

Algoritmo para la aproximación de la velocidad de giro de un eje mediante un encoder incremental

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 delta 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.

Abstract

The proposed algorithm in this paper approximates angular speed by means of the signals coming from an incremental encoder. This algorithm removes the oscillations that appear, even for constant speeds, when using traditional fix-step or fix-time approximation algorithms. The key idea for eliminating oscillations lies in the synchronization of a signal every delta time and the pulses from the encoder. The most notable result in this paper is that the harmonic mean between the output of the proposed algorithm and the output of a first modification of the same algorithm has the smallest relative error possible. The relative error of this new algorithm is always smaller than half the relative error using any of the two traditional algorithms, which indicates optimality. At the end of the paper, and using pulses coming from an encoder, some tests are run on Simulink of Matlab. These tests provide experimental proof of the performance of the algorithms.

Palabras clave:

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

Key Words:

Velocity approximation, incremental encoder, fix-space, harmonic mean, fix-time

I. Introducción

Los encoders incrementales son la opción más popular para medir la velocidad en aplicaciones industriales, dado su costo bajo 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, Jeon, 2011), donde un algoritmo optimiza la estimación cuando la salida es senoidal. 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, usando la superposición de medidas distanciadas cada delta de tiempo. De otra parte, en cuanto al área 3) los encoders son usados en robótica, control de movimiento, radares, vehículos, CNC, elevadores, entre muchas otras

Eliminado: En este trabajo se presenta un algoritmo para la aproximación de la velocidad de giro en un eje, mediante el tren de pulsos de un encoder incremental. El algoritmo elimina las oscilaciones que generan los algoritmos conocidos como espacio fijo o tiempo fijo, incluso cuando la velocidad es constante, y en lugar de ello entrega un valor único, gracias a la sincronización de una señal que mide deltas de tiempo con la aparición de los impulsos provenientes del encoder. Además se propone una modificación a esta primera propuesta, y se concluye que el promedio armónico entre las dos alternativas presenta el menor error relativo posible, y que este siempre es menor que la mitad de lo que puede lograrse con los dos algoritmos tradicionales. Al final del artículo se presentan resultados experimentales, los cuales consisten en implementar el algoritmo en Simulink y adquirir datos reales de un encoder para aproximar la velocidad de giro.¶

Eliminado: In

Eliminado: we present an algorithm to approximate the

Eliminado: velocity

Eliminado: impulses

Eliminado: The

Eliminado: show up

Eliminado: if the speed is

Eliminado: , for

Eliminado: and

Eliminado: besides this a unique value is generated due to synchronization between an impulse signal every delta time and encoder pulses. We propose a first modification of the algorithm and it is concluded

Eliminado: those alternatives

Eliminado: In addition it is shown that this

Eliminado: with

Eliminado: we present experimental results which are the implementation of our proposal in Simulink

Eliminado: the acquisition of true signals

Eliminado: to approximate a velocity

aplicaciones. Por ejemplo en el trabajo en (Voldán, 2012) se utiliza la estimación de la velocidad para calcular aceleración, y de ahí torque, lo cual puede ser utilizado en el control de motores. Finalmente, con respecto al área 4), por ahora se reporta un trabajo, aunque a lo largo del artículo se encuentran más. En (Negrea et. al., 2012) un algoritmo mejora los algoritmos tradicionales, por medio del aumento del 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 que se supondrán iguales. La parte electrónica se encarga de generar una luz, la cual es interpretada por un receptor, como uno cuando la luz pasa la ranura y llega al receptor y como cero en caso contrario.

La aproximación de la velocidad requiere de dos mediciones: una de posición y una de tiempo. El error en la aproximación tiene dos fuentes: la primera es que en la práctica sólo 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 deltas, lo cual conlleva a otro error. El delta 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 delta de tiempo, Δt (el cual se llamará dt en adelante), es un múltiplo entero del reloj del sistema, ts , 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 tiempo fijo es útil cuando la velocidad es alta (Hachiya y Ohmae, 2007).

Un concepto importante, relacionado con dt y pul es la que se denominará velocidad límite, ω_{lim} . Esta es la velocidad a la que el tiempo entre un impulso y el siguiente es exactamente dt , es decir, $\omega_{lim} = (1/pul)/dt$ 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 ningún pulso y que la medida de tiempo se hace a múltiplos exactos del reloj del sistema, ts , sin error en esa medida. Además se asume la distancia angular entre pulsos es igual, y para las demostraciones matemáticas se asume que la velocidad de giro es constante.

Eliminado: se

Eliminado: hacer

Con formato: Color de fuente: Automático

II. 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 delta 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 $\omega_m = Nep \cdot \omega_{lim}$.

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 delta de tiempo anterior.

Eliminado: anterior

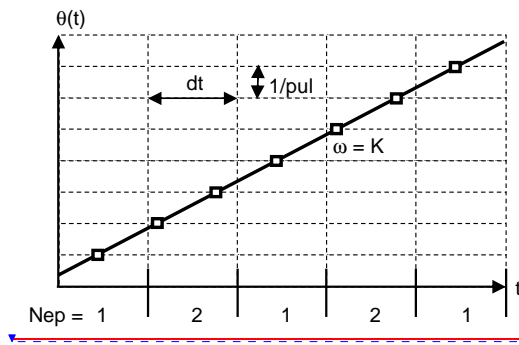


FIG 1. VELOCIDAD CONSTANTE Y CONTEO DE PULSOS DEL ENCODER. FUENTE: AUTORES DEL PROYECTO.

La variación en el conteo de Nep hace que ω_m presente saltos, como se muestra en la figura 2, en la cual la $\omega = 1,2\omega_{lim}$. Esos saltos no son deseables. En la figura se observa una velocidad superior, igual a 2 rev/s y una inferior, en 1 rev/s. Como hay dos valores para la velocidad 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.

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 100%, mientras que es 50% entre 2 y 3. Con esto en mente puede verse que ese error máximo es inverso a la relación ω/ω_{lim} .

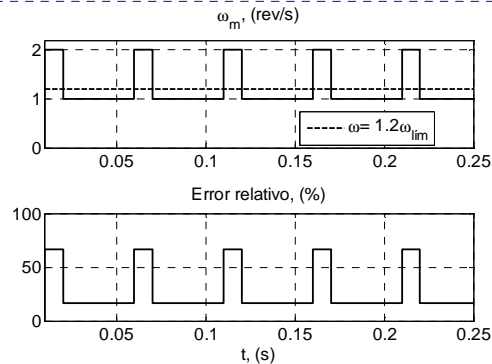


FIG 2. ERROR RELATIVO PARA EL ALGORITMO A TIEMPO FIJO. PUL = 100, DT = 0,01 s. FUENTE: AUTORES DEL PROYECTO.

De 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 $\omega_m = (1/Ndt) \cdot \omega_{lim}$, donde Ndt es el conteo de instantes de tiempo dt .

Eliminado: <sp>¶

Eliminado: velocidad real es

Eliminado: Se

Eliminado:)

Eliminado: busca resumir

Eliminado: posible

Eliminado: del algoritmo

Eliminado: De otra parte, en el algoritmo a espacio fijo se mide el tiempo entre un par de pulsos del encoder consecutivos (Merry, Molengraft y Steinbuch, 2007). Así $\omega_m = (1/Ndt) \cdot \omega_{lim}$, donde Ndt es el conteo de instantes de tiempo dt .

Eliminado: El efecto del residuo de posición que se vio en el caso del algoritmo a tiempo fijo también aparece aquí, pero esta vez es de tiempo y no de posición. Así el error de truncamiento genera dos valores de velocidad, incluso cuando la velocidad es constante. Una propuesta que se está haciendo popular para la solución de este problema es la utilización de circuitos dedicados, como procesadores digitales de señales, los cuales pueden medir diferencias de tiempo, incluso en el orden de los ns, como en (Boggarpu y Kavanagh, 2010); sin embargo, en este artículo se utilizará la sincronización de los impulsos del encoder y de la medida de tiempo cada dt .

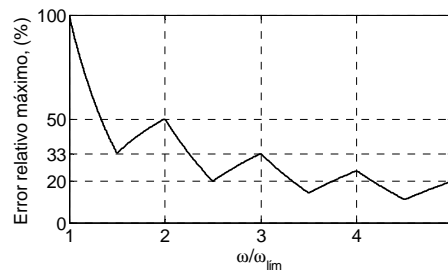


FIG 3. ERROR RELATIVO MÁXIMO PARA EL ALGORITMO A TIEMPO FIJO. FUENTE: AUTORES DEL PROYECTO.

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 y no de posición. 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, como procesadores digitales de señales, los cuales pueden medir diferencias de tiempo, incluso en el orden de los ns, como en (Boggarpu y Kavanagh, 2010). Así, trabajos como (Petrella y Tursini, 2008) dicen eliminar el error de truncamiento; sin embargo, en el caso de este artículo esa medición resulta dudosa, porque es hecha en un microcontrolador. 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. Resulta interesante ver que la medición precisa del tiempo genere problemas nuevos, como el cambio de la salida del algoritmo en función de la temperatura en el FPGA. 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, basados en medidas de posición. Un inconveniente del algoritmo propuesto en ese artículo es que requiere cerca de dos segundos para estabilizar la salida. Por último, los autores 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.

III. Presentación del algoritmo de aproximación nuevo

Con base en la identificación del problema de inestabilidad en la medida con los algoritmos tradicionales, 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, observadores de estado o del modelo del sistema bajo medición.

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. El número de pulsos es acumulado por medio de un acumulador, y eso genera una señal tipo escalera, C_{ep} . Cuando ese conteo se va a cero, 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} .

Eliminado: (tiempo fijo y espacio fijo)

Eliminado: asegurar la eliminación d

Eliminado: ya explicado

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 que el reloj llega a dt aparece un impulso, Idt , el cual es acumulado por un integrador en la variable Cdt . Cuando un impulso Iep reinicia el acumulador, entonces el retenedor almacena el último valor en Cdt . Así se han contado el número de impulsos dt en la variable Ndt .

La velocidad que entrega el algoritmo es $\omega_m = (Nep/Ndt) \cdot \omega_{lim} = n \cdot \omega_{lim}$. Por la naturaleza del conteo, Nep puede ser cualquier número completo, mientras que Ndt es un número natural. Si la velocidad de giro es superior o igual a ω_{lim} , entonces hay más impulsos Iep que Idt , y Ndt es uno. A su vez, si la velocidad de giro es inferior a ω_{lim} , y por tanto hay más pulsos Idt que Iep , Nep es uno.

Lo descrito hasta el momento es similar a los algoritmos a espacio o tiempo fijo, la diferencia fundamental está en que los impulsos Idt e Iep están sincronizados.

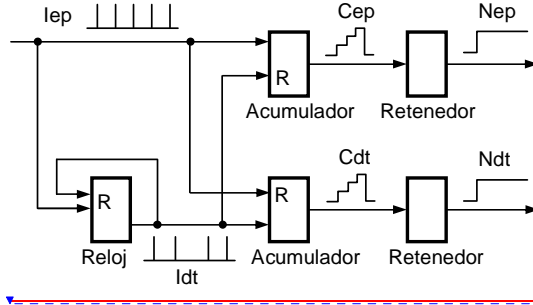


FIG 4. DIAGRAMA DE BLOQUES PARA EL ALGORITMO PROPUESTO. FUENTE: AUTORES DEL PROYECTO.

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 Iep 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. Los eventos que siguen, en orden cronológico son:

- (b) aparece el primer pulso Iep
- (b) se reinicia el reloj y aumenta el valor de Cep de 0 a 1
- (b) el reloj seguirá su curso hasta llegar a dt .
- (c) el segundo pulso Iep hace que Cep pase de 1 a 2
- (d) el reloj llega a dt y genera un impulso Idt
- (d) Idt reinicia Cep y pasa Cdt de 0 a 1
- (d) Nep toma el valor anterior de Cep , 2
- (d) el reinicio de Cep hace que Nep pase de 0 a 2
- (e) el tercer Iep reinicia el reloj y manda Cdt a 0
- (e) el reinicio de Cdt hace que Ndt tome el valor de Cdt

Este procedimiento se repetirá indefinidamente.

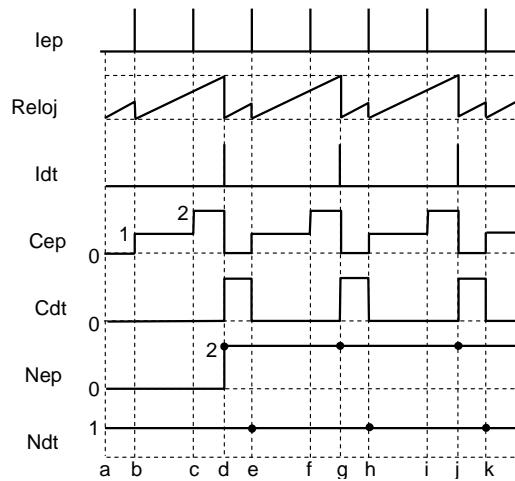
El conteo de tiempo en el reloj se reinicia para que ese conteo coincida con la aparición de un impulso Iep , de esta manera la misma cantidad de pulsos Iep será contada en el mismo tiempo, y se elimina la influencia de la incertidumbre por la aparición aleatoria del primer pulso Iep con respecto al tiempo inicial de ejecución del algoritmo. Si la velocidad es mayor que la velocidad límite, de igual manera se reinicia el conteo de Idt , y el resultado es otra vez, que el conteo es independiente de la aparición aleatoria del primer pulso Iep .

Eliminado: habrá

Eliminado: siempre será

Eliminado: siempre será

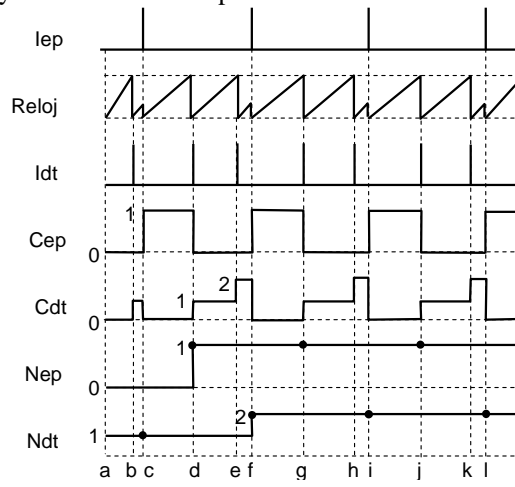
Eliminado: <sp>¶



Eliminado: <sp>¶

FIG 5. SEÑALES CUANDO $\omega = 1,5\omega_{lim}$. EL EJE X ES TIEMPO. FUENTE: AUTORES DEL PROYECTO.

En la figura 6 se presenta otro ejemplo con $\omega = 0,25\omega_{lim}$. 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.



Eliminado: <sp>¶

FIG 6. SEÑALES CUANDO $\omega = 0,25\omega_{lim}$. EL EJE X ES TIEMPO. FUENTE: AUTORES DEL PROYECTO.

La secuencia de eventos es la siguiente:

- (a) se inicia el reloj, al inicio del algoritmo
- (b) el reloj llega a dt y genera un impulso Idt
- (b) Idt hace que Cdt pase de 0 a 1
- (c) llega el primer pulso Iep
- (c) Iep reinicia el reloj, aumenta Cep y reinicia Cdt
- (c) el reinicio de Cdt fija Ndt en 1
- (d) el reloj llega a dt y genera un impulso Idt
- (d) Idt reinicia Cep y aumenta Cdt
- (d) el reinicio de Cep fija Nep en el anterior Cep, 1
- (e) el reloj llega a dt otra vez, y genera un impulso Idt

- (e) Cdt pasa de 1 a 2
 - (f) un impulso Iep aparece y reinicia el reloj
 - (f) Cep pasa de 0 a 1, y se reinicia Cdt
 - (f) el reinicio de Cdt fija Ndt en el anterior Cdt, 2
- Este procedimiento se repetirá indefinidamente.

IV. 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.

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

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 esa misma figura, la línea punteada representa la velocidad real, ω .

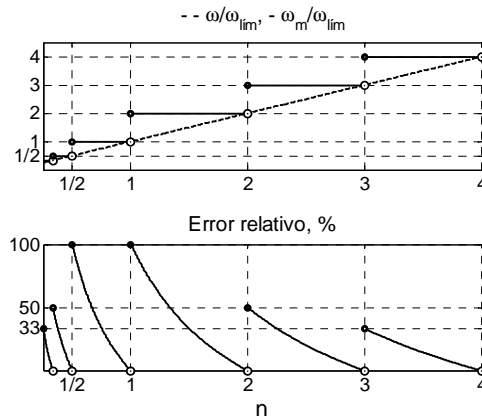


FIG 7. PRIMERA APROXIMACIÓN DE LA VELOCIDAD Y ERROR RELATIVO. FUENTE: AUTORES DEL PROYECTO.

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 $\omega_{lim} \leq \omega < 2\omega_{lim}$. De la figura 7 puede verse que ese error relativo máximo es el mismo para $\frac{1}{2}\omega_{lim} \leq \omega < \omega_{lim}$. 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 de 50%. En la ecuación 2 se encuentra la expresión que define el error relativo máximo.

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

El error relativo máximo, $Er_{M\acute{a}x}$, para $\omega < \omega_{lim}$ resulta ser igual al error relativo máximo con el algoritmo tradicional a tiempo fijo. De igual manera, $Er_{M\acute{a}x}$, 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.

Eliminado: La velocidad aproximada, ω_m , queda completamente definida por la expresión en la ecuación 2. Se escribe en términos de la función piso, y no parte entera, para unificar las expresiones a lo largo de este artículo.¶

$$\omega_m = \begin{cases} \frac{1}{\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor} \omega_{lim} & , \omega < \omega_{lim} \\ \left(\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1 \right) \omega_{lim} & , \omega \geq \omega_{lim} \end{cases} \quad (2)¶$$

A partir de la definición ω_m puede llegarse a una expresión para definir el

Eliminado: , según se define en la ecuación 3, y

Eliminado: más alto

Eliminado: ¶

Eliminado: 3

Eliminado: 3

V. Reducción del error relativo máximo

En esta sección se lleva el algoritmo de aproximación a su óptimo. Se dirá que el algoritmo es óptimo si presenta el $Er_{M\acute{a}x}$ más bajo posible. La minimización del $Er_{M\acute{a}x}$ comienza al tener en cuenta que ω_m siempre 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.

Eliminado: siempre

El $Er_{M\acute{a}x}$ 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 $Er_{M\acute{a}x}$ 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} .

Eliminado: será

Eliminado: rá

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

Eliminado: 4

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.

Con formato: Justificado

Eliminado: 4

Eliminado: rá

Eliminado: 5

Eliminado: 5

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

Véase que el error relativo pasa de 100%, con ω_m , a 50% con ω_{m2} , cuando $\omega_{lim} \leq \omega < 2\omega_{lim}$. La disminución del error relativo puede verse al evaluar el cociente entre los errores máximos para cada aproximación, como se muestra en la ecuación 5. Por medio de esa ecuación se verifica que a velocidades cercanas a ω_{lim} el error máximo se reduce a la mitad; sin embargo, cuando la velocidad es muy grande o muy pequeña comparada con ω_{lim} el error máximo tiende a ser igual.

Eliminado: 6.

$$\frac{Er2_{Sup}}{Er_{M\acute{a}x}} = \begin{cases} \frac{\left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor}{\left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} & , \omega < \omega_{lim} \\ \frac{\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor}{\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1} & , \omega \geq \omega_{lim} \end{cases} \quad (5)$$

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 $Er_{M\acute{a}x}$, mientras que cualquier aproximación que esté por debajo de ω_{m2} resultará en errores relativos mayores a $Er2_{Sup}$. Así queda demostrado que 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 es ahora, cuál es el valor intermedio ω_{m3} que minimiza el error relativo, como se presenta mediante la figura 8, en donde $c1$ es un número real entre cero y uno, y n es el piso de ω/ω_{lim} .

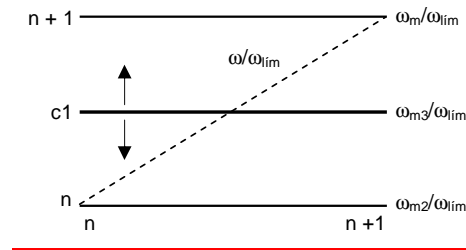


FIG 8. DETERMINACIÓN DE LA CONSTANTE C1, $\omega \geq \omega_{lim}$. FUENTE: AUTORES DEL PROYECTO.

El error relativo máximo por el lado izquierdo está en la ecuación 6, mientras que el error relativo superior, por el extremo derecho, es el que está en la ecuación 7.

$$Er3_I = \frac{c1 - n}{n} 100\% \quad (6)$$

$$Er3_D = \frac{(n+1) - c1}{n+1} 100\% \quad (7)$$

La observación del error relativo por la derecha $Er3_D$ y por la izquierda, $Er3_I$, permite ver que si uno de los aumenta el otro disminuye. Además, el mínimo se encuentra en $Er3_I = Er3_D$. Cualquier otro valor hace que uno de los dos errores sea mayor. Al igualar las ecuaciones 6, 7, resulta el valor de $c1$, como se presenta en la ecuación 8.

$$c1 = \frac{2n(n+1)}{2n+1}, \omega \geq \omega_{lim} \quad (8)$$

En el caso de las velocidades bajas se calcula el valor de una constante $c2$.

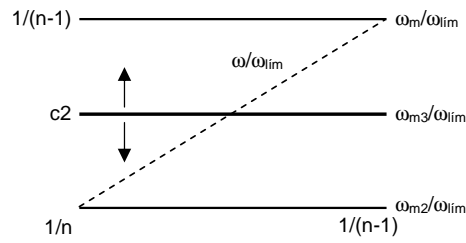


FIGURA 9. DETERMINACIÓN DE LA CONSTANTE C2, $\omega < \omega_{lim}$. FUENTE: AUTORES DEL PROYECTO.

Si se sigue el procedimiento que se realizó para $\omega \geq \omega_{lim}$ se encuentra que $c2$ es la constante en la ecuación 9.

$$c2 = \frac{2}{2n-1}, \omega < \omega_{lim} \quad (9)$$

Dada las constantes $c1$ y $c2$ en las ecuaciones 8 y 9 es posible definir la velocidad aproximada nueva, ω_{m3} , como se indica en la ecuación 10. Por definición, esta velocidad siempre es menor que ω_m y mayor que ω_{m2} .

Con formato: Disminuido 12 pto

Código de campo cambiado

Eliminado: La siguiente modificación al algoritmo, en busca de la optimización del error relativo máximo, proviene una vez más de la observación de los escalones que se generan al aproximar la velocidad. Con la propuesta original en este artículo, los pasos siempre están por encima de la velocidad real, y el error relativo máximo se presenta a la izquierda de cada paso. Con la modificación la aproximación siempre está por debajo de la velocidad real, y esta vez el error relativo supremo se encuentra a la derecha de cada paso. ¶
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_{2Sup} . Así queda demostrado que 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 es ahora, cuál es el valor intermedio ω_{m3} que minimiza el error relativo, como se presenta mediante la figura 8, en donde $c1$ es un número real entre cero y uno, y n es el piso de ω/ω_{lim} . ¶

FIG 8. DETERMINACIÓN DE LA CONSTANTE C1, $\omega \geq \omega_{lim}$. FUENTE: AUTORES DEL PROYECTO. ¶
El error relativo máximo por el lado izquierdo está en la ecuación 7, mientras que el error relativo superior, por el extremo derecho, es el que está en la ecuación 8. ¶

$$Er3_I = \frac{c1 - n}{n} 100\% \quad \dots \dots \dots (7) \quad \ddagger$$

$$Er3_D = \frac{(n+1) - c1}{n+1} 100\% \quad \dots \dots \dots (8) \quad \ddagger$$

Eliminado: , 8

Eliminado: 9

Eliminado: 9

Eliminado: <sp>

$$\omega_{m3} = \begin{cases} \frac{2}{2n-1} \omega_{lim} & \omega < \omega_{lim} \\ \frac{2n(n+1)}{2n+1} \omega_{lim} & \omega \geq \omega_{lim} \end{cases} \quad (10)$$

Con formato: Disminuido 35 pto

En la ecuación 10 debe tenerse en cuenta que la definición de n es la que se consigna en la ecuación 11.

$$n = \begin{cases} \left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1 & , \omega < \omega_{lim} \\ \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor & , \omega \geq \omega_{lim} \end{cases} \quad (11)$$

Eliminado: Dada las constantes c1 y c2 en las ecuaciones 9 y 10 es posible definir la velocidad aproximada nueva, ω_{m3} , como se indica en la ecuación 11. Por definición, esta velocidad siempre es menor que ω_m y mayor que ω_{m2} .

A continuación se muestra que ω_{m3} es el promedio armónico entre las aproximaciones ω_m y ω_{m2} . En la ecuación 12 se define y calcula el promedio armónico, indicada como ω_H . Primero para velocidades bajas.

$$\omega_H = \frac{2}{\frac{1}{\omega_m} + \frac{1}{\omega_{m2}}}, \quad \begin{cases} \omega_m = \frac{1}{n-1} \omega_{lim} \\ \omega_{m2} = \frac{1}{n} \omega_{lim} \end{cases} \quad \omega_H = \frac{2}{2n-1} \omega_{lim} = \omega_{m3} \quad \omega < \omega_{lim} \quad (12)$$

Y ahora para velocidades altas.

$$\omega_H = \frac{2}{\frac{1}{\omega_m} + \frac{1}{\omega_{m2}}}, \quad \begin{cases} \omega_m = (n+1)\omega_{lim} \\ \omega_{m2} = n\omega_{lim} \end{cases} \quad \omega_H = \frac{2n(n+1)}{2n+1} \omega_{lim} = \omega_{m3} \quad \omega \geq \omega_{lim} \quad (13)$$

Eliminado: En la ecuación 11 debe tenerse en cuenta que la definición de n es la que se consigna en la ecuación 12.

$$n = \begin{cases} \left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1 & , \omega < \omega_{lim} \\ \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor & , \omega \geq \omega_{lim} \end{cases} \quad (12)^*$$

Eliminado: 13

Eliminado: 13)

Eliminado: 14

Cualquier promedio diferente al armónico resulta en errores relativos máximos más grandes. El error relativo máximo por segmentos está definido en la ecuación 14. 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_{2sup} .

$$Er_{3M\acute{a}x} = \begin{cases} \frac{1}{2 \left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} \times 100\% & , \omega < \omega_{lim} \\ \frac{1}{2 \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1} \times 100\% & , \omega \geq \omega_{lim} \end{cases} \quad (14)$$

Eliminado: De otra parte, en la expresión en la ecuación 15 se define ω_{m3} en términos de ω y ω_{lim} , lo cual va a ser útil para generar una expresión general para el error relativo.

$$\omega_{m3} = \begin{cases} \left(\frac{2}{2 \left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} \right) \omega_{lim} \\ \left(\frac{2 \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor \left(\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1 \right)}{2 \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1} \right) \omega_{lim} \end{cases} \quad (15)^*$$

Eliminado: 16.

Eliminado: 16

Eliminado: 17.

Eliminado: 17)

De la misma forma en que se comparó Er_{2Sup} con $Er_{M\acute{a}x}$ ahora se compara $Er_{3M\acute{a}x}$ con $Er_{M\acute{a}x}$. El resultado está en la ecuación 15. Puede verse que esa relación siempre es menor a 1/2. Así que por medio del cálculo de ω_{m3} , se garantiza que error en cualquier caso es menor a la mitad de lo que se logra con los algoritmos tradicionales.

$$\frac{Er_{3M\acute{a}x}}{Er_{M\acute{a}x}} = \begin{cases} \frac{\left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor}{2 \left\lfloor \frac{\omega_{lim}}{\omega} \right\rfloor + 1} & , \omega < \omega_{lim} \\ \frac{\left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor}{2 \left\lfloor \frac{\omega}{\omega_{lim}} \right\rfloor + 1} & , \omega \geq \omega_{lim} \end{cases} \quad (15)$$

VI. 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 acoplado a un eje, y como resultado del giro del eje aparece una señal que es cero voltios, para el nivel bajo, y cinco, para el alto, en el encoder. Esa señal es capturada por una tarjeta de adquisición de datos, como lecturas de voltaje, y esos valores son almacenados en un archivo .mat, para su posterior uso en Simulink.

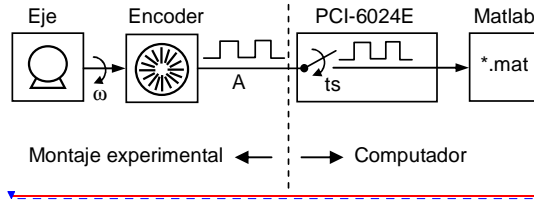


FIG 10. MONTAJE EXPERIMENTAL Y PROCESAMIENTO DE LAS SEÑALES. FUENTE: AUTORES DEL PROYECTO.

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 μ s.

El valor de velocidad que se asume como real es adquirido mediante la aplicación del mismo algoritmo aquí propuesto, pero con parámetros que hacen que el error relativo sea pequeño, en comparación con lo que puede esperarse para los experimentos realizados. Se decidió utilizar $pul = 160$ y $dt = 2$ s; de esta forma, por ejemplo, si la velocidad es 2 rev/s, el error relativo máximo es 0,07%, y este disminuye al aumentar la velocidad, por ejemplo a 24 rev/s el error relativo máximo es 0,0065%.

El primer experimento se realiza utilizando los parámetros $pul = 160$ y $dt = 1$ ms. Primero se fija el voltaje de alimentación del motor, por ejemplo en 5 V, y se espera hasta que el transitorio de velocidad haya pasado. Cuando esto sucede, se captura la señal del encoder por un intervalo de 10 s. Se almacena esa información en un archivo .mat, de Matlab, y luego se corre el algoritmo de aproximación en Simulink, utilizando como entrada los valores almacenados. Se repite el procedimiento 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.

Como se dijo al final de la sección V, el error relativo máximo debido a los algoritmos a tiempo y espacio fijo coincide con el error para la aproximación ω_m . Ahora con el fin de realizar una comparación adicional, se implementa el llamado algoritmo 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 18. 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 realizado por medio de la sumatoria hace que la aproximación dependa de instantes anteriores, y entonces, si bien se suaviza la aproximación el error máximo resulta similar al error con el algoritmo a tiempo fijo.

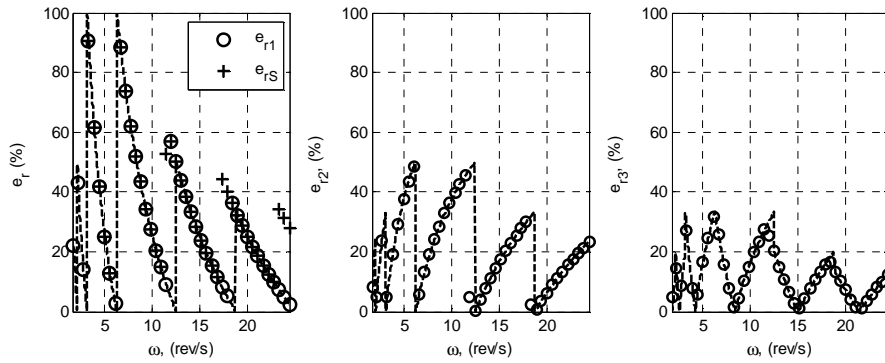
$$\omega_s = \frac{\sum_{j=0}^{Ndt-1} N_p(i-j)}{Ndt} \omega_{lim} \quad (18)$$

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. Exceptuando el salto entre velocidades $\omega =$

Eliminado: . Se
Eliminado: de tratamiento de las señales presentado
Eliminado: al
Eliminado: del cual se quiere aproximar su velocidad. Como
Eliminado: ,
Eliminado: . La
Eliminado: <sp>¶

Eliminado: 100
Eliminado: , así la
Eliminado: máxima
Eliminado: 62,5 rev/s.¶
 Con el fin de verificar el algoritmo, además de adquirir datos de un encoder real durante su funcionamiento, se utilizó el
Eliminado: muy
Eliminado: que se mide
Eliminado: realizó para
Eliminado: , es decir, apenas 10 veces más grande que ts. Se
Eliminado: No se realiza en tiempo real, por que el tiempo que se requiere para la ejecución del algoritmo no lo permite, dado que el computador es un Pentium 4 con 512 Mb en ram.
Eliminado: ¶
 Para poder verificar que el
Eliminado: que se encuentra es el esperado, se utiliza
Eliminado: lectura de velocidad supuesta como real (con $dt = 2$ s), y se utilizan las ecuaciones 2, 4, y 15 para encontrar los valores teóricos de ω_m , ω_{m2} y ω_{m3} . El resultado del cálculo del error relativo para esos valores teóricos se indica con un marcador circular
Eliminado: signo más
Eliminado: El error relativo e_{r1} corresponde a ω_m , e_{r2} a ω_{m2} y e_{r3} a ω_{m3} .
Eliminado: El error relativo se comporta tal como se espera. Existe una coincidencia perfecta entre el error relativo teórico y el experimental, excepto en un punto, cuando se presentan

$2\omega_{\text{lím}}$ a $\omega = 3\omega_{\text{lím}}$. Este salto es explicable dado que se está trabajando con datos provenientes de un encoder real, los cuales pueden incluir errores, como por ejemplo pérdidas de conteos.



Con formato: Centrado

FIG 11. ERROR RELATIVO TEÓRICO Y EXPERIMENTAL, DT = 0,001 s. FUENTE: AUTORES DEL PROYECTO.

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 $\omega_{\text{lím}}$. Para realizar esta prueba se disminuye el número de pulsos del encoder, pero en lugar de cambiar el encoder, se toma un impulso, I_{ep} , cada k número entero de impulsos. Por ejemplo, se utilizó k igual 40. Ahora, si dt es 1 ms, entonces $\omega_{\text{lím}} = 250$ rev/s, y si todas las velocidades están por debajo de 30 rev/s, entonces los errores relativos son menores a 10%, como efectivamente ocurre.

VII. 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.

Eliminado: gracias a la eliminación d

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 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 es siempre menor o igual. Se observa que el error relativo máximo siempre es menor que la mitad que el que se encuentra con los algoritmos tradicionales.

Si bien es posible garantizar los resultados del algoritmo para velocidades constantes, una vez cuando 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 sólo 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.

Eliminado: Además se observa experimentalmente el efecto del cambio del tiempo de discretización, así como del cambio del número de pulsos del encoder. Se concluye que el mejor resultado se logra cuando se utiliza el mayor número de pulsos y el mayor dt posible, o lo contrario, es decir, el menor número de pulsos del encoder y el menor dt posible. ¶

Es posible afirmar que las instrucciones del algoritmo son pocas y simples, sin embargo, la sincronización de las señales es un hecho crítico en la ejecución del algoritmo. Además de esto, la forma de mejorar la resolución en la aproximación exige que reloj del sistema, t_s , sea lo más pequeño y

preciso posible. Por tanto la aplicación del algoritmo propuesto en la práctica requiere del estudio de las capacidades de los circuitos electrónicos tradicionales para correr el algoritmo. Por ejemplo puede estudiarse la aplicación del algoritmo en procesadores digitales de señales DSP y FPGA.

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 de 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*, marzo 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, agosto 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*, septiembre 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*, August 2011, vol 58, núm 8, pp. 3634-3646.
- LIU, G. On velocity estimation using position measurements. *American Control Conference, 2002. Proceedings of the 2002*, mayo 2002, vol 2, pp. 1115 – 1120.
- LYGOURAS, J.; PACHIDIS, T.; TARCHANIDIS, K. y KODOGIANNIS V. Adaptive High-Performance Velocity Evaluation Based on a High-Resolution Time-to-Digital Converter. *IEEE Transactions on Instrumentation and measurement*, septiembre 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.; POP A. y SZABO C. 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*, septiembre/octubre 2008, vol 44, núm 5, pp. 1436 – 1444.
- SU, Y.; ZHENG, C.; SUN, D.; DUAN, B. 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.; MIZUOCHI, M. y OHNISHI, K. A Wide-Range Velocity Measurement Method for Motion Control. *IEEE Transactions on industrial electronics*, febrero 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.

Con formato: Sangría:
Izquierda: 0 cm, Primera línea:
0 cm