

Identificación y Control *Wavenet* de un Motor de AC

O. Islas-Gómez *, L. E. Ramos-Velasco, J. García-Lamont y J.C. Ramos Fernández**

Universidad Autónoma del Estado de Hidalgo, UAEH

Centro de Investigación en Tecnologías de la Información y Sistemas, CITIS

Carr. Pachuca-Tulancingo Km. 4.5, Pachuca, Hidalgo, México.

e-mails: {is104078, lramos, jglamont}@uaeh.edu.mx

**Universidad Politécnica de Pachuca, Carretera Pachuca-Cd. Sahagún, Km. 20, Rancho Luna, Ex-Hacienda de Sta. Bárbara, Municipio de Zempoala, Hidalgo, México.

e-mail: jramos@uaeh.edu.mx

Resumen: En el presente artículo se muestra un esquema de control que sintoniza en línea las ganancias proporcional, integral y derivativa de un controlador PID discreto, mediante la identificación de un sistema dinámico SISO a controlar de modelo matemático desconocido. Esto se logra empleando una red neuronal de base radial con funciones de activación wavelet hijas Morlet (*wavenet*) y un filtro de respuesta infinita al impulso (IIR) en cascada. Dicho esquema es aplicado para controlar la velocidad de un motor de inducción de CA trifásico del tipo jaula de ardilla, se presentan los resultados simulados y experimentales obtenidos, empleando para esto la plataforma de LabVIEW.

Palabras clave: Motor de inducción de CA, PID, *wavenet*.

I. INTRODUCCIÓN

Los controladores más utilizados actualmente en la industria son los controladores PID o alguna de sus modificaciones. Sin embargo, el algoritmo PID lineal es difícil de utilizar cuando el proceso a controlar presenta dinámicas complejas, zonas muertas y características altamente no lineales. El funcionamiento del controlador PID en general se basa en actuar en forma proporcional, integral y derivativa sobre la señal de error $e(t)$, definida como la diferencia entre la señal de referencia $y_{ref}(t)$ y la señal de salida del proceso $y(t)$, con la finalidad de efectuar la señal de control $u(t)$ que manipula en forma deseada la salida del proceso, como se ve en la Figura 1.

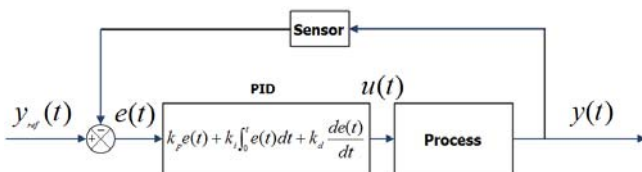


Figura 1: Esquema del controlador PID.

donde k_p , k_i y k_d son las ganancias del controlador. Existen distintas técnicas analíticas y experimentales con el fin de sintonizar dichas ganancias [2]. Una alternativa es auto-sintonizar en línea dichas ganancias como en [4, 5, 15]

*Islas-Gómez es actualmente estudiante de la maestría en Automatización y Control que se ofrece en el CITIS de la UAEH.

donde utilizan redes neuronales artificiales *wavenet* para identificar la planta y calcular esas ganancias, esta alternativa es utilizada en el presente trabajo usando la *wavelet* Morlet, con el fin de controlar la velocidad de un motor de inducción de AC trifásico. Dichos motores son mejor conocidos como los caballos de batalla de la industria [3], son muy utilizados debido a sus ventajas como la fiabilidad, eficiencia y bajo costo en comparación a motores utilizados en aplicaciones similares. Existen algunas técnicas para manipular la velocidad de estas máquinas eléctricas como el control de campo-orientado o también llamado control-vectorial, el control sin sensores y algunos basados en redes neuronales, entre otros [3, 8, 12]. La propuesta de control descrita en este trabajo pretende ser una técnica alternativa a estas, en la cual, solo se requiere tener las mediciones de la entrada y salida del motor de inducción para llevar a cabo la tarea de control, sin hacer uso de su modelo matemático ni de los parámetros del mismo.

II. CONTROLADOR PID WAVENET

El esquema del controlador PID *wavenet* se muestra en la Figura 2. Se pueden identificar en su arquitectura que dicho controlador consta de tres etapas para poder manipular la salida de un sistema SISO no lineal en una forma deseada.

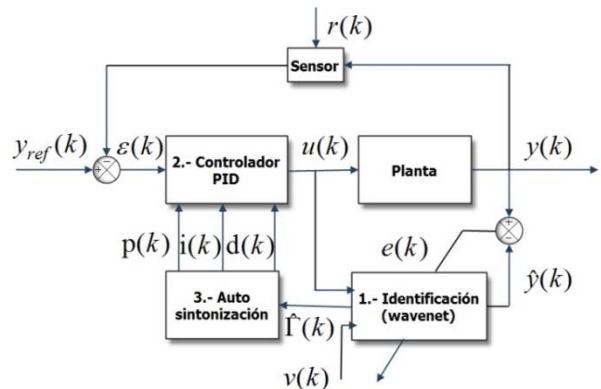


Figura 2: Esquema a bloques del controlador PID auto-sintonizado mediante una red neuronal *wavenet* para un sistema dinámico SISO

II.1. IDENTIFICACIÓN DEL SISTEMA

El proceso de identificación se hace mediante una red neuronal de base radial en la que las funciones de activación son funciones *wavelet* hijas $\psi_j(\tau)$ del tipo Morlet, además cuenta con un filtro IIR en cascada que tiene como función filtrar (podar) las neuronas que tienen “poca” contribución en el proceso de identificación, permitiendo con esto reducir el número de iteraciones en el proceso de aprendizaje [7]. Dichos elementos se observan en las Figuras 3 y 4 respectivamente. La función *wavelet* $\psi(k)$ es llamada *wavelet* madre porque las diferentes *wavelets* generadas a partir de ella, por su dilatación o contracción y translación, se les llama *wavelets* hijas $\psi_{a,b}(\tau)$, representadas matemáticamente como [6]:

$$\psi_{a,b}(\tau) = \frac{1}{\sqrt{a}}\psi\left(\frac{\tau - b}{a}\right) \quad (1)$$

con $a \neq 0$; $a, b \in \mathcal{R}$ y

$$\tau = \left(\frac{k - b}{a}\right) \quad (2)$$

donde a es la variable de escala, que permite hacer dilataciones y contracciones; y b es la variable de translación, que permite el desplazamiento en el instante k . La representación matemática de la *wavelet* Morlet es [6]

$$\psi_{a,b} = \cos(\omega_0\tau)e^{-0.5\tau^2} \quad (3)$$

donde su derivada parcial con respecto a b es

$$\frac{\partial \psi_{a,b}}{\partial b} = \frac{1}{a} [\omega_0 \sin(\omega_0\tau)e^{-0.5\tau^2} + \tau h(\tau)]$$

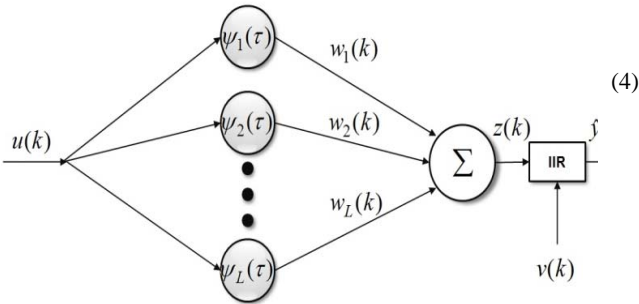


Figura 3: Esquema de una red neuronal *wavenet* con filtro IIR

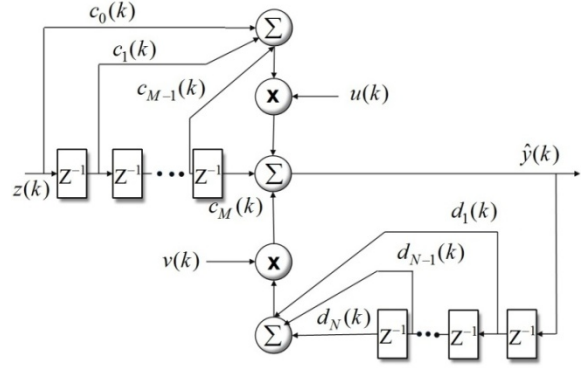


Figura 4: Estructura del filtro IIR

La señal de aproximación de la *wavenet* con filtro IIR $\hat{y}(k)$ puede ser calculada como:

$$\hat{y}(k) = \sum_{i=0}^M c_i z(k-i)u(k) + \sum_{j=1}^N d_j \hat{y}(k-j)v(k) \quad (5)$$

donde

$$z(k) = \sum_{l=1}^L w_l \psi_l(k) \quad (6)$$

L es el número de *wavelets*, w_l son los pesos de cada neurona en la *wavenet*, c_m y d_n son los coeficientes de adelanto y atraso del filtro IIR respectivamente, M y N son el número de los coeficientes de adelanto y atraso del filtro IIR, respectivamente. Los parámetros de la *wavenet*, en forma vectorial están dados por:

$$\mathbf{A}(k) \triangleq [a_1(k), a_2(k) \dots a_L(k)]^T \quad (7)$$

$$\mathbf{B}(k) \triangleq [b_1(k), b_2(k) \dots b_L(k)]^T \quad (8)$$

$$\mathbf{W}(k) \triangleq [w_1(k), w_2(k) \dots w_L(k)]^T \quad (9)$$

y los parámetros del filtro IIR, representados de la misma forma:

$$\mathbf{C}(k) \triangleq [c_1(k), c_2(k) \dots c_M(k)]^T \quad (10)$$

$$\mathbf{D}(k) \triangleq [d_1(k), d_2(k) \dots d_N(k)]^T \quad (11)$$

son optimizados mediante un algoritmo de aprendizaje basado en mínimos cuadrados medios LMS, mediante la minimización de una función de costos \mathbf{E} . Para lograr esto se define el error de estimación $e(k)$ entre la salida de la red neuronal $\hat{y}(k)$ y la salida real de la planta $y(k)$, como

$$e(k) = y(k) - \hat{y}(k) \quad (12)$$

La función de energía se define como

$$\mathbf{E} = \frac{1}{2} \sum_{k=1}^T e^2(k) \quad (13)$$

Para minimizar \mathbf{E} se aplica el método del gradiente de pasos descendentes, que utiliza las siguientes derivadas

$$\frac{\partial \mathbf{E}}{\partial w_l(k)} = -e(k)\mathbf{C}(k)^T \Psi_l(\tau)u(k) \quad (14)$$

$$\frac{\partial \mathbf{E}}{\partial b_l(k)} = -e(k)\mathbf{C}(k)^T \Psi_{bl}(\tau)w_l(k)u(k) \quad (15)$$

$$\frac{\partial \mathbf{E}}{\partial a_l(k)} = \tau_l \frac{\partial \mathbf{E}}{\partial b_l(k)} \quad (16)$$

$$\frac{\partial \mathbf{E}}{\partial c_m(k)} = -e(k)z(k-m)u(k) \quad (17)$$

$$\frac{\partial \mathbf{E}}{\partial d_n(k)} = -e(k)\hat{y}(k-m)v(k) \quad (18)$$

donde

$$\Psi_l(\tau) = [\psi_l(\tau), \psi_l(\tau-1), \dots, \psi_l(\tau-M)]^T \quad (19)$$

$$\Psi_{bl}(\tau) = \left[\frac{\partial \psi_l(\tau)}{\partial b_l(k)}, \frac{\partial \psi_l(\tau-1)}{\partial b_l(k)}, \dots, \frac{\partial \psi_l(\tau-M)}{\partial b_l(k)} \right]^T \quad (20)$$

La actualización de los parámetros cumple con la siguiente regla [4, 5, 15]

$$\Delta \boldsymbol{\theta}(k) = -\frac{\partial \mathbf{E}}{\partial \boldsymbol{\theta}(k)} \quad (21)$$

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \mu_\theta \Delta \boldsymbol{\theta}(k) \quad (22)$$

donde $\boldsymbol{\theta}$ puede ser cualquiera de los parámetros ajustados $\mathbf{W}(k), \mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)$ o $\mathbf{D}(k)$. El valor de $\mu_\theta \in \mathcal{R}$ representa el coeficiente de velocidad de aprendizaje para cada uno de los parámetros.

II.II. CONTROLADOR PID DISCRETO

Considerando un sistema dinámico SISO no lineal, puedes ser representado por la ecuación de estado discreto siguiente [10, 11]

$$x(k+1) = f[x(k), u(k), k] \quad (23)$$

$$y(k) = g[x(k), k] \quad (24)$$

donde $x(k) \in \mathcal{R}^n$, $u(k), y(k) \in \mathcal{R}$ y $f, g \in \mathcal{C}$ son desconocidas. La entrada $u(k)$ y la salida del sistema $y(k)$ son los únicos datos accesibles. Si el sistema linealizado alrededor del punto de equilibrio es observable [10, 11], existe una representación entrada-salida dada por

$$y(k+1) = \boldsymbol{\beta}[\mathbf{Y}(k), \mathbf{U}(k)] \quad (25)$$

donde

$$\mathbf{Y}(k) = [y(k) \ y(k-1) \ \dots \ y(k-n+1)] \quad (26)$$

$$\mathbf{U}(k) = [u(k) \ u(k-1) \ \dots \ u(k-n+1)] \quad (27)$$

es decir, sí existe una función $\boldsymbol{\beta}$ que mapea la salida $y(k)$ y la entrada $u(k)$ y sus $n-1$ valores pasados, en $y(k+1)$. Un modelo alternativo de una planta desconocida que

puede simplificar el algoritmo de la señal de control es el siguiente

$$y(k+1) = \Phi[\mathbf{Y}(k), \mathbf{U}(k)] + \Gamma[\mathbf{Y}(k), \mathbf{U}(k)]u(k) \quad (28)$$

sí los términos Φ y Γ son exactamente conocidos, el control $u(k)$ que obtiene una salida deseada $y_{ref}(k+1)$ es

$$u(k) = \frac{y_{ref}(k+1) - \Phi[\mathbf{Y}(k), \mathbf{U}(k)]}{\Gamma[\mathbf{Y}(k), \mathbf{U}(k)]} \quad (29)$$

Sin embargo los términos Φ y Γ son desconocidos. Por lo tanto se utiliza una red neuronal *wavenet* para aproximar las dinámicas del sistema como

$$\hat{y}(k+1) = \hat{\Phi}[y(k), \vartheta_\phi] + \hat{\Gamma}[y(k), \vartheta_\Gamma]u(k) \quad (30)$$

comparado la expresión anterior (30) con la salida de la *wavenet* con filtro IIR (5), se concluye

$$\hat{\Phi}[y(k), \vartheta_\phi] = \sum_{i=0}^N d_j \hat{y}(k-i)v(k) \quad (31)$$

$$\hat{\Gamma}[y(k), \vartheta_\Gamma] = \sum_{i=0}^M c_i z(k-i) \quad (32)$$

$$z(k) = \sum_{l=1}^L w_l \psi_l(k) \quad (33)$$

Por lo tanto, si las dos no linealidades Φ y Γ son estimadas por las dos funciones de la *wavenet* $\hat{\Phi}$ y $\hat{\Gamma}$ con parámetros ajustables ϑ_ϕ y ϑ_Γ respectivamente, la señal de control del PID que sigue la referencia deseada $y_{ref}(k)$ puede ser calculada como [1, 9, 13]:

$$u(k+1) = u(k) + p(k)\{\varepsilon(k) - \varepsilon(k-1)\} + i(k)\varepsilon(k) + d(k)\{\varepsilon(k) - 2\varepsilon(k-1) + \varepsilon(k-2)\} \quad (34)$$

donde $p(k), i(k)$ y $d(k)$ son las ganancias proporcional, integral y derivativas del controlador PID, $u(k)$ es la entrada de la planta al instante k y

$$\varepsilon(k) = y(k) - y_{ref}(k) \quad (35)$$

II.III. AUTO-SINTONIZACIÓN

Debido a que las ganancias $p(k), i(k)$ y $d(k)$ fueron consideradas dentro de la función de costo \mathbf{E} (13) pueden ser actualizadas de la siguiente manera

$$p(k) = p(k-1) + \mu_p e(k) \hat{\Gamma}(k) \{\varepsilon(k) - \varepsilon(k-1)\} \quad (36)$$

$$i(k) = i(k-1) + \mu_i e(k) \hat{\Gamma}(k) \varepsilon(k) \quad (37)$$

$$d(k) = d(k-1) + \mu_d e(k) \hat{\Gamma}(k) \{\varepsilon(k) - 2\varepsilon(k-1) + \varepsilon(k-2)\} \quad (38)$$

donde $\hat{\Gamma}(k)$ es parte de la identificación del sistema descrita en la ecuación (32). Y las constantes μ son las tasas de aprendizaje de las ganancias del controlador PID.

III. RESULTADOS SIMULADOS

El controlador PID *wavenet* fue simulado para controlar la velocidad de un motor de CA trifásico jaula de ardilla, utilizando un bloque *The asynchronous Machine* de *simulink* (MATLAB). El cual tiene los siguientes valores por unidad: $Lm = 2.0$, $Lls = 0.15$, $Llr = 0.15$, $Rs = 0.03$, $Rr = 0.03$, $2H = 0.3$ y $B = 0$. Los parámetros para la red neuronal *wavenet* y de la simulación se ven en la Tabla 1. Los valores iniciales y tasas de aprendizaje de los parámetros ajustados están en la Tabla 2.

Neuronas	3
Wavelet madre	Morlet
Coefficientes C del filtro IIR	3
Coefficientes D del filtro IIR	2
Épocas	20
Periodo de muestreo T	0.035 s
Señal de persistencia v	0.1

Tabla 1.- Parámetros de la *wavenet*, el filtro IIR y de la simulación.

W	[3.78, -3.36, -1.99]	μ_w	0.1
A	[-302.6, -55.5, -20]	μ_a	0.1
B	[92.7, 29.4, 107]	μ_b	0.1
C	[-0.4, -0.016, 0.64]	μ_c	0.1
D	[0.34, 1.66]	μ_d	0.1
p	0.02	μ_p	0.01
i	0.02	μ_i	0.007
d	0.003	μ_d	0.009

Tabla 2.- Valores iniciales y ganancias de aprendizaje de los parámetros ajustados en el controlador PID *wavenet* en la simulación.

III.I. RESULTADOS DE LA IDENTIFICACIÓN DEL SISTEMA

Se puede ver en la Figura 5 los resultados simulados de la identificación del motor de CA, en línea. El error de estimación $e(k)$ se muestra en la Figura 6.

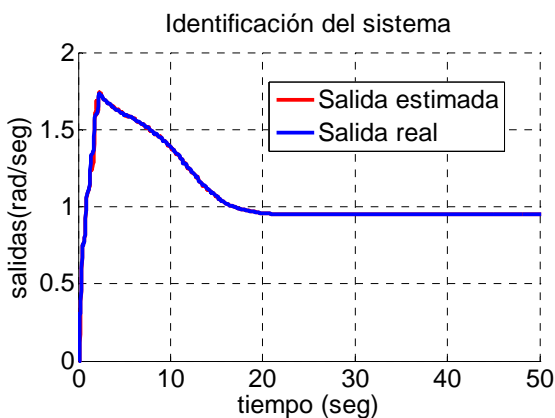


Figura 5.- Identificación del motor de CA, en línea

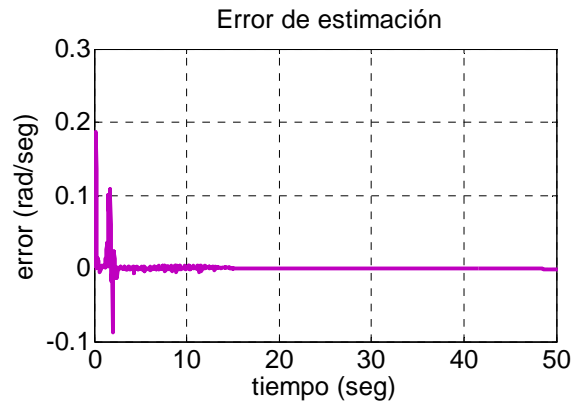


Figura 6.- Error de estimación $e(k)$ del motor de CA

A continuación se muestran las trayectorias que siguieron los parámetros que fueron ajustados en esta etapa. En la Figura 7 los parámetros de la *wavenet* (7), (8) y (9). Y en la Figura 8 los coeficientes de adelanto (10) y atraso del (11) del filtro IIR

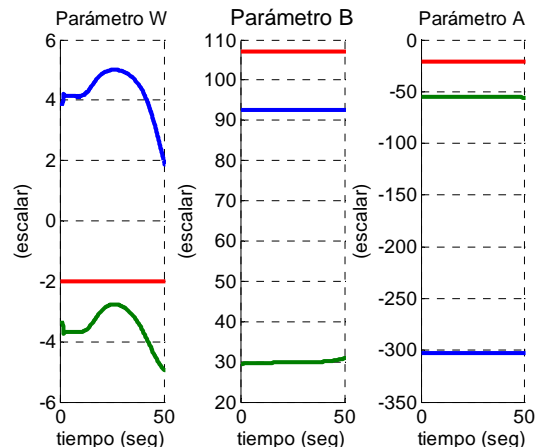


Figura 7.- Comportamiento de los parámetros de la *wavenet* $\{A(k), B(k) \text{ y } C(k)\}$

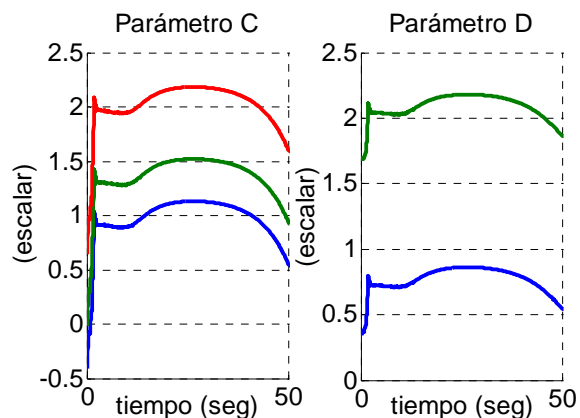


Figura 8.- Comportamiento de los parámetros del filtro IIR $\{C(k) \text{ y } D(k)\}$

III.II. RESULTADOS DEL CONTROLADOR PID WAVENET

En la Figura 9, se presenta la trayectoria de seguimiento de la salida del sistema $y(k)$ para alcanzar la referencia deseada $y_{ref}(k)$.

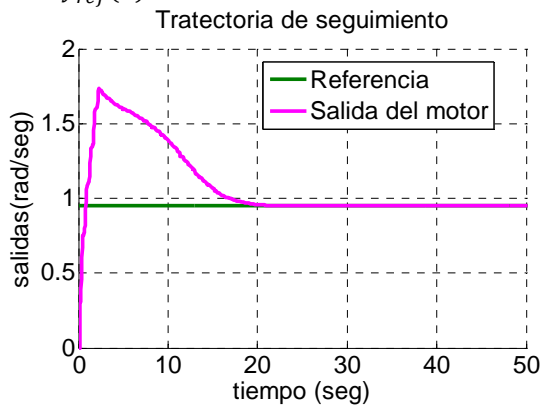


Figura 9.- Trayectoria de seguimiento

En la Figura 10 se observa a la izquierda la señal de control $u(k)$, el comportamiento del error de seguimiento $\varepsilon(k)$ a la derecha.

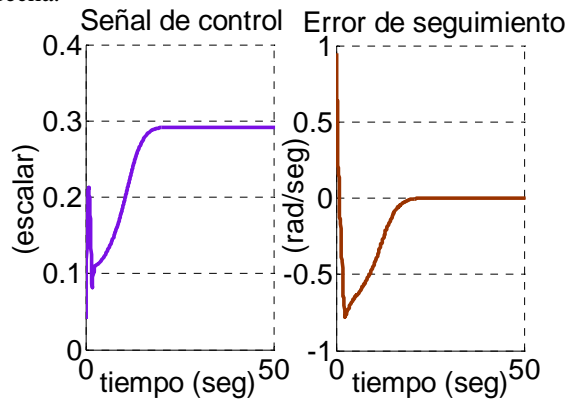


Figura 10.- Señal de control y error de seguimiento

III.III RESULTADOS DE LA AUTO-SINTONIZACIÓN

La Figura 11 corresponde al comportamiento que presentaron las ganancias del PID, auto-sintonizadas en línea.

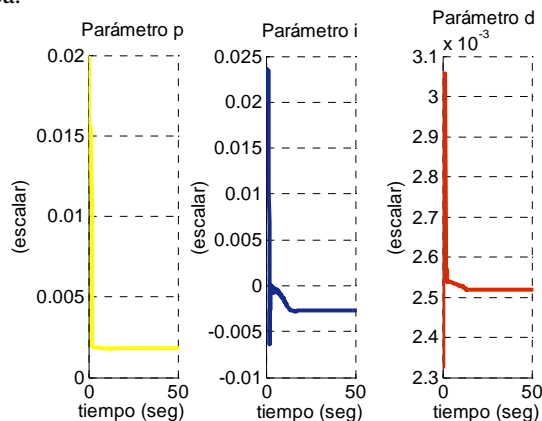


Figura 11.- Comportamiento de las ganancias del PID

IV. RESULTADOS EXPERIMENTALES

El PID *wavenet* se aplica experimentalmente para controlar la velocidad de un motor de inducción de AC trifásico del tipo jaula de ardilla (X), el cual está montado a una caminadora. Cuenta con un motor de CD (Y) como medio de sensado y un variador de frecuencia regadrive FVR G5 de la marca Regatron (Z). Estos elementos se ven en la Figura 12.



Figura 12.- Plataforma experimental.

El algoritmo se implementó en la plataforma de LabVIEW 9.0 y se utilizó una tarjeta de adquisición de datos NI-USB 6008 para manipular el variador de frecuencia (Z). Los parámetros para la red neuronal *wavenet* y del experimento se ven en la Tabla 3. Los valores iniciales y tasas de aprendizaje aparecen en la Tabla 4.

Neuronas	3
Wavelet madre	Morlet
Coefficientes C del filtro IIR	3
Coefficientes D del filtro IIR	2
Épocas	20
Periodo de muestreo T	0.035 s
Señal de persistencia ν	0.1

Tabla 3.- Parámetros de la *wavenet*, el filtro IIR y del experimento

W	[3.78, -3.36, -1.99]	μ_w	0.1
A	[-302.6, -55.5, -20]	μ_a	0.1
B	[92.7, 29.4, 107]	μ_b	0.1
C	[-0.4, -0.016, 0.64]	μ_c	0.1
D	[0.34, 1.66]	μ_d	0.1
p	0.02	μ_p	0.01
i	0.02	μ_i	0.007
d	0.003	μ_d	0.009

Tabla 4.- Valores iniciales y ganancias de aprendizaje de los parámetros ajustados del controlador PID *wavenet* experimentales.

IV.I. RESULTADOS DE LA IDENTIFICACIÓN DEL SISTEMA

Se muestran en las Figura 13 y 14 los resultados de la identificación del motor de CA, en línea y del comportamiento del error de estimación de la planta $e(k)$, respectivamente.

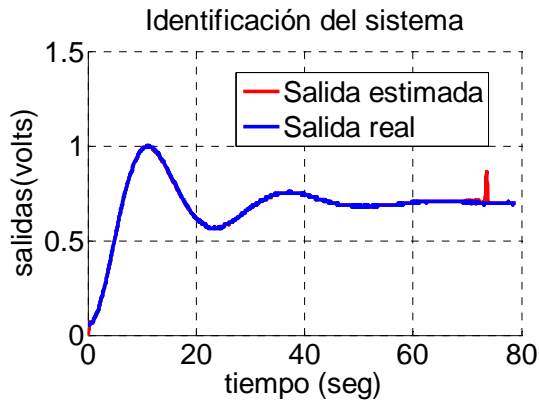


Figura 13.- Identificación del motor de CA, en línea

Además se presentan las trayectorias que siguieron los parámetros que fueron ajustados durante el experimento para lograr la identificación del motor de CA. Por un lado en la Figura 15 se observan los parámetros de la *wavenet* (7), (8) y (9). Y por el otro, el comportamiento de los coeficientes de adelanto (10) y atraso del (11) del filtro IIR, los cuales aparecen en la Figura 16.

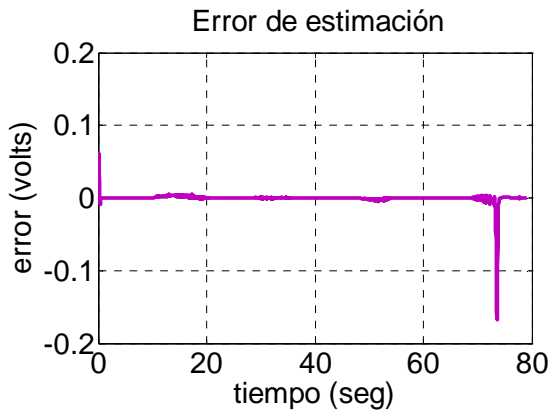


Figura 14.- Error de estimación $e(k)$ del motor de CA

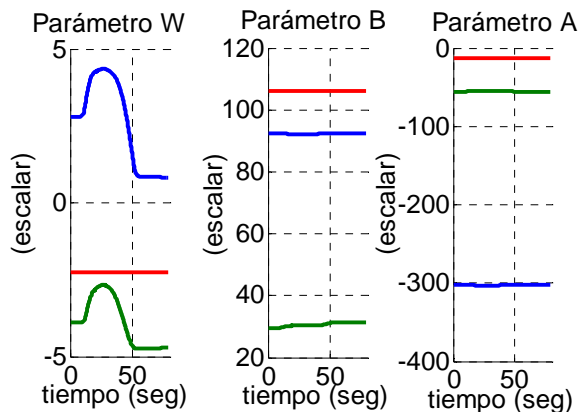


Figura 15.- Comportamiento de los parámetros de la *wavenet* $\{A(k), B(k) \text{ y } C(k)\}$

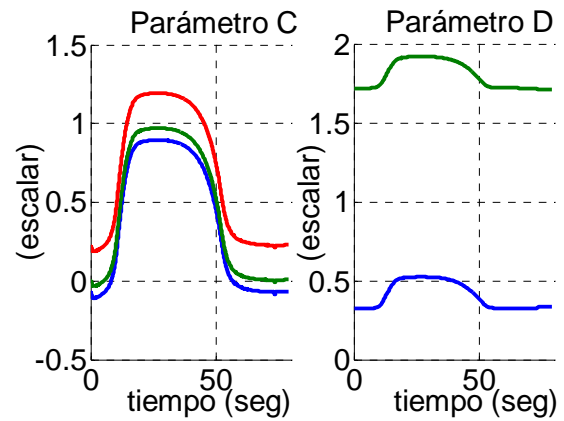


Figura 16.- Comportamiento de los parámetros del filtro IIR $\{C(k) \text{ y } D(k)\}$

IV.II. RESULTADOS DEL CONTROLADOR PID *WAVENET*

En la Figura 17 se presenta la trayectoria de seguimiento de la salida del sistema $y(k)$ para alcanzar la referencia deseada $y_{ref}(k)$. Se observa que se cumple el objetivo de seguimiento a los 60 segundos. En la Figura 18 se observan señal de control que experimentó la entrada de la planta $u(k)$ y el comportamiento del error de seguimiento $\varepsilon(k)$.

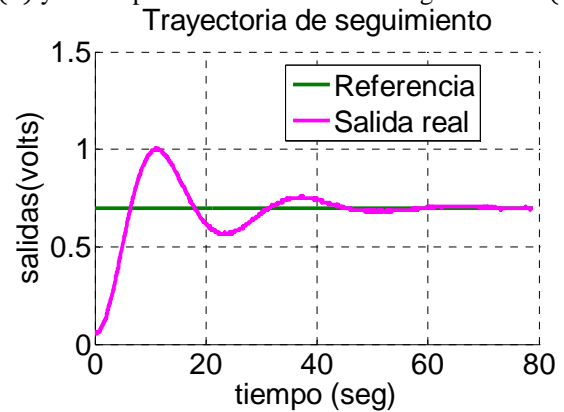


Figura 17.- Trayectoria de seguimiento.

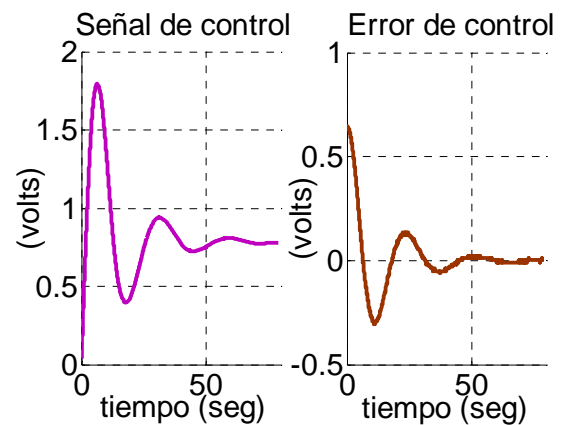


Figura 18.- Señal de control y error de seguimiento.

IV.III RESULTADOS DE LA AUTO-SINTONIZACIÓN

Por último en la Figura 19 se presenta la auto-sintonización de las ganancias del controlador PID, en línea, durante el experimento.

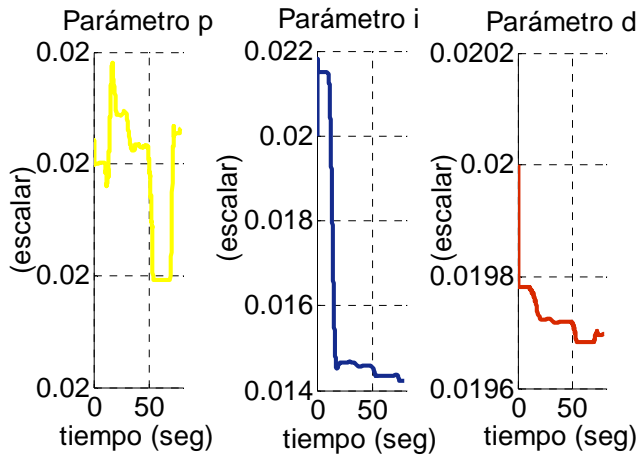


Figura 19.- Auto-sintonización de las ganancias del PID, en línea.

V. CONCLUSIONES

De los resultados presentados en este artículo se concluye que la relevancia está en la identificación y control de un sistema dinámico SISO desconocido, usando wavelets, redes neuronales y filtros IIR como herramientas para lograr el objetivo de control. Se emplea una red neuronal de base radial cuyas funciones de activación son *wavelets* hijas Morlet realiza la identificación del motor de inducción de CA trifásico del tipo jaula de ardilla, además que las ganancias del PID discreto son sintonizadas en función de la identificación, todo esto se hace en línea.

Actualmente se está desarrollando el diseño digital del PID *wavenet* presentado en este artículo, todo gracias al paralelismo de procesamiento que existe en las redes neuronales y el paralelismo de funcionamiento del diseño digital, el cual, se implementará en un FPGA de la familia Virtex-4 de Xilinx.

REFERENCIAS

- [1] Astrom K. (1997). *Computer-Controlled Systems, theory and design*. Prentice-Hall.
- [2] Astrom K. y T. Hagglund (2006). *Advanced PID Control*. ISA.
- [3] Böcker J. y Mathapati S. (2007). *State of the Art of Induction Motor Control*. in Electric Machines and Drives Conference, 2007. IEDMC'07. IEEE International. Paderborn, Germany. pp. 1459–1464.
- [4] Cruz J., L. Ramos y M. Espejel (2009). "PID Wavelet Auto-Sintonizado con una Red Neuronal Wavenet". V Semana Nacional de Ingeniería Electrónica SENIE09, Ocotlán, Jalisco, México, 7-9 Octubre. ISBN 978-607-477-073-5, pp 23-31.
- [5] Cruz J., L. Ramos y M. Espejel (2010). *A Self-Tuning Wavelet PID Controller Using Wavelet Networks*. 20th International Conference on Electronics Communications and Computers (CONIELE-COMP 2010). Puebla, México.
- [6] Daubechies I. (1992). *Ten Lectures on Wavelets*. SIAM.
- [7] Haykin S. (2001). *Kalman Filtering and Neural Networks*. Wiley.
- [8] Holtz J. (2002). *Sensorless Control of Induction Motor Drives*. in Proceedings of the IEEE. Wuppertal, Germany. Vol. 90, No. 8, pp. 1359-1394.
- [9] Kuo B. (1997). *Sistemas de Control Digital*. C.E.C.S.A.
- [10] Levin A. y K. Narendra (1993). *Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization*. IEEE Transactions on Neural Networks, Vol. 4, pp. 192-206.
- [11] Levin A. y K. Narendra (1996). *Control of Nonlinear Dynamical Systems Using Neural Networks - Part II: Observability, Identification, and Control*. IEEE Transactions on Neural Networks, Vol. 7, pp. 30-42.
- [12] Merabet A., Ouhrouche M. y Bui R. (2006). *Neural Generalized Predictive Controller for Induction Motor*. in International Journal of Theoretical and Applied Computer Sciences. Quebec, Canada. pp. 83-100.
- [13] Ogata K. (1996). *Sistemas de Control en Tiempo Discreto*. Prentice-Hall.
- [14] Parvez S. y Z. Gao (2005) *A Wavelet-Based Multiresolution PID Controller*. IEEE Transaction on Industry Applications, Vol 41, pp. 537-543.
- [15] Sedighzadeh M. y A Rezaadeh (2008). *Adaptive PID Control of Wind Energy Conversion Systems. Using RASPI Mother Wavelet Basis Function Network*. Proceeding of World Academy of Science, Engineering and Technology, pp. 269-273.