco, S.; Piegari, L.; Rizzo, R. "An optimized algorithm for velocity estimation method for motor drives", 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.
- [2] Ramón Payas Areny, "Sensores y Acondicionadores de Señal" 3ª edición. Alfa Omega Marcombo. pag.455-456.
- [3] <http://www.gpi-encoders.com/>, <http://www.servotek.com/>, Enero 2008.
- [4] Richard C. Dorf, "Sistemas Modernos de Control, Teoría y Práctica". 2ª edición. Addison Wesley Iberoamerica. pag. 508.
- [5] Katsuhiko Ogata, "Sistemas de Control en Tiempo Discreto" 2ª Edición. pag 8.