

## Estudio comparativo de controladores PID WaveNet-IIR aplicado a un helicóptero de 2 GDL

### Comparative study of WaveNet-IIR PID controllers applied to a 2 GDL helicopter

O. F. Garcia-Castro <sup>a,\*</sup>, L. E. Ramos-Velasco <sup>a</sup>, R. Garcia-Rodriguez <sup>a</sup>, M. A. Vega-Navarrete <sup>b</sup>, E. Escamilla-Hernández <sup>1</sup>, N. Oliva-Moreno <sup>d</sup>

<sup>a</sup>Maestría en Ingeniería Aeroespacial, Universidad Politécnica Metropolitana de Hidalgo, 43860, Tolcayuca, Hidalgo, México.

<sup>b</sup>Programa de Ingeniería en Aeronáutica, Universidad Politécnica Metropolitana de Hidalgo, 43860, Tolcayuca, Hidalgo, México.

<sup>c</sup>Escuela Superior de Ingeniería Mecánica Culhuacán, Instituto Politécnico Nacional, 04440, Ciudad de México, México.

<sup>d</sup>Unidad Profesional Interdisciplinaria de Ingeniería Campus Hidalgo, Instituto Politécnico Nacional, 42162, Pachuca, Hidalgo, México.

#### Resumen

En aplicaciones de control inteligente, uno de los problemas es determinar el número de capas y de neuronas en cada capa. Esto se vuelve más complejo con ciertos tipos de redes neuronales como las basadas en *Wavelets* donde las traslaciones y dilataciones son parámetros adicionales. Este artículo presenta un estudio comparativo para determinar el tipo de *wavelet* y número de neuronas que muestran el mejor desempeño para controlar un helicóptero Quanser de dos grados de libertad (GDL). Se presenta un controlador tipo *PID-WaveNet-IIR* el cuál se compone de controladores PID discretos en el tiempo, con ganancias auto-sintonizables por una red neuronal de base radial cuyas funciones de activación son *wavelets* y un par de filtros de respuesta de impulso infinito (IIR) para “podar” algunas neuronas. Por medio de simulaciones numéricas, usando LabVIEW, se presenta el desempeño del sistema en lazo cerrado para: diferentes condiciones de operación, tipos de familia *wavelets* donde se fijan valores mínimos de los errores de seguimiento, tipo de *wavelet*, el número de neuronas de la red y número de coeficientes de adelanto y atraso de los filtros IIR.

**Palabras Clave:** Controladores inteligentes, Identificación para control, Control adaptativo lineal, Control adaptativo por redes neuronales

#### Abstract

In intelligent control applications, one problem is determining the number of layers and neurons in each layer. The problem becomes even more complex when the neural network includes functions like WaveNets, where translations and dilations are additional parameters. This article presents a comparative study to determine the type of wavelet and number of neurons that best perform to approximate the Quanser helicopter dynamics with two degrees of freedom (DOF). The identification is based on a radial-based neural network whose activation functions are wavelets together, a pair of infinite impulse response (IIR) filters to “prune” some neurons. Additionally, a *PID-WaveNet-IIR* is presented, composed of a set of discrete PID controllers with self-tuning gains. Through numerical simulations using LabVIEW, the performance of the closed-loop system is presented under different operating conditions, types of family wavelets, where the minimum values of tracking errors are given previously, the number of neurons in the network, and the number of IIR filter lead and lag coefficients.

**Keywords:** Intelligent control, Identification for control, Linear adaptive control, Adaptive control by neural networks

#### 1. Introducción

Desde su aparición, la teoría de *wavelet* se ha vuelto una herramienta valiosa para el procesamiento de señales (Mallat,

2009); p. ej., es usada para la supresión de señales, la eliminación de ruidos, la compresión de señales, entre otras aplicaciones. Las *wavelets* se caracterizan por formar una familia a partir de una *wavelet* “madre” aplicando operaciones de dilata-

\*Autor para correspondencia: 213220003@upmh.edu.mx

**Correo electrónico:** 213220003@upmh.edu.mx (Oscar Federico Garcia-Castro), lramos@upmh.edu.mx (Luis Enrique Ramos-Velasco), rogarcia@upmh.edu.mx (Rodolfo Garcia-Rodriguez), mvega@upmh.edu.mx (Mario Alejandro Vega-Navarrete), eescamillah@ipn.mx (Enrique Escamilla-Hernández), loliva@ipn.mx (Luz Noé Oliva Moreno).

ción y traslación. Sin embargo, la elección de esta función no es única y dependerá en gran medida de la señal con la que se trabaja. Además, las funciones *wavelet* son una alternativa a la Transformada de Fourier, TF, para el análisis de señales con la diferencia de que la TF solo permite la localización en frecuencia de una señal pero no en el tiempo. Algunas de las *wavelets* más comunes son: Shannon, Har, Mexican hat, Spline, Morlet, Rasp y Polywog.

En este sentido, se busca aprovechar las propiedades de las *wavelets* junto a las redes neuronales de función de base radial (RBF, por sus siglas en inglés) (Sundararajan et al., 2002; Sanner y Slotine, 1992; Seshagiri y Khalil, 2000; Huang et al., 2002; Ge y Wang, 2002; Li et al., 2004; Huang et al., 2007; Wang y Yu, 2007; Zhu et al., 2008) en especial su capacidad de aproximadores de funciones. Lo que ha permitido que en los últimos años se haya extendido sus aplicaciones en identificación y control, (Abuhamdia y Taheri, 2017; Tangirala et al., 2013).

Por consiguiente, en este trabajo se presenta un controlador denominado *PID-WaveNet-IIR*, el cual consta de un método de auto-sintonización en línea (Barrón-Gómez et al., 2017) de las ganancias  $\mathbf{p}(k)$ ,  $\mathbf{i}(k)$  y  $\mathbf{d}(k)$  de un conjunto de controladores discretos del tipo proporcional-integral-derivativo (PID) (Ramos-Velasco et al., 2016). La adaptación de la red RBF se utilizará para mejorar el rendimiento del control aprovechando a las *wavelets* como funciones de activación y la salida de la red se conectará a un par de filtros de respuesta infinita al impulso (IIR) para podar algunas neuronas.

La utilidad de los controladores PID es que, para su diseño se puede prescindir del modelo matemático de la planta, permitiéndole ser una alternativa generalizada para la mayoría de los sistemas de control donde la sintonización de ganancias pueda llevarse a cabo de forma manual o automática (Åström, 1996, 1995). Aunque, uno de los problemas presentes con el esquema de control *PID-WaveNet-IIR* es que el diseñador debe sintonizar varios parámetros, los cuales regularmente se determinan a prueba y error, tales como: tipo de *wavelet* “madre”, número de neuronas, cantidad de coeficientes de adelanto y atraso de los filtros IIR, así como las tasas de aprendizaje.

Con el fin de resolver este problema, la contribución de este artículo es generar un método para determinar la *wavelet* “hija” que presente el mejor desempeño ante ciertas condiciones de operación. El estudio comparativo se llevará a cabo tomando como referencia un conjunto de *wavelets* “madres” agrupadas en cuatro familias. Para encontrar la mejor configuración, primero se hacen pruebas en lazo abierto con cada función *wavelet* “madre”; enseguida se calcula la media aritmética de los parámetros para cada familia; y al final, se varía el número de neuronas para observar el comportamiento del sistema en lazo cerrado. Y para determinar qué función *wavelet* “madre” es la mejor para el esquema de control y la planta propuestos, se registra el tiempo que tarda en reducir el error de seguimiento a una tolerancia dada por el diseñador. Dicho esquema de control *PID-WaveNet-IIR* se aplicará en el helicóptero *Quanser* de dos grados de libertad (GDL), presentando los resultados de simulación numérica en LabVIEW 2013 bajo diferentes condiciones de operación.

El artículo está organizado de la siguiente manera: en la Sección 2 se exponen los algoritmos de redes neuronales direc-

tamente sobre el helicóptero; mientras que, en la Sección 3, se muestran los resultados obtenidos de las simulaciones numéricas en lazo cerrado. Finalmente, en la Sección 4 se dan las conclusiones y trabajos futuros.

## 2. Esquema de control *PID-WaveNet-IIR* para el helicóptero *Quanser*

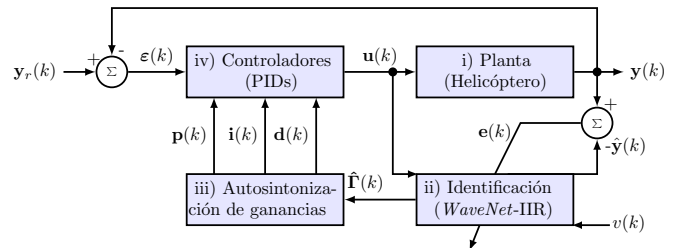


Figura 1: Diagrama a bloques del esquema *PID-WaveNet-IIR* para el helicóptero *Quanser* de 2 GDL.

En la Figura 1 se muestra el esquema de control *PID-WaveNet-IIR* propuesto, donde se aprecian cuatro bloques: *i*) la planta, la cual consiste de un helicóptero de 2 GDL, cuya dinámica es representada por ecuaciones diferenciales lineales; *ii*) *WaveNet-IIR*, encargado de realizar la identificación del modelo lineal; *iii*) Autosintonización, que ajusta las ganancias de los controladores PID cada periodo de muestreo  $T$ ; finalmente, se tienen a los *iv*) Controladores del tipo PID discreto para cada uno de los grados de libertad.

A continuación se realizará una descripción detallada de cada uno de los bloques.

### *i*) Planta (Helicóptero)

Consiste de dos ecuaciones diferenciales lineales utilizados para controlar el cabeceo  $\theta$  y guiñada  $\phi$  de un helicóptero de la empresa *Quanser* mostrado en la Figura 2. Algunos parámetros del helicóptero se indican en la Tabla 1 (Inc., 2012).

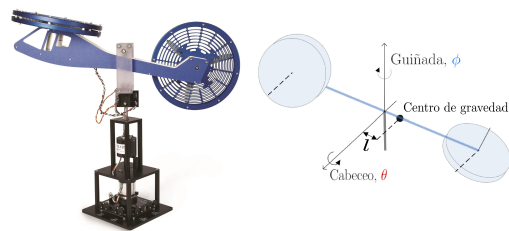


Figura 2: Modelo experimental del helicóptero *Quanser* de 2 GDL y su diagrama de cuerpo libre (Inc., 2012).

Tabla 1: Parámetros del modelo del helicóptero *Quanser* de 2 GDL

Símbolo	Descripción	Valor
$K_{pp}$ ,	Constante de torque debido	0.2040 N m/V
$K_{py}$ ,	a la fuerza de empuje sobre	0.0068 N m/V
$K_{yy}$ , $K_{yp}$	los ejes de cabeceo y guiñada	0.0720 N m/V
	desde las hélices de cabeceo	0.0219 N m/V
	y guiñada, resp.	
$V_{\theta}$ , $V_{\phi}$	Voltaje en el motor de cabeceo y guiñada, respectivamente.	$\pm 24$ V y $\pm 15$ V

Para llevar a cabo la simulación de la dinámica del helicóptero se utilizaron sus funciones de transferencia, las cuales son (Inc., 2012):

$$H_{\theta\theta}(s) = \frac{y_\theta(s)}{V_\theta(s)} = \frac{2.361}{s^2 + 9.26s}, \quad H_{\phi\theta}(s) = \frac{y_\phi(s)}{V_\theta(s)} = \frac{0.2402}{s^2 + 3.487} \quad (1)$$

$$H_{\theta\phi}(s) = \frac{y_\theta(s)}{V_\phi(s)} = \frac{0.07871}{s^2 + 9.26s}, \quad H_{\phi\phi}(s) = \frac{y_\phi(s)}{V_\phi(s)} = \frac{0.7895}{s^2 + 3.487s} \quad (2)$$

donde  $y_\theta$  y  $y_\phi$  son las salidas del modelo para cada eje de movimiento. Es importante mencionar que, el modelo lineal usado se obtuvo mediante una expansión en series de Taylor del modelo no lineal y después se le aplica la transformada de Laplace. Además, en la Figura 3 se aprecia el acoplamiento del modelo.

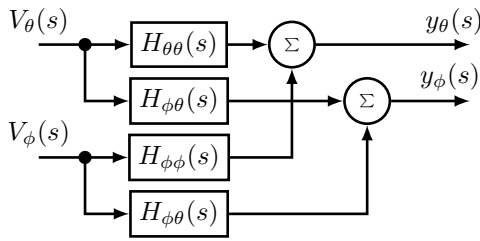


Figura 3: Diagrama de bloques basado en las funciones de transferencia.

Adicionalmente, el helicóptero posee un bloque que se realiza la interpretación eléctrica de la señal de control  $\mathbf{u} = [u_\theta, u_\phi]^T$ , es decir, la relación mecánica-eléctrica dada por

$$V_\theta = 0.4899u_\theta - 0.7387u_\phi, \quad (3)$$

$$V_\phi = 0.0793u_\phi - 0.7387u_\theta. \quad (4)$$

donde  $u_\theta$  y  $u_\phi$  son las señales de cada controlador PID, mientras que  $V_\theta$  y  $V_\phi$  son los voltajes para los motores de cada eje.

#### ii) Identificación (WaveNet-IIR)

En este bloque se identifica la dinámica entrada-salida del sistema y su estructura está compuesta por una red neuronal de base radial (RBF) (Haykin, 1999) cuyas funciones de activación son *wavelets* “hijas”, como se muestra en la Figura 4.

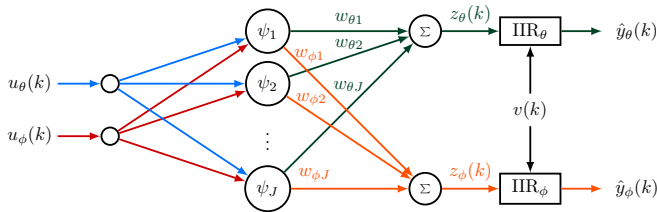


Figura 4: Diagrama de la red neuronal WaveNet con filtros IIR.

De igual manera, en la Figura 5 se observa la configuración interna de los bloques de filtro IIR cuyas salidas son las posiciones del helicóptero  $\hat{y}_\theta$  y  $\hat{y}_\phi$ . Se debe destacar que estos filtros “podan” a las neuronas que tienen poca contribución en el proceso de identificación y con él se tiene una estructura entrada-salida (Vega-Navarrete et al., 2018).

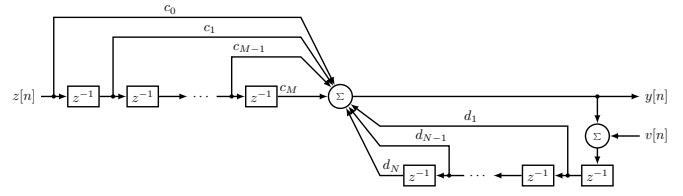


Figura 5: Estructura del filtro IIR donde  $M$  y  $N$  representan el número de coeficientes de adelanto y retraso, respectivamente.

Los parámetros de la red neuronal y de los filtros IIR en su forma matricial son: la señal de control  $\mathbf{u} = [u_\theta, u_\phi]^T$ , el parámetro de traslación  $\mathbf{b} = [b_1, b_2, \dots, b_J]^T$ , el parámetro de dilatación  $\mathbf{a} = [a_1, a_2, \dots, a_J]^T$ , las *wavelets* “hijas”  $\boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_J]^T$ , la directriz de la red neuronal  $\hat{\mathbf{F}} = [\hat{F}_\theta, \hat{F}_\phi]^T$ , la salida de la red neuronal  $\mathbf{z} = [z_\theta, z_\phi]^T$ , la posición estimada  $\hat{\mathbf{y}} = [\hat{y}_\theta, \hat{y}_\phi]^T$ , y las matrices de los pesos sinápticos  $\mathbf{W} \in \mathbb{R}^{2 \times J}$ ; y coeficientes  $\mathbf{C} \in \mathbb{R}^{2 \times M}$  y  $\mathbf{D} \in \mathbb{R}^{2 \times N}$ .

Para el estudio comparativo se emplean las *wavelets* “madre” de la Tabla 2 (Domínguez-Mayorga et al., 2012) y se estudia el desempeño de cada una en el esquema de control propuesto.

Tabla 2: Cuatro familias de *wavelets* madre usadas en este estudio

Familia	Expresión matemática, $\psi(\tau)$
<i>Morlet</i>	$\cos(w_0\tau)e^{-0.5\tau^2}$
<i>Rasp1</i>	$\tau(\tau^2 + 1)^{-2}$
<i>Rasp2</i>	$\tau \cos \tau(\tau^2 + 1)^{-1}$
<i>Rasp3</i>	$\sin \pi\tau(\tau^2 - 1)^{-1}$
<i>Shannon</i>	$(\sin 2\pi\tau - \sin \pi\tau)(\pi\tau)^{-1}$
<i>Polywog1</i>	$\tau e^{-0.5\tau^2}$
<i>Polywog2</i>	$(\tau^3 - 3\tau)e^{-0.5\tau^2}$
<i>Polywog3</i>	$(\tau^4 - 6\tau^2 - 3)\tau e^{-0.5\tau^2}$
<i>Polywog4</i>	$(1 - \tau^2)e^{-0.5\tau^2}$
<i>Polywog5</i>	$(3\tau^2 - \tau^4)e^{-0.5\tau^2}$

La función  $\psi(\tau)$  se llama *wavelet* “madre” porque a partir de ella se generan otras funciones al establecer valores que dilatan en un factor  $a_i$  y trasladen en un factor  $b_i$  a la señal *wavelet* “madre”, todo esto sobre el eje del tiempo. Estas funciones resultantes se nombran *wavelets* “hijas”  $\psi(\tau_i)$ , las cuales se representan de la siguiente manera (García-Castro, 2020; Carrillo-Santos et al., 2018):

$$\psi(\tau_i) = \frac{1}{\sqrt{|a_i|}} \psi(\tau_i), \quad \tau_i(k, a_i, b_i) = \left( \frac{k - b_i}{a_i} \right), \quad a_i \neq 0 \quad (5)$$

donde  $k$  se relaciona con el instante en el que se realiza la operación y  $\tau_i$  es una variable discreta en el tiempo. Por ejemplo, la  $i$ -ésima *wavelet* “hija” de la función *Morlet* está dada por

$$\psi_i(\tau_i) = \cos(w_0\tau_i)e^{-0.5\tau_i^2} \quad (6)$$

donde  $i \in \{1, 2, \dots, J\}$  siendo  $J$  el número de *wavelets* “hijas” o el número de neuronas en la red. Además, la derivada parcial de esta función con respecto a la  $i$ -ésima variable  $\tau_i$  es

$$\frac{\partial \psi_i(\tau_i)}{\partial \tau_i} = \frac{1}{a_i} \left[ w_0 \sin(w_0\tau_i)e^{-0.5\tau_i^2} + \tau_i \psi_i(\tau_i) \right]. \quad (7)$$

La salida  $\mathbf{z}(k)$  de la *WaveNet* está dada por

$$\mathbf{z}(k) = \mathbf{u}^T(k) \mathbf{W}(k) \boldsymbol{\psi}^T(k), \quad (8)$$

dicha salida  $\mathbf{z}(k)$  pasa a través de los filtros IIR para obtener la posición estimada  $\hat{\mathbf{y}}(k)$ , la cual es

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{\Gamma}}(k) + \text{diag}(\mathbf{D}(k)) \hat{\mathbf{Y}}(k) \mathbf{v}(k) \quad (9)$$

donde

$$\hat{\mathbf{\Gamma}}(k) = \text{diag}(\mathbf{C}(k) \mathbf{Z}(k)), \quad (10)$$

$$\hat{\mathbf{Y}}(k) = \begin{bmatrix} \hat{y}_\theta(k-1) & \hat{y}_\theta(k-2) & \cdots & \hat{y}_\theta(k-N) \\ \hat{y}_\phi(k-1) & \hat{y}_\phi(k-2) & \cdots & \hat{y}_\phi(k-N) \end{bmatrix}, \quad (11)$$

$$\mathbf{Z}(k) = \begin{bmatrix} z_\theta(k) & z_\theta(k-1) & \cdots & z_\theta(k-M) \\ z_\phi(k) & z_\phi(k-1) & \cdots & z_\phi(k-M) \end{bmatrix}, \quad (12)$$

y  $\mathbf{v}(k) = [v(k), v(k)]^T$  es la señal persistente de los filtros.

El ajuste de los parámetros de la *WaveNet-IIR* se realiza con un proceso de optimización estática basada en el algoritmo de retropropagación, donde la función de costo  $\mathbf{E}(k)$  se define como:

$$\mathbf{E}(k) = \frac{1}{2} \sum_t [\mathbf{e}_t(k)]^2 \quad (13)$$

para la cual se requiere el error de estimación  $\mathbf{e}_t(k) = \mathbf{y}_t(k) - \hat{\mathbf{y}}_t(k)$ , donde  $\mathbf{y}_t$  indica la salida del sistema y  $\hat{\mathbf{y}}_t$  es la salida estimada de la red neuronal con  $t = \{\theta, \phi\}$ . La minimización la función de costo  $\mathbf{E}(k)$  se realiza aplicando el método del gradiente descendente a través de las siguientes derivadas:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{W}(k)} = \mathbf{u}(k) \mathbf{e}^T(k) \mathbf{C}(k) \boldsymbol{\psi}_a^T(k) = U \mathbf{I}_e (\mathbf{C} \boldsymbol{\psi}_a)^T \quad (14)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{b}(k)} = -\mathbf{e}(k) \mathbf{u}(k) \mathbf{C}^T(k) \boldsymbol{\psi}_b = U (\mathbf{I}_f (\mathbf{C} \boldsymbol{\psi}_b)^T) \quad (15)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{a}(k)} = \tau \frac{\partial \mathbf{E}}{\partial \mathbf{q}(k)} = \frac{\partial \mathbf{E}}{\partial \mathbf{q}(k)} \odot \tau \quad (16)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{C}(k)} = -\mathbf{u}(k) \mathbf{e}^T(k) \mathbf{z}(k-M) = U \mathbf{I}_e \mathbf{z}^T \quad (17)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{D}(k)} = -\mathbf{v}(k) \mathbf{e}^T(k) \hat{\mathbf{y}}(k-N) = v \mathbf{I}_e (\hat{\mathbf{y}} \boldsymbol{\psi}_a)^T \quad (18)$$

donde

$$U = u_\theta(k) + u_\phi(k), \quad \mathbf{I}_e = \begin{bmatrix} e_\theta(k) & 0 \\ 0 & e_\phi(k) \end{bmatrix}, \quad (19)$$

$$\mathbf{I}_f = [e_\theta(k) \ e_\phi(k)]^T, \quad \tau = [\tau_1 \ \tau_2 \ \cdots \ \tau_J], \quad (20)$$

$$\boldsymbol{\psi}_a = \begin{bmatrix} \psi_1(\tau_1) & \psi_2(\tau_2) & \cdots & \psi_J(\tau_J) \\ \psi_1(\tau_1-1) & \psi_2(\tau_2-1) & \cdots & \psi_J(\tau_J-1) \\ \vdots & \cdots & \ddots & \vdots \\ \psi_1(\tau_1-M) & \psi_2(\tau_2-M) & \cdots & \psi_J(\tau_J-M) \end{bmatrix}, \quad (21)$$

$$\boldsymbol{\psi}_b = \begin{bmatrix} \frac{\partial \psi_1(\tau_1)}{\partial \tau_1} & \cdots & \frac{\partial \psi_J(\tau_J)}{\partial \tau_J} \\ \frac{\partial \psi_1(\tau_1-1)}{\partial \tau_1} & \cdots & \frac{\partial \psi_J(\tau_J-1)}{\partial \tau_J} \\ \vdots & \ddots & \vdots \\ \frac{\partial \psi_1(\tau_1-M)}{\partial \tau_1} & \cdots & \frac{\partial \psi_J(\tau_J-M)}{\partial \tau_J} \end{bmatrix}. \quad (22)$$

De este modo, la actualización de los parámetros cumple con la siguiente regla:

$$\lambda(k+1) = \lambda(k) + \mu_\lambda \Delta \lambda(k), \quad \Delta \lambda(k) = -\frac{\partial \mathbf{E}}{\partial \lambda(k)} \quad (23)$$

donde  $\lambda$  puede ser alguno de los parámetros  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  y  $\mathbf{W}$ , con su correspondiente tasa de aprendizaje  $\mu_\lambda$ .

Para fines de programación, en la plataforma LabVIEW 2013 se utilizó la tercera representación de cada una de las ecuaciones (14)–(18) donde  $\odot$  representa el operador Hadamard<sup>1</sup>.

iii) *Auto sintonización de las ganancias proporcional, integral y derivativa*

Este bloque se encarga de ajustar las ganancias  $\mathbf{p}(k) = [p_\theta, p_\phi]^T$ ,  $\mathbf{i}(k) = [i_\theta, i_\phi]^T$  y  $\mathbf{d}(k) = [d_\theta, d_\phi]^T$  del controlador de la siguiente manera (Ramos-Velasco et al., 2016):

$$\mathbf{p}(k) = \mathbf{p}(k-1) + \mu_p \mathbf{e}(k) \hat{\mathbf{\Gamma}}(k) [\boldsymbol{\varepsilon}(k) - \boldsymbol{\varepsilon}(k-1)] \quad (24)$$

$$\mathbf{i}(k) = \mathbf{i}(k-1) + \mu_i \mathbf{e}(k) \hat{\mathbf{\Gamma}}(k) \boldsymbol{\varepsilon}(k) \quad (25)$$

$$\mathbf{d}(k) = \mathbf{d}(k-1) + \mu_d \mathbf{e}(k) \hat{\mathbf{\Gamma}}(k) [\boldsymbol{\varepsilon}(k) - 2\boldsymbol{\varepsilon}(k-1) + \boldsymbol{\varepsilon}(k-2)] \quad (26)$$

donde  $\mu_p$ ,  $\mu_i$  y  $\mu_d$  son las tasas de actualización y  $\hat{\mathbf{\Gamma}}(k)$  es la directriz de la red neuronal dada por (10).

iv) *Controladores (PIDs)*

En este bloque se genera la señal de control  $\mathbf{u}(k) = [u_\theta, u_\phi]^T$  la cual hace que varíe la salida del sistema  $\mathbf{y}(k)$  con el objetivo de aproximar a la señal de referencia  $\mathbf{y}_r(k)$ . Dicha etapa solo necesita los valores de las ganancias  $\mathbf{p}(k)$ ,  $\mathbf{i}(k)$  y  $\mathbf{d}(k)$ ; así como el error de seguimiento  $\boldsymbol{\varepsilon}(k) = \mathbf{y}_r(k) - \mathbf{y}(k)$ . Entonces, la señal de control  $\mathbf{u}$  en el instante  $k+1$  está dada por (Ogata, 1996):

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \mathbf{p}(k) [\boldsymbol{\varepsilon}(k) - \boldsymbol{\varepsilon}(k-1)] + \mathbf{i}(k) [\boldsymbol{\varepsilon}(k)] + \mathbf{d}(k) [\boldsymbol{\varepsilon}(k) - 2\boldsymbol{\varepsilon}(k-1) + \boldsymbol{\varepsilon}(k-2)] \quad (27)$$

Cabe mencionar que, los bloques de identificación, autosintonización y controladores realizan sus cálculos correspondientes cada periodo de muestreo  $T$ .

### 3. Resultados en simulación numérica

La programación del controlador *PID-WaveNet-IIR* se realiza en la plataforma de LabVIEW 2013 y se aplica al modelo lineal del helicóptero *Quanser* para el control de las posiciones en los ejes de cabeceo  $\theta$  y guiñada  $\phi$ .

Primero se realiza el entrenamiento de la *WaveNet-IIR* para establecer sus condiciones iniciales de operación y, enseguida, se aplica el esquema de control en lazo cerrado para elegir la mejor combinación  $\psi - J$ , es decir, qué *wavelet* utilizar y la cantidad de neuronas  $J$ .

<sup>1</sup> Sean los vectores  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^{n \times 1}$  donde  $c_x = a_x \cdot b_x$  con  $x \in \{1, 2, \dots, n\}$  (Deisenroth et al., 2020).

### 3.1. Simulación en lazo abierto

Para encontrar las condiciones iniciales de los parámetros del esquema *WaveNet-IIR*, primero se hace un pre-entrenamiento de la red neuronal en lazo abierto solo para tareas de identificación, y se busca obtener las condiciones iniciales de los siguientes parámetros para cada función *wavelet*: cantidad de coeficientes de adelanto  $M$  y de retardo  $N$  de los filtros IIR, y las tasas de aprendizaje  $\mu_\lambda$ .

Entonces, primero se fija un error de tolerancia de estimación  $e_{tol} = 0.08$  grados, por ser el nivel de resolución del decodificador óptico (Inc., 2012). Y se realizan pruebas con cada *wavelet* “madre” de la Tabla 2 proponiendo valores iniciales para  $M, N, v$ ; además, se inicializan las tasas de aprendizaje igual a 5,  $\mu_\lambda = 5$ . Cabe mencionar que, dichas asignaciones se harán para cada combinación  $\psi - J$ , es decir, para cada *wavelet*  $\psi$  y cantidad de neuronas  $J$  donde  $J = 1, 2, \dots, J_{max}$ , siendo  $J_{max} = 10$ .

Después, se harán simulaciones con cada combinación  $\psi - J$  hasta cumplir la condición  $\|e(k)\| \leq e_{tol}$ : en cada prueba se deben variar los valores de  $M, N, \mu_\lambda$  y  $v$ . Una vez que se han configurado todas las combinaciones  $\psi - J$ , por familia se calculan las medias aritméticas de cada parámetros, cuyos resultados se muestran las Tablas 3 y 4.

Tabla 3: Tasas de actualización obtenidas y señal persistencia propuesta de la WaveNet-IIR para cada familia wavelet

Parámetro	Familia			
	Rasp	Polywog	Shannon	Morlet
$\mu_w$	$3.80 \times 10^{-8}$	$6.89 \times 10^{-17}$	$5.00 \times 10^{-9}$	$7.02 \times 10^{-5}$
$\mu_b$	$3.80 \times 10^{-8}$	$6.92 \times 10^{-17}$	$5.00 \times 10^{-9}$	$6.89 \times 10^{-5}$
$\mu_a$	$3.80 \times 10^{-8}$	$7.02 \times 10^{-18}$	$5.00 \times 10^{-9}$	$6.89 \times 10^{-5}$
$\mu_c$	$5.15 \times 10^{-8}$	$1.26 \times 10^{-13}$	$2.18 \times 10^{-8}$	$1.26 \times 10^{-10}$
$\mu_d$	$5.90 \times 10^{-5}$	$3.22 \times 10^{-11}$	$2.93 \times 10^{-6}$	$3.22 \times 10^{-6}$
$v$	$3.36 \times 10^{-2}$	$1.25 \times 10^{-1}$	$2.00 \times 10^{-1}$	$5.00 \times 10^{-1}$

Tabla 4: Cantidad de coeficientes obtenidos para los filtros IIR en cada familia wavelet

Parámetro	Familia			
	Rasp	Polywog	Shannon	Morlet
Retrasos, $M$	5	4	4	4
Adelantos, $N$	4	3	2	2

### 3.2. Simulación en lazo cerrado

Partiendo de los resultados anteriores, ahora el diseñador propone las condiciones iniciales de las ganancias  $\mathbf{p}(k)$ ,  $\mathbf{i}(k)$  y  $\mathbf{d}(k)$  con sus respectivas tasas de actualización, dichos valores se indican en la Tabla 5.

Tabla 5: Valores iniciales propuestos de las ganancias de los controles PID y sus tasas de actualización

Ganancias	Tasas de aprendizaje
$\mathbf{p}(0) = [2.90 \times 10^{-2}, 9.04 \times 10^{-3}]$	$\mu_p = [1.00, 0.90]$
$\mathbf{i}(0) = [1.58 \times 10^{-2}, 1.10 \times 10^{-2}]$	$\mu_i = [0.80, 0.60]$
$\mathbf{d}(0) = [1.45 \times 10^{-2}, 5.50 \times 10^{-3}]$	$\mu_d = [0.75, 0.80]$

También, a criterio del diseñador, se definen las señales de

referencia para el eje de cabeceo  $\theta$  y guiñada  $\phi$ , y estas son

$$y_{r_\theta} = \begin{cases} -30, & 0 < k \leq 2.5 \\ 10, & 2.5 < k \leq 5 \\ 25, & 5 < k \leq 7.5 \\ -20, & 7.5 < k \leq 10 \end{cases}, \quad y_{r_\phi} = \begin{cases} 0, & 0 < k \leq 2.5 \\ 80, & 2.5 < k \leq 5 \\ -120, & 5 < k \leq 7.5 \\ 110, & 7.5 < k \leq 10 \end{cases}, \quad (28)$$

cuyas gráficas se muestran en la Figura 6.

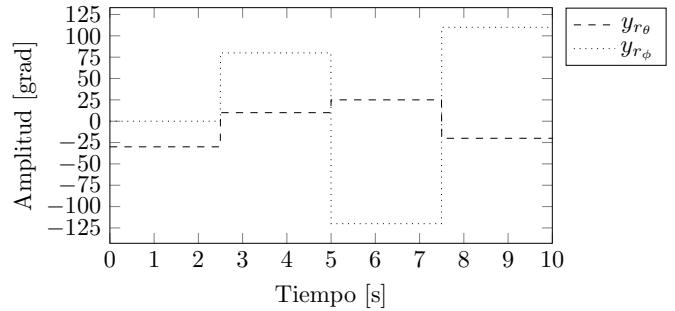


Figura 6: Señales de referencia para cada grado de libertad.

Para definir los ángulos de movimiento se consideraron las limitaciones físicas del modelo para el cabeceo, pues su rango de operación es  $-40 \leq \theta \leq 30$  grados; para la guiñada no existe dicha limitación, ya que gira libremente.

Una vez que ya se tienen definidos todos los parámetros del *PID-WaveNet-IIR*, así como las señales de referencia, se realizan simulaciones en lazo cerrado fijando un error de seguimiento  $\varepsilon_{tol} = 1 \times 10^{-5}$  grados. Al mismo tiempo, se toman como condiciones iniciales los valores de las Tablas 3, 4 y 5, y también el periodo de muestreo  $T = 5$  ms.

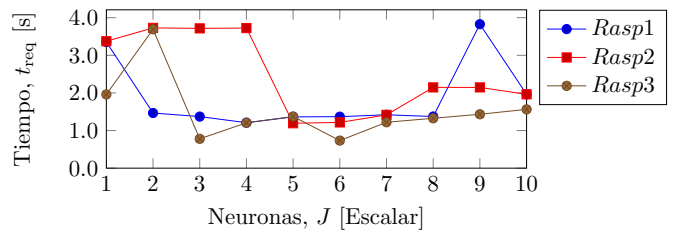


Figura 7: Familia Rasp. Tiempo requerido por las *PID-WaveNet-IIR* para aplicar el control en lazo cerrado con  $J_{max} = 10$ .

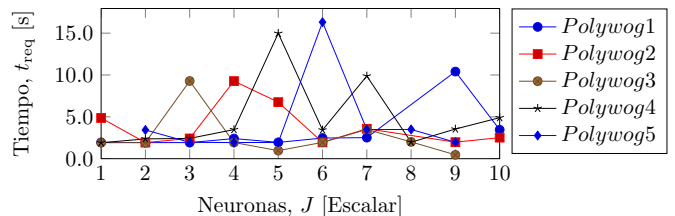


Figura 8: Familia Polywog. Tiempo requerido por las *PID-WaveNet-IIR* para aplicar el control en lazo cerrado con  $J_{max} = 10$ .

Ahora, con cada combinación  $\psi - J$  donde  $J = 1, 2, \dots, J_{max}$ , se realizan  $\eta$  simulaciones y se registra el tiempo requerido  $t_{req}$  por el esquema de control para cumplir con que  $\|e(k)\| \leq \varepsilon_{tol}$ . Enseguida, se calcula el tiempo promedio requerido,  $t_{prom}$ , aplicando la media aritmética por cada combinación

$\psi - J$ . De lo anterior, se obtienen las Figuras 7 y 8 para las familias *Rasp* y *Polywog*, cuyos tiempos promedios requeridos de cada integrante se especifica en la Tabla 6.

Tabla 6: Tiempo promedio requerido por las familias *Rasp* y *Polywog* para alcanzar  $\varepsilon(k) \leq \varepsilon_{tol}$

Wavelet	Tiempo	Wavelet	Tiempo
$\psi$	$t_{prom}$ (s)	$\psi$	$t_{prom}$ (s)
<i>Rasp1</i>	1.8715	<i>Polywog1</i>	3.2218
<i>Rasp2</i>	2.4628	<i>Polywog2</i>	3.9163
<i>Rasp3</i>	1.5285	<i>Polywog3</i>	2.6570
		<i>Polywog4</i>	4.8876
		<i>Polywog5</i>	4.3140

Luego, de cada familia se toma a la *wavelet* con el menor tiempo promedio: *Rasp3* y *Polywog3*, y se comparan con las funciones *Shannon* y *Morlet*. En la Figura 9 se aprecia dicha comparación y el tiempo promedio requerido por cada una de ellas se indica en la Tabla 7. Se determina que la mejor opción de las funciones *wavelets* empleadas en este estudio comparativo es la *Rasp3*, por alcanzar en menor tiempo la condición  $\|\varepsilon(k)\| \leq \varepsilon_{tol}$ .

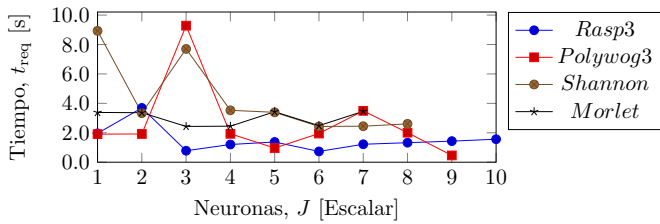


Figura 9: *Wavelets* seleccionadas. Tiempo requerido por el *PID-WaveNet-IIR* para aplicar el control en lazo cerrado con  $J_{max} = 10$ .

Tabla 7: Tiempo promedio requerido por las cuatro mejores *wavelets* para alcanzar  $\varepsilon(k) \leq \varepsilon_{tol}$

Tiempo $t_{prom}$ (s)	Wavelet			
	<i>Rasp3</i>	<i>Polywog3</i>	<i>Shannon</i>	<i>Morlet</i>
	1.5285	2.6570	4.3140	4.2955

### 3.3. Simulación en lazo cerrado utilizando la *Rasp3*

En esta parte se presentan los resultados en simulación numérica en lazo cerrado empleando la configuración del *PID-WaveNet-IIR* obtenida.

En la Figura 10 se muestra el seguimiento de una trayectoria  $y_r$  del helicóptero y se observa que la *PID-WaveNet-IIR* es capaz de identificar la dinámica del modelo lineal y acotar al error de seguimiento  $\varepsilon(k)$ . Además, en las Figuras 11 y 12 se muestra el comportamiento de los pesos sinápticos de la red y los parámetros de la función *wavelet* *Rasp3*. Mientras que, en las Figuras 13 y 14, se aprecia la variación de los coeficientes de adelanto y retardo de los filtros IIR durante la simulación.

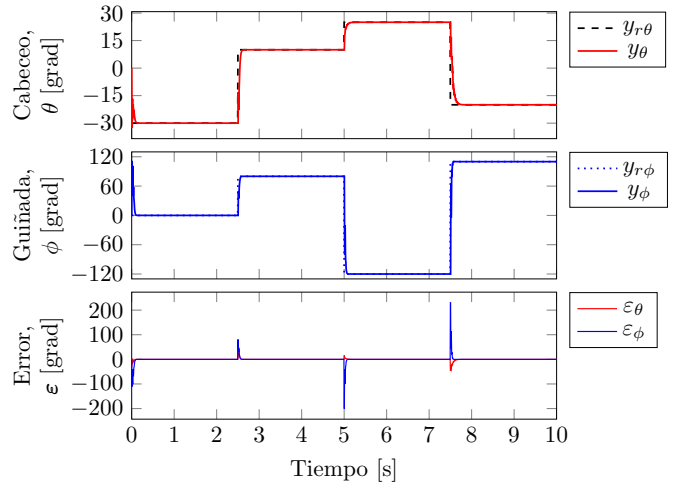


Figura 10: Identificación del modelo lineal en lazo cerrado y errores de seguimiento, donde  $y_{r\theta}$  y  $y_{r\phi}$  son las señales referencia en el cabeceo y guiñada, respectivamente.

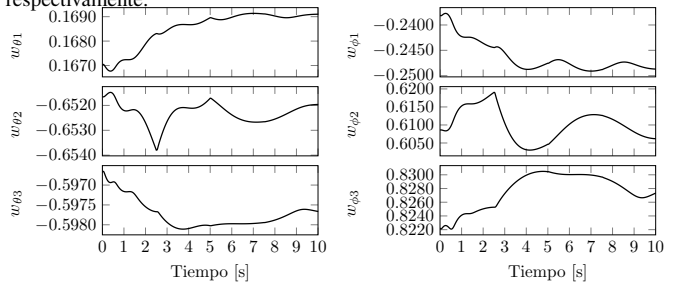


Figura 11: Comportamiento de los pesos sinápticos en cada eje.

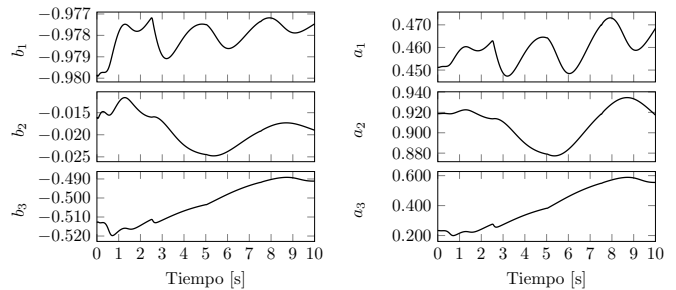


Figura 12: Comportamiento de los parámetros de traslación  $b$  y escalamiento  $a$  en cada *wavelet* "hija".

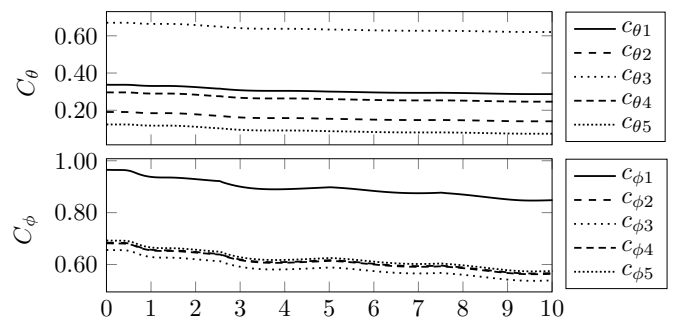


Figura 13: Comportamiento de los coeficientes de retardo en cada eje de movimiento.

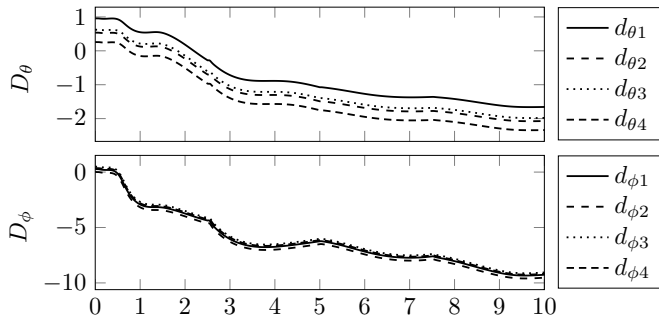


Figura 14: Comportamiento de los coeficientes de adelanto en cada eje de movimiento.

### 3.3.1. Auto sintonización de ganancias PID

A partir de la simulación mostrada en la Figura 10 donde se utiliza a la *Rasp3*, se obtiene el comportamiento de las ganancias  $\mathbf{p}(k)$ ,  $\mathbf{i}(k)$  y  $\mathbf{d}(k)$ , las cuales mantienen cierto valor y solo cambian ante algún cambio discontinuo en la trayectoria deseada, como se aprecia en la Figura 15.

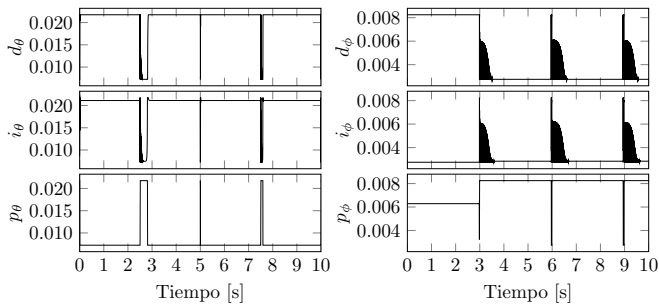


Figura 15: Comportamiento de las ganancias de los controladores PID para los eje de cabeceo  $\theta$  y guiñada  $\phi$ .

Además, se genera la señal de control  $\mathbf{u}(k)$  que, al ser transformada con (3) y (4) se obtiene la Figura 16; donde se aprecia que los voltajes de los motores están dentro de los parámetros de operación indicados en la Tabla 1.

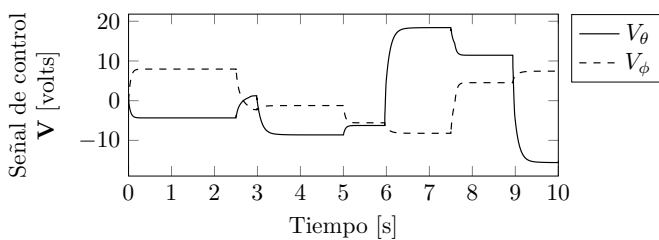


Figura 16: Comportamiento de la señal de control  $\mathbf{u}(k)$ .

## 4. Conclusiones y trabajos futuros

### Conclusiones

A partir de los resultados mostrados, es posible concluir que el esquema de control *PID-WaveNet-IIR* para el modelo lineal del helicóptero de 2 GDL es factible, debido a que se tiene buen desempeño en lazo cerrado. Además, del estudio comparativo entre las *wavelets* empleadas, la función que mejor desempeño mostró es la *Rasp3*, pues mantiene un tiempo requerido menor

a 2 segundos para todos los posibles casos estudiados de variación en el número de neuronas.

### Trabajos futuros

Actualmente, se trabaja con el esquema de control propuesto *PID-WaveNet-IIR* para controlar su modelo no lineal con la finalidad de hacer un estudio comparativo similar al presentado en este artículo. Para su posterior implementación en el prototipo real del laboratorio.

## Referencias

- Abuhamdia, T. y Taheri, S. (2017). Wavelets as a tool for systems analysis and control. *Journal of Vibration and Control*, 23:1377 – 1416.
- Åström, K. (1995). *PID Controllers*. Instrument Society of America.
- Åström, K. (1996). *Advanced PID control*. Instrument Society of America.
- Barrón-Gómez, R., Ramos-Velasco, L. E., Espinoza-Quesada, E., y García-Carrillo, L. (2017). Wavelet neural network pid controller for a uas transporting a cable-suspended load. *20th World Congress, IFAC*, pp. 2371–2376.
- Carrillo-Santos, C., Seck-Tuoh-Mora, J., Hernández-Romero, N., y Ramos-Velasco, L. (2018). Wavenet identification of dynamical systems by a modified PSO algorithm. *Eng Appl Artif Intell*, 79:1–9.
- Deisenroth, M., Faisal, A., y Ong, C. (2020). *Mathematics for machine learning*. Cambridge University Press.
- Domínguez-Mayorga, C. R., Espejel-Rivera, M., Ramos-Velasco, L. E., Ramos-Fernández, J., y Escamilla Hernández, E. (2012). Wavenet algorithms with applications in approximation signals: a comparative study. *RIAI*, 9:347–358.
- García-Castro, O. F. (2020). Controladores PID wavenet aplicados a un modelo de helicóptero de 2 GDL.
- Ge, S. S. y Wang, C. (2002). Direct adaptive nn control of a class of nonlinear systems. *IEEE Trans Neural Netw*, 13(1):214–221.
- Haykin, S. (1999). *Neural networks: a comprehensive foundation*. Pearson Education Inc, EUA.
- Huang, S., Tan, K. K., y Lee, T. H. (2002). Adaptive motion control using neural network approximations. *Automatica*, 38(2):227–233.
- Huang, S., Tan, K. K., y Putra, A. S. (2007). Adaptive control of mechanical systems using neural networks. *IEEE Trans Syst Man*, 37(5):897–903.
- Inc., Q. (2012). *User manual 2 DOF helicopter experiment, set up and configuration*. Markham, Ontario.
- Li, Y., Qiang, S., Zhuang, X., y Kaynak, O. (2004). Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. *IEEE Trans Neural Netw*, 14(3):693–701.
- Mallat, S. (2009). *A wavelet tour of signal processing*. Elsevier.
- Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Prentice Hall.
- Ramos-Velasco, L. E., Domínguez-Ramírez, O., y Parra-Vega, V. (2016). Wavenet fuzzy PID controller for nonlinear mimo systems: experimental validation on a high-end haptic robotic interface. *Applied Soft Computing*, pp. 199–205.
- Sanner, R. M. y Slotine, J. E. (1992). Gaussian networks for direct adaptive control. *IEEE Trans Neural Netw*, 3(6):837–863.
- Seshagiri, S. y Khalil, H. K. (2000). Output feedback control of nonlinear systems using RBF networks. *IEEE Trans Neural Netw*, 11(1):69–79.
- Sundararajan, N., Saratchandran, P., y Li, Y. (2002). *Fully tuned radial basis function neural networks for flight control*. Kluwer, Boston.
- Tangirala, A. K., Mukhopadhyay, S., y Tiwari, A. P. (2013). Chapter three - wavelets applications in modeling and control. *Control and Optimisation of Process Systems*, 43:107–204.
- Vega-Navarrete, M. A., Ramos-Velasco, L. E., Arizpe-Carreón, P. A., Delgado-Romero, V., Valle-Hernández, J., y Parra-Vega, V. (2018). Output feedback self-tuning wavenet control for underactuated euler-lagrange systems. *The 2nd IFAC Conference on Modelling (MICNON 2018)*, pp. 663–668.
- Wang, S. W. y Yu, D. L. (2007). Adaptive RBF network for parameter estimation and stable air–fuel ratio control. *Neural Netw*, 21(1):102–112.
- Zhu, Q., Fei, S., Zhang, T., y Li, T. (2008). Adaptive RBF neural-networks control for a class of timedelay nonlinear systems. *Neural Netw*, 71(16–18):3617–3624.