

Implementación Experimental de un Controlador de Posición para un Motor de Corriente Directa

R. Salas Cabrera, J. C. Mayo Maldonado, E. Y. Rendón Fraga,
A. González Rodríguez, E. N. Salas Cabrera, O. F. Ruíz Martínez
Maestría en Ingeniería Eléctrica, Departamento de Ingeniería Electrónica
Instituto Tecnológico de Ciudad Madero
Av. 1o. de Mayo S/N, Cd. Madero, Mexico
salascabrera@aol.com

Resumen—Este artículo presenta los resultados experimentales del control de la posición del rotor de un motor de corriente directa. Para el control en lazo cerrado se utiliza una herramienta en tiempo real basada en una distribución del sistema operativo Linux. Se propone una ecuación dinámica para el par mecánico. Se llevan a cabo pruebas dinámicas y de estado estacionario para indentificar los parámetros del sistema, incluyendo los de la ecuación para el par mecánico. Se implementa un observador de estado en tiempo discreto para determinar experimentalmente las variables de estado del sistema. Se construye un actuador electrónico como interfaz de potencia y se diseña un sensor para medir la posición del rotor.

Palabras clave: tiempo real, observador de estado, motor de corriente directa, electrónica.

I. INTRODUCCIÓN

La contribución principal de este trabajo está relacionada con la implementación experimental de un controlador y un observador de estado empleando una herramienta en tiempo real. El énfasis de este trabajo no es el análisis y deducción teórica de una nueva ley de control sino la implementación experimental utilizando diversos componentes de software y hardware. El objetivo de control es la posición del rotor de un motor de corriente directa con excitación separada. Se utiliza la herramienta de control en tiempo real llamada RTAI y el software Scilab como interfase de programación (Bucher, Mannori y Netter, 2008). Los resultados de algunos trabajos previos han demostrado que RTAI es una herramienta confiable para fines de control (Jugo, 2004), (Boizot, Busvelle y Sachau, 2007), (Salas Cabrera, Mayo Maldonado, Gonzalez Rodriguez, Rendon Fraga y Salas Cabrera, 2008). También se utiliza una tarjeta de adquisición de datos National Instruments PCI-6024E (National Instruments, 2006) como hardware computacional. La tarjeta de adquisición de datos posee entradas y salidas tanto analógicas como digitales a las que tiene acceso el programa en tiempo real. La plataforma en tiempo real RTAI-Lab que fue utilizada para la implementación del controlador de posición, así como sus herramientas, están descritas en trabajos anteriores (Salas Cabrera, Mayo Maldonado, Gonzalez Rodriguez, Rendon Fraga y Salas Cabrera, 2008).

Existen aproximaciones para el modelado del par mecánico del motor de corriente directa. Algunas contribuciones muestran descripciones algebraicas (Ruderman, Krettek y Hoffmann, 2008), otros presentan aproximaciones dinámicas (Tjahjowidodo, Al Bender y Van Brussel, 2005). En este trabajo se propone una aproximación dinámica lineal con el fin de mejorar el desempeño del sistema en lazo cerrado de forma experimental. Se sabe que los encoders incrementales presentan imperfecciones que hacen que se pierda la confiabilidad cuando se mide la velocidad del rotor (Merry, Van de Molengraft y Steinbuch, 2007). Con el propósito de disminuir al mínimo el número de sensores, así como los errores de medición, se implementa un observador de estado en tiempo discreto. La posición del rotor es la única variable de estado medida en la implementación de este controlador. En la sección III se presenta la contribución principal de este trabajo, es decir la implementación experimental de este controlador basado en un observador.

II. MODELADO DINÁMICO DEL MOTOR DE CORRIENTE DIRECTA

Aplicando leyes fundamentales de ingeniería eléctrica y mecánica se pueden definir las siguientes ecuaciones para el motor de corriente directa (Krause, Wasynczuk y Sudhoff, 2004):

$$V_f = r_f i_f + L_{ff} \frac{d}{dt} i_f \quad (1)$$

$$V_a = r_a i_a + L_{aa} \frac{d}{dt} i_a + L_{af} i_f \omega_r \quad (2)$$

$$J \frac{d}{dt} \omega_r + T_L = L_{af} i_f i_a \quad (3)$$

$$\frac{d}{dt} \theta_r = \omega_r \quad (4)$$

Se llevaron a cabo diferentes pruebas transitorias y en estado estacionario para determinar los valores de los parámetros de las ecuaciones (1)-(4). Una de estas pruebas se describe en la siguiente subsección.

TABLA I
 NOMENCLATURA

ω_r	Velocidad del Rotor
i_a	Corriente de Armadura
$i_f(0.46 \text{ Amps})$	Corriente de Campo
V_f	Voltaje de Campo
V_a	Voltaje de Armadura
$r_a(6.615 \text{ Ohms})$	Resistencia de Armadura
$L_{aa}(0.0645 \text{ H})$	Inductancia de Armadura
$L_{af}(1.7686 \text{ H})$	Inductancia Mutua
$J(0.0038 \text{ kg/m}^2)$	Inercia
T_L	Par Mecánico
$K_0(0.20907), K_1(-9.8297)$	Parámetros del Par Mecánico

II-A. Modelo del Par Mecánico

Se considera hacer una aproximación del modelo del par mecánico. Modelar la dinámica del par mecánico nos proporciona mayor información del sistema a controlar, por lo tanto se espera que los resultados experimentales sean similares a las simulaciones previas. A continuación se propone la siguiente ecuación dinámica para el par mecánico:

$$\frac{d}{dt}T_L = K_0\omega_r + K_1T_L \quad (5)$$

La ecuación (5) es una aproximación dinámica lineal del par mecánico la cual contiene 2 parámetros: K_0 y K_1 . Con el fin de obtener un modelo que permita obtener los valores numéricos de los parámetros en la expresión anterior, se utilizan las ecuaciones (3) y (5) y se forma el siguiente subsistema

$$\frac{d}{dt} \begin{bmatrix} \omega_r \\ T_L \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{J} \\ K_0 & K_1 \end{bmatrix} \begin{bmatrix} \omega_r \\ T_L \end{bmatrix} + \begin{bmatrix} \frac{L_{af}i_f}{J} \\ 0 \end{bmatrix} i_a \quad (6)$$

donde se puede considerar que la corriente de armadura i_a es la entrada y la velocidad del rotor ω_r es la salida.

A continuación de forma experimental, con ayuda de un microcontrolador ATMEGA8 (Atmel Corporation, 2006) y un puente H como actuador de potencia, se aplican aleatoriamente entradas de corriente de armadura al motor de corriente directa. La corriente de armadura es medida utilizando un sensor de corriente por efecto Hall Nana Electronics SHR-100 (Nana Electronics, 2004). De forma similar se mide la velocidad del rotor, la cual es la salida de la ecuación (6), utilizando un encoder incremental (Pepperl+Fuchs Group, 2004). Las variables medidas son capturadas dinámicamente haciendo uso de la tarjeta de adquisición de datos National Instruments PCI-6024E (National Instruments, 2006) y utilizando la herramienta FIFO incluida en la plataforma en tiempo real RTAI-Lab (Bucher, Mannori y Netter, 2008). Una vez que se obtienen los vectores de datos experimentales, se utiliza el método denominado error predictivo para la identificación dinámica de sistemas. Este método se encuentra disponible en el software Matlab. De este modo se obtiene la versión numérica de la ecuación (6). Comparando la ecuación (5) y los

valores numéricos obtenidos para la ecuación (6) se pueden determinar los valores numéricos para los parámetros K_0 y K_1 .

II-B. Representación en Espacio de Estado

En esta implementación se utiliza un modelo que considera que el motor tiene aplicada una corriente de campo constante. En particular, $i_f = 0.46$ amps. Esta consideración implica que $di_f/dt = 0$ para todo tiempo t , por tanto su ecuación de estado no se incluye en el modelo empleado. Los parámetros que se muestran en la Tabla I se pueden sustituir en las ecuaciones (2)-(5) y se puede deducir la representación en espacio de estado en tiempo continuo. Utilizando una frecuencia de muestreo de 5 kHz, es también posible obtener el modelo dinámico nominal en espacio de estado en tiempo discreto, el cual resulta ser lineal e invariante en el tiempo. Es decir

$$\begin{bmatrix} \theta_r(k+1) \\ \omega_r(k+1) \\ i_a(k+1) \\ T_L(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0,000200 & 0,000004 & -0,000005 \\ 0 & 0,999945 & 0,042383 & -0,052578 \\ 0 & -0,002497 & 0,979644 & 0,000065 \\ 0 & 0,000041 & 0,9 \times 10^{-6} & 0,998035 \end{bmatrix} \begin{bmatrix} \theta_r(k) \\ \omega_r(k) \\ i_a(k) \\ T_L(k) \end{bmatrix} + \begin{bmatrix} 4,4 \times 10^{-9} \\ 0,0000659 \\ 0,0030691 \\ 9,201 \times 10^{-10} \end{bmatrix} V_a(k) \quad (7)$$

$$\theta_r(k) = [1 \ 0 \ 0 \ 0] [\theta_r(k) \ \omega_r(k) \ i_a(k) \ T_L(k)]^T \quad (8)$$

Se observa claramente que la entrada al sistema es V_a y la salida es θ_r . Con el propósito de eliminar el error en estado estacionario y compensar las variaciones paramétricas

$$x_I(k+1) = x_I(k) + e(k) = x_I(k) + y(k) - r(k) \quad (9)$$

donde x_I es la variable de estado del integrador, $y = \theta_r$ es la salida a controlar y r es el set point.

II-C. Observador de Estado

En esta implementación se utiliza un observador de estado cuya expresión es de la siguiente forma

$$\tilde{x}(k+1) = G\tilde{x}(k) + Hu(k) + K_e[y(k) - C\tilde{x}(k)] \quad (10)$$

Se considera que la posición del rotor es la única variable de estado medida. El vector de ganancias K_e asociado al término correctivo $y(k) - C\tilde{x}(k) = \theta_r(k) - \hat{\theta}_r(k)$ se obtiene básicamente proponiendo una constante de tiempo y siguiendo el procedimiento descrito en (Franklin, Powell y Workman, 1998). También es posible calcular las ganancias utilizando el comando *ppol* del software Scilab, como se explica en (Campbell, Chancellier y Nikoukhah, 2006). En este trabajo se seleccionaron los siguientes polos: $z_{1,2,3,4} =$

0,994017964, 0,994017963, 0,994017962, 0,994017961,
con los cuales se obtiene el siguiente vector de ganancias:

$$K_e = \begin{bmatrix} 0,0015523 \\ 0,1544085 \\ -0,0392419 \\ -0,0014389 \end{bmatrix} \quad (11)$$

En esta implementación del controlador se utiliza solo un sensor, el cual es empleado para medir la posición del rotor. Es importante aclarar que las mediciones de velocidad y corriente de armadura que se mencionan en la subsección II-A, se utilizan únicamente para determinar los parámetros del par mecánico.

II-D. Controlador

En este trabajo se utiliza la técnica denominada reubicación de polos para calcular las ganancias de la retroalimentación de estado definida en la ecuación (12).

$$u(k) = - [K_I \quad K] \begin{bmatrix} x_I(k) \\ x(k) \end{bmatrix} \quad (12)$$

donde $u(k) = V_a(k)$, K_I es la ganancia asociada a la variable de estado del integrador y K es un vector que contiene las ganancias asociadas a las variables de estado del sistema original: posición del rotor, velocidad del rotor, corriente de armadura y par mecánico. Las ganancias son calculadas de acuerdo al procedimiento descrito en (Franklin, Powell y Workman, 1998). Para determinar los polos del sistema en lazo cerrado, se propone una constante de tiempo relacionada a cada uno de ellos. Se utiliza también la siguiente ecuación (13) que relaciona a los planos complejos z y s , (Franklin, Powell y Workman, 1998).

$$z = e^{-\frac{1}{\tau}T} \quad (13)$$

donde τ es la constante de tiempo propuesta y T es el periodo de muestreo, que para este caso es igual a 0,0002 segundos. Para calcular z_1 utilizamos una constante de tiempo igual a 0,1 segundos, por lo tanto: $z_1 = e^{-\frac{1}{0,1}(0,0002)} = 0,998001998$, siguiendo esta misma metodología se obtienen todos los polos del sistema en lazo cerrado: $z_{1,2,3,4,5} = [0,998001998, 0,998001997, 0,998001996, 0,998001995, 0,998001994]$, los cuales corresponden a las siguientes ganancias:

$$[K_I, K] = [0,0006168, 1,2288494, -0,6467532, -4,021708, -2,4009488] \quad (14)$$

III. DESARROLLO EXPERIMENTAL

III-A. Plataforma en Tiempo Real

Se utiliza RTAI-Lab como plataforma de control, con el propósito de resolver el algoritmo de control en tiempo real. Como se menciona en la sección II, se define una frecuencia de muestreo de 5 kHz para nuestro modelo discreto. La plataforma RTAI-Lab tiene como función, resolver el algoritmo de control bajo requerimientos estrictos de tiempo. Además

deberá acceder en tiempo real a los convertidores Análogo-Digital y Digital-Análogo de la tarjeta de adquisición de datos, para este caso una National Instruments PCI-6024E (National Instruments, 2006).

Para este trabajo se utilizó la distribución Knoppix 5.0 como sistema operativo Linux, la cual incluye todas las herramientas suficientes para llevar a cabo la programación y puesta en marcha del sistema de control, como se describe en trabajos anteriores (Salas Cabrera, Mayo Maldonado, Gonzalez Rodriguez, Rendon Fraga y Salas Cabrera, 2008). La herramienta RTAI-Lab incluye también la librería COMEDI que contiene los drivers (controladores) para la mayoría de las tarjetas de adquisición de datos (Bucher, Mannori y Netter, 2008). Se utiliza Scilab como interfase de programación, específicamente su herramienta Scicos, la cual se basa en bloques que cumplen con diferentes funciones matemáticas. RTAI-Lab a su vez, proporciona bloques adicionales para la programación en tiempo real en Scicos. Dentro del grupo de herramientas contenidas en RTAI-Lab, se encuentra un compilador para convertir los programas construidos en Scicos, a programas ejecutables para Linux, el cual es llamado RTAIcodegen.

En la Figura 1 se muestra el programa principal en Scicos utilizado para implementar el controlador. Para hacer posible la presentación del programa en una sola figura, se utilizan super bloques de Scicos, los cuales contienen varios bloques que incluyen programas mas extensos como se explica en (Campbell, Chancellier y Nikoukhah, 2006). El programa principal se puede dividir en 3 partes. La primera es un bloque que contiene un algoritmo utilizado para la medición de posición, el cual será explicado mas adelante. La segunda parte es el observador de estado que contiene la síntesis en bloques de la ecuación (10) y que incluye al vector de ganancias K_e de la ecuación (11). El programa completo para el observador de estado se muestra en la Figura 2 y se representa como un super bloque en la Figura 1. Por último la tercera parte del programa, la cual no está contenida en un super bloque, contiene el algoritmo de control con las ganancias de la ecuación (14). También incluye la ecuación de estado del integrador, así como el bloque STEP utilizado para definir el valor del set point y el bloque COMEDI D/A que es el bloque que enlaza al programa con el convertidor digital-análogo (DAC) de la tarjeta de adquisición de datos. Puesto que el controlador define el voltaje de armadura que debe aplicarse al motor de corriente directa, se incluye una ganancia A que tiene la función de escalar el voltaje a un rango más pequeño que pueda ser escrito en el DAC de la tarjeta de adquisición de datos. El voltaje escalado posteriormente será aplicado a la etapa de control de un actuador de potencia, el cual aplicará finalmente el voltaje definido originalmente por el controlador al motor de corriente directa.

III-B. Medición de la Posición del Rotor

Para medir la posición del rotor se utilizaron diferentes componentes de software y hardware. En primer lu-

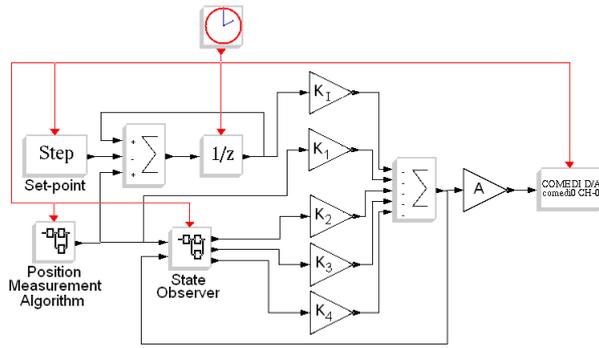


Figura 1. Programa principal en Scicos

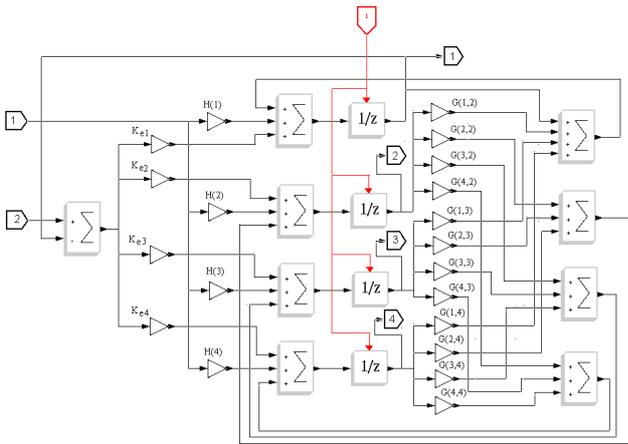


Figura 2. Programa para el Observador de Estado en Scicos

gar fue necesario utilizar un encoder incremental T2406 (Pepperl+Fuchs Group, 2004) el cual está acoplado mecánicamente a la flecha del rotor. Anteriormente se comentaba que existen problemas con la medición de velocidad utilizando encoders incrementales, debido a que es necesario medir la frecuencia de los pulsos. Sin embargo para la medición de posición no es necesario medir la frecuencia que proporciona el encoder incremental, sino la cantidad de pulsos. A partir de esto se concluye que es necesario emplear un contador de pulsos que haga posible guardar en memoria el número de pulsos proporcionados por el encoder y posteriormente convertir este valor a radianes.

La frecuencia que proporciona el encoder incremental para todos los rangos de velocidad es mayor a la frecuencia de muestreo del programa en tiempo real. Por lo tanto es necesario implementar un circuito independiente que haga un conteo de los pulsos del encoder y que proporcione un valor equivalente al conteo que pueda ser capturado por la tarjeta de adquisición de datos. Se utiliza un microcontrolador ATMEGA8535 (Atmel Corporation, 2008) el cual es programado como contador ascendente/descendente de 16 bits, es decir, cuando el rotor gira en una dirección el microcontrolador cuenta de manera ascendente el número de pulsos suministrados por el encoder. Del mismo modo,

el microcontrolador cuenta de forma descendente si el rotor del motor gira en sentido contrario. El encoder incremental empleado en este trabajo proporciona 1024 pulsos por revolución. Existe un bit de control que determina el sentido de giro del rotor al igual que el conteo ascendente o descendente del microcontrolador. El bit de control se genera con un circuito integrado flip-flop, el cual tiene como entrada los canales A y B del encoder incremental, en su salida proporciona un "1" o un "0" binario dependiendo del sentido de giro del rotor del motor de corriente directa. El conteo de 16 bits se escribe en 16 terminales del microcontrolador, esto quiere decir que el microcontrolador proporciona un número binario equivalente a la posición del rotor, el cual puede detectarse en tiempo real con la tarjeta de adquisición de datos. El diagrama completo del hardware utilizado para la medición de la posición del rotor se observa en la Figura 3.

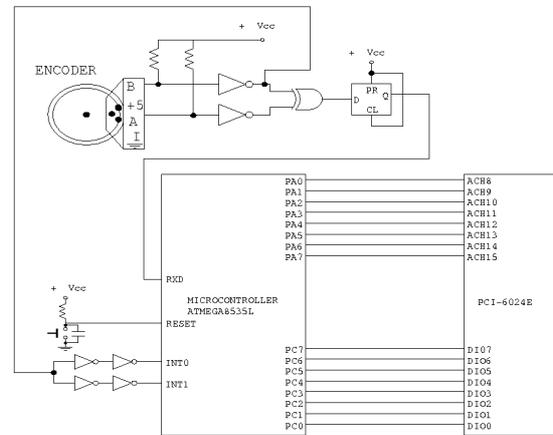


Figura 3. Esquemático del sensor de posición

El microcontrolador proporciona un número binario de 16 bits el cual puede ser leído por las entradas digitales de la tarjeta de adquisición de datos National Instruments PCI-6024E, sin embargo esta solo posee 8 entradas digitales y 16 analógicas, por lo tanto es necesario hacer un algoritmo dentro del programa principal para acondicionar algunas entradas analógicas haciendo posible que funcionen como entradas digitales. Posteriormente es necesario convertir el valor binario de la posición del rotor a radianes.

Las 8 entradas digitales de la tarjeta de adquisición de datos son leídas por los bloques COMEDI DI del software Scilab/Scicos, esto se muestra en la Figura 4. El algoritmo consiste en multiplicar cada bit (1 ó 0) por su valor equivalente en el sistema decimal. Con el propósito de acondicionar las entradas analógicas para la lectura de datos digitales, se empleó una subrutina que convierte un nivel menor a 2.5 volts en un "0" binario y un valor mayor a 2.5 volts en un "1" binario, como se muestra en la Figura 5. Posteriormente, al igual que como se hizo con las entradas digitales, los valores binarios de cada entrada analógica acondicionada son multiplicados por su valor equivalente

en el sistema decimal. Por último se hace una suma de todos los valores y se escala para obtener su equivalente en radianes como se observa en la Figura 6. Se hace uso nuevamente de super bloques, los cuales contienen la suma de los valores capturados por las entradas digitales así como de las analógicas.

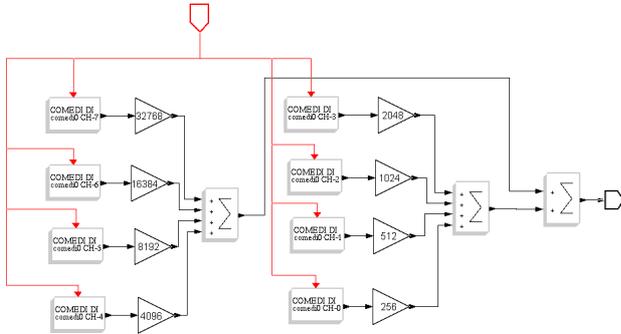


Figura 4. Algoritmo de medición asociado a las entradas digitales.

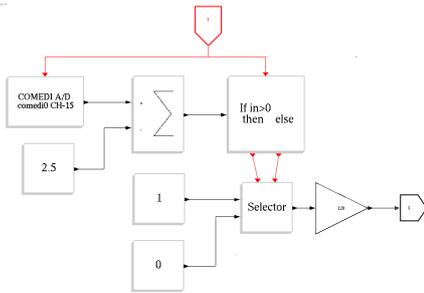


Figura 5. Algoritmo de medición asociado a las entradas analógicas.

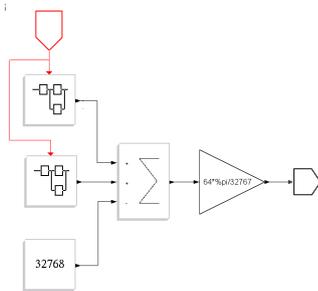


Figura 6. Algoritmo para convertir a radianes el número binario leído.

III-C. Actuador de Potencia

Como actuador de potencia se emplea un Puente H basado en MOSFETs y la modulación por ancho de pulso (PWM). Este convertidor electrónico es normalmente utilizado para controlar la velocidad o la posición de los motores de corriente directa de forma bidireccional (Rashid Muhammed, 2004). La ecuación que determina el voltaje de armadura aplicado al motor de corriente directa es:

$V_a = \delta V_{cc}$, donde δ es el ciclo de trabajo de los MOSFETs y V_{cc} es el voltaje de la fuente de potencia. El voltaje calculado por el controlador, que posteriormente es escrito en el convertidor digital-análogo de la tarjeta de adquisición de datos, determina el ciclo de trabajo de los MOSFETs. Por lo tanto, un voltaje de armadura equivalente al calculado por el controlador, es aplicado a las terminales físicas del motor de corriente directa.

El diagrama del actuador de potencia se observa en la Figura 7. Se divide en 5 partes. La primera parte (a), se encarga de acondicionar el voltaje suministrado por la tarjeta de adquisición de datos (Coughlin y Driscoll, 2001). La parte (b) se encarga de generar un PWM con un ciclo de trabajo que depende del voltaje aplicado a la parte (a). La parte (b) también está configurada para proporcionar un tiempo de recuperación para proteger a los MOSFETs durante el periodo de conmutación (Boylestad y Nashelsky, 2006). La etapa (c) aísla eléctricamente las etapas de control (a) y (b) de la etapa de potencia (Coughlin y Driscoll, 2001). La parte (d) es la etapa del puente H construido con MOSFETs, la cual está conectada directamente a las terminales del motor de corriente directa. Por último la parte (e) contiene protecciones de sobrecorriente para la etapa de potencia (d).

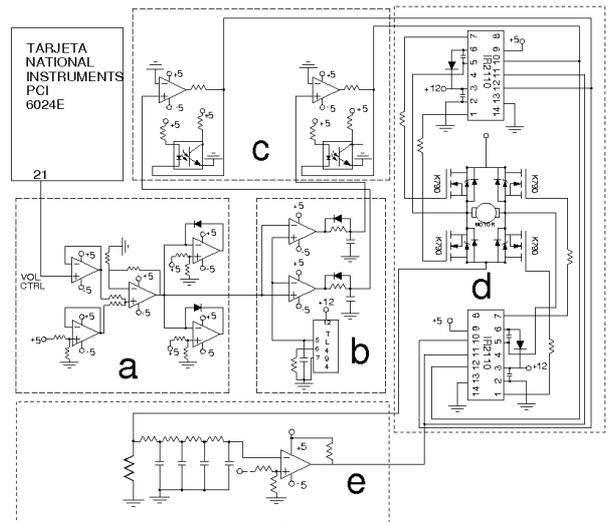


Figura 7. Diagrama del puente H como actuador de potencia.

IV. RESULTADOS EXPERIMENTALES

A continuación se presentan algunos resultados experimentales de la implementación del controlador de posición del rotor del motor de corriente directa. El periodo de muestreo para el controlador y el observador discreto es de 0.0002 segundos, que a su vez es el tiempo definido por el programa en tiempo real para leer y escribir las entradas y salidas de la tarjeta de adquisición de datos. El sistema dinámico discreto (observador-controlador) que se resuelve en tiempo real a una frecuencia de 5 kHz es de orden 5. Los registros transitorios fueron obtenidos

utilizando la herramienta FIFO de la plataforma RTAI-Lab (Bucher, Mannori y Netter, 2008). En la primera prueba se define un setpoint de 18.85 radianes (3 revoluciones). En la Figura 8, con línea continua se presenta el registro experimental transitorio de la posición del rotor medida en lazo cerrado; con asteriscos se presenta el resultado obtenido en la simulación correspondiente. En la Figura 9 se presenta el registro experimental transitorio de la velocidad del rotor observada; con asteriscos los resultados de la simulación bajo las mismas condiciones de la prueba. En la Figura 10 se presenta el voltaje de armadura experimental que es calculado por el controlador en el programa en tiempo real. Este voltaje posteriormente se aplica al motor de corriente directa por medio del actuador de potencia; con asteriscos se presenta el resultado de la simulación correspondiente. Por último en la Figura 11 se muestran los resultados de una prueba en donde se varia el valor del setpoint a 6π , 12π y 18π (3, 6 y 9 revoluciones) respectivamente.

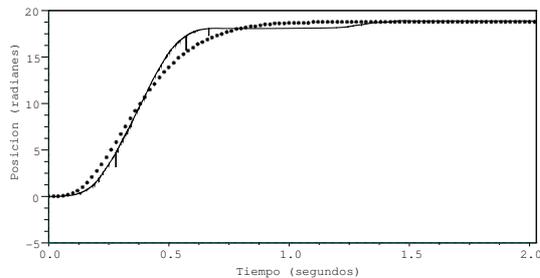


Figura 8. Posición del rotor –experimental medida y **simulada.

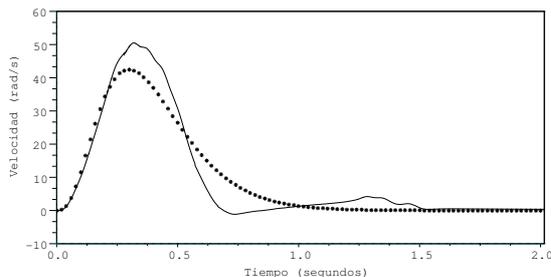


Figura 9. Velocidad del rotor –experimental observada y **simulada.

V. AGRADECIMIENTOS

Los autores desean agradecer a Michael Griffin y Carlos A. Cruz-Villar por su apoyo desinteresado cuando recién se iniciaba este trabajo.

REFERENCIAS

Bucher, R., Mannori, S. y T. Netter (2008). *RTAI-Lab tutorial: Scilab, Comedi and real-time control*.

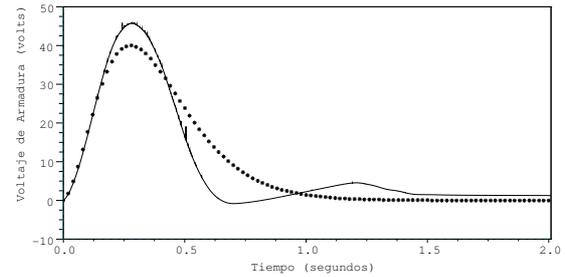


Figura 10. Voltaje de armadura –experimental calculado y **simulado.

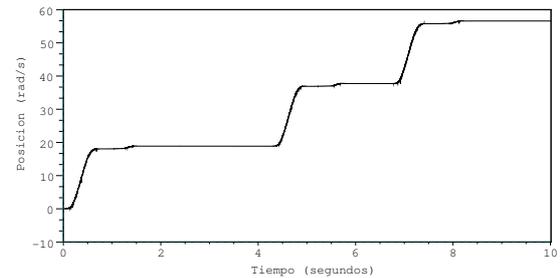


Figura 11. Registro de la posición empleando varios set points.

- J. Jugo (2004). Real-time control of a DC motor using Scilab and RTAI. *Scilab INRIA Rocquencourt*.
- Boizot, N., Busvelle, E. y J. Sachau (2007). High-gain observers and Kalman filtering in hard-time. *RTL 9th Workshop*.
- Salas Cabrera, R., Mayo Maldonado, J. C., Gonzalez Rodriguez, A., Rendon Fraga, E. Y. y Salas Cabrera N. (2008). Implementacion experimental de un controlador de velocidad para un motor de corriente directa en tiempo real. *Asociacion de Mexico de control Automatico 2008*.
- National Instruments (2006). *PCI-6024E National Instruments Manual*.
- Merry, R., Van de Molengraft, R. y M. Steinbuch (2007). Error modeling and improved position estimation for optical incremental encoders by means of time stamping. *American Control Conference*
- Ruderman, M., Krettek, J. y F. Hoffmann (2008). Optimal state space control of DC motor. *IFAC 17th World Congress*.
- Krause, P. C., Wasynczuk, O., y S. D. Sudhoff (2004). *Analysis of Electrical Machinery and Drive Systems*. John Wiley and Sons, IEEE Press Power Engineering.
- Tjahjowidodo, T., Al Bender, F. y H. Van Brussel (2005). Friction identification and compensation in a DC motor. *IFAC 16th World Congress*.
- Atmel Corporation (2006). *8 bit Avr Microcontroller Atmega8. Data sheet*.
- Nana Electronics (2004). *Nana Electronics SHR-100 current sensor. Data sheet*
- Pepperl Fuchs Group (2004). *10****X series Incremental rotatory encoder. Manual*.
- Franklin, G. F., Powell, J.D., y M. Workman (1998). *Digital Control of Dynamic Systems*. Addison Wesley, Menlo Park, CA.
- Campbell, S. L., Chancellier, J. P. y R. Nikoukhah (2006). *Modeling and Simulation in Scilab/Scicos*. Springer.
- Atmel Corporation (2008). *8 bit Avr Microcontroller Atmega8535. Data sheet*.
- H. Rashid Muhammed (2004). *Power Electronics: circuits, devices and applications*. Prentice Hall.
- Coughlin, R. F. y F. F. Sudhoff (2001). *Operational amplifiers and linear integrated circuits*. Prentice Hall.
- Boylestad, R. L. y L. Nashelsky (2006). *Electronics devices and circuit theory*. Prentice Hall.