

# Identificación Neuronal de un Generador Síncrono de Imanes Permanentes

Marlo A. Medina

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional  
Av. del Bosque 1145, colonia el Bajío, Zapopan , 45019, Jalisco, México.  
E-mail: Amedina@gdl.cinvestav.mx

Edgar N. Sánchez

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional  
Av. del Bosque 1145, colonia el Bajío, Zapopan , 45019, Jalisco, México.

Luís J. Ricalde

Universidad Autónoma de Yucatán

Av. Industrias no Contaminantes por Periférico Norte Apdo. Postal 150 Cordemex Mérida, Yucatán, México

**Resumen**—En éste trabajo, se presenta un identificador basado en una Red Neuronal Recurrente de Alto Orden Tipo Onduleta (RHONNW) Discreta, con entrenamiento basado en el algoritmo de Filtro de Kalman Extendido (FKE). El identificador, se aplico a un Generador Síncrono de Imanes Permanentes (GSIP), obteniendo resultados de simulación con errores despreciables.

**Palabras clave:** Red Neuronal Recurrente, Identificador Neuronal, Filtro de Kalman, Generador Síncrono de Imanes Permanentes.

## I. INTRODUCCIÓN

El no conocer el modelo matemático de un sistema no lineal, ha llevado a la construcción de identificadores que ofrezcan identificaciones rápidas y aproximaciones más exactas a dichos sistemas. Las redes neuronales, han sido ampliamente utilizadas para aproximar funciones; Por ello son muy útiles para la identificación de sistemas no lineales. Un Identificador Neuronal no es difícil de programar, ofrece robustez, sus aproximaciones son rápidas y muy exactas.

Las funciones tipo onduleta, tienen grandes propiedades como: escalamiento y traslación, que combinadas con las RHONNW discretas ofrecen grandes ventajas para de identificación.

Debido a las grandes ventajas de los GSIP, se a aumentado su uso en los aerogeneradores por lo que en este trabajo, se identifican el comportamiento de las señales que se pueden utilizar para poder aplicar una ley de control.

## II. PRELIMINARES MATEMÁTICAS

### II-A. Red Neuronal Recurrente de Alto Orden Tipo Onduleta Discreta

Las Redes Neuronales Recurrentes de Alto Orden (RHONN) son usadas por que presentan mayores interacciones entre neuronas, como las presentadas en (Narendra y parthasarathy, 1990) y (Rovithakis y Chrisodoulou, 2000). El modelo de una RHONN es muy flexible y permite incorporar

al modelo neuronal la información a priori acerca de la estructura del sistema.

La diferencia entre una RHONN y una RHONNW, es que la función de activación de la neurona es una función tipo onduleta.

Considérese la siguiente RHONNW en tiempo discreto:

$$x_i(k+1) = w_i^T z_i(x(k), u(k)), \quad i = 1, \dots, n \quad (1)$$

donde  $x_i(k)$  es el estado de la  $i$ -ésima neurona en la iteracion  $k$  con  $i = 1, \dots, n$ , siendo  $n$  la dimension del estado;  $w_i$  es el vector de pesos adaptables en linea y  $z_i(x(k), u(k))$  esta definida como:

$$z_i(x, u) = \begin{bmatrix} z_{I_1} \\ z_{I_2} \\ \vdots \\ z_{I_L} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_j^{d_j(1)} \\ \prod_{j \in I_2} y_j^{d_j(2)} \\ \vdots \\ \prod_{j \in I_L} y_j^{d_j(L)} \end{bmatrix} \quad (2)$$

donde  $d_j(k)$  son enteros no negativos;  $L$  es el numero de conexiones de alto orden;  $\{I_1, I_2, \dots, I_L\}$  es una colección de  $L$  subconjuntos no ordenados  $\{1, 2, \dots, m+n\}$ ; e  $y$  es un vector construido por las entradas a cada neurona, definido como:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ S(u_1) \\ \vdots \\ S(u_m) \end{bmatrix} \quad (3)$$

con  $u = [u_1 u_2 \dots u_m]^T$  como el vector de entradas a la red neuronal.

$S(\bullet)$  es una función "wavelet" que se define por:

$$\psi_{a,b}(x) = \psi\left(\frac{x-b}{a}\right) \quad (4)$$

donde  $a$  y  $b$ , son parámetros de escalamiento y traslación respectivamente.

En este trabajo, se manejan varias "wavelets" pero en especial la llamada sombrero mexicano definida como:

$$w_{avemex}(x) = \frac{2}{\sqrt{3}}\pi^{1/4}(1-x^2)\exp\left(\frac{-x^2}{2}\right) \quad (5)$$

donde como en (4) se aplicaran los parámetros  $a$  y  $b$ , con valores constantes.

Consideremos el problema de aproximar un sistema no lineal discreto de la forma:

$$x(k+1) = F(x(k), u(k)) \quad (6)$$

donde  $x \in \mathbb{R}^n, u \in \mathbb{R}^m$  y  $F \in \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  es una función no lineal.

Se aproximara con una RHONNW es su representación serie-paralelo (Rovithakis y Chrisodoulou, 2000), (Ramon y Felix, 2003):

$$\hat{x}(k+1) = w_1^* z_i(x(k), u(k)) + \epsilon_{zi}, \quad i = 1, \dots, n \quad (7)$$

donde  $x_i (i = 1, \dots, n)$  es el  $i$ -ésimo estado de la planta,  $\epsilon_{zi}$  es el error de aproximación acotado, el cual puede ser reducido al incrementar el numero de pesos adaptables (Rovithakis y Chrisodoulou, 2000). Asumimos que existe un vector de pesos ideales  $w_i^*$  tal que  $\|\epsilon_{zi}\|$  puede ser minimizado sobre un conjunto compacto  $\Omega_{zi} \subset \mathbb{R}$ . El vector de pesos ideal es una cantidad artificial requerida para propósito analítico (Rovithakis y Chrisodoulou, 2000). En general se asume que este vector existe y es constante pero desconocido. Se define el vector de pesos estimado como  $w_i$  y el error de estimación entre pesos como

$$\tilde{w}_i(k) = w_i^* - w_i(k) \quad (8)$$

El vector de pesos estimado  $w_i$  es usado para el análisis de estabilidad que será discutido después. Puesto que  $w_i^*$  es constante, entonces

$$\tilde{w}_i(k+1) - \tilde{w}_i(k) = w_i(k+1) - w_i(k) \quad \forall k \in 0 \cup \mathbb{Z}^+ \quad (9)$$

## II-B. FKE Algoritmo de Entrenamiento

Recientemente, algoritmos basados en el FKE han sido introducidos para entrenar redes neuronales (Feldkamp y Prokhorov, 2003), (Sánchez y Alanis, 2004), asegurando la convergencia de aprendizaje (Leung y Chan, 2003). El entrenamiento de redes neuronales por medio de FKE, tanto para hacia adelante y para redes recurrentes, ha sido probado y éstas redes han sido prácticas para muchas aplicaciones desde hace diez años (Feldkamp y Prokhorov, 2003).

Es conocido que el Filtro de Kalman (KF) estima el estado de un sistema lineal con ruido aditivo en el estado y la salida (Brown y Hwang, 1997), (Felix y Sánchez, 2005). Para el entrenamiento de la red neuronal (NN) basado en el FKE, los pesos de la red se convierten en los estados a ser estimados. En este caso, el error entre la salida de la NN y la salida de

la planta se considera como ruido blanco aditivo. Debido a hecho de que el mapeo de la NN es no lineal, es necesario utilizar un FKE.

La meta de entrenamiento es encontrar los valores óptimos de los pesos, los cuales minimizan el error de predicción. El algoritmo de entrenamiento basado en el FKE está descrito por (Rovithakis y Chrisodoulou, 2000) y se describe a continuación

$$M_i(k) = [R_i(k) + H_i^T(k)P_i(k)H_i(k)]^{-1}$$

$$K_i(k) = P_i(k)H_i(k)M_i(k)$$

$$w_i(k+1) = w_i(k) + \eta_i K_i(k) e_i(k) \quad (10)$$

$$P_i(k+1) = P_i(k) - K_i(k)H_i^T(k)P_i(k) + Q_i(k)$$

con

$$e_i(k) = [y(k) - \tilde{y}(k)] \quad (11)$$

donde  $e - i(k)$  es el error de estimación,  $P_i(k)$  es la matriz de covarianza del error de predicción asociado,  $w_i(k)$  es el vector de peso (estado),  $L_i$  es el numero total de pesos de la red neuronal.  $\eta_i$  es el parámetro de diseño tal que  $0 \leq \eta_i \leq 1$ ;  $y(k)$  es el vector de salida medido,  $\hat{y}(k)$  es el estado de la red neuronal,  $n$  es el número de estados,  $K_i(k)$  es la matriz de ganancia de Kalman,  $Q_i(k)$  es la matriz de covarianza del estado de medida del ruido,  $R_i(k)$  es la matriz de covarianza del ruido en el estado,  $H_i(k)$  es una matriz, en la cual cada entrada de  $H_i^j$  es la derivada de la salida de la red neuronal  $\hat{y}(k)$ , con respecto a los pesos ajustables  $w_{ij}$ , como sigue

$$H_{ij}(k) = \left[ \frac{\partial \hat{y}_{ij}(k)}{\partial w_{ij}(k)} \right]_{w(k)=\hat{w}(k+1)} \quad (12)$$

donde  $i = 1, \dots, m$  y  $j = 1, \dots, L_1$

El algoritmo FKE es usado sólo para entrenar los pesos de la red neuronal, los cuales vienen a ser los estados estimados por el FKE.

## III. IDENTIFICACIÓN

En esta sección, retomaremos el problema de identificación de un sistema no lineal discreto (6). Para la identificación utilizamos RHONNW definida en (1). El identificador sera entrenado con el FKE.

La dinámica de (11) puede ser expresada por

$$e_1(k+1) = \tilde{w}_i(k) z_i(x(k), u(k)) + \epsilon_{zi} \quad (13)$$

por otro lado la dinámica de (8) es

$$\tilde{w}_i(k+1) = \tilde{w}_i(k) - \eta_i K_i(k) e_i(k) \quad (14)$$

Teorema 1. (Sánchez y Alanis, 2008). La red neuronal recurrente de alto orden (1), entrenada con el algoritmo basado en el filtro de Kalman extendido definido en la sección II-B, para identificar la planta no lineal (6), asegura que el error de identificación (11) es semiglobalmente uniformemente últimamente acotado; por otra parte los pesos de la red neuronal recurrente de alto orden deben permanecer acotados.

La prueba de este teorema puede verse en (Sánchez y Alanis, 2008).

#### IV. GENERADOR SÍNCRONO DE IMANES PERMANENTES

El modelo matemático reducido del GSIP, habiendo utilizado la transformada de la park y considerando, que el sistema esta balanceado tenemos las siguientes ecuaciones

$$\dot{i}_d = [v_d - Ri_d + L_q(pw_r i_q)] \frac{1}{L_d} \quad (15)$$

$$\dot{i}_q = [v_q - Ri_q - L_d(pw_r i_d) - \lambda pw_r] \frac{1}{L_q} \quad (16)$$

$$\dot{w}_r = [Te - Fw_r - T_m] \frac{1}{J} \quad (17)$$

donde  $i_d$  = corriente de eje directo,  $i_q$  = corriente de eje en cuadratura,  $w_r$  = velocidad angular del rotor,  $T_e$  = torque Electromagnético,  $T_m$  = torque mecánico  $J$  = inercia del rotor,  $P$  = número de pares de polos,  $\lambda$  = flujo inducido por los imanes permanentes,  $L_d$  = inductancia de eje directo,  $L_q$  = inductancia del eje en cuadratura,  $v_q$  = voltaje de eje en cuadratura y  $v_d$  = voltaje de eje directo.

#### V. IDENTIFICADOR NEURONAL

El identificador neuronal propuesto es

$$x_1(k+1) = w_{11}S(x_1(k)) + w_{12}S(x_2(k)) + w_{13}vd + w_{14}S(vd) + w_{15}S(vd)^2 + w_{16}S(x_1(k))^2 + w_{17}S(x_3(k))S(x_2(k))$$

$$x_2(k+1) = w_{21}S(x_2(k)) + w_{22}S(x_3(k))S(x_2(k)) + w_{23}S(vq) + w_{24}S(x_2(k))^2 + w_{25}vq + w_{26}S(vq)^2$$

$$x_3(k+1) = w_{31}S(x_3(k)) + w_{32}S(x_3(k))^2 + w_{34}S(vd) + w_{33}S(x_2(k))^2 + w_{35}S(vd)^2 + w_{36}vd + w_{37}S(x_3(k))S(x_2(k)) + w_{38}(x_1(k))^2$$

Los parámetros del algoritmo de entrenamiento FKE son

$$\begin{aligned} P_1(0) &= P_2(0) = P_3(0) = 1 \times 10^9 \\ Q_1(0) &= Q_2(0) = Q_3(0) = 2 \times 10^{15} \\ R_1(0) &= R_2(0) = R_3(0) = 2 \times 10^{15} \\ w_1(0) &= w_2(0) = w(0) = [0000] \\ \eta &= 1 \end{aligned}$$

Los valores de escalamiento y traslación son  $a = 1.2$  y  $b = 0$  respectivamente. Las condiciones iniciales del GSIP son  $i_q(0) = .1$ ,  $i_d(0) = .1$ ,  $w_r(0) = .1$ .

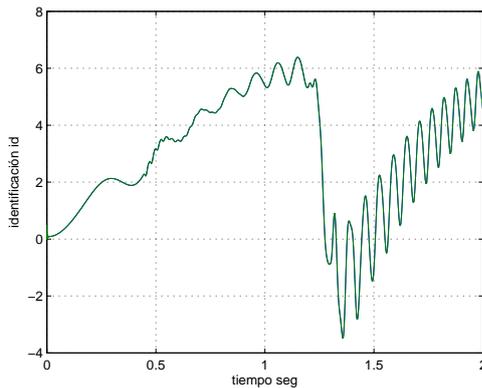


Fig1. Identificación del estado  $i_d$

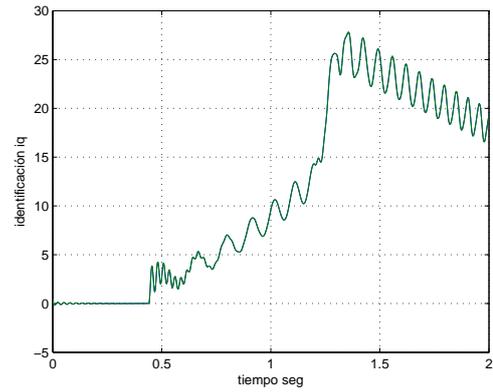


Fig2. Identificación del estado  $i_q$

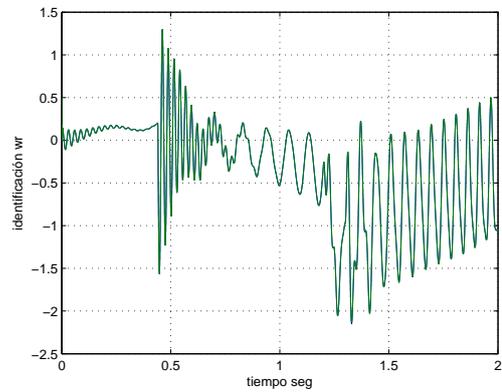


Fig3. Identificación del estado  $w_r$

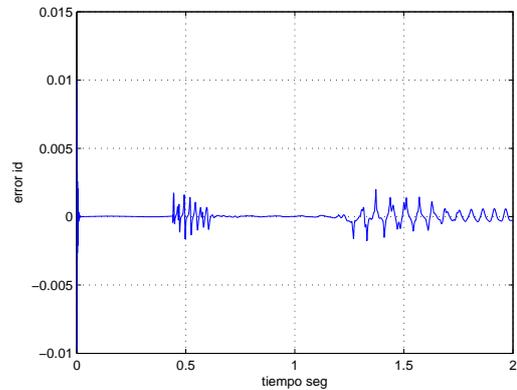


Fig4. Error de identificación de estado  $i_d$

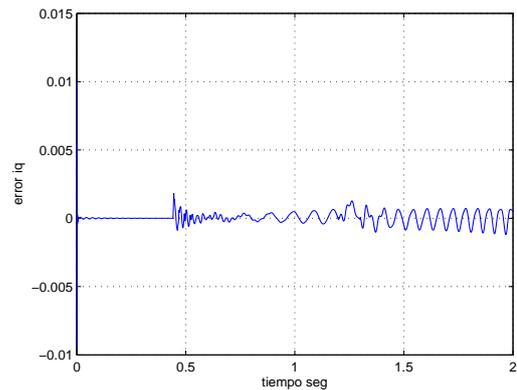


Fig5. Error de identificación de estado  $i_q$

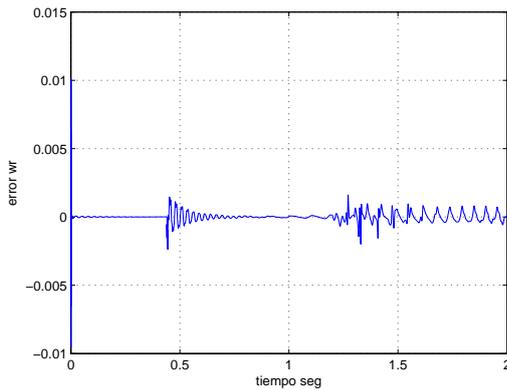


Fig6. Error de identificación de estado  $w_r$

## VI. RESULTADOS DE LA SIMULACIÓN

La siguiente tabla, presenta resultados de funciones de activación tipo onduleta y una clásica donde los valores presentados, es el máximo error de identificación

	$i_d$	$i_q$	$w_r$
wavemex	$1,9 \times 10^{-3}$	$1,5 \times 10^{-3}$	$1,5 \times 10^{-3}$
morlet	$9 \times 10^{-3}$	$6 \times 10^{-3}$	$6 \times 10^{-3}$
tanh	$14 \times 10^{-3}$	$14 \times 10^{-3}$	2

Tabla 1. resultado de identificación con una velocidad con variaciones grandes.

	$i_d$	$i_q$	$w_r$
wavemex	$3 \times 10^{-5}$	$7,3 \times 10^{-5}$	$6,5 \times 10^{-5}$
morlet	$1,5 \times 10^{-4}$	$3,1 \times 10^{-4}$	$2,9 \times 10^{-4}$
tanh	$13 \times 10^{-3}$	$14 \times 10^{-3}$	3,5

Tabla 2. resultado de identificación con una velocidad con variaciones medias.

## VII. CONCLUSIONES

Los resultados obtenidos en la identificación de los estados del GSIP, son muy exactos y la convergencia es rápida; Por lo que la identificación presentada en este trabajo, se puede utilizar para aplicar una ley de control.

## VIII. AGRADECIMIENTOS

Los autores agradecemos el apoyo de Conacyt México, al proyecto 131678. También agradecemos los útiles comentarios de los revisores anónimos que ayudaron a mejorar el artículo.

## REFERENCIAS

- [1] R. Brown and P. Hwang, (1997). *Introduction to Random Signals and applied Kalman filtering*, volume 2, John Wiley and Sons.
- [2] L. Feldkamp, D. Prokhorov, and T. Feldkamp,(2003). *Simple and conditioned adaptive behavior from kalman filter trained recurrent networks\* 1*. Neural networks, 16(5-6):683689.
- [3] R. Felix, E. Sánchez, and A. Loukianov. *Avoiding controller singularities in adaptiverecurrent neural control*. In Proceedings IFAC, volume 5.
- [4] C. Leung and L. Chan, (2003). *Dual extended kalman filtering in recurrent neural networks1*. Neural networks, 16(2):223239.

- [5] B. Muirhead,(1997). *Mars pathfinder flight system integration and test*. In AerospaceConference, 1997. Proceedings, IEEE, volume 4, pages 191205. IEEE.
- [6] K. S. Narendra and K. Parthasarathy,(1990). *Identification and control of dynamical systems using neural networks*. IEEE Transactions on Neural Networks, 1(1):427.
- [7] G. Rovithakis and M. Christodoulou, (2000). *Adaptive control with recurrent high-order neural networks: theory and industrial applications*. Springer.
- [8] E. Sánchez, A. Alanis, A. García, and A. Loukianov, (2008). *Discrete-time high order neuralcontrol: trained with Kalman filtering*, volume 112. Springer Verlag.
- [9] E. N. Sánchez, A. Y. Alanis, and J. Rico, (2004). *Electric load demand prediction using neuralnetwork trained by kalman filtering*. In Proc. IEEE Int Neural Networks Joint Conf, volume 4, pages 27712775.
- [10] A. Ramon and C. Felix, (2003). *Variable structure neural control*. PhD thesis, Ph. D. dissertation, CINVESTAV, available: www. gdl. cinvestav. mx.