



UNIVERSIDAD AUTÓNOMA  
DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

---

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL

DISEÑO Y APLICACIONES DE UN CONTROLADOR PID *WAVELET*

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN  
CIENCIAS EN AUTOMATIZACIÓN Y CONTROL

**PRESENTA:**

**ING. JOSÉ ALBERTO CRUZ TOLENTINO**

ASESOR:

**DR. LUIS ENRIQUE RAMOS VELASCO**

PACHUCA HGO., MÉXICO 16 DE DICIEMBRE DE 2009





Ing. José Alberto Cruz Tolentino

**DISEÑO Y  
APLICACIONES DE UN  
CONTROLADOR PID  
*WAVELET***



*El presente trabajo, fruto del esfuerzo de dos años,  
está dedicado a mis padres,  
quienes me han educado y enseñado,  
me han dado su cariño y comprensión,  
me han dado los medios para concluir  
de manera exitosa todos mis proyectos.  
Por todo eso y mucho más, gracias.*



Gracias a

**CONACYT** por la beca de Maestría  
otorgada durante el periodo  
Febrero 2008 - Junio 2009, con  
número de registro 216424.





# Agradecimientos

Siempre fue mi deseo hacer de esta sección lo más breve posible, sin embargo, hay tantas personas que de una u otra manera han dado vida a este trabajo, que no puedo dejar de extenderme con el fin de hacer llegar mi más sincera gratitud a cada uno de ellos, porque precisamente es eso lo menos que puedo hacer.

Agradezco a la Universidad Autónoma del Estado de Hidalgo, institución que me dio los medios para culminar de manera exitosa la *Maestría en Automatización y Control* y de este modo crecer personal y profesionalmente.

Gracias a mis padres por su amor incondicional, quienes me han enseñado a vivir, quienes han velado la realización de mis metas, quienes me han cuidado y educado, quienes han dado sangre y sudor para mantenernos unidos, no encuentro palabras suficientes para expresar mi gratitud, seguro estoy de que esté y muchos otros proyectos no habrían alcanzado feliz término si ellos no estuvieran ahí. Gracias a mis hermanos por su amor, por su amistad, por saber escuchar y darme tantos y tantos momentos de felicidad, por sus consejos y confidencias, por su paciencia, porque mi vida sería tan vacía sin ellos.

En la elaboración y redacción de esta tesis he de agradecer de especial manera a mi asesor Dr. Luis Enrique Ramos Velasco, por todo el tiempo dedicado a desarrollar este trabajo de investigación, por su paciencia, por sus atinados comentarios y sobre todo por su amistad. Del mismo modo agradezco a la M. en C. María Angélica Espejel Rivera quien siempre tuvo algo que aportar, dedicando tiempo y paciencia y sin el cual este trabajo no sería el mismo.

A el Dr. Omar Arturo Domínguez Ramírez, Dr. Omar Jacobo Santos Sánchez, Dr. Virgilio López Morales, quienes revisaron este trabajo y realizaron importantes acotaciones, sus observaciones y comentarios mejoraron de sobremanera la esencia de esta tesis.

Finalmente, pero no menos importante, agradezco a todos los que de manera indirecta hicieron una contribución a este trabajo, gracias a sus estudios y publicaciones, a todos los hombres de ciencia y letras que de uno u otro modo han dado vida al presente documento y que además han aportado algo a mi formación.



# Resumen

## DISEÑO Y APLICACIONES DE UN CONTROLADOR PID *WAVELET*

En el presente trabajo de tesis se diseñaron dos controladores del tipo multiresolución, uno llamado PID *wavelet*, el cual hace uso de la técnica multiresolución basada en la teoría *wavelet* para la descomposición de la señal de error, para obtener varias señales a diferentes escalas-frecuencias las cuales son escaladas y sumadas para generar una señal de control que compensa las incertidumbres del sistema. El otro controlador se le llama PID *wavenet*, que básicamente consiste de un PID *wavelet* cuyas ganancias son sintonizadas y adaptadas mediante redes neuronales *wavelet*. Además se presentan las siguientes aplicaciones de los controladores diseñados:

1. PID *wavelet* en simulación numérica y experimentación:
  - Control de posición de un motor de CD.
  - Control de posición de un robot planar.
  - Control de temperatura de un sistema térmico, donde sólo se aplico en simulación numérica.
2. PID *wavenet* en simulación numérica:
  - Control de posición de un motor de CD.



# Abstract

## Design and applications of an *wavelet* PID controller

In this thesis we designed two controllers of the type multiresolution, The first controller is named *wavelet* PID, this controller uses multiresolution technique based on wavelet to decompose the error signal into different scale-frequency and then the resultant signals are added and scaled by their respective gains to generate the control signal to compensate the uncertainties of the system. The second controller is named *wavenet* PID, this controller consists of an *wavelet* PID whose gains are tuning and adaptive using *wavelet* neural networks. In addition we present the applications of the controllers designed:

1. *Wavelet* PID in simulate numerical and experimentation:
  - Position Control of the a motor of CD.
  - Position Control of a planar robot.
  - Temperature Control of a thermal system, where only it apply in simulation numerical.
2. *Wavenet* PID in simulate numerical:
  - Position Control of a motor of CD.



# Publicaciones

Como parte de los resultados del trabajo de investigación desarrollado en esta tesis, se obtuvieron los siguientes artículos:

## Artículos en congresos nacionales

### Aceptados

- **J. A. Cruz Tolentino, L. E. Ramos Velasco & M. A. Espejel Rivera.**, *PID Wavelet Auto-Sintonizado con una Red Neuronal Wavenet*, V Semana Nacional de Ingeniería Electrónica (SENIE'09), Ocotlán, Jalisco, México 7-9 Noviembre 2009.
- **J. A. Cruz Tolentino, L. E. Ramos Velasco & M. A. Espejel Rivera.**, *Una aplicación de las wavelets en Control Automático*, 3er. Congreso Nacional de Mecatrónica, Zempoala, Hidalgo, México 25-27 Noviembre 2009.

## Artículos en congresos internacionales

### Sometidos

- **J. A. Cruz Tolentino, L. E. Ramos Velasco & M. A. Espejel Rivera.**, *A Self-Tuning Wavelet PID Controller Using Wavelet Networks*, 20th International Conference on Electronics Communications and Computers (CONIELECOMP 2010), Cholula, Puebla, México Febrero 2010.





# Índice general

Dedicatoria . . . . .	III
Agradecimientos . . . . .	VII
Resumen . . . . .	IX
Abstract . . . . .	XI
Publicaciones . . . . .	XIII
Índice general . . . . .	XV
Índice de figuras . . . . .	XIX
Índice de tablas . . . . .	XXV
<b>Acrónimos y abreviaturas</b>	<b>xxvii</b>
<b>Notación</b>	<b>xxix</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción de la problemática . . . . .	2
1.2. Descripción de la propuesta de solución . . . . .	2
1.3. Objetivos de la tesis . . . . .	2
1.4. Hipótesis . . . . .	3
1.5. Justificación . . . . .	3
1.6. Aportaciones . . . . .	4
1.7. Organización de la tesis . . . . .	4
<b>2. Estudio del estado del arte</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Teoría <i>wavelet</i> . . . . .	6
2.2.1. Trabajos previos sobre <i>wavelet</i> . . . . .	7
2.3. Redes neuronales artificiales . . . . .	8
2.4. Teoría <i>wavenet</i> . . . . .	9
2.4.1. Trabajos previos sobre <i>wavenet</i> . . . . .	9
2.4.2. Trabajos previos realizados por alumnos del CITIS . . . . .	10
2.5. Control clásico . . . . .	10

2.6. Comentarios . . . . .	12
<b>3. Desarrollo de un control PID <i>wavelet</i></b>	<b>13</b>
3.1. Introducción . . . . .	13
3.2. Control PID multiresolución . . . . .	14
3.3. Análisis multiresolución . . . . .	17
3.4. Algoritmos del PID <i>wavelet</i> . . . . .	24
3.5. Resultados de simulación e implementación . . . . .	24
3.5.1. Motor de CD . . . . .	26
3.5.2. Sistema térmico . . . . .	33
3.5.3. Robot planar de 2 grados de libertad . . . . .	36
3.6. Comentarios . . . . .	43
<b>4. Desarrollo de un control PID <i>wavenet</i></b>	<b>45</b>
4.1. Introducción . . . . .	45
4.2. Sistemas no lineales discretos . . . . .	46
4.3. Algoritmo de sintonización del control <i>wavenet</i> . . . . .	47
4.4. Resultados de simulación . . . . .	57
4.5. Comentarios . . . . .	62
<b>5. Resultados de laboratorio</b>	<b>63</b>
5.1. Implementación del control PID <i>wavelet</i> . . . . .	63
5.1.1. Implementación sobre un sistema de control de posición . . . . .	63
5.1.2. Implementación sobre un robot planar de dos grados de libertad . . . . .	68
<b>6. Conclusiones y trabajos futuros</b>	<b>73</b>
6.1. Conclusiones. . . . .	73
6.2. Trabajos futuros. . . . .	73
<b>Glosario</b>	<b>75</b>
<b>Bibliografía</b>	<b>81</b>
<b>A. Conceptos previos</b>	<b>89</b>
A.1. Conceptos generales . . . . .	89
A.2. Comentarios . . . . .	103
<b>B. Teoría wavelet</b>	<b>105</b>
B.1. Introducción . . . . .	105
B.2. Análisis de Fourier . . . . .	106
B.2.1. Funciones periódicas . . . . .	106

B.2.2.	Coeficientes y series de Fourier . . . . .	106
B.2.3.	Transformada de Fourier . . . . .	108
B.2.4.	Limitaciones del análisis de Fourier . . . . .	108
B.2.5.	Transformada de Fourier ventaneada . . . . .	109
B.3.	Teoría <i>wavelet</i> . . . . .	110
B.3.1.	Transformada <i>wavelet</i> . . . . .	113
B.3.1.1.	Transformada <i>wavelet</i> continua . . . . .	114
B.3.1.2.	Transformada <i>wavelet</i> discreta . . . . .	116
B.4.	Análisis multiresolución basada en <i>wavelets</i> . . . . .	117
B.5.	Análisis sub-banda . . . . .	120
B.5.1.	Análisis de descomposición . . . . .	122
B.5.2.	Síntesis de descomposición . . . . .	124
B.6.	Comentarios . . . . .	126
<b>C.</b>	<b>Redes neuronales artificiales</b>	<b>127</b>
C.1.	Introducción . . . . .	127
C.2.	El perceptrón y el percentrón multicapa . . . . .	128
C.3.	Redes neuronales con funciones de base radial . . . . .	132
C.4.	Comentarios . . . . .	134
<b>D.</b>	<b>Teoría <i>wavenet</i></b>	<b>135</b>
D.1.	Introducción . . . . .	135
D.2.	Algoritmos <i>wavenets</i> . . . . .	136
D.3.	Comentarios . . . . .	140
<b>E.</b>	<b>Implementación de un filtro <i>wavelet</i></b>	<b>141</b>
E.1.	Introducción . . . . .	141
E.2.	Filtro <i>wavelet</i> . . . . .	141
E.3.	Implementación . . . . .	142
<b>F.</b>	<b>Algoritmos programados en MATLAB y <i>simulink</i></b>	<b>145</b>
F.1.	Algoritmos del PID <i>wavelet</i> empleados para la simulación . . . . .	145
F.1.1.	MATLAB . . . . .	145
F.1.2.	<i>simulink</i> . . . . .	147
F.2.	Algoritmos del PID <i>wavelet</i> empleados para la implementación experimental en <i>simulink</i> . . . . .	151
F.3.	Algoritmo del filtro <i>wavelet</i> . . . . .	158
F.4.	Algoritmos del PID <i>wavenet</i> en Matlab . . . . .	164



# Índice de figuras

1.1. Esquema simple de un control clásico, donde $y_{ref}$ , $e$ , $u$ , $y$ , $P$ y $r$ , son: la señal de referencia del sistema, el error entre la salida y la referencia, la señal de control, la salida del sistema, las perturbaciones externas al sistema y el ruido en la medición, respectivamente. . . . .	1
2.1. Áreas requeridas para el desarrollo de la tesis. . . . .	6
3.1. Esquema simple de un control PID clásico, donde $y_{ref}$ , $e$ , $u$ , $y$ , $P$ y $r$ , son la referencia del sistema, el error entre la salida y la referencia, la señal de control, la salida del sistema, las perturbaciones externas al sistema y el ruido en la medición, respectivamente. . . . .	14
3.2. Diagrama simple del control PIDMR, donde $y_{ref}$ , $e$ , $u$ y $y$ , son la señal de referencia del sistema, señal de error entre la salida y la referencia, la señal de control y la señal de salida del sistema, respectivamente. Aquí $K_H, K_{M_1}, \dots, K_{M_{N-1}}$ y $K_L$ son las ganancias a sintonizar del control PIDMR [69]. . . . .	17
3.3. Señal de entrada $e(x)$ original. . . . .	22
3.4. Descomposición de la señal $e(x)$ a diferentes niveles de resolución, donde $x$ es el número de muestras y $e_H^{(3)}, e_L^{(3)}, e_L^{(2)}, e_L^{(1)}$ las componentes de la señal original. . . . .	23
3.5. Diagrama de un esquema de decodificación sub-banda, para el análisis y síntesis de la descomposición en la parte superior y en la parte inferior de la figura, respectivamente. Donde $h$ y $g$ , representan los filtros pasa bajas y pasa altas, respectivamente y $\overline{h[k]}$ , $\overline{g[k]}$ representan sus complejos conjugados, respectivamente, $\downarrow 2$ y $\uparrow 2$ , representan la decimación e interpolación a 2, respectivamente. . . . .	25
3.6. Respuesta del sistema con control PID clásico y el PID <i>wavelet</i> para una referencia de $\pi/2$ rad. . . . .	27
3.7. Señal de control del PID clásico y del PID <i>wavelet</i> para una referencia de $\pi/2$ rad. . . . .	27

3.8. Señal de error del sistema con un control del PID clásico y un PID <i>wavelet</i> para una referencia de $\pi/2$ rad. . . . .	29
3.9. Señales generadas a partir de la descomposición a un nivel 3 del error del sistema con un control PID <i>wavelet</i> para una referencia de $\pi/2$ rad. . . . .	29
3.10. Señal de ruido blanco introducido en el sistema. . . . .	30
3.11. Respuesta del sistema con control PID clásico y el PID <i>wavelet</i> para una referencia de $\pi/2$ rad. ante la presencia de ruido blanco. . . . .	31
3.12. Señal de control del PID clásico y del PID <i>wavelet</i> con ruido blanco. . . . .	31
3.13. Señales de error para un sistema con un control PID clásico y un PID <i>wavelet</i> ante la presencia de ruido blanco. . . . .	32
3.14. Señales generadas a partir de la descomposición a un nivel 3 del error del sistema con un control PID <i>wavelet</i> con ruido. . . . .	32
3.15. Sistema térmico, donde $T_e$ , $T_{ec}$ , $M$ son la temperatura del tanque, la temperatura a la salida de la válvula mezcladora, la masa del fluido contenido en el tanque, respectivamente. . . . .	33
3.16. Señal de ruido blanco introducido en el sistema con retardo. . . . .	34
3.17. Respuesta del sistema con retardo con control PID clásico y del PID <i>wavelet</i> para una referencia de 100 grados centígrados, con un señal de ruido inyectado. . . . .	35
3.18. Señal de control del PID clásico y la señal de control del PID <i>wavelet</i> , ante la presencia de un señal de ruido. . . . .	35
3.19. Señal de error del sistema con retardo controlado con un PID clásico y con un PID <i>wavelet</i> , ante la presencia de un señal de ruido blanco. . . . .	36
3.20. Respuesta del sistema con retardo con control PID clásico y del PID <i>wavelet</i> para una referencia de 100 grados centígrados, con una señal de ruido inyectado. . . . .	37
3.21. Señal de control del PID clásico y la señal de control del PID <i>wavelet</i> , ante la presencia de una señal de ruido. . . . .	37
3.22. Señal del error del sistema con retardo controlado con un PID clásico y la señal del error del sistema con retardo controlado con un PID <i>wavelet</i> , ante la presencia de un señal de ruido. . . . .	38
3.23. Señales generadas a partir de la descomposición a un nivel 3 de la señal del error del sistema con retardo controlado con un PID <i>wavelet</i> , ante la presencia de una señal de ruido. . . . .	38
3.24. Sistema del robot planar de dos grados de libertad sin acción de gravedad, donde $m_1$ , $m_2$ , $l_1$ , $l_2$ , $q_1$ , $q_2$ son la masa en el eslabón 1, la masa en el eslabón 2, la longitud del eslabón 1, la longitud del eslabón 2, la variable articular 1 y la variable articular 2, respectivamente; y $u_1$ , $u_2$ son el par aplicado al actuador 1 y al actuador 2, respectivamente. . . . .	39
3.25. Señal de ruido blanco introducido al robot planar. . . . .	40

3.26. Descripción de la tarea a seguir del robot con un control PID clásico y un control PID <i>wavelet</i> . . . . .	41
3.27. Comportamiento de las variables articulares $q_1$ y $q_2$ con un control PID clásico y con un control PID <i>wavelet</i> . . . . .	41
3.28. Señales de control para los actuadores 1 y 2, generados con un control PID clásico y con un control PID <i>wavelet</i> , donde: $u_1$ y $u_2$ son las señales de control para $\tau_1$ y $\tau_2$ , respectivamente. . . . .	42
3.29. Señales de error $e_1$ y $e_2$ . . . . .	42
4.1. Diagrama a bloques del PID <i>wavenet</i> , donde $e$ y $e_n$ son el error entre la señal de referencia $y_{ref}$ y la salida de la plana $y$ , y el error entre la salida de la planta $y$ ; y la salida estimada por la red neuronal <i>wavenet</i> $\hat{y}$ , respectivamente; $u$ , $v$ y $\hat{\Gamma}$ , son la señal de control, el coeficiente de velocidad de aprendizaje y el coeficiente de no linealidades estimado, respectivamente; $\mathbf{K}$ y $\mathbf{E}_m$ son el vector de las ganancias del controlador <i>wavelet</i> actualizados y el vector de los componentes de la señal del error $e$ , respectivamente; $P$ y $r$ son las perturbaciones externas al sistema y el ruido en la medición, respectivamente. . . . .	48
4.2. Señal de salida del sistema con un control PID <i>wavenet</i> y con un control PID <i>wavelet</i> . . . . .	59
4.3. Señal de control generada por el control PID <i>wavenet</i> y el PID <i>wavelet</i> . . . . .	59
4.4. Descomposición de la señal del error del sistema con un control PID <i>wavenet</i> . . . . .	60
4.5. Comportamiento de los parámetros de la red neuronal <i>wavelet</i> , durante la simulación. . . . .	60
4.6. Comportamiento de los parámetros del filtro IIR, durante la simulación. . . . .	61
4.7. Comportamiento de las ganancias del control PID <i>wavenet</i> , durante la simulación. . . . .	61
5.1. Sistema de control de movimiento. . . . .	64
5.2. Objetivo de control. . . . .	64
5.3. Diagrama a bloques del sistema de control de movimiento. . . . .	65
5.4. Resultados de la implementación para la señal de posición del sistema de posición. . . . .	66
5.5. Resultados de la implementación para la señal de control del sistema de posición. . . . .	66
5.6. Resultados de la implementación para la señal de error en la posición del sistema de posición. . . . .	67
5.7. Resultados de la implementación para la señal de error a un nivel de descomposición $N = 3$ . . . . .	67

5.8. Robot planar de dos grados de libertad. . . . .	69
5.9. Resultados de la posición para el actuador 1 y 2. . . . .	69
5.10. Resultados de la señal de control para el actuador 1 y 2. . . . .	70
5.11. Resultados de la señal de error en el actuador 1 y 2. . . . .	70
5.12. Señal de error del actuador 1 y 2 a un nivel de descomposición $N = 3$ . . . . .	71
B.1. Representación de una función periódica $f(x)$ , con período $P$ . . . . .	106
B.2. Representación gráfica de (B.13), donde $g(t)$ es la función de Gauss, $t$ es el tiempo, $\sigma$ es el factor de desplazamiento. . . . .	110
B.3. Gráficas de las <i>wavelets</i> madre dadas en la Tabla B.1. . . . .	113
B.4. Plano tiempo-frecuencia para diferentes bases ortogonales: a) base de Fourier, b) base de Gabor y c) base <i>wavelet</i> . . . . .	118
B.5. Arquitectura de una red neuronal <i>wavelet</i> SISO. . . . .	121
B.6. Análisis de descomposición de la función $f[n]$ . . . . .	121
B.7. Síntesis de descomposición de la señal $f[n]$ . . . . .	122
B.8. Diagrama de una convolución seguida de una decimación a dos, donde $c_{i-1}[n]$ , $c_{i,k}[n]$ y $\nu_{i,k}[n]$ son la señal de entrada, la señal de salida y la señal resultante de la convolución entre la entrada y el filtro conjugado $\overline{h[k]}$ de longitud $\kappa$ . . . . .	123
B.9. Diagrama de una convolución seguida de una decimación a dos, donde $d_{i-1}[n]$ , $d_{i,k}[n]$ y $\nu_{i,k}[n]$ son la señal de entrada, la señal de salida y la señal resultante de la convolución entre la entrada y el filtro conjugado $\overline{g[k]}$ de longitud $\kappa$ . . . . .	124
B.10. Diagrama de una interpolación a dos seguida de una convolución, donde $c_i[n]$ , $c'_{i,k}[n]$ y $\xi_{i,k}[n]$ son la señal de entrada, la señal de salida y la señal resultante de la interpolación a dos de la entrada, y $g[k]$ es el filtro de longitud $\kappa$ . . . . .	124
B.11. Diagrama de una interpolación a dos seguida de una convolución, donde $d_i[n]$ , $d'_{i,k}[n]$ y $\zeta_{i,k}[n]$ son la señal de entrada, la señal de salida y la señal resultante de la interpolación a dos de la entrada, y $g[k]$ es el filtro de longitud $\kappa$ . . . . .	125
C.1. Modelo de McCulloch-Pitts de una neurona, donde $x_l$ , $w_l$ , $f(\cdot)$ , $y$ , y $v_t$ , son la $l$ -ésima señal de entrada, el peso que conecta la $l$ -ésima entrada con el CPU, la función de activación, la señal de salida y el umbral, respectivamente. . . . .	128
C.2. Estructura de un perceptrón de una capa, donde $x_j$ , $w_{j,l}$ , $f(\cdot)$ , $y_l$ , $v_t$ , son la $j$ -ésima señal de entrada, el peso que conecta la $j$ -ésima entrada con $l$ -ésimo nodo de la capa de salida, la función de activación, la $l$ -ésima señal de salida y el umbral, respectivamente. . . . .	129



C.3.	Estructura de un perceptrón multicapa de tres capas, donde $x_j$ , $w_{j,l}$ , $w_{l,k}$ , $f(\cdot)$ , $y_l$ , $v_t$ , son la $j$ -ésima señal de entrada, el peso que conecta la $j$ -ésima entrada con $l$ -ésimo nodo en la primera capa, el peso que conecta el $l$ -ésimo nodo de la primera capa con el $k$ -ésimo nodo de la capa de salida, la función de activación, la $l$ -ésima señal de salida y el umbral, respectivamente. . . . .	130
C.4.	Estructura de una red neuronal con funciones de base radial, donde $x_j$ , $\mathbf{C}_l$ , $\mathbf{W}_i$ , $\varphi(\cdot)$ y $y_i$ son la entrada en el nodo $j$ , el vector de pesos del nodo $l$ , el vector de pesos del nodo $i$ , la función de activación y la salida en el nodo $i$ , respectivamente. . . . .	133
D.1.	Diagrama de una red neuronal, donde $u(t)$ , $\hat{y}$ son la señal entrada, la señal de salida de la red neuronal, respectivamente y $\psi(\tau)$ , $w$ son la función de activación y los pesos que hay entre las neuronas, respectivamente. . . . .	136
D.2.	Diagrama del filtro IIR, donde $z(t)$ , $\hat{y}$ son la señal de entrada, la señal de salida del filtro IIR, respectivamente y $c$ , $d$ , son los coeficientes del filtro. . . . .	137
D.3.	Diagrama de la <i>wavenet</i> con el filtro IIR, donde $u(t)$ , $\hat{y}$ son la señal de entrada, la señal de salida de la <i>wavenet</i> con un filtro IIR, respectivamente, $a$ , $b$ , $w$ y $c$ , $d$ , son los parámetros de la <i>wavenet</i> y coeficientes del filtro IIR, respectivamente. . . . .	138
E.1.	Diagrama de descomposición de una señal $f(x) = C_0$ en una señal de baja frecuencia (alta escala) $C_1$ y una señal de alta frecuencia (baja escala) $D_1$ . . . . .	142
E.2.	Diagrama del filtro <i>wavelet</i> con un nivel de filtrado de $N = 5$ . . . . .	142
E.3.	Señal de velocidad articular $\dot{q}_1$ con y sin filtro digital <i>wavelet</i> . . . . .	143
E.4.	Señal de velocidad articular $\dot{q}_2$ con y sin filtro digital <i>wavelet</i> . . . . .	143
E.5.	Señal de velocidad articular $\dot{q}_3$ con y sin filtro digital <i>wavelet</i> . . . . .	144
F.1.	Bloque del control <i>wavelet</i> . . . . .	148
F.2.	Bloque del control <i>wavelet</i> tipo I. . . . .	149
F.3.	Bloque del control PID clásico. . . . .	149
F.4.	Bloque para la simulación de los controles en un motor de CD. . . . .	150
F.5.	Bloque para la simulación de los controles en un motor de CD con ruido blanco inyectado en la medición. . . . .	150
F.6.	Bloque para la simulación de los controles en un sistema con retardo y con ruido blanco inyectado en la medición. . . . .	151
F.7.	Bloque para la simulación del control PID para el robot planar de 2 grados de libertad, con ruido blanco inyectado en la medición. . . . .	152

---

F.8. Bloque para la simulación del control PID <i>wavelet</i> para el robot planar de 2 grados de libertad, con ruido blanco inyectado en la medición. . .	152
F.9. Bloque para la simulación de la trayectoria para el robot planar de 2 grados de libertad. . . . .	153
F.10. Bloque para la simulación del robot planar de 2 grados de libertad. . .	153
F.11. Bloque de las constantes $a_1$ , $a_2$ y $a_3$ en el robot planar de 2 grados de libertad. . . . .	154
F.12. Bloque del Sistema A en el robot planar de 2 grados de libertad. . . .	155
F.13. Bloque del Sistema B en el robot planar de 2 grados de libertad. . . .	156
F.14. Bloque del Sistema C en el robot planar de 2 grados de libertad. . . .	156
F.15. Bloque del control PID para el motor de CD. . . . .	157
F.16. Bloque del control PID <i>wavelet</i> para el motor de CD. . . . .	157
F.17. Bloque del control PID para el robot planar de 2 grados de libertad. .	157
F.18. Bloque del control PID <i>wavelet</i> para el robot planar de 2 grados de libertad. . . . .	158

# Índice de tablas

3.1. MATLAB: Comandos para la descomposición . . . . .	21
3.3. ALGORITMO: Algoritmo de control PIDMR . . . . .	24
3.2. Coeficientes del filtro Daubechies de orden 2. . . . .	26
3.4. Parámetros del motor de CD. . . . .	28
3.5. Valores de las ganancias para el control PID clásico y el PID <i>wavelet</i> para el sistema (3.44). . . . .	28
3.6. Valores de las ganancias para el control PID clásico y el PID <i>wavelet</i> para el sistema con retardo (3.45). . . . .	33
3.7. Valores de las ganancias para el PID <i>wavelet</i> para el sistema con retardo (3.45). . . . .	34
3.8. Valores de las ganancias para el PID clásico para el robot planar (3.47). . . . .	40
3.9. Valores de las ganancias para el PID <i>wavelet</i> para el robot planar (3.47). . . . .	40
4.1. Algunas <i>Wavelet</i> madres y su respectiva derivada con respecto a $b$ . . . . .	57
4.2. Parámetros de la red neuronal <i>wavelet</i> . . . . .	57
4.3. Valores iniciales de los parámetros de la red neuronal <i>wavelet</i> y las ganancias del control PID <i>wavenet</i> . . . . .	58
4.4. Valores finales de los parámetros de la red <i>wavenet</i> y las ganancias del control PID <i>wavenet</i> . . . . .	58
5.1. Valores de las ganancias para el control PID clásico y el PID <i>wavelet</i> para el sistema para la plataforma mostrada en la Figura 5.1. . . . .	65
5.2. Valores de las ganancias para el PID clásico para el robot planar. . . . .	68
5.3. Valores de las ganancias para el PID <i>wavelet</i> para el robot planar. . . . .	68
B.1. Algunos ejemplos comunes de <i>wavelets</i> madre. . . . .	112
B.2. ALGORITMO: Algoritmo de análisis de descomposición . . . . .	124
B.3. ALGORITMO: Algoritmo de síntesis de descomposición . . . . .	125
C.1. Algunos ejemplos comunes de funciones de activación . . . . .	131
C.2. ALGORITMO: Algoritmo <i>K-means</i> . . . . .	133

F.1. MATLAB: Función de descomposición . . . . .	145
F.2. MATLAB: Función de síntesis . . . . .	146
F.3. MATLAB: Función para un nivel de descomposición $N = 3$ de la señal de error $e$ . . . . .	146
F.4. MATLAB: Empleando el control <i>wavelet</i> . . . . .	147
F.5. MATLAB: Control <i>wavelet</i> para emplearse en <i>simulink</i> . . . . .	147
F.6. MATLAB: Comandos para la función “ <i>fcn</i> ” para el modelo cinemático inverso de posición . . . . .	148
F.7. ALGORITMO: <i>wavelet.h</i> . . . . .	158
F.8. MATLAB: Función para calcular las <i>wavelets</i> hijas a partir de una <i>wavelet</i> madre Morlet . . . . .	164
F.9. MATLAB: Algoritmo de para calcular la salida de la red neuronal <i>wavelet</i> en cada interacción $k$ . . . . .	164
F.10. MATLAB: Algoritmo de para calcular el error entre la salida deseada $y$ y la salida de la red neuronal <i>wavelet</i> <i>gest</i> en el instante $k$ . . . . .	165
F.11. MATLAB: Algoritmo para el cálculo de los gradientes en el instante $k$	165
F.12. MATLAB: Algoritmo para el cálculo del gradiente para las ganancias del control PID <i>wavelet</i> en el instante $k$ . . . . .	166
F.13. MATLAB: Algoritmo para el cálculo de las actualizaciones de los parámetros de la red neuronal <i>wavelet</i> en el instante $k$ . . . . .	166
F.14. MATLAB: Algoritmo para el cálculo de las actualizaciones de lass ganancias del control PID <i>wavelet</i> en el instante $k$ . . . . .	167

# Acrónimos y abreviaturas

Siglas	Descripción
CD	Corriente directa.
CPU	Unidad central de procesamiento.
FEC	Filtro de espejo de cuadratura.
IIR	Filtro de respuesta al impulso infinito.
LMS	Método de mínimos cuadrados.
MATLAB	Laboratorio de matrices.
PID	Control proporcional integral y derivativo.
PIDMR	Controlador PID multiresolución.
RBF	Funciones de base radial.
RMS	Raíz cuadrática media.
RNA	Redes neuronales artificiales.
RPDGDL	Robot planar de 2 grados de libertad.
SISO	Sistema de una entrada y una salida.
TTCF	Transformada de tiempo corto de Fourier.
TWC	Transformada <i>wavelet</i> continua.
TWD	Transformada <i>wavelet</i> discreta.
UAEH	Universidad Autónoma del Estado de Hidalgo.



# Notación

Símbolo	Descripción
$A \triangleq B$	$A$ es por definición igual a $B$ .
$a \in A$	$a$ es un elemento de $A$ .
$A^{-1}$	Inversa de la matriz $A$ .
$A^{n \times m}$	$A$ es una matriz real de dimensión $n \times m$ .
$A \subset B$	$A$ es un subconjunto propio de $B$ .
$A \subseteq B$	$A$ es un subconjunto de $B$ .
$A^T$	$A$ transpuesta.
$\mathbb{C}$	Espacio de todos los números complejos.
$\mathbf{F}$	Campo.
$\mathbf{J}$	Conjunto de todos los números enteros positivos.
$\bar{x}$	Complejo conjugado de $x$ .
$\neq$	Desigualdad.
$A \sim B$	$A$ es equivalente a $B$ .
$\mathbb{K}$	Es $\mathbb{R}$ o $\mathbb{C}$ .
$(V, \langle \cdot, \cdot \rangle)$	Espacio con producto interno.
$(X, d)$	Espacio métrico.
$e^x$	Exponencial de $x$ .
$f : A \rightarrow B$	$f$ mapea el dominio $A$ en $B$ .
$I$	Matriz identidad.
$\mathbf{i}$	$\sqrt{-1}$ .
$\langle \cdot, \cdot \rangle$	Producto interno.
$\cap$	Intersección.
$\mathbb{N}$	Campo de todos los números naturales.
$b \notin A$	$b$ no es un elemento de $A$ .
$\forall$	Para todo.
$A \perp B$	$A$ es perpendicular a $B$ .
$\mathbb{Q}$	Campo de todos los números racionales.
$\mathbb{R}$	Espacio de todos los números reales.
$\mathbb{R}^n$	Espacio vectorial de los reales de dimensión $n$ .

Símbolo	Descripción
$\sum$	Sumatoria.
$\oplus$	Sumatoria directa de subespacios.
$x \rightarrow x_n$	$x$ tiende a $x_n$ .
$\cup$	Unión.
$\emptyset$	Conjunto vacío.
$\hat{x}$	$x$ estimada.



# Capítulo 1

## Introducción

Un esquema simple de un control básico está conformado por tres bloques como se muestra en la Figura 1.1: La planta que es el sistema a ser controlado, el cual puede ser afectado por perturbaciones externas, un sensor que mide la variable de interés y finalmente el controlador que hace que la planta se comporte de una manera predeterminada.

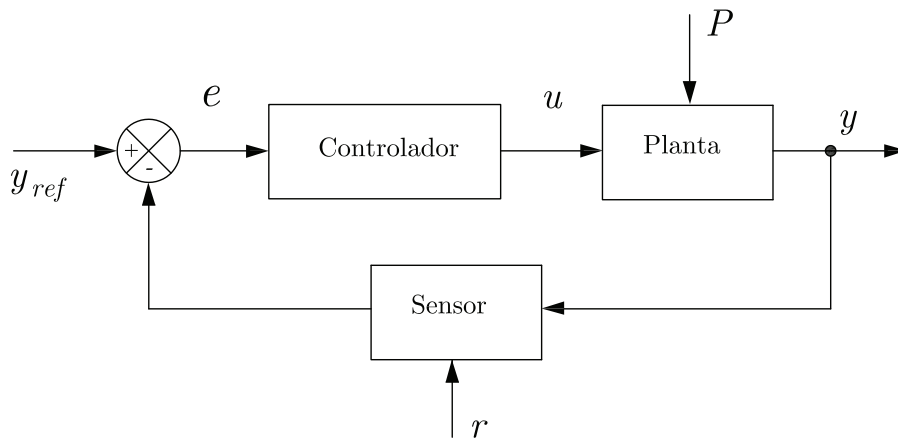


Figura 1.1: Esquema simple de un control clásico, donde  $y_{ref}$ ,  $e$ ,  $u$ ,  $y$ ,  $P$  y  $r$ , son: la señal de referencia del sistema, el error entre la salida y la referencia, la señal de control, la salida del sistema, las perturbaciones externas al sistema y el ruido en la medición, respectivamente.

Uno de los controladores más empleados en la industria moderna y en la antigua es el control Proporcional, Integral y Derivativo ( PID ), debido a la facilidad de con la que realiza e implanta, ya que sólo requiere hacer pruebas básicas, además que presenta robustez [82].

Inspirado en la estructura de un PID clásico se propone en este trabajo de tesis una

versión modificada de un PID clásico discreto, llamado PID *wavelet*. Esta estructura hace uso de la teoría *wavelet* para tener un mejor comportamiento del sistema en lazo cerrado.

En la estructura del PID *wavelet* propuesto se hace una descomposición de la *señal de error*, el cual permite tener diferentes detalles de dicha señal que se está tratando [43, 69, 75].

El diseño y la simulación de los algoritmos de los controladores PID propuestos se desarrolla bajo el entorno de programación de MATLAB y Simulink.

La validación del PID *wavelet* propuesto se hace mediante diferentes aplicaciones: control de posición de un motor de CD, un sistema térmico y un robot planar, mientras que el PID *wavenet* sólo se aplica en simulación numérica para el control de posición de un motor de CD.

Con base a este panorama general en la siguiente sección se enuncia el problema a trabajar.

## 1.1. Descripción de la problemática

Entonces, el problema que se plantea es cómo estabilizar un sistema cuando no se conoce su modelo matemático.

## 1.2. Descripción de la propuesta de solución

La solución que se propone al problema planteado es desarrollar algoritmos de dos controladores del tipo multiresolución, uno llamado PID *wavelet*, el cual hace uso la técnica multiresolución basada en la teoría *wavelet* para la descomposición de la señal de error, para obtener varias señales a diferentes escalas-frecuencias las cuales son escaladas y sumadas para generar una señal de control que compensa las incertidumbres del sistema. El otro llamado PID *wavenet* que se basa en el aprovechamiento que tienen las *wavelets* en conjunción con las redes neuronales y los algoritmos de adaptación LMS todos esto para obtener un controlador PID multiresolución adaptable siendo su principal característica, el emplear una red neuronal *wavelet* con funciones *wavelet* madre Morlet como funciones de activación, aplicado a un sistema no lineal SISO en donde el modelo matemático se desconoce.

## 1.3. Objetivos de la tesis

Los objetivos de este trabajo de investigación son los siguientes:

## Objetivo general

Diseñar e implementar algoritmos de dos controladores PID's multiresolución, uno que hace uso de la teoría *wavelet* y el otro que además hace uso de la teoría de redes neuronales *wavelet*.

## Objetivos específicos

Los objetivos específicos de la tesis son los siguientes:

- Analizar los algoritmos clásicos de los controladores PID, tanto en caso continuo como discreto.
- Diseñar un algoritmo de un controlador PID *wavelet*.
- Diseñar un algoritmo de un controlador PID multiresolución adaptable empleando redes neuronales *wavelet*.
- Implantar algoritmos de un controlador PID multiresolución adaptable empleando redes neuronales *wavelet*.

## 1.4. Hipótesis

Haciendo uso de controladores del tipo multiresolución es posible estabilizar sistemas de posición sin requerir uso del modelo matemático para el diseño.

## 1.5. Justificación

Un método de autosintonización es una consideración importante en el diseño de un controlador adaptable de un sistema desconocido con variaciones lentas. La idea básica en control adaptable es estimar los parámetros de la planta que varían y ajustar los parámetros del controlador en línea, basados en las señales del sistema usando los parámetros estimados en el cálculo de entrada de control. Sin embargo, las técnicas tradicionales de control adaptable pueden sólo tratar con sistemas lineales o en su defecto casos especiales de sistemas no lineales. Típicamente, esas técnicas asumen que el modelo del sistema está operando en una región lineal. Los parámetros de un modelo de la planta linealizado son estimados de manera recursiva y son usados para actualizar el controlador. Comúnmente no es posible representar adecuadamente las características del sistema, tales como no linealidades y en general la complejidad del sistema. Dada la importancia del control adaptable, es necesario desarrollar una técnica en la cual la estructura del modelo de un sistema puede ser identificado por un proceso adaptable.

## 1.6. Aportaciones

Las aportaciones de este trabajo de investigación son las siguientes:

- Puesta en operación de un sistema de control posición de un grado de libertad.
- Puesta en operación de un robot planar.
- Implantación de un controlador PID *wavelet* en las plataformas anteriores.
- Aplicación de un controlador PID *wavenet* a un sistema de control de posición.
- Implantación de un filtro digital *wavelet* en un dispositivo háptico PHANToM.
- Publicación de los resultados en dos congresos nacionales y un congreso internacional.

## 1.7. Organización de la tesis

En el Capítulo 2, *Estudio del estado del arte*, se analizan los trabajos relacionados con el desarrollo de esta investigación. En el Capítulo 3, *Desarrollo de un control PID wavelet*, se desarrolla un controlador PID *wavelet* y se presentan los resultados de simulación sobre algunos sistemas, tales como: un motor de CD, un sistema térmico y un robot planar. El Capítulo 4, *Desarrollo de un control PID wavenet*, se presenta las características del controlador PID *wavelet* propuesto y se presentan los resultados en simulación sobre un sistema de control de posición de un motor de CD, mientras que en el Capítulo 5, *Resultados de laboratorio*, Se muestran los resultados experimentales y las plataformas experimentales, donde son obtenidos los resultados, demostrándose la mejora en el comportamiento del sistema global. Finalmente, el Capítulo 6, *Conclusiones*, se discuten los principales resultados del trabajo de investigación, también se presentan los trabajos futuros relacionados con este estudio.

# Capítulo 2

## Estudio del estado del arte

El objetivo de este capítulo es presentar los principales resultados que están estrechamente relacionados con este trabajo de tesis. La organización del capítulo es la siguiente: en la Sección 2.1 se presenta una introducción, haciendo énfasis en las diferentes áreas de trabajo que interactúan en el desarrollo de la tesis: teoría *wavelet*, redes neuronales, teoría *wavenet* y control de sistemas. Mientras que en las Secciones 2.2, 2.3, 2.4 y 2.5 se describen los trabajos de cada una de estas áreas. Finalmente, en la Sección 2.6 se presentan los comentarios.

### 2.1. Introducción

Una de las etapas fundamentales en el trabajo de tesis es el conocer las investigaciones que se han venido realizando en los últimos años en las áreas de:

- Teoría *wavelet*.
- Redes neuronales.
- Teoría *wavenet*.
- Control de sistemas.

Con la finalidad de tener un conocimiento bien fundamentado y así poder desarrollar, aplicar y/o aportar algo sobre las áreas en cuestión, en la Figura 2.1 se muestran estas áreas. A continuación se mencionan los trabajos que se consideran más importantes para el desarrollo de la tesis.

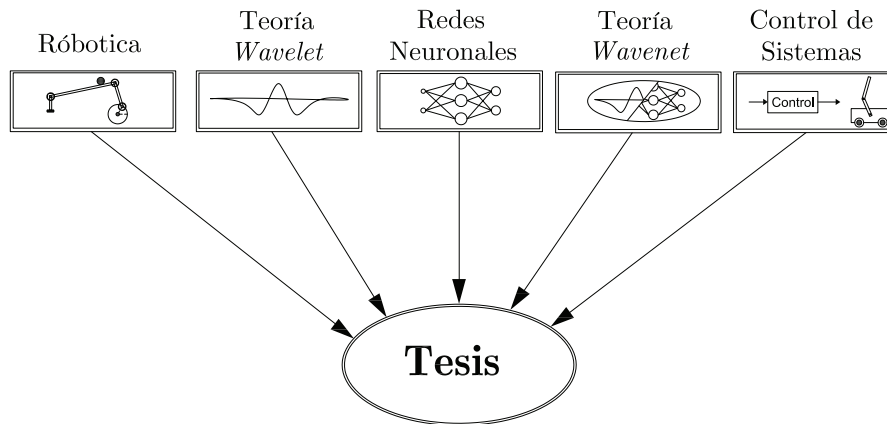


Figura 2.1: Áreas requeridas para el desarrollo de la tesis.

## 2.2. Teoría *wavelet*

El desarrollo de las *wavelets* empieza a partir del trabajo de Alfred Haar, de quien se conoce la primera *wavelet* [27] la cual lleva su nombre y fue propuesta a principios del siglo XX. Contribuyeron de modo notable al avance de la teoría Goupillaud, Grossmann y Morlet [25, 26] con su formulación de lo que hoy se conoce como transformada *wavelet* continua TWC Jan Olov-Strömberg con su temprano trabajo en 1983 sobre *wavelets* discretas, Ingrid Daubechies [13], con su propuesta de *wavelets* ortogonales con soporte compacto publicado en 1988, Stephane Mallat y Yves Meyer [60], con su marco multirresolución publicado en 1989 y muchos otros trabajos desde entonces.

En la actualidad existen una gran cantidad de artículos, libros y software que hacen de este tema toda una área de investigación. Se tiene por ejemplo los artículos de Daubechies [11, 12, 13] que son un semillero de generación de conocimiento en el área, en este mismo tenor se tienen los artículos de Meyer [59, 60, 61]. En cuanto a libros se refiere se han venido publicando alguno de ellos con el objetivo de ser usado en función del área de aplicación, se tiene por ejemplo los libros [20, 38, 86] que están enfocados a dar las bases matemáticas sobre el tema, mientras que [29, 31, 49] presentan la teoría y sus diferentes aplicaciones en áreas de la ingeniería. Es importante mencionar que no se encontró un libro que trate la teoría *wavelet* aplicada al control de sistemas dinámicos. Sin embargo, sí se tienen una gran cantidad de publicaciones tanto en congresos internacionales como revistas sobre este gran tema de interés para el autor de la tesis. Por mencionar algunos, por ejemplo en [14, 45, 91] y [18] se presentan aplicaciones de la teoría *wavelet* al procesamiento de señales médicas y localización de características faciales, respectivamente.

Por otro lado en control existen diferentes trabajos previos, que utilizan trans-

formadas *wavelet* para: análisis multiresolución, aproximación de funciones, identificación de parámetros, sintonización de PID, clasificación de perturbaciones, por mencionar algunos. Más adelante haremos referencia de algunos de esos artículos que utilizan transformadas *wavelets*.

En cuanto a software se refiere se ha de mencionar, que existe una *toolbox* de MATLAB [54] para la teoría *wavelet*, el cual cuenta con algoritmos numéricos programados en forma de comandos, que pueden ser usados en procesamiento de señales e imágenes.

### 2.2.1. Trabajos previos sobre *wavelet*

Un novedoso controlador PID basado sobre análisis multiresolución basado en *wavelet* fue presentado en [69]. El controlador es similar a un controlador PID en su principio y aplicación. Dicho controlador es simulado para controlar la velocidad de un servo sistema, que manifiesta resonancia de baja frecuencia, que son causados por los componentes de transmisión y el mal acoplamiento del motor con la carga, el objetivo es llevar al motor a una velocidad de referencia dada. Se prueba de forma física el controlador en dicho sistema para controlar la posición, donde el objetivo es rotar la carga una revolución en un segundo sin sobre paso. De los resultados obtenidos en ambos casos, se resalta el hecho de que la señal de control es mucho más suave y sin cambios bruscos que utilizando un PID clásico. Aunque en ambos casos se lograron los objetivos cabe mencionar que se probó en sistemas lineales y con sus modelos matemáticos bien conocidos.

En [37] hacen uso de la transformada Haar para un método computacional que determina las constantes de un control con retroalimentación del estado de un sistema con retardo y variante en el tiempo, mientras que en [67] utilizan un Algoritmo de Colocación Sucesivo *wavelet* para resolver ecuaciones Hamilton-Jacobi-Bellman Generalizadas que aparecen en problemas de control. Por otro lado en [23] se propuso un algoritmo genético para obtener un control en la reorientación de una nave espacial subactuada con dos volantes, que proveen dos señales de control de entrada, produciendo momentos sobre los ejes  $xyz$ , así permite cualquier orientación de la nave espacial, el cual utiliza la transformada *wavelet* discreta TWD para aproximar la ley de control. En los tres artículos anteriores hacen uso de la transformada *wavelet* para el diseño de un controlador sub-óptimo y para la solución de ecuaciones que se presentan en problemas de control sub-óptimo.

En el control de robots se sabe que ciertos tipos de perturbaciones ocurren, en [32] presenta un clasificador y un estimador difuso de las perturbaciones que actualmente afectan el proceso. El clasificador utiliza TWD para extraer las características de la señal medida, basado en estas características, un sistema difuso da una estimación de la proporción de diferentes perturbaciones que se presentan en la señal. La salida del

clasificador es usado para seleccionar un control para una perturbación en particular, esto a su vez es usado para controlar el proceso durante la perturbación.

En el trabajo de [56], muestran y concluyen, la ventaja de usar la transformada *wavelet* en lugar de la transformada de Fourier en el análisis de procesos químicos, debido que en la transformada *wavelet* puede visualizarse el comportamiento dinámico del proceso en el dominio de tiempo-frecuencia, la relación interna entre los estados del proceso pueden ser clasificados y la señal de propagación puede ser identificada.

La transformada *wavelet* también es usada en un método adaptable en la simulación de circuitos no lineales [90], al igual que para desarrollar un algoritmo para estimación de la escala y el tiempo transcurrido entre dos señales recibidas y subsecuentemente para extraer el tiempo de retardo y la compresión doppler, específicamente en tiempo de retardo para movimiento lineal, en procesamiento de señales de radar y sonar [64].

### 2.3. Redes neuronales artificiales

El interés de científicos e ingenieros durante muchos años por lograr máquinas inteligentes (que puedan emular el conocimiento humano), han llevado a desarrollar lo que se le conoce como Redes Neuronales Artificiales RNA un estudio sistemático fué publicado por primera vez por McCulloch y Pitts [58] y cuatro años después los mismos autores exploraron los paradigmas de la red para un reconocimiento de patrones usando un percentrón de una simple capa [57].

Una red neuronal se define como un procesador distribuido en paralelo, que puede filtrar y procesar la información y tener una decisión [83]. Una red neuronal se parece al cerebro en dos ideas básicas:

- La red necesita una cantidad de información para entrenarse.
- Las conexiones entre las neuronas se usan para almacenar la información.

Desde entonces y hasta la fecha se han publicado: libros y artículos, por mencionar algunos por ejemplo: en libros [17, 28, 30, 92] y en artículos [1, 52, 73] y [3, 34, 35] que combina sistemas difusos con redes neuronales. Una de las herramientas de programación para facilitar el desarrollo de redes neuronales es la *toolbox* incluida en MATLAB [53]. La teoría básica sobre RNA se presenta en el Apéndice C, que da las bases necesarias al autor para el desarrollo de este trabajo de tesis.

Debido a la estructura de las redes neuronales, éstas permiten el uso de la transformada *wavelet*, conocida como red neuronal *wavelet* o *wavenet*.



## 2.4. Teoría *wavenet*

Una red neuronal posee: una capa de entrada, una o varias capas ocultas y una capa de salida, cada capa posee varias neuronas las cuales tienen una función de activación: función sigmoïdal o función tangente hiperbólica, que son las más comúnmente empleadas [87]. Pero existen trabajos de investigación donde dicha función de activación es sustituida por una función *wavelet*, por lo que es conocida como *wavenet*. A continuación se presentan los artículos más relevantes y los cuales inspiran este trabajo de tesis.

### 2.4.1. Trabajos previos sobre *wavenet*

Como se mencionó anteriormente, una *wavenet* no es más que la estructura de una red neuronal radial con funciones *wavelet* como funciones de activación, un ejemplo es [46] donde utilizan funciones *wavelets* Gaussianas como funciones de activación para diseñar un controlador adaptable para robots.

Una de las aplicaciones donde emplean las *wavenets*, es en [85], donde un control PID es auto-sintonizable empleando una *wavenet*. Además, pueden utilizarse para identificación de parámetros de sistemas dinámicos no lineales, como [43], donde se utilizan dos estructuras *wavenet*: una para identificación y otra para sintonizar un PID. Puesto que las *wavenet* combinan las ventajas ofrecidas en el aprendizaje de una red neuronal y la representación de una *wavelet*, ofrecen una aproximación eficiente en sistemas de control dinámicos que usualmente poseen complejidad, no linealidades e incertidumbres.

También, las *wavenets* pueden usar como control inteligente, para tener más robustez en un sistema donde utiliza un vector de control en máquinas de inducción [62], y para aprendizaje en línea y la cancelación de errores repetitivos en lectura de unidades de discos [4].

Algunos trabajos como [5, 6, 71, 88], hacen uso de *wavenet* para el control adaptable en sistemas dinámicos no lineales. Los resultados basados en simulaciones en modelos perturbados como el péndulo invertido y otros modelos de sistemas no lineales, demuestran la eficacia de este tipo de control adaptable *wavenet*.

Un controlador PID adaptable presentado en [75], mediante una red neuronal adaptable RASPl para controlar un sistema de conversión de energía eólica. Para el proceso de identificación utiliza una estructura de una red neuronal de una simple capa con alimentación hacia adelante con transformadas *wavelet* incrustadas en las neuronas de la capa oculta, en cascada con un bloque de un filtro IIR (Filtro de Respuesta al Impulso Infinito).

En algunos trabajos como [2, 44, 48], usan *wavenet* para la identificación y aproximación de sistemas no lineales, mientras que [22] las usan para aproximación y apren-

dizaje de series caóticas, simulando en dos de las más famosas series caóticas: ecuaciones de Lorenz y Mackey-Glass.

Algunas otras áreas en donde se emplean *wavenets* son: en un método de diagnóstico aplicado en la detección del golpeteo del motor en la industria automotriz [84], mientras que en [33] es empleado para control longitudinal de un pelotón vehicular vía *wavenet* para un sistema de prevención de coalición de carros y modelado de señales en el campo de procesamiento de señales electrofisiológicas y análisis tiempo-frecuencia en [91].

Existen algunos trabajos tales como: [76], donde presentan un trabajo de investigación para comparar redes neuronales (en particular las de funciones de base radial) y *wavenet*. Los resultados experimentales demuestran que la *wavenet* son un caso específico de la red neuronal de base radial, así que las redes neuronales de base radial en su forma más general puede ser específicamente aplicada también para aproximaciones de funciones pero depende de que tan bien se elija la función de activación. También en [80], donde hacen una comparación entre funciones *wavelet* y series de Fourier para una red neuronal en un clasificador de cromosomas, dando como resultados en simulación que, al usar *wavenet* converge mucho más rápido al error RMS de 0.00001 y el clasificador *wavenet* usa solamente 16 coeficientes, mientras el clasificador de Fourier requiere 32 coeficientes y converge en el doble de épocas al mismo valor de error ( RMS ). Dejando claro que es mejor usar funciones *wavelet* como funciones de activación que series de Fourier.

### 2.4.2. Trabajos previos realizados por alumnos del CITIS

Existe una tesis previa a éste trabajo de investigación dado en [15]. Aquí básicamente lo que se hace es emplear las *wavenet* para la aproximación de funciones desconocidas, obteniéndose resultados en simulación numérica. En [74] se presenta un esquema de aprendizaje reforzado derivado del método *Q-Learning*, el cual emplea RNA, cuyas funciones de activación son funciones *wavelets* y los algoritmos desarrollados son validados en la aplicación de control de un sistema subactuado. También, es importante mencionar que la plataforma sobre la que se trabaja es un *ball and bean* que fue diseñado y construido por estudiantes de la carrera de Ingeniería en Electrónica y Telecomunicaciones [10] que se ofrece en la UAEH .

## 2.5. Control clásico

Uno de los controladores más empleados en la industria moderna es el PID (Proporcional Integral-Derivativo) [82]. El controlador PID en adición a su eficacia como un medio de control, que lo ha mantenido como el más empleado desde que apareció

por los años 40's [82], es por el hecho de ser muy práctico de implementar en circuitos analógicos y ser muy intuitivo, el cual lo hace fácil de sintonizar.

En general, un controlador PID toma como entrada al error,  $e$  y actuar sobre él para generar una señal de control como salida  $u$ , como se muestra

$$u = k_P e + k_I \int_0^t e dt + k_D \frac{de}{dt}, \quad (2.1)$$

donde  $k_P$ ,  $k_I$  y  $k_D$  son las ganancias del PID a ser sintonizadas, los tres términos en este controlador representan intuitivamente el actual, pasado y la tendencia del error. En una idea general un controlador PID trabaja de la siguiente forma: la parte proporcional  $k_P e$  provee la fuerza necesaria para llevar la señal del error a cero, aunque sí la función de transferencia en lazo cerrado del sistema no posee un integrador  $1/s$  existe un error en estado estacionario  $e_{ss}$ , o desplazamiento (offset), es aquí donde la parte integral,  $k_I \int e dt$  actúa eliminando o reduciendo el error en estado estacionario, esto lo logra, haciendo que la señal de control  $u$  sea diferente de cero cuando el error  $e$  es cero y la parte derivativa  $k_D \frac{de}{dt}$  aporta una alta sensibilidad, debido a que responde a la velocidad del cambio del error y produce una corrección significativa antes que la magnitud del error se vuelva demasiado grande, es decir, prevé el error, inicia una acción correctiva oportuna y tiende a aumentar la estabilidad del sistema [65, 66].

La forma de un PID discreto es [39]:

$$u(k) = u(k-1) + k_P [e(k) - e(k-1)] + k_I e(k) + k_D [e(k) - 2e(k-1) + e(k-2)], \quad (2.2)$$

cuya función de transferencia esta dada por

$$\frac{u(z)}{e(z)} = k_P + k_I \frac{T}{2} \frac{z+1}{z-1} + k_D \frac{1}{T} \frac{z-1}{z}, \quad (2.3)$$

y su funcionamiento es de la misma forma que el PID continuo.

En la actualidad existen una gran variedad de variantes del PID clásico [16], así como también diferentes técnicas de sintonización [66]. Básicamente se puede clasificar las variantes del PID clásico como:

- PID no lineal.
- PID robusto.
- PID genético.
- PID difuso.
- PID neuro-difuso.

- PID *wavelet*.
- PID *wavenet*.

Es en estos dos últimos donde se centra el trabajo de investigación de esta tesis.

## 2.6. Comentarios

Es importante comentar que existen un gran número de trabajos de investigación donde se involucran redes neuronales en combinación con la transformada *wavelet*, y sólo se mencionan aquí las que se consideran más importantes para desarrollar el PID *wavenet*. Para el mejor entendimiento de la combinación entre redes neuronales y la transformada *wavelet*, en el Apéndice D se dan las herramientas matemáticas necesarias, mientras que en el Capítulo 4 se presenta el controlador PID *wavenet*, el cual está inspirado en [75].

# Capítulo 3

## Desarrollo de un control PID *wavelet*

El objetivo de este capítulo es presentar el desarrollo de un control PID *wavelet* para sistemas SISO, el cual es simulado e implementado en diferentes sistemas como: un motor de CD, un sistema térmico y un robot planar, para definir las ventajas y desventajas de este controlador en ambos sistemas. Para poder lograr el objetivo, se ha organizado este capítulo de la siguiente forma: se da una breve introducción en la Sección 3.1, en la Sección 3.2 se da la estructura del control PID *wavelet*, llamado PID multiresolución PIDMR . El análisis multiresolución es un tema muy importante para el desarrollo del PID *wavelet*, siendo empleado para la descomposición de la señal del error, por tal motivo en la Sección 3.3 es visto, los algoritmos para el desarrollo del PID *wavelet* para la simulación e implementación son dados en la Sección 3.4, los resultados de las simulaciones son dados en la Sección 3.5 y las conclusiones basadas en los resultados se dan en la Sección 3.6.

### 3.1. Introducción

El controlador *wavelet* es similar a un control PID en principio y aplicación, como se muestra en la Figura 3.1. La salida  $y$  de un sistema de control representa los efectos acumulativos de incertidumbre tales como: el ruido en el estado, variación en la fricción y perturbaciones externas, las cuales se manifiestan a diferentes escalas. El control *wavelet* utiliza el análisis multiresolución para descomponer la señal de error  $e$  a diferentes escalas, las señales resultantes son entonces escaladas por sus respectivas ganancias  $K_i$  y adicionadas juntas para generar la señal de control  $u$  para compensar

las incertidumbres del sistema, es decir

$$u = \sum_{i=1}^{N+1} K_i e_i, \quad (3.1)$$

donde  $e_i$  es la señal del error a su  $i$ -ésima escala,  $K_i$  es su respectiva ganancia y  $N$  es el nivel de descomposición. Para el caso del PID clásico discreto  $N = 2$ .

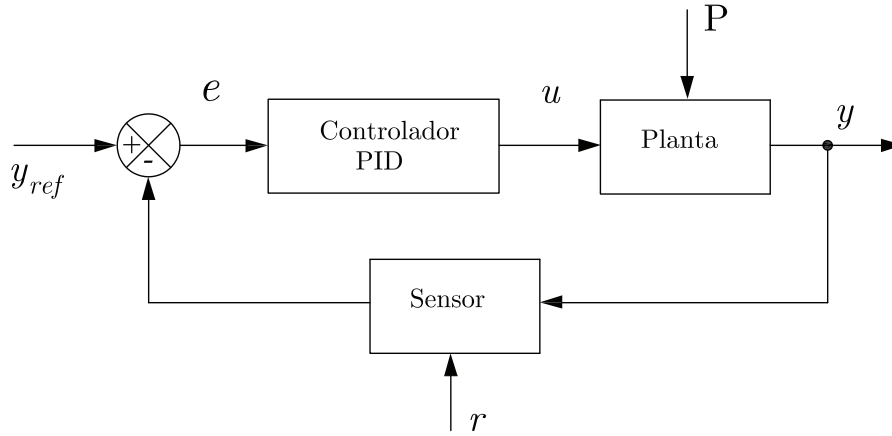


Figura 3.1: Esquema simple de un control PID clásico, donde  $y_{ref}$ ,  $e$ ,  $u$ ,  $y$ ,  $P$  y  $r$ , son la referencia del sistema, el error entre la salida y la referencia, la señal de control, la salida del sistema, las perturbaciones externas al sistema y el ruido en la medición, respectivamente.

## 3.2. Control PID multiresolución

Un control PID clásico, en esencia toma la señal del error  $e$  y entonces actúa sobre ésta para generar una señal de control, es decir

$$u(t) = k_P e(t) + k_I \int_0^t e(t) dt + k_D \frac{de(t)}{dt} \quad (3.2)$$

donde  $k_P$ ,  $k_I$  y  $k_D$  son las ganancias proporcional, integral y derivativa, respectivamente. En términos de información de frecuencia los términos integral y derivativo, tienden a capturar la información de la baja frecuencia y la información de la alta frecuencia de la señal del error  $e$ , respectivamente. En el dominio del tiempo discreto  $t = kT$ , la ley de control de un PID es expresada como [65]

$$u(k) = k_P e(k) + k_I \sum_{i=0}^k e(i) + k_D [e(k) - e(k-1)], \quad (3.3)$$

donde  $k_P$ ,  $k_I$  y  $k_D$  son los coeficientes proporcional, integral y derivativo, respectivamente y donde por facilidad de cálculos se toma  $T = 1$ . La expresión integral de la ley de control del PID se escribe como:

$$\begin{aligned}\Delta u(k) &= u(k) - u(k-1), \\ \Delta u(k) &= k_P e(k) + k_I \sum_{i=0}^k e(i) + k_D [e(k) - e(k-1)] \\ &\quad - k_P e(k-1) - k_I \sum_{i=0}^{k-1} e(i) - k_D [e(k-1) - e(k-2)], \\ \Delta u(k) &= k_P [e(k) - e(k-1)] + k_I e(k) + k_D [e(k) - 2e(k-1) + e(k-2)],\end{aligned}\tag{3.4}$$

o equivalente

$$u(k) = u(k-1) + k_P [e(k) - e(k-1)] + k_I e(k) + k_D [e(k) - 2e(k-1) + e(k-2)].\tag{3.5}$$

Tomando los parámetros del PID,  $k_P$ ,  $k_I$  y  $k_D$  como las variables de ajuste, entonces (3.5) puede ser descrita como

$$\begin{aligned}u(k) &= u(k-1) + (k_P + k_I + k_D)e(k) + (-k_P - 2k_D)e(k-1) + k_D e(k-2), \\ u(k) &= u(k-1) + \sum_{i=0}^2 k_i e(k-i),\end{aligned}\tag{3.6}$$

ó equivalente

$$\Delta u(k) = \sum_{i=0}^2 k_i e(k-i),\tag{3.7}$$

donde  $k_0 = k_P + k_I + k_D$ ,  $k_1 = -k_P - 2k_D$  y  $k_2 = k_D$ .

De la ecuación (3.7), se puede observar que la ley de control de un PID clásico es una descomposición lineal del error, sólo que esta descomposición es fija, es decir, siempre tiene 3 términos, ésto hace la diferencia con la descomposición multiresolución, donde aquí el número de descomposiciones puede ser infinito y aun más que cada una de ellas está a diferentes escalas de tiempo-frecuencia.

En una manera similar, un control PIDMR descompone la señal de error  $e$  para obtener frecuencias altas, bajas e intermedias, haciendo uso del análisis multiresolución para la descomposición, como se muestra en la Sección 3.3. Así cada uno de estos

componentes son escalados con sus respectivas ganancias y adicionados todos juntos para generar la señal de control  $u$ , como sigue:

$$u = K_H e_H + K_{M_1} e_{M_1} + \cdots + k_i e_i + \cdots + K_{M_{N-1}} e_{M_{N-1}} + K_L e_L, \quad (3.8)$$

$$u = [K_H \ K_{M_1} \ \cdots \ K_i \ \cdots \ K_{M_{N-1}} \ K_L] \begin{bmatrix} e_H \\ e_{M_1} \\ \vdots \\ e_i \\ \vdots \\ e_{M_{N-1}} \\ e_L \end{bmatrix}, \quad (3.9)$$

$$u = \mathbf{K} \mathbf{E}_m, \quad (3.10)$$

donde

$$\mathbf{K} = [K_H \ K_{M_1} \ \cdots \ K_i \ \cdots \ K_{N-1} \ K_L], \quad (3.11)$$

$$\mathbf{E}_m = [e_H \ e_{M_1} \ \cdots \ e_i \ \cdots \ e_{N-1} \ e_L]^T, \quad (3.12)$$

donde  $N$  es el nivel del control PID multiresolución *wavelet*.

Mientras que un control PID clásico tiene tres parámetros para ser sintonizado  $k_P$ ,  $k_I$  y  $k_D$ , el control PIDMR posee dos ó más parámetros y este número de parámetros depende del nivel de descomposición que se aplica a la señal del error  $e$ . El diagrama esquemático de una planta usando un control PIDMR es mostrado en la Figura 3.2.

Como puede verse en el Apéndice B, existe un gran número de *wavelets* diferentes, la selección de la *wavelet* afecta el funcionamiento del controlador. Por eso, existen características que se deben tomar en cuenta, tales como:

- El tipo de sistema de representación (continuo o discreto).
- Las propiedades de la *wavelet* a ser empleada.
- Las dinámicas del sistema.

Para mayores detalles sobre la selección de la *wavelet*, ver [68]. En la construcción del PIDMR en este capítulo se emplea la transformada *wavelet* Daubechies de orden 2, basado en los resultados de [69], al encontrarse muy conveniente para control automático de un sistema de control de movimiento.

Todos los sistemas físicos están sujetos a algún tipo de señales externas o ruidos durante la operación. Por lo tanto, al diseñar un sistema de control, se debe considerar



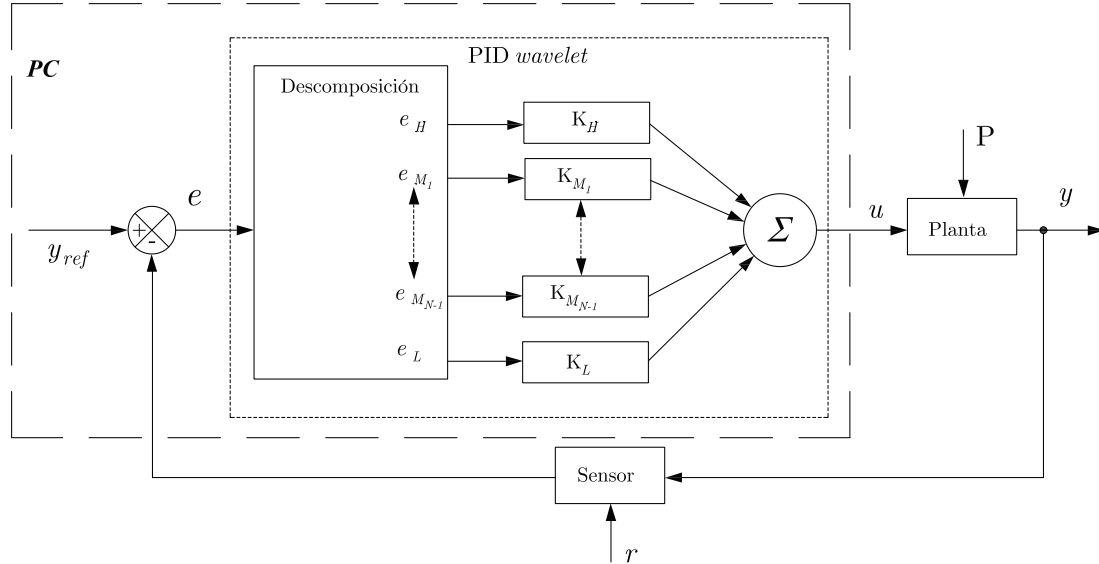


Figura 3.2: Diagrama simple del control PIDMR, donde  $y_{ref}$ ,  $e$ ,  $u$  y  $y$ , son la señal de referencia del sistema, señal de error entre la salida y la referencia, la señal de control y la señal de salida del sistema, respectivamente. Aquí  $K_H, K_{M_1}, \dots, K_{M_{N-1}}$  y  $K_L$  son las ganancias a sintonizar del control PIDMR [69].

si el sistema proporcionará mayor sensibilidad al ruido o a las perturbaciones. En la práctica, las perturbaciones y comandos son algunas veces señales de baja frecuencia y el ruido es de alta frecuencia, con un control PIDMR se puede manipular las señales, sintonizando las ganancias del PIDMR casi de una manera directa. Por ejemplo: ajustando la ganancia de la escala baja a zero  $K_L = 0$ , se logra producir una señal de control que reduzca los efectos del ruido en la salida de la planta  $y$ , y por lo tanto la señal suave de control produce un esfuerzo mínimo mejorando la vida de los actuadores y todo el comportamiento de la planta [39, 66].

### 3.3. Análisis multiresolución

El análisis multiresolución es una estructura conveniente para la representación jerárquica de funciones o señales en diferentes escalas. La idea básica del análisis multiresolución es representar una función  $e(x)$  como un límite de aproximaciones sucesivas, cada una de estas aproximaciones sucesivas  $e_i$  es una versión suave de la función original con más de los finos detalles, es decir

$$e(x) = \sum_{i=1}^{N+1} e_i(x). \quad (3.13)$$

En el Apéndice B se dan las herramientas matemáticas para el mejor entendimiento del tema de análisis multiresolución. Las *wavelets* son usadas para generar las funciones base las cuales son empleadas para representar una señal usando un conjunto de coeficientes.

Sea una señal  $e(x)$  a descomponer, siendo ésta la señal de error, la cual se considera que cumple con

$$\int_{-\infty}^{\infty} \|e(x)\|^2 dx < \infty, \quad (3.14)$$

por lo tanto, de la Definición 38 (ver Apéndice B) implica que  $e(x) \in L^2(\mathbb{R})$ . Si  $e(x)$  es la señal original a una resolución 0 en el subespacio  $V_0$ , donde el subespacio  $V_0$  cumple con (B.28) y (B.29), por lo que  $V_0 \in L^2(\mathbb{R})$ . De (B.25) se tiene que las funciones base *wavelet* que generan al subespacio  $V_0$  son:

$$\phi_{0,k}(x) = \phi(x - k); \quad k \in \mathbb{Z}, \quad (3.15)$$

donde  $\phi \in L^2(\mathbb{R})$  es llamada *función de escalamiento* (B.26).

De (B.27) implica que existe un subespacio cerrado  $V_1$  dentro del subespacio  $V_0$  y el cual es generado por:

$$\phi_{1,k}(x) = 2^{-\frac{1}{2}}\phi(2^{-1}x - k), \quad (3.16)$$

sea  $c_{1,k}$  los coeficientes llamados *coeficientes de aproximación*, que contiene información menos fina necesaria para la reconstrucción a una resolución 1 en el espacio  $V_1$  de la señal original  $e(x)$ , por la Definición 42 (ver Apéndice B), los coeficientes se calculan como

$$c_{1,k} = \langle e(x), \phi_{1,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\phi_{1,k}(x)}, \quad (3.17)$$

para la reconstrucción de la señal de aproximación a una resolución 1, definida como  $e_H^{(1)}(x)$ , es obtenida por medio de (B.24)

$$e_H^{(1)}(x) = \sum_{k=-\infty}^{\infty} c_{1,k} \phi_{1,k}(x). \quad (3.18)$$

La información más detallada de la función original está contenida dentro del subespacio complementario de  $V_1$ , definido como  $W_1$  el cual cumple con (B.30) y (B.31), las funciones *wavelet* base para generar el subespacio  $W_1$  son

$$\psi_{1,k}(x) = 2^{-\frac{1}{2}}\psi(2^{-1}x - k), \quad (3.19)$$

donde  $\psi \in L^2(\mathbb{R})$  es llamada *función wavelet* (B.35).

Sea  $d_{1,k}$  los coeficientes que contienen la información más detallada, llamados *coeficientes de detalle*, obtenidos por

$$d_{1,k} = \langle e(x), \psi_{1,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\psi_{1,k}(x)}, \quad (3.20)$$

para la reconstrucción de la señal de detalles a una resolución 1, definida como  $e_L^{(1)}(x)$ , es obtenida por

$$e_L^{(1)}(x) = \sum_{k=-\infty}^{\infty} d_{1,k} \psi_{1,k}(x). \quad (3.21)$$

Por lo tanto la señal original es calculada por (B.38), el cual se obtiene

$$e(x) = e_H^{(1)}(x) + e_L^{(1)}(x), \quad (3.22)$$

$$e(x) = \sum_{k=-\infty}^{\infty} c_{1,k} \phi_{1,k}(x) + \sum_{k=-\infty}^{\infty} d_{1,k} \psi_{1,k}(x), \quad (3.23)$$

donde  $e_H^{(1)}(x)$  tiene la característica de ser una señal de alta escala a baja frecuencia, y  $e_L^{(1)}(x)$  es una señal de baja escala a alta frecuencia.

La señal  $e_H^{(1)}(x)$  contiene gran parte de la señal original  $e(x)$  (B.40), por lo que se puede seguir descomponiendo en dos señales de la misma forma que la señal original, de la siguiente manera:

- Sea  $V_2$  un subespacio cerrado en  $V_1$ , con su complemento  $W_2$ , generados por las siguientes funciones base *wavelet*

$$\phi_{2,k}(x) = 2^{-1} \phi(2^{-2}x - k), \quad (3.24)$$

$$\psi_{2,k}(x) = 2^{-1} \psi(2^{-2}x - k), \quad (3.25)$$

con coeficientes

$$c_{2,k} = \langle e(x), \phi_{2,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\phi_{2,k}(x)}, \quad (3.26)$$

$$d_{2,k} = \langle e(x), \psi_{2,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\psi_{2,k}(x)}. \quad (3.27)$$

- Las señales a un nivel de resolución 2,  $e_H^{(2)}$  y  $e_L^{(2)}$  son obtenidas por

$$e_H^{(2)}(x) = \sum_{k=-\infty}^{\infty} c_{2,k} \phi_{2,k}(x), \quad (3.28)$$

$$e_L^{(2)}(x) = \sum_{k=-\infty}^{\infty} d_{2,k} \psi_{2,k}(x). \quad (3.29)$$

- Finalmente, la señal  $e_H^{(1)}$  es

$$e_H^{(1)}(x) = e_H^{(2)}(x) + e_L^{(2)}(x), \quad (3.30)$$

$$e_H^{(1)}(x) = \sum_{k=-\infty}^{\infty} c_{2,k} \phi_{2,k}(x) + \sum_{k=-\infty}^{\infty} d_{2,k} \psi_{2,k}(x), \quad (3.31)$$

sustituyendo (3.22) en (3.30), se tiene

$$e(x) = e_H^{(2)}(x) + e_L^{(2)}(x) + e_L^{(1)}(x), \quad (3.32)$$

considerando (3.23) y (3.31),  $e(x)$  puede ser expresado como

$$e(x) = \sum_{k=-\infty}^{\infty} c_{2,k} \phi_{2,k}(x) + \sum_{k=-\infty}^{\infty} d_{2,k} \psi_{2,k}(x) + \sum_{k=-\infty}^{\infty} d_{1,k} \psi_{1,k}(x), \quad (3.33)$$

$$e(x) = \sum_{k=-\infty}^{\infty} c_{2,k} \phi_{2,k}(x) + \sum_{i=1}^2 \sum_{k=-\infty}^{\infty} d_{i,k} \psi_{i,k}(x), \quad (3.34)$$

donde (3.34) es la representación matemática de la señal del error  $e(x)$  a un nivel de descomposición 2.

Este procedimiento de la descomposición de la señal del error  $e(x)$  puede repetirse hasta el infinito. Por lo que (3.34) para un nivel de descomposición  $N$ , queda de la siguiente forma

$$e(x) = \sum_{k=-\infty}^{\infty} c_{N,k} \phi_{N,k}(x) + \sum_{i=1}^N \sum_{k=-\infty}^{\infty} d_{i,k} \psi_{i,k}(x), \quad (3.35)$$

donde los coeficientes de tendencia y de detalles en cada  $m$ -ésimo nivel de resolución, son calculados por

$$c_{m,k} = \langle e(x), \phi_{m,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\phi_{m,k}(x)}, \quad (3.36)$$

$$d_{m,k} = \langle e(x), \psi_{m,k}(x) \rangle = \sum_{k=-\infty}^{\infty} e(x) \overline{\psi_{m,k}(x)}. \quad (3.37)$$

Entonces, una señal  $e(x) \in L^2(\mathbb{R})$  puede ser descompuesta a un nivel  $N$ , en  $N + 1$  componentes, donde  $e_H^{(N)}$  es la componente de más alta escala y de menor frecuencia definido como  $e_H(x)$ ,  $e_L^{(1)}$  es la componente de más baja escala y alta frecuencia definido como  $e_L(x)$ , y las componentes restantes  $e_L^{(N)}$ ,  $e_L^{(N-1)}$ ,  $\dots$ ,  $e_L^{(3)}$ ,  $e_L^{(2)}$  son las componentes de mediana escala y mediana frecuencia, definidas como  $e_{M_1}$ ,  $e_{M_2}$ ,  $\dots$ ,  $e_{M_{N-2}}$ ,  $e_{M_{N-1}}$ , respectivamente, es decir  $e(x)$  puede ser descompuesta como

$$e(x) = e_H^N(x) + e_L^N(x) + \dots + e_L^2(x) + e_L^1(x), \quad (3.38)$$

$$e(x) = e_H(x) + e_{M_1}(x) + \dots + e_{M_{N-1}}(x) + e_L(x), \quad (3.39)$$

donde  $e_H$  es la señal de escala alta,  $e_L$  es la señal de baja escala y  $e_{M_i}$ ,  $i = 1, \dots, N - 1$ , son las señales de escala media, donde  $N$  es el número de niveles descomposición, como se puede observar de (3.39) para todo los niveles de descomposición siempre se va tener una componente en alta  $e_H$  y otra en baja  $e_L$ , y el resto son señales de escala media  $e_{M_1}$ ,  $e_{M_2}$ ,  $\dots$ ,  $e_{M_{N-1}}$ .

**Ejemplo 1** Con la finalidad de mostrar la descomposición, se presenta el siguiente ejemplo que involucra una señal del consumo eléctrico sobre un periodo de 3 días y que contiene 3920 muestras. Esta señal es tomada desde MATLAB<sup>®</sup> [55], y la cual es mostrada en la Figura 3.3, en la cual se procedió a la descomposición de ésta empleando las líneas de comando mostrados en MATLAB 3.1, obteniendo los resultados mostrados en la Figura 3.4 para diferentes descomposiciones. De esta figura se puede observar que las gráficas de los detalles son señales de alta frecuencia y las aproximaciones son señales de baja frecuencia.

<i>MATLAB 3.1: Comandos para la descomposición</i>
% Primero se carga la señal e(x)

```

>> load leleccum;
>> ex = leleccum(1:3920)
% Se realiza el análisis de la señal e(x) a un nivel 3 de descomposición usando
la wavelet madre Daubechies de orden 1
>> [C,L] = wavedec(ex,3,'db1');
% Reconstrucción de las componentes de la señal e(x)
% Señal eH(1)
>> ex1H = wrcoef('a',C,L,'db1',1);
% Señal eH(2)
>> ex2H = wrcoef('a',C,L,'db1',2);
% Señal eH(3)
>> ex3H = wrcoef('a',C,L,'db1',3);
% Señal eL(1)
>> ex1L = wrcoef('d',C,L,'db1',1);
% Señal eL(2)
>> ex2L = wrcoef('d',C,L,'db1',2);
% Señal eL(3)
>> ex3L = wrcoef('d',C,L,'db1',3);

```

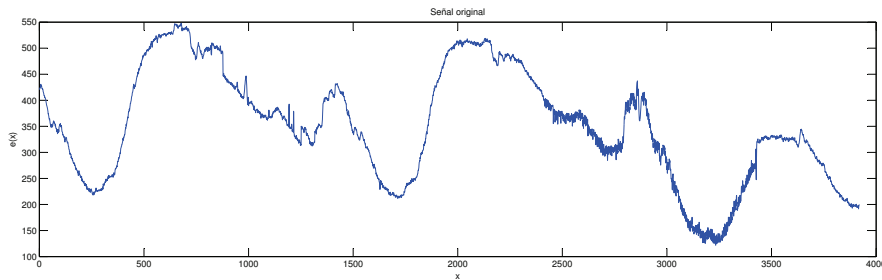


Figura 3.3: Señal de entrada  $e(x)$  original.

Un acercamiento eficiente para calcular la descomposición en *wavelet* es usar el esquema de codificación de sub-banda [86] el cual usa únicamente los filtros  $h(k)$  y  $g(k)$ , los cuales son encontrados por [11] como:

$$\begin{aligned}
 h(k) &= \sqrt{2} \sum_x \phi(x) \overline{\phi(2x - k)}, \\
 g(k) &= \sqrt{2} \sum_x \psi(x) \overline{\psi(2x - k)}, \\
 g(x) &= (-1)^k \overline{h(-k + 1)}.
 \end{aligned} \tag{3.40}$$

Las ecuaciones (3.35), (3.36) y (3.40) proveen un esquema rápido y jerárquico para calcular los coeficientes de una función dada.

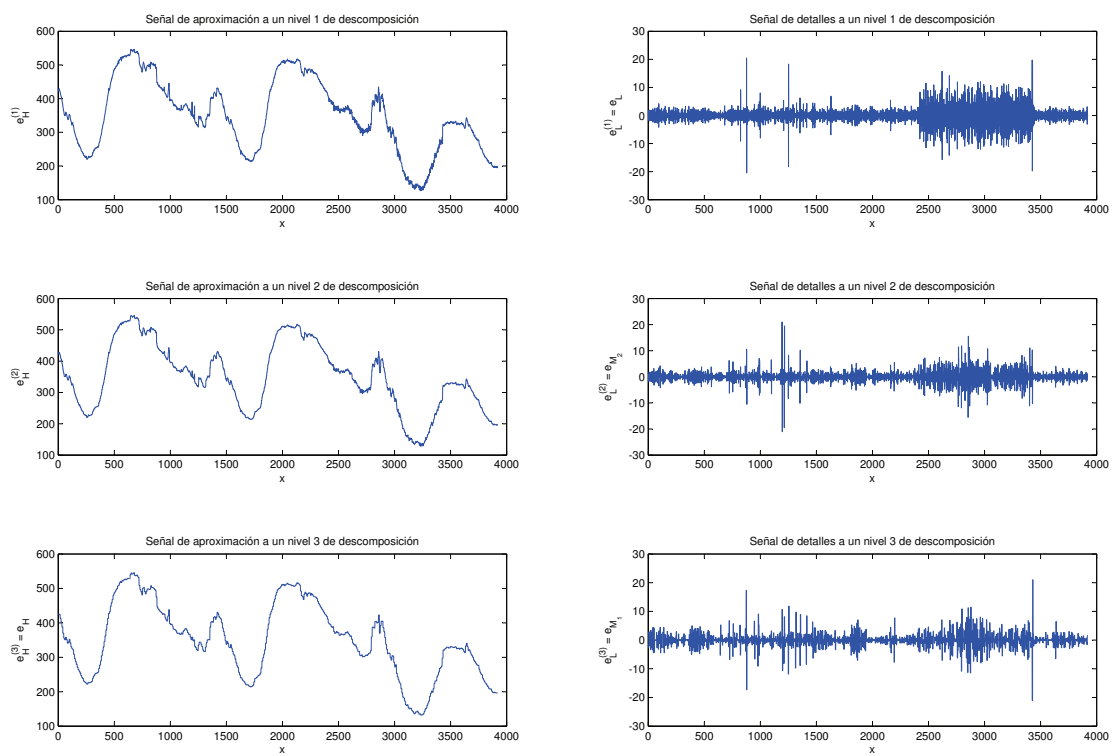


Figura 3.4: Descomposición de la señal  $e(x)$  a diferentes niveles de resolución, donde  $x$  es el número de muestras y  $e_H^{(3)}, e_L^{(3)}, e_L^{(2)}, e_L^{(1)}$  las componentes de la señal original.

### 3.4. Algoritmos del PID *wavelet*

Sea  $e[n]$  la señal muestreada del error  $e(t)$  a descomponer a un nivel  $N$  de resolución, donde  $n = kT_m$ , para  $k = 1, 2, 3, \dots$  y  $T_m$  es el periodo de muestreo. Primero se emplea el proceso del análisis de descomposición dado en el Apéndice B, y para recuperar cada señal de forma independiente para obtener la señal del error de la forma de (3.39), como

$$e[n] = e_H[n] + e_{M_1}[n] + \dots + e_{M_{N-1}}[n] + e_L[n], \quad (3.41)$$

las señales descompuestas en el nivel  $N$  de resolución  $c_{N,k}$  y  $d_{N,k}$ , pertenecientes al espacio de escalonamiento y el espacio *wavelet*, son recuperadas empleando las ecuaciones (B.48) y (B.49), respectivamente. Resultando las señales  $c'_{N,k}$  y  $d'_{N,k}$ , estas señal juntas generar a  $c_{N-1,k}$  por medio de (B.51), debido a que se requiere ambas señales por separado, y debido a que hasta este punto ambas señales pertenecen al nivel de resolución  $N - 1$  en el espacio de escalonamiento, ambas son recuperadas al siguiente nivel ( $N - 2$ ) con (B.48), dando  $c''_{N,k}$  y  $d''_{N,k}$ , que junto con  $d'_{N-1}$  dan la señal  $c_{N-2}$ , donde la señal  $d'_{N-k}$  es obtenida de  $d_{N-1,k}$  por medio de (B.49). Este proceso se sigue hasta obtener todas las señales desde sus respectivos de niveles de resolución hasta el nivel 0, que es el espacio de escalonamiento donde existe la señal del error  $e[n]$ .

La Figura 3.5 muestra el proceso de análisis de descomposición y síntesis de descomposición a un nivel 3, empleada para el control PID *wavelet* en un esquema de codificación sub-banda.

### 3.5. Resultados de simulación e implementación

Con base a las secciones anteriores se diseñó un control PIDMR, utilizando la *wavelet* Daubechies de 2 orden para la descomposición multiresolución de la señal del error  $e$ . Los coeficientes del filtro Daubechies de orden 2 usados en la descomposición multiresolución del control están dados en la Tabla 3.2 y cuya estructura de los filtros  $h$  y  $g$  en el plano  $z$  son dados por (3.42) y (3.43), mientras que el algoritmo del controlador PIDMR está dado en el Algoritmo 3.3.

$$h(z) = 0.4839 + 0.8365z^{-1} + 0.2241z^{-2} - 0.1294z^{-3}, \quad (3.42)$$

$$g(z) = -0.1294 + 0.2241z^{-1} + 0.8365z^{-2} + 0.4839z^{-3}. \quad (3.43)$$



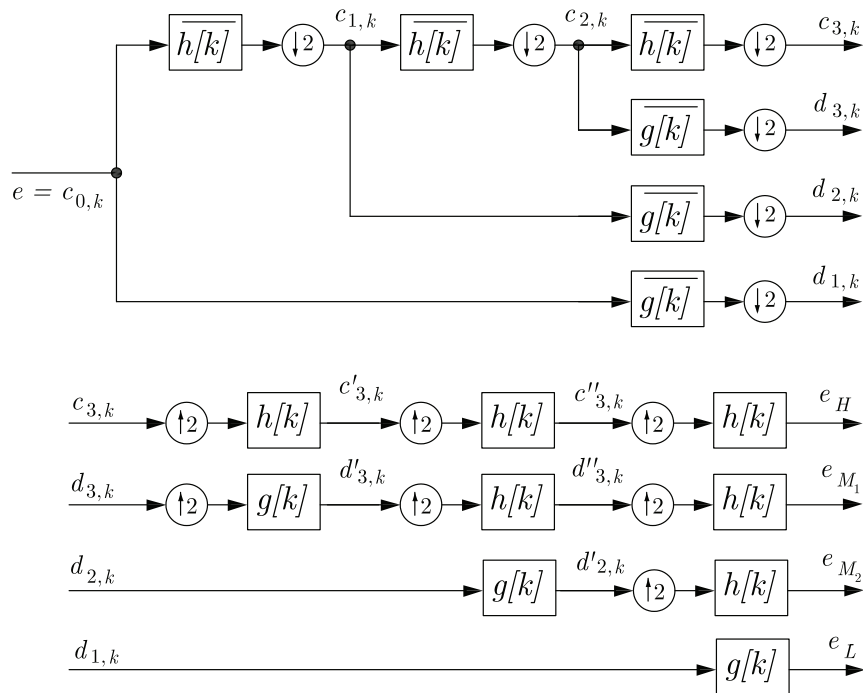


Figura 3.5: Diagrama de un esquema de decodificación sub-banda, para el análisis y síntesis de la descomposición en la parte superior y en la parte inferior de la figura, respectivamente. Donde  $h$  y  $g$ , representan los filtros pasa bajas y pasa altas, respectivamente y  $\overline{h[k]}$ ,  $\overline{g[k]}$  representan sus complejos conjugados, respectivamente,  $\downarrow 2$  y  $\uparrow 2$ , representan la decimación e interpolación a 2, respectivamente.

$h$	0.4830	0.8365	0.2241	-0.1294
$g$	-0.1294	0.2241	0.8365	0.4830

Tabla 3.2: Coeficientes del filtro Daubechies de orden 2.

## ALGORITMO 3.3: Algoritmo de control PIDMR

1. Se obtiene los datos de la señal de salida de la planta  $y$ , en un periodo de muestreo  $T_m$ .
2. Se calcula la señal del error  $e$  para cada valor  $y$  muestreado.
3. Posteriormente, se procede a descomponer la señal del error  $e$ , dando como resultado  $e_H$ ,  $e_{M_1}$ ,  $e_{M_2}$  y  $e_L$ . Esto corresponde para un nivel 3 de descomposición.
4. Se calcula la señal del control  $u$ , con la ecuación (3.8), para un nivel 3 de descomposición.
5. Por último se procede a inyectar la señal de control  $u$  a la planta.
6. Repetir los pasos 1 al 5 para cada periodo de muestreo  $T_m$ .

A continuación se dan los resultados obtenidos de la simulación del control PID *wavelet* con diferentes tipos de sistemas tales como un motor de CD, un sistema térmico y un robot planar.

### 3.5.1. Motor de CD

El control PID *wavelet* aplicado en el control de posición de un motor de CD con la siguiente función de transferencia [81]:

$$G(s) = \frac{b}{s(Js + c)} \quad (3.44)$$

donde  $b$  constante de par,  $c$  coeficiente de fricción y  $J$  inercia total del motor y la carga, cuyos valores están dados en la Tabla 3.4. Este sistema es considerado para agregar un control PID clásico y un PID *wavelet* de nivel  $N = 3$ , los valores de las ganancias son mostradas en la Tabla 3.5, las cuales son obtenidas a prueba y error.

En la Figura 3.6, se observa cómo responde el sistema con un control PID *wavelet*, que tiene similitud con la respuesta con el control PID clásico, de la misma forma se comportan la señal de control del PID clásico y la señal de control generada por el PID *wavelet* como se muestra en la Figura 3.7.

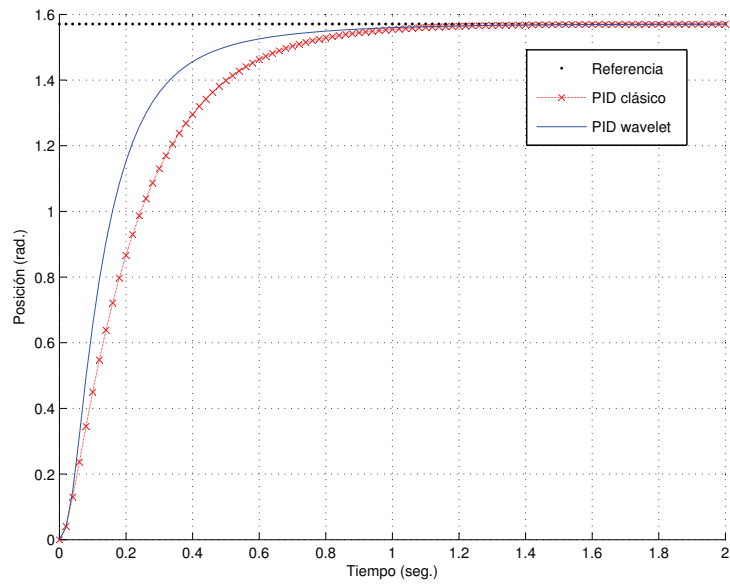


Figura 3.6: Respuesta del sistema con control PID clásico y el PID *wavelet* para una referencia de  $\pi/2$  rad.

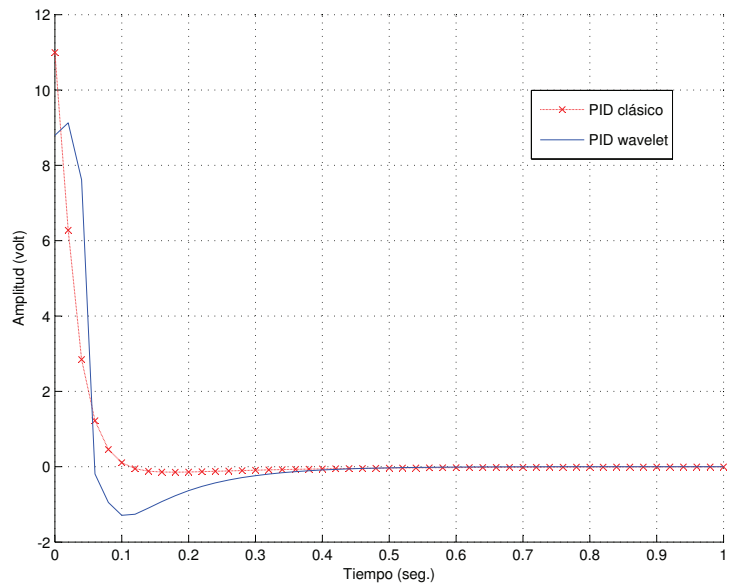


Figura 3.7: Señal de control del PID clásico y del PID *wavelet* para una referencia de  $\pi/2$  rad.

Parámetros	Valores	Unidades
$b$	22	$N \cdot m/volts$
$c$	4	$kg \cdot m^2/seg \cdot rad$
$J$	1	$kg \cdot m^2/rad$

Tabla 3.4: Parámetros del motor de CD.

Ganancias del PID clásico	$K_P$	$K_D$	$K_I$	
	7	1.5	0	
Ganancias del PID <i>wavelet</i>	$k_H$	$K_{M_1}$	$K_{M_2}$	$K_L$
	0.15	4	10	0

Tabla 3.5: Valores de las ganancias para el control PID clásico y el PID *wavelet* para el sistema (3.44).

En la Figura 3.8, se observa el comportamiento del error, siendo esta señal empleada por el control PID clásico utiliza para generar la señal de control al derivarla, integrarla y escalarla. Mientras que el control PID *wavelet* la descompone y posteriormente la escala, como se puede observar en la Figura 3.9

En la Figura 3.9, se observa como la componente  $e_H$  es la que mayor contribuye dentro de la ley de control, como lo hace la parte proporcional de un PID clásico, alcanzando su máximo valor en 0.1 segundos. Al inicio la componente  $e_{M_2}$  es el que se encarga de vencer la inercia que tiene el sistema y se puede ver como el componente  $e_{M_1}$  está siempre en oposición a  $e_{M_2}$  para evitar un sobre impulso después de alcanzado la referencia y tratar de mantener el sistema en estado estable. Interactuando, para lograr llegar de manera rápida a la referencia y evitando un sobre impulso si se sintonizan las ganancias  $K_{M_1}$  y  $K_{M_2}$  de manera adecuada. La componente  $e_L$  no es considerado debido a que se desprecia al elegir su ganancia  $K_L = 0$ , al generar cierta oscilación y sobre impulso, además de que este componente hablando en términos de frecuencia contienen las frecuencias altas, y para visualizar este fenómeno se realiza una simulación inyectando ruido al sistema.

Para analizar el comportamiento del sistema ante la presencia de ruido blanco, se inyecta una señal de ruido blanco con amplitud máxima de  $\pm 0.1$  radianes, como se muestra en la Figura 3.10, en la medición de la salida tanto para el control PID clásico como para el control PID *wavelet*, como se muestra en la Figura 3.1 y 3.2. Los resultados son mostrados en las Figuras 3.11, 3.12, 3.13 y 3.14.

En la Figura 3.11, se observa como la señal de salida del sistema con un control

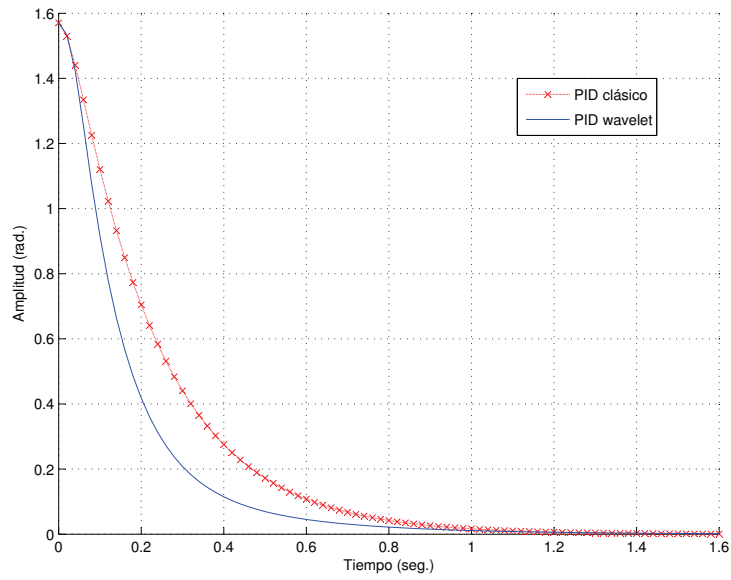


Figura 3.8: Señal de error del sistema con un control del PID clásico y un PID *wavelet* para una referencia de  $\pi/2$  rad.

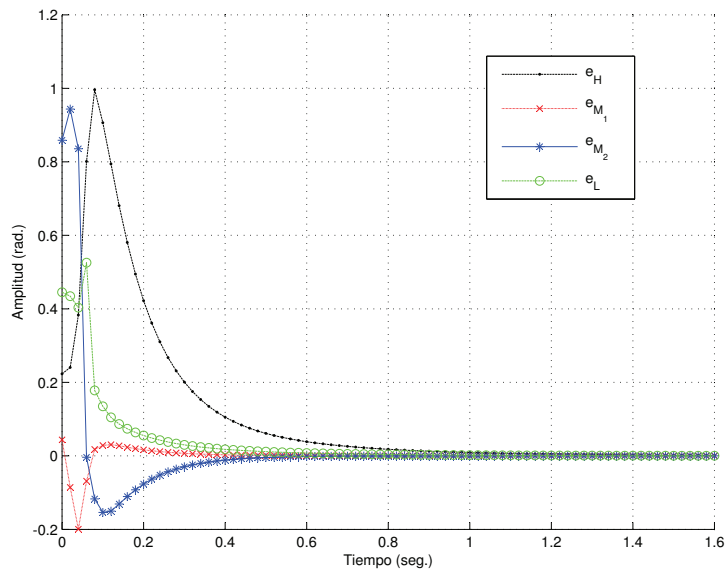


Figura 3.9: Señales generadas a partir de la descomposición a un nivel 3 del error del sistema con un control PID *wavelet* para una referencia de  $\pi/2$  rad.

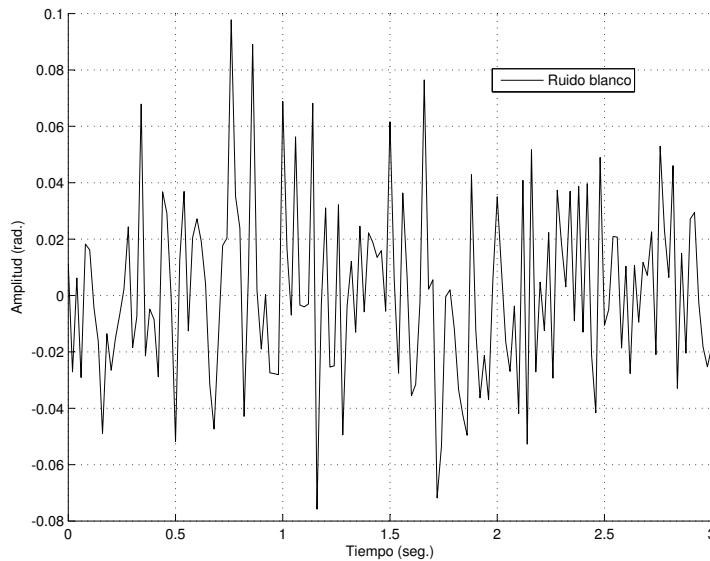


Figura 3.10: Señal de ruido blanco introducido en el sistema.

PID *wavelet* es similar en comportamiento que en la Figura 3.6, mientras que en la señal de salida del sistema con un control PID clásico se ve ciertas variaciones generadas por el ruido en la medición de la señal de salida.

En la Figura 3.12, se observa como la señal de control del PID clásico varía alrededor de  $\pm 15$  volts con variaciones abruptas, lo que puede generar esfuerzo y desgastar poco a poco la vida del motor. Mientras que la señal de control de un PID *wavelet* es un señal más suave comparada con la señal del PID clásico.

Lo anterior se debe a que aunque las señales de error son similares como se muestra en la Figura 3.13, donde se ve que ambas contienen el ruido, el control PID *wavelet* descompone la señal como se observa en la Figura 3.14 y discrimina al ruido contenido en la señal del error, al escalar al componente  $e_L$  con  $K_L = 0$  que es la señal que contiene gran parte del ruido, lo que no se puede hacer con el componente  $e_{M_2}$ , por que se requiere al inicio para darle la potencia necesaria a la señal de control para vencer la inercia del sistema.

Como se puede observar en las simulaciones anteriores el control PID *wavelet* tiene la característica de ser inmune ante la presencia del ruido.

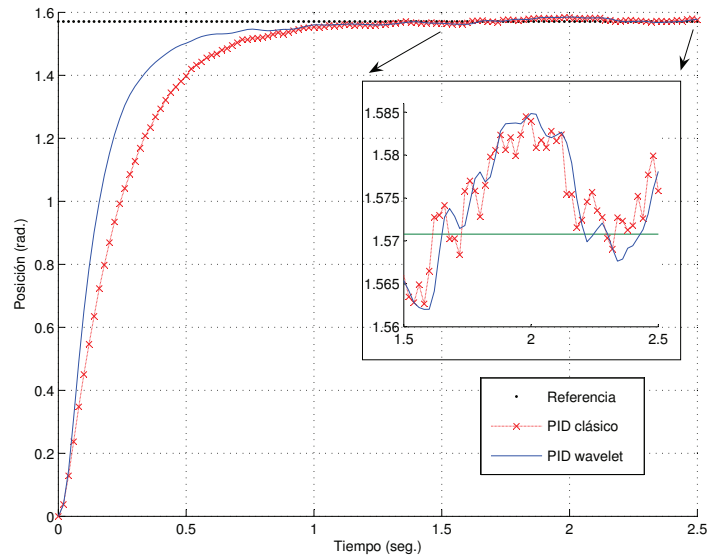


Figura 3.11: Respuesta del sistema con control PID clásico y el PID *wavelet* para una referencia de  $\pi/2$  rad. ante la presencia de ruido blanco.

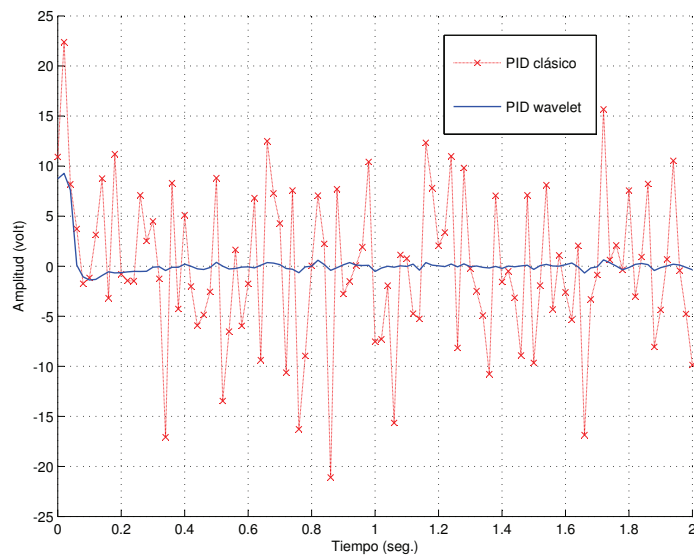


Figura 3.12: Señal de control del PID clásico y del PID *wavelet* con ruido blanco.

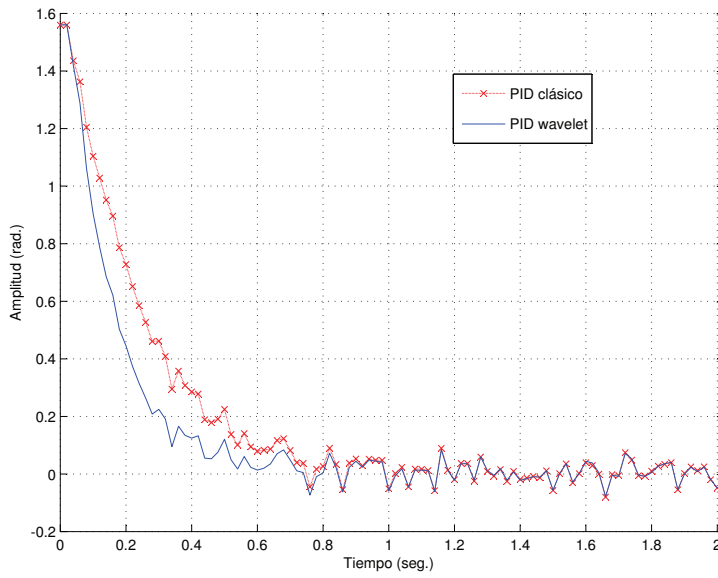


Figura 3.13: Señales de error para un sistema con un control PID clásico y un PID *wavelet* ante la presencia de ruido blanco.

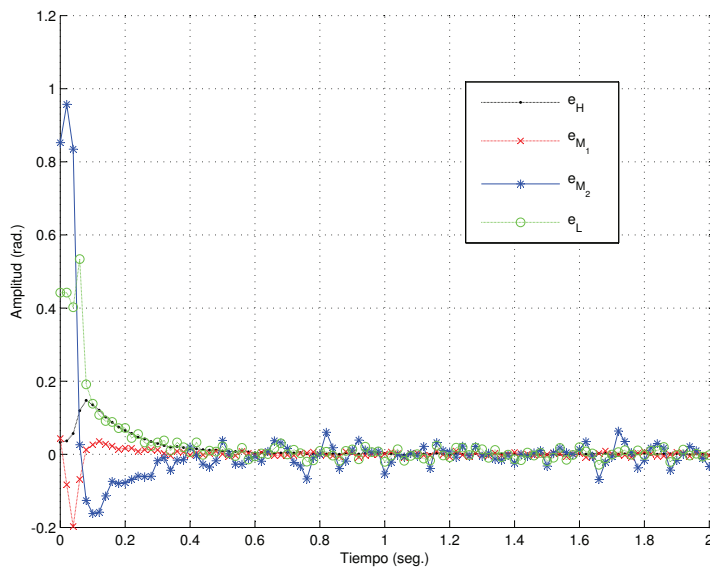


Figura 3.14: Señales generadas a partir de la descomposición a un nivel 3 del error del sistema con un control PID *wavelet* con ruido.



### 3.5.2. Sistema térmico

El siguiente es un sistema común en problemas de control de temperatura, mostrado en la Figura 3.15. El sistema es tomado de [21], cuya función de transferencia es

$$G(s) = \frac{T_e(s)}{T_{ec}(s)} = \frac{e^{-s\tau}}{\frac{s}{a} + 1}, \quad (3.45)$$

donde  $T_e$  es la temperatura del tanque,  $T_{ec}$  temperatura a la salida de la válvula mezcladora,  $\tau$  es el tiempo de retardo en segundos por la transportación del material en el ducto, que conecta a la válvula mezcladora con el tanque,  $a = \frac{\dot{m}}{M}$ ,  $\dot{m}$  es la razón del flujo de la masa en Kg./seg. y  $M$  es la masa del fluido contenido en el tanque en Kg.

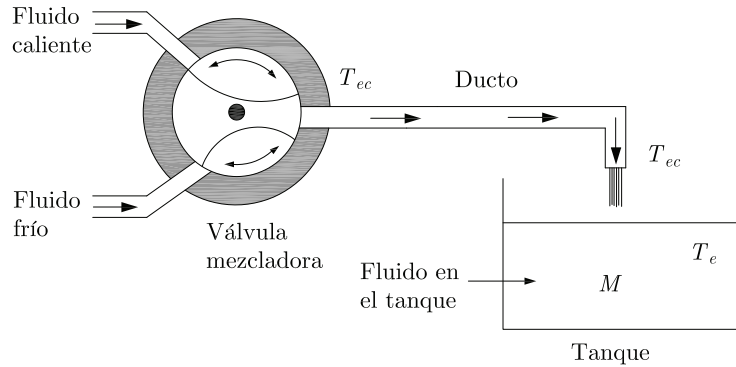


Figura 3.15: Sistema térmico, donde  $T_e$ ,  $T_{ec}$ ,  $M$  son la temperatura del tanque, la temperatura a la salida de la válvula mezcladora, la masa del fluido contenido en el tanque, respectivamente.

El sistema es expuesto ante una señal de ruido mostrado en la Figura 3.16, los valores del sistema con retardo empleados son:  $a = 1$ ,  $\tau = 5 \text{ seg}$  y en la Tabla 3.6 se muestran los valores de las ganancias del control PID clásico y del control PID *wavelet*, con el método heurístico y a prueba y error, respectivamente.

Ganancias de PID clásico	$K_P$	$K_D$	$K_I$	
	0.2	0.04	0.1	
Ganancias del PID <i>wavelet</i>	$k_H$	$K_{M_1}$	$K_{M_2}$	$K_L$
	1	85	10	1

Tabla 3.6: Valores de las ganancias para el control PID clásico y el PID *wavelet* para el sistema con retardo (3.45).

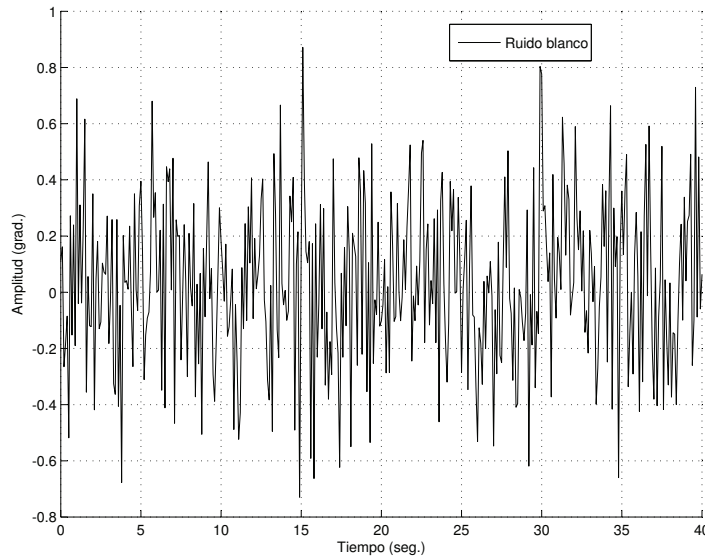


Figura 3.16: Señal de ruido blanco introducido en el sistema con retardo.

En la Figura 3.17, se observa que al llegar cerca de la referencia la salida del sistema con un control PID *wavelet* decae y empieza a oscilar, lo cual ya no describe de manera correcta el comportamiento del sistema térmico, lo mismo ocurre con la señal de error, como se muestra en la Figura 3.19, todo esto ocasionado por la señal de control, como se observa en la Figura 3.18.

Para eliminar las oscilaciones a la salida del sistema térmico, se propone integrar el error en bajas frecuencias  $e_H$ , es decir

$$u = K_H \int_0^t e_H dt + K_{M_1} e_{M_1} + \dots + K_{M_{N-1}} e_{M_{N-1}} + K_L e_L, \quad (3.46)$$

Las ganancias del control PID *wavelet* son dadas en la Tabla 3.7, mientras que las ganancias del control PID clásico son los mismo a los empleados anteriormente.

Ganancias del PID <i>wavelet</i>	$k_H$	$K_{M_1}$	$K_{M_2}$	$K_L$
	0.12	1.2	1.2	0

Tabla 3.7: Valores de las ganancias para el PID *wavelet* para el sistema con retardo (3.45).

En la Figura 3.20 se observa como el sistema con retardo llega a la referencia casi al mismo tiempo, tanto empleando un control PID clásico como con un control PID *wavelet* con la ley de control (3.46), mientras que en la Figura 3.21, se observa que la

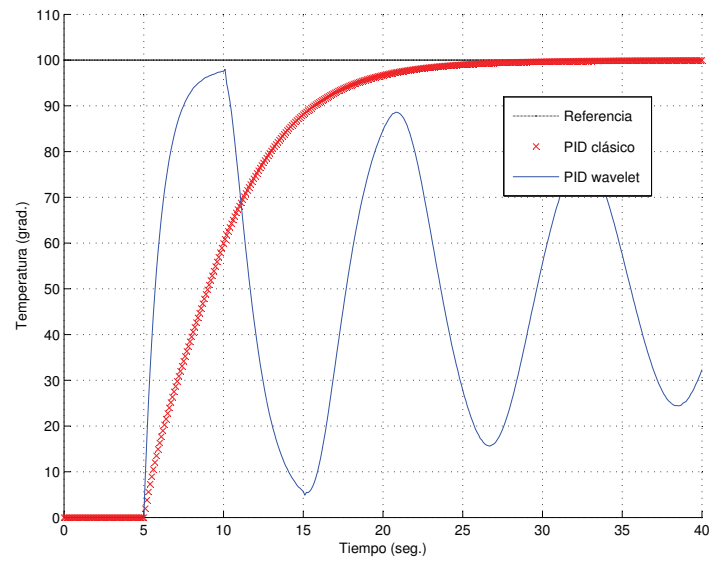


Figura 3.17: Respuesta del sistema con retardo con control PID clásico y del PID *wavelet* para una referencia de 100 grados centígrados, con un señal de ruido inyectado.

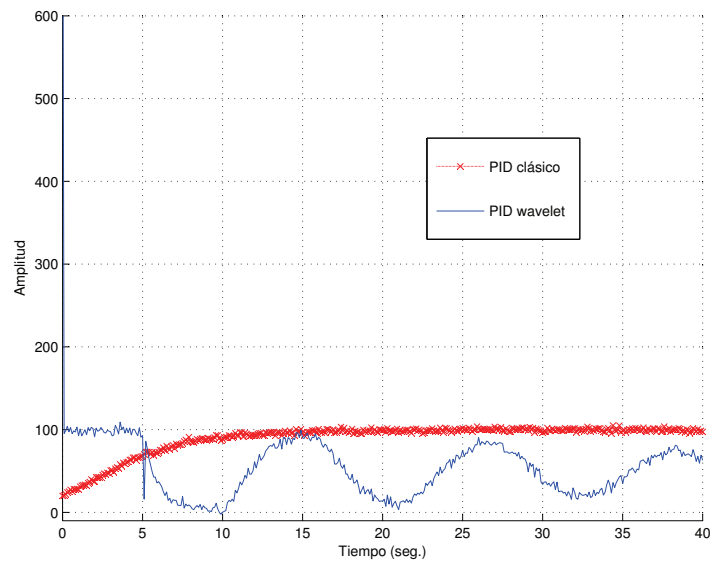


Figura 3.18: Señal de control del PID clásico y la señal de control del PID *wavelet*, ante la presencia de un señal de ruido.

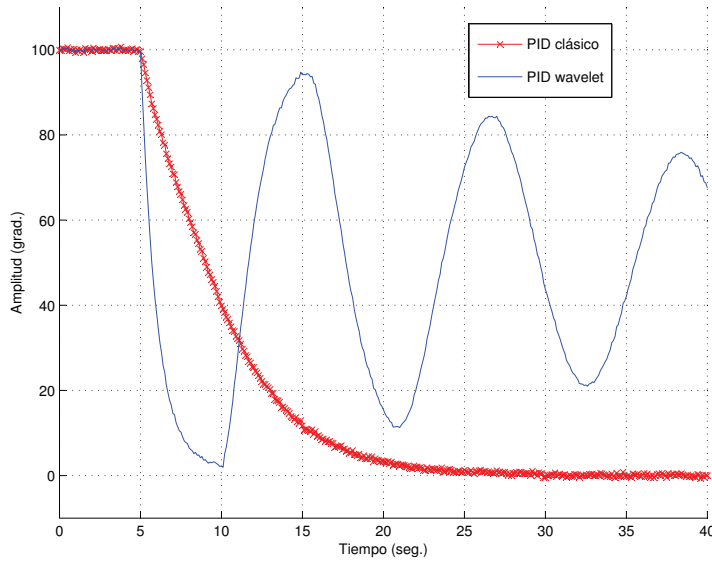


Figura 3.19: Señal de error del sistema con retardo controlado con un PID clásico y con un PID *wavelet*, ante la presencia de un señal de ruido blanco.

señal de control del PID *wavelet* es mucho más suave que la señal de control del PID clásico, siendo ésta la característica más destacada del control PID *wavelet*.

La señal del error del sistema con retardo con el control PID clásico y la señal del error del sistema con retardo con el control PID *wavelet*, son mostrados en la Figura 3.22, en la cual se puede observa que son similares y ambas señales contienen la señal de ruido, la diferencia como se ha mencionado antes es que el control PID *wavelet*, al descomponer la señal del error en sus diferentes componentes permite ser selectivo y encontrar una ley de control adecuada para el sistema con retardo. En la Figura 3.23, se muestran las señales de los componentes de la señal de error del sistema con retardo controlado con el PID *wavelet*.

### 3.5.3. Robot planar de 2 grados de libertad

La siguiente simulación se realiza sobre una un sistema denominado RPDGDL *robot planar de 2 grados de libertad*, mostrado en la Figura 3.27, para seguimiento de una trayectoria dada, el modelo matemático que representa a este sistema está dado por:

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\mathbf{q}) [\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}], \quad (3.47)$$

donde  $\mathbf{q} \in \mathbb{R}^{2 \times 1}$  es el vector de coordenadas articulares,  $\mathbf{H} \in \mathbb{R}^{2 \times 2}$  es la matriz de inercia,  $\mathbf{C} \in \mathbb{R}^{2 \times 2}$  es la matriz de fuerzas de coriolis y centrípetas,  $\mathbf{F} \in \mathbb{R}^{2 \times 1}$  es el

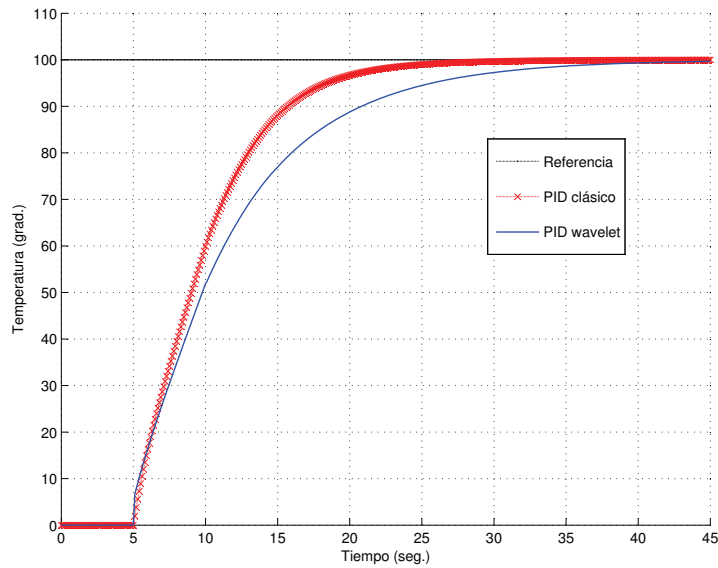


Figura 3.20: Respuesta del sistema con retardo con control PID clásico y del PID *wavelet* para una referencia de 100 grados centígrados, con una señal de ruido inyectado.

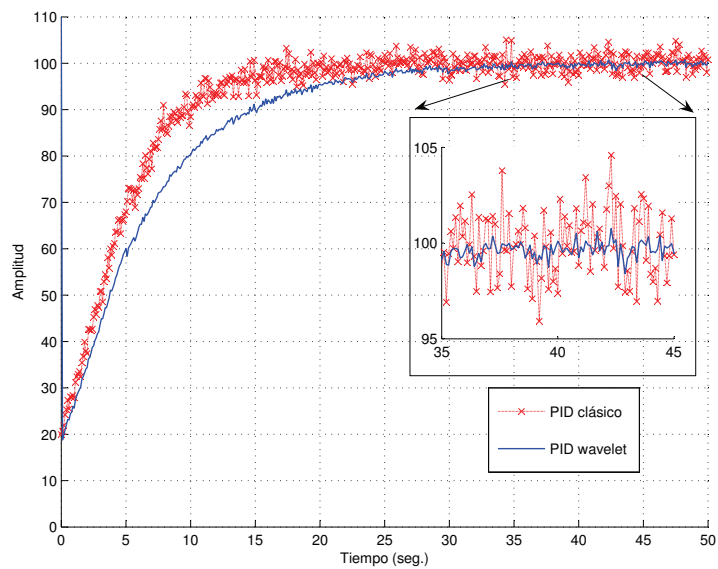


Figura 3.21: Señal de control del PID clásico y la señal de control del PID *wavelet*, ante la presencia de una señal de ruido.

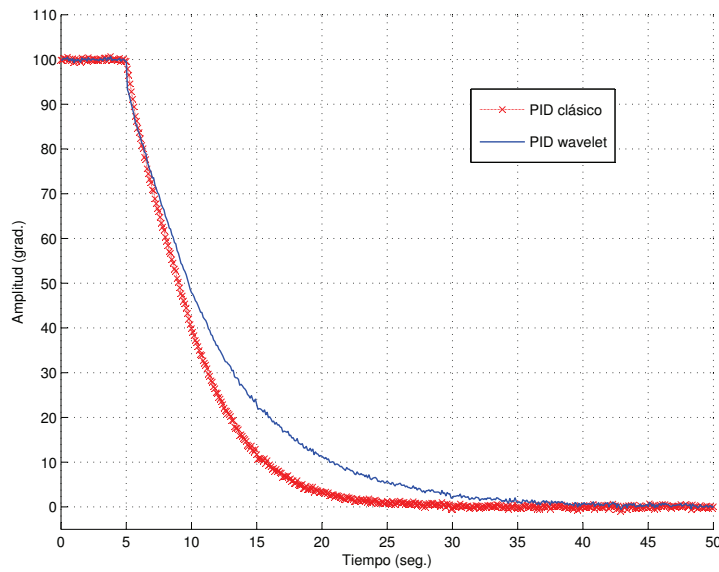


Figura 3.22: Señal del error del sistema con retardo controlado con un PID clásico y la señal del error del sistema con retardo controlado con un PID *wavelet*, ante la presencia de un señal de ruido.

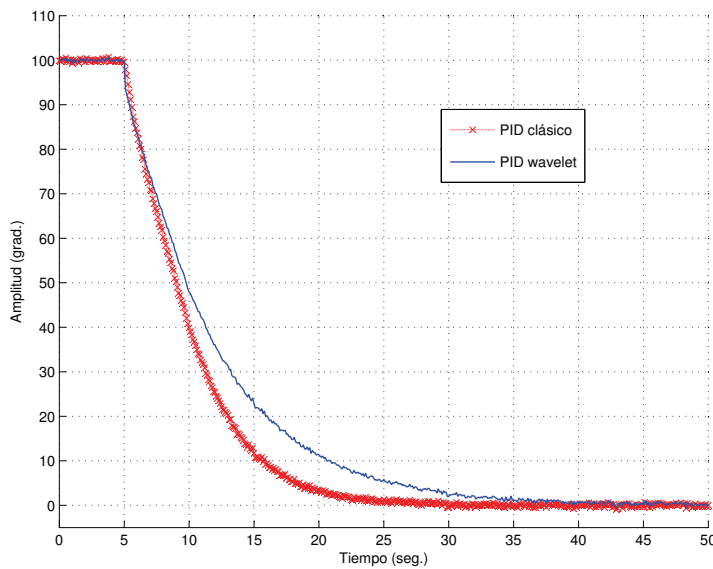


Figura 3.23: Señales generadas a partir de la descomposición a un nivel 3 de la señal del error del sistema con retardo controlado con un PID *wavelet*, ante la presencia de una señal de ruido.

vector de de fuerzas de fricción y  $\tau \in \mathbb{R}^{2 \times 1}$  es el vector de par generalizado. Donde

$$\mathbf{H} = \begin{bmatrix} a_1 + 2a_2 \cos(x_2) + a_3 & a_2 \cos(x_2) + a_3 \\ a_2 \cos(x_2) + a_3 & a_3 \end{bmatrix}, \quad (3.48)$$

$$\mathbf{C} = \begin{bmatrix} -2a_2 \sin(x_2)x_4 & -a_2 \sin(x_2)x_4 \\ a_2 \sin(x_2)x_3 & 0 \end{bmatrix}, \quad (3.49)$$

$$\mathbf{F} = \begin{bmatrix} b_1 x_3 + K_1 \tanh(\beta x_3) \\ b_1 x_4 + K_1 \tanh(\beta x_4) \end{bmatrix}, \quad (3.50)$$

$$\tau = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (3.51)$$

con:

$$a_1 = (m_1 + m_2)l_1^2, \quad (3.52)$$

$$a_2 = m_2 l_1 l_2, \quad (3.53)$$

$$a_3 = m_2 l_2^2, \quad (3.54)$$

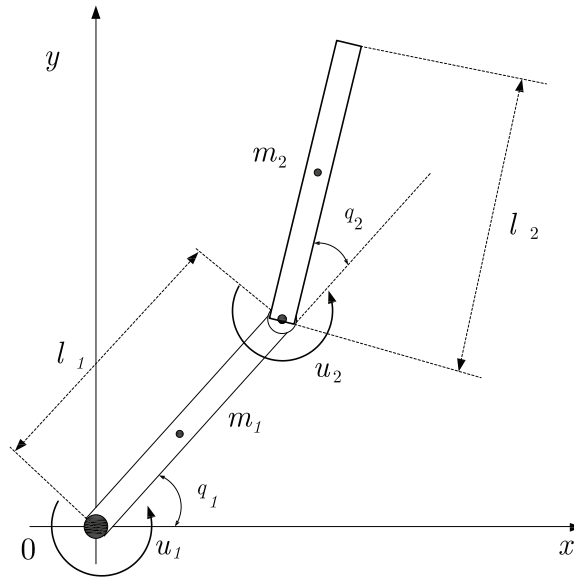


Figura 3.24: Sistema del robot planar de dos grados de libertad sin acción de gravedad, donde  $m_1$ ,  $m_2$ ,  $l_1$ ,  $l_2$ ,  $q_1$ ,  $q_2$  son la masa en el eslabón 1, la masa en el eslabón 2, la longitud del eslabón 1, la longitud del eslabón 2, la variable articular 1 y la variable articular 2, respectivamente; y  $u_1$ ,  $u_2$  son el par aplicado al actuador 1 y al actuador 2, respectivamente.

donde los valores de las constantes son:  $m_1 = m_2 = 0.45Kg.$ ,  $l_1 = l_2 = 0.35mts.$ ,  $\beta = 10$ ,  $K_1 = K_2 = 0.15 N/rad.$ ,  $b_1 = b_2 = 0.25 N \cdot seg./rad.$ , para ver más a detalle sobre este sistema ver [42, 79]. La Figura 3.25, muestra el ruido inyectado al sistema.

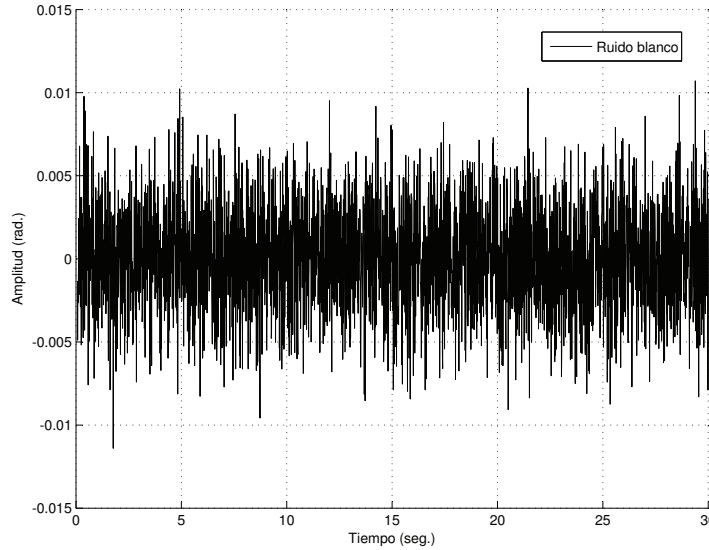


Figura 3.25: Señal de ruido blanco introducido al robot planar.

Las ganancias empleadas en el PID clásico para  $u_1$  y  $u_2$  son mostradas en la Tabla 3.8 y las ganancias empleadas en el PID *wavelet* son mostradas en la Tabla 3.9.

Ganancias del PID clásico	$k_P$	$K_D$	$K_I$
Ganancias para el control $u_1$	6	2	0
Ganancias para el control $u_2$	6	2	0

Tabla 3.8: Valores de las ganancias para el PID clásico para el robot planar (3.47).

Ganancias del PID <i>wavelet</i>	$k_H$	$K_{M_1}$	$K_{M_2}$	$K_L$
Ganancias para el control $u_1$	2.5	6	0	0
Ganancias para el control $u_2$	2.5	6	0	0

Tabla 3.9: Valores de las ganancias para el PID *wavelet* para el robot planar (3.47).

En las Figuras 3.27, 3.26, 3.28 y 3.29 se puede observar que el control *wavelet* aplicado a un robot planar, preserva sus características, de mejorar la señal de control



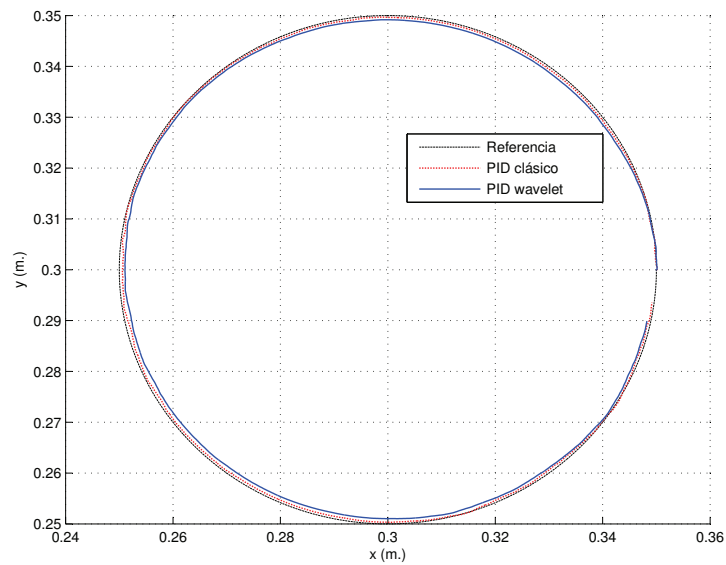


Figura 3.26: Descripción de la tarea a seguir del robot con un control PID clásico y un control PID *wavelet*.

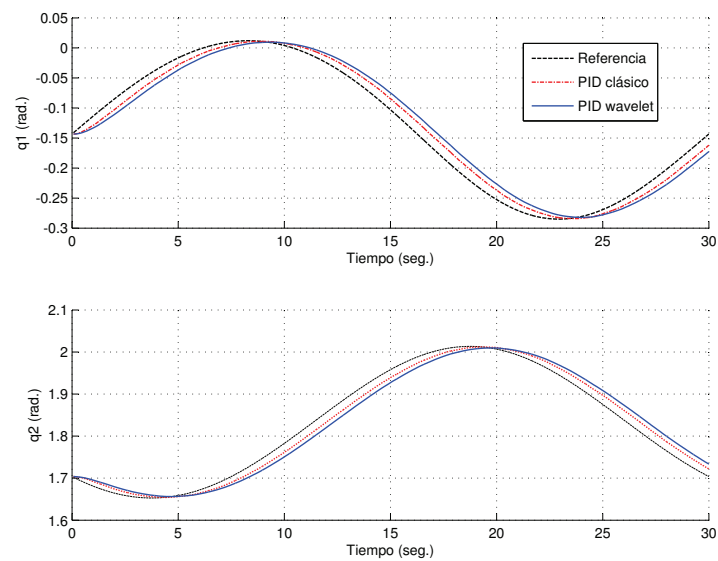


Figura 3.27: Comportamiento de las variables articulares  $q_1$  y  $q_2$  con un control PID clásico y con un control PID *wavelet*.

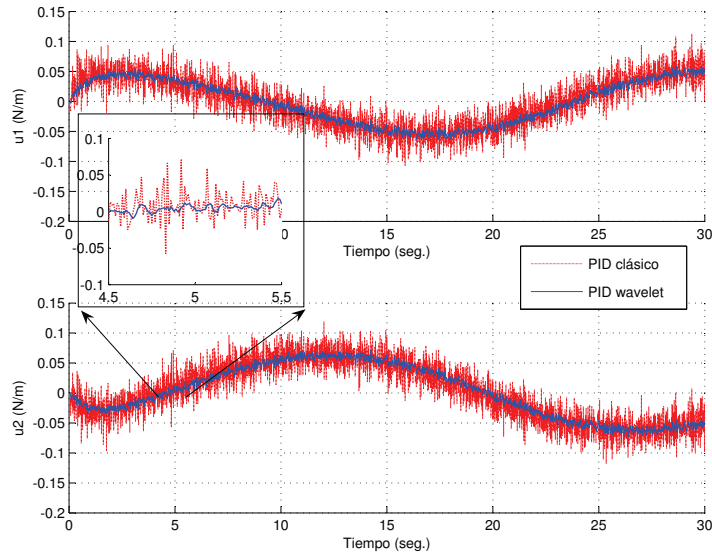


Figura 3.28: Señales de control para los actuadores 1 y 2, generados con un control PID clásico y con un control PID *wavelet*, donde:  $u_1$  y  $u_2$  son las señales de control para  $\tau_1$  y  $\tau_2$ , respectivamente.

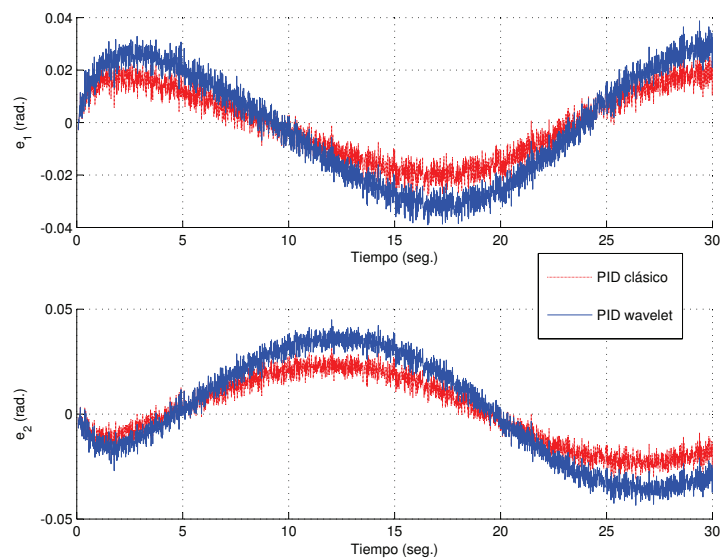


Figura 3.29: Señales de error  $e_1$  y  $e_2$ .

y ser inmune al ruido, y además de que la planta tiene un comportamiento aceptable y permite eliminar las señales de ruido no deseadas.

### 3.6. Comentarios

Del trabajo realizado en este capítulo se puede comentar que el PID *wavelet* presenta resultados aceptables sobre los sistemas donde fue implementado, como en el motor de CD, el sistema térmico y el robot planar. Los resultados obtenidos se debe a la descomposición que el diseñador puede realizar, en teoría la descomposición puede ser infinita, pero en la práctica es recomendable que sea finita, debido a las limitaciones computacionales, además que entre más grande es la descomposición mayor es el número de ganancias a sintonizarse. Es importante mencionar que se hicieron varias pruebas en simulación numérica empleando el PID *wavelet* en otras plantas en las que no fue posible sintonizar el PID *wavelet* para lograr estabilizar dichas planta es por esta razón que no se presentan resultados. Es importante comentar que para la sintonización del PID clásico existen métodos de sintonización como por ejemplo Ziegler-Nichols, por asignación de polos, mientras que para el PIDMR no existe ningún método de sintonización similar. Una alternativa a estos métodos es emplear redes neuronales para el cálculo de las ganancias del controlador PIDMR, lo cual se verá en el capítulo siguiente.



# Capítulo 4

## Desarrollo de un control PID *wavenet*

En este capítulo se presenta un controlador PID *wavenet* basado en redes neuronales *wavelets* cuyas funciones de activación son *wavelet* hijas. La organización del presente capítulo es: en la Sección 4.1 se da una breve introducción del controlador *wavenet*, en la Sección 4.2 se presenta la estructura del controlador *wavenet* y su algoritmo. El algoritmo del control *wavenet*, para sintonizar los parámetros en línea del control *wavelet*, es presentado en la Sección 4.3, los resultados de simulación son mostrados en la Sección 4.3 y por último en la Sección 4.5 se dan algunos comentarios acerca del capítulo presentado.

### 4.1. Introducción

Los enfoques de control adaptables tradicionales son limitados en lo que respecta a no poder tratar con sistemas no lineales complejos. Típicamente, estas técnicas suponen que el modelo de control está operando en una región lineal. Los parámetros del modelo de la planta linealizada son calculados recursivamente y usados para actualizar el controlador. En general, el diseño de un controlador basado en el análisis matemático para tales plantas que constan de la no linealidad e incertidumbres, es muy complicado. El problema empeora cuando las funciones que describen la planta son desconocidas y variantes en el tiempo. Tales problemas de control adaptable no lineal variantes en el tiempo no lineales están surgiendo con una frecuencia creciente en la tecnología de hoy. Por esto es importante desarrollar una técnica eficaz en la que la estructura de los modelos desconocidos de la planta ya sea lineales o no lineales pueden ser identificados como un proceso adaptable; y los controladores tienen que ser diseñados para actuar rápidamente, con exactitud y en un modo estable.

La propuesta de esta tesis, es emplear *wavenets*, es decir redes neuronales con

bases *wavelets* como funciones de activación, como en [43, 75, 85], para sintonizar los parámetros de un PID. De la misma forma el control multiresolución propuesto en esta tesis es el descrito por (3.10), el cual es sintonizado por una red neuronal *wavenet*.

## 4.2. Sistemas no lineales discretos

Dado un sistema dinámico general SISO (una entrada - una salida) representado en tiempo discreto por las siguientes ecuaciones de estado [40, 41]:

$$x(k+1) = f[x(k), u(k), k], \quad (4.1)$$

$$y(k) = g[x(k), k], \quad (4.2)$$

donde  $x \in \mathbb{R}^n$  y  $u, y \in \mathbb{R}$ , además las funciones  $f, g \in \mathbb{R}$  son desconocidas, y los únicos datos accesibles son la entrada  $u$  y la salida  $y$ . En [40] se demostró que si el sistema linealizado de (4.1) y (4.2) alrededor del estado de equilibrio es observable, existe una representación entrada-salida la cual tiene la siguiente forma:

$$y(k+1) = \Omega[\mathbf{Y}(k), \mathbf{U}(k)], \quad (4.3)$$

$$\mathbf{Y}(k) = [y(k) \ y(k-1) \ \cdots \ y(k-n+1)], \quad (4.4)$$

$$\mathbf{U}(k) = [u(k) \ u(k-1) \ \cdots \ u(k-n+1)], \quad (4.5)$$

i.e., existe una función  $\Omega$  que mapea al par  $(y(k), u(k))$  y sus valores  $n-1$  pasados, dentro de  $y(k+1)$ .

Entonces un modelo de red neuronal *wavelet*  $\hat{\Omega}$ , puede ser entrenado para aproximar  $\Omega$  sobre el dominio de interés. Prácticamente, incluso si el modelo de la planta está disponible, los modelos aproximados se adaptan para actualizar los parámetros de control en línea. Las consideraciones están basadas sobre el diseño del control de red neuronal del sistema de control. Un modelo alternativo de una planta desconocida que puede simplificar el algoritmo de la entrada de control es descrito por la ecuación siguiente:

$$y(k+1) = \Phi[\mathbf{Y}(k), \mathbf{U}(k)] + \Gamma[\mathbf{Y}(k), \mathbf{U}(k)] \cdot u(k) \quad (4.6)$$

donde  $y(k)$  y  $u(k)$  denotan la entrada y la salida en el instante  $k$  de tiempo. Si los términos  $\Phi(\cdot)$  y  $\Gamma(\cdot)$  son exactamente conocidos, el control requerido  $u(k)$  para lograr una salida deseada  $y_{ref}(k+1)$ , se obtiene igualando  $y(k+1) = y_{ref}(k+1)$ , es decir

$$y(k+1) = \Phi[\mathbf{Y}(k), \mathbf{U}(k)] + \Gamma[\mathbf{Y}(k), \mathbf{U}(k)] \cdot u(k) = y_{ref}, \quad (4.7)$$

resolviendo para  $u(k)$ , resulta

$$u(k) = \frac{y_{ref}(k+1) - \Phi[\mathbf{Y}(k), \mathbf{U}(k)]}{\Gamma[\mathbf{Y}(k), \mathbf{U}(k)]}, \quad (4.8)$$

sin embargo, si los términos  $\Phi(\cdot)$  y  $\Gamma(\cdot)$  son desconocidos, entonces aquí es donde se emplea un modelo de redes neuronales *wavelets* adaptable, para aproximar el sistema dinámico, de la siguiente manera

$$\hat{y}(k+1) = \hat{\Phi}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Phi}] + \hat{\Gamma}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Gamma}] \cdot u(k), \quad (4.9)$$

comparando el modelo de (4.9) con el (D.12) dado por

$$\hat{y}(t) = \mathbf{C}^T \mathbf{Z}(t) u(t) + \mathbf{D}^T \hat{\mathbf{Y}}(t) v(t), \quad (4.10)$$

se concluye que

$$\hat{\Phi}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Phi}] = \mathbf{D}^T \hat{\mathbf{Y}}(k) v(k), \quad (4.11)$$

$$\hat{\Gamma}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Gamma}] = \mathbf{C}^T \mathbf{Z}(k), \quad (4.12)$$

$$z(k) = \Psi^T \mathbf{W}. \quad (4.13)$$

Después de que las no linealidades  $\Phi(\cdot)$  y  $\Gamma(\cdot)$  son aproximados con las dos distintas funciones de la red neuronal  $\hat{\Phi}(\cdot)$  y  $\hat{\Gamma}(\cdot)$  con parámetros ajustables (como son: los pesos  $w$ , las dilataciones  $a$ , las traslaciones  $b$ , los coeficientes del filtro IIR  $c$  y  $d$ ), representados por  $\Theta_{\Phi}$  y  $\Theta_{\Gamma}$  respectivamente, la señal de control  $u(k)$  para lograr una señal de salida  $y_{ref}(k+1)$  puede ser obtenida a partir de:

$$u(k) = \frac{y_{ref}(k+1) - \hat{\Phi}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Phi}]}{\hat{\Gamma}[\mathbf{Y}(k), \mathbf{U}(k), \Theta_{\Gamma}]}. \quad (4.14)$$

El esquema del control *wavenet*, es mostrado en la Figura 4.1, y en la siguiente sección se muestra el algoritmo para sintonizar los parámetros del control *wavelet* y los parámetros de la red neuronal *wavenet*, como los pesos, los valores de dilatación y traslación de cada *wavelet* hija.

### 4.3. Algoritmo de sintonización del control *wavenet*

El control *wavelet* visto en el capítulo anterior, descrito por (3.8), posee la dificultad para ser sintonizado, además de ser un control lineal y el cual no puede tratar con procesos con dinámicas complejas como aquellas con mayor tiempo muerto, respuesta inversa y características de altas no linealidades. Para mejorar el comportamiento del control, se emplea un algoritmo para autosintonizar los valores de cada ganancia en

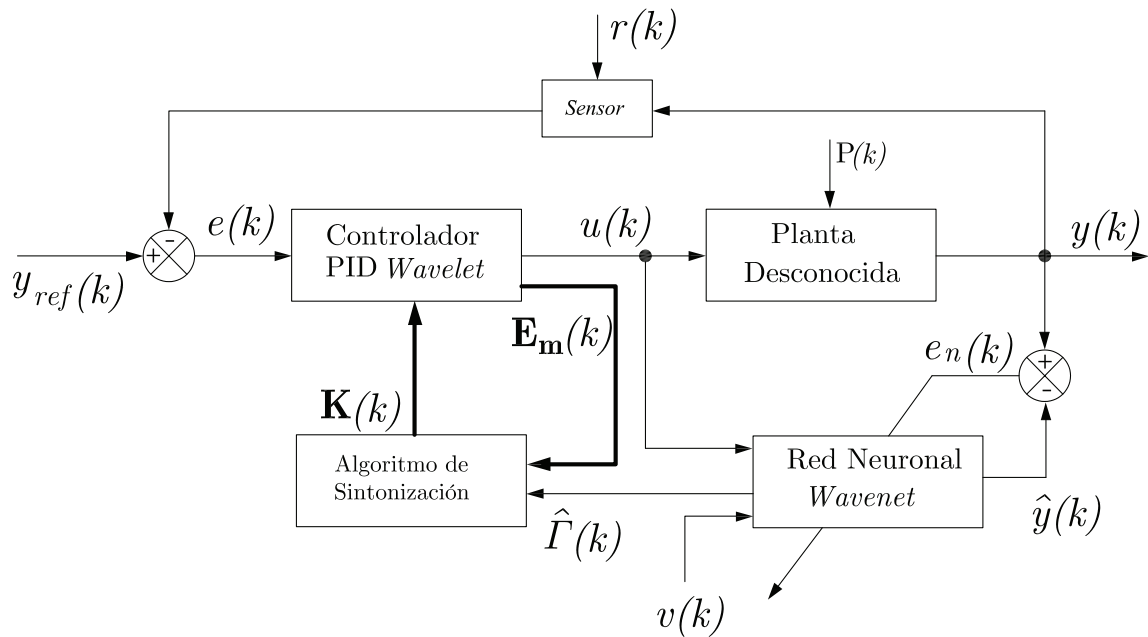


Figura 4.1: Diagrama a bloques del PID *wavenet*, donde  $e$  y  $e_n$  son el error entre la señal de referencia  $y_{ref}$  y la salida de la planta  $y$ , y el error entre la salida de la planta  $y$  y la salida estimada por la red neuronal *wavenet*  $\hat{y}$ , respectivamente;  $u$ ,  $v$  y  $\hat{\Gamma}$ , son la señal de control, el coeficiente de velocidad de aprendizaje y el coeficiente de no linealidad estimado, respectivamente;  $\mathbf{K}$  y  $\mathbf{E}_m$  son el vector de las ganancias del controlador *wavelet* actualizados y el vector de los componentes de la señal del error  $e$ , respectivamente;  $P$  y  $r$  son las perturbaciones externas al sistema y el ruido en la medición, respectivamente.



$\mathbf{K}$  del control *wavelet*, de la misma forma que se autosintonizan los parámetros de un PID como en [75], con la diferencia que aquí se pueden tener más de tres parámetros según el nivel de descomposición del control *wavelet*. Antes se define:

$$\mathbf{W}(k) \triangleq [w_1(k) \ w_2(k) \ \cdots \ w_l(k) \ \cdots \ w_{L-1}(k) \ w_L(k)]^T, \quad (4.15)$$

$$\mathbf{A}(k) \triangleq [a_1(k) \ a_2(k) \ \cdots \ a_l(k) \ \cdots \ a_{L-1}(k) \ a_L(k)]^T, \quad (4.16)$$

$$\mathbf{B}(k) \triangleq [b_1(k) \ b_2(k) \ \cdots \ b_l(k) \ \cdots \ b_{L-1}(k) \ b_L(k)]^T, \quad (4.17)$$

$$\mathbf{C}(k) \triangleq [c_0(k) \ c_1(k) \ \cdots \ c_m(k) \ \cdots \ c_{M-1}(k) \ c_M(k)]^T, \quad (4.18)$$

$$\mathbf{D}(k) \triangleq [d_1(k) \ d_2(k) \ \cdots \ d_j(k) \ \cdots \ d_{J-1}(k) \ d_J(k)]^T, \quad (4.19)$$

$$\mathbf{\Psi}(k) \triangleq [\psi_1(\tau) \ \psi_2(\tau) \ \cdots \ \psi_l(\tau) \ \cdots \ \psi_{L-1}(\tau) \ \psi_L(\tau)]^T, \quad (4.20)$$

$$\mathbf{Z}(k) \triangleq [z(k) \ z(k-1) \ \cdots \ z(k-m) \ \cdots \ z(k-M+1) \ z(k-M)]^T, \quad (4.21)$$

$$\hat{\mathbf{Y}}(k) \triangleq [\hat{y}(k-1) \ \hat{y}(k-2) \ \cdots \ \hat{y}(k-j) \ \cdots \ \hat{y}(k-J+1) \ \hat{y}(k-J)]^T, \quad (4.22)$$

en la cual  $\psi_l(\tau) = \psi\left(\frac{k-b_l(k)}{a_l(k)}\right)$  para  $l = 1, 2, \dots, L$ , donde  $L \in \mathbb{Z}$  es el número de neuronas en la capa de la red neuronal;  $I = N + 1$ ,  $I, N \in \mathbb{R}$  donde  $N$  es el nivel del control PID multiresolución *wavelet*, y

$$\hat{y}(k) \triangleq \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k), \quad (4.23)$$

$$u(k) \triangleq \mathbf{K}(k) \mathbf{E}_m(k), \quad (4.24)$$

$$z(k) \triangleq \mathbf{\Psi}^T(k) \mathbf{W}(k), \quad (4.25)$$

donde  $\mathbf{K}$  y  $\mathbf{E}_m$  están dados por (3.11) y (3.12), respectivamente.

Los parámetros de la red *wavenet* y los parámetros del control *wavelet* son optimizados con el algoritmo de mínimos cuadrados medios LMS por medio de la minimización de un función de costo o función de energía  $E$ , a lo largo de todo el tiempo  $t$ . Si se define al error  $e_n$  entre la salida de la planta  $y$  y la salida de la red *wavenet*  $\hat{y}$ , como:

$$e_n(k) = y(k) - \hat{y}(k), \quad (4.26)$$

que es una función de error en un tiempo  $k$ , la función de energía esta definida por:

$$E = \frac{1}{2} E_n^T E_n, \quad (4.27)$$

con

$$E_n = [e_n(1) \ e_n(2) \ \cdots \ e_n(k) \ \cdots \ e_n(T)]^T. \quad (4.28)$$

Entonces, para minimizar  $E$  se usa el método del gradiente de pasos descendentes, el cual requiere los gradientes  $\frac{\partial E}{\partial \mathbf{A}(k)}$ ,  $\frac{\partial E}{\partial \mathbf{B}(k)}$ ,  $\frac{\partial E}{\partial \mathbf{W}(k)}$ ,  $\frac{\partial E}{\partial \mathbf{C}(k)}$ ,  $\frac{\partial E}{\partial \mathbf{D}(k)}$  y  $\frac{\partial E}{\partial \mathbf{K}(k)}$  para actualizar los cambios incrementales de cada parámetro en particular. Los cuales son expresados como:

$$\frac{\partial E}{\partial \mathbf{W}(k)} = \left[ \frac{\partial E}{\partial w_1(k)} \quad \frac{\partial E}{\partial w_2(k)} \quad \cdots \quad \frac{\partial E}{\partial w_l(k)} \quad \cdots \quad \frac{\partial E}{\partial w_{L-1}(k)} \quad \frac{\partial E}{\partial w_L(k)} \right]^T, \quad (4.29)$$

$$\frac{\partial E}{\partial \mathbf{A}(k)} = \left[ \frac{\partial E}{\partial a_1(k)} \quad \frac{\partial E}{\partial a_2(k)} \quad \cdots \quad \frac{\partial E}{\partial a_l(k)} \quad \cdots \quad \frac{\partial E}{\partial a_{L-1}(k)} \quad \frac{\partial E}{\partial a_L(k)} \right]^T, \quad (4.30)$$

$$\frac{\partial E}{\partial \mathbf{B}(k)} = \left[ \frac{\partial E}{\partial b_1(k)} \quad \frac{\partial E}{\partial b_2(k)} \quad \cdots \quad \frac{\partial E}{\partial b_l(k)} \quad \cdots \quad \frac{\partial E}{\partial b_{L-1}(k)} \quad \frac{\partial E}{\partial b_L(k)} \right]^T, \quad (4.31)$$

$$\frac{\partial E}{\partial \mathbf{C}(k)} = \left[ \frac{\partial E}{\partial c_1(k)} \quad \frac{\partial E}{\partial c_2(k)} \quad \cdots \quad \frac{\partial E}{\partial c_m(k)} \quad \cdots \quad \frac{\partial E}{\partial c_{M-1}(k)} \quad \frac{\partial E}{\partial c_M(k)} \right]^T, \quad (4.32)$$

$$\frac{\partial E}{\partial \mathbf{D}(k)} = \left[ \frac{\partial E}{\partial d_1(k)} \quad \frac{\partial E}{\partial d_2(k)} \quad \cdots \quad \frac{\partial E}{\partial d_j(k)} \quad \cdots \quad \frac{\partial E}{\partial d_{J-1}(k)} \quad \frac{\partial E}{\partial d_J(k)} \right]^T, \quad (4.33)$$

$$\frac{\partial E}{\partial \mathbf{K}(k)} = \left[ \frac{\partial E}{\partial K_1(k)} \quad \frac{\partial E}{\partial K_2(k)} \quad \cdots \quad \frac{\partial E}{\partial K_i(k)} \quad \cdots \quad \frac{\partial E}{\partial K_{I-1}(k)} \quad \frac{\partial E}{\partial K_I(k)} \right]^T. \quad (4.34)$$

Los cambios incrementales de cada coeficiente son simplemente el negativo de sus gradientes,

$$\Delta \mathbf{W}(k) = -\frac{\partial E}{\partial \mathbf{W}(k)}, \quad (4.35)$$

$$\Delta \mathbf{A}(k) = -\frac{\partial E}{\partial \mathbf{A}(k)}, \quad (4.36)$$

$$\Delta \mathbf{B}(k) = -\frac{\partial E}{\partial \mathbf{B}(k)}, \quad (4.37)$$

$$\Delta \mathbf{C}(k) = -\frac{\partial E}{\partial \mathbf{C}(k)}, \quad (4.38)$$

$$\Delta \mathbf{D}(k) = -\frac{\partial E}{\partial \mathbf{D}(k)}, \quad (4.39)$$

$$\Delta \mathbf{K}(k) = -\frac{\partial E}{\partial \mathbf{K}(k)}. \quad (4.40)$$

Así cada coeficiente de la red *wavenet* y ganancias del control *wavelet* son actualizados en concordancia con las siguientes reglas:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu_W \Delta \mathbf{W}(k), \quad (4.41)$$

$$\mathbf{A}(k+1) = \mathbf{A}(k) + \mu_A \Delta \mathbf{A}(k), \quad (4.42)$$

$$\mathbf{B}(k+1) = \mathbf{B}(k) + \mu_B \Delta \mathbf{B}(k), \quad (4.43)$$

$$\mathbf{C}(k+1) = \mathbf{C}(k) + \mu_C \Delta \mathbf{C}(k), \quad (4.44)$$

$$\mathbf{D}(k+1) = \mathbf{D}(k) + \mu_D \Delta \mathbf{D}(k), \quad (4.45)$$

$$\mathbf{K}(k+1) = \mathbf{K}(k) + \mu_K \Delta \mathbf{K}(k), \quad (4.46)$$

donde  $\mu \in \mathbb{R}$  es el coeficiente de velocidad de aprendizaje, para cada uno de los parámetros.

A continuación se calculan los gradientes requeridos por (4.29) - (4.34).

### Cálculo del gradiente $\frac{\partial E}{\partial w_l(k)}$

Derivando (4.27) con respecto a  $w_l(k)$ , se obtiene

$$\frac{\partial E}{\partial w_l(k)} = \frac{\partial}{\partial w_l(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.47)$$

$$\frac{\partial E}{\partial w_l(k)} = \frac{\partial}{\partial w_l(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.48)$$

$$= e_n(k) \frac{\partial}{\partial w_l(k)} e_n(k), \quad (4.49)$$

$$= e_n(k) \frac{\partial}{\partial w_l(k)} [y(k) - \hat{y}(k)], \quad (4.50)$$

$$= -e_n(k) \frac{\partial}{\partial w_l(k)} \hat{y}(k), \quad (4.51)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial w_l(k)} = -e_n(k) \frac{\partial}{\partial w_l(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.52)$$

ya que sólo  $\mathbf{Z}(k)$  depende de  $w_l(k)$ , entonces la ecuación se simplifica,

$$\frac{\partial E}{\partial w_l(k)} = -e_n(k) \mathbf{C}^T(k) \frac{\partial}{\partial w_l(k)} [\mathbf{Z}(k)] u(k), \quad (4.53)$$

si definimos

$$\Psi_l(\tau) = \frac{\partial}{\partial w_l(k)} \mathbf{Z}(k), \quad (4.54)$$

$$= \left[ \frac{\partial z(k)}{\partial w_l(k)} \quad \frac{\partial z(k-1)}{\partial w_l(k)} \quad \dots \quad \frac{\partial z(k-m)}{\partial w_l(k)} \quad \dots \quad \frac{\partial z(k-M)}{\partial w_l(k)} \right]^T, \quad (4.55)$$

sea

$$\frac{\partial z(k)}{\partial w_l(k)} = \frac{\partial}{\partial w_l(k)} [\Psi^T(k) \mathbf{W}(k)], \quad (4.56)$$

$$= \psi_l(\tau) \quad (4.57)$$

entonces

$$\Psi_l(\tau) = [\psi_l(\tau) \ \psi_l(\tau - 1) \ \cdots \ \psi_l(\tau - m) \ \cdots \ \psi_l(\tau - M)]^T, \quad (4.58)$$

por lo tanto la ecuación del gradiente para cada  $w_l$  es

$$\frac{\partial E}{\partial w_l(k)} = -e_n(k) \mathbf{C}^T \Psi_l(\tau) u(k). \quad (4.59)$$

### Cálculo del gradiente $\frac{\partial E}{\partial b_l}$

Derivando (4.27) con respecto a  $b_l(k)$ , se obtiene

$$\frac{\partial E}{\partial b_l(k)} = \frac{\partial}{\partial b_l(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.60)$$

$$\frac{\partial E}{\partial b_l(k)} = \frac{\partial}{\partial b_l(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.61)$$

$$= e_n(k) \frac{\partial}{\partial b_l(k)} e_n(k), \quad (4.62)$$

$$= e_n(k) \frac{\partial}{\partial b_l(k)} [y(k) - \hat{y}(k)], \quad (4.63)$$

$$= -e_n(k) \frac{\partial}{\partial b_l(k)} \hat{y}(k), \quad (4.64)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial b_l(k)} = -e_n(k) \frac{\partial}{\partial b_l(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.65)$$

ya que sólo  $\mathbf{Z}(k)$  depende de  $b_l(k)$ , entonces la ecuación se simplifica,

$$\frac{\partial E}{\partial b_l(k)} = -e_n(k) \mathbf{C}^T(k) \frac{\partial}{\partial b_l(k)} [\mathbf{Z}(k)] u(k), \quad (4.66)$$

sea

$$\frac{\partial \mathbf{Z}(k)}{\partial b_l(k)} = \left[ \frac{\partial z(k)}{\partial b_l(k)} \ \frac{\partial z(k-1)}{\partial b_l(k)} \ \cdots \ \frac{\partial z(k-m)}{\partial b_l(k)} \ \cdots \ \frac{\partial z(k-M)}{\partial b_l(k)} \right]^T, \quad (4.67)$$

si

$$\frac{\partial z(k)}{\partial b_l(k)} = \frac{\partial}{\partial b_l(k)} \mathbf{\Psi}^T(k) \mathbf{W}(k), \quad (4.68)$$

$$= \left[ \frac{\partial \psi_1(\tau)}{\partial b_l(k)} \quad \frac{\partial \psi_2(\tau)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_L(\tau)}{\partial b_l(k)} \right] \mathbf{W}(k), \quad (4.69)$$

$$= \frac{\partial \psi_l(\tau)}{\partial b_l(k)} w_l(k), \quad (4.70)$$

si definimos

$$\mathbf{\Psi}_{b_l}(\tau) = \left[ \frac{\partial \psi_l(\tau)}{\partial b_l(k)} \quad \frac{\partial \psi_l(\tau-1)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau-m)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau-M)}{\partial b_l(k)} \right]^T, \quad (4.71)$$

por lo tanto la ecuación del gradiente para cada  $b_l$  es

$$\frac{\partial E}{\partial b_l(k)} = -e_n(k) \mathbf{C}^T \mathbf{\Psi}_{b_l}(\tau) w_l(k) u(k). \quad (4.72)$$

### Cálculo del gradiente $\frac{\partial E}{\partial a_l}$

Derivando (4.27) con respecto a  $a_l(k)$ , se obtiene

$$\frac{\partial E}{\partial a_l(k)} = \frac{\partial}{\partial a_l(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.73)$$

$$\frac{\partial E}{\partial a_l(k)} = \frac{\partial}{\partial a_l(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.74)$$

$$= e_n(k) \frac{\partial}{\partial a_l(k)} e_n(k), \quad (4.75)$$

$$= e_n(k) \frac{\partial}{\partial a_l(k)} [y(k) - \hat{y}(k)], \quad (4.76)$$

$$= -e_n(k) \frac{\partial}{\partial a_l(k)} \hat{y}(k), \quad (4.77)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial a_l(k)} = -e_n(k) \frac{\partial}{\partial a_l(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.78)$$

ya que sólo  $\mathbf{Z}(k)$  depende de  $a_l(k)$ , entonces la ecuación se simplifica,

$$\frac{\partial E}{\partial a_l(k)} = -e_n(k) \mathbf{C}^T(k) \frac{\partial}{\partial a_l(k)} [\mathbf{Z}(k)] u(k), \quad (4.79)$$

sea

$$\frac{\partial \mathbf{Z}(k)}{\partial a_l(k)} = \left[ \frac{\partial z(k)}{\partial a_l(k)} \quad \frac{\partial z(k-1)}{\partial a_l(k)} \quad \dots \quad \frac{\partial z(k-m)}{\partial a_l(k)} \quad \dots \quad \frac{\partial z(k-M)}{\partial a_l(k)} \right]^T, \quad (4.80)$$

si

$$\frac{\partial z(k)}{\partial a_l(k)} = \frac{\partial}{\partial a_l(k)} \boldsymbol{\Psi}^T(k) \mathbf{W}(k), \quad (4.81)$$

$$= \left[ \frac{\partial \psi_1(\tau)}{\partial a_l(k)} \quad \frac{\partial \psi_2(\tau)}{\partial a_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau)}{\partial a_l(k)} \quad \dots \quad \frac{\partial \psi_L(\tau)}{\partial a_l(k)} \right] \mathbf{W}(k), \quad (4.82)$$

$$= \frac{\partial \psi_l(\tau)}{\partial a_l(k)} w_l(k), \quad (4.83)$$

como

$$\frac{\partial \psi_l(\tau)}{\partial a_l(k)} = \tau_l \frac{\partial \psi_l(\tau)}{\partial b_l(k)}, \quad (4.84)$$

entonces

$$\frac{\partial z(k)}{\partial a_l(k)} = \tau_l \frac{\partial \psi_l(\tau)}{\partial b_l(k)} w_l(k), \quad (4.85)$$

y

$$\frac{\partial \mathbf{Z}(k)}{\partial a_l(k)} = \tau_l \left[ \frac{\partial \psi_l(\tau)}{\partial b_l(k)} \quad \frac{\partial \psi_l(\tau-1)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau-m)}{\partial b_l(k)} \quad \dots \quad \frac{\partial \psi_l(\tau-M)}{\partial b_l(k)} \right] w_l, \quad (4.86)$$

$$= \tau_l \boldsymbol{\Psi}_{b_l} w_l, \quad (4.87)$$

por lo tanto la ecuación del gradiente para cada  $a_l$  es

$$\frac{\partial E}{\partial a_l(k)} = -\tau_l e_n(k) \mathbf{C}^T \boldsymbol{\Psi}_{b_l}(\tau) w_l(k) u(k), \quad (4.88)$$

$$= \tau_l \frac{\partial E}{\partial b_l(k)}, \quad (4.89)$$

## Cálculo del gradiente $\frac{\partial E}{\partial d_j}$

Derivando (4.27) con respecto a  $d_j(k)$ , se obtiene

$$\frac{\partial E}{\partial d_j(k)} = \frac{\partial}{\partial d_j(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.90)$$

$$= \frac{\partial}{\partial d_j(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.91)$$

$$= e_n(k) \frac{\partial}{\partial d_j(k)} [y(k) - \hat{y}(k)], \quad (4.92)$$

$$= -e_n(k) \frac{\partial}{\partial d_j(k)} \hat{y}(k), \quad (4.93)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial d_j(k)} = -e_n(k) \frac{\partial}{\partial d_j(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.94)$$

ya que sólo el primer término del lado derecho de (4.94) depende de  $d_j(k)$ , entonces la ecuación se simplifica, y por lo tanto la ecuación del gradiente para cada  $d_j$  es (4.96)

$$\frac{\partial E}{\partial d_j(k)} = -e_n(k) \frac{\partial}{\partial d_j(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) \right], \quad (4.95)$$

$$= -e_n(k) \hat{y}(k - j) v(k), \quad (4.96)$$

### Cálculo del gradiente $\frac{\partial E}{\partial c_m}$

Derivando (4.27) con respecto a  $c_m(k)$ , se obtiene

$$\frac{\partial E}{\partial c_m(k)} = \frac{\partial}{\partial c_m(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.97)$$

$$= \frac{\partial}{\partial c_m(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.98)$$

$$= e_n(k) \frac{\partial}{\partial c_m(k)} [y(k) - \hat{y}(k)], \quad (4.99)$$

$$= -e_n(k) \frac{\partial}{\partial c_m(k)} \hat{y}(k), \quad (4.100)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial c_m(k)} = -e_n(k) \frac{\partial}{\partial c_m(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.101)$$

ya que sólo el segundo término del lado derecho de (4.101) depende de  $c_m(k)$ , entonces la ecuación se simplifica, y por lo tanto la ecuación del gradiente para cada  $c_m$  es (4.103)

$$\frac{\partial E}{\partial c_m(k)} = -e_n(k) \frac{\partial}{\partial c_m(k)} [\mathbf{C}^T(k) \mathbf{Z}(k) u(k)], \quad (4.102)$$

$$= -e_n(k) z(k-m) u(k), \quad (4.103)$$

### Calculando del gradiente $\frac{\partial E}{\partial K_i}$

Derivando (4.27) con respecto a  $K_i(k)$ , se obtiene

$$\frac{\partial E}{\partial K_i(k)} = \frac{\partial}{\partial K_i(k)} \left[ \frac{1}{2} E_n^T E_n \right], \quad (4.104)$$

$$= \frac{\partial}{\partial K_i(k)} \left[ \frac{1}{2} e_n^2(k) \right], \quad (4.105)$$

$$= e_n(k) \frac{\partial}{\partial c_m(k)} [y(k) - \hat{y}(k)], \quad (4.106)$$

$$= -e_n(k) \frac{\partial}{\partial K_i(k)} \hat{y}(k), \quad (4.107)$$

sustituyendo (4.23) en  $\hat{y}$ , resultando

$$\frac{\partial E}{\partial K_i(k)} = -e_n(k) \frac{\partial}{\partial K_i(k)} \left[ \mathbf{D}^T(k) \hat{\mathbf{Y}}(k) v(k) + \mathbf{C}^T(k) \mathbf{Z}(k) u(k) \right], \quad (4.108)$$

ya que sólo  $u(k)$  depende de  $K_i(k)$ , entonces la ecuación se simplifica,

$$\frac{\partial E}{\partial K_i(k)} = -e_n(k) \frac{\partial}{\partial K_i(k)} [\mathbf{C}^T(k) \mathbf{Z}(k) u(k)], \quad (4.109)$$

$$= -e_n(k) \mathbf{C}^T(k) \mathbf{Z}(k) \frac{\partial}{\partial K_i(k)} u(k), \quad (4.110)$$

sustituyendo (3.10) y (4.12)

$$\frac{\partial E}{\partial K_i(k)} = -e_n(k) \hat{\Gamma} [Y(k), U(k), \Theta_\Gamma] \frac{\partial}{\partial K_i(k)} [\mathbf{K}^T(k) \mathbf{E}_m(k)], \quad (4.111)$$

si



$$\frac{\partial u(k)}{\partial K_i(k)} = \frac{\partial}{\partial K_i(k)} [\mathbf{K}^T(k)\mathbf{E}_m(k)], \quad (4.112)$$

$$\frac{\partial u(k)}{\partial K_i(k)} = e_i(k), \quad (4.113)$$

por lo tanto la ecuación del gradiente para cada  $K_i$  es

$$\frac{\partial E}{\partial K_i(k)} = -e_n(k)\hat{\Gamma}[Y(k), U(k), \Theta_\Gamma]e_i(k), \quad (4.114)$$

En la Tabla 4.1 se muestra algunas *wavelet* madres con sus derivadas con respecto a  $b$ , empleadas para calcular (4.71).

Nombre	$\psi(\tau)$	$\frac{\partial \psi(\tau)}{\partial b}$
Morlet	$\cos(\omega_0\tau)\exp(-0.5\tau^2)$	$\frac{1}{a}[\omega_0 \sin(\omega_0\tau)\exp(-0.5\tau^2) + \tau\psi(\tau)]$
RASP1	$\frac{\tau}{(\tau^2+1)^2}$	$\frac{3\tau^2-1}{a(\tau^2+1)^3}$

Tabla 4.1: Algunas *Wavelet* madres y su respectiva derivada con respecto a  $b$ .

## 4.4. Resultados de simulación

Para validar los algoritmos, éstos son aplicados al siguiente sistema lineal estable descrito por la ecuación (3.44) con los parámetros de la red neuronal mostrados en la Tabla 4.2 y con valores iniciales mostrados en la Tabla 4.3.

Neuronas	5
<i>Wavelet</i> madre	<i>wavelet</i> Morlet
Coefficientes de alimentación hacia adelante $c$	2
Coefficientes de alimentación hacia atrás $d$	2

Tabla 4.2: Parámetros de la red neuronal *wavelet*.

El sistema es simulado con una red neuronal *wavenet* sin entrenamiento previo a ser utilizado con el Esquema de la Figura 4.1 para sintonizar las ganancias del control PID *wavelet*, pero si se tiene un periodo de aprendizaje  $0 \leq t \leq 120$  segundos para identificar al sistema.

<b>W</b>	[-0.5 -0.5 -0.5 -0.5 -0.5]	$\Delta \mathbf{W}$	0.2
<b>A</b>	[10 10 10 10 10]	$\Delta \mathbf{A}$	0.2
<b>B</b>	[0 30 60 90 120]	$\Delta \mathbf{B}$	0.2
<b>C</b>	[0.1 0.1]	$\Delta \mathbf{C}$	0.2
<b>D</b>	[0.1 0.1]	$\Delta \mathbf{D}$	0.2
<b>K</b>	[0.1 0.5 0.5 0.1]	$\Delta \mathbf{K}$	0.05

Tabla 4.3: Valores iniciales de los parámetros de la red neuronal *wavelet* y las ganancias del control PID *wavenet*.

En la Figura 4.2, se observa como después del periodo de aprendizaje se obtiene una respuesta del PID *wavenet* aceptable con un sobre-impulso no mayor de 0.2.

En la Figura 4.3, se observa como la señal de control es mucho menor que la que se generaba con el control PID *wavelet* y aun así se tiene una rápida respuesta como se ve en la Figura 4.2, y esto se debe a la sintonización de las ganancias realizada por el algoritmo de sintonización.

En la descomposición de la señal del error mostrada en la Figura 4.4, es muy similar a la descomposición del error mostrada en la Figura 3.14 generada por el PID *wavelet* sin la autosintonización.

En las Figuras 4.5, 4.6 y 4.6, se muestran los comportamientos de los parámetros de la red neuronal *wavenet* y del filtro IIR, así como el comportamiento de las ganancias del PID *wavenet* que se están sintonizando durante el periodo de entrenamiento. Observándose que con poco periodo de aprendizaje y sin previo entrenamiento de la red neuronal *wavenet*, se logra tener una buena respuesta en la salida del sistema como se muestra en la Figura 4.2.

Los valores finales son mostrados en Tabla 4.4:

<b>W</b>	[-0.426 -0.47 -0.54 -0.56 -1.25]
<b>A</b>	[10.057 10.014 10.004 9.98 9.962]
<b>B</b>	[0.003 30.019 60.065 90.15 120.018]
<b>C</b>	[0.775 0.8]
<b>D</b>	[0.0973 0.0948]
<b>K</b>	[0.615 0.45 0.448 0.2]

Tabla 4.4: Valores finales de los parámetros de la red *wavenet* y las ganancias del control PID *wavenet*.

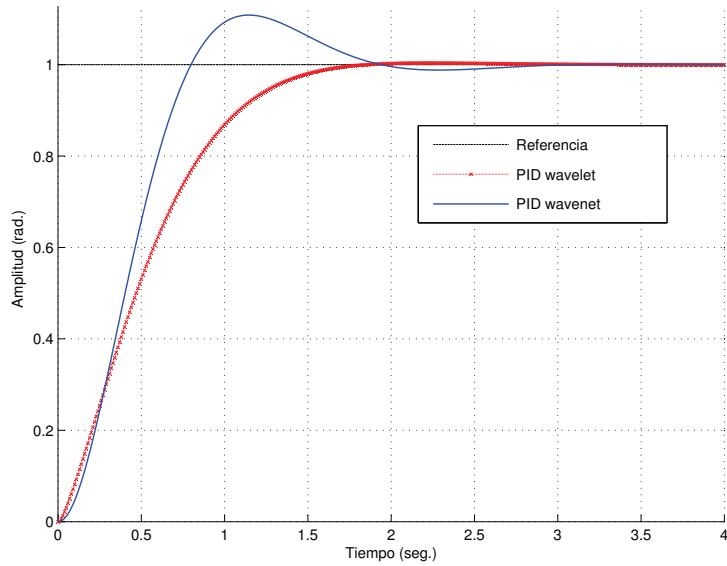


Figura 4.2: Señal de salida del sistema con un control PID *wavenet* y con un control PID *wavelet*.

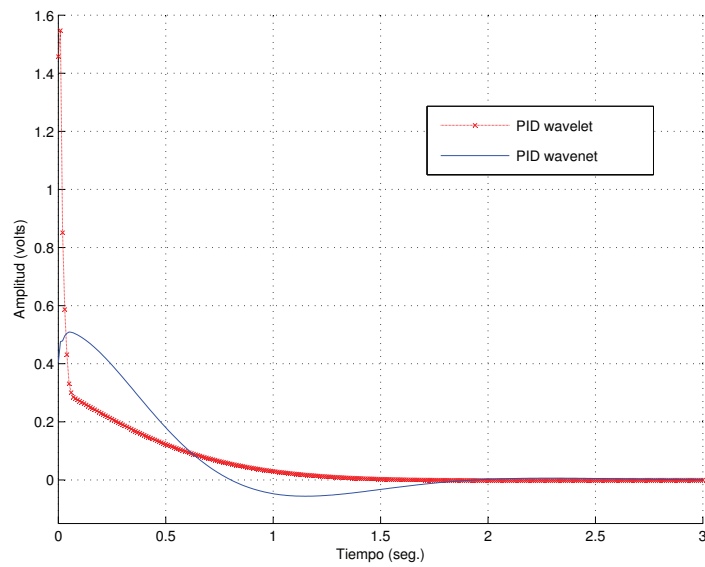


Figura 4.3: Señal de control generada por el control PID *wavenet* y el PID *wavelet*.

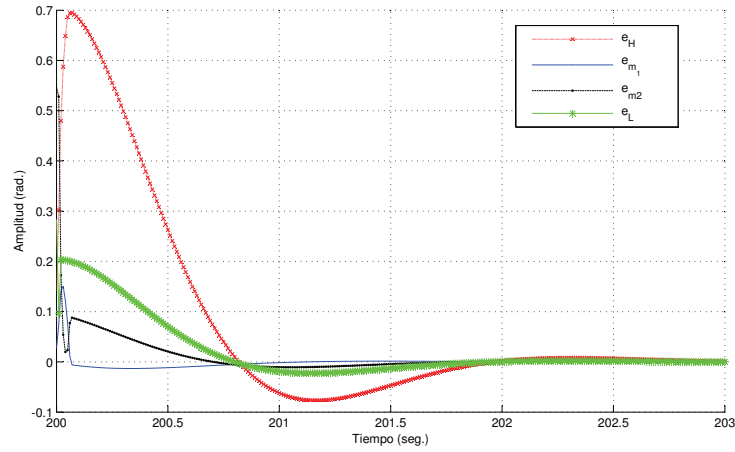


Figura 4.4: Descomposición de la señal del error del sistema con un control PID *wavenet*.

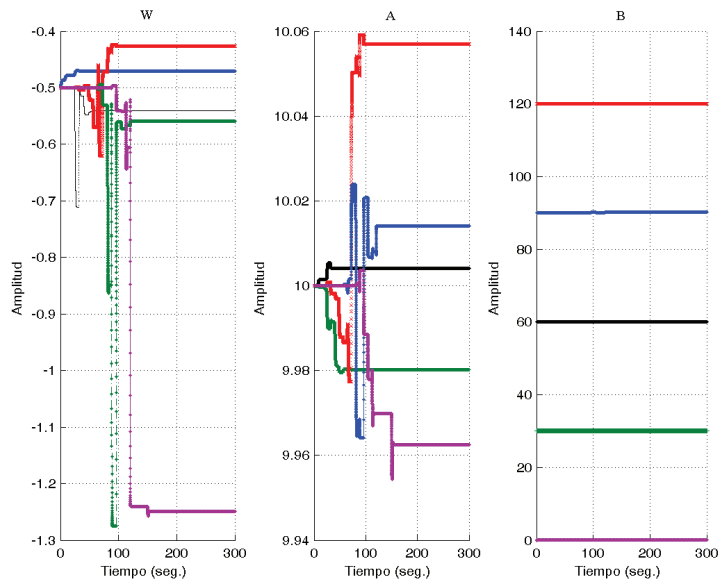


Figura 4.5: Comportamiento de los parámetros de la red neuronal *wavenet*, durante la simulación.

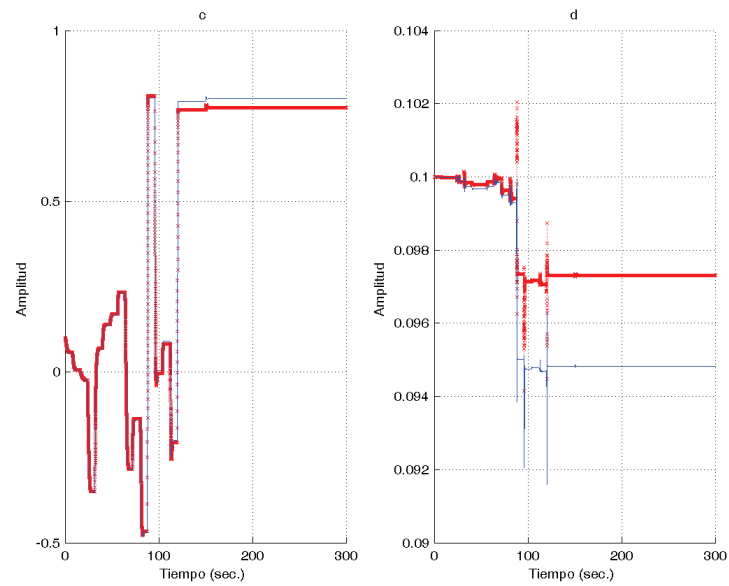


Figura 4.6: Comportamiento de los parámetros del filtro IIR, durante la simulación.

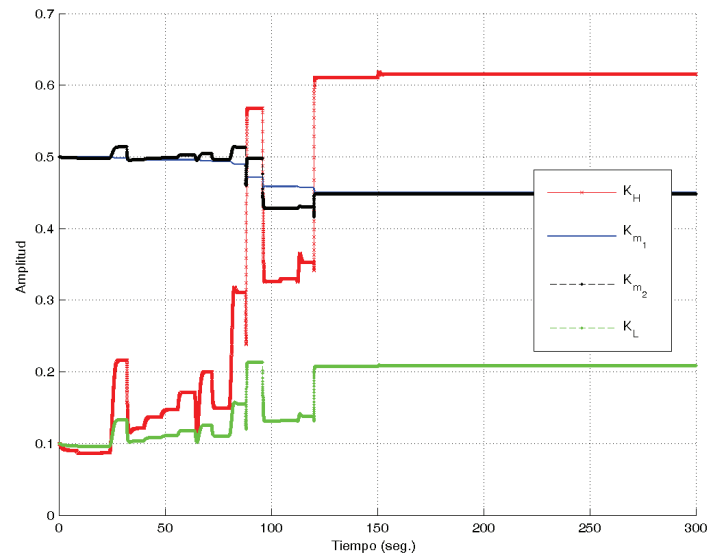


Figura 4.7: Comportamiento de las ganancias del control PID *wavenet*, durante la simulación.

## 4.5. Comentarios

Las redes neuronales *wavelets* adaptables (*wavenets*) combinan las ventajas que ofrece el procesamiento con redes neuronales y la representación *wavelet*; y la estructura y parámetros de los modelos *wavenets* son fáciles de obtener, de esta manera, los diferentes PID *wavelet* y PID *wavenet* presentados en esta tesis ofrecen un método eficiente en sistemas de control que usualmente poseen complejidad, no linealidades, incertidumbres y retardos.

# Capítulo 5

## Resultados de laboratorio

En este capítulo, se muestran los resultados obtenidos en la implementación del control PID *wavelet*, sobre un sistema de control de posición y un robot planar.

### 5.1. Implementación del control PID *wavelet*

#### 5.1.1. Implementación sobre un sistema de control de posición

Los siguientes resultados son obtenidos en la implementación del control PID *wavelet*, en un sistema de control de posición, mostrado en la Figura 5.1, la función de transferencia del voltaje  $v$  para una posición  $x$  puede ser modelado como:

$$G(s) = \frac{X(s)}{V(s)} = \frac{b}{s(Js + c)}. \quad (5.1)$$

donde:  $b$ ,  $J$ ,  $c$  son la constante del par, la inercia total del motor y la carga, y el coeficiente de fricción viscosa, respectivamente. El objetivo del control es rotar al motor de un punto A a un punto B, como se muestra en la Figura 5.2. Es importante mencionar que los parámetros de la planta no son requeridos en la sintonización de los controladores propuestos.

La prueba se realizó con una plataforma experimental sobre una PC con tarjeta adquisición de datos de *National Instrument* PCI-6024E y un servomotor. El controlador está programado en un entorno de programación en Simulink, sobre una PC Intel pentium 4 de 2.0 GHz y 1 GB de memoria RAM, la cual contiene la tarjeta de adquisición de datos, para leer la posición, con una frecuencia de muestreo de 100 Hz, y también es empleada para calcular la ley de control, como se muestra en la Figura 5.3

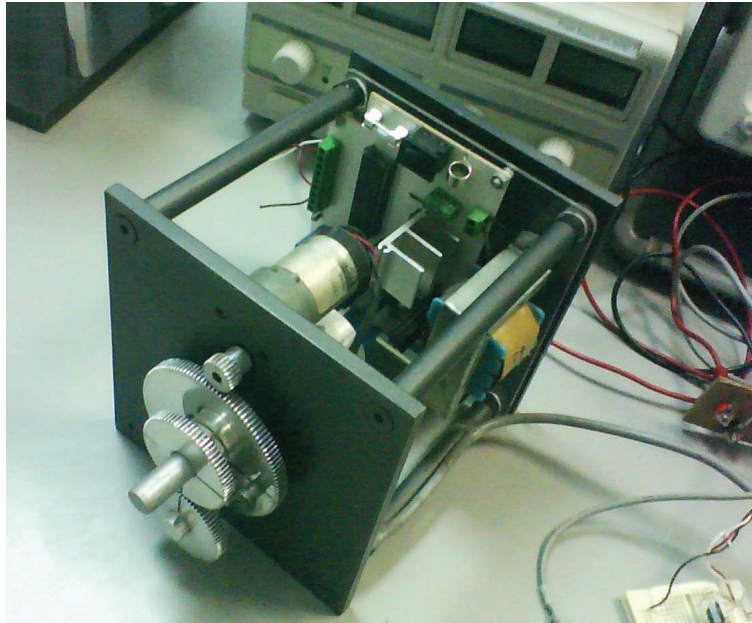


Figura 5.1: Sistema de control de movimiento.

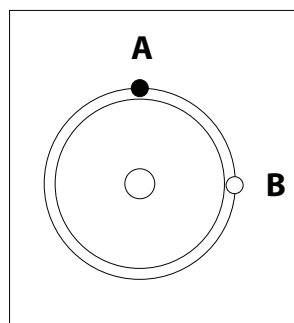


Figura 5.2: Objetivo de control.



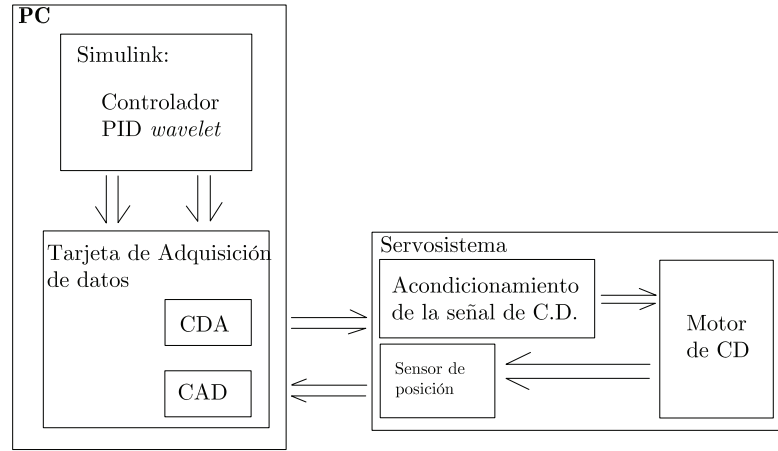


Figura 5.3: Diagrama a bloques del sistema de control de movimiento.

El algoritmo del PID *wavelet* fue realizado en Simulink, empleando un esquema de codificación de sub-banda usando los coeficientes Daubechies de orden 2, con un nivel de descomposición  $N = 3$  y los parámetros del PID *wavelet* son mostrados en la Tabla 5.1, donde las ganancias del PID clásico son sintonizadas de manera herística y para el PID *wavelet* se sintonizarón de a prueba y error.

Ganancias del PID clásico	$K_P$	$K_D$	$K_I$	
	10	1.6	6	
Ganancias del PID <i>wavelet</i>	$k_H$	$K_{m_1}$	$K_{m_2}$	$K_L$
	1.6	18	1.6	0

Tabla 5.1: Valores de las ganancias para el control PID clásico y el PID *wavelet* para el sistema para la plataforma mostrada en la Figura 5.1.

Las gráficas obtenidas, son mostradas en las Figuras 5.4, 5.5, 5.6 y 5.7, donde se puede observar la característica particular del control PID *wavelet*, al generar una señal de control suave e impidiendo generar esfuerzos y desgaste en el motor. Además de tener un mejor comportamiento en la salida de la planta que cuando es controlada con un PID clásico, ya que al exigirle tal repuesta la señal de control que generaría contendría alta frecuencia dañando al actuador, y algunos casos no se podría permitir que esto ocurriera, tal es el caso de esta planta, por que el motor sólo vibraría sin generar ningún movimiento.

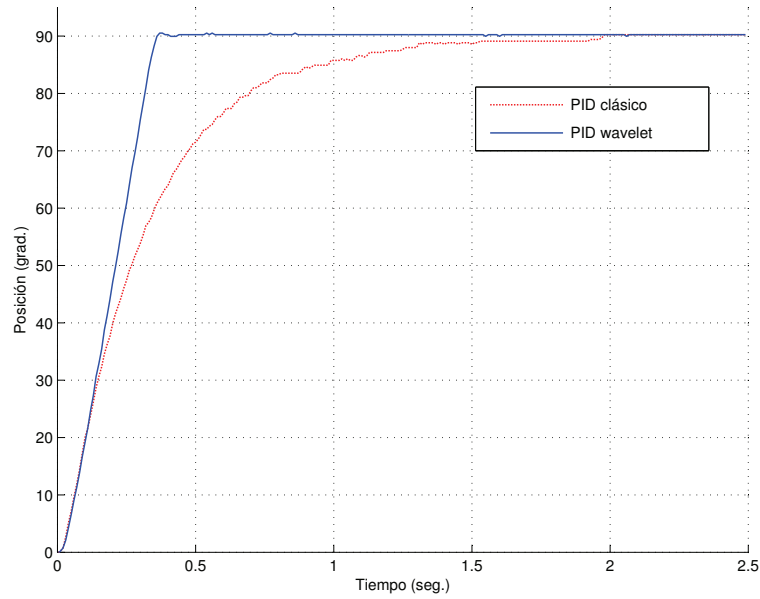


Figura 5.4: Resultados de la implementación para la señal de posición del sistema de posición.

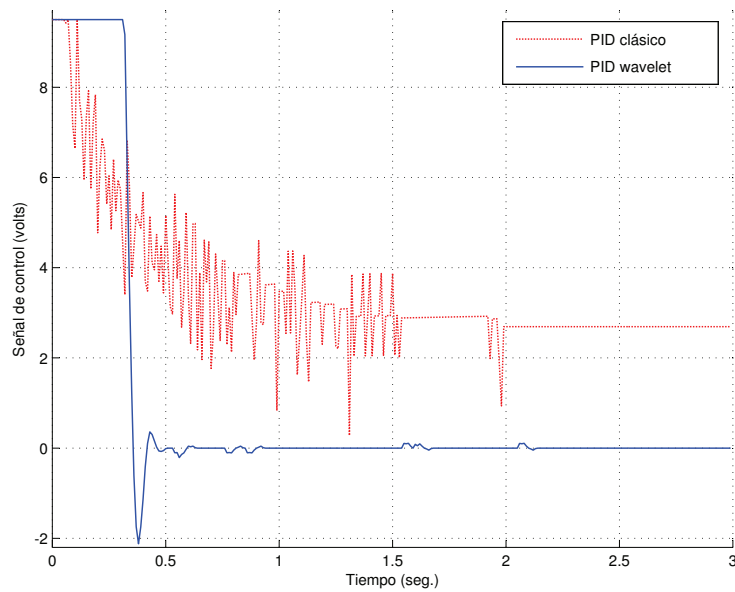


Figura 5.5: Resultados de la implementación para la señal de control del sistema de posición.

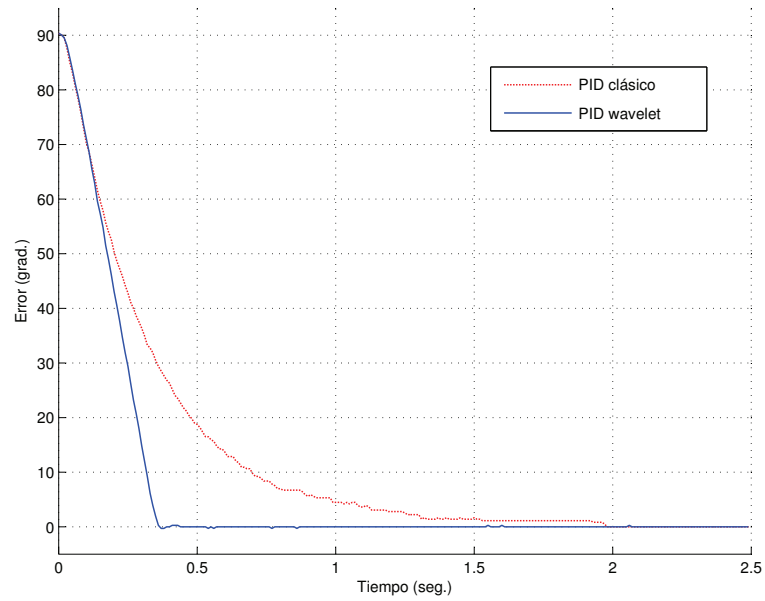


Figura 5.6: Resultados de la implementación para la señal de error en la posición del sistema de posición.

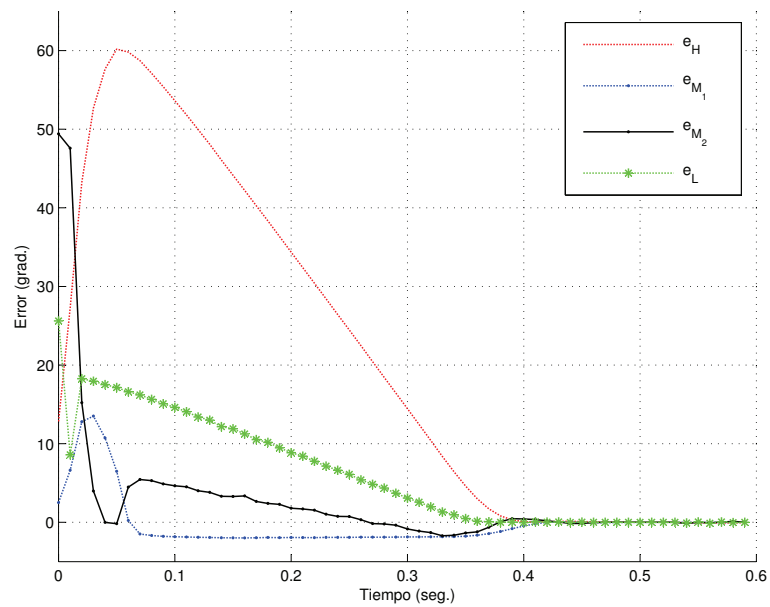


Figura 5.7: Resultados de la implementación para la señal de error a un nivel de descomposición  $N = 3$ .

### 5.1.2. Implementación sobre un robot planar de dos grados de libertad

Esta aplicación se realiza sobre un robot planar de dos grados de libertad, mostrada en la Figura 5.8, elaborado por alumnos de la Licenciatura de Sistemas Computacionales en la especialidad de mecatrónica que se ofrece en la UAEH. Donde sus actuadores son controlados por medio de un PID *wavelet*, ambos actuadores son llevados a su correspondiente referencia al mismo tiempo para llevar el extremo del eslabón 2 (efector final) a una referencia dada, los resultados son comparados con un control PID clásico. De la misma forma que la implementación anterior, aquí se emplea la misma tarjeta de adquisición de datos *National Instrument PCI-6024E*, sobre la misma PC, y con los algoritmos son programados en *Simulink*. Las ganancias empleadas en el control PID clásico son mostradas en la Tabla 5.2 y las ganancias empleadas en el control PID *wavelet* son mostradas en la Tabla 5.3.

Ganancias del PID clásico	$k_P$	$K_D$	$K_I$
Ganancias para el control $u_1$	3.5	2.3	0.5
Ganancias para el control $u_2$	9.5	2.3	0.5

Tabla 5.2: Valores de las ganancias para el PID clásico para el robot planar.

Ganancias del PID <i>wavelet</i>	$k_H$	$K_{M_1}$	$K_{M_2}$	$K_L$
Ganancias para el control $u_1$	4.7	4.8	0.8	0
Ganancias para el control $u_2$	12	12	0.8	0

Tabla 5.3: Valores de las ganancias para el PID *wavelet* para el robot planar.

Los resultados son mostrados en la Figuras 5.9, 5.10, 5.11 y 5.12.

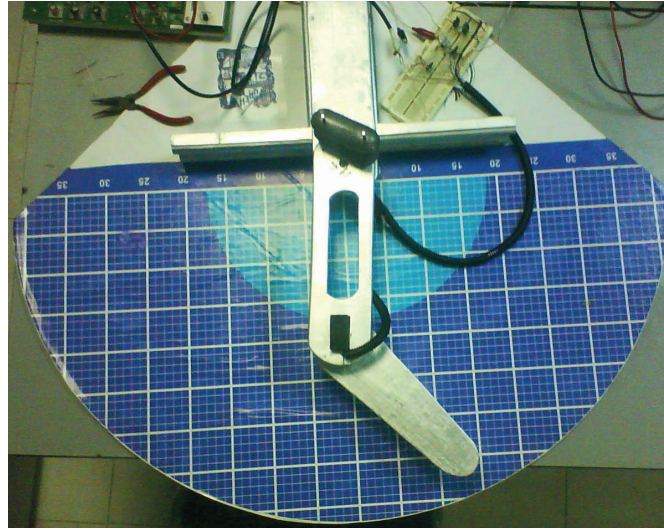


Figura 5.8: Robot planar de dos grados de libertad.

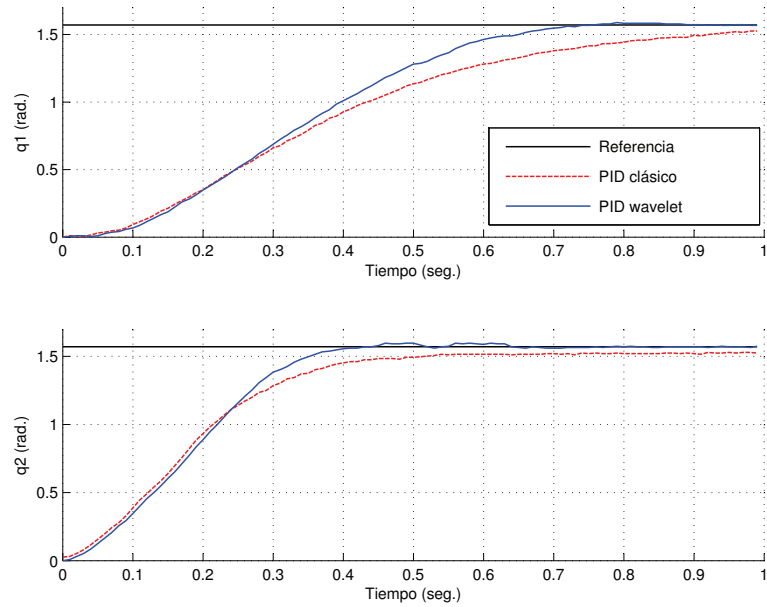


Figura 5.9: Resultados de la posición para el actuador 1 y 2.

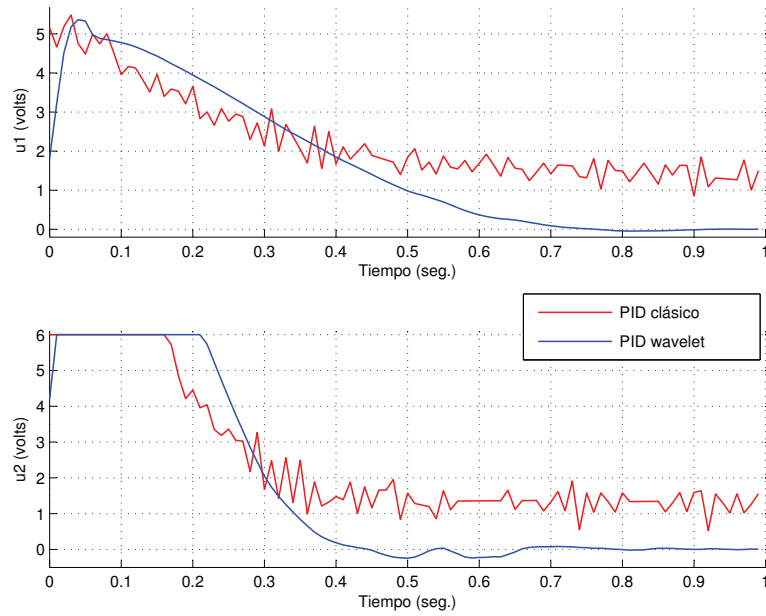


Figura 5.10: Resultados de la señal de control para el actuador 1 y 2.

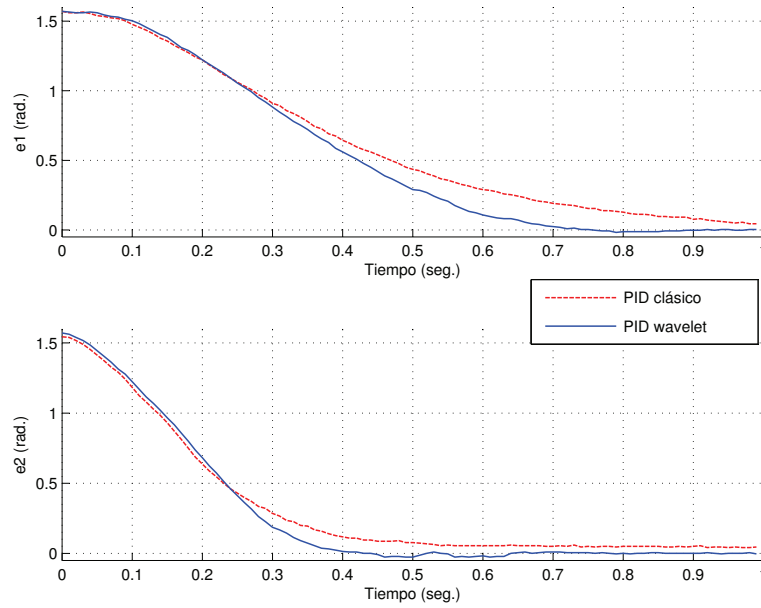


Figura 5.11: Resultados de la señal de error en el actuador 1 y 2.

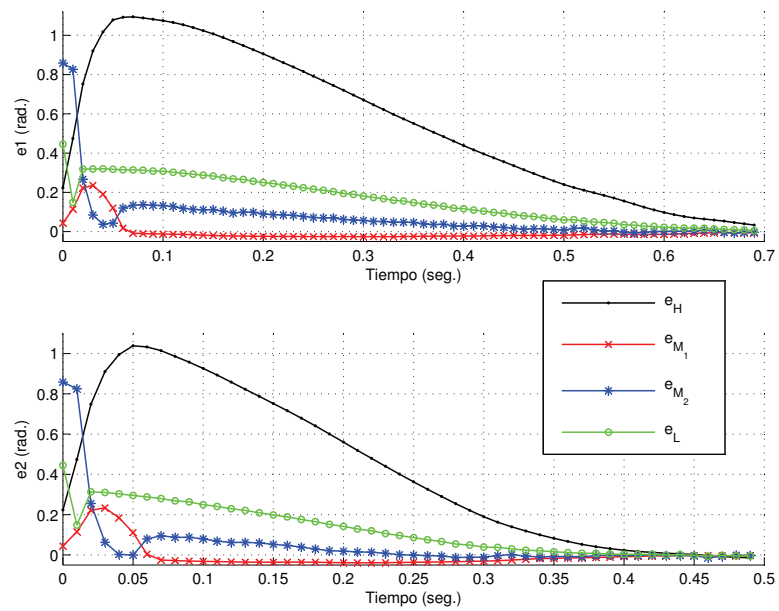


Figura 5.12: Señal de error del actuador 1 y 2 a un nivel de descomposición  $N = 3$ .





# Capítulo 6

## Conclusiones y trabajos futuros

### 6.1. Conclusiones.

En este trabajo de investigación se diseñaron dos variantes del controlador PID que hacen uso de la teoría *wavelet*. El primero es el PIDMR que hace uso de una descomposición de la señal del error, donde dicha descomposición se efectúa mediante la técnica multiresolución y decodificación sub-banda, este controlador se emplea para controlar un motor de CD presentando buenos resultados tanto en simulación como en laboratorio, el segundo controlador llamado *wavenet* hace uso de redes neuronales de base radial con funciones de activación que son *wavelets* hijas, aquí las redes se emplean para dos propósitos (ver Figura 4.1), una para la identificación del sistema y otra para la sintonización de los parámetros del PID *wavenet*, este controlador también se aplicó a un motor de CD obteniéndose resultados satisfactorios, de lo anterior se concluye que el objetivo de la tesis se cumplió satisfactoriamente.

### 6.2. Trabajos futuros.

Algunos de los trabajos futuros que se desprenden de esta tesis son:

- Encontrar un método de sintonización del PIDMR.
- Hacer un análisis riguroso de la convergencia de los algoritmos.
- Hacer un estudio sobre la estabilidad en lazo cerrado para ambos esquemas propuestos en la tesis.



# Glosario

**Actuador.** Dispositivo capaz de convertir energía eléctrica, hidráulica u otro tipo de energía en movimiento.

**Algoritmo.** Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**Aprendizaje no supervisado.** También conocido como aprendizaje auto-supervisado, no se requiere influencia externa para ajustar los pesos de las conexiones entre sus neuronas. No se recibe ninguna información por parte del entorno que indique si la salida generada en respuesta a una determinada entrada es correcta o no.

**Aprendizaje reforzado.** Forma de aprendizaje donde un mapeo entrada-salida se lleva a cabo a través de la continua interacción del entorno para minimizar un índice de desempeño. El uso de un crítico es una característica de este tipo de aprendizaje.

**Aprendizaje supervisado.** Tipo de aprendizaje basado en una respuesta preprogramada para una entrada particular. Es decir, cada par de entrenamiento está compuesto por los valores de entrada a la red y los valores de salida deseados para tales entradas.

**Aprendizaje.** Proceso por el cual los parámetros libres de una red neuronal son adaptados a través de estimulaciones generadas por el entorno en el cual la red neuronal está empotrada.

**Arquitectura.** En redes neuronales consiste en la organización y disposición de las neuronas formando capas y la forma en que se conectan entre sí. En este sentido, los parámetros fundamentales de la red que constituyen su arquitectura son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

**Automatización.** Ciencia que trata de sustituir en un proceso el operador humano por un determinado dispositivo, generalmente electromecánico.

**Codificador (Encoder).** Dispositivo de retroalimentación que convierte un movimiento mecánico en señales eléctricas, las cuales indican la posición de un actuador.

**Control adaptable.** Esquema de control en el cual los parámetros del controlador se ajustan ante cambios de la planta, de manera que el comportamiento en lazo cerrado conserva las características deseadas de diseño. El control adaptable es inherentemente no lineal. Se aplica a sistemas con parámetros que varían lentamente en el tiempo.

**Control.** Proceso mediante el cual un sistema es llevado a parámetros preestablecidos.

**Energía.** Capacidad de realizar un trabajo.

**Entrenamiento.** Se refiere al proceso consciente y planeado de transferir conocimiento, habilidades y capacidades.

**Época.** Presentación completa del conjunto de entrenamiento a la red neuronal.

**Espacio de estado.** Espacio de  $n$  dimensiones, donde cualquier estado de un sistema dado puede ser representado como un punto.

**Estabilidad.** Condición en la cual las variables críticas de un sistema dinámico se mantienen invariables o permanecen dentro de unos límites determinados.

**Estado.** Conjunto mínimo de variables de un sistema dado (variables de estado), tal que, conociendo su valor en un instante dado, permiten conocer la respuesta del sistema ante cualquier señal de entrada o perturbación.

**Función de activación.** Sirve para limitar la amplitud de la neurona de salida. La función de activación limita el rango de amplitud permisible de la señal de salida a algún valor finito.

**Función suave.** La función  $f$  se dice que es suave si sus derivadas parciales de cualquier orden existen y son continuas.

**Función wavelet hija.** Es una función wavelet resultado de dilatar o contraer y trasladar una función wavelet madre.

**Función wavelet madre.** Es una función wavelet que se puede dilatar o contraer y trasladar, generando funciones wavelet hijas.

**Función Wavelet.** Es una función de onda oscilatoria de duración muy corta, también se le llama ondoleta.

**Grados de libertad pasivos.** Grados de libertad que no se encuentran actuados.

**Grados de libertad.** Número de coordenadas que describen de manera única la posición de todos los eslabones de un sistema.

**Hardware.** Se dice de cualquier componente físico relacionado con cierta tecnología.

**Identificación.** Conjunto de métodos para la obtención de modelos matemáticos a partir de datos experimentales de las entradas y salidas de un sistema. Evitándose la obtención de un modelo a partir de complejas leyes.

**Incertidumbre.** En teoría de la información, la certidumbre es el grado cómo la información se puede considerar verdadera, completa y digna de fe. La incertidumbre se origina a partir de elementos de datos falsos o de un equívoco, a partir de datos incompletos o de un contexto ambiguo.

**Manipulador.** Mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

**Modelo matemático.** Es la representación por medio de ecuaciones de la dinámica de un sistema. Es el tipo de modelo más importantes para la ciencia y la tecnología.

**Neurona** Célula nerviosa, elemento fundamental de la arquitectura nerviosa. Es una unidad de procesamiento de información que es fundamental para la operación de una red neuronal.

**Perceptrón.** Dicese de una arquitectura de red neuronal estática multicapa, cuyos nodos ocultos tienen una función de activación suave (sigmoideal), mientras que los nodos de salida poseen una función de activación lineal.

**Perturbación.** Variable no deseada aplicada a un sistema y la cual tiende a afectar adversamente el valor de una variable controlada. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema y es una entrada.

**Punto de equilibrio.** El estado  $\mathbf{x}^*$  se dice que es un punto de equilibrio del sistema  $\dot{\mathbf{x}} = f(\mathbf{x})$  si  $\mathbf{x}(t)$  es igual a  $\mathbf{x}^*$ , para todo  $t \geq t_0$ .

**Redes neuronales dinámicas.** Redes neuronales que poseen dispositivos de memoria en sus conexiones internas, sus salidas son funciones de entradas presentes y pasadas.

**Redes neuronales estáticas.** Redes neuronales cuyas ecuaciones no lineales que las caracterizan no tienen memoria, es decir sus salidas son tan solo funciones de las entradas actuales.

**Redes neuronales.** Procesador distribuido masivamente paralelo constituido de unidades de procesamiento simples, el cual tiene una tendencia natural a almacenar conocimiento experimental y hacerlo viable para su posterior uso.

**Retroalimentación.** En el proceso de control consiste en la reinyección de la salida (o algunas salidas) a la entrada del sistema, con el fin de mantener cierto desempeño.

**Retropropagación.** Algoritmo de entrenamiento basado en el método del descenso del gradiente.

**Robot.** Dispositivo generalmente mecánico, que desempeña tareas automáticamente, ya sea de acuerdo a supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales. Generalmente estas tareas reemplazan, asemejan o extienden el trabajo humano, como ensamblaje en manufactura, manipulación de objetos pesados o peligrosos, trabajo en el espacio, etc.

**Ruido.** Típicamente es una perturbación aleatoria. Señal indeseada presente en un sistema físico.

**Ruidoblanco.** Típicamente es una perturbación aleatoria. Señal indeseada presente en un sistema físico, con una densidad espectral de potencia constante, es decir, la señal de ruido blanco contiene todas las frecuencias y todas ellas contienen la misma frecuencia.

**Sensor.** Dispositivo que convierte un parámetro físico (como temperatura, presión, flujo, velocidad, posición) en una señal eléctrica. En algunos casos se le considera un sinónimo de transductor, pero un verdadero sensor contiene un sistema de acondicionamiento de la señal, de manera que es mucho más sencillo realizar una medición.

**Sinapsis.** Son uniones especializadas mediante las cuales las células del sistema nervioso envían señales de unas a otras y a células no neuronales como las musculares o glandulares. Son la estructura elemental y unidad funcional que regulan la interacción entre neuronas.

**Sistema dinámico.** Sistema cuya respuesta depende de entradas presentes y pasadas (e incluso futuras), dicho sistema puede ser representado mediante ecuaciones diferenciales dinámicas.

**Sistema en tiempo real.** Un sistema de tiempo real STR es un sistema informático en el que es significativo el tiempo en el que se producen las acciones. Las acciones deben realizarse dentro de un intervalo de tiempo determinado.

**Sistema estático.** Sistema cuya respuesta depende únicamente de las entradas presentes.

**Sistema lineal.** Se dice que un sistema es lineal sí cumple con los principios de homogeneidad y superposición.

**Sistema no lineal.** Se dice que un sistema es lineal sí no cumple con los principios de homogeneidad o superposición.

**Sistema subactuado.** Sistema con menos actuadores que grados de libertad.

**Software.** Se dice de todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

**Umbral.** Tiene el efecto de incrementar o decrementar la entrada de la red de la función de activación, dependiendo de si es positivo o negativo, respectivamente.

**Variables de estado.** Conjunto más pequeño de variables que determinan el estado de un sistema dinámico.

**Vector de estado.** Si se necesitan  $n$  variables para describir el estado de un sistema dado, entonces estas  $n$  variables se pueden considerar como los componentes de un vector  $\mathbf{x}$ .





# Bibliografía

- [1] S. Balakrishnan. *Neurocontrol: A Literature*. Mathl. Comput. Modelling, Vol. 23, num. 1-2, pp. 101-117, 1996.
- [2] S. Billings and H. Wei. *A New Class of Wavelet Networks for Nonlinear System Identification*. IEEE Transactions on Neural Networks, Vol. 16, num. 4, pp. 862-874, July 2005.
- [3] C. Borgelt and R. Kruse. **Speeding Up Fuzzy Clustering with Neural Network Techniques**. In *The 12th IEEE International Conference*, Vol. 2, pp. 852-856 2003.
- [4] C. M. Chang and T. S. Lui. *A Wavelet Network Control Method for Disk Drives*. IEEE Transactions on Control Systems Technology, Vol. 14, num. 1, pp. 63-68, Taiwan, China, January 2006.
- [5] C. Chen and W. Lin. **Adaptive Wavelet Sliding Mode Control of Uncertain Multivariable Nonlinear Systems**. In *Proceeding of the American Control Conference*, pp. 180-185, Denver, Colorado, USA, June 2003.
- [6] Y. Cheng, B. Chen, and F. Shiau. *Adaptive Wavelet Network Control Design for Nonlinear Systems*. Proceedings of the National Science Council. ROCA, Vol. 22, num. 6, pp. 783-799, Kobe, Japan, May 1998.
- [7] R. Churchill. *Series de Fourier y Problemas de Contorno*. McGraw-Hill, 1978.
- [8] R. Crochiere, A. Webber, and J. Flanagan. *Digital Coding of Speech in Subbands*. Bell Systems Technical Journal, Vol. 55, num. 8, pp. 1069-1085, October 1976.
- [9] A. Croisier, D. Esteban, and C. Galand. **Perfect Channel Splitting by Use of Interpolation / Decimation / Tree Decomposition Techniques**. In *International Conference on Inform. Sciences and Systems*, pp. 443-446, Patras, Greece, August 1976.

- 
- [10] F. Cuevas, A. Mora, and C. Olvera. *Diseño, Construcción y Control de Ball and Beam*. Tesis de Licenciatura en Ingeniería Electrónica y Telecomunicaciones, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, México, Junio 2008.
- [11] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
- [12] I. Daubechies. *The Wavelet Transform, Time-Frequency Localization and Signal Analysis*. IEEE Trans. Inf. Theory, Vol. 36, num. 5, pp. 961-1005, September 1990.
- [13] I. Daubechies. *Othonormal Bases of Compactly Supported Wavelets*. Commun. on Pure and Appl. Math., Vol. 41, pp. 909-996, November 1988.
- [14] H. Dickhaus and H. Heinrich. **Identification of High Risk Patients In Cardiology By Wavelet Networks**. In *18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 923-924, Amsterdam, Netherlands, USA, 1996.
- [15] C. Domínguez. *Algoritmos Wavenet con Aplicaciones en la Aproximación de Señales*. Tesis de Maestría en Ciencias en Sistemas Computacionales, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, México, Octubre 2008.
- [16] I. Fantoni and R. Lozano. *Non-linear Control for Underactuated Mechanical Systems*. Springer Verlag, London, England, 2002.
- [17] L. Faussett. *Fundamentals of Neural Networks*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1994.
- [18] R. Feris, J. Kentaro, and W. Kruger. **Hierarchical Wavelet Networks for Facial Feature Localization**. In *Proceedings of the 5th IEEE International On Automatic Face and Gesture Recognition*, 2002.
- [19] J. Fourier. *Théorie Analytique de la Chaleur*. Firmin Didot, Paris, France, 1822.
- [20] M. Frazier. *An Introduction to Wavelets Through Linear Algebra*. Springer, 1999.
- [21] Z. Gao, T. Trautzsch, and J. Dawson. **A Stable Self-Tuning Fuzzy Logic Control System for Industrial Temperature Regulation**. In *Industrial Application Conference, 2000. Conference Record of the 2000 IEEE*, Vol. 2, pp. 1232-1240, 2000.
- [22] E. Garcia and V. Aquino. **Chaotic Time Series Approximation Using Iterative Wavelet-Network**. In *Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers*, México, 2006.

- 
- [23] X. Ge, L. Chen, and Y. Liu. Attitude Control of Underactuated Spacecraft through Flywheels Motion Using Genetic Algorithm with Wavelet Approximation. In *Proceeding of the 5th World Congress on Intelligent Control and Automation*, pp. 5466-5469, Hangzhou, P.R. China, June 2004.
- [24] J. Goswami and A. Chan. *Fundamentals of Wavelets - Theory, Algorithms and Applications*. John Wiley And Sons, 1999.
- [25] P. Goupillaud, A. Grossmann, and J. Morlet. *Cycle-Octave and Related Transforms in Seismic Signal Analysis Geop Exploration*. Elsevier Science, Vol. 23, pp. 85-102, 1984.
- [26] A. Grossman and J. Morlet. *Decomposition of Hardy Functions into Squared Integrable Wavelets of Constant Shape*. SIAM J. of Math. Annal., Vol. 15, num. 4, pp. 723-736, July 1984.
- [27] A. Haar. *Zur theorie der Orthogonalen Funktionssysteme*. Math. Annal., Vol. 64, pp. 331-373, 1910.
- [28] F. Ham and I. Kostanic. *Principles of Neurocomputing of Science and Engineering*. McGraw Hill, New York, USA, 2001.
- [29] G. Hans. *Wavelets and Signal Processing : An Application-Based Introduction*. Springer, 2005.
- [30] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1999.
- [31] A. Hojjat. *Wavelets in Intelligent Transportation System*. Wiley, 2004.
- [32] V. Holtto and J. Varso. *Wavelet-Based Disturbance Classification with Robot Arm Application Example*. Proceeding IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 175-179, May 2005.
- [33] C. Hsu, W. Wang, T. Lee, and C. Lin. Longitudinal Control of Vehicle Platoon via Wavelet Neuronal Network. In *International Conference on Systems, Man and Cybernetics*, pp. 3811-3816, The Hague, Netherlands, USA, 2004.
- [34] J. Jang. *ANFIS: Adaptive-Network-Based Fuzzy Inference System*. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, num. 3, pp. 665-685, May-June 1993.
- [35] J. Jang and C. Sun. *Neuro-Fuzzy Modeling and Control*. Proceeding of The IEEE, Vol. 83, num. 3, pp. 378-406, March 1995.

- [36] A. Jarillo. *Diseño e Implementación de Técnicas de Control No lineal Basada en Pasividad Aplicadas a Guiado Háptico*. Tesis de Maestría en Automatización y Control, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, México, Noviembre, 2009.
- [37] H. Karimi, P. Maralami, B. Moshiri, and B. Lohmann. *Haar Wavelet-Based Optimal Control of Time-Varying State-Delayed Systems: A Computational Method*. IEEE International Symposium on Computer-Aided Control Systems Design, pp. 1-6, Munich, Germany, October, 2006.
- [38] F. Keinert. *Wavelets and Multiwavelets*. Chapman and Hall, 2004.
- [39] B. Kuo. *Sistemas de Control Digital*. C.E.C.S.A., 1997.
- [40] A. U. Levin and K. S. Narendra. *Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization*. IEEE Transactions on Neural Networks, Vol. 4, pp. 192-207, 1993.
- [41] A. U. Levin and K. S. Narendra. *Control of Nonlinear Dynamical Systems Using Neural Networks—Part II: Observability, Identification and Control*. IEEE Transactions on Neural Networks, Vol. 7, pp. 30-42, 1996.
- [42] F. Lewis, D. Dawson, and C. Abdallah. *Robot Manipulator Control Theory and Practice*. Marcel Dekker, Inc. New, York, USA, 1992.
- [43] H. Li, H. Jin, and C. Guo. PID Control Based on Wavelet Neural Network Identification and Tuning and Its Application to Fin Stabilizer. In *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 1907-1911, Niagara Falls, Canada, July 2005.
- [44] S. Li and S. Chen. Function Approximation Using Robust Wavelet Neural Networks. In *14th IEEE International Conference on Tools with Artificial Intelligence*, 2002.
- [45] C. Lin, Y. Lee P. Chen, and T. Chen. Multiple Cardiac Arrhythmia Recognition Using Adaptive Wavelet Network. In *Proceedings of the 2005 IEEE Engineering in Medicine and Biology, 27th Annual Conference*, pp. 5655-5659, Shanghai, China, September 2005.
- [46] C. K. Lin. Adaptive Tracking Controller Design for Robotic Systems Using Gaussian Wavelet Networks. In *IEEE Proceedings Control Theory Applications, Depto. of Electrical Engineering Chinese Naval Academy of Kaohsiung*, Vol. 149, num. 4, pp. 316-322, China, July 2002.

- 
- [47] R. Lippmann. *An Introduction to Computing with Neural Networks*. IEEE Signal Processing Magazine, pp. 4-22, April 2002.
- [48] G. Liu, S. Billings, and V. Kadiramanathan. **Nonlinear System Identification Using Wavelet Networks**. In *UKACC International Conference on Control'98*, pp. 1248-1253, UK, September 1998.
- [49] S. Mallat. *A Wavelet Tour of Signal Processing*. Prentice Hall PTR, New York, 1999.
- [50] S. Mallat. *A Theory Multiresolution Signal Decomposition: The Wavelet Representation*. IEEE Trans. Pattern. Anal. Machine Intell., Vol. 11, num. 7, pp. 674-693, July 1989.
- [51] S. Mallat. *Multiresolution Aproximations and Wavelet Orthonormal Bases of  $L^2(R)$* . Trans. Amer. Math. Soc., Vol. 315, pp. 69-87, September 1989.
- [52] S. Masri, A. Chassiakos, and T. Caughey. *Identification of Nonlinear Dynamic System Using Neural Networks*. Journal of Applied Mechanics, Vol. 60, pp. 123-133, March 1993.
- [53] The MathWorks. *The Neural Network Toolbox*, 1995-2004.
- [54] The MathWorks. *The Wavelet Toolbox version 3.0*, 1995-2004.
- [55] The MathWorks. *MATLAB<sup>®</sup> version 7.0*, 2007.
- [56] T. Matsuo. *Application of Wavelet Transform to Control System Diagnosis*. The IEEE Control and Automation Profesional Network, pp. 81-88, Fukuoka, Japan.
- [57] W. McCulloch and W. Pitts. *How we Know Universal*. Bull. Math. Biophys, pp. 127-147, 1947.
- [58] W. McCulloch and W. Pitts. *A Logical Calculus of Ideas Imminent in Nervous Activity*. Bull. Math. Biophys, Vol. 5, pp. 115-133, 1943.
- [59] Y. Meyer. *Méthodes Temps-Fréquence et Méthodes Temps-Échelle en Traitement du Signal et de L'imagen*. INRIA lectures, 1990.
- [60] Y. Meyer. *Ondelettes et Opérateurs*. Hermann, Paris, Francia, 1990.
- [61] Y. Meyer. *Wavelets, Algorithms and Applications*. SIAM, Philadelphia, USA, 1993.

- [62] H. Moghbelli, A. Rahideh, and A. Safavi. **Vector Control of Induction Machines Using Wavenet Based Controller For Traction Applications**. In *IEEE Annual Meeting, Conference record of the 2007, Industry Applications Conference*, pp. 761-767, University of Technology of Isfahan, Iran 2003.
- [63] M. Moreno. *Operación Óptima de Robots Manipuladores Basada en la Planificación de Movimiento*. Tesis de Licenciatura en Ingeniería en Electrónica y Telecomunicaciones, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, México, Noviembre, 2009.
- [64] X.Ñiu, P. Ching, and Y. Chan. **A Wavelet-Based Algorithm for Time Delay and Doppler Measurements**. In *Proceedings Internacional Symposium on Signals, Systems and Electronics*, pp. 485-488, San Francisco, USA 1995.
- [65] K. Ogata. *Sistemas de Control en Tiempo Discreto*. Prentice-Hall, Naucalpan, Edo. de México, 1996.
- [66] K. Ogata. *Ingeniería de Control Moderna*. Prentice-Hall, Naucalpan, Edo. de México, 1998.
- [67] C. Park and P. Tsiotras. **Approximations to Optimal Feedback Control Using a Succesive Wavelet Collocation Algorithm**. In *Proceeding of the American Control Conference*, pp. 1950-1955, Atlanta, Giorgia. USA, June 2003.
- [68] S. Parvez. *Advanced Control Techniques for Motion Control Problem*. Ph. D. Dissertation, Cleveland, Ohio, USA, December 2003.
- [69] S. Parvez and Z. Gao. *A Wavelet-Based Multiresolution PID Controller*. IEEE Transactions on Industry Applications, Vol. 41, num. 2, pp. 537-543, Hills, Ohio, USA, March-April 2005.
- [70] R. Polikar. *The Wavelet Tutorial*. Dept. of Electrical an Computer Engineering, Rowan University, 1996.
- [71] M. Polycarpou, M. Mears, and S. Weaver. **Adaptive Wavelet Control of Non-linear Systems**. In *Proceeding of the 36th Conference on Desicion and Control*, pp. 3890-3895, Sandiego, California, USA, December 1997.
- [72] J. Proakis. *Introduction to Digital Signal Processing*. Macmillan publising, 1988.
- [73] B. Horne R. Hush. *Progress in Supervised Neural Network. What's New Since Lippmann?* IEEE Signal Processing Magazine, pp. 8-39, January 1993.

- [74] I. Razo. *Aprendizaje Reforzado para el Control de un Sistema Subactuado*. Tesis de Maestría en Ciencias Computacionales, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, México, Enero, 2007.
- [75] M. Sedighizadeh and A. Rezazadeh. *Adaptive PID Control of Wind Energy Conversion Systems Using RASPl Mother Wavelet Basis Function Networks*. Proceedings of World Academy of Science, Engineering and Technology, Vol. 27, pp. 269-273, 2008.
- [76] M. Sgarbi, V. Colla, and L. Reyneri. *A Comparison Between Weighted Radial Basis Functions and Wavelet Networks*. Dipartimento di Elettronica di Politecnico di Torino, Torino, Italy.
- [77] Y. Sheng. *The Transforms and Applications Handbook*. CRC Press, 1996.
- [78] M. Spiegel. *Teoría y Problemas de Análisis de Fourier*. McGraw-Hill serie de compendios Shaum, 1981.
- [79] M. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley and Sons, New York, USA, 1989.
- [80] N. Sweeney, R. L. Becker, and B. Sweeney. **A Comparison of Wavelet and Fourier Descriptors for Neural Network Chromosome Classifier**. In *Proceedings 19th International Conference - IEEE/EMBS*, pp. 1359-1362, Chicago, Illinois, USA, October 1997.
- [81] J. Tang. *PID Controller Using The TMS320C31 DSK With On-Line Parameter Adjustment For Real-Time DC Motor Speed And Position Control*. ISIE, pp. 786-791, Pusan, Korea 2001.
- [82] K. Åstrom and T. Hagglund. *PID Controllers: Theory, Design and Tuning*. International, USA, 2007.
- [83] R. Nielsen. *Neurocomputing*. Addison Wesley Publishing Company Inc., 1991.
- [84] J. Thomas and B. Dubuisson. **A Diagnostic Method Using Wavelets Networks Application to Engine Knock Detection**. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 244-249, Beijing, China 1996.
- [85] Y. Tong, Q. Dao, and F. Xu. **AC Motor Control Based on Wavelet Network**. In *Proceeding of the Third International Conference on Machine Learning and Cybernetics*, pp. 861-865, Shanghai, China, August 2004.
- [86] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice Hall PTR, Englewood Cliffs, New Jersey, USA, 2007.

- 
- [87] P. Viñuela and I. León. *Redes de Neuronas Artificiales un Enfoque Práctico*. Person Prentice Hall, España, 2004.
- [88] J. Xu and Y. Tan. **Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks**. In *Proceedings of the American Control Conference on Electronics*, pp. 624-629, Arlington, USA, June 2001.
- [89] S. Zarantonello. *Theory and Application of Wavelets*. Santa Clara University, 1997.
- [90] D. Zhou, W. Cai, and W. Zhang. *An Adaptive Wavelet Method for Nonlinear Circuit Simulation*. IEEE Transactions on Circuits and Systems: Fundametal Theory and Applications, Vol. 46, num. 8, pp. 931-938, August 1999.
- [91] W. Zhou and L. Jia. **A Modeling Method Based on Wavelet Networks**. In *ICSP'04 Proceedings*.
- [92] A. Zilouchian and M. Jamshidi. *Intelligent Control Systems Using Softcomputing Methodologies*. CRC Press, N. Y., 2001.



# Apéndice A

## Conceptos previos

### A.1. Conceptos generales

Para una mayor comprensión sobre el tema de la teoría *wavelet*, se resumen en este apéndice algunos conceptos matemáticos necesarios.

**Definición 1 [Relación de equivalencia]** Si existe un mapeo 1-1 de  $A$  sobre  $B$ , se dice que  $A$  y  $B$  se pueden poner en **correspondencia** ó que  $A$  y  $B$  tienen el mismo **número de cardinalidad** ó simplemente, que  $A$  y  $B$  son **equivalentes**, y se escribe  $A \sim B$ . Esta relación tiene las siguientes propiedades:

- Ésta es reflexiva:  $A \sim B$ .
- Ésta es simétrica: si  $A \sim B$ , entonces  $B \sim A$ .
- Ésta es transitiva: si  $A \sim B$  y  $B \sim C$ , entonces  $A \sim C$ .

cualquier relación con estas tres propiedades se llama una **relación de equivalencia**.

**Definición 2 [Conjuntos contables]** Para algún entero positivo  $n$ , sea  $J_n$  el conjunto cuyos elementos son los enteros  $1, 2, \dots, n$ ; sea  $J$  el conjunto de todos los enteros positivos. Para algún conjunto  $A$ , se dice:

- (a)  $A$  es finito si  $A \sim J_n$  para algún  $n$ .
- (b)  $A$  es infinito si  $A$  es no finito.
- (c)  $A$  es contable si  $A \sim J$ .
- (d)  $A$  es no contable si  $A$  es no finito.

Los **conjuntos contables** son algunas veces llamados **conjuntos enumerables**.

**Ejemplo 2** Sea  $A$  el conjunto de todos los enteros. Entonces  $A$  es **contable**. Para ésto considere los siguientes arreglos de los conjuntos  $A$  y  $J$ :

$$\begin{aligned} A &= \{ 0, 1, -1, 2, -2, 3, -3, \dots \} \\ J &= \{ 1, 2, 3, 4, 5, 6, 7, \dots \} \end{aligned}$$

donde se puede observar que existe una correspondencia 1-1 entre los elementos de ambos conjuntos.

**Definición 3 [Campo]** Es una colección de elementos de un conjunto  $\mathbf{F}$  que posee dos operaciones binarias, llamadas suma y producto, tal que el resultado de estas operaciones de dos elementos cualquiera de  $\mathbf{F}$  da un elemento también de  $\mathbf{F}$ ,  $(\alpha + \beta) \in \mathbf{F}$  y  $(\alpha \cdot \beta) \in \mathbf{F}$ , es decir, es cerrado bajo las operaciones de suma y producto. Además un **campo** cumple con los siguientes axiomas:

1. Operación binaria **adición**, satisfaciendo:

- La adición es conmutativa,  $\alpha + \beta = \beta + \alpha$ ,  $\forall \alpha, \beta \in \mathbf{F}$ .
- La adición es asociativa,  $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$ ,  $\forall \alpha, \beta, \gamma \in \mathbf{F}$ .
- Existe un elemento neutro único, llamado neutro aditivo y denotado por 0, tal que  $\alpha + 0 = \alpha$ ,  $\forall \alpha \in \mathbf{F}$ .
- Existe un elemento llamado inverso aditivo, único a cada elemento del campo  $\alpha \in \mathbf{F}$ , tal que  $\alpha + (-\alpha) = 0$ ,  $\forall \alpha \in \mathbf{F}$ .

2. Operación binaria **producto**, satisfaciendo:

- El producto es conmutativo,  $\alpha \cdot \beta = \beta \cdot \alpha$ ,  $\forall \alpha, \beta \in \mathbf{F}$ .
- El producto es asociativo,  $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$ ,  $\forall \alpha, \beta, \gamma \in \mathbf{F}$ .
- Existe un elemento único, llamado neutro multiplicativo y denotado por 1, tal que  $\alpha \cdot 1 = \alpha$ ,  $\forall \alpha \in \mathbf{F}$ .
- Existe un elemento llamado inverso multiplicativo único para cada elemento del campo  $\alpha \in \mathbf{F}$ , tal que,  $\alpha \cdot \alpha^{-1} = 1$ ,  $\forall \alpha, \alpha^{-1} \in \mathbf{F}$ .

3. El **producto** es distributivo con respecto a la **adición**, es decir

$$\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma), \quad \forall \alpha, \beta, \gamma \in \mathbf{F}.$$

**Observación 1**   ▪ Usualmente se escribe (en cualquier campo)

$$\alpha - \beta, \frac{\alpha}{\beta}, \alpha + \beta + \gamma, \alpha\beta\gamma, \alpha^2, \alpha^3, 2\alpha, 3\alpha, \dots$$

en lugar de

$$\alpha + (-\beta), \alpha \cdot \left(\frac{1}{\beta}\right), \alpha + (\beta + \gamma), \alpha\alpha, \alpha\alpha\alpha, \alpha + \alpha, \alpha + \alpha + \alpha, \dots$$

- Ejemplos de campos son: los números reales  $\mathbb{R}$ ; los números racionales  $\mathbb{Q}$ ; los números complejos  $\mathbb{C}$ .

Un espacio vectorial puede ser definido sobre cualquier campo  $\mathbf{F}$ , sólo que para propósitos de esta tesis los campos a considerar son los reales y los complejos, es decir  $\mathbf{F}\mathbb{R}$  ó  $\mathbf{F}\mathbb{C}$ , dando como resultado un espacio vectorial lineal real y un espacio vectorial lineal complejo, respectivamente.

**Definición 4 [Espacio vectorial lineal]** Un **espacio vectorial lineal real**  $\mathbb{R}$  (espacio lineal complejo  $\mathbb{C}$ , respectivamente) es un conjunto cuyos elementos  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  son llamados vectores que junto con las operaciones de adición

$$(+): V \times V \rightarrow V$$

y multiplicación

$$\begin{aligned} (\cdot): \mathbb{R} \times V &\rightarrow V \\ ((\cdot): \mathbb{C} \times V &\rightarrow V, \text{ respectivamente.}) \end{aligned}$$

cumplen con los siguientes axiomas  $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in V$  y  $r_1, r_2 \in \mathbb{R}$ ,  $(c_1, c_2 \in \mathbb{C})$

1. Ley conmutativa,  $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$ ,
2. Ley asociativa,  $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$ ,
3. Ley asociativa con respecto a los elementos del campo,  $r_1 \cdot (r_2 \cdot \mathbf{x}) = (r_1 r_2) \mathbf{x}$ ,  
 $(c_1 \cdot (c_2 \cdot \mathbf{x})) = (c_1 c_2) \mathbf{x}$ , respectivamente)
4. Ley distributiva con respecto a los elementos del campo,  $r_1 \cdot (\mathbf{x} + \mathbf{y}) = r_1 \cdot \mathbf{x} + r_1 \cdot \mathbf{y}$ ,  
 $(c_1 \cdot (\mathbf{x} + \mathbf{y})) = c_1 \cdot \mathbf{x} + c_1 \cdot \mathbf{y}$ , respectivamente)
5. Ley distributiva con respecto a los elementos del campo,  $(r_1 + r_2) \cdot \mathbf{x} = r_1 \cdot \mathbf{x} + r_2 \cdot \mathbf{x}$ ,  
 $((c_1 + c_2) \cdot \mathbf{x}) = c_1 \cdot \mathbf{x} + c_2 \cdot \mathbf{x}$ , respectivamente)
6. Existe un elemento neutro único, llamado neutro aditivo y denotado por  $0_v \in V$ , tal que  $\mathbf{x} + 0_v = \mathbf{x}$ ,  $\forall \mathbf{x} \in V$
7. Existe un elemento llamado inverso aditivo, único a cada elemento de  $V$ , tal que  $\mathbf{x} + (-\mathbf{x}) = 0_v$ ,

8. Existe un elemento llamado neutro multiplicativo, único a cada elemento de  $V$ , tal que  $\mathbf{x} \cdot 1 = \mathbf{x}$ .

Un espacio vectorial  $V$  es cerrado bajo las operaciones binarias de adición y producto por un escalar, es decir, una operación binaria de un vector  $\mathbf{x}$  y un elemento  $\alpha$  del campo  $\mathbb{R}$ .

Existen propiedades importantes de un espacio vectorial  $V$  que se obtienen de los axiomas de la definición, y se enuncian a continuación:

1.  $\mathbf{x} + \mathbf{y} = \mathbf{x} + \mathbf{z} \Rightarrow \mathbf{y} = \mathbf{z}$ ,
2.  $\alpha \cdot \mathbf{x} = \alpha \cdot \mathbf{y}, \alpha \neq 0_v \Rightarrow \mathbf{x} = \mathbf{y}$ ,
3.  $\alpha \cdot \mathbf{x} = \beta \cdot \mathbf{x}, \mathbf{x} \neq 0_v \Rightarrow \alpha = \beta$ ,
4.  $(\alpha - \beta) \cdot \mathbf{x} = \alpha \cdot \mathbf{x} - \beta \cdot \mathbf{x}$ ,
5.  $\alpha \cdot (\mathbf{x} - \mathbf{y}) = \alpha \cdot \mathbf{x} - \alpha \cdot \mathbf{y}$ ,
6.  $\alpha \cdot 0_v = 0_v$ .

**Definición 5 [Subespacio]** Un subconjunto  $M$  de un espacio vectorial  $V$  es llamado **subespacio** de  $V$  si  $M$  satisface las siguientes condiciones  $\forall \mathbf{x}, \mathbf{y} \in M$

1.  $\mathbf{x} + \mathbf{y} \in M$ ,
2.  $\alpha \cdot \mathbf{x} \in M$ ,

de manera equivalente  $\alpha \cdot \mathbf{x} + \beta \cdot \mathbf{y} \in M$ .

**Observación 2** Debido a que  $M \neq \emptyset \Rightarrow \exists \mathbf{x} \in M \Rightarrow (-1\mathbf{x} + 1\mathbf{x}) \in M$ , entonces

$$(-1 + 1)\mathbf{x} = 0 \cdot \mathbf{x} = 0 \in M,$$

es decir cualquier subespacio contiene el vector nulo.

**Definición 6 [Combinación lineal]** Una **combinación lineal** está formada por la suma de elementos de un espacio vectorial  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n \in V$  en combinación con escalares del campo  $\mathbf{F}$ , es decir

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \alpha_3 \mathbf{x}_3 + \dots + \alpha_n \mathbf{x}_n.$$

Una combinación lineal es usada para construir un subespacio de un subconjunto arbitrario en un espacio vectorial. Una combinación lineal de un subespacio  $M$  está contenido en el mismo subespacio.

**Definición 7 [Generador o Span]** Sea  $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  un conjunto finito de vectores en un espacio vectorial  $V$ . El subconjunto  $W$  de  $V$  generado (spanned) por  $S$  es el conjunto de todas las combinaciones de  $S$ . Este conjunto es llamado **generador** ó **span** de  $S$  y se denota por:

$$[S] = \text{span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$$

**Teorema 1** El **generador** o **span** de cualquier conjunto finito  $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  de vectores en  $V$  es un subconjunto de  $V$ .

**Definición 8 [Dependencia lineal]** Un vector  $\mathbf{x}$  se dice **linealmente dependiente** de un conjunto finito de vectores  $S$ , si  $\mathbf{x}$  puede ser expresado como una combinación lineal de  $S \subset V$ , es decir

$$\mathbf{x} = \sum_i \alpha_i \mathbf{s}_i; \quad \mathbf{s}_i \in S.$$

**Definición 9 [Independencia lineal]** Un conjunto finito de vectores  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n\}$  es **linealmente independiente** si

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \alpha_3 \mathbf{x}_3 + \dots + \alpha_n \mathbf{x}_n = 0,$$

implica que

$$\alpha_1 = \alpha_2 = \alpha_3 = \dots = \alpha_n = 0.$$

Por negación de la Definición 8, un vector  $\mathbf{x}$  se dice **linealmente independiente** del conjunto  $S$  si no es linealmente dependiente de  $S$ .

**Definición 10 [El conjunto vacío,  $\emptyset$  es linealmente dependiente]**  $\emptyset$  es linealmente dependiente de cualquier conjunto de vectores.  $\emptyset$  es considerado un conjunto dependiente. El conjunto de un sólo elemento diferente de cero es linealmente independientes, es decir  $\{V \neq 0\}$  es un conjunto independiente.

**Teorema 2** El conjunto de vectores  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n\}$  son linealmente independientes si y sólo si  $\sum_{k=1}^n \alpha_k \mathbf{x}_k = 0 \Rightarrow \alpha_k = 0; \quad k = 1, 2, 3, \dots, n$ .

**Corolario 1** El conjunto de vectores  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n\}$  son linealmente independientes, si:

$$\sum_{k=1}^n \alpha_k \mathbf{x}_k = \sum_{k=1}^n \beta_k \mathbf{x}_k \Rightarrow \alpha_k = \beta_k; \quad k = 1, 2, 3, \dots, n.$$

**Definición 11 [Base]** Un conjunto finito de vectores  $S$  linealmente independiente se dice una **base** de  $V$  si  $S$  genera  $V$ , es decir:  $[S] \supset V$ .

**Definición 12 [Dimensión]** Un espacio vectorial  $V$  que tiene una base finita se dice **finito dimensional**, si no es finito dimensional se dice **infinito dimensional**.

**Teorema 3** Cualesquiera dos bases de un espacio vectorial tienen el mismo número de elementos.

El número de vectores en una base es igual al número de dimensión.

**Definición 13 [Transformación]** Sean  $V$  y  $Y$  espacios vectoriales y  $D \subset V$

$$\begin{aligned} T : D &\longrightarrow Y, \\ \mathbf{x} &\mapsto \mathbf{y}, \end{aligned}$$

con  $\mathbf{y} = T(\mathbf{x})$ , a  $T$  se llama una **transformación** de  $V$  a  $Y$  con dominio  $D$ .

**Definición 14 [Funcional]** Una transformación de un espacio vectorial  $V$  al espacio de los reales o los complejos es una **funcional** en  $V$ , es decir

$$f : V \longrightarrow \mathbb{K} \quad \text{es una funcional en } V.$$

donde  $\mathbb{K}$  representa el espacio real  $\mathbb{R}$  o el espacio complejo  $\mathbb{C}$ .

**Definición 15 [Transformación lineal]** Sean  $V$  y  $Y$  espacios vectoriales sobre el mismo campo  $\mathbf{F}$ . Se dice que  $T : V \longrightarrow Y$  es **lineal** si  $\forall \mathbf{x}_1, \mathbf{x}_2 \in V$  y  $\forall \alpha_1, \alpha_2 \in \mathbf{F}$  se cumple

$$T(\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2) = \alpha_1 T(\mathbf{x}_1) + \alpha_2 T(\mathbf{x}_2).$$

**Ejemplo 3** A continuación se dan algunos ejemplos de transformaciones lineales:

- Transformada de Laplace

$$L\{\cdot\} = \int_{-\infty}^{\infty} e^{st} dt \{\cdot\},$$

- Transformada de Fourier

$$F\{\cdot\} = \int_{-\infty}^{\infty} e^{-i\omega t} dt \{\cdot\},$$

- Transformada  $Z$

$$Z\{\cdot\} = \sum_{k=-\infty}^{\infty} z^{-k} \{\cdot\},$$

- *Transformada wavelet*

$$W\{\cdot\} = \int_{-\infty}^{\infty} \psi_{a,b}(t) dt \{\cdot\}.$$

Una vez dado una serie de definiciones sobre espacio vectorial como son: dependencia e independencia lineal, base, dimensión, serie, transformación, funcional y transformación lineal. Se observa en todas estas definiciones que no se hace uso de la noción de distancia, para esto es importante dotar al espacio vectorial de una forma de medir.

**Definición 16 [Métrica]** *X es cualquier conjunto no vacío, un **métrica** es una función de dos argumentos  $d(x, y)$ , llamada también función de distancia,*

$$d : X \times X \longrightarrow \mathbb{R}_+,$$

que satisface

- $d(x, y) \geq 0, \quad \forall x, y \in \mathbb{R},$  (Positividad)
- $d(x, y) = 0, \quad$  si y sólo si  $x = y,$
- $d(x, y) = d(y, x),$  (Simetría)
- $d(x, y) \leq d(x, z) + d(z, y),$  (Desigualdad del triángulo).

**Definición 17 [Espacio métrico]** *Un espacio métrico es el par  $(X, d)$ , es decir un conjunto no vacío  $X$  donde se ha definido una métrica.*

*Sea  $X$  un un espacio métrico. Todos los puntos y conjuntos mencionados abajo se entiende elementos y conjuntos de  $X$ .*

- Una **vecindad** de un punto  $p$  es un conjunto  $N_r(p)$  conteniendo todos los puntos  $q$  tal que  $d(p, q) < r$ . El número  $r$  es llamado el **radio** de  $N_r(p)$ .
- Un punto  $p$  es un **punto límite** del conjunto  $E$  si cualquier vecindad de  $p$  contiene un punto  $q \neq p$  tal que  $q \in E$ .
- Si  $p \in E$  y  $p$  no es un punto límite de  $E$ , entonces  $p$  se llama un **punto aislado** de  $E$ .
- $E$  es **cerrado** si cualquier punto límite de  $E$  es un punto de  $E$ .
- La **clausura** de  $E$  es la unión de dicho conjunto con todos sus puntos límites.
- Un punto  $p$  es un **punto interior** de  $E$  si existe una vecindad  $N$  de  $p$  tal que  $N \subset E$ .

- (g)  $E$  es **abierto** si cualquier punto de  $E$  es un punto interior de  $E$ .
- (h) El **complemento** de  $E$  (denotado por  $E^c$ ) es el conjunto de todos los puntos  $p \in X$  tal que  $p \notin E$ .
- (i)  $E$  es **perfecto** si  $E$  es cerrado y cualquier punto de  $E$  es un punto límite de  $E$ .
- (j)  $E$  está **acotado** si hay un número real  $m$  y un punto  $q \in X$  tal que  $d(p, q) < m$  para todo  $p \in E$ .
- (k)  $E$  es **denso** en  $X$  si cualquier punto de  $X$  es un punto límite de  $E$ , o un punto de  $E$  (o ambos).

**Definición 18 [Espacio normado]** Un **espacio normado** es un par ordenado  $(V, \|\cdot\|)$  donde  $V$  es un espacio vectorial real (o complejo) y  $\|\cdot\| : V \rightarrow \mathbb{R}$  es una función valuada en los reales definida sobre  $V$  llamada norma que satisface los siguientes axiomas:

1.  $\|\mathbf{x}\| \geq 0$ ;  $\|\mathbf{x}\| = 0$ , si y sólo si  $\mathbf{x} = 0$ ; (Definida positiva)
2.  $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ ,  $\forall \alpha \in \mathbb{R}$ ; (Homogénea)
3.  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ ; (Desigualdad del triángulo)

Una norma  $\|\cdot\|$  en  $V$  define una métrica  $d$  en  $V$  dada por  $d(x, y) = \|x - y\|$  y es llamada la métrica inducida por la norma.

A continuación se dan algunos ejemplos de espacios normados:

**Ejemplo 4** *Espacio Euclidiano y espacio unitario.* El espacio vectorial lineal real de  $n$ -tuplas ordenadas de números reales  $\mathbb{R}^n$  (o complejos  $\mathbb{C}$ ), donde para  $\mathbf{x} = x_1, x_2, \dots, x_n$  junto con la función

$$\|\cdot\|_2 : \mathbb{R}^n \rightarrow \mathbb{R}, \text{ (o } \|\cdot\|_2 : \mathbb{C}^n \rightarrow \mathbb{C}), \text{ definida por}$$

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2},$$

en el caso real se llama  $n$ -espacio Euclidiano y se denota por  $(\mathbb{R}^n, \|\cdot\|_2)$  y en el caso complejo  $n$ -espacio unitario y se denota por  $(\mathbb{C}^n, \|\cdot\|_2)$ .

**Ejemplo 5** *Considere nuevamente el espacio vectorial lineal  $\mathbb{R}^n$  (o  $\mathbb{C}^n$ ), junto con la función*

$$\|\cdot\|_p : \mathbb{R}^n \rightarrow \mathbb{R}, \text{ (o } \|\cdot\|_p : \mathbb{C}^n \rightarrow \mathbb{R}), \text{ definida por}$$

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}},$$



donde  $p$  es cualquier número en el intervalo  $[1, \infty]$ . Si  $p = 1$ , entonces  $\|\cdot\|_p$  está dada por

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|,$$

mientras que si  $p \rightarrow \infty$ , entonces la norma  $\|\cdot\|_p$  se aproxima a

$$\begin{aligned} \|\mathbf{x}\|_\infty &= \text{máx}|x_i|, \quad 1 \leq i \leq n \\ &= \text{máx}\{|x_i| \mid i \in \{1, 2, \dots, n\}\}. \end{aligned}$$

Una vez dada la definición de norma sobre un espacio vectorial, éste se ha dotado de una forma de medir y ahora se está en posición de estudiar conceptos como convergencia, continuidad, límite, etc.

**Definición 19 [Transformación continua]** Sean  $V$  y  $Y$  espacios normados, se dice  $T : V \rightarrow Y$  es **continuo** en  $\mathbf{x}_0 \in V$ , si  $\forall \epsilon > 0 \exists \delta > 0$ , tal que  $\|\mathbf{x} - \mathbf{x}_0\|_V < \delta \Rightarrow \|T(\mathbf{x}) - T(\mathbf{x}_0)\|_Y < \epsilon$ .

Como se puede observar la continuidad depende de la norma en ambos espacios  $V$  y  $Y$ . Se debe notar que  $T$  tiene que estar definida en el punto  $\mathbf{x}_0$  para que sea continua en  $\mathbf{x}_0$ .

**Definición 20 [Sucesión]** Una **sucesión** de elementos de un espacio normado ( $V, \|\cdot\|$ ) es una función cuyo dominio es el conjunto de los números naturales o un subconjunto propio de  $\mathbb{N}$ .

$$\begin{aligned} f : \mathbb{N} &\longrightarrow V, \\ n &\mapsto x_n, \end{aligned}$$

y se escribe  $x_n = f(n)$ . Una sucesión teniendo dominio a un subconjunto finito de  $\mathbb{N}$  son llamados sucesiones finitas. Es común en la práctica una notación de subíndice para una sucesión y escribir  $f(n)$  como  $x_n$  denotando la sucesión por

$$\{x_n\} = x_1, x_2, \dots, x_n, \dots$$

Cualquier subconjunto infinito de  $S$  de números naturales puede ser representado como el rango de una subsucesión donde los términos preservan el orden de sistema de números naturales  $\mathbb{N}$ ; esto es, hay una función  $g$  que preserva el orden

$$\begin{aligned} S &\subset \mathbb{N}, \\ f : \mathbb{N} &\longrightarrow V, \\ f : \mathbb{N} &\longrightarrow S, \\ f \circ g(n) &= f(g(n)). \end{aligned}$$

**Definición 21 [Subsucesión]** Sea  $\{x_n\}$  una sucesión en  $V$ . Sea  $\{x_{n_k}\} = n_1, n_2, \dots, n_k, \dots$  una sucesión de enteros positivos que son estrictamente crecientes, es decir,  $n_j > n_k \quad \forall j > k$ .

Entonces la sucesión  $\{x_{n_k}\}$  se le llama una **subsucesión** de  $\{x_n\}$ .

**Definición 22 [Límite de la subsucesión]** Si la subsucesión  $\{x_{n_k}\}$  converge, entonces su límite se le llama el **límite de la subsucesión** de  $\{x_n\}$ .

**Definición 23 [Serie]** Sea  $\{x_n\}$  una sucesión, se puede construir otra sucesión  $\{s_n\}$  considerando la suma de los primeros  $n$  términos.

$$s_n = \sum_{i=1}^n x_i = x_1 + x_2 + \dots + x_n$$

A una sucesión obtenida así  $\{s_n\}$  se llama **serie** y se dice que  $s_n$  es su  $n$ -ésima suma parcial.

**Definición 24 [Serie convergente]** Dada una serie

$$\sum_{i=1}^n x_i = x_1 + x_2 + \dots + x_n,$$

se denotará mediante el símbolo  $x_n$ . Si la sucesión  $\{x_n\}$  es **convergente** y si existe el

$$\lim_{n \rightarrow \infty} x_n = x,$$

entonces la serie  $\sum_{i=1}^n x_i$  se llama **convergente** y se escribe

$$x_1 + x_2 + \dots + x_n + \dots = x \quad \text{ó} \quad \sum_{i=1}^n x_n = x,$$

El vector  $x$  se denomina **suma** ó **límite** de la serie. Si es otro caso, la serie se llamará **divergente**.

**Observación 3** El conjunto de todos los puntos  $x_n$  ( $n = 1, 2, 3, \dots$ ) es el **rango** de  $\{x_n\}$ . El rango de una sucesión puede ser un conjunto finito o éste puede ser infinito. La sucesión  $\{x_n\}$  se dice **acotado** si su rango es acotado.

Empleando los conceptos de sucesiones es posible relacionar la definición de continuidad y convergencia, la cual está dada en la siguiente proposición.

**Proposición 1** Sea  $V, Y$  espacios normados, y sea  $T : V \rightarrow Y$ ;  $T$  es continuo, si y sólo si  $\mathbf{x}_n \rightarrow \mathbf{x}_0 \Rightarrow T(\mathbf{x}_n) \rightarrow T(\mathbf{x}_0)$ .

**Definición 25 [Convergencia]** En un espacio vectorial normado  $V$ , una sucesión infinita de vectores  $\{\mathbf{x}_n\}$  se dice que **converge** a  $\mathbf{x}$ , si la sucesión de números reales  $\{\|\mathbf{x} - \mathbf{x}_n\|\}$  converge a cero. Se escribe  $\mathbf{x}_n \rightarrow \mathbf{x}$ .

Ya que se dotó al espacio vectorial de una forma de medir (norma), se estudian conceptos que involucran la forma de medir, sin embargo existen otros conceptos que son de vital importancia para el entendimiento del comportamiento de los elementos del espacio o de una sucesión. tal concepto se da a continuación.

**Definición 26 [Sucesión de Cauchy]** Una sucesión  $\{\mathbf{x}_n\}$  en un espacio normado  $(V, \|\cdot\|)$  se dice sucesión de **Cauchy** si  $\|\mathbf{x}_n - \mathbf{x}_m\| \rightarrow 0$ , cuando  $n, m \rightarrow \infty$ ; es decir dado  $\epsilon > 0 \exists N \in \mathbb{N}$  tal que

$$\|\mathbf{x}_n - \mathbf{x}_m\| < \epsilon,$$

si  $n \geq N$  y  $m \geq N$ .

**Observación 4** La diferencia entre la definición de convergencia y la definición de sucesión de Cauchy es que en el primer caso el límite está explícitamente especificado, mientras que en el caso de una sucesión de Cauchy no es así.

**Corolario 2** En un espacio normado toda sucesión convergente es de Cauchy, es decir, si  $\mathbf{x}_n \rightarrow \mathbf{x}$ , entonces

$$\|\mathbf{x}_n - \mathbf{x}_m\| = \|\mathbf{x}_n - \mathbf{x} + \mathbf{x} - \mathbf{x}_m\| \leq \|\mathbf{x}_n - \mathbf{x}\| + \|\mathbf{x} - \mathbf{x}_m\| \rightarrow 0.$$

En general una secuencia de Cauchy puede no ser convergente.

**Definición 27 [Espacio de Banach]** Un espacio normado  $V$  es completo si cada sucesión de Cauchy de  $V$  tiene un límite en  $V$ . Un espacio vectorial completo se le llama **espacio de Banach**.

Los espacios de Banach son importantes por dos razones:

- i) Si  $(V, \|\cdot\|)$  es un espacio Banach entonces todas sucesiones de Cauchy son convergentes. Esta propiedad provee un medio de prueba si una sucesión es convergente sin tener a la mano un candidato para el límite de la sucesión.
- ii) Aún si un espacio vectorial normado  $(V, \|\cdot\|)$  no es completo, este puede hacerse un espacio Banach añadiendo algunos elementos, por razones obvias, este proceso es conocido como *acompletar* el espacio.

En base al Corolario 2 se tiene que en un espacio completo las sucesiones de Cauchy y las convergentes son las mismas, es decir, una sucesión converge si y sólo si es de Cauchy. Análogamente en un espacio incompleto deben existir sucesiones de Cauchy no convergentes.

**Definición 28 [Producto interno]** Sea  $V$  un espacio vectorial. Una función  $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{K}$  se llama **producto interno** si cumple con los siguientes axiomas para todo  $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$  y  $\lambda \in \mathbb{K}$ :

$$[P_1] \langle \mathbf{x}, \mathbf{x} \rangle > 0 \quad \forall \mathbf{x} \neq 0 \quad \text{y} \quad \langle \mathbf{x}, \mathbf{x} \rangle = 0 \quad \text{si} \quad \mathbf{x} = 0; \quad (\text{Definida positiva})$$

$$[P_2] \langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}; \quad (\text{Hermítica})$$

$$[P_3] \langle \lambda \mathbf{x}, \mathbf{y} \rangle = \lambda \langle \mathbf{x}, \mathbf{y} \rangle; \quad (\text{Homogénea})$$

$$[P_4] \langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle; \quad (\text{Aditiva ó lineal en } \mathbf{x})$$

**Definición 29 [Espacio con producto interior]** Un par  $(V, \langle \cdot, \cdot \rangle)$  donde  $\langle \cdot, \cdot \rangle$  es un producto interior en  $V$ , se llama **espacio con producto interior**.

La importancia de este tema es que todo espacio con producto interno  $(V, \langle \cdot, \cdot \rangle)$  tiene una norma asociada a él, que es inducida por el producto interno  $\langle \cdot, \cdot \rangle$ .

**Proposición 2** Sea  $(V, \langle \cdot, \cdot \rangle)$  un espacio con producto interno. Entonces la función  $\|\cdot\| : V \rightarrow \mathbb{R}$  definida  $\forall \mathbf{x} \in V$  por  $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$  es una norma; es decir, se cumplen los 3 axiomas de la norma dadas en la Definición 18.

**Definición 30 [Espacio pre-Hilbert]** Un espacio normado  $(V, \|\cdot\|)$  en el cual la norma  $\|\cdot\|$  es inducida por el producto interior, se llama **espacio pre-Hilbert**.

**Definición 31 [Espacio Hilbert]** Un espacio normado completo con producto interior se llama **espacio Hilbert**.

Un espacio Hilbert es un espacio pre-Hilbert completo (respecto a la norma asociada). Por lo tanto, todo espacio Hilbert es un espacio de Banach en el cual se ha definido un producto interior.

Un espacio de Banach  $(V, \|\cdot\|)$ , que a la vez es un espacio pre-Hilbert, se llama espacio Hilbert. En otras palabras, un espacio Hilbert es un espacio pre-Hilbert completo.

**Proposición 3** Sea  $(V, \|\cdot\|)$  un espacio pre-Hilbert y sea  $\langle \cdot, \cdot \rangle$  el producto interior que induce la norma  $\|\cdot\|$ , si  $\mathbf{x}_n \rightarrow \mathbf{x}$ ,  $\mathbf{y}_n \rightarrow \mathbf{y}$ , entonces

$$\langle \mathbf{x}_n, \mathbf{y}_n \rangle \rightarrow \langle \mathbf{x}, \mathbf{y} \rangle,$$

es decir, la función  $\langle \cdot, \cdot \rangle$  es continua en  $V \times V$ .

**Observación 5** Como se ha visto que no todos los espacios normados son espacios pre-Hilbert.

Ahora se da una condición necesaria y suficiente para que una norma sea inducida por un producto interior.

**Definición 32 [Elementos ortogonales]** Dado un espacio pre-Hilbert  $(V, \langle \cdot, \cdot \rangle)$ . Sea  $\mathbf{x}, \mathbf{y} \in V$ . Se dice que  $\mathbf{x}$  y  $\mathbf{y}$  son **ortogonales**, si

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0,$$

y se escribe  $\mathbf{x} \perp \mathbf{y}$ . Se dice que  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in V$  son ortogonales dos a dos si  $\mathbf{x}_i \perp \mathbf{x}_j$  para  $i \neq j$ .

**Definición 33 [Elemento ortonormal]** Sea  $(V, \langle \cdot, \cdot \rangle)$  un espacio pre-Hilbert. Se dice que un elemento  $\mathbf{x} \in V$  es **ortonormal** si  $\|\mathbf{x}\| = 1$ .

**Definición 34 [Conjunto ortonormal]** Sea  $(V, \langle \cdot, \cdot \rangle)$  un espacio pre-Hilbert. Sea  $A \subset V$ ,  $A \neq \emptyset$ . Se dice que  $A$  es **ortonormal** si  $A$  es ortogonal y además se tiene que  $\|\mathbf{x}\| = 1$  para cada  $\mathbf{x} \in A$ .

**Teorema 4** Sea  $(V, \langle \cdot, \cdot \rangle)$  un espacio con producto interno. Si  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in V$  son no nulos y ortogonales dos a dos, entonces son **linealmente independientes**.

**Corolario 3** En un espacio con producto interior de dimensión  $n$ , se cumple

- (a) Cualquier colección de  $n$  vectores ortogonales no nulos constituyen una base.
- (b) No hay colección de vectores ortogonales no nulos con más de  $n$  elementos.

**Definición 35 [Base ortogonal y base ortonormal]** Una **base ortogonal** en un espacio con producto interno  $(V, \langle \cdot, \cdot \rangle)$  es una base que consta de vectores ortogonales dos a dos. Si además de ser ortogonales son ortonormales, entonces se llama una **base ortonormal**.

**Ejemplo 6** Dada una base ortogonal cualquiera  $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$  en un espacio con producto interior, es fácil obtener las coordenadas de cualquier vector.

Si  $\mathbf{x} \in V$ , entonces

$$\mathbf{x} = \sum_{i=1}^n \alpha_i \mathbf{e}_i = \alpha_1 \mathbf{e}_1 + \dots + \alpha_n \mathbf{e}_n, \quad (\text{A.1})$$

formando el producto interno por  $\mathbf{e}_i$  en ambos miembros, se obtiene

$$\langle \mathbf{x}, \mathbf{e}_i \rangle = \left\langle \sum_{i=1}^n \alpha_i \mathbf{e}_i, \mathbf{e}_i \right\rangle, \quad (\text{A.2})$$

debido a que  $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 0$  si  $i \neq j$ , entonces  $\forall i = 1, 2, \dots, n$

$$\alpha_i = \frac{\langle \mathbf{x}, \mathbf{e}_i \rangle}{\|\mathbf{e}_i\|^2}. \quad (\text{A.3})$$

La expresión para  $\alpha_i$  se simplifica considerablemente si los vectores de la base son unitarios, es decir  $\|\mathbf{e}_i\| = 1$ , entonces

$$\alpha_i = \langle \mathbf{x}, \mathbf{e}_i \rangle \quad (\text{A.4})$$

y sustituyendo (A.4) en (A.1) resulta

$$\mathbf{x} = \sum_{i=1}^n \langle \mathbf{x}, \mathbf{e}_i \rangle \mathbf{e}_i,$$

los términos  $\alpha_i$ , reciben un nombre especial dado en la siguiente definición.

**Definición 36 [Coeficiente de Fourier]** Sea  $V$  un espacio Hilbert,  $\{\mathbf{e}_n\}$  una sucesión ortogonal y  $\mathbf{x} \in V$ . A los elementos

$$\alpha_n = \langle \mathbf{x}, \mathbf{e}_n \rangle,$$

se les llama el **coeficiente de Fourier** de  $\mathbf{x}$ , relativo a  $\{\mathbf{e}_n\}$ .

Todos estos antecedentes matemáticos están formando la base, tal como ahora se puede observar que las series de Fourier son un caso particular de los Espacios de Hilbert. Así también como es conocida la Igualdad de Parseval, muy usada en el procesamiento de señales, la cual se puede interpretar con estos conocimientos.

**Definición 37 [Igualdad de Parseval]**  $\int_0^\infty |f(t)|^2 dt = \|\alpha\|^2$  En donde se observa que la energía no se altera en los 2 espacios, en el tiempo en el primer miembro de la ecuación y del espacio de la frecuencia en el segundo miembro. La conservación de energía está dada por los coeficientes de Fourier  $\alpha$ , una vez transformada la señal en el tiempo.

**Definición 38 [Espacio de funciones de cuadrado integrable]** Es el espacio vectorial de todas las funciones  $f(t)$  definidas sobre  $\mathbb{R}$ , es decir, están en el espacio  $L^2(\mathbb{R})$ , si  $\|f(t)\|^2$  es integrable, como

$$\int_{-\infty}^{\infty} \|f(t)\|^2 dt < \infty,$$

por lo que, el espacio de funciones de cuadrado integrable se define como

$$L^2(\mathbb{R}) = \{f(t) : f \text{ es medible y } \int_{-\infty}^{\infty} \|f(x)\|^2 dx < \infty\}.$$

## **A.2. Comentarios**

Los conceptos presentados en este apéndice son importantes para el entendimiento de la teoría *wavelet*, las redes neuronales y la teoría *wavenet*, empleadas para el desarrollo del control PID *wavelet* y el control PID *wavenet*.





# Apéndice B

## Teoría wavelet

El objetivo de este apéndice es presentar la teoría *wavelet*, necesaria para el desarrollo de un controlador *wavelet* y un controlador *wavenet*, siendo este último el objetivo principal de la tesis. La organización del presente apéndice es de la siguiente forma: en la Sección A.1 se da una introducción a la transformada *wavelet*, en la Sección A.2 se ve de manera breve el análisis de Fourier, la teoría *wavelet* es presentada en la Sección A.3, en la Sección A.4 se presenta el análisis multiresolución basada en *wavelet*, la codificación sub-banda es presentada en la Sección A.5 y por último en la Sección A.6 se dan algunos comentarios referente al apéndice presentado.

### B.1. Introducción

Este apéndice se enfoca a la teoría de la transformada *wavelet*, que es una herramienta matemática generada a mediados de los 80's. Siendo ésta muy eficiente para el análisis local de señales no estacionarias y de rápida transitoriedad, mapea la señal en una representación tiempo-escala que provee un análisis multiresolución con ventanas dilatadas, similar a una transformada de Fourier ventaneada, pero el análisis de frecuencias de mayor rango se realiza usando ventanas angostas y el análisis de las frecuencias de menor rango se hace utilizando ventanas anchas, y no fijas como se realiza en la transformada de Fourier. Además la transformada *wavelet* no es solamente local en frecuencia, sino también en tiempo.

Dentro de los usos de esta poderosa herramienta se puede nombrar, además del análisis local de señales no estacionarias, el análisis de señales electrocardiográficas, sísmicas, de sonido, de radar, así como también es utilizada para la compresión y procesamiento de imágenes y reconocimiento de patrones, etc.

## B.2. Análisis de Fourier

Jean Baptiste J. Fourier matemático y físico francés quien demostró que una función podía ser desarrollada en términos de series trigonométricas convergentes llamadas Series de Fourier, publicando en 1807, desarrolló esta teoría cuando estudiaba la ecuación del calor el cual fue publicado en 1822 en su celebre "*Théorie Analytique de la Chaleur*" (Teoría Analítica del Calor) [19].

### B.2.1. Funciones periódicas

Dado que los términos de las series trigonométricas son periódicas es lógico deducir que las funciones que se van a desarrollar mediante dichas series deben ser también periódicas. Se dice que una función  $f(x)$  tiene un período  $P$  o es periódica con un período  $P$  si para todo  $x$ ,  $f(x + P) = f(x)$ , donde  $P$  es una constante Positiva, en la Figura B.1 se muestra una señal periódica. El menor valor de  $P > 0$  se llama el período mínimo o período fundamental de  $f(x)$  [78].

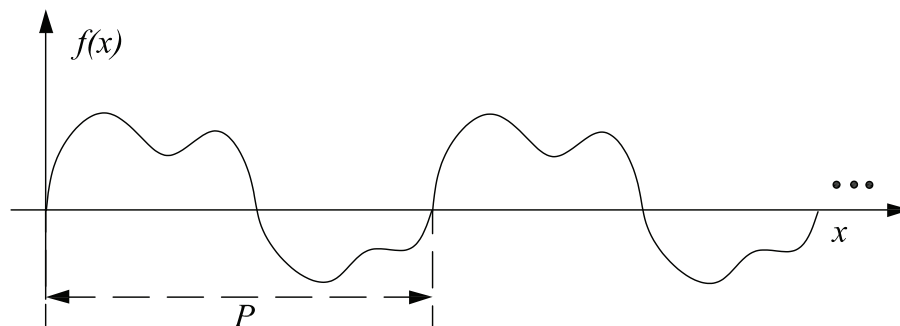


Figura B.1: Representación de una función periódica  $f(x)$ , con período  $P$ .

### B.2.2. Coeficientes y series de Fourier

Los desarrollos en series de Fourier, tienen dos aplicaciones fundamentales:

- Representar una función  $f(x)$  definida en el intervalo  $(-c, c)$ , para los valores de  $x$  en ese intervalo.
- Representar una función periódica con un período  $2c$  para todos los valores de  $x$ .

La función  $f(x)$  puede ser proyectada en una base ortonormal de funciones  $\{\phi_k(x)\}$ , de la siguiente forma [7]:

$$\hat{f}(x) = c_1\phi_1(x) + c_2\phi_2(x) + \dots + c_k\phi_k(x) + \dots, \quad (-c < x < c); \quad k = 1, 2, 3 \dots \quad (\text{B.1})$$

se espera que el desarrollo de  $\hat{f}(x)$  converja a la función original  $f(x)$ .

Se puede demostrar que los coeficientes  $c_k$  de la suma son los coeficientes de Fourier de  $f(x)$  con respecto a la base ortonormal  $\{\phi_k(x)\}$ . Estos coeficientes pueden ser expresarse como:

$$c_k = \int_{-c}^c f(x) \overline{\phi_k(x)} dx, \quad k = 1, 2, 3 \dots \quad (\text{B.2})$$

siendo  $\hat{\phi}$  el complejo conjugado de  $\phi$ .

La serie de (B.1) con estos coeficientes es la serie de Fourier genérica correspondiente a la función  $f(x)$ , y se define como:

$$f(x) = \sum_{k=1}^{\infty} c_k \phi_k(x), \quad (\text{B.3})$$

si  $f(x)$  está definida en el intervalo  $(0, 2\pi)$  y determinada fuera de ese intervalo por  $f(x + 2\pi) = f(x)$ , esto es,  $f(x)$  tiene periodo  $2\pi$ , la serie de Fourier que corresponde a  $f(x)$  sobre la base ortogonal de senos y cosenos se define como:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx), \quad (\text{B.4})$$

donde los coeficientes de Fourier  $a_k$  y  $b_k$  se definen como:

$$\begin{cases} a_k &= \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx dx, \\ b_k &= \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx dx, \end{cases} \quad \text{con } k = 1, 2, 3, \dots \quad (\text{B.5})$$

Puede observarse que los coeficientes de Fourier de la función transformada representan la contribución de cada función seno y coseno para cada frecuencia [78].

Usando la identidad de Euler:  $e^{ikx} = \cos kx + i \sin kx$  se puede escribir la serie de Fourier de  $f(x)$  como una combinación lineal de funciones exponenciales complejas:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}, \quad (\text{B.6})$$

donde las funciones  $\phi_k(x) = \frac{e^{ikx}}{\sqrt{2\pi}}$  constituyen un conjunto ortonormal [89].

Los coeficientes de Fourier de  $f(x)$ , respecto de esta base, pueden expresarse como:

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} f(x)e^{-ikx} dx. \quad (\text{B.7})$$

Es evidente que la serie de Fourier, (B.3), no puede representar una función para todos los valores de  $x$  si la función no es periódica.

### B.2.3. Transformada de Fourier

Para obtener una representación que pueda ser válida para todos los valores de  $x$  cuando  $f(x)$  no es periódica, es natural intentar extender la representación anterior dejando que  $c$  tienda a infinito, lo que da lugar a la transformada de Fourier [7].

La transformada de Fourier de una función no periódica  $f(x)$  esta definida por

$$F(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-iwx} dx, \quad (\text{B.8})$$

donde  $w$  es una variable compleja continua.

La función puede ser reconstruida a partir de sus componentes de Fourier, por medio de la transformada inversa de Fourier:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(w)e^{iwx} dw \quad (\text{B.9})$$

La transformada de Fourier en  $L^2(\mathbb{R})$   $[-\infty, \infty]$  satisface las siguientes propiedades:

- Es una Transformación de Fourier uno-a-uno de  $L^2(\mathbb{R})$   $[-\infty, \infty]$  en sí mismo
- Preserva la norma

$$\int_{-\infty}^{\infty} \|f(x)\|^2 dx = \int_{-\infty}^{\infty} \|F(w)\|^2 dw \quad (\text{B.10})$$

- Preserva el producto interno

$$\int_{-\infty}^{\infty} f(x) * g(x) dx = \int_{-\infty}^{\infty} F(w) * G(w) dw \quad (\text{B.11})$$

### B.2.4. Limitaciones del análisis de Fourier

La transformada de Fourier es ampliamente utilizada en el procesamiento y análisis de señales y con resultados satisfactorios en los casos en que estas señales son periódicas y lo suficientes regulares, pero no ocurre lo mismo para el análisis de señales cuyo espectro varía con el tiempo (señales no estacionarias) [72].

Tomando el caso en el que la función  $f$  a descomponer es una señal dependiente del tiempo, puede decirse que las funciones de la base de Fourier son de duración infinita en el tiempo, pero locales en frecuencia.

La transformada de Fourier detecta la presencia de una determinada frecuencia pero no brinda información acerca de la evolución en el tiempo de las características espectrales de la señal. Muchos aspectos temporales de la señal, tales como el comienzo y el fin de una señal finita y el instante de aparición de una singularidad en una señal transitoria, no puede ser analizados adecuadamente por el análisis de Fourier.

Para los casos de señales no estacionarias y transitorias se utiliza generalmente la transformada de Fourier ventaneada.

### B.2.5. Transformada de Fourier ventaneada

Una forma de analizar una señal no estacionaria es realizar un análisis espectral dependiente del tiempo. Una señal estacionaria es dividida en una secuencia de segmentos de tiempo en los cuales la señal puede ser considerada como una cuasi-estacionaria y la transformada de Fourier es aplicada a cada segmento local de la señal. Gabor, en 1940, fue el primero en introducir la transformada de Fourier de tiempo corto, conocida como la transformada de Fourier con ventana deslizante, definida como

$$S_f(\omega, \sigma) = \int f(t) * g(t - \sigma) e^{-i\omega t} dt, \quad (\text{B.12})$$

donde  $g(t)$  es una ventana deslizante, la cual tiene un ancho fijo y cambia a lo largo del eje  $x$  por un factor  $\sigma$  [77], la función está definida como (B.13). Así, propuso a la función Gaussiana como la función ventana  $g(t)$  y demostró que la transformada de Fourier de una ventana Gaussiana continúa siendo Gaussiana [77].

$$g(t) = \frac{1}{\sigma} e^{-\frac{\pi t^2}{\sigma^2}} \quad (\text{B.13})$$

Con la transformada de Fourier ventaneada se logra una mejor localización de la aparición de una singularidad en una señal. Pero solo conocerá en que intervalo de tiempo se produce la singularidad, debido a que la localización depende del ancho elegido para la función ventana. Además, los eventos no podrán ser resueltos si aparecen muy cerca unos de otros, ya que no será posible distinguir diferentes comportamientos dentro de una misma amplitud de ventana [86]. Dada las desventajas que se presenta con la transformada de Fourier ventaneada, surge la transformada *wavelet* como alternativa a Fourier.

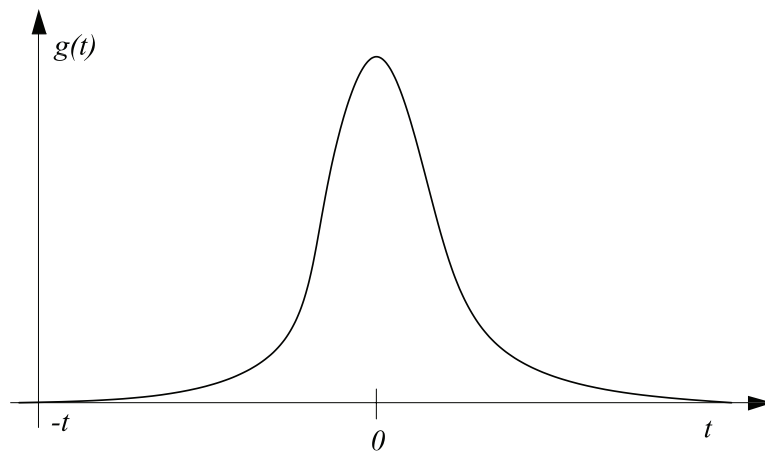


Figura B.2: Representación gráfica de (B.13), donde  $g(t)$  es la función de Gauss,  $t$  es el tiempo,  $\sigma$  es el factor de desplazamiento.

### B.3. Teoría *wavelet*

Aunque la primera *wavelet* que se conoce fue descubierta en 1910, Alfred Haar, un matemático Húngaro, descubre una “base” de funciones que se reconocen actualmente como las primeras *wavelets*, denominadas “*wavelets* de Haar”[27]. Tiempo después, el ingeniero petrolífero Jean Morlet en 1981, desarrolló su propia forma de analizar las señales sísmicas para crear componentes que estuvieran localizados en el espacio, a los que denominó “*wavelets* de forma constante”, actualmente se conocen como “*wavelets* de Morlet”, independientemente de que los componentes se dilaten, compriman o desplacen en el tiempo, mantienen la misma forma. Alex Grossmann, un físico de Marsella, trabajó con Morlet durante un año para confirmar que las ondas se podían reconstruir a partir de sus descomposiciones en *wavelets*. Demostrando, que la transformada *wavelet* funciona mucho mejor que la transformada de Fourier, porque eran mucho menos susceptibles a pequeños errores de cómputo. Un error o un truncamiento indeseados de los coeficientes de Fourier pueden transformar una señal suave en una saltarina o viceversa; las *wavelets* evitan tales consecuencias desastrosas. Publicó su artículo [26] en 1984, en el cual se utilizó por primera vez la palabra “*wavelet*”.

En 1985, Yves Meyer, reconocido ampliamente como uno de los fundadores de la teoría *wavelet* [59, 60, 61], descubre las primeras *wavelets* ortogonales suaves. “Ortogonalidad” significa que la información capturada por una *wavelet* es completamente independiente de la información capturada por otra. En 1986, Stéphane Mallat, un antiguo alumno de Meyer, vinculó la teoría de *wavelets* a la literatura existente sobre codificación de subbanda y filtros de duplicación de cuadratura, que son las versiones de las *wavelets* de la comunidad de procesamiento de imágenes. La idea

del análisis multiresolución (es decir, la observación de señales a distintas escalas de resolución) ya era familiar para los expertos en procesamiento de imágenes. Mallat, en colaboración con Meyer, demostraron que las *wavelets* están implícitas en el proceso del análisis multiresolución [50, 51].

Gracias al trabajo de Mallat, las *wavelets* se convirtieron en algo mucho más sencillo. Ya se podía hacer un análisis con las *wavelets* sin necesidad de conocer la fórmula de una *wavelet* madre. El proceso se redujo a sencillas operaciones, el lenguaje de las *wavelets* también resultó más cómodo para los ingenieros eléctricos, que adoptaron términos familiares como “filtros”, “altas frecuencias” y “bajas frecuencias”[49].

En 1987, Ingrid Daubechies, descubrió una clase completamente nueva de *wavelets* [12, 13], que no sólo eran ortogonales (como las de Meyer) sino que también se podían implementar mediante sencillas ideas de filtrado digital. Las nuevas *wavelets* eran casi tan sencillas de programar y utilizar como las *wavelets* de Haar, pero eran suaves, sin los saltos de las *wavelets* de Haar. Los procesadores de señales disponían ahora de una herramienta de ensueño: una manera de descomponer datos digitales en contribuciones de diversas escalas. Al combinar las ideas de Daubechies y Mallat, se disponía de una transformación ortogonal y sencilla que se podía calcular rápidamente en las modernas computadoras digitales.

Los trabajos importantes surgidos en la década de los 80's, dieron origen a lo que hoy en día se le conoce como la Teoría *wavelet*.

Literalmente el término *wavelet* significa pequeña onda u ondoleta. Dentro de un contexto más general, una *wavelet* es una función que satisface las condiciones:

- tiene una explosión concentrada pequeña de energía finita en el dominio del tiempo, y
- exhibe una cierta oscilación en el tiempo.

La primera condición hace que la *wavelet* sea pequeña en el sentido que es bien localizada en el tiempo, mientras que la segunda condición permite observar la ondulación que la hace ser así llamada *wavelet*. Existen algunas características importantes de las *wavelets*, tales como el soporte compacto, que es la propiedad de que la *wavelet* sea de duración finita, lo que permite una menor complejidad en los cálculos, mejor resolución en el tiempo y pobre resolución en frecuencia. Otra característica para ser empleada con filtros digitales es la simetría que permite que los filtros sean de fase lineal.

Después de haber dado un poco de historia sobre las *wavelets*, se procede a dar la teoría *wavelet* de forma formal, por lo que primero se da la definición de función *wavelet*.

**Definición 39** [*Función wavelet,  $\psi$* ] Una función wavelet es una función  $\psi \in L^2(\mathbb{R})$  que cumple con la siguiente condición de admisibilidad

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0, \quad (\text{B.14})$$

La función wavelet es tradicionalmente llamada wavelet madre, por que a partir de ella se pueden generar una familia de funciones de doble índice llamadas funciones wavelets hijas.

A continuación se dan algunos ejemplos de funciones wavelets madre más comunes, entre estas se encuentran: Haar, Mexican hat, Morlet, Meyer, Daubechies, Shannon [49, 54]. La definición y la gráfica correspondiente a cada una de dichas wavelets madre se muestran en la Tabla B.1 y en la Figura B.3.

Nombre	Definición
Haar	$\psi(t) = \begin{cases} 1, & \text{si } t \in [0, \frac{1}{2}] \\ -1, & \text{si } t \in (\frac{1}{2}, 1] \\ 0, & \text{en otro caso} \end{cases}$
Mexican hat	$\psi(t) = \frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} (1 - t^2) e^{(-\frac{1}{2}t^2)}$
Morlet	$\psi(t) = e^{-\frac{t^2}{2}} \cos(5t)$
Meyer	$\psi(\omega) = \begin{cases} (2\pi)^{-\frac{1}{2}} e^{\frac{i\omega}{2}} \sin(\frac{\pi}{2} v (\frac{3}{2\pi} \omega  - 1)), & \text{si } \frac{2\pi}{3} \leq  \omega  \leq \frac{4\pi}{3} \\ (2\pi)^{-\frac{1}{2}} e^{\frac{i\omega}{2}} \cos(\frac{\pi}{2} v (\frac{3}{4\pi} \omega  - 1)), & \text{si } \frac{4\pi}{3} \leq  \omega  \leq \frac{8\pi}{3} \\ 0, & \text{en otro caso} \end{cases}$ $v = a^4(35 - 84a + 70a^2 - 20a^3), \quad a \in [0, 1]$
Daubechies	$P(y) = \sum_{k=0}^{N-1} C_k^{N-1+k} y^k,$ $C_k^{N-1+k}$ son coeficientes binomiales, N es el orden de la wavelet
Shannon	$\psi(t) = \frac{\text{sen}(\frac{\pi}{2}t)}{\frac{\pi}{2}t} \cos(\frac{3\pi}{2}t)$

Tabla B.1: Algunos ejemplos comunes de wavelets madre.

**Definición 40** [*función wavelet hija,  $\psi_{a,b}$* ] Una función wavelet hija, es una función generada a partir de la dilatación-contracción y traslación de una función wavelet madre, y se denota como

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a > 0; \quad a, b \in \mathbb{R}, \quad (\text{B.15})$$



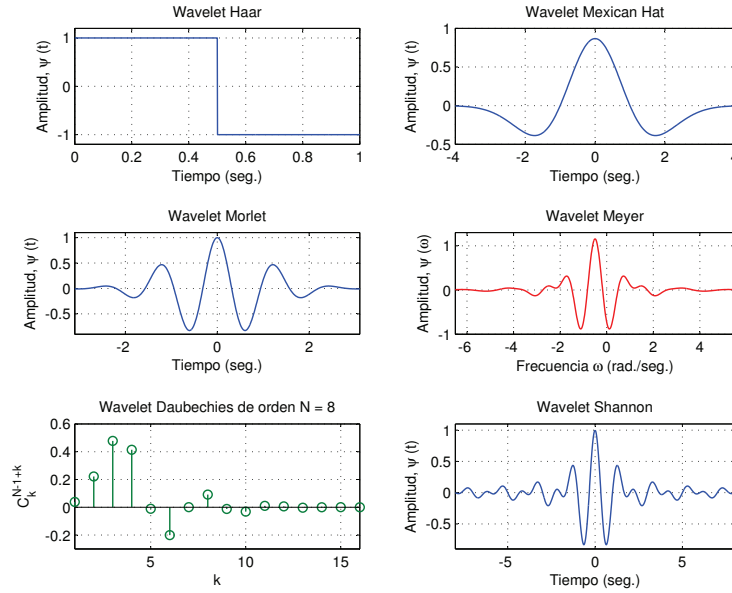


Figura B.3: Gráficas de las *wavelets* madre dadas en la Tabla B.1.

donde la variable escala “ $a$ ” permite hacer dilataciones y las contracciones de la señal, y la variable de traslación “ $b$ ” permite el desplazamiento en el tiempo.

El factor de normalización  $(\sqrt{a})^{-1}$  ha sido seleccionada tal que

$$\|\psi_{a,b}\| = \|\psi\|, \quad \forall a, b \in \mathbb{R}$$

por lo que se puede asumir que  $\|\psi\| = 1$ .

### B.3.1. Transformada *wavelet*

Existen diferentes tipos de transformada *wavelet*, todas parten de las *wavelets* básicas anteriores. Principalmente, se distingue entre transformada *wavelet* continua y transformada *wavelet* discreta y dentro de ésta última se clasifican los sistemas discretos redundantes o frames y las bases *wavelets* ortonormales. Las funciones bases *wavelets* ortonormales inducen el análisis multiresolución, que nos permiten descomponer una función  $f \in L^2(\mathbb{R})$ , lo cual es muy importante en el desarrollo del control *wavelet*.

Antes de tratar el análisis multiresolución basada *wavelet*, trataremos la transformada *wavelet* continua y la transformada *wavelet* discreta,

### B.3.1.1. Transformada *wavelet* continua

Sea  $V_1$  y  $V_2$ , espacios Hilbert, la transformada *wavelet* continua,  $W_f(a, b)$  es descrita como un mapeo

$$\begin{aligned} W_f &: V_1 \rightarrow V_2 \\ f(t) &\mapsto \langle f, \psi_{a,b} \rangle \end{aligned}$$

donde  $f \in L^2(\mathbb{R}) = V_1$  y  $\psi_{a,b}$  es una función *wavelet* hija. Es decir, la transformada *wavelet* continua es definida como un mapeo  $W_f(a, b)$  que es dependiente de las especificaciones de una función *wavelet* madre  $\psi$ .

A continuación se da la definición formal

**Definición 41** [*Transformada wavelet continua*]

La transformada *wavelet* continua de una función  $f \in L^2(\mathbb{R})$  correspondiente a una determinada *wavelet*  $\psi$  es

$$W_f(a, b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi \left( \frac{t-b}{a} \right) dt. \quad (\text{B.16})$$

Para este caso específico los parámetros de dilatación  $a$  y traslación  $b$  varían continuamente sobre  $\mathbb{R}$ , con la restricción  $a \neq 0$ . El parámetro  $b$  es utilizado para trasladar en el tiempo la función *wavelet* madre, tal que para un valor fijo de  $a$ , la transformada *wavelet* puede ser vista como una convolución de  $f(t)$  y la *wavelet* escalada con el tiempo invertido, de la siguiente manera

$$W_f(a, t) = |a|^{-\frac{1}{2}} f(t) \psi_a(t); \quad \psi_a(t) = \psi\left(\frac{-t}{a}\right).$$

El efecto de desplazamiento produce como resultado que la transformada *wavelet* realice un efecto de ampliación en fenómenos de alta frecuencia y de muy breve duración, tal como en transiciones de la señal o singularidades en funciones.

**Observaciones.** De la definición de la transformada *wavelet* se dan las siguientes observaciones:

- El análisis wavelet es frecuentemente llamado análisis tiempo-escala mientras que en Fourier es llamado análisis tiempo-frecuencia.
- La correspondencia  $f(t) \rightarrow W_f(a, b)$  indica un cambio de una función de una variable por una función a dos variables, dentro de los cuales son construidos porciones de correlaciones. Esta redundancia de la representación es explotada en una aplicación conocida como concepto esqueleto de una señal extraída de la transformada *wavelet* continua, la cual puede ser usada para filtrado no lineal [86].

### Propiedades

A continuación se mencionan algunas de las propiedades que posee la transformada *wavelet* continua, todas ellas surgen como relación directa de la base matemática en la que se apoya la transformada *wavelet*.

Considerando funciones arbitrarias  $f, g \in L^2(\mathbb{R})$ , y escalares  $\alpha, \beta \in \mathbb{R}$ , se satisfacen las siguientes propiedades

[P<sub>1</sub>]  $W[\alpha f + \beta g](a, b) = \alpha W_f(a, b) + \beta W_g(a, b)$ ; (Linealidad)

[P<sub>2</sub>]  $W_{f'}(a, b) = W_f(\frac{a}{\alpha}, \frac{b}{\alpha})$ ; para  $f' = \frac{1}{\sqrt{\alpha}} f(\frac{t}{\alpha})$ ; (Escalamiento)

[P<sub>3</sub>]  $W_{f'}(a, b) = W_f(a, b \Leftrightarrow b_0)$ ; para  $f' = f(t \Leftrightarrow b_0)$ ; (Traslación en el tiempo)

[P<sub>4</sub>]  $\sum_{a,b \in \mathbb{Z}} |\langle \psi_{a,b}, f \rangle|^2 = \|f\|^2$ ; (Igualdad de Parseval)

[P<sub>5</sub>] Para localizar los valores de  $W_g(a, b)$  que contengan información alrededor del tiempo  $t_0$ , dada una *wavelet*  $\psi(a, b)$  y una función  $g = f(t_0)$ ; en la escala  $a$ , los coeficientes *wavelets* que satisfacen el índice  $b$  serán influenciados  $(-b_1 + b) 2^a \leq t_0 \leq (b_2 + b) 2^a$  o bien en término de  $b$ ,  $2^{-a}t_0 - b_2 \leq b \leq 2^{-a}t_0 + b_1$ ; (Localización en el tiempo)

[P<sub>6</sub>] Para localizar los valores de  $W_f(a, b)$  que contengan información alrededor de una componente de frecuencia  $\omega_0$ , dada una *wavelet*  $\psi(a, b)$  y una función  $f(t)$ ; en la escala  $a$ , los coeficientes *wavelets* que satisfacen el siguiente rango de escalas son afectados  $\frac{\omega_{min}}{2^a} \leq \omega_0 \leq \frac{\omega_{max}}{2^a}$  dado en término de  $a$ ,  $\log_2(\frac{\omega_{min}}{\omega_0}) \leq a \leq \log_2(\frac{\omega_{max}}{\omega_0})$ ; (Localización en la frecuencia)

[P<sub>7</sub>]  $\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |W_f(a, b)|^2 \frac{da db}{a^2}$ , donde  $C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$  y  $\Psi(\omega)$  es la transformada de Fourier de  $\psi$ ; (Conservación de la energía)

[P<sub>8</sub>] Para asegurar la perfecta reconstrucción, la transformada *wavelet* deberá cumplir  $\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$ , donde  $\Psi(\omega)$  es la transformada de Fourier de  $\psi$  y  $\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0$ . (Admisibilidad)

Una vez que se comprende el proceso de análisis de la transformada *wavelet*, así como sus propiedades, es importante conocer el proceso inverso que nos permite reconstruir la señal. Se considerarán dos variantes de transformadas *wavelets* continuas que difieren en la forma de realizar la reconstrucción. Específicamente una reconstrucción integral desde un plano completo de tiempo-frecuencia y una reconstrucción semi-discreto. Aquí se da a conocer la reconstrucción integral.

El producto interno de dos señales  $f(t)$  y  $y(t)$  está relacionado a el producto interno de su transformada *wavelet* como:

$$\langle f, y \rangle = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b) \overline{W_y(a, b)} \frac{da db}{a^2}, \quad (\text{B.17})$$

con

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty.$$

Dado el producto interno, se obtiene una ecuación de síntesis, de la siguiente manera:

Primeramente, definiendo

$$y_t(t') = \delta(t' - t), \quad (\text{B.18})$$

porque además la siguiente relación se cumple

$$\langle f, y_t \rangle = \int_{-\infty}^{\infty} f(t') \delta(t' - t) dt' = f(t), \quad (\text{B.19})$$

substituyendo (B.18), en ??, se obtiene

$$\langle f, y_t \rangle = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b) |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} \delta(t' - t) \psi\left(\frac{t' - b}{a}\right) dt' \frac{da db}{a^2}, \quad (\text{B.20})$$

de aquí obtenemos la fórmula de reconstrucción

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a, b) \psi\left(\frac{t - b}{a}\right) \frac{da db}{a^2}, \quad (\text{B.21})$$

### B.3.1.2. Transformada *wavelet* discreta

Para *wavelets* discretas los parámetros de escala y traslación son elegidos tal que en el nivel  $m$  la *wavelet*  $a_0^m \psi(a_0^{-m}t)$ , es  $a_0^m$  veces el ancho de  $\psi(t)$ . Esto significa que el parámetro de escala es  $a = a_0^m$  :  $m \in \mathbb{Z}$  y el parámetro de traslación  $b = kb_0 a_0^m$  :  $m, k \in \mathbb{Z}$ . Por lo tanto la familia de *wavelets* está dada por

$$\psi_{m,k}(t) = \frac{1}{\sqrt{a_0^m}} \psi(a_0^{-m}t - kb_0). \quad (\text{B.22})$$

A partir de estas consideraciones se formula la definición de la transformada discreta, que se da a continuación.

**Definición 42 [Transformada wavelet discreta]** La transformada wavelet discreta, se obtiene de la restricción de que  $a, b$  son solamente valores discretos  $a = a_0^m$ ,  $b = kb_0 a_0^m$ , donde  $a_0 > 1$ ,  $b_0 > 0$  son valores fijos. Dada una función  $f \in L^2(\mathbb{R})$  su transformada wavelet discreta se define como

$$W_f[m, k] = \frac{1}{\sqrt{a_0^m}} \int_{-\infty}^{\infty} f(t) \psi \left( \frac{t}{a_0^m} - kb_0 \right) dt; \quad m, k \in \mathbb{Z}$$

donde  $\psi$  es una wavelet madre.

Para recuperar  $f(t)$  de los coeficientes  $W_f[m, k]$  debe existir la siguiente condición de estabilidad,

$$A \|f(t)\|^2 \leq \frac{2}{A+B} \sum_m \sum_k |W_f[m, k]|^2 \leq B \|f(t)\|^2 \quad (\text{B.23})$$

es decir, que la norma al cuadrado de la señal original está acotada por arriba y por abajo para  $A > 0$  y  $B < \infty$  para todas las señales  $f(t)$  en  $L^2(\mathbb{R})$ . Entonces la fórmula de reconstrucción está determinada por

$$f(t) \approx \frac{2}{A+B} \sum_m \sum_k W_f[m, k] \psi_{m,k}(t), \quad (\text{B.24})$$

entre más cercanos sean los valores de  $A$  y  $B$  más aproximada será la reconstrucción. Cuando  $A = B = 1$  la familia de *wavelets* es ortonormal.

## B.4. Análisis multiresolución basada en *wavelets*

El análisis multiresolución, que como lo dice su nombre es un análisis de la señal a estudiar de tal modo que cada componente de frecuencia es analizado con una resolución diferente. Esto es una alternativa más sobre la transformada en tiempo corto de Fourier TTCF que analiza todas las componentes de frecuencia a una misma resolución dada.

En la Figura B.4 los ejes que se manejan son tiempo y frecuencia, aquí se ejemplifica lo que se mencionó anteriormente. Cabe señalar que para la TTCF las ventanas serían de igual dimensión para todos los casos ya que la función utilizada para ese análisis no se dilata ni se contrae (ver Figura B.4a), lo que representa una ventaja de la transformada *wavelet* sobre la TTCF, ya que se puede tener mayor detalle del comportamiento de la señal, dada la resolución que puede alcanzar.

La función base  $e^{i\omega_k t}$  (empleada en la transformada de Fourier) representa líneas horizontales, infinitamente pequeñas y rectángulos infinitamente largos. La base *wavelet* corresponderá a rectángulos finitos, estrechos y altos; rectángulos delgados para altas

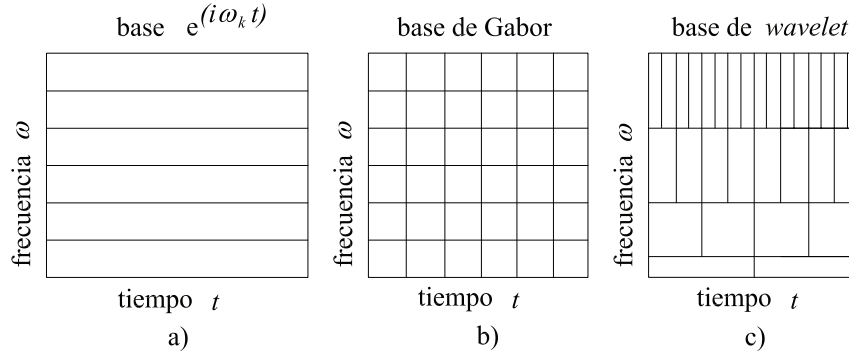


Figura B.4: Plano tiempo-frecuencia para diferentes bases ortogonales: a) base de Fourier, b) base de Gabor y c) base *wavelet*.

frecuencias, y anchos y pequeños para bajas frecuencias, como se muestra en la Figura B.4c,

Una primera tentativa de alcanzar tal objetivo es dada por la transformada ventaneada de Fourier o por la TTCF. Aquí las funciones de la base  $e^{i\omega t}$  se substituyen por versiones ventaneadas  $\psi_{\omega,b}(t) = g(t-b)e^{i\omega t}$ , donde  $g$  es una ventana. En el caso de la transformada Gabor la correspondiente ventana es  $g(t) = e^{(-t^2)}$ , cuyo plano tiempo frecuencia es una cuadrícula mostrada en la Figura B.4b.

Para el análisis multiresolución basada en *wavelet*, se tiene:

Sea  $\phi(x) \in L^2(\mathbb{R})$  la cual se llama la *función de escalonamiento*. Definiendo las siguientes funciones

$$\phi_{j,k}(x) = 2^{-\frac{j}{2}}\phi(2^{-j}x - k); \quad j, k \in \mathbb{Z}, \quad (\text{B.25})$$

donde la base

$$\{\phi_{j,k}(x) | k \in \mathbb{Z}\}, \quad (\text{B.26})$$

es estándar y ortonormal,  $j$  y  $k$  son la dilatación y traslación de la función de escalonamiento, respectivamente.

Ahora, sea  $V_j$  el subespacio escalado de  $L^2(\mathbb{R})$  generado con la base ortonormal estándar, es decir, una cadena anidada de subespacios cerrados

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots, \quad (\text{B.27})$$

tal que

$$\bigcap_j V_j = \{0\}, \quad (\text{B.28})$$

$$\text{cl} \left\{ \bigcup_j V_j \right\} = L^2(\mathbb{R}), \quad \text{cl} = \text{clausura}, \quad (\text{B.29})$$

donde  $V_j$  es el subespacio generado por  $\{\phi_{j,k}\}_{k=0}^\infty$ .

Para obtener un subespacio ortonormal complementario de  $V_j$ , es decir

$$V_{j-1} = V_j \oplus W_j; \quad V_j \perp W_j; \quad j \in \mathbb{Z}, \quad (\text{B.30})$$

$$L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j. \quad (\text{B.31})$$

El análisis multiresolución cumple con la propiedad

$$V_0 = V_1 \oplus W_1 = V_2 \oplus W_2 \oplus W_1 = V_3 \oplus W_3 \oplus W_2 \oplus W_1 = \dots, \quad (\text{B.32})$$

y

$$V_j = \text{span}\{2^{-\frac{j}{2}}\phi(2^{-j}x - k)\}; \quad j \in \mathbb{Z}, \quad (\text{B.33})$$

$$W_j = \text{span}\{2^{-\frac{j}{2}}\psi(2^{-j}x - k)\}; \quad j \in \mathbb{Z}. \quad (\text{B.34})$$

Entonces existe una función  $\psi(x) \in W_0$  tal que el conjunto

$$\{\psi(x - k), k \in \mathbb{Z}\}, \quad (\text{B.35})$$

cuyas funciones

$$\psi_{j,k}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k); \quad j, k \in \mathbb{Z}, \quad (\text{B.36})$$

se hace para obtener una base estándar ortonormal de  $W_0$ . Tradicionalmente se llama la función *wavelet* o función madre. Ahora se emplea este andameaje matemático para la representación de funciones dada las bases estándares construidas previamente. Para efectuar dicha representación se proponen dos formas distintas de acuerdo con (B.30 - B.31).

Dada  $f(x) \in L^2(\mathbb{R})$  se puede descomponer por:

(a)

$$\begin{aligned} f(x) &= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \psi_{j,k}(x), \\ f(x) &= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \psi_{j,k}, \end{aligned} \quad (\text{B.37})$$

(b)

$$\begin{aligned}
f(x) &= \sum_{k=-\infty}^{\infty} \langle f, \phi_{J,k}(x) \rangle \phi_{J,k}(x) + \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \psi_{j,k}(x), \\
f(x) &= \sum_{k=-\infty}^{\infty} c_{J,k} \phi_{J,k}(x) + \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \psi_{j,k}.
\end{aligned} \tag{B.38}$$

Lo importante aquí es que la descomposición de la función  $f(x)$  para valores lo suficientemente grandes de  $J$ , puede ser aproximada arbitrariamente cerca de  $V_J$ . Esto que  $\exists$  algún  $\epsilon > 0$

$$\|f(x) - \sum_{k=-\infty}^{\infty} c_{J,k} \phi_{J,k}(x)\| < \epsilon. \tag{B.39}$$

La aproximación por el truncamiento de la descomposición *wavelet* puede ser aproximada como:

$$f(x) \approx \sum_{k=-\infty}^{\infty} c_{J,k} \phi_{J,k}(x). \tag{B.40}$$

Esta expresión indica que algunos componentes finos (frecuencias altas) que pertenecen al espacio *wavelet*  $W_j$  para la función  $f(x)$  son eliminados y los componentes *coarse* (frecuencias bajas) que pertenecen al espacio escalado  $V_j$  son preservados para aproximar la función original a una escala  $J$ . En realidad, la mayoría de los procesos dinámicos son en general pasa bajos.

Entonces (B.40) nos dice que cualquier función  $f(x) \in L^2(\mathbb{R})$  puede ser aproximada por una combinación lineal finita.

**Observación 6** De la ecuación (B.40) se puede observar que la expresión de la derecha tiene la estructura adecuada para ser una red neuronal wavelet de 3 capas, mostrada en la Figura B.5, lo cual será usada como base en el diseño de un PID wavenet.

## B.5. Análisis sub-banda

Una forma eficiente para realizar el análisis multiresolución basada en *wavelets* es desarrollarla en una estructura de codificación sub-banda, la condificación sub-banda fue propuesta por Croisier, Esteban y Galand [9], usando una clase especial de filtros llamados *filtros de espejo de cuadratura* FEC y por Crochiere, Webber y Flanagan



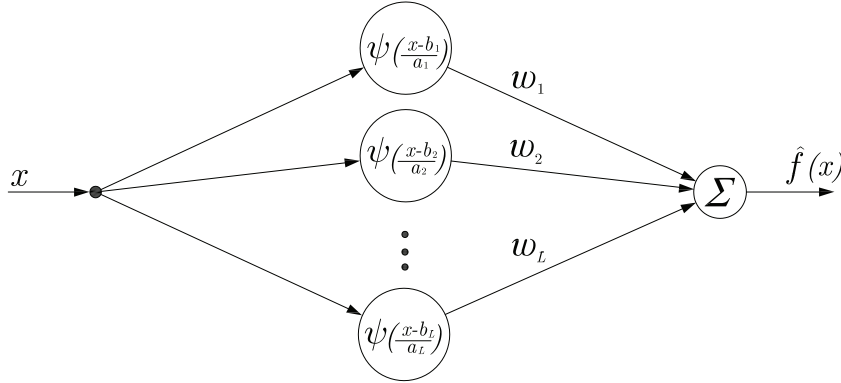


Figura B.5: Arquitectura de una red neuronal *wavelet* SISO.

[8], esto permitió el estudio de banco de filtros de reconstrucción perfecta. Un banco de filtro puede ser usado bajo ciertas condiciones para generar bases *wavelets*, como lo mostró I. Daubechies [12], los cuales permiten realizar un descomposición de una señal  $f(x) \in L^2(\mathbb{R})$ .

El proceso requerido para el análisis multiresolución empleando el esquema de codificación sub-banda, está compuesto de dos etapas: El análisis de descomposición mostrado en la Figura B.6, y la síntesis de descomposición mostrado en la Figura B.7.

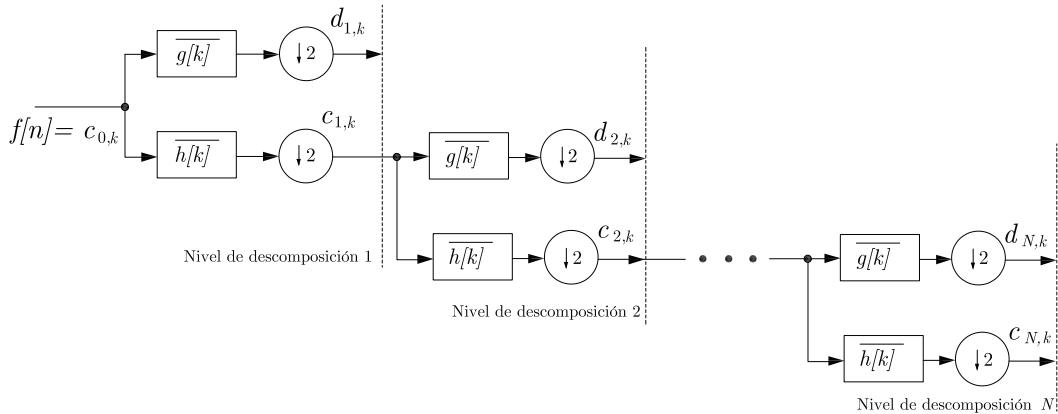


Figura B.6: Análisis de descomposición de la función  $f[n]$ .

En el análisis de descomposición la señal de entrada  $f[n]$  (donde  $f[n]$  es la señal muestreada de  $f(x)$ ) es pasada a través de un filtro pasa bajas  $\overline{h[k]}$  y el otro es un filtro pasa altas  $\overline{g[k]}$  (que son los filtros conjugados de  $h[k]$  y  $g[k]$ ), los cuales manejan la mitad del ancho de banda de la señal de entrada  $f[n]$ . El símbolo  $\downarrow 2$  representa la operación de decimación a dos (*down sampling*). La salida del filtro pasa bajas se vuelve a enviar a otro par de filtros de las mismas características. De este modo se va

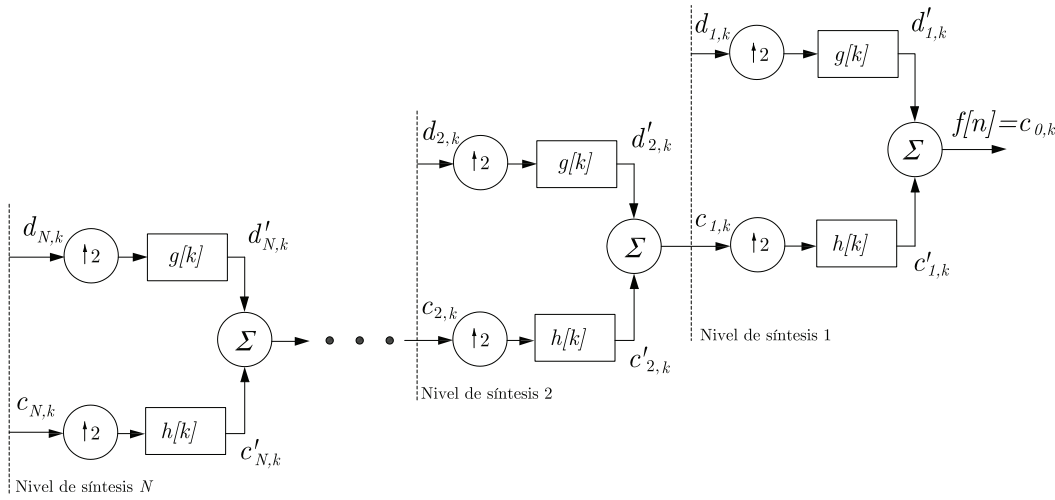


Figura B.7: Síntesis de descomposición de la señal  $f[n]$ .

reduciendo el ancho de banda de la señal y eso se traduce en la reducción a la mitad de la resolución. Lo que significa que a mayor número de etapas de filtrado se tendrá una mayor resolución [70].

En la síntesis de descomposición se recupera la señal  $f[n]$  a través de las señales descompuestas  $c_{m,k}$  y  $d_{m,k}$ , pasando primero por una interpolación a dos (*up sampling*) representado por el símbolo  $\uparrow 2$ , seguida de un filtro pasa bajas  $h[k]$  para  $c_{m,k}$  y un filtro pasa altas  $g[k]$  para  $d_{m,k}$ .

A continuación se desglosa un poco más estas etapas, para tratar con la convolución seguida de una decimación (como se muestra en la Figura B.6) la cual es requerida para la descomposición de la señal, la interpolación a dos seguida de una convolución (como se muestra en la Figura B.7) requerida para la recuperación de la señal.

### B.5.1. Análisis de descomposición

El proceso del análisis de descomposición consiste en tomar una señal  $f[n]$  y descomponerla hasta un nivel de descomposición  $N$ . Pasando la señal  $f[n]$  a través de un par de filtros conjugados  $\overline{h[k]}$  y  $\overline{g[k]}$ , generando  $c_{1,k}$  y  $d_{1,k}$ , para posteriormente tomar la señal  $c_{1,k}$  y pasarla a través de un par de filtros conjugados de las mismas características que los anteriormente mencionados, generando las señales  $c_{2,k}$  y  $d_{2,k}$ , y así hasta obtener las señales  $c_{N,k}$  y  $d_{N,k}$  en el nivel de descomposición  $N$ , como se muestra en la Figura B.6. De manera general el proceso consiste de tomar una señal  $c_{i-1,k}$  pasarla a través del par de filtros conjugados  $\overline{h[k]}$  y  $\overline{g[k]}$  y entregar las señales  $c_{i,k}$  y  $d_{i,k}$ . Estos pasos son repetitivos en cada nivel de descomposición, además el proceso

de tomar la señal y pasarla a través del filtro conjugado, también es repetitivo.

Dado  $i = 1, 2, \dots, N$ , donde  $N$  es el nivel de descomposición. Para un nivel de descomposición  $i$ . Se tiene que,  $c_{i-1,k}$  es la señal de entrada y  $\overline{h[k]}$  el filtro conjugado, que genera la señal  $c_i$  (para generar la señal  $d_{i,k}$  se emplea el filtro conjugado  $\overline{g[k]}$ ) como se muestra en la Figura B.8.

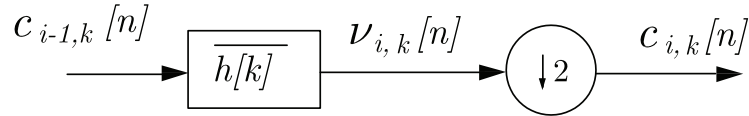


Figura B.8: Diagrama de una convolución seguida de una decimación a dos, donde  $c_{i-1}[n]$ ,  $c_{i,k}[n]$  y  $\nu_{i,k}[n]$  son la señal de entrada, la señal de salida y la señal resultante de la convolución entre la entrada y el filtro conjugado  $\overline{h[k]}$  de longitud  $\kappa$ .

El diagrama mostrado en la Figura B.8, es una convolución seguida de una decimación a dos, cuya representación matemática puede ser expresada como:

$$c_i[n] = \{c_{i-1,k}[n] * \overline{h[n]}\}_{\downarrow 2}, \quad (\text{B.41})$$

donde  $c_i$  es la señal de salida después de ser descompuesta,  $c_{i-1,k}$  es la señal de entrada a ser descompuesta,  $\overline{h[k]}$  es filtro conjugado de longitud  $\kappa$ . Si  $\nu_i[n]$  es la salida de la convolución entre  $c_{i-1}[n]$  y  $\overline{h[n]}$  representado matemáticamente como [24]:

$$\nu_{i,k}[n] = \sum_{k=0}^{\kappa-1} \overline{h[k]} c_{i-1,k}[n-k], \quad (\text{B.42})$$

entonces la decimación a dos de  $\nu_{i,k}[n]$  es:

$$c_i[n] = \nu[2n] = \sum_{k=0}^{\kappa-1} \overline{h[k]} c_{i-1,k}[2n-k], \quad (\text{B.43})$$

de la misma forma para calcular  $d_{i,k}$  se calcula como (B.44) y la Figura B.9 muestra su diagrama para el proceso.

$$d_i[n] = v[2n] = \sum_{k=0}^{\kappa-1} \overline{g[k]} c_{i-1,k}[2n-k], \quad (\text{B.44})$$

donde

$$v_{i,k}[n] = \sum_{k=0}^{\kappa-1} \overline{g[k]} c_{i-1,k}[n-k]. \quad (\text{B.45})$$

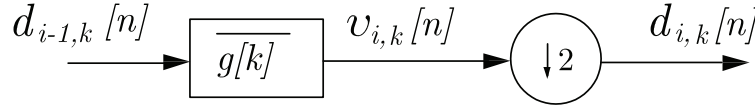


Figura B.9: Diagrama de una convolución seguida de una decimación a dos, donde  $d_{i-1}[n]$ ,  $d_{i,k}[n]$  y  $v_{i,k}[n]$  son la señal de entrada, la señal de salida y la señal resultante de la convolución entre la entrada y el filtro conjugado  $\overline{g[k]}$  de longitud  $\kappa$ .

Para el caso en que el  $i = 1$  considera a  $c_{0,k}$  como  $f[n]$ . El algoritmo para el análisis de descomposición es dado por el Algoritmo B.2.

ALGORITMO B.2: Algoritmo de análisis de descomposición
<ol style="list-style-type: none"> <li>1. Se toma la señal <math>f[n]</math> y se calcula <math>c_{1,k}[n]</math> y <math>d_{1,k}[n]</math> con (B.43) y (B.44), respectivamente. Considerando a <math>f[n] = c_{0,k}</math>.</li> <li>2. Ahora se toma <math>c_{i-1,k}[n]</math>, y se calcula <math>c_i[n]</math> y <math>d_i[n]</math> con (B.43) y (B.44), respectivamente, para <math>i = 2, 3, \dots, N</math>.</li> </ol>



### B.5.2. Síntesis de descomposición

El proceso de la síntesis de descomposición consiste en tomar las señales  $c_{N,k}$  y  $d_{N,k}$  desde el nivel de descomposición para recuperar la señal  $c_{N-1}$ , posteriormente ahora tomar las señales  $c_{N-1}$  y  $d_{N-1}$  para obtener  $c_{N-2}$  y así hasta obtener  $f[n]$ , como se muestra en la Figura B.7. Para la elaboración de este proceso en cada nivel de síntesis se requiere de hacer una interpolación a dos seguida de una convolución. Ésto para recuperar la señal  $c_{i,k}$  y  $d_{i,k}$  y obtener  $c_{i-1,k}$  en un nivel de síntesis  $i$ , para  $i = 1, 2, \dots, N$ .

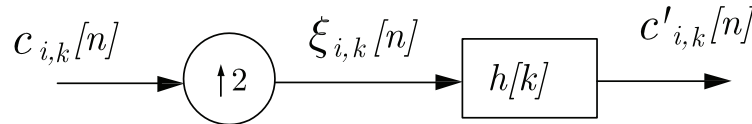


Figura B.10: Diagrama de una interpolación a dos seguida de una convolución, donde  $c_i[n]$ ,  $c'_{i,k}[n]$  y  $\xi_{i,k}[n]$  son la señal de entrada, la señal de salida y la señal resultante de la interpolación a dos de la entrada, y  $g[k]$  es el filtro de longitud  $\kappa$ .

En la Figura B.10, se muestra una interpolación a dos seguida de una convolución,

cuya representación matemática es dada por:

$$c'_{i,k} = \{h[k] * c_{i,k}\}_{\uparrow 2}, \quad (\text{B.46})$$

donde  $c'_{i,k}$  es la señal de salida,  $h[k]$  es el filtro de longitud  $\kappa$ ,  $c_{i,k}$  es la señal de entrada (para el caso de  $d'_{i,k}$  como salida y  $c_{i,k}$  como entrada el filtro utilizado es  $g[n]$ ). Si  $v_{i,k}[n]$  es el resultado de interpolar a dos a  $c_{i,k}$ , y cuya representación matemática es:

$$\xi_{i,k}[n] = \begin{cases} c_{i,k}[\frac{n}{2}], & \text{para } n = 2\alpha, \quad \alpha \in \mathbb{Z}, \\ 0, & \text{en otro caso,} \end{cases} \quad (\text{B.47})$$

por lo que la salida  $c'_{i,k}$  es dada por:

$$c'_{i,k}[n] = \sum_{k=0}^{\kappa-1} h[k]\xi_{i,k}[n-k]. \quad (\text{B.48})$$

Para  $d'_{i,k}$  se recupera de la misma forma con (B.49), y la Figura B.11 muestra el diagrama para el proceso

$$d'_{i,k}[n] = \sum_{k=0}^{\kappa-1} g[k]\zeta_{i,k}[n-k], \quad (\text{B.49})$$

donde

$$\zeta_{i,k}[n] = \begin{cases} d_{i,k}[\frac{n}{2}], & \text{para } n = 2\alpha, \quad \alpha \in \mathbb{Z}, \\ 0, & \text{en otro caso.} \end{cases} \quad (\text{B.50})$$

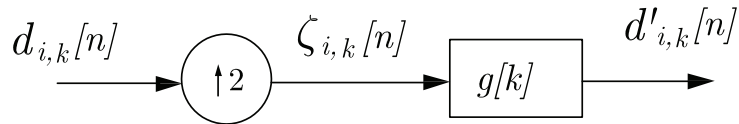


Figura B.11: Diagrama de una interpolación a dos seguida de una convolución, donde  $d_i[n]$ ,  $d'_{i,k}[n]$  y  $\zeta_{i,k}[n]$  son la señal de entrada, la señal de salida y la señal resultante de la interpolación a dos de la entrada, y  $g[k]$  es el filtro de longitud  $\kappa$ .

Una vez calculadas  $c'_{i,k}[n]$  y  $d'_{i,k}[n]$ , la señal recuperada  $c_{i-1,k}[n]$  para el siguiente nivel de síntesis, es obtenida por:

$$c_{i-1,k}[n] = c'_{i,k}[n] + d'_{i,k}[n]. \quad (\text{B.51})$$

ALGORITMO B.3: Algoritmo de síntesis de descomposición
--

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Se toma la señal <math>c_{N,k}[n]</math> y <math>d_{N,k}[n]</math> y se calcula <math>c_{N-1,k}</math>, con (B.47) - (B.51).</li> <li>2. Ahora se toma <math>c_{i,k}</math> y <math>d_{i,k}[n]</math>, y se calcula <math>c_{i-1}[n]</math> y <math>d_{i-1}[n]</math> con (B.47) - (B.51), para <math>i = 1, 2, 3, \dots, N - 1</math>.</li> </ol> |
|--|

El algoritmo para recuperar un señal desde el nivel de síntesis  $N$ , es dado en el Algoritmo B.3.

## B.6. Comentarios

En este apéndice se presentan los fundamentos básicos de la teoría *wavelet*, que son necesarios para el entendimiento de la descomposición multiresolución de una señal, que en el caso de control clásico es la señal de error, que permite el desarrollo de un control PID *wavelet*. Además, se puede observar que la aproximación de una función B.40 está dada como una combinación lineal de *wavelets* ponderadas por los coeficientes  $c_i$ , esta representación permite tener una estructura de una red neuronal *wavelet*. En el siguiente apéndice se presenta los fundamentos de redes neuronales para tener un mayor entendimiento sobre las *wavenets* que son presentadas en el Apéndice D.

# Apéndice C

## Redes neuronales artificiales

El objetivo de este apéndice es dar un breve repaso sobre el tema de redes neuronales artificiales, las cuales son empleadas para generar redes *wavelets* y que éstas a su vez, se emplean dentro del esquema de un control PID *wavelet* para sintonizar sus parámetros. De aquí la importancia de este apéndice sobre el trabajo de la tesis, para dar un panorama muy general sobre las partes en que está conformado una red neuronal de simple capa, multicapa y las redes neuronales con funciones de base radial.

El apéndice se encuentra organizado de la siguiente manera: Primeramente en la Sección C.1 se presenta una introducción sobre el origen del estudio de las redes neuronales artificiales, posteriormente se muestra el esquema del perceptrón de simple capa y multicapa, en la Sección C.3 se muestra el esquema de las redes de base radial, y por último en la Sección C.4 se dan los comentarios referentes a este apéndice.

### C.1. Introducción

Una red neuronal es una red interconectada de elementos de procesamiento simple, i.e. escalados y filtrados. Estos elementos de procesamiento interactúan a lo largo de rutas estrechas las cuales cuando son adaptadas adecuadamente pueden producir colectivamente un comportamiento complejo deseado. El diseño de la red neuronal fue inspirado por la investigación biológica de cómo trabaja el cerebro humano. El objetivo de la red neuronal es imitar a las neuronas del cerebro humano por el enlace de muchos procesadores simple, llamadas neuronas artificiales o nodos, las cuales se interconectan a través de los pesos [30, 87].

Uno de los primeros modelos de neuronas fue desarrollado por McCulloch y Walter Pitts, que era un modelo simplificado de una neurona biológica. El modelo consiste de múltiples entradas y una salida con una unidad central de procesamiento CPU [47].

La Figura C.1 muestra el modelo para una neurona, la cual es descrita por

$$y = f \left( \sum_{l=1}^L w_l x_l - v_t \right), \quad (\text{C.1})$$

donde  $x_l$  es la  $l$ -ésima señal de entrada,  $w_l$  es el peso que conecta la  $l$ -ésima entrada con el CPU,  $v_t$  es un umbral o bias, que proporciona una tendencia a la función de activación  $f(\cdot)$ , y  $y$  es la señal de salida de la neurona. En 1949, Hebb sugiere una simple fórmula matemática para adaptar los cambios de los pesos en proporción a la actividad entre la pres-inápsis y post-sinápsis de la neurona

$$\Delta w_i(k) = \mu y(k) x_i(k), \quad (\text{C.2})$$

donde  $0 < \mu \in \mathbb{R}$  es la velocidad de aprendizaje para todo el tiempo  $k$ .

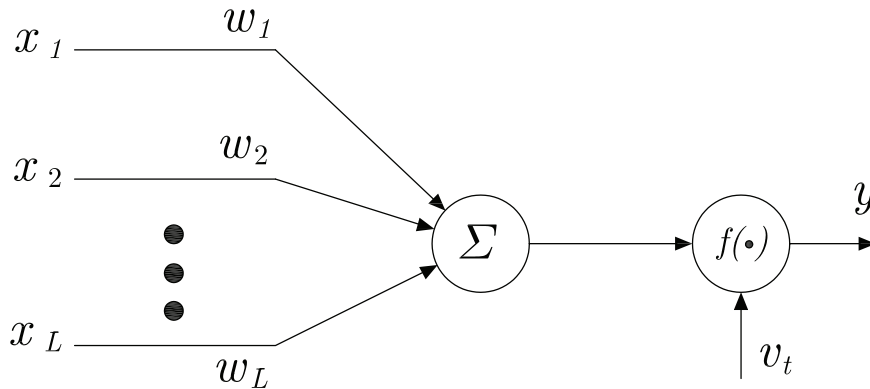


Figura C.1: Modelo de McCulloch-Pitts de una neurona, donde  $x_l$ ,  $w_l$ ,  $f(\cdot)$ ,  $y$ , y  $v_t$ , son la  $l$ -ésima señal de entrada, el peso que conecta la  $l$ -ésima entrada con el CPU, la función de activación, la señal de salida y el umbral, respectivamente.

## C.2. El perceptrón y el percentrón multicapa

Otro de los modelos de neuronas que mejora al modelo propuesto por McCulloch y Pitts, es el perceptrón. En 1958, Rosenblat demostró la eficacia de este modelo sobre algunas aplicaciones prácticas [73], el perceptrón es una conexión de nivel simple de neuronas de McCulloch-Pitts algunas veces llamadas redes de simple capa. Esta red es capaz de separar linealmente los vectores de entrada en patrones de clases por un hiperplano. La Figura C.2 muestra la red neuronal del tipo percentrón de una capa,



y su representación matemática de la  $l$ -ésima salida está dada por

$$y_l = f \left( \sum_{j=1}^J w_{j,l} x_j - v_t \right), \quad (\text{C.3})$$

donde  $l = 1, 2, \dots, L$ ,  $j = 1, 2, \dots, J$ ,  $L$  es el número de nodos en la salida y  $J$  el número de entradas, y la regla de aprendizaje es basada en el ajuste de los pesos en proporción al error entre las neuronas de salida y la salida deseada, la adaptación de los pesos es dada por:

$$\Delta w_{j,l}(k) = \mu [d_l(k) - y_l(k)] x_j(k), \quad (\text{C.4})$$

donde  $0 < \mu \in \mathbb{R}$  es la velocidad de aprendizaje para todo el tiempo  $k$  y  $d_l$  es la salida deseada en el nodo  $l$  al tiempo  $k$ .

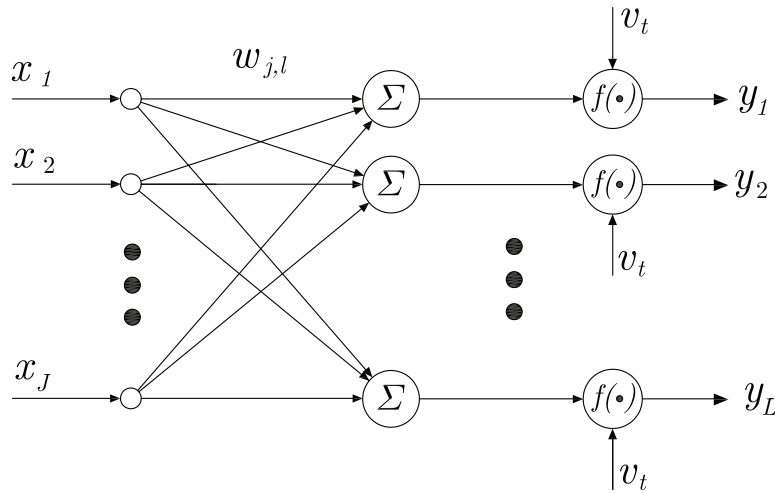


Figura C.2: Estructura de un perceptrón de una capa, donde  $x_j$ ,  $w_{j,l}$ ,  $f(\cdot)$ ,  $y_l$ ,  $v_t$ , son la  $j$ -ésima señal de entrada, el peso que conecta la  $j$ -ésima entrada con  $l$ -ésimo nodo de la capa de salida, la función de activación, la  $l$ -ésima señal de salida y el umbral, respectivamente.

Para procesos donde la red de simple capa no es la adecuada por sus limitaciones fundamentales que posee [30], las redes multicapas fueron sugeridas por Minsky y Papert en 1969. El perceptrón multicapa introduce una o más capas ocultas, llamadas neuronas ocultas. La función de las neuronas ocultas es el de intervenir entre las entradas externas y la salida de la red. La Figura C.3 muestra un perceptrón multicapa de tres capas, que consta de una capa de entrada, una capa oculta y una capa de salida. Los nodos de entrada en la capa de entrada consiste de  $J$  elementos de los patrones los cuales constituyen las señales de entrada aplicado a  $L$  neuronas en la primera

capa oculta. La señal de salida de las  $L$  neuronas en la capa final oculta  $H$  de la red neuronal constituye la respuesta completa de la red para los patrones suministrados en los nodos de entrada. La función  $f(\cdot)$ , es llamada función de activación, la cual define la salida de una neurona en términos del nivel de actividad a su entrada (rangos de -1 a 1 o 0 a 1). En la Tabla C.1 se dan algunos ejemplos de funciones de activación, donde las más empleadas son la función sigmoideal y la función de la tangente hiperbólica.

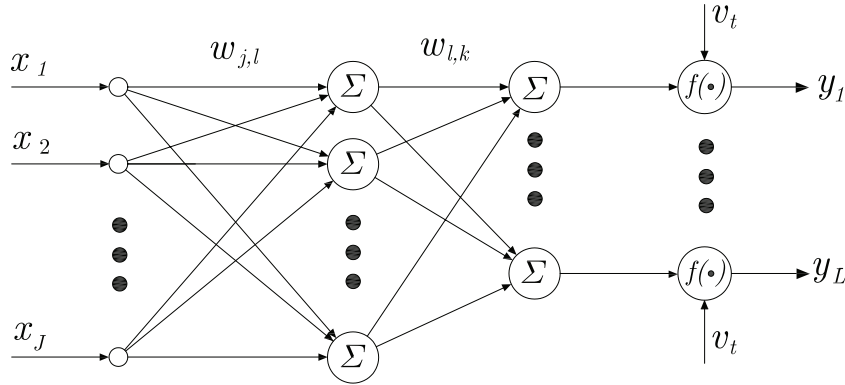


Figura C.3: Estructura de un perceptrón multicapa de tres capas, donde  $x_j$ ,  $w_{j,l}$ ,  $w_{l,k}$ ,  $f(\cdot)$ ,  $y_l$ ,  $v_t$ , son la  $j$ -ésima señal de entrada, el peso que conecta la  $j$ -ésima entrada con  $l$ -ésimo nodo en la primera capa, el peso que conecta el  $l$ -ésimo nodo de la primera capa con el  $k$ -ésimo nodo de la capa de salida, la función de activación, la  $l$ -ésima señal de salida y el umbral, respectivamente.

Un método para el aprendizaje supervisado que actualiza los pesos de la red, es el algoritmo *backpropagation* [30], desarrollado por R. Hinton y Williams en 1986. El algoritmo emplea un método interactivo del gradiente descendente de minimización, la cual minimiza el error cuadrático medio entre la salida deseada y la salida de la red, definido como

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{l=1}^L e^2(n), \quad (\text{C.5})$$

con

$$e_l(n) = d_l(n) - y_l^{(H)}(n), \quad (\text{C.6})$$

donde  $L$  es el número de neuronas en la capa de salida,  $N$  es el número de datos de entrenamiento en la entrada,  $n$  es el número de interacciones, cuando  $H$  es igual a la capa de salida, se tiene

$$v_l^{(h)}(n) = \sum_{j=1}^J w_{l,j}^{(h)}(n) y_j^{(h-1)}(n), \quad (\text{C.7})$$

Nombre	Definición
Lineal	$f(x) = \iota x$
Paso	$f(x) = \begin{cases} \beta, & \text{si } x \geq x_k, \\ \delta, & \text{si } x < x_k, \end{cases}$ comúnmente $\beta = 1, \delta = 0, x_k = 0,$
Rampa	$f(x) = \begin{cases} \rho, & \text{si } x > \rho, \\ x, & \text{si }  x  < \rho, \\ -\rho, & \text{si } \leq -\rho, \end{cases}$
Sigmoidal	$f(x) = \frac{1}{1+e^{-2\alpha x}}, \alpha > 0,$
Hiperbólica	$f(x) = \tanh(\gamma x) = \frac{1-e^{-2\gamma x}}{1+e^{-2\gamma x}}, \gamma > 0,$
Racional	$f(x) = \begin{cases} \frac{x^2}{1+x^2}, & \text{si } x > 0, \\ 0, & \text{en otro caso,} \end{cases}$
Gaussiana	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)},$ donde $\mu$ es el centro de la función Gaussiana y $\sigma$ es la media de propagación.

Tabla C.1: Algunos ejemplos comunes de funciones de activación

donde  $y_j^{(h-1)}(n)$  es la señal de función de la neurona  $j$  en la capa previa  $h - 1$  a la interacción  $n$ ,  $w_{l,j}^{(h)}(n)$  es el peso que conecta la neurona  $l$  en la capa  $h$  con la neurona  $j$  en la capa  $h - 1$ .

Entonces la señal de salida de la neurona  $l$  en la capa  $h$  es

$$y_l^{(h)} = f(v_l^l(n)), \tag{C.8}$$

$f(\cdot)$  es la función de activación, si la neurona  $l$  está en la primera capa oculta ( $h = 1$ ), entonces el conjunto  $y_l^{(0)}(n) = x_l(x)$ .

Haciendo el cálculo hacia atrás, se tiene

$$\delta_l(n) = -\frac{\partial E_l}{\partial v_l}, \tag{C.9}$$

el cual es llamado el error local o los gradientes locales, se puede simplificar (C.9) como:

(b) para la neurona  $l$  en la capa de salida  $H$

$$\delta_l^H(n) = e_l^H(n) f'(v_l^H(n)), \tag{C.10}$$

(b) para la neurona  $l$  en la capa de salida  $h$ ,

$$\delta_l^h(n) = f'(v_l^h(n)) \sum_k \delta_k^{(h+1)}(n) w_{k,l}^{(h+1)}(n). \quad (\text{C.11})$$

donde  $f'(\cdot)$  es la derivada de la función de activación con respecto a  $v(n)$ ,

La actualización de los pesos  $w$  de la red en la capa  $h$  se actualizan de acuerdo a la *regla delta generalizada*:

$$w_{i,j}^{(h)}(n+1) = w_{i,j}^{(h)}(n) + \mu \gamma_l^{(h)}(n) y_j^{(h-1)}(n), \quad (\text{C.12})$$

donde:  $0 < \mu \in \mathbb{R}$  es la velocidad de aprendizaje, comúnmente de 0.01. Si después de actualizar los pesos, el error  $E$  no se minimiza, una nueva interacción (época) se requiere hasta que el error  $E$  sea mínimo.

### C.3. Redes neuronales con funciones de base radial

Las redes neuronales con funciones de base radial RBF fueron empleadas por primera vez por Broomhed y Lowe en 1988 [30]. Las redes de RBF básicamente están formadas de tres capas completamente diferentes como se muestra en la Figura C.4; una capa de entrada, una capa oculta de dimensiones suficientes y una capa de salida. La transformación desde la capa oculta a la capa de salida es en forma lineal, sin embargo, la transformación desde la entrada a la capa oculta es no lineal. Cada neurona o nodo en la capa oculta forma una combinación lineal de la función base, la cual produce una respuesta bien localizada con respecto a la señal de entrada. Esto es que la RBF producen una respuesta significativa diferente de cero únicamente cuando la entrada cae dentro de una pequeña región localizada en el espacio de entrada.

La función base más común de la RBF es la función Gaussiana, la cual tiene la siguiente representación

$$\varphi_l(\mathbf{X}) = e^{-\left(\frac{(\mathbf{x}-\mathbf{c}_l)^T(\mathbf{x}-\mathbf{c}_l)}{2\sigma_l^2}\right)}, \quad (\text{C.13})$$

donde  $\varphi_l$  es la salida del  $l$ -ésimo nodo en la capa oculta,  $\mathbf{X}$  es el patrón de entrada,  $\mathbf{C}_l$  es el vector de pesos para el  $l$ -ésimo nodo en la capa oculta, i.e. es el centro de la función Gaussiana para el nodo  $l$ ;  $\sigma_l$  es el parámetro de normalización o la media de propagación para el  $l$ -ésimo nodo, y  $l = 1, 2, \dots, L$ . Siendo  $L$  el número de neuronas en la capa oculta. La salida  $i$ -ésima de la red de funciones de base radial está dada por

$$y_i = \mathbf{W}_i^T \varphi(\mathbf{X}_i), \quad (\text{C.14})$$

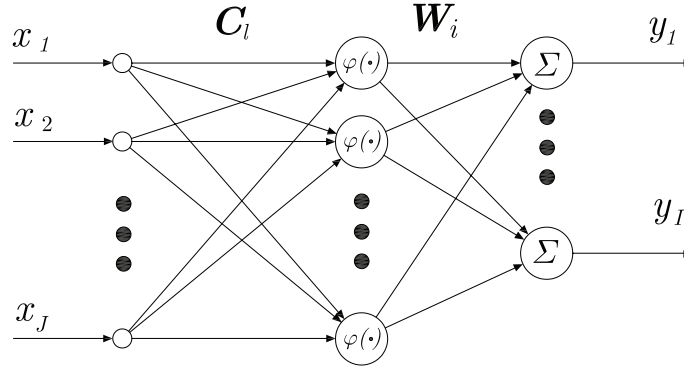


Figura C.4: Estructura de una red neuronal con funciones de base radial, donde  $x_j$ ,  $\mathbf{C}_l$ ,  $\mathbf{W}_i$ ,  $\varphi(\cdot)$  y  $y_i$  son la entrada en el nodo  $j$ , el vector de pesos del nodo  $l$ , el vector de pesos del nodo  $i$ , la función de activación y la salida en el nodo  $i$ , respectivamente.

donde  $y_i$  es la salida del  $i$ -ésimo nodo en la capa de salida,  $\mathbf{W}_i$  es el vector de pesos del nodo  $i$ ,  $\varphi(\mathbf{X})$  es el vector de salidas desde la capa oculta, y  $i = 1, 2, \dots, I$ . Siendo  $I$  el número de nodos en la capa de salida.

Existen dos forma comunes de calcular la media de propagación  $\sigma_l^2$  de los datos asociado a cada nodo:

1. Encuentra la media de propagación desde el conjunto de todos los patrones de entrenamiento agrupados con cada centro *cluster*, i.e., los hace igual a la distancia del centro *cluster* y los patrones de entrenamiento:

$$\sigma_l^2 = \frac{1}{N_l} \sum_{\mathbf{X}_k \in \mathbf{C}_l} (\mathbf{X}_k - \mathbf{C}_l)^T (\mathbf{X}_k - \mathbf{C}_l), \quad l = 1, 2, \dots, k, \dots, L, \quad (\text{C.15})$$

donde  $N_l$  es el número de patrones que pertenecen al  $l$ -ésimo *cluster*,  $k$  es el índice de un patrón que pertenece a el  $l$ -ésimo *cluster*.

2. Encontrar la media de propagación desde el centro (heurística del vecino  $p$  más cercano):

$$\sigma_l^2 = \frac{1}{L} \sum_{k=1}^L (\mathbf{C}_k - \mathbf{C}_l)^T (\mathbf{C}_k - \mathbf{C}_l), \quad l = 1, 2, \dots, k, \dots, L, \quad (\text{C.16})$$

Una elección popular de algoritmos de *clustering* usados en el entrenamiento de redes RBF (para dar una clase  $k$ ) es el algoritmo Lloyd generalizado o el algoritmo de *clustering*  $K$ -means. El algoritmo diseñado por J. MacQueen en 1967 provee un mecanismo simple para minimizar la suma de los errores cuadráticos con  $k$  *clusters*, donde cada *cluster* consiste en un conjunto de  $N_l$  muestras  $x_1, x_2, \dots, K, \dots, N_l$ , que son similares con cada una de las otras. El algoritmo  $K$ -means es:

ALGORITMO C.2: Algoritmo *K-means*

1. Seleccione un conjunto de *clusters*  $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K, \dots, \mathbf{C}_k\}$  arbitrario.
2. Asigne los  $N_l$  muestras a el  $k$  *clusters* usando la regla de distancia mínima Euclidiana: Si  $\|\mathbf{X} - \mathbf{C}_l\| = \min_l \|\mathbf{X} - \mathbf{C}_l\| \Rightarrow \mathbf{X} \in \mathbf{C}_l$ .
3. Calcular el nuevo centro *cluster*  $\mathbf{C}_k = \frac{1}{N_l} \sum_{\mathbf{X}_k \in \mathbf{C}_l} \mathbf{X}_k$
4. Si cualquier centro *cluster* cambia, regrese al paso 2; en otro caso, terminar.

## C.4. Comentarios

En este apéndice se da una introducción sobre las redes neuronales artificiales, las cuales junto con la teoría *wavelet* vistas en el Apéndice B, forma las redes *wavelets* (*wavenets*), las cuales son dadas en el Apéndice D, y sirven para el desarrollo del control PID *wavenet* que es el tema principal de esta tesis.

# Apéndice D

## Teoría *wavenet*

El objetivo de este apéndice es desarrollar los algoritmos que hacen uso de la combinación de redes neuronales y las *wavelets*, para modificar las ganancias del controlador *wavelet* propuesto. Las redes neuronales *wavelets* adaptable, no sólo son empleadas para control adaptable, si no también, para la identificación de sistemas no lineales y aproximación de funciones no lineales, debido a la gran capacidad de rápido aprendizaje y a las propiedades de las *wavelets*. Un filtro IIR es empleado en cascada, para garantizar la estabilidad y la convergencia de la red neuronal *wavelet* adaptable.

### D.1. Introducción

Combinando la teoría de la transformada *wavelet* con el concepto básico de redes neuronales, se propone un nuevo mapeo de red llamado red neuronal *wavelets* adaptable o *wavenet* como una alternativa a las redes neuronales de alimentación hacia adelante para aproximar funciones arbitrarias no lineales. Los algoritmos *wavenet* consisten básicamente de dos procesos: la auto-construcción de las redes y la minimización del error. En el primer proceso, las estructuras de las redes aplicadas para la representación son determinadas usando análisis *wavenet*. La red gradualmente combina unidades ocultas para cubrir eficiente y suficientemente la región tiempo-frecuencia ocupada por una meta dada. Simultáneamente, los parámetros de la red son actualizados para conservar la topología de la red y aprovechar el proceso posterior. En el segundo proceso, las aproximaciones de los errores instantáneos son minimizadas usando una técnica de adaptación basada en los algoritmos LMS. Los parámetros de la red inicializada son actualizados usando el método del gradiente de pasos descendentes. Cada unidad oculta tiene una ventana cuadrada en el plano tiempo-frecuencia. La regla de optimización es solamente aplicada a las unidades ocultas donde el punto seleccionado cae en sus ventanas. Por lo tanto, el costo del aprendizaje puede ser

reducido. Todas estas ventajas que se tienen de las redes *wavenets* son aprovechadas en el controlador *wavenet* que se propone en este trabajo de tesis.

## D.2. Algoritmos *wavenets*

Como se vio en la Sección B.4, una función  $f(t) \in L^2(\mathbb{R})$  puede ser aproximada mediante una combinación lineal empleando (B.40), la cual tiene similitud con una red neuronal de base radial. La Figura D.1 es la arquitectura de la red neuronal *wavelet* adaptable, que aproxima cualquier señal deseada  $y(t)$  mediante la generalización de una combinación lineal de un conjunto de *wavelets* hijas  $\psi(\tau)$ , donde éstas son generadas por una dilatación  $a$  y una traslación  $b$  de la *wavelet* madre  $\psi(t)$ :

$$\psi(\tau) = \psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbb{R}, \quad \tau = \frac{t-b}{a}, \quad (\text{D.1})$$

con el factor de dilatación  $a > 0$ . Se puede observar que (D.1) es similar a la Definición 39, Apéndice B pero sin la normalización de energía.

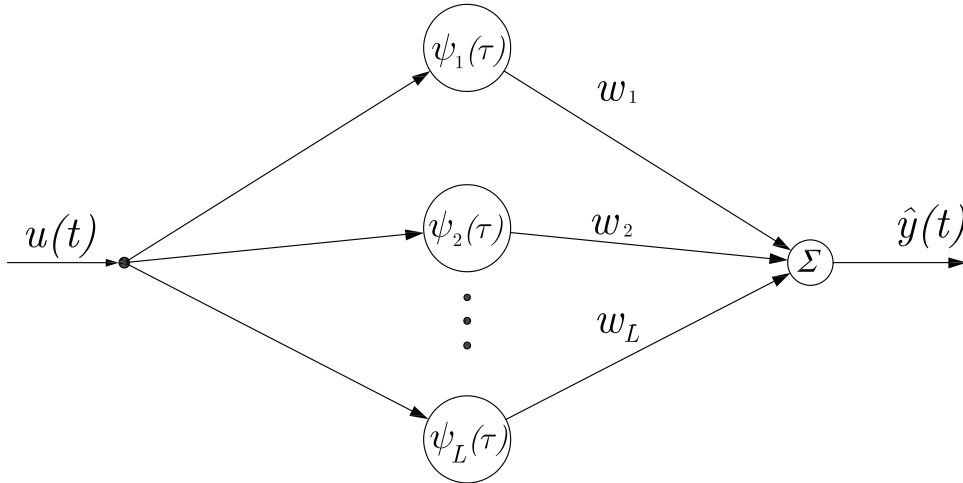


Figura D.1: Diagrama de una red neuronal, donde  $u(t)$ ,  $\hat{y}$  son la señal entrada, la señal de salida de la red neuronal, respectivamente y  $\psi(\tau)$ ,  $w$  son la función de activación y los pesos que hay entre las neuronas, respectivamente.

Y, para lograr la aproximación, se asume que la función de salida de la red satisface la condición de admisibilidad y la red aproxima suficientemente la meta, es decir, que la región tiempo-frecuencia es cubierta efectivamente por sus  $L$  ventanas. La señal



aproximada de la red  $\hat{y}(t)$  puede ser representada por:

$$\hat{y}(t) = u(t) \sum_{l=1}^L w_l \psi_l(\tau), \quad y, u, w \in \mathbb{R}, \quad (\text{D.2})$$

en la cual  $\psi_l(\tau) = \psi\left(\frac{t-b_l}{a_l}\right)$  para  $l = 1, 2, \dots, L$ , donde  $L \in \mathbb{Z}$  es el número de neuronas en la capa de la red neuronal wavenet.

Definiendo los vectores  $\mathbf{W}$ ,  $\mathbf{A}$ ,  $\mathbf{B}$  y  $\Psi$  como

$$\mathbf{W} \triangleq [w_1 \ w_2 \ \dots \ w_l \ \dots \ w_{L-1} \ w_L]^T, \quad (\text{D.3})$$

$$\mathbf{A} \triangleq [a_1 \ a_2 \ \dots \ a_l \ \dots \ a_{L-1} \ a_L]^T, \quad (\text{D.4})$$

$$\mathbf{B} \triangleq [b_1 \ b_2 \ \dots \ b_l \ \dots \ b_{L-1} \ b_L]^T, \quad (\text{D.5})$$

$$\Psi(\tau) \triangleq [\psi_1(\tau) \ \psi_2(\tau) \ \dots \ \psi_l(\tau) \ \dots \ \psi_{L-1}(\tau) \ \psi_L(\tau)]^T, \quad (\text{D.6})$$

ahora (D.2), puede ser representada por:

$$\hat{y}(t) = \Psi^T(\tau) \mathbf{W} u(t), \quad (\text{D.7})$$

y, como una *wavenet* es una red local en la cual la función de salida está bien localizada en ambos tiempo-frecuencia. En adición, una red local doble puede ser lograda por la combinación de una red neuronal en cascada con un filtro de respuesta infinita al impulso (IIR), el cual provee un método computacional eficiente para el aprendizaje del sistema [15]. En la Figura D.2 se muestra la estructura del filtro IIR y en la Figura D.3 se muestra la estructura final, de la *wavenet* con el filtro IIR.

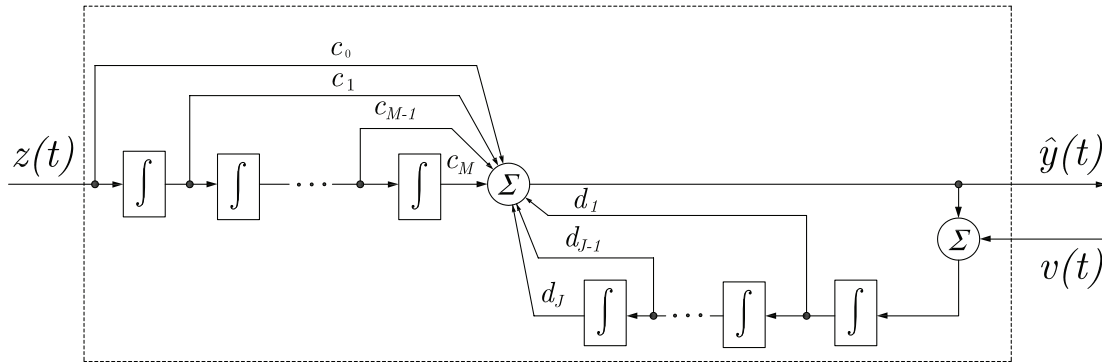


Figura D.2: Diagrama del filtro IIR, donde  $z(t)$ ,  $\hat{y}$  son la señal de entrada, la señal de salida del filtro IIR, respectivamente y  $c, d$ , son los coeficientes del filtro.

Definiendo

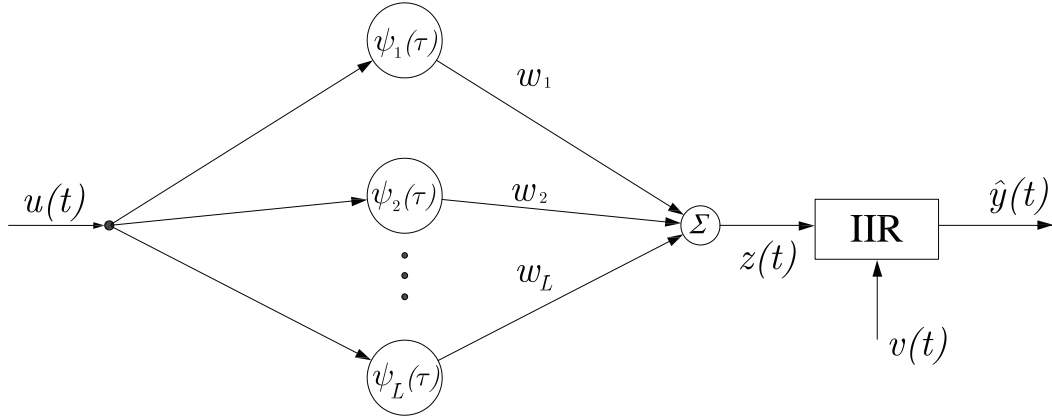


Figura D.3: Diagrama de la *wavenet* con el filtro IIR, donde  $u(t)$ ,  $\hat{y}$  son la señal de entrada, la señal de salida de la *wavenet* con un filtro IIR, respectivamente,  $a$ ,  $b$ ,  $w$  y  $c$ ,  $d$ , son los parámetros de la *wavenet* y coeficientes del filtro IIR, respectivamente.

$$\mathbf{C} \triangleq [c_0 \ c_1 \ \cdots \ c_m \ \cdots \ c_{M-1} \ c_M]^T, \quad (\text{D.8})$$

$$\mathbf{D} \triangleq [d_1 \ d_2 \ \cdots \ d_j \ \cdots \ d_{J-1} \ d_J]^T, \quad (\text{D.9})$$

$$\mathbf{Z}(t) \triangleq [z(t) \ z(t-1) \ \cdots \ z(t-m) \ \cdots \ z(t-M+1) \ z(t-M)]^T, \quad (\text{D.10})$$

$$\hat{\mathbf{Y}}(t) \triangleq [\hat{y}(t-1) \ \hat{y}(t-2) \ \cdots \ \hat{y}(t-j) \ \cdots \ \hat{y}(t-J+1) \ \hat{y}(t-J)]^T, \quad (\text{D.11})$$

con  $c$ ,  $d$ ,  $z$ ,  $v \in \mathbb{R}$ . Ahora la señal aproximada  $\hat{y}(t)$  con el filtro IIR en cascada, puede ser expresada en forma vectorial como

$$\hat{y}(t) = \mathbf{C}^T \mathbf{Z}(t) u(t) + \mathbf{D}^T \hat{\mathbf{Y}}(t) v(t), \quad (\text{D.12})$$

donde

$$z(t) = \mathbf{\Psi}^T(\tau) \mathbf{W}. \quad (\text{D.13})$$

Los parámetros de la *wavenet* IIR  $w$ ,  $a$ ,  $b$ ,  $c$  y  $d$  pueden ser optimizados en el sentido de LMS minimizando la función de energía del error  $E$ , definida como

$$E = \frac{1}{2} \sum_{t=1}^{\tau} e^2(t), \quad (\text{D.14})$$

$$e(t) = y(t) - \hat{y}(t), \quad (\text{D.15})$$

donde  $y(t)$  es la respuesta deseada.

Se minimiza a  $E$  con el método del gradiente de pasos descendentes, el cual requiere los gradientes  $\frac{\partial E}{\partial \mathbf{W}}$ ,  $\frac{\partial E}{\partial \mathbf{A}}$  y  $\frac{\partial E}{\partial \mathbf{B}}$  de los parámetros de la red *wavenet* y  $\frac{\partial E}{\partial \mathbf{C}}$ ,  $\frac{\partial E}{\partial \mathbf{D}}$  de los coeficientes de la estructura IIR, los cuales son expresados como:

$$\frac{\partial E}{\partial \mathbf{W}} = \left[ \frac{\partial E}{\partial w_1} \frac{\partial E}{\partial w_2} \cdots \frac{\partial E}{\partial w_l} \cdots \frac{\partial E}{\partial w_{L-1}} \frac{\partial E}{\partial w_L} \right]^T, \quad (\text{D.16})$$

$$\frac{\partial E}{\partial \mathbf{A}} = \left[ \frac{\partial E}{\partial a_1} \frac{\partial E}{\partial a_2} \cdots \frac{\partial E}{\partial a_l} \cdots \frac{\partial E}{\partial a_{L-1}} \frac{\partial E}{\partial a_L} \right]^T, \quad (\text{D.17})$$

$$\frac{\partial E}{\partial \mathbf{B}} = \left[ \frac{\partial E}{\partial b_1} \frac{\partial E}{\partial b_2} \cdots \frac{\partial E}{\partial b_l} \cdots \frac{\partial E}{\partial b_{L-1}} \frac{\partial E}{\partial b_L} \right]^T, \quad (\text{D.18})$$

$$\frac{\partial E}{\partial \mathbf{C}} = \left[ \frac{\partial E}{\partial c_1} \frac{\partial E}{\partial c_2} \cdots \frac{\partial E}{\partial c_m} \cdots \frac{\partial E}{\partial c_{M-1}} \frac{\partial E}{\partial c_M} \right]^T, \quad (\text{D.19})$$

$$\frac{\partial E}{\partial \mathbf{D}} = \left[ \frac{\partial E}{\partial d_1} \frac{\partial E}{\partial d_2} \cdots \frac{\partial E}{\partial d_j} \cdots \frac{\partial E}{\partial d_{J-1}} \frac{\partial E}{\partial d_J} \right]^T, \quad (\text{D.20})$$

donde:

$$\frac{\partial E}{\partial w_l} = - \sum_{t=1}^{\tau} e(t)u(t) \sum_{m=0}^M c_m \psi(\tau - m), \quad (\text{D.21})$$

$$\frac{\partial E}{\partial b_l} = - \sum_{t=1}^{\tau} e(t)u(t) \sum_{m=0}^M c_m w_l \frac{\partial \psi(\tau - m)}{\partial b_l}, \quad (\text{D.22})$$

$$\frac{\partial E}{\partial a_l} = - \sum_{t=1}^{\tau} e(t)u(t) \sum_{m=0}^M c_m w_l \tau \frac{\partial \psi(\tau - m)}{\partial b_l} = \tau \frac{\partial E}{\partial b_l}, \quad (\text{D.23})$$

$$\frac{\partial E}{\partial c_m} = - \sum_{t=1}^{\tau} e(t)u(t)z(t - m), \quad (\text{D.24})$$

$$\frac{\partial E}{\partial d_j} = - \sum_{t=1}^{\tau} e(t)v(t)\hat{y}(t - j), \quad (\text{D.25})$$

de manera similar, los cambios incrementales de cada parámetro son sólo el negativo de sus gradientes:

$$\Delta \mathbf{W} = -\frac{\partial E}{\partial \mathbf{W}}, \quad (\text{D.26})$$

$$\Delta \mathbf{A} = -\frac{\partial E}{\partial \mathbf{A}}, \quad (\text{D.27})$$

$$\Delta \mathbf{B} = -\frac{\partial E}{\partial \mathbf{B}}, \quad (\text{D.28})$$

$$\Delta \mathbf{C} = -\frac{\partial E}{\partial \mathbf{C}}, \quad (\text{D.29})$$

$$\Delta \mathbf{D} = -\frac{\partial E}{\partial \mathbf{D}}, \quad (\text{D.30})$$

de este modo, cada coeficiente es actualizado acorde a la siguiente regla:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu_W \Delta \mathbf{W}, \quad (\text{D.31})$$

$$\mathbf{A}(n+1) = \mathbf{A}(n) + \mu_A \Delta \mathbf{A}, \quad (\text{D.32})$$

$$\mathbf{B}(n+1) = \mathbf{B}(n) + \mu_B \Delta \mathbf{B}, \quad (\text{D.33})$$

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu_C \Delta \mathbf{C}, \quad (\text{D.34})$$

$$\mathbf{D}(n+1) = \mathbf{D}(n) + \mu_D \Delta \mathbf{D}, \quad (\text{D.35})$$

donde  $\mu \in \mathbb{R}$  es el coeficiente de velocidad de aprendizaje, para cada uno de los parámetros.

### D.3. Comentarios

En este apéndice se resumen los resultados básicos de las redes *wavenets* para la aproximación de funciones, es decir dada una señal de entrada  $u(t)$  arbitraria se aproximó por  $\hat{y}(t)$  que es la salida de la red neuronal *wavenet*. Esta configuración de red es empleada en el Capítulo 4 para la identificación de la planta y otra más para la sintonización del controlador PID *wavelet*.

# Apéndice E

## Implementación de un filtro *wavelet*

### E.1. Introducción

Otra de las aplicaciones que se generaron de esta tesis a partir de la teoría *wavelet*, fue la implementación de un filtro *wavelet* para decrecer frecuencias en las señales de velocidad articular de los decodificadores ópticos un dispositivo háptica PHANTOM premium 1.0 [36, 63], el cual está conformado de posicionadores mecánicos con propósitos de retroalimentación de fuerza kinestética. Estos posicionadores mecánicos contiene decodificadores ópticos para la lectura de velocidad articular para fines de control, las lecturas obtenidas a través de los decodificadores ópticos están contaminadas con señales de ruido de alta frecuencia, por lo que un filtro digital sería una solución adecuada, para disminuir dicho ruido.

### E.2. Filtro *wavelet*

Un filtro *wavelet* es implementado basado en el análisis multiresolución, considerando que (B.38) permite descomponer una señal  $f(x)$  a un nivel de resolución  $N$ , el resultado de esta descomposición en una señal de alta escala (baja frecuencia) y  $N$  señales de mediana y baja escala (mediana y alta frecuencia). Si consideramos a éstas últimas como señales no deseadas, i.e. señal de ruido. Una función  $f(x) = C_0$ , se descompone en dos señales  $C_1$  y  $D_1$ , donde  $C_1$  es una señal de baja frecuencia (escala alta) y  $D_1$  es una señal de alta frecuencia (baja escala), por lo tanto una señal  $f(x)$  a un nivel 1 de descomposición sin señal de baja escala es como se muestra en la Figura

E.1

$$f^*(x) = C_1, \quad (\text{E.1})$$

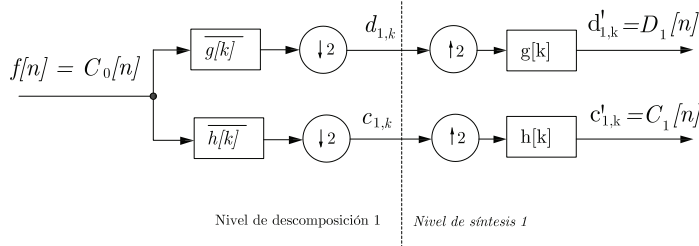


Figura E.1: Diagrama de descomposición de una señal  $f(x) = C_0$  en una señal de baja frecuencia (alta escala)  $C_1$  y una señal de alta frecuencia (baja escala)  $D_1$ .

la señal  $C_1$  se puede seguir descomponiendo de forma infinita hasta  $C_N$ , donde  $N$  es el nivel de descomposición. Entre mayor sea el nivel de descomposición, la señal  $f^*(x)$  será mejor filtrada

$$f^*(x) = \sum_{k \in \mathbb{Z}} c_{N,k} \phi_{N,k}(x) = C_N \quad (\text{E.2})$$

### E.3. Implementación

El filtro implementado sobre el dispositivo háptico PHANToM, es de nivel  $N=5$ , con coeficientes Daubechies ( $m = 7$ ) [11] y el diagrama del filtro es mostrado en la Figura E.2.

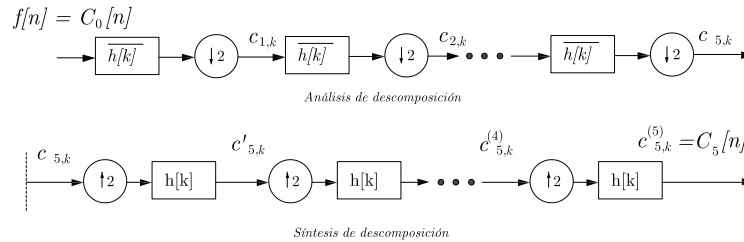


Figura E.2: Diagrama del filtro *wavelet* con un nivel de filtrado de  $N = 5$ .

En las Figuras E.3, E.4 y E.5, se muestran los resultados experimentales del filtro *wavelet* de velocidad con un nivel de descomposición de  $N = 5$ . Los resultados muestran las señales filtradas y no filtradas de las velocidades articulares de la interfaz PHANToM, en la se cual muestra la disminución del *chattering* en las tres señales de velocidad [36].

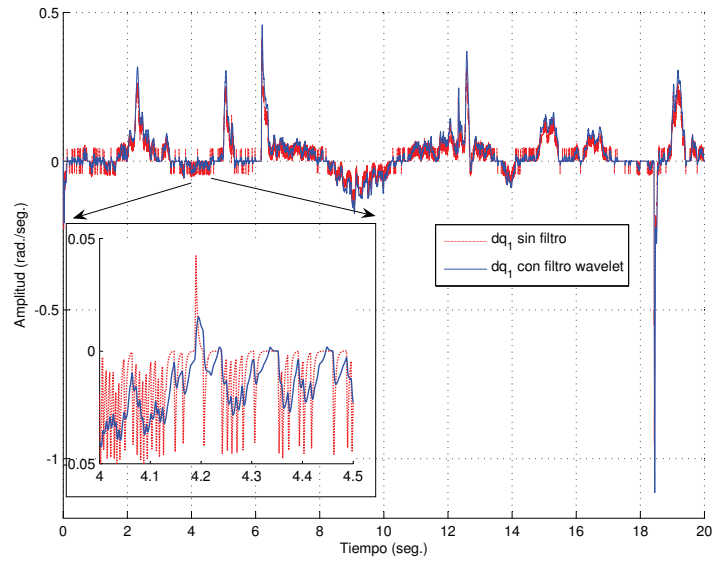


Figura E.3: Señal de velocidad articular  $\dot{q}_1$  con y sin filtro digital *wavelet*.

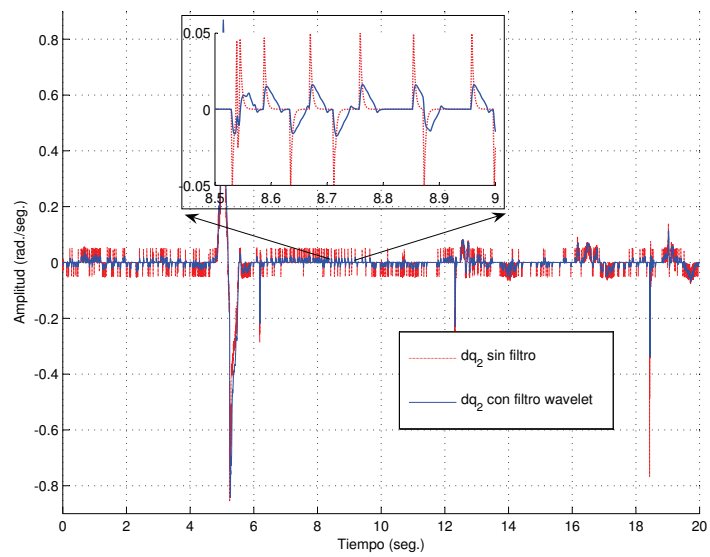


Figura E.4: Señal de velocidad articular  $\dot{q}_2$  con y sin filtro digital *wavelet*.

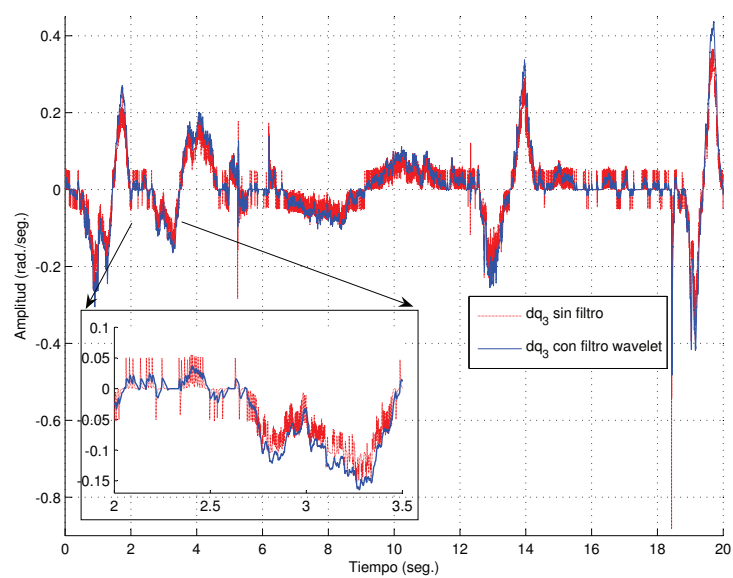


Figura E.5: Señal de velocidad articular  $\dot{q}_3$  con y sin filtro digital *wavelet*.



# Apéndice F

## Algoritmos programados en MATLAB y *simulink*

El objetivo de este apéndice es de mostrar los algoritmos empleados para la implementación para control PID *wavelet*, el filtro digital y el control PID *wavenet* desarrollados en esta tesis. Los algoritmos de los controladores PID *wavelet* y PID *wavenet* están programados en MATLAB 7.0 y *simulink*, mientras que los algoritmos del filtro digital *wavelet* están programados en C++.

### F.1. Algoritmos del PID *wavelet* empleados para la simulación

#### F.1.1. MATLAB

MATLAB F.1: Función de descomposición
<pre>function [a0, w0] = FWD(s) % Filtros de Descomposición y Reconstrucción g0 = [0.68301; 1.18301; 0.31699; -0.18301]; k = [0; 1; 2; 3]; g1 = flipud(g0) .* (-1).^k; h0 = flipud(g0)/2; h1 = flipud(g1)/2; % Proceso de Descomposición</pre>

```

% Sólo un nivel de Descomposición

x = conv(s, h0);
a0 = x(1 : 2 : length(x)); % muestreo a dos
x = conv(s, h1);
w0 = x(1 : 2 : length(x)); % muestreo a dos

```

---

MATLAB F.2: Función de síntesis

---

```

function a0 = FWS(a1, z)
% Opcion: 1.- g0 (h)
%          2.- g1 (g)
% Filtros de Descomposición y Reconstrucción
g0 = [0.68301; 1.18301; 0.31699; -0.18301];
k = [0; 1; 2; 3];
g1 = flipud(g0) .* (-1).^k;
h0 = flipud(g0)/2;
h1 = flipud(g1)/2;
% Proceso de Reconstrucción
if z == 1
    p = g0;
else
    p = g1;
end
% Reconstrucción a un sólo nivel
x = zeros(2 * length(a1), 1);
x(1 : 2 : 2 * length(a1)) = a1(1 : length(a1));
a = conv(x, p);
a0 = a(4 : length(a) - 4);
a0 = a0';

```

---

MATLAB F.3: Función para un nivel de descomposición  $N = 3$  de la señal de error  $e$ 


---

```

function eMR = ControlMR(ew)

% Vector Buffer de la señal de error a un nivel de descomposición  $N = 3$ 
% ew = [e(1, t), e(1, t - 1), e(1, t - 2), e(1, t - 3), e(1, t - 4), e(1, t - 5),
% e(1, t - 6), e(1, t - 7)];

```

```

% Descomposición de la señal
[a1, w1] = FWD(ew);
[a2, w2] = FWD(a1);
[a3, w3] = FWD(a2);
% Reconstrucción de las señales
a31 = FWS(a3, 1);
w31 = FWS(w3, 2);
a32 = FWS(a31, 1);
w32 = FWS(w31, 1);
w21 = FWS(w2, 2);
a33 = FWS(a32, 1);
w33 = FWS(w32, 1);
w22 = FWS(w21, 1);
w11 = FWS(w1, 2);
% Componentes de la señal de error
eMR = [a33(1, 1), w33(1, 1), w22(1, 1), w11(1, 1)];

```

Los algoritmos pueden ser usados desde línea de comandos ó desde un archivo *.m* de MATLAB, de la siguiente forma mostrada en MATLAB F.4.

MATLAB F.4: Empleando el control *wavelet*

```

% Ganancias empleadas para el control wavelet
K = [2.5, 12, 18, 0]; %K = [KH, KM1, KM2, KL]
% Señal de error a descomponer de longitud ebuf = 2^N
% donde N es el nivel de descomposición
ew = [error(1, n), error(1, n - 1), error(1, n - 2), error(1, n - 3),
      error(1, n - 4), error(1, n - 5), error(1, n - 6), error(1, n - 7)];
% Componentes de la señal de error
eMR = ControlMR(ew);
% Ley de control
u = K * eMR';

```

### F.1.2. *simulink*

Los diagramas a bloques del control PID *wavelet* son mostrados en las Figuras F.1, F.2. La función empleada en “MATLAB Function” en las Figuras F.1 y F.2 es dada por MATLAB F.5.

MATLAB F.5: Control *wavelet* para emplearse en *simulink*

```

% Descomposición wavelet
function [u] = deswave(block)

ebuff = 8;
e = zeros(1,ebuff);
for ii = 1 : 8
    e(1,ii) = block(ii);
end

% Descomposición de la señal
[a1,w1] = FWD(ew);
[a2,w2] = FWD(a1);
[a3,w3] = FWD(a2);
% Reconstrucción de las señales
a31 = FWS(a3,1);
w31 = FWS(w3,2);
a32 = FWS(a31,1);
w32 = FWS(w31,1);
w21 = FWS(w2,2);
a33 = FWS(a32,1);
w33 = FWS(w32,1);
w22 = FWS(w21,1);
w11 = FWS(w1,2);
% Componentes de la señal de error
eMR = [a33(1,1), w33(1,1), w22(1,1), w11(1,1)];
% *****
% Ley de control
u = [eMR];

```

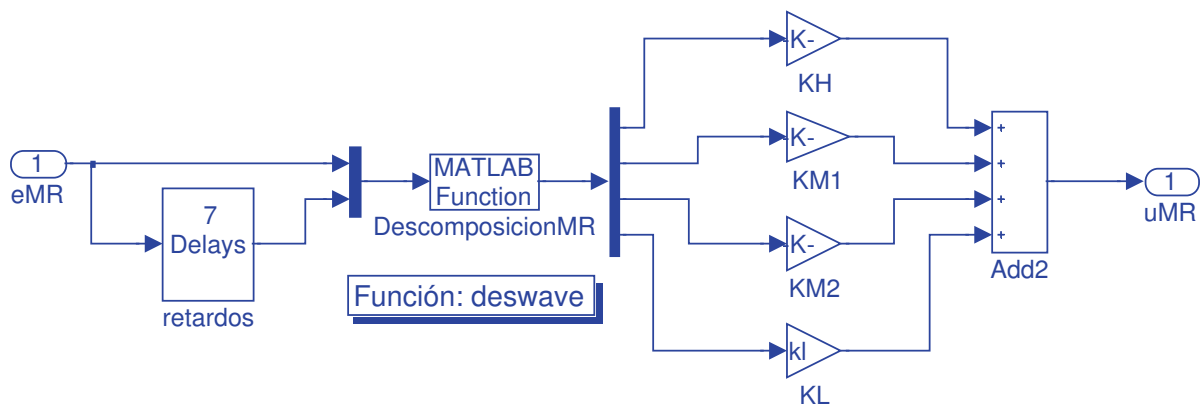


Figura F.1: Bloque del control *wavelet*.

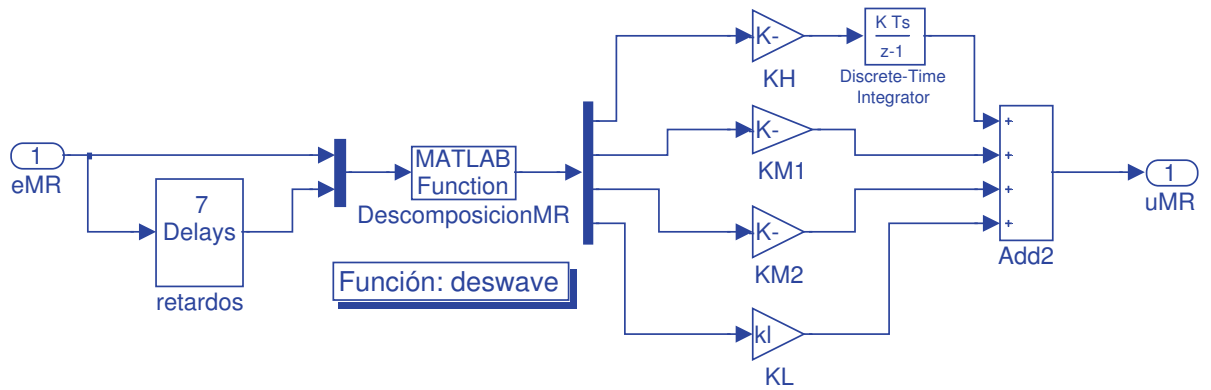


Figura F.2: Bloque del control *wavelet* tipo I.

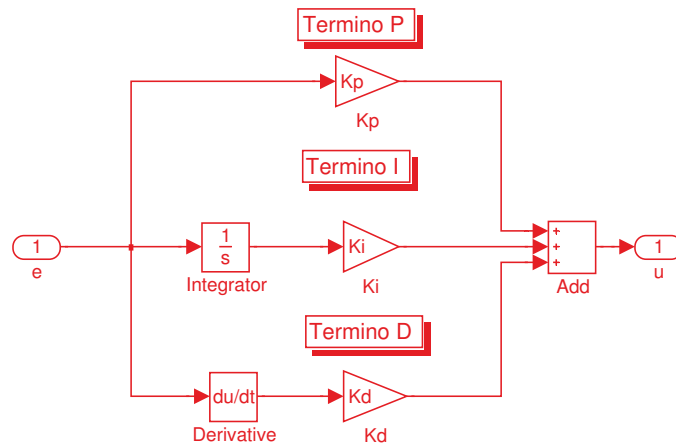


Figura F.3: Bloque del control PID clásico.

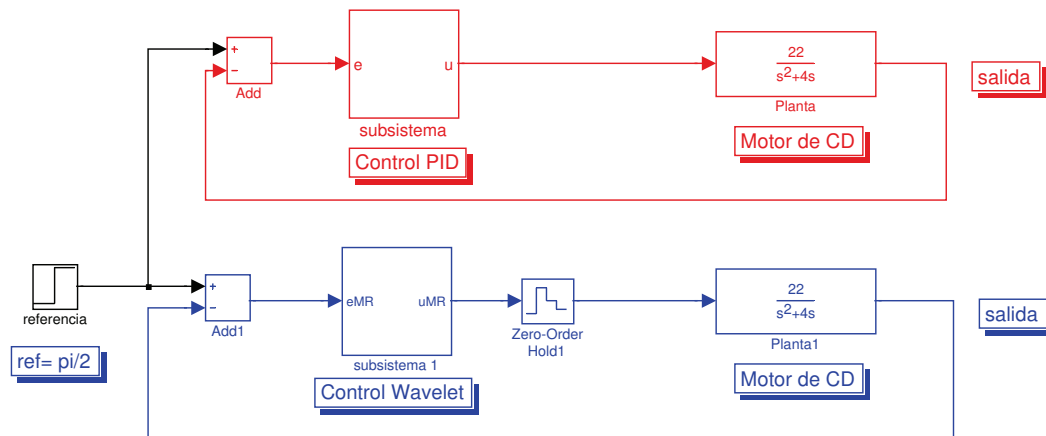


Figura F.4: Bloque para la simulación de los controles en un motor de CD.

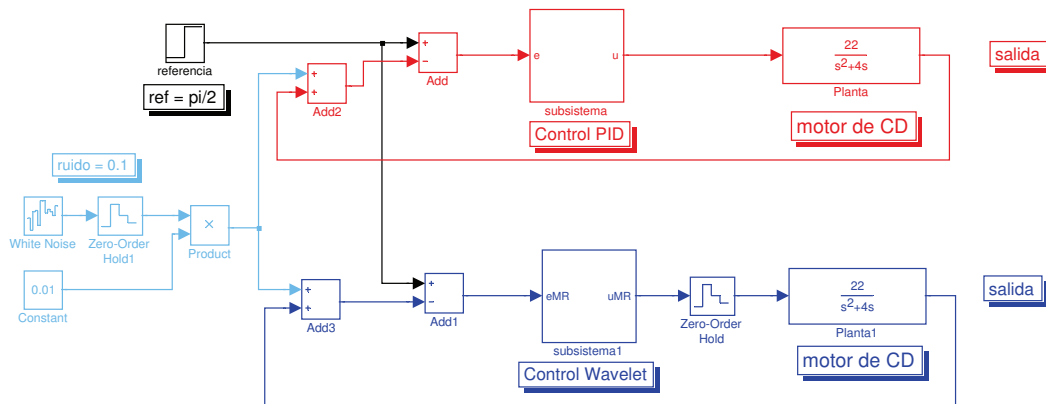


Figura F.5: Bloque para la simulación de los controles en un motor de CD con ruido blanco inyectado en la medición.

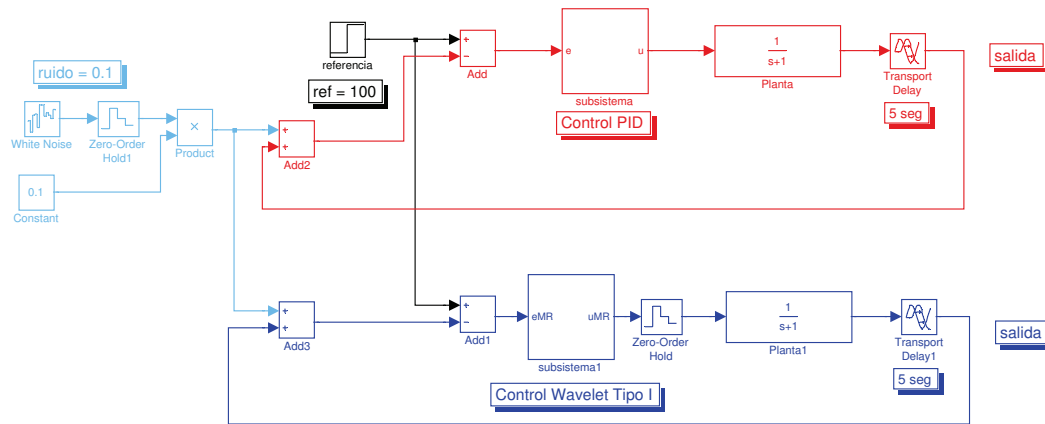


Figura F.6: Bloque para la simulación de los controles en un sistema con retardo y con ruido blanco inyectado en la medición.

MATLAB F.6: Comandos para la función “*fcn*” para el modelo cinemático inverso de posición

```
function [teta1, teta2] = fcn(L1, L2, x, y)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.

teta2 = acos(((x^2 + y^2) - (L1^2 + L2^2))/(2 * L1 * L2));
beta = atan2(y, x); alfa = asin((L2 * sin(teta2))/(sqrt(x^2 + y^2)));
if teta2 > 0
    teta1 = beta - alfa;
else
    teta1 = beta + alfa;
end
```

## F.2. Algoritmos del PID *wavelet* empleados para la implementación experimental en *simulink*

La tarjeta empleada para la implementación experimental y sus características son dadas en el Apéndice 5, y los bloques empleados son mostrados en las Figuras F.15, F.16, F.17, y F.18.

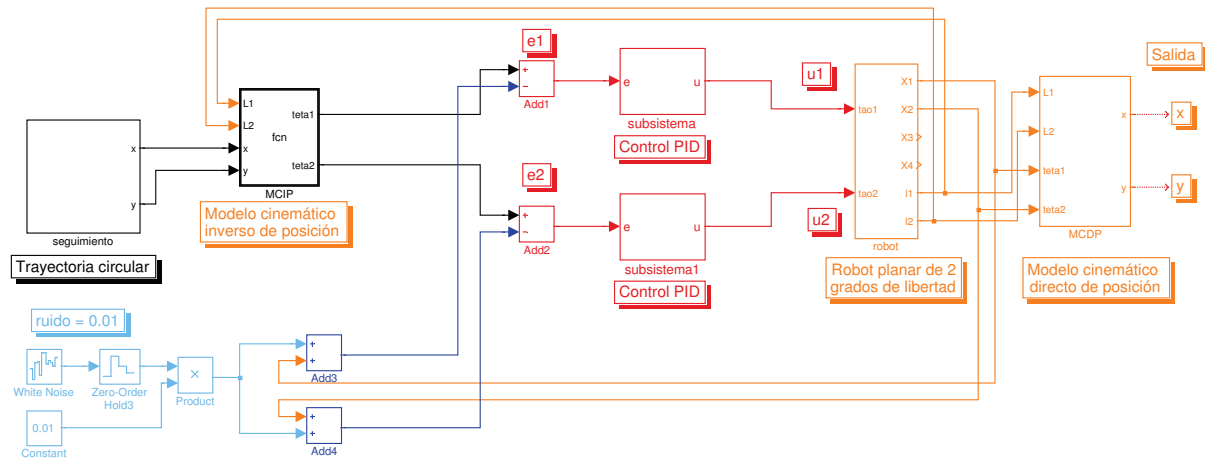


Figura F.7: Bloque para la simulación del control PID para el robot planar de 2 grados de libertad, con ruido blanco inyectado en la medición.

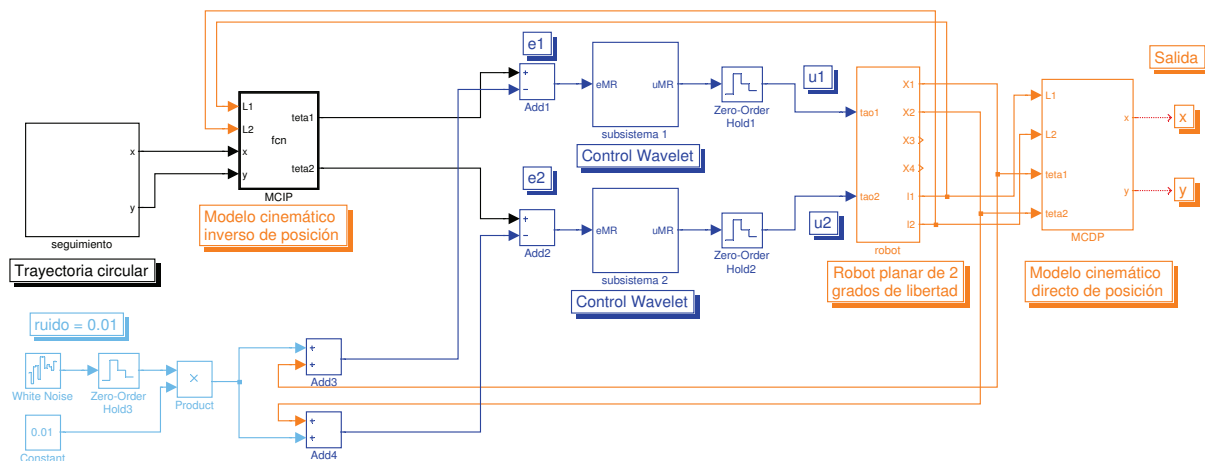


Figura F.8: Bloque para la simulación del control PID *wavelet* para el robot planar de 2 grados de libertad, con ruido blanco inyectado en la medición.



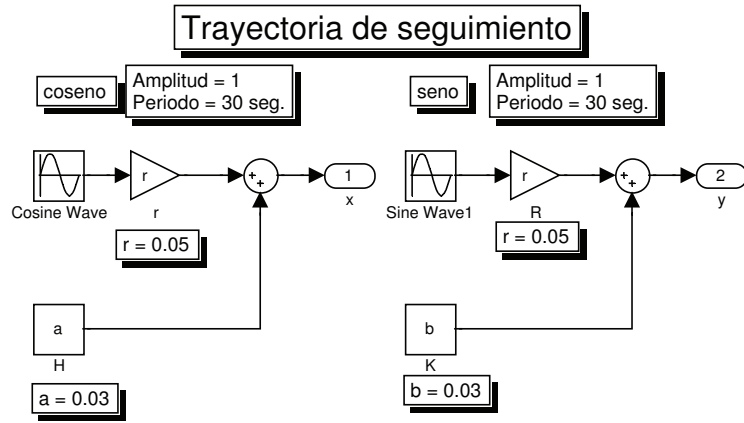


Figura F.9: Bloque para la simulación de la trayectoria para el robot planar de 2 grados de libertad.

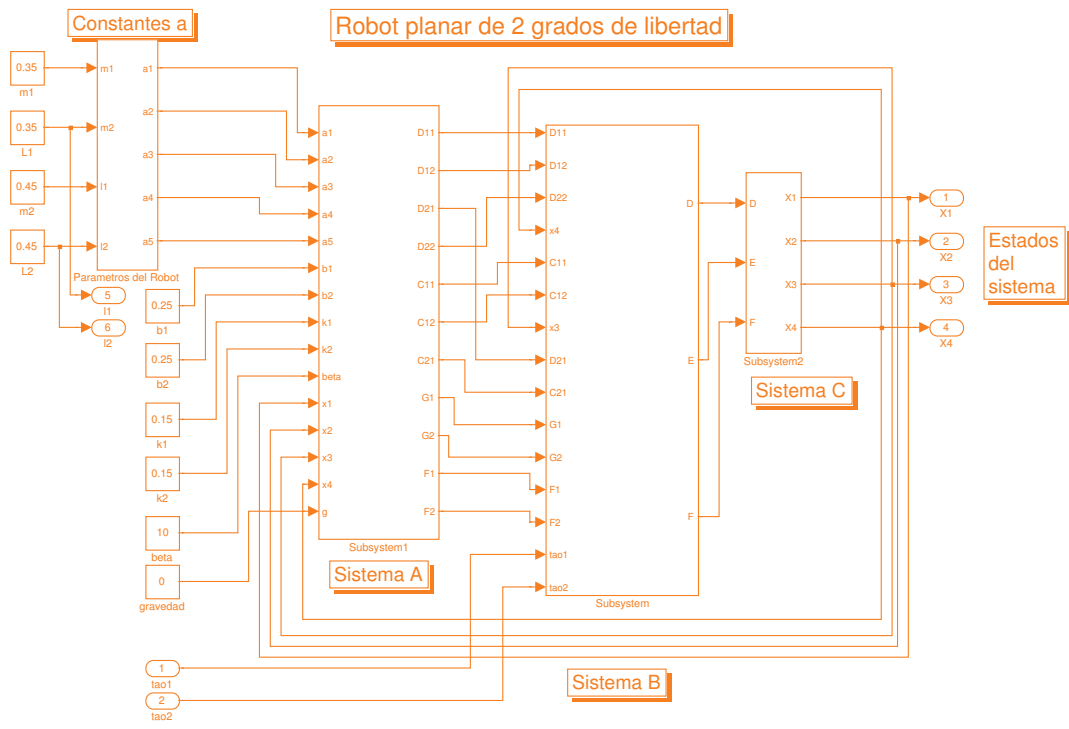


Figura F.10: Bloque para la simulación del robot planar de 2 grados de libertad.

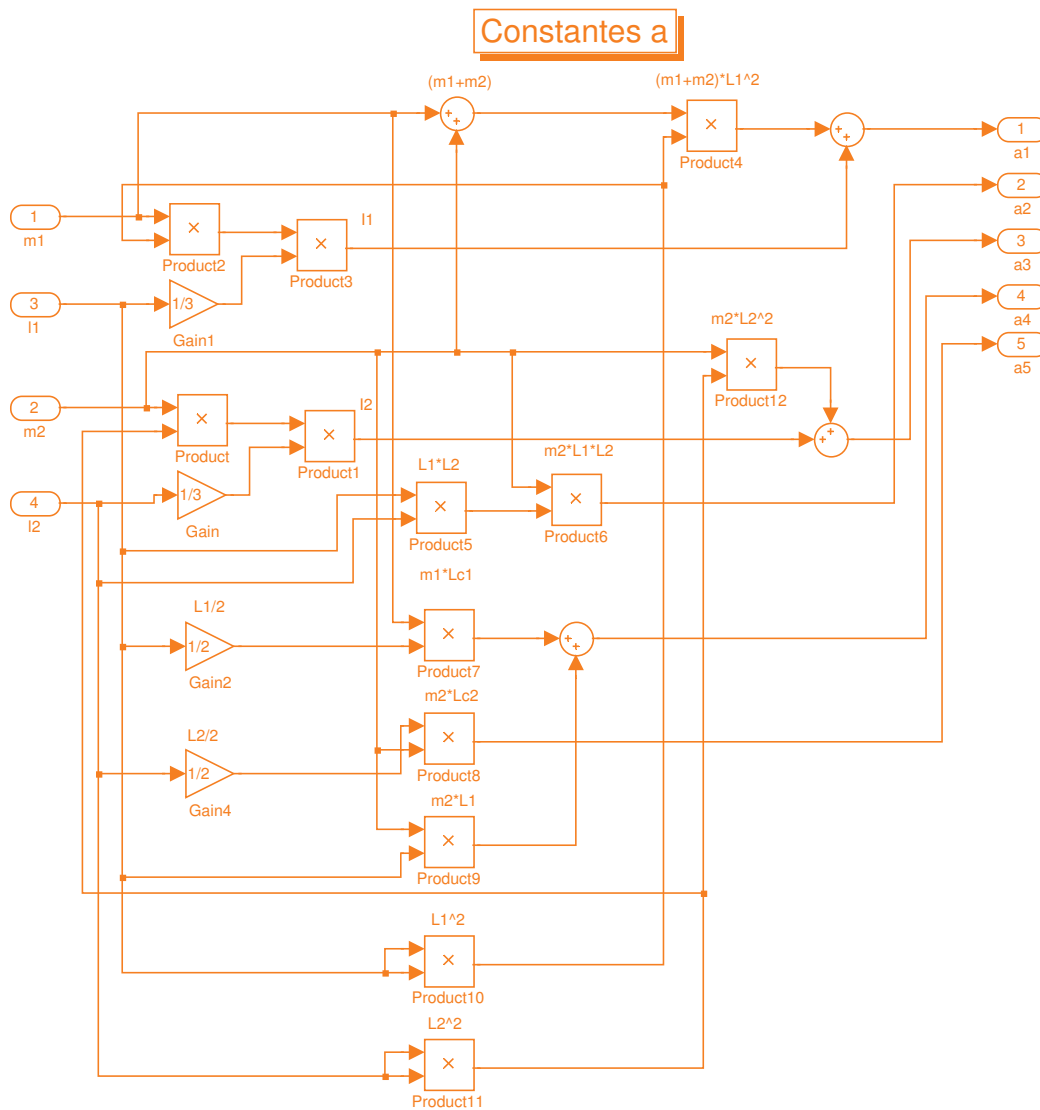


Figura F.11: Bloque de las constantes  $a_1$ ,  $a_2$  y  $a_3$  en el robot planar de 2 grados de libertad.

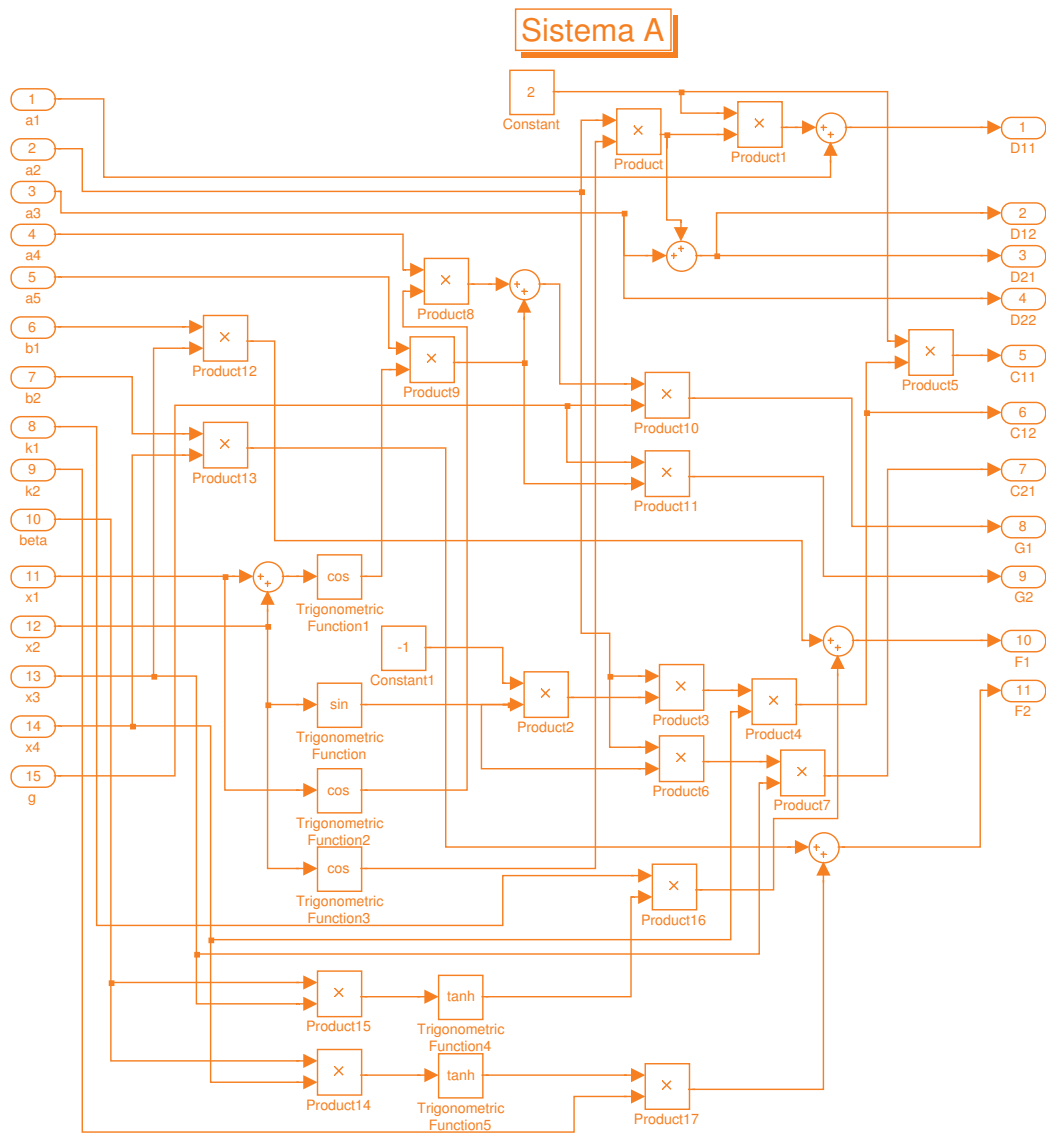


Figura F.12: Bloque del Sistema A en el robot planar de 2 grados de libertad.

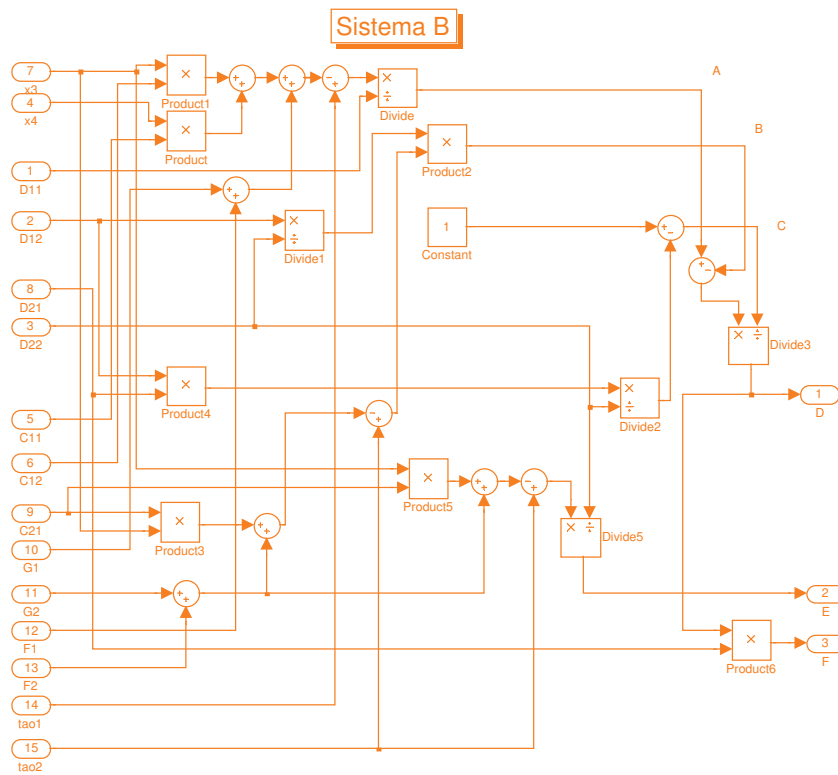


Figura F.13: Bloque del Sistema B en el robot planar de 2 grados de libertad.

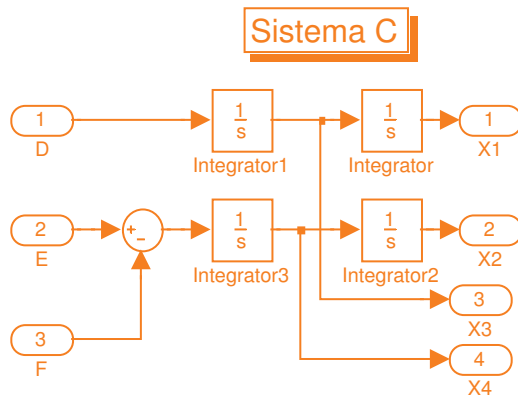


Figura F.14: Bloque del Sistema C en el robot planar de 2 grados de libertad.

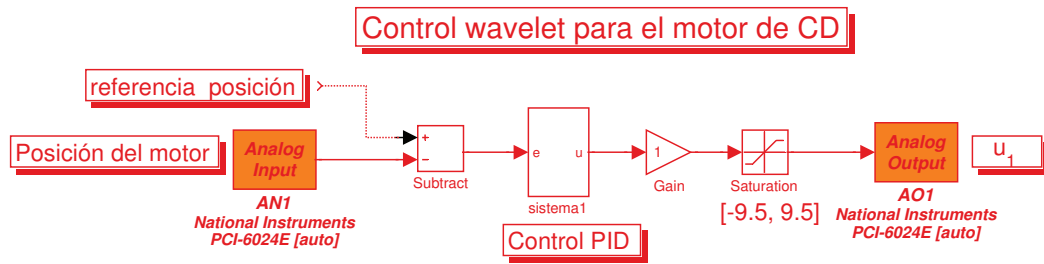


Figura F.15: Bloque del control PID para el motor de CD.

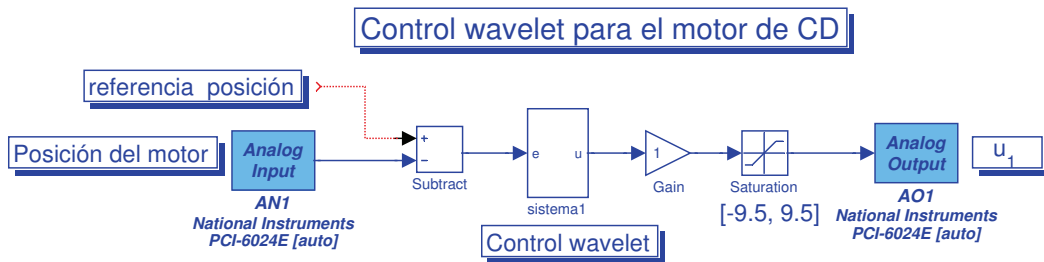


Figura F.16: Bloque del control PID *wavelet* para el motor de CD.

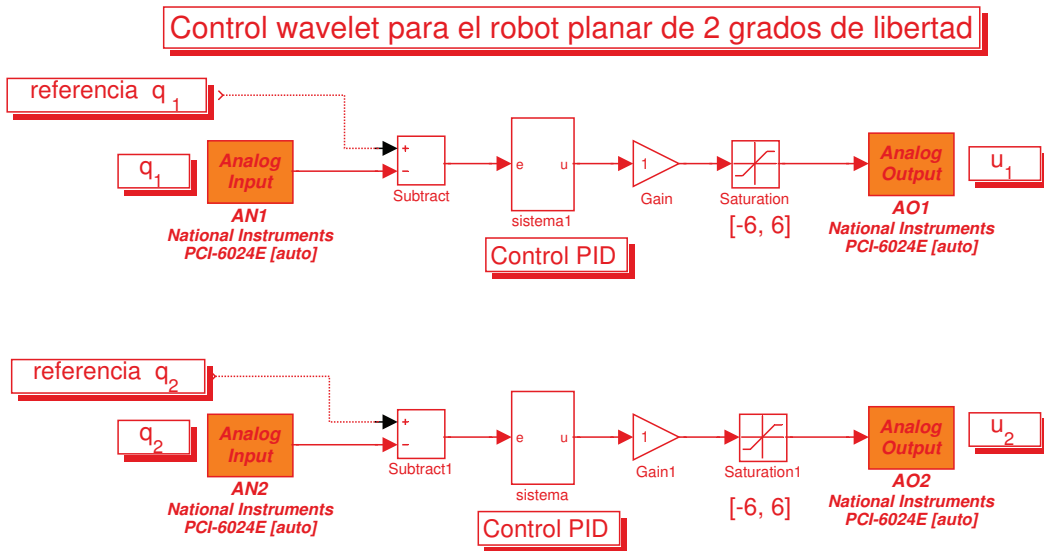


Figura F.17: Bloque del control PID para el robot planar de 2 grados de libertad.

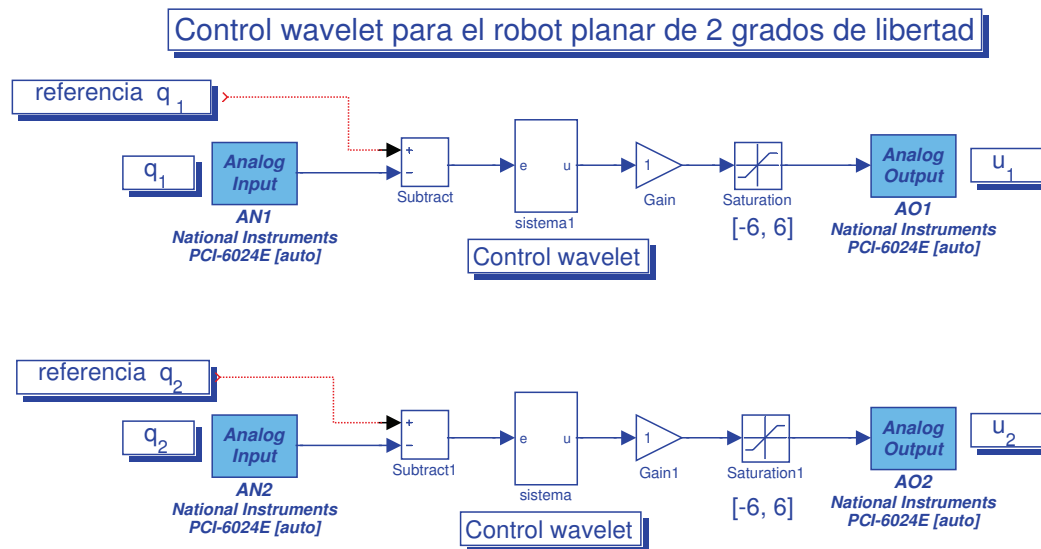


Figura F.18: Bloque del control PID *wavelet* para el robot planar de 2 grados de libertad.

### F.3. Algoritmo del filtro *wavelet*.

Los algoritmos fueron programados en un lenguaje de programación en  $C++$ , los cuales fueron incluidos dentro de una librería llamada “wavelet.h”, y la cual se muestra en Algoritmo F.7.

#### ALGORITMO F.7: wavelet.h

```
// wavelet.h
// Función para la el analisis multirresolución
#define Mbuff 64 // tamaño del buffer
#define Nfilt 14 // tamaño del filtro
/*
// Filtro con Dauechies (m=2)
float G0[] = { 0.6830127, 1.1830127, 0.31698729, -0.1830127 }; // h síntesis
float G1[] = { G0[3], -G0[2], G0[1], -G0[0] }; // g síntesis
float H0[] = { G0[3]/2, G0[2]/2, G0[1]/2, G0[0]/2 }; // h descomposición
float H1[] = { G1[3]/2, G1[2]/2, G1[1]/2, G1[0]/2 }; // g descomposición
```

```

*/
// Filtro con Dauechies (m=7)
float G0[ ] = {0.0778521, 0.396539, 0.729132, 0.469782, -0.143906, -0.224036,
0.0713092, 0.0806126, -0.0380299, -0.0165745, 0.012551, 0.00042957,
-0.0018016, 0.00035371}; // h sintesis
float G1[ ] = {G0[13], -G0[12], G0[11], -G0[10], G0[9], -G0[8],
G0[7], -G0[6], G0[5], -G0[4], G0[3], -G0[2],
G0[1], -G0[0]}; // g sintesis
float H0[ ] = {G0[13]/2, G0[12]/2, G0[11]/2, G0[10]/2, G0[9]/2, G0[8]/2,
G0[7]/2, G0[6]/2, G0[5]/2, G0[4]/2, G0[3]/2, G0[2]/2,
G0[1]/2, G0[0]/2}; // h descomposición
float H1[ ] = {G1[13]/2, G1[12]/2, G1[11]/2, G1[10]/2, G1[9]/2, G1[8]/2,
G1[7]/2, G1[6]/2, G1[5]/2, G1[4]/2, G1[3]/2, G1[2]/2,
G1[1]/2, G1[0]/2}; // g descomposición
//-----
float VectBuff[Mbuff];
float VectX[Mbuff];
float VectY[Mbuff]; float VectZ[Mbuff];
int Vectl[Mbuff]; float wave=0;
// almacenamiento de la señal de descomposición
float Matdesc[Mbuff][Mbuff];
float Matsint[Mbuff][Mbuff];
// arrays temporales
float temp1[2*Mbuff];
float tempX[2*Mbuff];
float temp3[2*Mbuff];
float temps[2*Mbuff];
//*****
// Inicializando Vectores
//*****
void temp1init( ) {
for(int w=0; w<2*Mbuff; w++)
temp1[w] = 0;}
void tempXinit( ) {
for(int w=0; w<2*Mbuff; w++)
tempX[w] = 0;}
void temp3init( ) {
for(int w=0; w<2*Mbuff; w++)
temp3[w] = 0;}
void tempsinit( ) {
for(int w=0; w<2*Mbuff; w++)
temps[w] = 0;}
void Vectlinit(int LV) {
Vectl[0] = Mbuff;

```

```

for (int w=1; w<LV+1; w++)
Vectl[w] = (Vectl[w-1]+Nfilt)/2;}
void VectBuffinit() {
for(int w=0; w<Mbuff; w++)
VectBuff[w] = 0;}
void VectXinit() {
for(int w=0; w<Mbuff; w++)
VectX[w] = 0;}
void VectYinit() {
for(int w=0; w<Mbuff; w++)
VectY[w] = 0;}
void VectZinit() {
for(int w=0; w<Mbuff; w++)
VectZ[w] = 0;}
//*****
// Funciones auxiliares
//*****
// Función para la convolución de dos vectores
void convxy(float *x, float *y, int lx, int ly) {
templinit();
tempxinit();
for (int wi=(ly-1); wi<(lx+ly-1); wi++){
temp1[wi] = x[(wi-ly+1)];}
float suma = 0;
for ( wi=0; wi<(lx+ly-1); wi++){
for(int wj=0; wj<ly; wj++){
suma += y[wj]*temp1[(ly-1+wi-wj)];}
tempx [wi] = suma;
suma = 0;} // Resultado almacenado en tempx[lx+ly-1]
}
// Función para la decimación dos abajo
void diez2down(float *x, int lx0) {
templinit();
for (int w=0; w<lx0; w++)
temp1[w] = x[2*w];
// Resultado almacenado en temp1[lx0]}
// Función para la interpolación a dos
void diez2up(float *x, int lx) {
templinit();
for (int w=0; w<lx; w++)
temp1[2*w] = x[w];
// Resultado almacenado en temp1[2*lx]
}
// Función para la descomposición de una señal X

```



```

void descomp(float *senX, float *filtroY, int lsen, int lfiltro){
convxy(senX,filtroY,lsen,lfiltro);
diez2down(tempx,(lsen+lfiltro)/2);
// Resultado almacenado en temp1[(lsen+lfiltro)/2]
}
// Función para la síntesis de una señal X
void síntesis(float *senX, float *filtroY, int lsen, int lfiltro, int l) {
diez2up(senX,lsen);
temp3init( );
for (int w=0; w<2*lsen; w++)
temp3[w] = temp1[w];
convxy(temp3,filtroY,2*lsen,lfiltro);
temp3init( );
for (w=0; w<l; w++)
temp3[w] = tempx[lfiltro-1+w];
// Resultado almacenado en temp1[lsen+lfiltro-1]
}
//*****
// Funciones para el filtrado de las señales
// Filtro wavelet para la variable articular de velocidad q1
//*****
float filtrowaveX(float WaveDato, int Nivel) {
int L0, L1,a1,b1,c1,d1,q;
wave = WaveDato;
for (a1=0; a1<Mbuff/2-1; a1++) {
VectX[a1] = VectX[a1+1];
VectX[Mbuff-1-a1] = VectX[a1+1]; }
VectX[Mbuff/2-1] = wave;
VectX[Mbuff/2] =wave;
Vectlinit(Nivel);
temp1init( );
tempxinit( );
temp3init( );
// Descomposición para "a"
// cargar datos
for(a1=0; a1<Mbuff; a1++){
Matdesc[0][a1] = VectX[a1]; }
for (a1=0; a1<Nivel; a1++){
L0 = Vectl[a1];
L1 = Vectl[a1+1];
for (b1=0; b1<L0; b1++){
temp3[b1] = Matdesc[a1][b1]; }
descomp(temp3,H0,L0,Nfilt);
for (b1=0; b1<L1; b1++){

```

```

Matdesc[a1+1][b1] = temp1[b1];} }
// Síntesis para "a"
temp1init( );
tempxinit( );
temp3init( );
tempsinit( ); // cargar datos
for(a1=0; a1<Vectl[Nivel]; a1++){
Matsint[Nivel][a1] = Matdesc[Nivel][a1];}
// Interacciones
for(d1=0; d1<Nivel; d1++){
L1 = Vectl[Nivel-d1];
L0 = Vectl[Nivel-1-d1];
for(b1=0; b1<L1; b1++){
temps[b1] = Matsint[Nivel-d1][b1];
síntesis(temps,G0,L1,Nfilt,L0);
for(c1=0; c1<L0; c1++){
Matsint[Nivel-1-d1][c1] = temp3[c1]; }
wave = (Matsint[0][Mbuff/2]+Matsint[0][Mbuff/2-1])/2;
return wave; }
// Filtro wavelet para la variable articular de velocidad q2
//*****
float filtrowaveY(float WaveDato, int Nivel) {
int L0, L1, a1,b1,c1,d1,q;
wave = WaveDato;
for (a1=0; a1<Mbuff/2-1; a1++){
VectY[a1] = VectY[a1+1];
VectY[Mbuff-1-a1] = VectY[a1+1];}
VectY[Mbuff/2-1] = wave;
VectY[Mbuff/2] =wave;
Vectlinit(Nivel);
temp1init( );
tempxinit( );
temp3init( );
// Descomposición para "a"
// cargar datos
for(a1=0; a1<Mbuff; a1++){
Matdesc[0][a1] = VectY[a1];}
for (a1=0; a1<Nivel; a1++){
L0 = Vectl[a1];
L1 = Vectl[a1+1];
for (b1=0; b1<L0; b1++){
temp3[b1] = Matdesc[a1][b1];}
descomp(temp3,H0,L0,Nfilt);
for (b1=0; b1<L1; b1++){

```

```

Matdesc[a1+1][b1] = temp1[b1];} }
// Síntesis para "a"
temp1init( );
tempxinit( );
temp3init( );
tempsinit( );
// cargar datos
for(a1=0; a1<Vectl[Nivel]; a1++){
Matsint[Nivel][a1] = Matdesc[Nivel][a1];}
// Interacciones
for(d1=0; d1<Nivel; d1++){
L1 = Vectl[Nivel-d1];
L0 = Vectl[Nivel-1-d1];
for(b1=0; b1<L1; b1++){
temps[b1] = Matsint[Nivel-d1][b1];
síntesis(temps,G0,L1,Nfilt,L0);
for(c1=0; c1<L0; c1++){
Matsint[Nivel-1-d1][c1] = temp3[c1];}
wave = (Matsint[0][Mbuff/2]+Matsint[0][Mbuff/2-1])/2;
return wave;}
// Filtro wavelet para la variable articular de velocidad q3
//*****
float filtrowaveZ(float WaveDato, int Nivel) {
int L0, L1,a1,b1,c1,d1,q;
wave = WaveDato;
for (a1=0; a1<Mbuff/2-1; a1++){
VectZ[a1] = VectZ[a1+1];
VectZ[Mbuff-1-a1] = VectZ[a1+1];}
VectZ[Mbuff/2-1] = wave;
VectZ[Mbuff/2] =wave;
Vectlinit(Nivel);
temp1init( );
tempxinit( );
temp3init( );
// Descomposición para "a"
// cargar datos
for(a1=0; a1<Mbuff; a1++){
Matdesc[0][a1] = VectZ[a1];}
for (a1=0; a1<Nivel; a1++){
L0 = Vectl[a1];
L1 = Vectl[a1+1];
for (b1=0; b1<L0; b1++){
temp3[b1] = Matdesc[a1][b1];}
descomp(temp3,H0,L0,Nfilt);

```

```

for (b1=0; b1<L1; b1++){
Matdesc[a1+1][b1] = temp1[b1];} }
// Sintesis para "a"
templinit( );
tempxinit( );
temp3init( );
tempsinit( );
// cargar datos
for(a1=0; a1<Vectl[Nivel]; a1++){
Matsint[Nivel][a1] = Matdesc[Nivel][a1];}
// Interacciones
for(d1=0; d1<Nivel; d1++){
L1 = Vectl[Nivel-d1];
L0 = Vectl[Nivel-1-d1];
for(b1=0; b1<L1; b1++){
temps[b1] = Matsint[Nivel-d1][b1];
sintesis(temps,G0,L1,Nfilt,L0);
for(c1=0; c1<L0; c1++){
Matsint[Nivel-1-d1][c1] = temp3[c1];}
wave = (Matsint[0][Mbuff/2]+Matsint[0][Mbuff/2-1])/2;
return wave;}
//*****

```

## F.4. Algoritmos del PID *wavenet* en Matlab

MATLAB F.8: Función para calcular las *wavelets* hijas a partir de una *wavelet* madre Morlet

```

% función para calcular las wavelet hijas de una wavelet madre morlet
function waveh = wavemorlet(t,a,b)
w0 = 1;
tao = (t-b)/a; % cambio de variable
%wavelet morlet
h = cos(w0*tao)*exp(-0.5*tao^2);
dh = (1/a)*(w0*sin(w0*tao)*exp(-0.5*tao ^2) + tao*h );
waveh = [h;dh];

```

MATLAB F.9: Algoritmo de para calcular la salida de la red neuronal *wavelet* en cada interacción  $k$

```

% Calculando h(tao) y dh(tao) en el instante k
for ii=1:K
hh = wavemorlet(k,ap(ii,k),bp(ii,k));
wavep(ii,k) = hh(1,1);
dwavep(ii,k) = hh(2,1);
end
% Calculando z(k) en el instante k
suma = 0;
for ii=1:K
suma = suma + wp(ii,k)*wavep(ii,k);
end
z(1,k) = suma;
% Calculando Fi(k)
suma = 0;
for ii=1:D
suma = suma + d(ii,k)*yest(1,k-ii);
end
Fi(1,k) = suma*v(1,k);
% Calculando Gamma(k)
suma = 0;
for ii=0:C
suma = suma + c(ii+1,k)*z(1,k-ii);
end
Gamma(1,k) = suma;
% Calculando yest(k) en el instante k
yest(1,k) = Gamma(1,k)*u(1,k) + Fi(1,k);

```

MATLAB F.10: Algoritmo de para calcular el error entre la salida deseada  $y$  y la salida de la red neuronal *wavelet yest* en el instante  $k$

```

% Calculando e(k) en el instante
en(1,k) = y(1,k) - yest(1,k);

```

MATLAB F.11: Algoritmo para el cálculo de los gradientes en el instante  $k$

```

% Calculando deltawp(k)
for jj=1:K
suma = 0;

```

```

for ii=0:C
suma = suma + c(ii+1,k)*wavep(jj,k-ii);
end
deltawp(jj,k) = en(1,k)*u(1,k)*suma;
end
% Calculando deltabp(k)
for jj=1:K
suma = 0;
for ii=0:C
suma = suma + c(ii+1,k)*dwavep(jj,k-ii);
end
deltabp(jj,k) = en(1,k)*u(1,k)*wp(jj,k)*suma;
end
% Calculando deltaap(k)
for jj=1:K
tao = (k-bp(jj,k))/ap(jj,k);
deltaap(jj,k) = tao*deltabp(jj,k);
end
% Calculando deltad(k)
for jj=1:D
deltad(jj,k) = en(1,k)*v(1,k)*yest(1,k-jj);
end
% Calculando deltac(k)
for jj=0:C
deltac(jj+1,k) = en(1,k)*u(1,k)*z(1,k-jj);
end

```

MATLAB F.12: Algoritmo para el cálculo del gradiente para las ganancias del control PID *wavelet* en el instante  $k$

```

% Calculando kpact(k)
for jj=1:N+1
deltakp(jj,k) = en(1,k)*Gamma(1,k)*em(jj,k);
end

```

MATLAB F.13: Algoritmo para el cálculo de las actualizaciones de los parámetros de la red neuronal *wavelet* en el instante  $k$

```
% Actualizando parámetros
% wp ap bp
for jj=1:K
wp(jj,k+1) = wp(jj,k) + muw*deltawp(jj,k);
ap(jj,k+1) = ap(jj,k) + mua*deltaap(jj,k);
bp(jj,k+1) = bp(jj,k) + mub*deltabp(jj,k);
end
% d
for jj=1:D
d(jj,k+1) = d(jj,k) + mud*deltad(jj,k);
end
% c
for jj=0:C
c(jj+1,k+1) = c(jj+1,k) + muc*deltac(jj+1,k);
end
```

MATLAB F.14: Algoritmo para el cálculo de las actualizaciones de las ganancias del control PID *wavelet* en el instante  $k$

```
% Actualizando parámetros
% ki
for jj=1:N+1
kp(jj,k+1) = kp(jj,k) + muk(jj,1)*deltakp(jj,k);
end
```