

# Implementación en tiempo real de un identificador neuronal discreto para un motor de inducción lineal

J. D. Rios\*, A. Y. Alanis\*, J. Rivera\*, M. Hernandez-Gonzalez\*\*

\* Universidad de Guadalajara

Apartado Postal 51-71, Col. Las Águilas, C.P. 45080

Zapopan, Jalisco, México

e-mail: almayalanis@gmail.com.

\*\* Universidad Autonoma de Nuevo Leon

Pedro de Alba s/n

San Nicolás de Los Garza, Nuevo León, México

**Resumen**—Este trabajo presenta un identificador neuronal discreto en tiempo real para la identificación de un motor de inducción lineal (MIL). El identificador está basado en una red neuronal recurrente de alto orden discreta entrenada en línea. El algoritmo de entrenamiento está basado en el filtro extendido de Kalman. Un observador de orden reducido es usado para estimar los flujos magnéticos del secundario. La implementación en tiempo real es realizada usando un MIL modelo LAB-Volt 8228, MATLAB/Simulink, la tarjeta controladora dSPACE DS1104 y las bibliotecas dSPACE RTI como interfaz de software. El rendimiento es mostrado con gráficas.

**Palabras clave:** Filtro extendido de Kalman, Motor de inducción lineal, Red neural recurrente de alto orden, identificación en tiempo real.

## I. INTRODUCCIÓN

Los motores de inducción lineal (MIL) pertenecen al grupo especial de máquinas eléctricas que convierten la energía eléctrica directamente en energía mecánica de movimiento translacional. El interés en este tipo de máquinas creció a principios de los años 70, sin embargo a finales de esa década el número de investigaciones y publicaciones disminuyó (Gieras, 1994). No fue hasta después de 1980 que aparecieron las primeras aplicaciones notables de los MIL, las cuales se encuentran en transporte, automatización, industria y electrodomésticos (Gieras, 1994), (Boldea y Nasar, 2005). Los MIL poseen características sobresalientes, como los son: alta fuerza inicial de empuje, no necesita de engranajes entre el motor y dispositivos móviles, reducción de pérdidas mecánicas, reducción del tamaño de los dispositivos de movimiento y alta velocidad de operación (Boldea y Nasar, 2005). Los principios de operación de los MIL son similares a los de los motores de inducción rotacional (MIR), pero el control y las características de los MIL son más complicados que en los MIR, debido a que en los MIL los parámetros varían con el tiempo por los cambios en las condiciones de operación (Benitez, 2002).

Por otro lado, los sistemas modernos de control requieren de un conocimiento estructurado del sistema a ser controlado; tal conocimiento debe ser representado en términos de ecuaciones diferenciales o ecuaciones en diferencias. A esta descripción matemática del sistema dinámico se le conoce como el modelo del sistema. Existen diferentes motivaciones para establecer una descripción matemática de sistemas dinámicos, por mencionar algunas: simulación, predicción, detección de fallos y diseño de sistemas de control.

Existen dos maneras para obtener el modelo del sistema; el primer método consiste en derivar el modelo a partir de las leyes de la física, pero en muchos casos el tiempo requerido es excesivo y es poco probable obtener un modelo preciso del sistema. El segundo método conocido como identificación de sistema, consiste en deducir el modelo a partir de un conjunto de datos obtenidos de un experimento práctico que lleve al sistema a través de su rango de operación y aunque no siempre resulta en un modelo preciso del sistema, se puede lograr un modelo satisfactorio con un esfuerzo razonable (Farrell y Polycarpou, 2006), (Norgaard, 2000).

Las redes neuronales artificiales se han establecido como una herramienta para resolver problemas de la ingeniería, por ejemplo: la identificación y control de sistemas no lineales complejos (Lewis *et al.*, 1999). Las redes neuronales que involucran elementos dinámicos en forma de conexiones de retroalimentación, son conocidas como redes neuronales recurrentes (Kosmatopoulos *et al.*, 1995).

Las redes neuronales recurrentes de alto orden ofrecen ventajas como su excelente capacidad de aproximación y robustez contra el ruido, que son ideales para el modelado de sistemas no lineales complejos (Lewis *et al.*, 1999). La principal técnica de entrenamiento para las redes neuronales recurrentes es el aprendizaje por retropropagación (Williams y Zipser, 1989), (Haykin, 2004). Además se han propuesto métodos de entrenamiento basados en el filtro de Kalman

(FK) (Sanchez *et al.*, 2008). Pero debido al hecho que el entrenamiento de una red neuronal comúnmente resulta en un problema no lineal, se hace uso del filtro de Kalman extendido (FKE) (Haykin, 2004).

El entrenamiento basado en el FKE para redes neuronales reduce el número de épocas y el número de neuronas requeridas, además mejora convergencia (Haykin, 2004) y ha demostrado ser confiable y práctico (Williams y Zipser, 1989).

Por otra parte, la tarjeta controladora dSPACE DS1104 R&D está diseñada para la implementación de controladores multivariados de alta velocidad y simulaciones en tiempo real. El panel conector RTI1104 da acceso a las señales de entrada y salida de la tarjeta. Las bibliotecas de software dSPACE RTI permiten la comunicación entre MATLAB/Simulink y la tarjeta controladora y provee funciones para monitorear las señales (dSPACE DS1104, 2009).

Este trabajo presenta el uso de una red neuronal recurrente de alto orden discreta entrenada en línea con un algoritmo basado en el FKE, para la identificación en tiempo real de un MIL modelo Lab-Volt 8228, utilizando la tarjeta controladora dSPACE DS1104 R&D, el software MATLAB/Simulink y las bibliotecas dSPACE RTI.

## II. PRELIMINARES MATEMÁTICOS

### II-A. Redes neuronales recurrentes de alto orden discretas

Considere el siguiente sistema no lineal con varias entradas y salidas:

$$x(k+1) = F(x(k), u(k)) \quad (1)$$

donde  $k$  es el tiempo de muestreo,  $k \in 0 \cup \mathbb{Z}^+$ ,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ ,  $F: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  es una función no lineal.

Es conocido que el uso de las redes neuronales multicapa para el reconocimiento de patrones y modelado de sistemas estáticos conduce a buenos resultados. Las redes neuronales son entrenadas para aprender un mapeo entrada – salida. Se ha probado que con el uso de una sola capa oculta se puede aproximar cualquier función continua sobre un dominio compacto (Haykin, 2004).

Para propósito de control, las redes neuronales recurrentes de alto orden presentan una mayor interacción entre las neuronas (Narendra y Parthasarathy, 1990), (Rovithakis y Christodoulou, 2000) y permiten la incorporación de información a priori de la estructura del sistema (Alanis *et al.*, 2006).

Considere la siguiente red neuronal recurrente de alto orden discreta:

$$x_i(k+1) = \omega_i^T z_i(x(k), u(k)) \quad (2)$$

donde  $x_i$  ( $i = 1, 2, \dots, n$ ) es el estado de la  $i$ -ésima neurona,  $\omega_i$  ( $i = 1, 2, \dots, n$ ) es el respectivo vector de pesos adapta-

dos en línea y  $z_i(x(k), u(k))$  es dado por

$$z_i(x(k), u(k)) = \begin{pmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{iL} \end{pmatrix} = \begin{pmatrix} \prod_{j \in I_1} \xi_{ij}^{d_{ij}^{(1)}} \\ \prod_{j \in I_2} \xi_{ij}^{d_{ij}^{(2)}} \\ \vdots \\ \prod_{j \in I_L} \xi_{ij}^{d_{ij}^{(L)}} \end{pmatrix} \quad (3)$$

$$\xi_i = \begin{pmatrix} \xi_{i1} \\ \vdots \\ \xi_{in} \\ \xi_{i_{n+1}} \\ \vdots \\ \xi_{i_{n+m}} \end{pmatrix} = \begin{pmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{pmatrix} \quad (4)$$

$$S(v) = \alpha \tanh(\beta v) \quad (5)$$

donde  $L$  es el número de conexiones de alto orden,  $\{I_1, I_2, \dots, I_L\}$  es una colección de conjuntos no ordenados de dimensión  $\{1, 2, \dots, n+m\}$ , con  $d_{ij}(k)$  siendo enteros no negativos y  $\xi_i$  es descrito por (4);  $u = [u_1, u_2, \dots, u_m]^T$  es el vector de entrada a la red neuronal y  $S(\cdot)$  descrito por (5) donde  $v$  es cualquier valor real.

Considere el problema de aproximar el sistema no lineal (1) con la red neuronal de alto orden discreta en configuración serie paralelo (6) (Rovithakis y Christodoulou, 2000), donde  $\chi_i$  es el  $i$ -ésimo estado de la planta,  $\epsilon_{z_i}$  es una aproximación acotada del error que puede ser reducido aumentando el número de pesos ajustables (Rovithakis y Christodoulou, 2000). Se supone que existe un vector de pesos ideal  $\omega_i^*$  tal que  $\|\epsilon_{z_i}\|$  puede ser minimizado sobre un conjunto compacto  $\Omega_{z_i} \subset \mathbb{R}^{L_i}$ . El vector de pesos ideal  $\omega_i^*$  es una entidad artificial requerida para propósito de análisis (Rovithakis y Christodoulou, 2000). En general, se considera que el vector existe y es constante, pero desconocido.

$$\chi_i(k+1) = \omega_i^T z_i(x(k), u(k)) + \epsilon_{z_i} \quad i = 1, \dots, n \quad (6)$$

### II-B. Algoritmo de entrenamiento basado en el filtro de Kalman extendido

El método de entrenamiento por retropropagación es el más utilizado para las redes neuronales recurrentes, sin embargo, es un método de gradiente descendiente de primer orden, por lo que su velocidad de aprendizaje puede llegar a ser lenta (C. Leunga, 2003). Recientemente, se han propuestos algoritmos de entrenamiento basados en el filtro de Kalman extendido (FKE) para el entrenamiento de redes neuronales (Alanis *et al.*, 2010), (Feldkamp *et al.*, 2003), los cuales mejoran la convergencia (C. Leunga, 2003), (Feldkamp *et al.*, 2003).

El filtro de Kalman (FK) estima el estado de un sistema lineal con ruido blanco aditivo (Brown y Hwang, 1997), (Song y Grizzle, 1992). En el entrenamiento basado en el FKE los pesos se convierten en los estados a estimar y el error entre la salida de la red neuronal y las mediciones de la planta se consideran ruido blanco aditivo. El uso del FKE es debido al hecho de que el mapeo de la red neuronal es no lineal (Poznyak *et al.*, 2001).

El objetivo del entrenamiento es encontrar los valores óptimos de pesos los cuales minimicen el error de predicción. El algoritmo de entrenamiento utilizado es el descrito por (7) (Williams y Zipser, 1989).

$$\begin{aligned} \omega_i(k+1) &= \omega_i(k) + \eta_i K_i(k) e_i(k) \\ K_i(k) &= P_i(k) H_i(k) M_i(k) \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^T(k) P_i(k) + Q_i(k) \end{aligned} \quad (7)$$

$$\begin{aligned} M_i(k) &= [R_i(k) + H_i^T(k) P_i(k) H_i(k)]^{-1} \\ e_i(k) &= x_i(k) - \chi_i(k) \end{aligned} \quad (8)$$

$$H_{ij} = \left[ \frac{\partial \chi_i(k)}{\partial \omega_{ij}(k)} \right]^T \quad (9)$$

donde  $i = 1, \dots, n$ ,  $e_i(k) \in \mathfrak{R}$  es el error de identificación,  $P_i(k) \in \mathfrak{R}^{L_i \times L_i}$  es la matriz de covarianza del error de estimación al paso  $k$ ,  $\omega_i \in \mathfrak{R}^{L_i}$  es el vector de pesos adaptado en línea,  $\chi_i$  es el  $i$ -ésimo estado de la red neuronal,  $K_i \in \mathfrak{R}^{L_i}$  es el vector de ganancias de Kalman,  $Q_i \in \mathfrak{R}^{L_i \times L_i}$  es la matriz de covarianzas del ruido de estimación,  $R_i \in \mathfrak{R}$  es la covarianza del ruido de medición,  $H_i \in \mathfrak{R}^{L_i}$  es un vector, el que cada entrada  $H_{ij}$  es la derivada del estado de la red neuronal ( $\chi_i$ ) con respecto a los pesos ( $\omega_{ij}$ ) y está dado por (9), donde  $i = 1, \dots, n$  y  $j = 1, \dots, L_i$ . Usualmente  $P_i$  y  $Q_i$  son inicializadas como matrices diagonales, con entradas  $P_i(0)$  y  $Q_i(0)$ , respectivamente. Es importante recordar que  $H_i(k)$ ,  $K_i(k)$  y  $P_i(k)$  para el FKE son acotados (Song y Grizzle, 1992).

### III. MODELO DEL MOTOR DE INDUCCIÓN LINEAL

El modelo  $\alpha - \beta$  del motor de inducción lineal es discretizado utilizando el método de Euler en (Benitez, 2002), (Kazantzis y c. Kravaris, 1999), (Hernandez-Gonzalez *et al.*, 2008) y se muestra a continuación

$$\begin{aligned} q_m(k+1) &= q_m(k) + v(k)T \\ v(k+1) &= (1 - k_2 T)v(k) - k_1 T \lambda_{r\alpha}(k) \rho_1 i_{s\alpha}(k) \\ &\quad - k_1 T \lambda_{r\beta}(k) \rho_2 i_{s\alpha}(k) \\ &\quad + k_1 T \lambda_{r\alpha}(k) \rho_2 i_{s\beta}(k) \\ &\quad - k_1 T \lambda_{r\beta}(k) \rho_1 i_{s\beta}(k) - k_3 T F_L \\ \lambda_{r\alpha}(k+1) &= (1 - k_6 T) \lambda_{r\alpha}(k) + k_4 T v(k) \rho_1 i_{s\alpha}(k) \\ &\quad - k_4 T \rho_1 i_{s\alpha}(k) + k_5 T \rho_2 i_{s\alpha}(k) \\ &\quad + k_4 T \rho_2 i_{s\beta}(k) - k_4 T v(k) \rho_2 i_{s\beta}(k) \\ &\quad + k_5 T \rho_1 i_{s\beta}(k) \\ \lambda_{r\beta}(k+1) &= (1 - k_6 T) \lambda_{r\beta}(k) + k_4 T v(k) \rho_2 i_{s\alpha}(k) \\ &\quad - k_4 T \rho_2 i_{s\alpha}(k) - k_5 T \rho_1 i_{s\alpha}(k) \\ &\quad - k_4 T \rho_1 i_{s\beta}(k) + k_4 T v(k) \rho_1 i_{s\beta}(k) \\ &\quad + k_5 T \rho_2 i_{s\beta}(k) \\ i_{s\alpha}(k+1) &= (1 + k_9 T) i_{s\alpha}(k) - k_7 T \lambda_{r\alpha}(k) \rho_2 \\ &\quad - k_8 T \lambda_{r\alpha}(k) v(k) \rho_1 + k_7 T \lambda_{r\beta}(k) \rho_1 \\ &\quad - k_8 T \lambda_{r\beta}(k) v(k) \rho_2 + k_{10} T u_\alpha(k) \\ i_{s\beta}(k+1) &= (1 + k_9 T) i_{s\beta}(k) + k_8 T \lambda_{r\alpha}(k) v(k) \rho_2 \\ &\quad - k_7 T \lambda_{r\alpha}(k) \rho_1 - k_7 T \lambda_{r\beta}(k) \rho_2 \\ &\quad - k_8 T \lambda_{r\beta}(k) v(k) \rho_1 - k_{10} T u_\beta(k) \end{aligned} \quad (10)$$

$$\begin{aligned} \rho_1 &= \sin(n_p q_m(k)) & \rho_2 &= \cos(n_p q_m(k)) & k_1 &= \frac{n_p L_{sr}}{D_m L_r} \\ k_2 &= \frac{R_m}{D_m} & k_3 &= \frac{1}{D_m} & k_4 &= n_p L_{sr} & k_5 &= \frac{R_r L_{sr}}{L_r} \\ k_6 &= \frac{R_r}{L_r} & k_7 &= \frac{L_{sr} R_r}{L_r (L_{sr}^2 - L_s L_r)} & k_8 &= \frac{L_{sr} n_p}{L_{sr}^2 - L_s L_r} \\ k_9 &= \frac{L_r^2 R_s + L_{sr}^2 R_r}{L_r (L_{sr}^2 - L_s L_r)} & k_{10} &= \frac{L_r}{L_{sr}^2 - L_s L_r} \end{aligned}$$

donde  $q_m(k)$  ( $m$ ) es la posición del sector secundario,  $v(k)$  ( $m/s$ ) es la velocidad lineal,  $\lambda_{r\alpha}$  y  $\lambda_{r\beta}$  ( $Wb$ ) son respectivamente los flujos  $\alpha$  y  $\beta$ ,  $i_{s\alpha}$  y  $i_{s\beta}$  ( $A$ ) son respectivamente las corrientes  $\alpha$  y  $\beta$  del primario,  $u_\alpha$  y  $u_\beta$  ( $V$ ) son respectivamente los voltajes  $\alpha$  y  $\beta$  del primario,  $R_s$  ( $\Omega$ ) es la resistencia en el primario por fase,  $R_r$  ( $\Omega$ ) es la resistencia en el secundario por fase,  $L_{sr}$  ( $H$ ) es la inductancia magnetizante por fase,  $L_s$  ( $H$ ) es la inductancia en el primario por fase,  $L_r$  ( $H$ ) es la inductancia en el secundario por fase,  $F_L$  ( $N$ ) es la perturbación en la carga,  $R_m$  ( $Kg/s$ ) es el coeficiente de fricción viscosa y pérdidas del hierro,  $D_m$  ( $Kg$ ) es la masa del sector secundario y  $n_p$  es el número de pares de polos,  $T$  ( $s$ ) es el periodo de muestreo (Benitez, 2002).

### IV. OBSERVADOR DE FLUJOS

Los flujos del secundario no pueden ser medidos, por lo que se hace uso de un observador de orden reducido (Sanchez *et al.*, 2008), (Castillo-Toledo *et al.*, 2008). Las dinámicas de los flujos en (10) pueden ser descritos como sigue (Hernández-González, 2008).

$$\Psi(k+1) = \Psi(k) - k_6 T \Psi(k) - k_4 T \Theta^T J I_s(k) \quad (11)$$

$$+ k_4 T \Theta^T J I_s(k) v(k) + k_4 T \Theta^T I_s(k)$$

con

$$\Theta(k) = \begin{pmatrix} \cos(n_p q_m) & -\sin(n_p q_m) \\ \sin(n_p q_m) & \cos(n_p q_m) \end{pmatrix}$$

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$\Psi(k) = \begin{pmatrix} \psi_{r\alpha}(k) \\ \psi_{r\beta}(k) \end{pmatrix}$$

$$I(k) = \begin{pmatrix} I_{s\alpha}(k) \\ I_{s\beta}(k) \end{pmatrix}$$

El observador propuesto en [27] es (12)

$$\tilde{\Psi}(k+1) = \tilde{\Psi}(k) - k_6 T \tilde{\Psi}(k) - k_4 T \Theta^T J I_s(k) \quad (12)$$

$$+ k_4 T \Theta^T J I_s(k) v(k) + k_4 T \Theta^T I_s(k)$$

Las pruebas de convergencia del observador pueden ser encontradas en (Sanchez *et al.*, 2008), (Castillo-Toledo *et al.*, 2008), (Hernández-González, 2008).

## V. IDENTIFICACIÓN NEURONAL

El identificador neuronal discreto propuesto basado en una red neuronal recurrente de alto orden para el modelo (10) del MIL se define como

$$\begin{aligned} \chi_1(k+1) &= \omega_{11}(k)S(v(k)) + \omega_{12}(k)S(\psi_{r\alpha}(k)) \\ &+ \omega_{13}(k)S(\psi_{r\beta}(k)) \\ &- \omega_{14}(k)S(\psi_{r\alpha}(k))\rho_1 \dot{i}_{s\alpha}(k) \\ &- \omega_{14}(k)S(\psi_{r\beta}(k))\rho_2 \dot{i}_{s\alpha}(k) \\ &+ \omega_{15}(k)S(\psi_{r\alpha}(k))\rho_2 \dot{i}_{s\beta}(k) \\ &- \omega_{15}(k)S(\psi_{r\beta}(k))\rho_1 \dot{i}_{s\beta}(k) \\ \chi_2(k+1) &= \omega_{21}(k)S(v(k)) + \omega_{22}(k)S(\psi_{r\alpha}(k)) \\ &+ \omega_{23}(k)S(\psi_{r\beta}(k)) + \omega_{24}(k)\rho_2 \dot{i}_{s\alpha}(k) \\ &+ \omega_{25}(k)\rho_1 \dot{i}_{s\beta}(k) \\ \chi_3(k+1) &= \omega_{31}(k)S(v(k)) + \omega_{32}(k)S(\psi_{r\alpha}(k)) \\ &+ \omega_{33}(k)S(\psi_{r\beta}(k)) - \omega_{34}(k)\rho_1 \dot{i}_{s\alpha}(k) \\ &+ \omega_{35}(k)\rho_2 \dot{i}_{s\beta}(k) \\ \chi_4(k+1) &= \omega_{41}(k)S(v(k)) + \omega_{42}(k)S(\psi_{r\alpha}(k)) \\ &+ \omega_{43}(k)S(\psi_{r\beta}(k)) + \omega_{44}(k)S(\dot{i}_{s\alpha}(k)) \\ &+ \omega_{45}(k)u_\alpha(k) \\ \chi_5(k+1) &= \omega_{51}(k)S(v(k)) + \omega_{52}(k)S(\psi_{r\alpha}(k)) \\ &+ \omega_{53}(k)S(\psi_{r\beta}(k)) + \omega_{54}(k)S(\dot{i}_{s\beta}(k)) \\ &+ \omega_{55}(k)u_\beta(k) \\ \chi_6(k+1) &= \omega_{65}S(q_m(k)) + \omega_{66}v(k) \end{aligned} \quad (13)$$

donde  $\chi_1(k)$  identifica la velocidad lineal,  $\chi_2(k)$  y  $\chi_3(k)$  identifican los flujos  $\alpha$  y  $\beta$ ,  $\chi_4(k)$  y  $\chi_5(k)$  identifican las corrientes  $\alpha$  y  $\beta$  en el primario,  $\chi_6(k)$  identifica la posición.

Los pesos  $\omega_{ij}(k)$  son actualizados por (7),  $S(\cdot)$  es descrita por (5), los pesos  $\omega_{14}(k)$ ,  $\omega_{15}(k)$ ,  $\omega_{24}(k)$ ,  $\omega_{25}(k)$ ,  $\omega_{34}(k)$ ,  $\omega_{35}(k)$ ,  $\omega_{45}(k)$  y  $\omega_{55}(k)$  son fijos, además  $\omega_{24}(k) = \omega_{25}(k) = \omega_{34}(k) = \omega_{35}(k) = \omega_f$  es una constante,  $\alpha$  y  $\beta$  son parámetros de diseño.

Es importante notar que para el identificador (13) el modelo matemático (10) es considerado desconocido.

## VI. PRUEBA EN TIEMPO REAL

Para la prueba en tiempo real el modelo en MATLAB/Simulink (Figura 1) es cargado en la tarjeta controladora dSPACE1104 (Figura 2) con ayuda de las librerías RTI1104 como interfaz, El motor de inducción lineal utilizado es el modelo LAB-Volt 8228 (Figura 3). Las entradas al MIL son una señal variante en frecuencia proporcionada por el autotransformador (Figura 4) y una señal de control PWM, ambas señales pasan por el módulo de potencia (Figura 5) que permite el manejo de los voltajes y corrientes necesarias para mover la parte móvil del MIL. Todas las entradas y salidas a la tarjeta controladora pasan por el panel conector RTI1104 (Figura 6).

Las corrientes  $\alpha$  y  $\beta$  son obtenidas a partir de las corrientes reales del MIL convertidas al modelo  $\alpha$  y  $\beta$  [4], la posición y la velocidad son obtenidas del encoder de precisión lineal SENC 150 que se encuentra montado sobre el MIL.

Los valores fijos son:  $\omega_{14} = 0,001$ ,  $\omega_{15} = 0,001$ ,  $\omega_f = 0,001$ ,  $\omega_{45} = 0,02178$  y  $\omega_{55} = 0,02178$ .

Los parámetros de entrada al observador son: la posición, la velocidad y las corrientes para obtener una medición de los flujos  $\alpha$  y  $\beta$ . Posteriormente, las mediciones de la posición, velocidad, corrientes y flujos son las entradas al identificador que es entrenado en línea por el algoritmo basado en el FKE (7) para ajustar los pesos  $\omega_{ij}(k)$  en cada instante de tiempo  $k$ .

Los errores de identificación de velocidad, flujo  $\alpha$ , flujo  $\beta$ , corriente  $\alpha$ , corriente  $\beta$  y posición entre el identificador (13) y el MIL son mostrados en las Figuras 7 a 12. La tabla I muestra los errores medios cuadráticos (EMC) entre el identificador neuronal y las señales de los estados del MIL. La prueba en tiempo real se realiza con un tiempo de muestreo de  $k = 0,3 \text{ ms}$  con un tiempo total de 11 s.

TABLA I  
ERRORES MEDIO CUADRÁTICO DE IDENTIFICACIÓN DEL MIL

Error de identificación	EMC
Velocidad	0,0089
Flujo $\alpha$	5,2903e - 005
Flujo $\beta$	8,3943e - 005
Corriente $\alpha$	0,2063
Corriente $\beta$	0,1657
Posición	1,4944e - 005

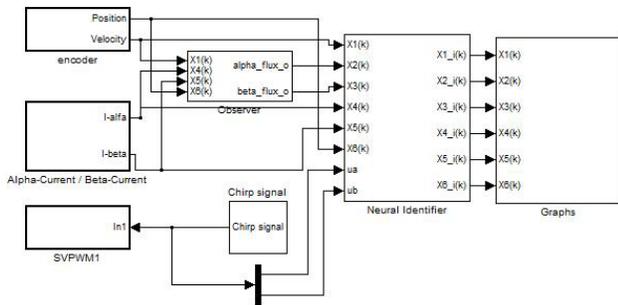


Figura 1. Modelo MATLAB/Simulink.



Figura 2. Tarjeta controladora dSPACE DS1104.



Figura 3. Motor de inducción lineal LAB Volt 8228.



Figura 4. Autotransformador.

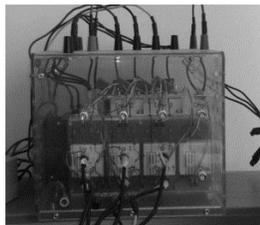


Figura 5. Módulo de potencia.

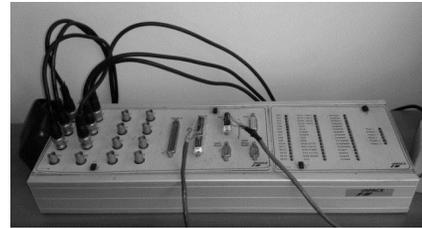


Figura 6. Panel conector RTI 1104.

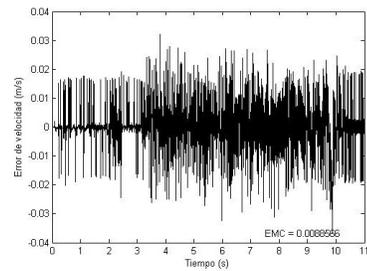


Figura 7. Error de identificación de la velocidad.

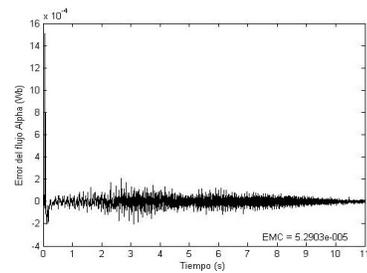


Figura 8. Error de identificación del flujo  $\alpha$ .

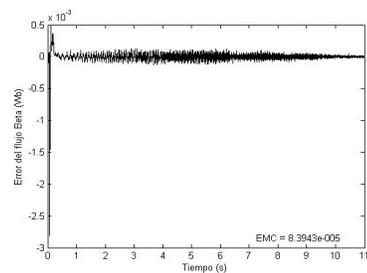


Figura 9. Error de identificación del flujo  $\beta$ .

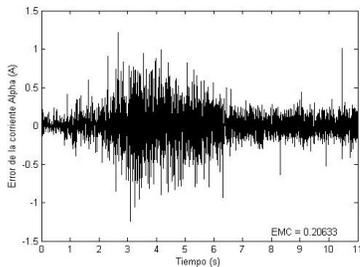


Figura 10. Error de identificación de la corriente  $\alpha$ .

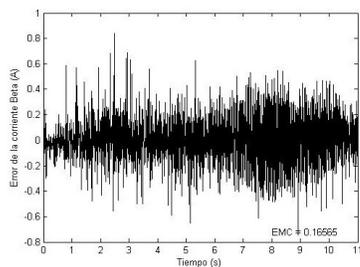


Figura 11. Error de identificación de la corriente  $\beta$ .

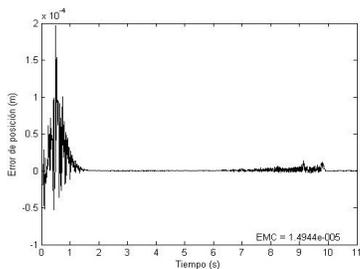


Figura 12. Error de identificación de la posición.

## VII. CONCLUSIONES

Este trabajo propone el uso de una red neuronal recurrente de alto orden discreta para la identificación en tiempo real de un motor de inducción lineal. El identificador neuronal propuesto es entrenado en línea con un algoritmo basado en el filtro de Kalman extendido y es implementado en tiempo real utilizando la tarjeta controladora dSPACE DS1104, con lo que se muestra su aplicabilidad. Actualmente, los autores trabajan en mejorar el rendimiento del identificador y en un esquema de control para el motor de inducción lineal.

## VIII. AGRADECIMIENTOS

Los autores agradecen a CONACYT por su apoyo a través del proyecto 103191Y.

## REFERENCIAS

Alanis, Alma Y., Edgar N. Sanchez, Alexander G. Loukianov y M.A. Perez-Cisneros (2010). Real-time discrete neural block control using sliding modes for electric induction motors. *Control Systems Technology, IEEE Transactions on* **18**(1), 11–21.

- Alanis, Alma Y., Edgar N. Sanchez y Alexander G. Loukianov (2006). Discrete-time recurrent neural induction motor control using kalman learning. En: *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. pp. 1993–2000.
- Benitez, V. H. (2002). Control neuronal a bloques: Aplicación a un motor de inducción de desplazamiento lineal. Tesis de maestría. Centro de Investigación y Estudios Avanzados del IPN. Guadalajara, México.
- Boldea, I. y S.A. Nasar (2005). *Linear Electric Actuators and Generators*. Cambridge University Press.
- Brown, R.G. y P.Y.C. Hwang (1997). *Introduction to random signals and applied Kalman filtering*. número v. 1 En: *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions*. Wiley.
- C. Leunga, y L. Chan (2003). Dual extended kalman filtering in recurrent neural networks. *Neural Networks* **6**, 223–239.
- Castillo-Toledo, B., S. Di Gennaro, A.G. Loukianov y J. Rivera (2008). Discrete time sliding mode control with application to induction motors. *Automatica* **44**(12), 3036 – 3045.
- dSPACE DS1104 (2009). *Hardware Installation and Configuration and ControlDesk Experimental Guide*. DSPACE GmbH. London.
- Farrell, J.A. y M.M. Polycarpou (2006). *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley.
- Feldkamp, L.A., D.V. Prokhorov y T.M. Feldkamp (2003). Conditioned adaptive behavior from kalman filter trained recurrent networks. En: *Neural Networks, 2003. Proceedings of the International Joint Conference on*. Vol. 4. pp. 3017–3021 vol.4.
- Gieras, J.F. (1994). *Linear induction drives*. Monographs in electrical and electronic engineering. Oxford University Press on Demand.
- Haykin, S. (2004). *Kalman Filtering and Neural Networks*. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley.
- Hernández-González, M. (2008). Control Neuronal Discreto del Motor de Inducción de Desplazamiento Lineal. Tesis de maestría. Centro de Investigación y Estudios Avanzados del IPN. Guadalajara, México.
- Hernandez-Gonzalez, M., E.N. Sanchez y A.G. Loukianov (2008). Discrete-time neural network control for a linear induction motor. En: *Intelligent Control, 2008. ISIC 2008. IEEE International Symposium on*. pp. 1314–1319.
- Kazantzis, N. y c. Kravaris (1999). Time-discretization of nonlinear control systems via taylor methods. *Computer and Chemical Engineering* **23**, 763–784.
- Kosmatopoulos, E.B., M.M. Polycarpou, M.A. Christodoulou y P.A. Ioannou (1995). High-order neural network structures for identification of dynamical systems. *Neural Networks, IEEE Transactions on* **6**(2), 422–431.
- Lewis, F.L., S. Jagannathan y A. Yesildirek (1999). *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Series in Systems and Control. Taylor & Francis.
- Narendra, K.S. y K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *Neural Networks, IEEE Transactions on* **1**(1), 4–27.
- Norgaard, M. (2000). *Neural Networks for Modelling and Control of Dynamic Systems.: A Practitioner's Handbook*. Advanced textbooks in control and signal processing. Springer Verlag.
- Poznyak, A.S., W. Yu y E.N. Sanchez (2001). *Differential neural networks for robust nonlinear control: identification, state estimation and trajectory tracking*. World Scientific Publishing Company Incorporated.
- Rovithakis, G.A. y M.A. Christodoulou (2000). *Adaptive Control with Recurrent High-order Neural Networks: Theory and Industrial Applications*. Advances in Industrial Control Series. Springer London, Limited.
- Sanchez, E.N., A.Y. Alanis y A.G. Loukianov (2008). *Discrete-Time High Order Neural Control: Trained with Kalman Filtering*. Studies in Computational Intelligence. Springer.
- Song, Yongkyu y J.W. Grizzle (1992). The extended kalman filter as a local asymptotic observer for nonlinear discrete-time systems. En: *American Control Conference, 1992*. pp. 3365–3369.
- Williams, R. J. y D. Zipser (1989). Neural computation,. *Neural Computation* **1**, 270–280.