

Algoritmo para la aproximación de la velocidad de giro de un eje mediante un *encoder* incremental¹

Algorithm to approximate the rotational speed of an axis by an incremental *encoder*²

Algoritmo para a aproximação da velocidade de rotação de um eixo mediante um *encoder* incremental³

José Danilo Rairan-Antolines⁴

José Miguel Fonseca-Gómez⁵

SICI: 0123-2126(201307)17:2<293:AAVGEE>2.0.TX;2-G

¹ Fecha de recepción: 13 de diciembre de 2011. Fecha de aceptación: 6 de diciembre de 2012. Este artículo se deriva del proyecto de investigación *Planteamiento de un algoritmo de control de sistemas dinámicos nuevo, basado en el proceso de toma de decisiones humano*. Registro: 1-72-176-09. Desarrollado por el Grupo de Investigación en Control Electrónico (GICE), de la Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

² Received: December 13, 2011. Accepted: December 6, 2012. This Article is a result of the research project entitled "*Planteamiento de un algoritmo de control de sistemas dinámicos nuevo, basado en el proceso de toma de decisiones humano*" (Approach to a control algorithm of new dynamic systems, based on the process of human decision making). Registration: 1-72-176-09. Developed by the Research Group on Electronic Control (GICE), University Francisco José de Caldas, Bogotá, Colombia.

³ Data de recebimento: 13 de dezembro de 2011. Data de aceitação: 6 de dezembro de 2012. Este artigo é derivado do projeto de pesquisa *Aproximação de um novo algoritmo de controle de sistemas dinâmicos, baseado no processo humano de tomada de decisões*. Registro: 1-72-176-09. Desenvolvido pelo Grupo de Investigación en Control Electrónico (GICE, Grupo de Pesquisa em Controle Eletrônico), da Universidade Distrital Francisco José de Caldas, Bogotá, Colômbia.

⁴ Ingeniero electricista, magíster en Automatización Industrial y doctor en Sistemas y Computación de la Universidad Nacional de Colombia. Docente de la Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: drairan@udistrital.edu.co.

⁵ Tecnólogo en Electricidad de la Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: jose60022@gmail.com.

Resumen

El algoritmo propuesto en este artículo aproxima la velocidad angular por medio de las señales que vienen de un *encoder* incremental. Este algoritmo remueve las oscilaciones que aparecen, incluso a velocidad constante, cuando se usan los algoritmos tradicionales, los cuales son conocidos como *a tiempo fijo* y *a espacio fijo*. La idea clave en la eliminación de las oscilaciones es la sincronización entre una señal cada δ de tiempo y los pulsos que vienen del *encoder*. El resultado más sobresaliente en este artículo dice que el promedio armónico entre la salida del algoritmo propuesto y la salida de una modificación de este mismo algoritmo resulta tener el error relativo más pequeño posible. El error relativo del algoritmo nuevo es siempre menor que la mitad de lo que sería usando los algoritmos tradicionales, lo cual demuestra que el algoritmo es óptimo. Al finalizar el artículo, y usando pulsos reales provenientes de un *encoder*, se corren algunos experimentos en Simulink de Matlab. Estos experimentos sirven como prueba experimental del rendimiento de los algoritmos.

Palabras clave

Aproximación de la velocidad, *encoder* incremental, espacio fijo, promedio armónico, tiempo fijo.

Abstract

The algorithm proposed in this paper approaches to the angular speed by signals coming from an incremental *encoder*. This algorithm removes the oscillations that appear even at constant speed, when using the traditional algorithms, which are known as *fixed time and at fixed space*. The key idea on the elimination of the oscillations is the synchronization between a signal every δ time and the pulses coming from the *encoder*. The most outstanding result in this paper says that the harmonic average between the output of the proposed algorithm and the output of a modification of this same algorithm turns out to have the smallest possible relative error. The relative error of the new algorithm is always smaller than half of what would be by using traditional algorithms, which shows that the algorithm is optimal. At the end of the article, and using real pulses from an *encoder*, some experiments were run in Matlab Simulink. These experiments serve as an experimental test of the algorithms performance.

Keywords

Approximation of the speed, incremental *encoder*, fixed space, harmonic average, fixed time.

Resumo

O algoritmo proposto neste artigo se aproxima da velocidade angular por intermédio dos sinais provenientes de um *encoder* incremental. Este algoritmo elimina as oscilações que aparecem mesmo a velocidade constante, quando se utilizam os algoritmos tradicionais, conhecidos como *a tempo fixo* e *a espaço fixo*. A ideia-chave na eliminação das oscilações é a sincronização entre um sinal cada δ de tempo e os pulsos provenientes do *encoder*. O resultado mais surpreendente no presente artigo diz que a média harmônica entre a saída do algoritmo proposto e a saída de uma modificação deste mesmo algoritmo acaba por ter um erro relativo tão pequeno quanto possível. Erro relativo do algoritmo novo é sempre inferior da metade do que seria usando algoritmos tradicionais, o que mostra que o algoritmo é ótimo. No final do artigo, e usando pulsos reais provenientes de um *encoder*, correm-se alguns experimentos em Simulink de Matlab. Estes experimentos servem como prova experimental do rendimento dos algoritmos.

Palavras-chave

Aproximação da velocidade, *encoder* incremental, espaço fixo, média harmônica, tempo fixo.

1. Introducción

Los *encoders* incrementales son la opción más popular para medir la velocidad en aplicaciones industriales, dado su bajo costo y su simplicidad, de ahí que cualquier propuesta para mejorar el sensor tendrá un impacto positivo en la industria. La revisión de la literatura permite identificar cuatro áreas de trabajo, por medio de las cuales puede evidenciarse la importancia de cualquier aporte en el área: 1. la generación de nuevos tipos de *encoders*; 2. la generación de propuestas para reducir los errores propios del sensor; 3. el reporte de aplicaciones; 4. la generación de algoritmos para mejorar la calidad de la estimación, de los cuales la propuesta en este artículo es uno más.

Las propuestas de *encoders* nuevos incluyen, por ejemplo, el uso del efecto *hall*, efecto capacitivo o salidas con forma de onda distinta a la cuadrada, como en Hoang y Jeon (2011). Acerca del área 2, el trabajo en Nandayapa *et al.* (2012) elimina la naturaleza discreta del sensor, por medio de la estimación de la posición. Por otra parte, en cuanto al área 3, los *encoders* son usados en robótica, control de movimiento y radares, entre muchas otras aplicaciones. Por ejemplo, en el trabajo de Voldán (2012) se utiliza la estimación de la velocidad para calcular la aceleración, y de ahí el torque. Finalmente, respecto al área 4, en Negrea *et al.* (2012) un algoritmo mejora los algoritmos tradicionales, cambiando el tiempo de discretización.

El *encoder* incremental es un dispositivo electromecánico que permite aproximar la velocidad angular de un eje. La parte mecánica consiste en un disco ranurado, a espacios iguales. La parte electrónica genera una luz, la cual es interpretada por un receptor como 1 cuando la luz pasa la ranura y llega al receptor, y como 0 en caso contrario.

La aproximación de la velocidad requiere dos mediciones: una de posición y una de tiempo. El error en la aproximación tiene dos fuentes: la primera es que en la práctica solo pueden hacerse medidas a tiempo discreto, y por tanto la velocidad medida, ω_m , es $\Delta\theta/\Delta t$; la segunda tiene que ver con la medida de esos δ , lo cual lleva a otro error. El δ de posición, $\Delta\theta$, es un múltiplo entero del

número de pulsos del *encoder*, pul, por tanto existe un error de truncamiento. De igual manera, el δ de tiempo, Δt (el cual se llamará dt en adelante), es un múltiplo entero del reloj del sistema, t_s , donde se implemente el algoritmo.

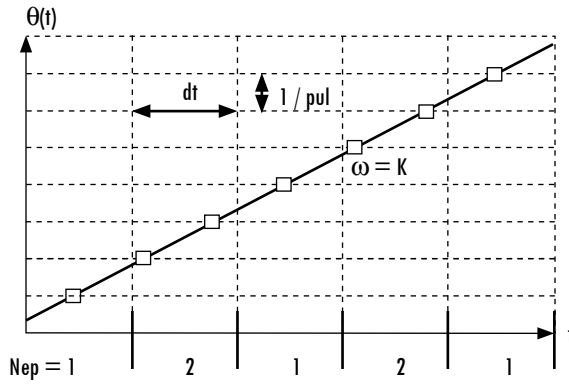
En la literatura son dos los algoritmos clásicos para la aproximación de la velocidad, estos se conocen como *a tiempo fijo* y *a espacio fijo* (D'Arco, Piegari, y Rivo, 2003). Cuando la medición se hace a tiempo fijo se define un dt y se cuenta el número de pulsos en ese intervalo de tiempo. Si la medida se hace a espacio fijo, entonces se mide el tiempo que hay entre cada par de pulsos (Liu, 2002). La medición a espacio fijo es recomendada cuando la velocidad es baja, mientras que la medición a tiempo fijo es útil cuando la velocidad es alta (Hachiya y Ohmae, 2007).

Un concepto importante, relacionado con dt y pul es la velocidad límite, ω_{lim} . Esta es la velocidad a la que el tiempo entre un impulso y el siguiente es exactamente dt , es decir, ω_{lim} es $(1/pul)/dt$ en rev/s. Definiciones similares pueden verse en Lygouras *et al.* (2008) y Tsuji *et al.* (2009), las cuales son utilizadas para definir lo que denominan la resolución de velocidad. Las velocidades menores a ω_{lim} se denominarán bajas, mientras que las superiores, altas. Ahora, con el propósito de exponer el algoritmo propuesto en este artículo, se asumirá que el *encoder* no pierde el conteo de pulsos y que la medida de tiempo se hace a múltiplos exactos del reloj del sistema, t_s , sin error en esa medida. Además, se asume que la distancia angular entre pulsos es igual, y para las demostraciones matemáticas se asume que la velocidad de giro es constante.

2. Algoritmos a tiempo fijo y a espacio fijo

En esta sección se detallan los dos algoritmos clásicos para aproximar la velocidad de giro, y se hace énfasis en el valor del error relativo máximo. Se comienza con el algoritmo a tiempo fijo. En este algoritmo se fija dt y se cuenta el número de impulsos, Nep , provenientes del *encoder* durante ese δ de tiempo. La velocidad aproximada es la relación entre el espacio recorrido por el número de impulsos contados y un tiempo de muestreo, dt , esto es ω_m es $Nep\omega_{lim}$. Si la velocidad es constante, y no es múltiplo entero de ω_{lim} , el número de impulsos contados varía, como se presenta en la figura 1, lo cual es el problema fundamental de este algoritmo. Durante el primer dt , en la figura 1, el conteo es uno, pero en el segundo es dos, y así sucesivamente, debido al residuo de desplazamiento que viene del δ de tiempo anterior.

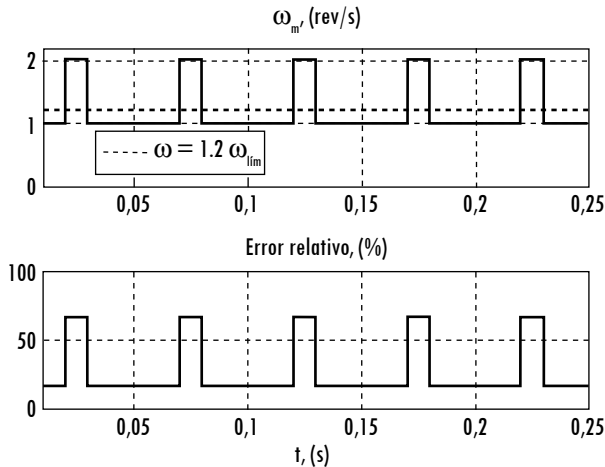
Figura 1. Velocidad constante y conteo de pulsos del *encoder*



Fuente: presentación propia de los autores.

La variación en el conteo de Nep hace que ω_m presente saltos, como se muestra en la figura 2, en la cual la $\omega = 1,2\omega_{lim}$. Como hay dos valores para la velocidad en la figura 2, entonces hay dos errores relativos. La variación ω_m obliga en la práctica a utilizar filtros para lograr una medida estable (Petrella y Tursini, 2008), lo cual retarda la salida del algoritmo, generando así un problema más en la aproximación.

Figura 2. Error relativo para el algoritmo a tiempo fijo. $Pul = 100, dt = 0,01$ s



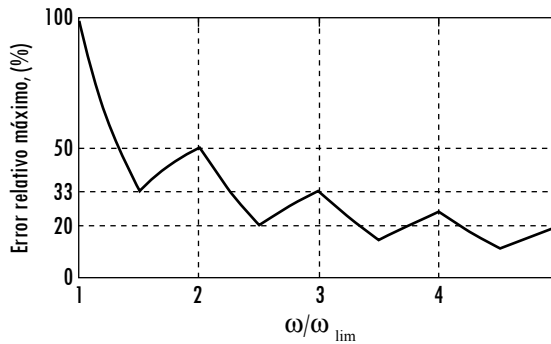
Fuente: presentación propia de los autores.

En la figura 3 se resume el comportamiento del algoritmo a tiempo fijo por medio del valor máximo del error relativo. Otra medida de rendimiento es el

error relativo máximo que se presenta entre cada múltiplo de ω/ω_{lim} , o también llamado en este artículo segmento de velocidad. Por ejemplo, para los valores entre 1 y 2 ese error es del 100%, mientras que es del 50% entre 2 y 3. Con esto en mente, puede verse que ese error máximo es inverso a la relación ω/ω_{lim} .

Por otra parte, en el algoritmo a espacio fijo se mide el tiempo entre un par de pulsos del *encoder* consecutivos, como se explica en Merry, Molengraft y Steinbuch (2007). Así, ω_m es igual a ω_{lim}/Ndt , donde Ndt es el conteo de instantes de tiempo dt .

Figura 3. Error relativo máximo para el algoritmo a tiempo fijo.



Fuente: presentación propia de los autores.

El efecto del residuo de posición en el algoritmo a tiempo fijo también aparece en espacio fijo, pero esta vez el residuo es de tiempo. Así, el error de truncamiento genera dos valores de velocidad, incluso cuando la velocidad es constante, lo cual ya se definió como el problema fundamental de los algoritmos tradicionales.

Una propuesta que se está haciendo popular para la solución del problema de truncamiento es la utilización de circuitos dedicados, los cuales pueden medir diferencias de tiempo, incluso en el orden de los ns, como en Boggarpu y Kavanagh (2010). Así, trabajos como los de Petrella y Tursini (2008) dicen eliminar el error de truncamiento. La propuesta en Lygouras *et al.* (2008) reconoce que el problema del residuo es difícil de atacar, y prefieren hacer una medida de tiempo tan precisa que la oscilación debida al truncamiento es despreciable. El trabajo en Negrea *et al.* (2012) ataca el problema de truncamiento por medio de la definición de un tiempo dt variable, de manera que siempre coincida con un pulso de posición. Sin embargo, el artículo no dice cómo puede medirse ese tiempo, por lo cual los resultados se quedan en idealizaciones de la situación real. El trabajo en Su *et al.* (2005) también reconoce el problema del truncamiento, pero se enfoca en la aproximación de velocidades bajas. Un inconveniente del

algoritmo propuesto en ese artículo es que requiere cerca de dos segundos para estabilizar la salida. Por último, en Tsuji *et al.* (2009) atacan el problema de truncamiento, sincronizando la medida de tiempo y espacio, de manera similar a lo que se propone en este artículo, pero su objetivo es aumentar la resolución de la medida, y no reducir o acotar el error.

3. Presentación del algoritmo de aproximación nuevo

Con base en la identificación del problema de inestabilidad en la medida, en esta sección se plantea un algoritmo nuevo. El corazón del algoritmo está en la sincronización de los impulsos que indican posición y tiempo. El objetivo fundamental es eliminar el error de truncamiento, y en su lugar tener una medida de velocidad estable, sin necesidad de filtros.

El único tratamiento que se le hace a la señal proveniente del *encoder* es la detección de los flancos de subida, con lo que se genera el tren de impulsos denominado I_{ep} , como se muestra en la figura 4. Si la velocidad es constante, la separación entre impulsos I_{ep} es constante. Los pulsos son contados por medio de un acumulador, lo cual genera la señal C_{ep} . Cuando ese conteo se va a 0, porque un evento hasta ahora no descrito reinicia el acumulador, el bloque retenedor conserva el último conteo y así se ha contado el número de pulsos, N_{ep} .

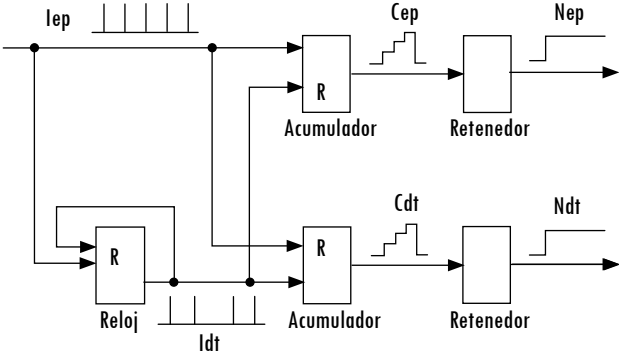
El esquema en la figura 4 presenta además una serie de bloques en cascada en la parte inferior; estos son los encargados de contar el número de instantes de tiempo, dt . Al comienzo del programa se inicia el reloj, y cada vez que el reloj llega a dt aparece un impulso, I_{dt} , el cual es acumulado por un integrador en la variable C_{dt} . Cuando un impulso I_{ep} reinicia el acumulador, entonces el retenedor almacena el último valor en C_{dt} . Así se ha contado el número de impulsos dt en la variable N_{dt} .

La velocidad que entrega el algoritmo es $\omega_m = (N_{ep}/N_{dt})\omega_{lim} = n\omega_{lim}$. Por la naturaleza del conteo, N_{ep} puede ser cualquier número completo, mientras que N_{dt} es un número natural. Si la velocidad de giro es superior o igual a ω_{lim} , entonces hay más impulsos I_{ep} que I_{dt} , y N_{dt} es 1. A su vez, si la velocidad de giro es inferior a ω_{lim} , y por tanto hay más pulsos I_{dt} que I_{ep} , N_{ep} es 1. Lo descrito hasta el momento es similar a los algoritmos a espacio o a tiempo fijo, la diferencia fundamental está en que los impulsos I_{dt} e I_{ep} están sincronizados.

La figura 5 presenta un ejemplo en el cual la velocidad que se está midiendo es $1,5\omega_{lim}$. Mientras que el primer pulso I_{ep} puede aparecer en cualquier momento, el reloj es iniciado con el programa, como indica el evento (a), según se indica en el eje x de la figura 5. El conteo de tiempo en el reloj se reinicia para

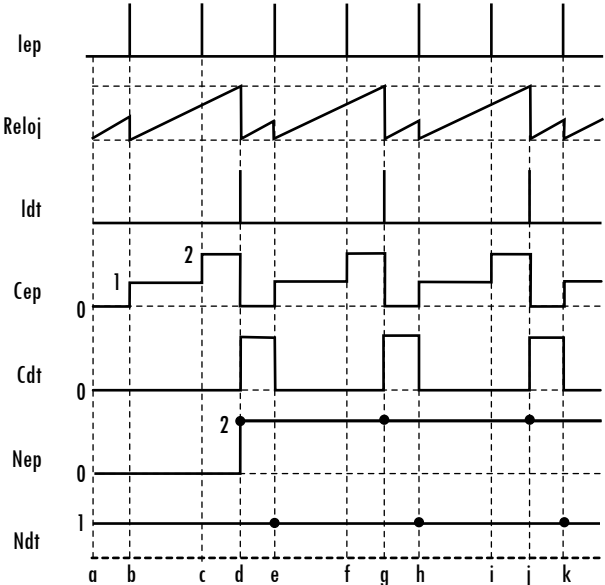
que ese conteo coincida con la aparición de un impulso I_{ep} , de esta manera la misma cantidad de pulsos I_{ep} será contada en el mismo tiempo, y se elimina la influencia de la incertidumbre por la aparición aleatoria del primer pulso I_{ep} . Si la velocidad es mayor que la velocidad límite, de igual manera se reinicia el conteo de I_{dt} , y el resultado es otra vez que el conteo es independiente de la aparición aleatoria del primer pulso I_{ep} .

Figura 4. Diagrama de bloques para el algoritmo propuesto



Fuente: presentación propia de los autores.

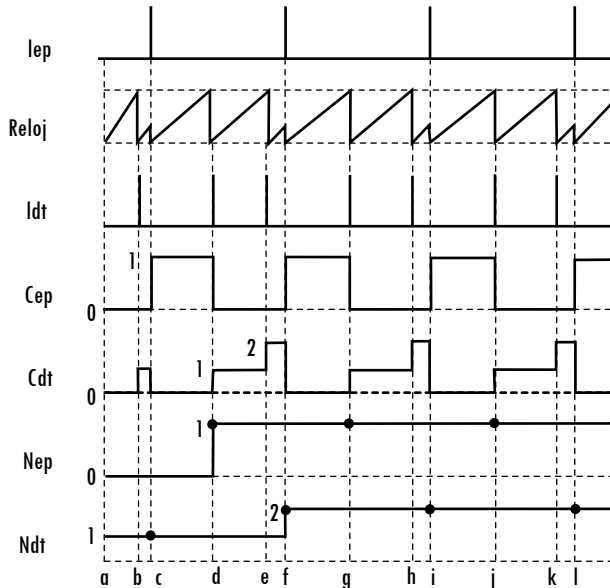
Figura 5. Señales cuando $w = 1,5 w_{lím}$. El eje X es tiempo



Fuente: presentación propia de los autores.

En la figura 6 se presenta otro ejemplo con $w = 0,25\omega_{lim}$. Como la velocidad es lenta, comparada con ω_{lim} , aparecen más impulsos *Idt* que *Iep*. Se supondrá, como ejemplo, que el primer impulso *Iep* aparece luego de que el reloj ya ha contado un impulso *Idt*.

Figura 6. Señales cuando $w = 0,25 w_{lim}$. El eje X es tiempo



Fuente: presentación propia de los autores.

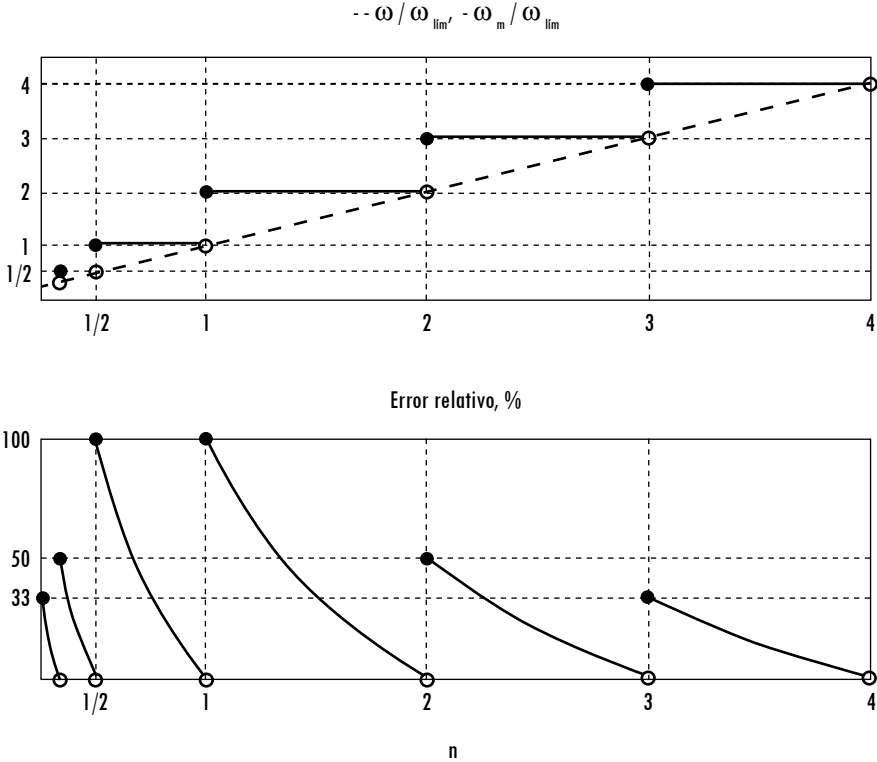
4. Cálculo del error relativo

La ecuación 1 indica que la aproximación de la velocidad se da a escalones, los cuales aumentan de a uno si $\omega \geq \omega_{lim}$, y disminuyen con el inverso de *Ndt* si $\omega < \omega_{lim}$. Esto significa que el error relativo puede ser definido para cada valor de velocidad, según se presenta en la figura 7.

$$\omega_m = \frac{Nep}{Ndt} \omega_{lim} \quad (rev/s) \tag{1}$$

Cada segmento de velocidad en la parte superior de la figura 7 tiene dos extremos, uno cerrado al lado izquierdo y otro abierto al lado derecho. En la figura 7, la línea punteada representa la velocidad real, ω .

Figura 7. Primera aproximación de la velocidad y error relativo



Fuente: presentación propia de los autores.

El error relativo se presenta en la parte inferior de la figura 7. Ahora, como medida del algoritmo se observará el error relativo máximo en cada segmento de velocidad. Por ejemplo, el error relativo máximo es 100% para velocidades entre $\omega_{lim} \leq \omega < 2\omega_{lim}$. En la figura 7 puede verse que ese error relativo máximo es el mismo para $\frac{1}{2}\omega_{lim} < \omega < \omega_{lim}$. Téngase en cuenta que el error nunca es superior a 100%. Las velocidades con valor de n dos, es decir $2\omega_{lim} \leq \omega < 3\omega_{lim}$, tienen un error relativo máximo del 50%. En la ecuación 2 se encuentra la expresión que define el error relativo máximo.

$$Er_{M\acute{a}x} = \begin{cases} \left\lceil \frac{1}{\frac{\omega_{lim}}{\omega}} \right\rceil \times 100\% & , \omega < \omega_{lim} \\ \left\lceil \frac{1}{\frac{\omega}{\omega_{lim}}} \right\rceil \times 100\% & , \omega \geq \omega_{lim} \end{cases} \quad (2)$$

El error relativo máximo, $Er_{Máx}$, para $\omega < \omega_{lim}$ resulta ser igual al error relativo máximo con el algoritmo tradicional a tiempo fijo. De igual manera, $Er_{Máx}$, para $\omega \geq \omega_{lim}$ es igual al error relativo máximo para el algoritmo a espacio fijo. Esto quiere decir que los tres algoritmos, espacio y tiempo fijo, más el propuesto aquí, son equivalentes en cuanto al error relativo máximo.

5. Reducción del error relativo máximo

En esta sección se lleva el algoritmo de aproximación a su punto óptimo. Se dirá que el algoritmo es óptimo si presenta el $Er_{Máx}$ más bajo posible. La minimización del $Er_{Máx}$ comienza al tener en cuenta que ω_m es mayor que la velocidad real. Esto hace que el error máximo esté a la izquierda de cada segmento, como se observa en la figura 7. Ahora, dado que la velocidad real más pequeña en cada segmento también está a la izquierda, entonces el error relativo máximo siempre está a la izquierda de cada segmento. El $Er_{Máx}$ se reduce si en lugar de tener la diferencia máxima a la izquierda de cada segmento, se tiene a la derecha. Si bien el error absoluto es el mismo, este se presenta a una velocidad mayor, y por tanto $Er_{Máx}$ se reduce. Esto es equivalente a bajar cada uno de los escalones de ω_m , lo cual se logra mediante la definición de ω_{m2} .

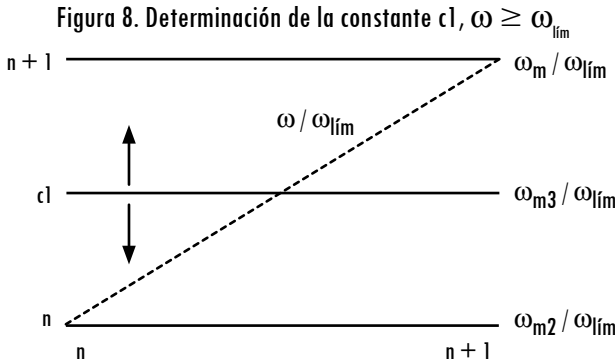
$$\omega_{m2} = \begin{cases} \frac{1}{\left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} \omega_{lim} & , \omega < \omega_{lim} \\ \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor \omega_{lim} & , \omega \geq \omega_{lim} \end{cases} \tag{3}$$

La velocidad ω_{m2} en la ecuación 3 siempre es inferior o igual a la velocidad real. Con ω_{m2} ya definida puede calcularse el error relativo, y a partir de ese error, puede encontrarse el error máximo en cada segmento, lo cual resulta en la expresión en la ecuación 4.

$$Er2_{sup} = \begin{cases} \frac{1}{\left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} \times 100\% & , \omega < \omega_{lim} \\ \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor \times 100\% & , \omega \geq \omega_{lim} \end{cases} \tag{4}$$

Véase que el error relativo pasa de 100%, con ω_{m1} , a 50% con ω_{m2} , cuando $\omega_{lim} \leq w < 2\omega_{lim}$. La siguiente modificación al algoritmo, en busca de la optimización del error relativo máximo, proviene de la observación de los escalones que se generan al aproximar la velocidad. Con la propuesta original, los pasos siempre están por encima de la velocidad real. Con la modificación, la aproximación siempre está por debajo de la velocidad real. Cualquier aproximación que esté por encima de ω_m resultará en errores relativos mayores al $Er_{Máx}$, mientras que cualquier aproximación que esté por debajo de ω_{m2} resultará en errores relativos mayores a $Er_{2_{sup}}$. Así, la única forma de encontrar un error relativo menor al de los dos errores relativos ya calculados es si la velocidad aproximada está en algún punto intermedio entre estas dos, es decir, algún tipo de promedio.

La pregunta ahora es cuál es el valor intermedio ω_{m3} que minimiza el error relativo, como se presenta mediante la figura 8, donde $c1$ es un número real entre 0 y 1, y n es el piso de ω/ω_{lim} .



Fuente: presentación propia de los autores.

El error relativo máximo por el lado izquierdo está en la ecuación 5, mientras que el error relativo superior, por el extremo derecho, es el que se presenta en la ecuación 6.

$$Er3_l = \frac{c1 - n}{n} 100\% \tag{5}$$

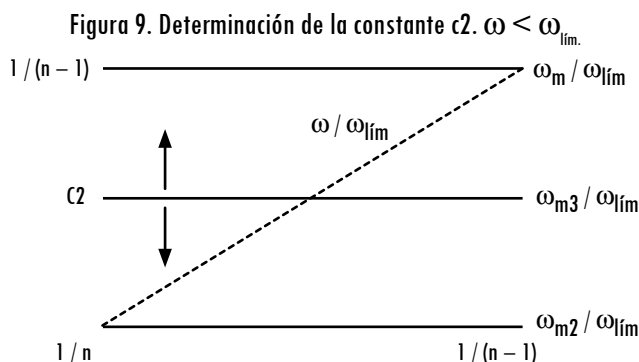
$$Er3_d = \frac{(n + 1) - c1}{n + 1} 100\% \tag{6}$$

La observación del error relativo por la derecha $Er3_d$ y por la izquierda, $Er3_l$, permite ver que si uno de ellos aumenta, el otro disminuye. Además, el mínimo

se encuentra en $Er3_I$ igual a $Er3_D$. Cualquier otro valor hace que uno de los dos errores sea mayor. Al igualar las ecuaciones 5 y 6, resulta el valor de $c1$, como se presenta en la ecuación 7.

$$c1 = \frac{2n(n+1)}{2n+1}, \omega \geq \omega_{lim} \tag{7}$$

En el caso de las velocidades bajas se calcula el valor de una constante $c2$, según se muestra en la figura 9.



Fuente: presentación propia de los autores.

Si se sigue el procedimiento que se realizó para $\omega \geq \omega_{lim}$, se encuentra que $c2$ es la constante en la ecuación 8.

$$c2 = \frac{2}{2n-1}, \omega < \omega_{lim} \tag{8}$$

Dadas las constantes $c1$ y $c2$ en las ecuaciones 7 y 8, es posible definir la velocidad aproximada nueva, ω_{m3} , como se indica en la ecuación 9. Por definición, esta velocidad siempre es menor que ω_m y mayor que ω_{m2} .

$$\omega_{m3} = \begin{cases} \frac{2}{2n-1} \omega_{lim} & \omega < \omega_{lim} \\ \frac{2n(n+1)}{2n+1} \omega_{lim} & \omega \geq \omega_{lim} \end{cases} \tag{9}$$

A continuación se muestra que ω_{m3} es el promedio armónico entre las aproximaciones ω_m y ω_{m2} . En las ecuaciones 10 y 11 se define y calcula el promedio armónico, indicado como ω_H . Primero para velocidades bajas.

$$\omega_H = \frac{2}{\frac{1}{\omega_m} + \frac{1}{\omega_{m2}}} \left| \begin{array}{l} \omega_m = \frac{1}{n-1} \omega_{lim} \\ \omega_{m2} = \frac{1}{n} \omega_{lim} \end{array} \right. \quad \omega_H = \frac{2}{2n-1} \omega_{lim} = \omega_{m3} \quad (10)$$

Y ahora para velocidades altas.

$$\omega_H = \frac{2}{\frac{1}{\omega_m} + \frac{1}{\omega_{m2}}} \left| \begin{array}{l} \omega_m = (n+1)\omega_{lim} \\ \omega_{m2} = n\omega_{lim} \end{array} \right. \quad \omega_H = \frac{2n(n+1)}{2n+1} \omega_{lim} = \omega_{m3} \quad \omega \geq \omega_{lim} \quad (11)$$

Cualquier promedio diferente al armónico resulta en errores relativos máximos más grandes. El error relativo máximo por segmentos está definido en la ecuación 12. Cuando ω es cercana a ω_{lim} , el error máximo es (100/3)%, lo cual representa una reducción a 1/3 del $Er_{m\acute{a}x}$, y a 2/3 de Er_{sup}^2 .

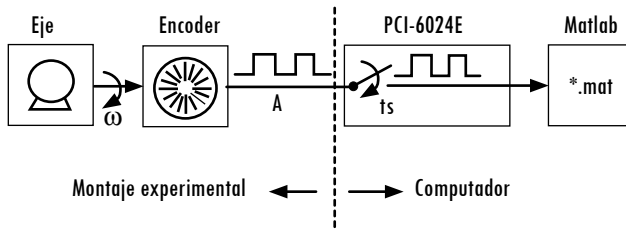
$$Er3_{M\acute{a}x} = \begin{cases} \frac{1}{2 \left[\frac{\omega_{lim}}{\omega} \right] + 1} \times 100\% & , \omega < \omega_{lim} \\ \frac{1}{2 \left[\frac{\omega}{\omega_{lim}} \right] + 1} \times 100\% & , \omega \geq \omega_{lim} \end{cases} \quad (12)$$

6. Resultados experimentales

A continuación se presenta el resultado de la aplicación del algoritmo utilizando señales de un *encoder* real, para lo cual se sigue el esquema en la figura 10. El *encoder* es óptico, de 160 ranuras, y se acopla al eje de un motor de corriente continua, con 24 V de voltaje nominal, con referencia Minertia 6GFMED. La tarjeta de adquisición de datos es una PCI-6024E de National Instruments, el tiempo de muestreo es de 10 ms.

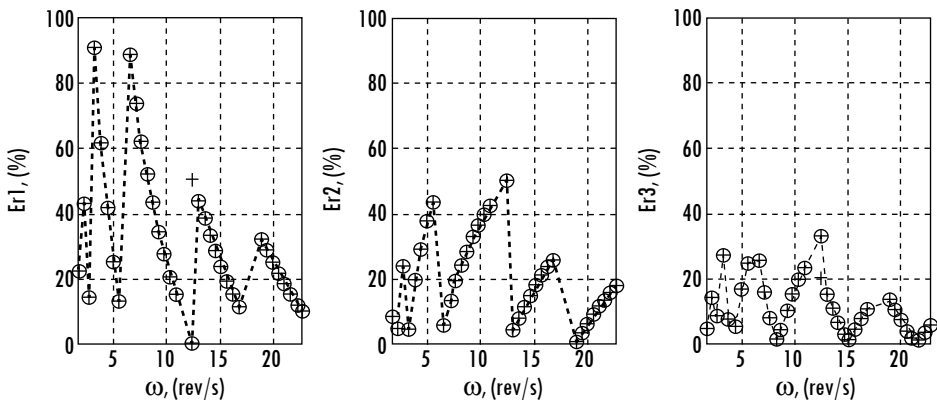
El primer experimento define pul como 160 y dt como 1 ms. Primero se fija el voltaje de alimentación del motor, y se captura la señal del *encoder* por un intervalo de 10 s. Esto se repite para voltajes entre 3 y 24 V, cada 0,5 V, y para cada voltaje de alimentación del motor se calcula el error relativo. El error relativo teórico se indica con la línea punteada en la figura 11. El error relativo experimental es indicado por un círculo como marcador.

Figura 10. Montaje experimental y procesamiento de las señales



Fuente: presentación propia de los autores.

Figura 11. Error relativo teórico y experimental, $dt = 0,001$ s



Fuente: presentación propia de los autores.

Ahora, con el fin de realizar una comparación adicional, se implementa el algoritmo llamado S (del inglés para método de sincronización de la medida), el cual busca aumentar la resolución de medida que se presenta para el algoritmo a espacio fijo. El algoritmo S, propuesto en Tsuji *et al.* (2009), requiere tres pasos: 1. contar N_p cada instante dt , 2. no actualizar la velocidad si N_p permanece constante, 3. actualizar la aproximación si N_p cambia. El valor de la aproximación se calcula como indica la ecuación 13. Para llegar al resultado en la figura 11 para el algoritmo S, el error relativo fue calculado como el máximo error relativo producido durante 2 s. Si bien ese error puede ser pequeño, el promedio, dada la sumatoria, hace que la aproximación dependa de instantes anteriores. Así, si bien se suaviza la aproximación, el error máximo resulta similar al error con el algoritmo a tiempo fijo.

$$\omega_s = \frac{\sum_{j=0}^{Ndt-1} Np(i-j)}{Ndt} \omega_{lim} \tag{13}$$

El error relativo con el algoritmo propuesto aquí se comporta como se espera, lo cual puede verse en que el error relativo teórico y el experimental coinciden.

Además del experimento a $dt = 1$ ms, se realizaron otros a 2, 5, 10, 20, 50 y 100 ms. Dado que se aumenta dt , la velocidad límite disminuye, e incluso está por debajo de todas las velocidades medidas en algunos casos, lo que hace que el error relativo disminuya. De igual manera, también es posible hacer que el error disminuya si todas las velocidades están por debajo de ω_{lim} .

7. Conclusiones

En este trabajo se presenta un algoritmo para la aproximación de la velocidad, el cual indica un valor de velocidad único cuando esta es constante, en contraposición con los algoritmos tradicionales que generan dos valores. Esto se logra eliminando el error de truncamiento de espacio y tiempo, mediante la sincronización de los impulsos provenientes de las medidas de espacio y de tiempo.

Se realiza el análisis teórico del error de estado estacionario de la propuesta, comparada con los algoritmos tradicionales, y se llega a que el error relativo máximo por cada segmento de velocidad es el mismo. Recuérdese que un segmento es el rango de velocidades que hay entre un múltiplo de la velocidad límite y otro.

Se propone, además, una modificación que reduce el error relativo máximo por segmento al máximo posible. Esto es por medio del cálculo del promedio entre una aproximación que es siempre mayor o igual que la velocidad real y otra que en todos los casos es menor o igual. Se observa que el error relativo máximo siempre es menor que la mitad de aquel que se encuentra con los algoritmos tradicionales.

Si bien es posible garantizar los resultados del algoritmo para velocidades constantes, una vez la aproximación ya ha alcanzado un estado estacionario, este trabajo no cubre el caso cuando las velocidades son variables. El método de sincronización de señales utilizado aquí hace que la aproximación sea aceptable solo bajo una aceleración máxima. En este sentido, un trabajo futuro debe enfocarse en determinar esa aceleración máxima, con el fin de definir si son necesarias modificaciones adicionales para utilizar el algoritmo en aplicaciones industriales, o cuando menos conocer los límites de aplicación del algoritmo.

Por último, un problema abierto, el cual resulta de la naturaleza del algoritmo propuesto aquí, es la interpolación de la salida del algoritmo. Así, en lugar de que la aproximación se dé a escalones fijos, esta aproximación contaría con cambios suaves, los cuales podrían, por ejemplo, anticipar los cambios que pudiera tener la velocidad real.

Referencias

- BOGGARPU, N. y KAVANAGH, R. New Learning Algorithm for High-Quality Velocity Measurement and Control When Using Low-Cost Optical Encoders. *IEEE Transactions on instrumentation and measurement*. 2010, vol. 59, núm. 3, pp. 565-574.
- D'ARCO, S.; PIEGARI, L. y RIVO, R. An Optimized Algorithm for Velocity Estimation Method for Motor Drives. *Symposium on Diagnostics for Electric Machines, Power Electronics and Drivers*. Atlanta, GA, USA, 2003, pp. 76-80.
- HACHIYA, K. y OHMAE, T. Digital speed control system for a motor using two speed detection methods of an incremental encoder. *Power Electronics and Applications, 2007 European Conference*. Aalborg, Dinamarca, 2007, pp. 1-10.
- HOANG H. y JEON J. An efficient approach to correct the signals and generate high-resolution quadrature pulses for magnetic encoders. *IEEE Transactions on industrial electronics*. 2011, vol. 58, núm. 8, pp. 3634-3646.
- LIU, G. On velocity estimation using position measurements. *American Control Conference, 2002. Proceedings of the 2002*. 2002, vol. 2, pp. 1115-1120.
- LYGOURAS, J.; PACHIDIS, T.; TARCHANIDIS, K., et al. Adaptive High-Performance Velocity Evaluation Based on a High-Resolution Time-to-Digital Converter. *IEEE Transactions on Instrumentation and measurement*. 2008, vol. 57, núm. 9, pp. 2035-2043.
- MERRY, R.; MOLENGRAFT, R. y STEINBUCH, M. Error modeling and improved position estimation for optical incremental encoders by means of time stamping. *Proceedings of the 2007 American Control Conference*. New York City, USA, julio 11-13, 2007, pp. 3570-3575.
- NANDAYAPA, M.; MITSANTISUK C.; y OHISHI K. High resolution position estimation for advanced motion control based on FPGA. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*. 2012, pp. 3808-3813.
- NEGREA, C.; INCZE, I.; IMECS, M., et al. An improved speed identification method using incremental encoder in electric drives. *IEEE International Conference on Automation Quality and Testing Robotics (AQTR)*. 2012, pp. 536-540.
- PETRELLA, R. y TURSINI, M. An Embedded System for Position and Speed Measurement Adopting Incremental Encoders. *IEEE Transactions on industry applications*. 2008, vol. 44, núm. 5, pp. 1436-1444.
- SU, Y.; ZHENG, C.; SUN, D., et al. A Simple Nonlinear Velocity Estimator for High-Performance Motion Control. *IEEE Transactions on Industrial Electronics*. 2005, vol. 52, núm. 4, pp. 1161-1169.
- TSUJI, T.; HASHIMOTO, T.; KOBAYASHI, H., et al. Wide-Range Velocity Measurement Method for Motion Control. *IEEE Transactions on industrial electronics*. 2009, vol. 56, núm. 2, pp. 510-519.
- VOLDÁN, J. Use encoder speed sensors for the calculation of torque moment. *13th International Carpathian Control Conference (ICCC)*. 2012, pp. 768-771.