



**EDUCACIÓN**

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

# Tecnológico Nacional de México

Centro Nacional de Investigación  
y Desarrollo Tecnológico

## Tesis de Maestría

**Propuesta de una Neurona Artificial Dinámica para  
Propósitos de Modelado Entrada-Salida**

presentada por

**Ing. Henry Magdaleno Luna Pliego**

como requisito para la obtención del grado de  
**Maestro en Ciencias en Ingeniería  
Electrónica**

Director de tesis

**Dr. Juan Reyes Reyes**

Cuernavaca, Morelos, México. Noviembre de 2019.



"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor.,  
No. de Oficio:  
Asunto:

**09/octubre/2019**

**DIE/133/2019**

**Aceptación de documentos de tesis**

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Henry Magdaleno Luna Pliego**, con número de control **M17CE058** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Propuesta de una neurona artificial dinámica para propósitos de modelado entrada-salida"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS


CODIRECTOR DE TESIS

  
Dr. Juan Reyes Reyes  
Doctor en Ciencias en la Especialidad de  
Control Automático  
Cédula profesional 4214833

REVISOR 1

REVISOR 2

  
Dr. Carlos Daniel García Beltrán  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 8699605

  
Dra. Gloria Lilia Osorio Gordillo  
Doctora en Ciencias en Ingeniería Electrónica  
Cédula profesional 9968714

C.p. M.E. Guadalupe Osorio Gordillo, Centro Nacional de Investigación y Desarrollo Tecnológico de Servicios Escolares  
Estudiante  
Expediente  
MPS/Irr.

**SEP TECNM**  
**RECIBIDO**  
30 OCT 2019  
L. R. S. G. h.  
**SUBDIRECCIÓN ACADÉMICA**

**EDUCACIÓN** | **TECNOLÓGICO NACIONAL DE MÉXICO**  
30 OCT 2019  
**RECIBIDO**  
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO  
SERVICIOS ESCOLARES





"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor.,

**30/octubre/2019**

No. de Oficio:

**SAC/286/2019**

Asunto:

**Autorización de  
impresión de tesis**

**ING. HENRY MAGDALENO LUNA PLIEGO**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS**  
**EN INGENIERÍA ELECTRÓNICA**  
**PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "Propuesta de una neurona artificial dinámica para propósitos de modelo entrada-salida", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**ATENTAMENTE**

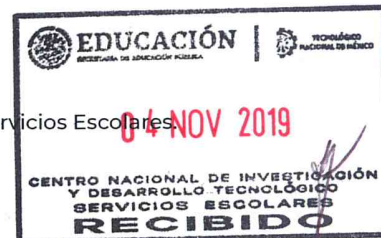
*Excelencia en Educación Tecnológica®*  
*"Conocimiento y tecnología al servicio de México"*

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**

ESTADOS UNIDOS MEXICANOS  
SEP TecNM  
CENTRO NACIONAL  
DE INVESTIGACIÓN  
Y DESARROLLO  
TECNOLÓGICO  
SUBDIRECCIÓN  
ACADÉMICA

C.p. Mtra. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares  
Expediente

GVGR/mcr



---

# Dedicatoria

---

*Este trabajo de investigación está dedicado enteramente y en su totalidad a mi señor padre  
Magdaleno Luna Sánchez, y mi señora madre Violeta Pliego Salazar. Dios los cuide y los  
proteja por siempre.*

Los científicos de hoy piensan en profundizar y no en esclarecer. Uno debe ser sensato para  
pensar con claridad, pero uno puede pensar con profundidad aun siendo un demente.  
Nikola Tesla



---

# Agradecimientos

---

En primer lugar doy gracias a Dios padre por prestarme vida y salud para culminar esta etapa de mi vida.

A mis padres por todo el apoyo económico, emocional y paternal durante todas las etapas de mi vida, que han culminado con este nuevo logro académico. Gracias apa' y gracias ama'.

A mi director de tesis el Dr. Juan Reyes Reyes por contar con su amistad, su asesoramiento y su paciencia a lo largo de este proyecto llamado maestría. Por todo, gracias Doc.

A mis profesores de Cenidet, los doctores: Carlos Daniel García Beltrán, Gloria Lilia Osorio Gordillo, José Francisco Gómez Aguilar, Luis Gerardo Vela Valdés, Gerardo Vicente Guerrero Ramírez, Carlos Manuel Astorga Zaragoza, Ricardo Fabricio Escobar Jiménez, Enrique Quintero-Mármol Márquez, Guadalupe Madrigal Espinoza, Manuel Adam Medina.

A la Dra. Citlaly Martínez García por todos sus sabios y acertados consejos en la escuela de la vida y sus valiosas contribuciones en mi formación profesional. Al Dr. Rodolfo Amalio Vargas Méndez por su amistad y su apoyo incondicional durante esta travesía.

A la Lic. Guadalupe Patricia Armas León por sus cursos de inglés que contribuyeron a mi formación técnica y científica. A la Lic. Lorena Ruiz Ramírez por su apoyo total y eficiente con todos los trámites y documentos necesarios durante mi estancia en Cenidet.

A todos mis amigos, personas importantes e inolvidables que han ocupado un fragmento en el espacio-tiempo de mi vida: Cynthia, Eusebio, Erick Michel, Jonathan, Christian, Isaac, José Juan, Diana Arleth, Elizabeth, Oyuki, Felipe, Luis Ignacio, Jesus Reyes, Miguel Francisco, Omar, Moises, Cesar, Ivonne, Alejandrina, Liz, Brian, Adrian, Daniel, Ramses, Polo, Manuel, Ángel, Gabriela, Nancy, Joaquín, José Campos, Luis Javier, Itzel, Yarely, Jaret, Alexia, Aneth, Roberto, Yenesey, Diana Tepozteco, Jesus Díaz, Gustavo Guerrero, Javier Sánchez, Julio, Juan Sánchez, Kelly, Julissa, Erick, Juliana, Aricel, Abigail, Maricruz, Dania, Janeth, Jessica, Deisy, Betzayra, Leobardo, Paula, Beatriz, Carlos, Citlaly Roldan, Eulalio, Javier Quintero, Valentin, Graciela, Deyaneira, Edith, Samantha, Mirna, Zinay, Arturo, y a todos los faltantes pero no menos importantes. Gracias por todo.

Al Tecnológico Nacional de México/Cenidet por recibirme en sus instalaciones y otorgarme las herramientas necesarias para mi formación académica y personal.

Al Consejo Nacional de Ciencia y Tecnología, por brindarme el sustento económico necesario para culminar en tiempo y forma mis estudios de posgrado.

---

# Resumen

---

Un modelo matemático es la aproximación de un sistema, y la precisión depende de distintos factores, como puede ser las suposiciones que se asumen en el proceso de modelado, la incertidumbre que se presente en el sistema, debidas a desgaste interno en los parámetros del sistema o dinámicas no modeladas. El modelo matemático de un sistema se puede obtener mediante algunas técnicas de modelado, como puede ser el que se basa en las leyes físicas que rigen al sistema y otra técnica, la cual se basa en el uso de datos de entrada y salida obtenidos directamente de un experimento, y recibe el nombre de identificación de sistemas.

Este trabajo de investigación propone una metodología alterna de modelado entrada-salida, la cual es formalizada por medio del planteamiento de un teorema, que manifiesta las condiciones necesarias para lograr la aproximación de sistemas estáticos y dinámicos utilizando una neurona artificial dinámica y una ley de aprendizaje.

Se verifica el buen desempeño de la neurona artificial dinámica y la metodología propuesta en esta tesis con la realización de distintos ejemplos de simulación, donde se obtiene como resultado el modelo matemático de sistemas con distinta naturaleza.

---

# Abstract

---

A mathematical model is the approximation of a system, and the precision depends on different factors like the assumptions that are assumed in the modeling process, the uncertainty presented in the system, due to internal wear in the system parameters or dynamics not modeled. The mathematical model of a system can be obtained with some modeling techniques like the based on the physical laws that govern the system and another technique based on the use of input and output data obtained directly from an experiment and it is the systems identification.

This research proposes an alternative methodology of input-output modeling, which is formalized through a theorem that manifests the necessary conditions to achieve the approximation of static and dynamic systems using a dynamic artificial neuron and a learning law.

The good performance of the dynamic artificial neuron and the methodology proposed in this thesis are verified with the realization of different simulation examples, where the mathematical model of systems with different nature is obtained.

---

# Índice general

---

Índice de tablas	VIII
Índice de figuras	IX
<b>1. Introducción</b>	<b>1</b>
1.1. Estado del arte. . . . .	1
1.1.1. Estabilidad estilo Lyapunov en Redes Neuronales Artificiales . . . . .	1
1.1.2. Aproximación de funciones con una neurona artificial . . . . .	2
1.1.3. Conclusión sobre el estado del arte . . . . .	3
1.2. Planteamiento del problema. . . . .	3
1.2.1. Identificación de sistemas . . . . .	4
1.2.2. Neuro-Identificación de sistemas . . . . .	5
1.3. Objetivo general. . . . .	5
1.3.1. Objetivos específicos. . . . .	6
1.4. Metas. . . . .	6
1.5. Justificación. . . . .	6
1.6. Alcances. . . . .	7
1.7. Hipótesis. . . . .	7
1.8. Estructura de la tesis. . . . .	7
<b>2. Marco Teórico</b>	<b>8</b>
2.1. Modelado de Neuronas Biológicas . . . . .	8
2.2. Modelado Estático de Neuronas Artificiales . . . . .	9
2.3. Modelado Dinámico de Neuronas Artificiales . . . . .	10
2.4. Puntos de Equilibrio . . . . .	11
2.4.1. Estabilidad de Lyapunov . . . . .	11
2.5. La Derivada Parcial . . . . .	12
2.6. Regla de la Cadena . . . . .	12
2.7. La Derivada Parcial de una Función Integral . . . . .	13
2.8. Derivación Matricial . . . . .	13
2.8.1. Definición 1. Derivada de un escalar con respecto a un vector . . . . .	13
2.8.2. Definición 2. Derivada de un vector con respecto a un escalar . . . . .	13
2.8.3. Definición 3. Derivada de un vector con respecto a un vector (Jacobiano) . . . . .	14
2.8.4. Definición 4. Derivada de un escalar con respecto a una matriz . . . . .	14
2.8.5. Definición 5. Derivada de una matriz con respecto a un escalar . . . . .	14
2.8.6. Definición 6. Derivada de un vector con respecto a una matriz . . . . .	14
2.8.7. Definición 7. Derivada de una matriz con respecto a un vector . . . . .	15
2.8.8. Definición 8. Derivada de una matriz con respecto a una matriz . . . . .	15
2.9. Propiedades de Derivación de Funciones Matriciales . . . . .	16
2.10. Método Directo de Lyapunov . . . . .	16

2.10.1. Principio del Teorema de Estabilidad de Lyapunov . . . . .	16
2.10.1.1. Función Candidata de Lyapunov . . . . .	17
2.10.1.2. Teorema de Estabilidad de Lyapunov . . . . .	17
2.10.2. El Algoritmo de Gradiente Descendente Desde la Perspectiva de Lyapunov	17
2.11. Aprendizaje en línea . . . . .	18
2.12. Aprendizaje fuera de línea . . . . .	18
<b>3. Modelado Entrada-Salida de Sistemas Estáticos y Dinámicos</b>	<b>20</b>
3.1. Estructura neuronal dinámica . . . . .	20
3.1.1. Puntos de equilibrio de la neurona artificial dinámica . . . . .	21
3.1.2. Constante de tiempo $\tau_p$ . . . . .	22
3.2. Teorema para aproximación de sistemas . . . . .	23
3.3. Demostración del teorema para aproximación de sistemas . . . . .	25
3.4. Observaciones del Teorema 3.1 . . . . .	28
3.5. Metodología propuesta para aproximación de sistemas . . . . .	29
3.5.1. Pasos para aproximación de sistemas . . . . .	29
<b>4. Simulaciones</b>	<b>31</b>
4.1. Simulación 1: Aproximación de un sistema estático lineal . . . . .	31
4.1.1. Objetivo de la simulación . . . . .	31
4.1.2. Configuración de la simulación . . . . .	31
4.1.3. Desarrollo de la simulación . . . . .	31
4.1.4. Conclusión de la simulación . . . . .	36
4.2. Simulación 2: Aproximación de un sistema estático no lineal . . . . .	37
4.2.1. Objetivo de la simulación . . . . .	37
4.2.2. Configuración de la simulación . . . . .	37
4.2.3. Desarrollo de la simulación . . . . .	37
4.2.4. Conclusión de la simulación . . . . .	40
4.3. Simulación 3: Aproximación de un sistema dinámico lineal . . . . .	40
4.3.1. Objetivo de la simulación . . . . .	40
4.3.2. Configuración de la simulación . . . . .	40
4.3.3. Desarrollo de la simulación . . . . .	41
4.3.4. Conclusión de la simulación . . . . .	46
4.4. Simulación 4: Aproximación de un sistema dinámico no lineal . . . . .	46
4.4.1. Objetivo de la simulación . . . . .	46
4.4.2. Configuración de la simulación . . . . .	47
4.4.3. Desarrollo de la simulación . . . . .	47
4.4.4. Conclusión de la simulación . . . . .	49
4.5. Simulación 5: Aproximación de un sistema estático no lineal con múltiples entradas	50
4.5.1. Objetivo de la simulación . . . . .	50
4.5.2. Configuración de la simulación . . . . .	50
4.5.3. Desarrollo de la simulación . . . . .	51
4.5.4. Conclusión de la simulación . . . . .	56
4.6. Simulación 6: Aproximación de la dinámica de un biorreactor híbrido lecho fijo- fluidizante inverso con datos experimentales . . . . .	56
4.6.1. Objetivo de la simulación . . . . .	56
4.6.2. Configuración de la simulación . . . . .	56
4.6.3. Desarrollo de la simulación . . . . .	57
4.6.4. Conclusión de la simulación . . . . .	60

<b>5. Neurona Artificial Dinámica vs Red Neuronal Artificial</b>	<b>61</b>
5.1. Simulación 7. RNA y NAD aproximando una función estática lineal . . . . .	62
5.1.1. Objetivo de la simulación . . . . .	62
5.1.2. Configuración de la simulación . . . . .	62
5.1.3. Desarrollo de la simulación . . . . .	63
5.1.4. Conclusión de la simulación . . . . .	64
5.2. Simulación 8. RNA y NAD aproximando una función dinámica (Circuito RC) . .	65
5.2.1. Objetivo de la simulación . . . . .	65
5.2.2. Configuración de la simulación . . . . .	65
5.2.3. Desarrollo de la simulación . . . . .	65
5.2.4. Conclusión de la simulación . . . . .	67
<b>6. Conclusiones</b>	<b>68</b>
6.1. Trabajos Futuros . . . . .	69
<b>Bibliografía</b>	<b>70</b>
<b>A. Funciones de activación</b>	<b>73</b>
<b>B. Manipulación algebraica para acotar el error de aproximación</b>	<b>74</b>
<b>C. Jornada de Ciencia y Tecnología Aplicada</b>	<b>76</b>



---

# Índice de tablas

---

4.5.1.Compuerta AND . . . . .	50
-------------------------------	----

---

# Índice de figuras

---

2.1.1.Neurona biológica [1]. . . . .	8
2.2.1.Neurona Artificial Estática. . . . .	10
2.3.1.Neurona Artificial Dinámica (Hopfield) [1]. . . . .	11
2.11.1.Error de aproximación 1 . . . . .	18
2.12.1.Error de aproximación 2 . . . . .	19
3.2.1.Eschema General . . . . .	24
4.1.1.Señal de entrenamiento . . . . .	32
4.1.2.Aprendizaje de la neurona dinámica . . . . .	32
4.1.3.Convergencia de los pesos sinápticos . . . . .	33
4.1.4.Error de entrenamiento . . . . .	33
4.1.5.Señal de validación 1 . . . . .	34
4.1.6.Señal de validación 2 . . . . .	34
4.1.7.Respuesta a la entrada de entrenamiento . . . . .	35
4.1.8.Respuesta a la entrada 2 . . . . .	35
4.1.9.Respuesta a la entrada 3 . . . . .	36
4.1.10.Función estática lineal. . . . .	36
4.2.1.Señal de entrada . . . . .	37
4.2.2.Aprendizaje de la neurona dinámica 2 . . . . .	38
4.2.3.Convergencia de los pesos sinápticos 2 . . . . .	38
4.2.4.Error de entrenamiento 2 . . . . .	39
4.2.5.NAD vs Función Estática No Lineal . . . . .	39
4.2.6.Función estática no lineal. . . . .	40
4.3.1.Señal senoidal de entrenamiento . . . . .	41
4.3.2.Aprendizaje de la neurona dinámica 3 . . . . .	42
4.3.3.Convergencia de los pesos sinápticos 3 . . . . .	42
4.3.4.Error de entrenamiento 3 . . . . .	43
4.3.5.Tren de pulsos cuadrado para validar . . . . .	44
4.3.6.Cresta de sierra para validar . . . . .	44
4.3.7.Respuesta a la señal senoidal . . . . .	45
4.3.8.Respuesta al tren de pulsos cuadrado . . . . .	45
4.3.9.Respuesta a la cresta de sierra . . . . .	46
4.3.10.Circuito RC . . . . .	46
4.4.1.Tren de pulsos cuadrado . . . . .	47
4.4.2.Aprendizaje de la neurona dinámica 4 . . . . .	48
4.4.3.Convergencia de los pesos sinápticos 4 . . . . .	48
4.4.4.Error de entrenamiento 4 . . . . .	48
4.4.5.Neurona entrenada . . . . .	49
4.4.6.Sistema no lineal . . . . .	50

4.5.1. Señal de entrada 1 . . . . .	51
4.5.2. Señal de entrada 2 . . . . .	51
4.5.3. Aprendizaje de la neurona dinámica 5 . . . . .	52
4.5.4. Error de entrenamiento 5 . . . . .	52
4.5.5. Entrenamiento de $w_a$ . . . . .	53
4.5.6. Entrenamiento de $w_b$ . . . . .	53
4.5.7. Entrenamiento de $\theta$ . . . . .	54
4.5.8. ND vs Compuerta AND . . . . .	55
4.5.9. Señal de entrada para validar 1 . . . . .	55
4.5.10. Señal de entrada para validar 2 . . . . .	55
4.5.11. ND vs Compuerta AND con señales de validación . . . . .	56
4.5.12. Compuerta AND . . . . .	56
4.6.1. Entrada del sistema . . . . .	57
4.6.2. Aprendizaje de la neurona dinámica 6 . . . . .	58
4.6.3. Convergencia de los pesos sinápticos 5 . . . . .	58
4.6.4. Error de entrenamiento 6 . . . . .	59
4.6.5. Neurona dinámica entrenada . . . . .	60
5.0.1. Neurona artificial dinámica . . . . .	61
5.0.2. RNA 2-1 de una entrada . . . . .	62
5.1.1. Señal de excitación . . . . .	63
5.1.2. comparación . . . . .	64
5.1.3. Error de entrenamiento 7 . . . . .	64
5.2.1. Señal de excitación 2 . . . . .	66
5.2.2. comparación 2 . . . . .	66
5.2.3. Error de entrenamiento 8 . . . . .	67

---

# Nomenclatura

---

## Letras mayúsculas

$\mathfrak{R}$	Conjunto de los números reales
$\Omega$	Subconjunto de la entrada
$\mathcal{C}$	Subconjunto de la salida
$W$	Matriz de pesos sinápticos
$R_i$	i-ésimo resistor
$C$	Capacitor
$D$	Conjunto D
$A$	Matriz de retroalimentación
$V$	Función de energía
$J$	Función costo
$T_e$	Paso de integración

## Letras minúsculas

$y_d$	Salida deseada del sistema físico
$y$	Salida de la neurona artificial dinámica
$u$	Señal de entrada
$x$	Estado de la neurona artificial dinámica
$\dot{x} = \frac{dx}{dt}$	Derivada del estado con respecto al tiempo
$f(\cdot)$	Función no lineal de retroalimentación
$g(\cdot)$	Función no lineal de salida
$\varphi(\cdot)$	Función de activación
$a$	Coefficiente de retroalimentación
$b$	Coefficiente de la entrada
$c$	Coefficiente de salida
$w_i$	i-ésimo peso sináptico
$w_j$	j-ésimo peso sináptico
$x_i$	i-ésimo estado
$\theta$	Umbral neuronal
$w_a$	Peso sináptico (combinación lineal con el estado)
$w_b$	Peso sináptico (combinación lineal con la entrada)
$w_a^*$	Peso sináptico entrenado (combinación lineal con el estado)
$w_b^*$	Peso sináptico entrenado (combinación lineal con la entrada)
$\theta^*$	Umbral neuronal entrenado
$k_0$	Elementos del conjunto omega
$\alpha$	Amplitud de la tangente hiperbólica
$\tau_p$	Constante de tiempo
$\tau_d$	Constante de tiempo deseada
$v$	Argumento de la función de activación de la neurona artificial dinámica
$x_0$	Punto de equilibrio
$\mu$	Matriz de coeficientes de convergencia
$z$	Integrador de la derivada de la función de activación de la neurona artificial dinámica
$e_y$	Error de aproximación
$t$	tiempo
$t_f$	Tiempo final
$v_c$	Voltaje del capacitor
$v_{in}$	Voltaje de entrada
$w$	Estado del sistema no lineal
$diag_n$	matriz diagonal de $n$ dimensiones
$vec$	vectorización
${}^N w_{ij}(t)$	Es el i-ésimo peso sináptico de la j-ésima neurona, de la N-ésima capa.

## Abreviaturas

NAD	Neurona Artificial Dinámica
SL	Sistema lineal
SNL	Sistema no lineal
MLP	Perceptrón multicapa
RBF	Función de base radial
RL	Aprendizaje por refuerzo
MDP	Procesos de decisión de Markov
FC	Célula de disparo
LTP	Potenciación sináptica de larga duración
LTI	Sistema lineal invariante en el tiempo
CVA	Carga volumétrica aplicada
DQO	Demanda química de oxígeno soluble
DQOs	Demanda química de oxígeno soluble a la salida
MSE	Error medio cuadrático
RNA	Red neuronal artificial



# Capítulo 1

---

## Introducción

---

Hoy en día las industrias implementan distintas técnicas de control para elevar la eficiencia, productividad y seguridad de sus diferentes procesos. Para los métodos de control basados en modelo se requiere una representación matemática del proceso o sistema que se busca controlar.

El modelo matemático de un sistema o proceso se puede obtener mediante algunas técnicas de modelado, como puede ser basándose en las leyes físicas que gobiernan al sistema. Otra metodología es la identificación de sistemas, el cual se basa en la obtención del modelo matemático por medio de los datos de entrada y salida de la planta, los cuales se obtienen mediante un experimento directo con el proceso real.

Como herramienta alterna a la metodología de identificación de sistemas se ha empleado el uso de redes neuronales artificiales, ya que su principio de funcionamiento se basa en el procesamiento de información del cerebro humano, las cuales son capaces de procesar información, aprender y adaptarse a distintas tareas aplicables a la teoría de identificación.

En este trabajo de tesis se propone una neurona artificial dinámica y una ley de aprendizaje para la aproximación de sistemas estáticos y dinámicos.

### 1.1. Estado del arte.

La obtención de modelos que describan la dinámica de distintos tipos de sistemas, mediante el empleo de neuronas artificiales y redes neuronales artificiales, es un tema que ha sido ampliamente abordado por la comunidad de control automático.

#### 1.1.1. Estabilidad estilo Lyapunov en Redes Neuronales Artificiales

En la referencia [2] la red neuronal que se utiliza para la identificación posee una modificación robusta con el objetivo de garantizar la estabilidad tipo Lyapunov. Se utiliza la técnica de estabilidad entrada a estado, esto para dar acceso a los algoritmos de entrenamiento robustos en dominio de tiempo discreto de las redes neuronales recurrentes.

En la referencia [3] se comparan diferentes tipos de algoritmos de aprendizaje, en la cual se concluye que el algoritmo de aprendizaje basado en un filtro de Kalman posee mejores características que otros algoritmos de aprendizaje, pero se debe recalcar la complejidad de este algoritmo y su sensibilidad a ruidos. Algunas de sus principales características es su rápida convergencia. Se emplea el método estilo Lyapunov para verificar y demostrar la estabilidad de este algoritmo

de aprendizaje.

En [4] se propone una nueva estructura de red neuronal recurrente con retardo (TDRNN). Algunas características de esta estructura es un mayor rango para la resolución de su memoria, es más simple de manejar su estructura y utilizan el algoritmo de retropropagación. Con el objetivo de obtener una convergencia rápida, las tasas óptimas de aprendizaje adaptable se obtienen en tiempo discreto, y se utiliza el método de Lyapunov para garantizar estabilidad.

En [5] proponen una diferencia en las escalas de tiempo para utilizar dos redes neuronales dinámicas en la identificación de sistemas no lineales. Se utiliza una función de Lyapunov en la primera red neuronal para calcular su ley de aprendizaje en línea. En la segunda red neuronal se emplea una función de zona muerta para el cálculo de su algoritmo de aprendizaje en línea, esto con el fin de mejorar la identificación del sistema no lineal.

En la referencia [6] se diseña un esquema de identificación en línea para disminuir el error del estado frente a distintas perturbaciones. La ley de aprendizaje modifica los pesos sinápticos con el fin de aproximar las no linealidades desconocidas con error acotado. Se anexa un modelo de identificación con retroalimentación para disminuir el error de estado. Dicho modelo basa su estructura en una función de delimitación para estimar un límite superior de los disturbios. Se utiliza una técnica de delimitación y el método de Lyapunov para mostrar las características del error.

En los trabajos expuestos se resalta la importancia de asegurar la estabilidad en redes neuronales dinámicas, para las cuales se emplea un análisis estilo Lyapunov. Se utilizan distintas arquitecturas de redes neuronales y algoritmos de aprendizaje para los que se exponen sus principales características en la identificación de sistemas.

### 1.1.2. Aproximación de funciones con una neurona artificial

En la referencia [7] se realizó la estimación de estados no lineales y algunos temas relacionados, como la estimación paramétrica, el diagnóstico de fallas y la atenuación de perturbaciones, se abordan a través de una nueva metodología de diferenciación numérica. Las correspondientes definiciones y propiedades teóricas del sistema básico se presentan dentro del marco del álgebra diferencial, que permite manejar las variables del sistema y sus derivadas de cualquier orden.

En [8] se centran en el aprendizaje por refuerzo (RL), el cual se ha centrado en la aproximación de la función en el aprendizaje de la predicción y el control de los procesos de decisión de Markov (MDP). El uso de técnicas de aproximación de funciones en RL será esencial para tratar con los MDP con espacios de acción y estados grandes o continuos. Se analiza la convergencia y la representación de características de los algoritmos RL. Desde un aspecto empírico, el rendimiento de diferentes algoritmos de RL se evaluó y comparó en varias tareas de predicción de aprendizaje de referencia y control de aprendizaje.

En [9] se propone un modelo computacional de neurona, llamado célula de disparo (FC), cuyas propiedades cubren fenómenos como la atenuación de los receptores para estímulos externos, el retraso y la descomposición de los potenciales postsinápticos, la modificación de los pesos internos debido a la propagación de los potenciales postsinápticos a través de la dendrita, modificación de las propiedades de la memoria analógica para cada entrada debido a un patrón de potenciación sináptica de corta duración o potenciación sináptica de larga duración (LTP),

generación de picos de salida cuando la suma de todas las entradas excede un umbral y refracción. La célula puede tomar una de las tres formas: excitatoria, inhibitoria y receptora. Las simulaciones por computadora mostraron que, dependiendo de la fase de las señales de entrada, la frecuencia de salida de la neurona artificial puede demostrar diversos comportamientos caóticos.

En [10] se muestra una neurona artificial de integración y disparo basada en un memristor de conmutación de umbral. Esta neurona muestra cuatro características críticas para la computación basada en el potencial de acción: el pico de todo o nada de un potencial de acción, el pico de umbral, un período refractario y una respuesta de frecuencia modulada por la fuerza. Como neurona post-sináptica, se demostró que la neurona diseñada era aplicable al reconocimiento de dígitos. Estos resultados demuestran que la neurona artificial desarrollada puede realizar las funciones básicas de las neuronas de punta y tiene un gran potencial para la computación neuromórfica.

En la referencia [11] se comparan los diferentes lenguajes de programación, como Java, C y C ++, que en este momento son los mejores en el ranking internacional para el desarrollo de programas, para observar cuál de ellos minimiza el tiempo de respuesta (variable de salida) de un algoritmo de neurona artificial (variable de entrada). Los resultados muestran cómo cada uno de los lenguajes de programación da un tiempo de respuesta al algoritmo propuesto. Se discuten los resultados en el marco de la controversia de la neurona artificial frente a los lenguajes de programación.

### 1.1.3. Conclusión sobre el estado del arte

Para la aproximación de sistemas utilizando redes neuronales dinámicas es de vital importancia asegurar la estabilidad de la estructura neuronal, para lo que se utiliza el método de estabilidad estilo Lyapunov.

A pesar de la antigüedad del campo, actualmente se tienen pocos trabajos sobre una neurona artificial dinámica, ya que la investigación se volcó a estructuras más complejas como Redes Neuronales Artificiales. Lo novedoso de este trabajo de tesis es la propuesta de una misma estructura neuronal, con una misma ley de aprendizaje para la aproximación de sistemas con ciertas características, las cuales son que sean estáticos, dinámicos, múltiples entradas, una salida, de primer orden, esto quiere decir, estado escalar.

## 1.2. Planteamiento del problema.

Un sistema es un grupo de elementos que interactúan entre sí con el objetivo de realizar una tarea específica. Para los cuales existen dos tipos, sistemas dinámicos y sistemas estáticos. Se entiende por sistema estático aquellos que no varían en el tiempo, dicho de otra forma, que la salida del sistema únicamente depende de las entradas actuales, significa que no tienen memoria y no almacenan energía. Los cuales pueden representar comportamientos lineales (1.1) y no lineales (1.2).

$$y_d = au \tag{1.1}$$

$$y_d = f(u) \tag{1.2}$$

Un sistema dinámico es aquel que evoluciona a través del tiempo, esto es, que sus salidas dependen de las entradas y salidas pasadas, y no solo de las entradas actuales. Para los que se conoce su representación lineal (1.3) y no lineal (1.4)

$$\Sigma_{SL} : \begin{cases} \dot{x}(t) = ax(t) + bu(t) \\ y_d(t) = cx(t) \end{cases} \quad (1.3)$$

$$\Sigma_{SNL} : \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y_d(t) = g(x(t)) \end{cases} \quad (1.4)$$

Un modelo matemático es una aproximación de un sistema, ya sea estático o que evolucione en el tiempo (dinámico), una representación exacta es imposible o difícil de obtener, los motivos son imprecisiones de cálculo en la determinación de los parámetros, cambio en las constantes de tiempo, desgaste interno, y dinámicas no modeladas.

Dicho modelo matemático se puede obtener mediante las leyes físicas que gobiernan al sistema, este método requiere de un conocimiento profundo y claro del sistema real, además dicho proceso de modelado es relativamente extenso y algunas veces imposible. El método de identificación de sistemas sólo requiere datos de entrada y salida del sistema real obtenidos directamente de un experimento, no es necesario un conocimiento tan profundo del proceso a diferencia del método que se basa en las leyes físicas del sistema. Otra ventaja es que el tiempo de modelado es relativamente más corto y el modelo obtenido es una aproximación adecuada del sistema real.

Otra opción como herramienta alterna en el método de identificación es el uso de redes neuronales artificiales, para las cuales existen diferentes esquemas y metodologías reportadas en la literatura. Por ejemplo las redes neuronales artificiales recurrentes para la identificación de sistemas dinámicos y las redes neuronales artificiales estáticas para la identificación de sistemas estáticos.

### 1.2.1. Identificación de sistemas

Los primeros trabajos de identificación surgieron en la época de los 60s, gracias a los artículos publicados por Ho y Kalman [12] y Åström y Bohlin [13], en cuyos artículos crean las bases para el desarrollo de dos técnicas de identificación: Identificación en sub-espacios (técnicas de proyección en el espacio euclidiano) e identificación mediante la predicción del error (criterio de minimización de dependencia de parámetros), siendo este último el método que predominaba a mediados de los 60s.

En 1970 Box y Jenkins publican un libro “Time-Series Analysis, Forecasting and Control”, en el cual proporcionan una metodología de identificación más completa, abarcando desde el análisis de datos hasta la estimación del modelo.

Durante esa época los investigadores se dedicaban a buscar el método para obtener el modelo que representara de manera exacta al sistema real, (método que aún no existe en la actualidad) fue a finales de los 70s [14, 15], cuando se consideró a la identificación una teoría de aproximación al sistema real.

En los 80s la identificación se convirtió en un problema de diseño, es decir se comenzó a tomar en cuenta el diseño de experimentos, la elección de la estructura y la determinación del

criterio de elección como parte del método de identificación.

Ljung en su libro “System Identification Theory For The User” [16], plantea una metodología de identificación en la cual desde el diseño de experimentos hasta la generación del modelo toma en cuenta el uso final del modelo.

### 1.2.2. Neuro-Identificación de sistemas

Existen trabajos sobre identificación utilizando redes neuronales, uno de ellos realizado por Faisal, quien realiza la identificación de un motor de inducción colocando en paralelo al motor y la red neuronal obteniendo y utilizando la diferencia (error) entre la salida del motor y la salida estimada por la red neuronal, en este trabajo se menciona que es posible identificar una amplia clase de funciones con redes neuronales de una sola capa oculta [17].

Otro trabajo es el realizado por David C. Hyland quien realiza de identificación de sistemas lineales, con las señales de salida afectadas por ruido de un sensor y desarrolla algoritmos que son capaces de minimizar el efecto del ruido de ese sensor. Él utiliza una estructura denominada ganglio, esta estructura permite el rechazo a ruido al forzar a la matriz de pesos sinápticos a tener una estructura Toeplitz [18].

Olivera Jovanović hace un estudio sobre la identificación de sistemas dinámicos aprovechando algunas de las características de las redes neuronales como su capacidad de auto aprendizaje, y su capacidad de manipular sistemas no lineales, él menciona que al hacer identificación en línea el algoritmo de retropropagación no tiene un desempeño adecuado ya que la convergencia de la red neuronal es muy lenta [19].

E. I. Gaura, realizó la identificación de un acelerómetro micro-mecanizado, basándose en la propiedad de las redes neuronales de aproximar modelos no lineales, el utilizó las estructuras MLP (Multi Layer Perceptron) y RBF (Radial Basis Function) [20], mencionando que obtuvo mejor resultado al utilizar la estructura MLP.

Yong-Geun Lee realizó la identificación de un rectificador de tres fases y menciona que en un momento de la identificación requiere un algoritmo de autoaprendizaje, por lo tanto decide utilizar redes neuronales [21].

Wen Yu y América Morales, realizan la identificación de un mezclador de gasolina utilizando una red neuronal dinámica para identificar la parte dinámica del sistema y una red neuronal estática para aproximar las partes estáticas del sistema, ellos mencionan en su trabajo que las redes neuronales son capaces de modelar una gran variedad de sistemas dinámicos complejos [22].

Wen Yu, Alexander S. Poznyak and Edgar N. Sánchez, utilizaron una red neuronal dinámica para realizar identificación en línea, y se basan en un análisis estilo Lyapunov para que el algoritmo de aprendizaje sea estable, y una ecuación algebraica de Riccati para construir el error de identificación [23].

## 1.3. Objetivo general.

Desarrollar una metodología que utilice una arquitectura neuronal para el modelado entrada-salida de sistemas dinámicos y estáticos, dentro de un subconjunto real, con una función de

activación dinámica y acotada, y los pesos sinápticos realizando una retroalimentación de la salida y una prealimentación de las entradas.

### 1.3.1. Objetivos específicos.

- Definir la estructura de la neurona artificial dinámica.
- Definir la ley de aprendizaje para los dos pesos sinápticos y el umbral.
- Definir la ley de aprendizaje del peso sináptico  $w_a$  (combinación lineal con el estado),  $w_b$  (combinación lineal con la entrada) y el umbral  $\theta$  con entrada escalar.
- Definir la ley de aprendizaje del peso sináptico  $w_a$  (combinación lineal con el estado),  $W_b$  (combinación lineal con las entradas) y el umbral  $\theta$  con múltiples entradas.
- Comparar el esquema de una sola neurona con redes neuronales artificiales.
- Proponer una metodología para la aproximación de sistemas utilizando una neurona artificial dinámica.

### 1.4. Metas.

- Formalizar el resultado indicando las condiciones necesarias para lograr la aproximación.
- Formalizar el resultado en un teorema que manifieste las condiciones necesarias para hacer estable el error de aproximación y acotarlo.
- Demostrar el teorema mediante un análisis estilo Lyapunov.
- Propuesta de una metodología para la aproximación de sistemas estáticos y dinámicos.

### 1.5. Justificación.

- La propuesta presentada en este trabajo de tesis corresponde a una sola estructura neuronal con una sola ley de aprendizaje para realizar el modelado de sistemas dinámicos y sistemas estáticos.
- La aproximación de sistemas mediante la neurona artificial dinámica propuesta es una técnica de modelado muy útil cuando las leyes físicas del comportamiento del sistema no son muy claras, o no se tiene conocimiento suficiente de la estructura interna o las variables y parámetros que interactúan dentro del sistema.
- La neurona artificial dinámica tiene una estructura paralela y permite aproximación en línea.
- La neurona artificial dinámica puede ajustar de manera automática sus parámetros durante el entrenamiento mediante la ley de aprendizaje.
- Para estructurar y proponer soluciones a problemas con un grado de dificultad mayor, se antepone el estudio preliminar que sirve como conocimiento de base, en este caso una neurona artificial dinámica para que de manera futura en otros trabajos se aborden redes neuronales artificiales dinámicas.



## 1.6. Alcances.

El desarrollo de esta tesis está acotado a realizar las siguientes actividades:

- Proponer una metodología para el modelado de sistemas estáticos y dinámicos utilizando una neurona artificial dinámica, para obtener aproximaciones que representen de forma adecuada al sistema que se está analizando, esto quiere decir que el modelo presente resultados positivos al realizar alguna prueba de desempeño.
- Programación de todas las funciones que se necesiten para efectuar los pasos necesarios que culminen con la aproximación de sistemas estáticos y dinámicos, desde la elección del sistema hasta las pruebas de desempeño de los modelos.
- Se establecerán distintos casos de estudio, en los cuales se definirán puntualmente cada uno de los parámetros y sistemas, tanto estáticos y dinámicos, como ejemplos para verificar que la estructura neuronal propuesta en esta tesis funciona de manera correcta.

## 1.7. Hipótesis.

La hipótesis de este trabajo de tesis se compone de una afirmación:

- Con la propuesta de una metodología, para el modelado entrada-salida de sistemas estáticos y dinámicos con una sola estructura neuronal dinámica, es posible realizar aproximaciones que resulten en un error estable y acotado.

## 1.8. Estructura de la tesis.

En el capítulo 2 se proporcionan conceptos y definiciones fundamentales relacionadas con esquemas de neuronas artificiales propuestos y el método del gradiente descendente aplicado al entrenamiento y ajuste de los parámetros de las neuronas artificiales propuestas. Además, se presentan algunas definiciones importantes de cálculo diferencial aplicado a vectores y matrices. Así mismo, se brindan conceptos básicos relacionados con el principio del teorema de estabilidad de Lyapunov, aprendizaje en línea y fuera de línea en el ámbito de redes neuronales artificiales.

En los capítulos 3 y 4 se muestran los resultados obtenidos a partir de la simulación de los casos de estudio propuestos en esta investigación, y la formalización del problema con un teorema que manifiesta las condiciones necesarias para lograr la obtención de modelos.

En el capítulo 5 se presenta una breve comparación de la neurona artificial dinámica propuesta en el trabajo de investigación con una RNA 2-1 de una entrada existente en la literatura, sus diferencias, sus ventajas y desventajas en la aproximación de sistemas estáticos y dinámicos.

Finalmente, en el capítulo 6 se proporcionan las conclusiones generales formuladas a partir de los resultados obtenidos, así como, los trabajos futuros propuestos para este tema de tesis.

# Capítulo 2

---

## Marco Teórico

---

En este capítulo se presentan las definiciones y conceptos que brindan el sustento teórico de este trabajo de investigación.

### 2.1. Modelado de Neuronas Biológicas

Se sabe que el cerebro humano posee de  $10^{10}$  a  $10^{11}$  neuronas comunicadas con una red nerviosa altamente interconectada. Una estructura simplificada de una neurona es la mostrada en la figura 2.1.1. Las partes básicas mostradas en el esquema son:

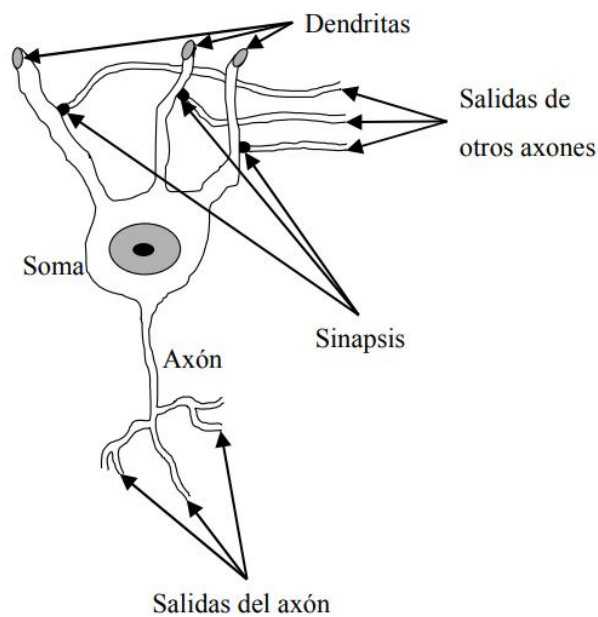


Figura 2.1.1: Neurona biológica [1].

- Soma o cuerpo celular, es la parte central de la neurona en la que son realizadas casi todas las funciones lógicas, es en esta parte donde se encuentra el mecanismo genético y metabólico que mantiene viva a la neurona, así como la síntesis de proteínas.
- El axón o salida es un nervio conectado al soma, que posteriormente está altamente ramificado, en el segmento inicial del axón la señal es transformada en impulsos nerviosos los cuales se propagan sin atenuación a células receptoras, musculares o bien otras neuronas.

- Las dendritas son ramificaciones densas de fibras nerviosas largas y de forma irregular y están conectadas al soma, por cada neurona hay de  $10^3$  a  $10^4$  dendritas. Las dendritas reciben señales de otras neuronas por medio de un contacto especializado conocido como sinapsis.
- Las sinapsis son los puntos terminales de los axones de otras neuronas. La función de la sinapsis tiene una naturaleza excitatoria o inhibitoria, por esto tiene la habilidad de incrementar o atenuar la excitación de la neurona. La sinapsis humana y en general la de los animales son un proceso químico muy complejo, pero la sinapsis de animales primitivos como los insectos están basados en transmisiones puramente eléctricas.

Cabe recordar que una neurona biológica posee procesos muy complejos que en su totalidad no han sido conocidos. Sin embargo, existe flexibilidad en el modelado de una neurona biológica, es decir, una neurona artificial es diseñada para realizar funciones o cálculos que sean específicos, de manera que la arquitectura de la misma es muy particular para cada problema que se desee resolver con esta técnica [1].

## 2.2. Modelado Estático de Neuronas Artificiales

Una neurona artificial estática muestra un mapeo estático de las entradas hacia la salida. Una forma básica de este modelo es como el que propuso McCulloch Pitts [24], [1] en la cual se expresa la salida  $y$  de la neurona de la siguiente manera

$$\begin{aligned}
 y &= \varphi(w_1x_1 + w_2x_2 + \dots + w_nx_n + \theta) = \varphi(Wx + \theta) \\
 &= \varphi\left(\sum_{i=1}^n w_i x_i + \theta\right).
 \end{aligned}
 \tag{2.1}$$

Otra forma que es común encontrar de los modelos de las neuronas artificiales, es a través de representaciones en esquemas gráficos, los cuales son muy útiles para visualizar la analogía que tienen con una neurona biológica. Para el caso del modelo McCulloch Pitts el esquema correspondiente es el de la figura 2.2.1, donde  $x_i$  es la  $i$ -ésima entrada de la neurona.  $w_i$  es el  $i$ -ésimo peso de la neurona y modela la acción de la sinapsis de una neurona biológica. El punto suma, el umbral  $\theta$  y la función de activación  $\varphi(u)$  corresponden a el cuerpo de la neurona. Finalmente la salida  $y$  representa el axón de la neurona biológica. La función de activación en general es no lineal y acotada, generalmente se escoge el perfil sigmoïdal, pero puede tener entre otros perfiles el de saturación o la función escalón (Anexo A).

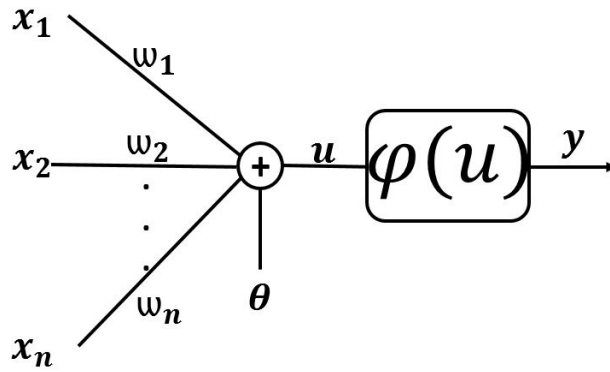


Figura 2.2.1: Neurona Artificial Estática.

El perfil sigmoideal (Anexo A) es utilizado en arquitecturas de neuronas artificiales en las que la ley de aprendizaje utiliza métodos que requieren de funciones suaves, como el algoritmo de retropropagación [25] o el de gradiente por mencionar algunos. Pero es posible que las leyes de aprendizaje no demanden que las funciones de activación sean suaves, tal es el caso del perceptrón ([26], [27], [28], [29]) que utiliza la función escalón. El problema de interés en cualquier arquitectura es el ajuste de las sinapsis o pesos para realizar una tarea determinada, a esta estrategia de ajuste se le conoce como ley de aprendizaje. La estrategia más básica de una ley de aprendizaje consiste en el ajuste de los pesos de una neurona artificial para minimizar una función de costo, por lo cual es fácil encontrar el uso de las neuronas artificiales para resolver problemas de optimización [30].

Si el ajuste de pesos se realiza antes de implementar la neurona artificial, se le conoce como ley de aprendizaje fuera de línea, y una vez realizado el ajuste, la neurona artificial se implementa con sus pesos fijos. Pero si este ajuste se realiza simultáneamente con la neurona artificial implementada, este ajuste es conocido como ley de aprendizaje en línea. En este caso los pesos o aprendizaje puede sufrir modificaciones ante posibles perturbaciones en la tarea que realizan [1].

### 2.3. Modelado Dinámico de Neuronas Artificiales

Las neuronas artificiales dinámicas poseen al menos una señal retroalimentada. Esta retroalimentación puede estar hecha en tiempo discreto o en tiempo continuo.

Uno de los modelos más populares de una neurona artificial dinámica es el propuesto por Hopfield [31], cuya estructura es la siguiente:

$$\begin{aligned} \dot{u} &= -w_a u + Wx + \theta \\ y &= \varphi(u) \end{aligned} \tag{2.2}$$

donde,  $x = [x_1, x_2, \dots, x_n]^T$  es el vector de entrada,  $W = [w_1, w_2, \dots, w_n]$  es el vector de pesos sinápticos,  $u$  es el argumento de la función de activación, ésta última es escogida con un perfil como el de la tangente hiperbólica, finalmente la señal  $y$  es la salida de la neurona. El modelo de Hopfield se puede implementar con un circuito eléctrico Resistivo-Capacitivo (RC) conectado a un amplificador no lineal con una función de perfil sigmoideal como se muestra en la figura (2.3.1). Donde la correspondencia de los parámetros del circuito con respecto al modelo matemático (2.2), se construye con

$$w_a = \frac{1}{CR_0}; \quad w_j = \frac{1}{CR_i}, \quad j = 1, 2, \dots, n; \quad (2.3)$$

el umbral simplemente es la fuente independiente de corriente, esto es,  $\theta = i$ ; una sinapsis excitatoria se construye escogiendo a la entrada correspondiente como  $+x_j$ , y a la acción inhibitoria de la sinapsis se construye escogiendo a la entrada correspondiente como  $-x_j$ .

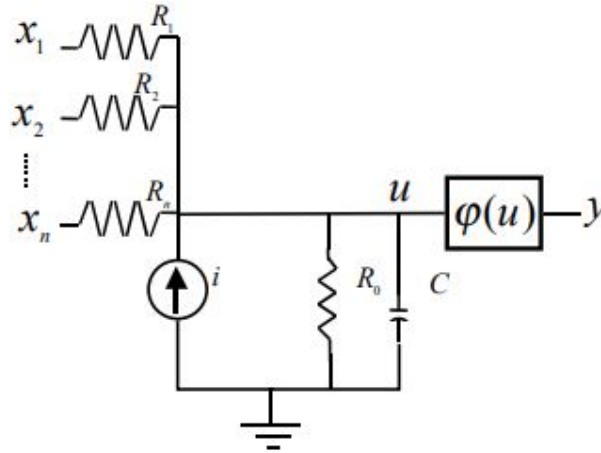


Figura 2.3.1: Neurona Artificial Dinámica (Hopfield) [1].

Como puede observarse una neurona artificial dinámica es un sistema no lineal dinámico, que tiene la característica de cambiar su comportamiento a lo largo del tiempo para lograr un propósito a través del ajuste de sus pesos (aprendizaje). Por esta razón es que se aprovechan las redes neuronales dinámicas en la identificación y control de sistemas con no linealidades complejas [1].

## 2.4. Puntos de Equilibrio

La riqueza dinámica de los sistemas no lineales presenta ciertos fenómenos que no son evidentes en los sistemas lineales [32]. Uno de estos fenómenos es la existencia de múltiples puntos de equilibrio aislados. Un sistema lineal puede tener un solo punto de equilibrio aislado, y por lo tanto un solo estado de régimen estacionario que -si el punto es estable- atrae al estado del sistema independientemente del estado inicial. En cambio, los sistemas no lineales pueden tener varios puntos de equilibrio, y la convergencia a uno estable depende del estado inicial. Es por ello que, resulta importante estudiar la estabilidad de los diferentes puntos de equilibrio de los sistemas no lineales con lo cual se logra un mejor entendimiento del comportamiento del mismo [33].

### 2.4.1. Estabilidad de Lyapunov

Un punto de equilibrio se dice estable si todas las soluciones que se inicien en las cercanías del punto de equilibrio permanecen en las cercanías del mismo, de otro modo el punto de equilibrio es inestable. Un punto de equilibrio se dice asintóticamente estable si todas las soluciones que inicien en las cercanías del punto de equilibrio no sólo permanecen en las cercanías del punto de equilibrio, sino que además tienden hacia el equilibrio a medida que el tiempo se aproxima a infinito.

**Teorema 2.1 (Método Directo).** Sea el origen  $x = 0$  un punto de equilibrio del sistema no lineal  $\dot{x} = f(x)$  donde  $f : D \subset \mathbb{R}^n$  es una función continuamente diferenciable y  $D \subset \mathbb{R}^n$  es un entorno del origen. Sea

$$A = \left. \frac{\partial f(x)}{\partial x} \right|_{x=0}$$

entonces, el origen es ASINTÓTICAMENTE ESTABLE si todos los valores propios de  $A$  tienen parte real negativa. El origen es INESTABLE si uno o más valores propios de  $A$  tienen parte real positiva.

**Teorema 2.2 (Método Indirecto).** Sea el origen  $x = 0$  un punto de equilibrio de  $\dot{x} = f(x)$  y sea  $D \subset \mathbb{R}^n$  un dominio que contiene al origen.  $D \subset \mathbb{R}^n$  es un entorno del origen. Sea  $V : D \rightarrow \mathbb{R}$  una función continuamente diferenciable tal que:

$$\begin{aligned} V(0) = 0, V(x) > 0 \quad \forall x \in D \setminus \{0\}, \\ \dot{V}(x) \leq 0 \quad \forall x \in D, \end{aligned}$$

entonces  $x = 0$  es ESTABLE, más aún, sí

$$\dot{V}(x) < 0,$$

entonces  $x = 0$  es ASINTÓTICAMENTE ESTABLE.

Los teoremas 2.1 y 2.2 con sus respectivas demostraciones se encuentran en [32], [34] y [35].

## 2.5. La Derivada Parcial

Para determinar la velocidad o el ritmo de cambio de una función de varias variables respecto a una de sus variables independientes se utiliza el proceso de derivación parcial [36].

Si  $z = f(x, y)$  la derivada parcial de  $f$  con respecto a la variable  $x$  es (2.4)

$$\frac{\partial z}{\partial x} = \frac{\partial f}{\partial x}(x, y) := \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h}, \quad (2.4)$$

y la derivada parcial de  $f$  con respecto a  $y$  es (2.5)

$$\frac{\partial z}{\partial y} = \frac{\partial f}{\partial y}(x, y) := \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h}, \quad (2.5)$$

siempre y cuando el límite en cada caso exista.

La definición indica que para calcular  $\frac{\partial f}{\partial x}$  se considera  $y$  constante derivando con respecto a  $x$  y para calcular  $\frac{\partial f}{\partial y}$  se considera  $x$  constante derivando con respecto a  $y$ . Pueden aplicarse por tanto las reglas usuales de derivación.

## 2.6. Regla de la Cadena

Si se tiene  $F(x) = h[g(x)]$ .

Donde  $x \in \mathfrak{R}^n$  y  $F(\cdot), h(\cdot), g(\cdot)$  son funciones escalares. La regla de la cadena de derivación puede ser escrita como se muestra en (2.6) [30]:

$$\frac{\partial F}{\partial x_i} = \frac{\partial h}{\partial g} \frac{\partial g}{\partial x_i}. \quad (2.6)$$

## 2.7. La Derivada Parcial de una Función Integral

La regla de derivación bajo la integral es la siguiente igualdad expresada en (2.7)[37]:

$$\frac{\partial}{\partial \lambda} \int f(x, \lambda) dx = \int \frac{\partial f}{\partial \lambda} (x, \lambda) dx, \quad (2.7)$$

que dice esencialmente que cuando se deriva una integral respecto de un parámetro distinto a la variable en la que se integra, derivada e integral pueden intercambiarse. Según el sentido que se quiera dar a la integral y a la derivada, este resultado requiere distintas condiciones sobre la función  $f$ . A veces esta regla se llama también regla de Leibniz.

## 2.8. Derivación Matricial

La derivada de una función matricial de variable matricial no difiere sustancialmente del concepto de derivada habitual. La diferencia y dificultad se presenta cuando es preciso manipular estas derivadas matriciales, debido a su dimensión y la ubicación de sus elementos [38].

En esta sección se indican las definiciones de funciones escalares, vectoriales y matriciales, de variable escalar, vectorial o matricial, adoptando un criterio común para la colocación de los distintos elementos que componen cada una de estas derivadas [39].

Cuando se ha de calcular el gradiente de  $y = f(x)$  se respeta la ordenación del vector fila  $x = (x_1, \dots, x_n)$  de las variables explicativas; luego parece lógico que este mismo criterio de ubicación de elementos se adopte para definir la derivada de una función escalar de variable matricial [38].

### 2.8.1. Definición 1. Derivada de un escalar con respecto a un vector

Sea  $y = f(x)$  una función real de variable vectorial, es decir,  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ . La derivada de  $y$  respecto de  $x$  es el vector fila

$$\frac{\partial y}{\partial x} = \left[ \frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right],$$

se adopta esta notación ya que se está derivando un escalar  $f(x)$  con respecto a un vector fila  $x = (x_1, \dots, x_n)$ . Obsérvese que  $\frac{\partial f(x)}{\partial x}$  es el vector gradiente de la función  $f(x)$ [39].

### 2.8.2. Definición 2. Derivada de un vector con respecto a un escalar

Sea  $y = f(x)$  una función vectorial de variable real, dicho de otro modo,

$$y = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$

lo cual significa que  $f : \mathfrak{R} \rightarrow \mathfrak{R}^m$ . La derivada del vector  $y$  respecto del escalar  $x$  es el vector columna [39]

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x} \\ \vdots \\ \frac{\partial f_m(x)}{\partial x} \end{bmatrix}$$

**2.8.3. Definición 3. Derivada de un vector con respecto a un vector (Jacobiano)**

Sea  $y = f(x)$  una función vectorial de variable vectorial, donde  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ . Como

$$y = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$

entonces:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x} \\ \vdots \\ \frac{\partial f_m(x)}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}.$$

Obsérvese que esta matriz de orden  $m \times n$  es el Jacobiano de la función  $f(x)$ , ya que cada una de sus filas es el gradiente de las funciones  $f_1(x), \dots, f_m(x)$ , que son las componentes de  $f(x)$  [39].

**2.8.4. Definición 4. Derivada de un escalar con respecto a una matriz**

Sea  $y = f(X)$  una función real de variable matricial, donde  $f : \mathfrak{R}^{p \times n} \rightarrow \mathfrak{R}$ . La derivada de  $y$  respecto de la matriz  $X$  de orden  $p \times n$ , es la matriz del mismo orden dado por [39]

$$\frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial f(X)}{\partial x_{11}} & \dots & \frac{\partial f(X)}{\partial x_{1n}} \\ \vdots & & \vdots \\ \frac{\partial f(X)}{\partial x_{p1}} & \dots & \frac{\partial f(X)}{\partial x_{pn}} \end{bmatrix}.$$

**2.8.5. Definición 5. Derivada de una matriz con respecto a un escalar**

Sea  $Y = F(x)$  una función matricial de variable escalar, donde  $F : \mathfrak{R} \rightarrow \mathfrak{R}^{m \times q}$ . La derivada de  $Y$  respecto de la variable  $x$ , es la matriz de orden  $m \times q$  dada por [39]

$$\frac{\partial Y}{\partial x} = \begin{bmatrix} \frac{\partial f_{11}(x)}{\partial x} & \dots & \frac{\partial f_{1q}(x)}{\partial x} \\ \vdots & & \vdots \\ \frac{\partial f_{m1}(x)}{\partial x} & \dots & \frac{\partial f_{mq}(x)}{\partial x} \end{bmatrix}.$$

**2.8.6. Definición 6. Derivada de un vector con respecto a una matriz**

Sea  $y = f(X)$  una función vectorial de variable matricial, donde  $f : \mathfrak{R}^{p \times n} \rightarrow \mathfrak{R}^m$ . La derivada de  $y$  respecto de la matriz  $X$  de orden  $p \times n$ , es la matriz de orden  $m \times n$  dada por

$$\frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial f_1(X)}{\partial X} \\ \vdots \\ \frac{\partial f_m(X)}{\partial X} \end{bmatrix},$$



ya que, de acuerdo con la definición 4 [39]

$$\frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial f_1(X)}{\partial x_{11}} & \dots & \frac{\partial f_1(X)}{\partial x_{1n}} \\ \vdots & & \vdots \\ \frac{\partial f_1(X)}{\partial x_{p1}} & \dots & \frac{\partial f_1(X)}{\partial x_{pn}} \\ \vdots & & \vdots \\ \frac{\partial f_m(X)}{\partial x_{11}} & \dots & \frac{\partial f_m(X)}{\partial x_{1n}} \\ \vdots & & \vdots \\ \frac{\partial f_m(X)}{\partial x_{p1}} & \dots & \frac{\partial f_m(X)}{\partial x_{pn}} \end{bmatrix}_{pm \times n}$$

### 2.8.7. Definición 7. Derivada de una matriz con respecto a un vector

Sea  $Y = F(x)$  una función matricial de variable vectorial, donde  $F : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times q}$ . La derivada de  $Y$  respecto del vector  $x$ , es la matriz de orden  $m \times q \times n$  cuya expresión es

$$\frac{\partial Y}{\partial x} = \begin{bmatrix} \frac{\partial f_{11}(x)}{\partial x} & \dots & \frac{\partial f_{1q}(x)}{\partial x} \\ \vdots & & \vdots \\ \frac{\partial f_{m1}(x)}{\partial x} & \dots & \frac{\partial f_{mq}(x)}{\partial x} \end{bmatrix},$$

donde, teniendo en cuenta la definición 1 [39],

$$\frac{\partial Y}{\partial x} = \begin{bmatrix} \frac{\partial f_{11}(x)}{\partial x_1} & \dots & \frac{\partial f_{11}(x)}{\partial x_n} & \dots & \frac{\partial f_{1q}(x)}{\partial x_1} & \dots & \frac{\partial f_{1q}(x)}{\partial x_n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial f_{m1}(x)}{\partial x_1} & \dots & \frac{\partial f_{m1}(x)}{\partial x_n} & \dots & \frac{\partial f_{mq}(x)}{\partial x_1} & \dots & \frac{\partial f_{mq}(x)}{\partial x_n} \end{bmatrix}_{m \times q \times n}$$

### 2.8.8. Definición 8. Derivada de una matriz con respecto a una matriz

Sea  $Y = F(X)$  una función matricial de variable matricial, donde  $F : \mathbb{R}^{p \times n} \rightarrow \mathbb{R}^{m \times q}$ . La derivada de  $Y$  respecto de la matriz  $X$ , es la matriz de orden  $p \times m \times n \times q$  cuya expresión es

$$\frac{\partial Y}{\partial X} = \begin{bmatrix} \frac{\partial f_{11}(X)}{\partial X} & \dots & \frac{\partial f_{1q}(X)}{\partial X} \\ \vdots & & \vdots \\ \frac{\partial f_{m1}(X)}{\partial X} & \dots & \frac{\partial f_{mq}(X)}{\partial X} \end{bmatrix}$$

ya que, teniendo en cuenta la definición 4, [39]

$$\frac{\partial Y}{\partial X} = \begin{bmatrix} \frac{\partial f_{11}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{11}(X)}{\partial x_{1n}} & \dots & \frac{\partial f_{1q}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{1q}(X)}{\partial x_{1n}} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial f_{11}(X)}{\partial x_{p1}} & \dots & \frac{\partial f_{11}(X)}{\partial x_{pn}} & \dots & \frac{\partial f_{1q}(X)}{\partial x_{p1}} & \dots & \frac{\partial f_{1q}(X)}{\partial x_{pn}} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial f_{m1}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{m1}(X)}{\partial x_{1n}} & \dots & \frac{\partial f_{mq}(X)}{\partial x_{11}} & \dots & \frac{\partial f_{mq}(X)}{\partial x_{1n}} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial f_{m1}(X)}{\partial x_{p1}} & \dots & \frac{\partial f_{m1}(X)}{\partial x_{pn}} & \dots & \frac{\partial f_{mq}(X)}{\partial x_{p1}} & \dots & \frac{\partial f_{mq}(X)}{\partial x_{pn}} \end{bmatrix}_{pm \times nq}$$

## 2.9. Propiedades de Derivación de Funciones Matriciales

En esta sección se recopilan algunas de las propiedades que aparecen con mayor frecuencia. Los resultados presentados se obtienen utilizando las reglas de derivación, las diferentes propiedades tanto del producto de Kronecker como de la vectorización de matrices [39].

Proposición. Sea  $A \in \mathfrak{R}^{m \times n}$  y sean  $x \in \mathfrak{R}^n$ ,  $z \in \mathfrak{R}^m$ . Entonces

1. Si  $y(x) = Ax$  se verifica que  $\frac{\partial y(x)}{\partial x} = A$ .

2. Si  $y(x) = Ax$  se verifica que  $\frac{\partial y(x)}{\partial A} = x^T$ .

3. Si  $y(x, z) = z^T Ax$  se verifica que

$$\frac{\partial y}{\partial x} = z^T A \quad \frac{\partial y}{\partial z} = x^T A^T.$$

4. Si  $m = n$  e  $y(x) = x^T Ax$ , se verifica que

$$\frac{\partial y(x)}{\partial x} = x^T (A + A^T), \quad \frac{\partial^2 y(x)}{\partial x \partial x} = (A + A^T),$$

y si además  $A$  es una matriz simétrica se tiene que

$$\frac{\partial y(x)}{\partial x} = 2x^T A, \quad \frac{\partial^2 y(x)}{\partial x \partial x} = 2A.$$

5. Si  $m = n$ ,  $A = A^T$ ,  $X \in \mathfrak{R}^{n \times p}$  e  $Y(X) = X^T AX$ , se verifica que

$$\frac{\partial Y(X)}{\partial X} = P_{n,p}(AX \otimes I_p) + (\text{vec}(AX))(\text{vec}(I_p))^T.$$

## 2.10. Método Directo de Lyapunov

El matemático Ruso A.M. Lyapunov formuló una poderosa teoría de estabilidad y propuso una técnica para probarla. Uno de los caminos más comunes para estudiar la convergencia de un sistema dinámico descrito por un conjunto de ecuaciones diferenciales es encontrar una función de Lyapunov, también llamada una función de energía desde el método de Lyapunov el cual se basa en el concepto de energía y la relación de la energía almacenada para la estabilidad del sistema [30].

### 2.10.1. Principio del Teorema de Estabilidad de Lyapunov

De manera general un sistema es estable si su función de energía total ( $J$ ), es una función definida positiva, además es continuamente decreciente, es decir, la derivada con respecto al tiempo de la  $J$  total es definida negativa, hasta que alcanza un estado de equilibrio. En un sistema no lineal no tiene por qué existir una manera simple de definir una función de energía, por ello, el método de estabilidad por Lyapunov introduce una función de energía auxiliar denominada función candidata de Lyapunov [40].

### 2.10.1.1. Función Candidata de Lyapunov

Suponga que  $J : \mathbb{R}^n \rightarrow \mathbb{R}$  es una función escalar.  $J$  es una función candidata de Lyapunov si es una función positiva definida localmente, es decir, la función candidata de Lyapunov es mayor que cero para todo valor en los reales excepto el cero, y el único valor donde tiene un valor de cero es el mismo cero (2.8):

$$\begin{aligned} J(0) &= 0, \\ J(x) &> 0 \quad \forall x \in \mathbb{R} \setminus \{0\}. \end{aligned} \tag{2.8}$$

### 2.10.1.2. Teorema de Estabilidad de Lyapunov

**Teorema 2.3.** Considere el sistema  $\dot{x} = f(x, t)$ , donde  $f(0, t) = 0$ . Si  $\exists$  una función escalar  $J(x, t)$  con las primeras derivadas parciales continuas que satisface las condiciones (2.9, 2.10 y 2.11)

$$J(0) = 0, \tag{2.9}$$

$$J(x, t) \text{ es definida positiva, es decir, } J(x, t) > 0 \quad \forall x, t \neq 0, \tag{2.10}$$

$$\dot{J}(x, t) \text{ es definida negativa, es decir, } \dot{J}(x, t) < 0 \quad \forall x, t \neq 0. \tag{2.11}$$

entonces el estado de equilibrio en el origen es uniforme y asintóticamente estable.

Si se satisface (2.9 y 2.10) pero  $\dot{J}$  es semidefinida negativa, es decir,  $\dot{J}(x, t) \leq 0 \quad \forall x$  entonces el estado de equilibrio en el origen es estable.

El teorema de Lyapunov no dice nada acerca de la forma de la función  $J$  o cómo construirla. Da sólo una condición suficiente para la convergencia cuando  $t \rightarrow \infty$  y no se preocupa por un tiempo finito de convergencia. Desafortunadamente, no hay un método general para la construcción de funciones de Lyapunov.

### 2.10.2. El Algoritmo de Gradiente Descendente Desde la Perspectiva de Lyapunov

Este método transforma un problema de minimización en un sistema asociado de ecuaciones diferenciales ordinarias de primer orden como se presenta en (2.12) [30].

$$\dot{x}_j = - \sum_{i=1}^n \mu_{ji} \frac{\partial J}{\partial x_i} \tag{2.12}$$

Con condiciones iniciales  $x_j(0) = x_j^0$ . El sistema (2.12) puede ser escrito en la forma matricial compacta como se expresa en (2.13).

$$\dot{x} = -\mu(x, t) \frac{\partial J}{\partial x} \tag{2.13}$$

Donde:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_n \end{bmatrix}^T, \\ x &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T, \\ \frac{\partial J}{\partial x} &= \begin{bmatrix} \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial x_2} & \cdots & \frac{\partial J}{\partial x_n} \end{bmatrix}^T. \end{aligned}$$

$\mu(x, t)$  es una matriz simétrica positiva definida también llamada matriz de aprendizaje con entradas dependientes. Además para encontrar el vector deseado  $x^*$  que minimiza la función

$J(x)$  necesitamos resolver o simular el sistema de ecuaciones diferenciales con condiciones iniciales. Esto quiere decir que el mínimo de la función de energía o la función objetivo es determinado por la siguiente expresión que se muestra en (2.14).

$$x^* = \lim_{t \rightarrow \infty} x(t) \tag{2.14}$$

Para mostrar que el sistema de ecuaciones diferenciales expresado en (2.13) es estable dejamos la derivada con respecto al tiempo de la función energía como se muestra en (2.15).

$$\dot{J} = \frac{\partial J}{\partial x} \dot{x} = - \left( \frac{\partial J}{\partial x} \right)^T \mu(x, t) \left( \frac{\partial J}{\partial x} \right) \leq 0 \tag{2.15}$$

Tomando la condición que la matriz  $\mu(x, t)$  es simétrica y positiva definida. La ecuación (2.15) garantiza que la función de energía  $J(x)$  decrece en el tiempo y converge a un mínimo local estable cuando  $t \rightarrow \infty$ . La velocidad de convergencia depende de la elección de la matriz de entradas  $\mu(x, t)$ . En el caso más simple la matriz  $\mu(x, t)$  es reducida a una matriz unitaria multiplicada por una constante positiva  $\mu_0$ , dando como resultado un sistema simplificado de la forma expresada en (2.16).

$$\dot{x} = -\mu_0 \frac{\partial J(x)}{\partial x}, \quad x(0) = x^0. \tag{2.16}$$

Donde el coeficiente positivo  $\mu_0$ , es llamado el parámetro de aprendizaje.

### 2.11. Aprendizaje en línea

Se basa en ajustar los parámetros de la neurona artificial dinámica en tiempo real, esto quiere decir que, se introduce una señal de excitación de forma paralela a la planta y a la neurona artificial dinámica, y en base a la corrección del error, que es igual a la salida de la planta menos la salida de la neurona artificial dinámica se ajustan los parámetros de la neurona con el objetivo de minimizar el error conforme transcurre el tiempo (ver figura 2.11.1).

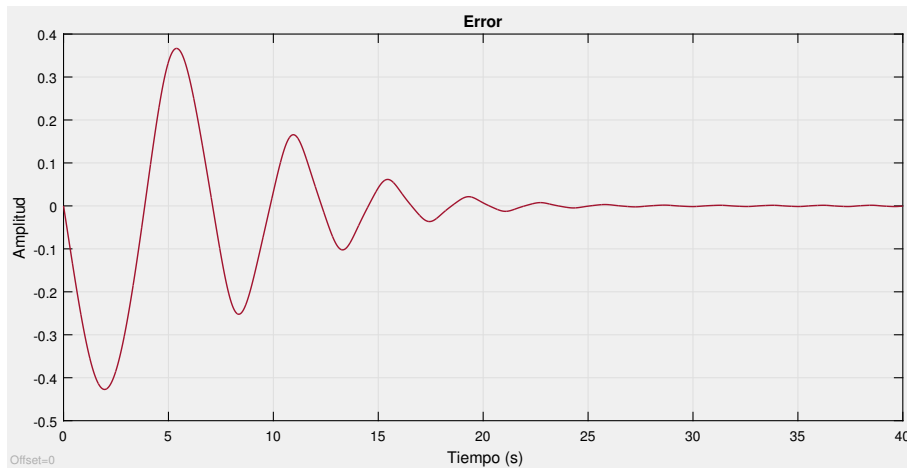


Figura 2.11.1: Error de aproximación 1

### 2.12. Aprendizaje fuera de línea

Modifica los parámetros de la neurona artificial dinámica después de la presentación de un patrón de datos entrada-salida, obtenidos mediante un experimento directo con el sistema real,

y utiliza el último valor como las condiciones iniciales de los parámetros de la neurona artificial dinámica en la siguiente evaluación del mismo patrón de datos entrada-salida, hasta lograr minimizar el error cuadrático medio sobre el patrón de datos. Esto se logra modificando todos los parámetros de la neurona en dirección opuesta al gradiente de la función de error (ver figura 2.12.1).

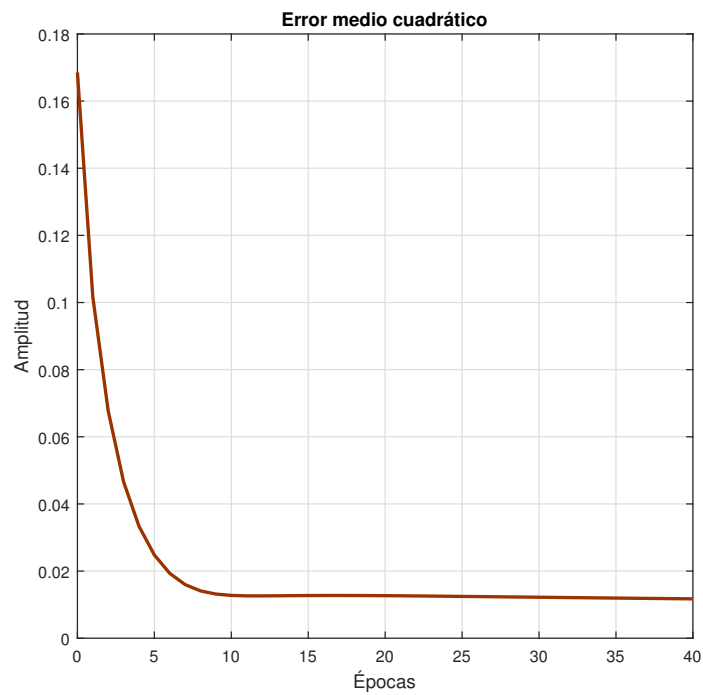


Figura 2.12.1: Error de aproximación 2

# Capítulo 3

---

## Modelado Entrada-Salida de Sistemas Estáticos y Dinámicos

---

El análisis de estabilidad para el error de aproximación de la neurona artificial dinámica se presenta mediante el método de estabilidad estilo Lyapunov, en el cuál se precisan las condiciones suficientes que se deben cumplir para hacer estable el error de aproximación y además acotarlo. Este análisis se presenta de manera formal con la realización de un teorema, que sería uno de los principales resultados de este trabajo de investigación. Además de que se presenta una metodología propuesta para la obtención de un sistema lineal invariante en el tiempo, un sistema no lineal afín con la entrada, un sistema estático lineal y un sistema estático no lineal con múltiples entradas.

Se desea aproximar sistemas estáticos y dinámicos con la misma estructura neuronal. La neurona artificial dinámica busca aproximar sistemas dinámicos lineales y no lineales. Bajo la postura filosófica de que un sistema dinámico que de manera muy rápida logra su estado estable (dinámicas muy rápidas) se le considera bajo ciertas condiciones prácticamente una función estática, se aprovecha dicha postura para que una neurona dinámica pueda aproximar sistemas estáticos lineales y no lineales.

**Suposición 1.** Suponga que la entrada del sistema que se busca aproximar está acotada y puede ser de  $m$  dimensiones, es decir  $u(t) \in \Omega^m \subset \mathfrak{R}^m$ . A su vez la salida se considera que también está acotada, es decir  $y_d(t) \in \mathcal{C} \subset \Omega$ .

El subconjunto  $\Omega$  se define como el conjunto de elementos  $k_0$  que van desde su valor mínimo  $\underline{k}_0$  hasta su valor máximo  $\bar{k}_0$ , dicho así

$$\Omega = \{k_0, \quad \underline{k}_0 \leq k_0 \leq \bar{k}_0\}.$$

### 3.1. Estructura neuronal dinámica

La estructura propuesta de la neurona artificial dinámica para realizar el modelado entrada-salida de sistemas estáticos y dinámicos es la presentada en (3.1)

$$\sum_{\phi} : \begin{cases} \dot{x}(t) = -\frac{1}{\tau_p}x(t) + \frac{1}{\tau_p}\alpha \tanh(v_1(t) - v_2(t)) \\ v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = w_a(t) \cdot x(t) + w_b(t) \cdot u(t) + \theta(t) \\ y(t) = x(t) \end{cases} \quad (3.1)$$

donde:

$\alpha$ = amplitud de la tangente hiperbólica.

$\tau_p$ = constante de tiempo de la neurona artificial dinámica.

$w_a(t) \in \mathfrak{R}^2$ = peso sináptico (combinación lineal con el estado) de la neurona dinámica.

$w_b(t) \in \mathfrak{R}^2$ = peso sináptico (combinación lineal con la entrada) de la neurona dinámica.

$\theta(t) \in \mathfrak{R}^2$ = umbral de la neurona dinámica.

$y(t) = x(t) \in \mathcal{C} \subset \Omega$ = estado de la neurona dinámica.

**Observación 1.** Nótese que el sistema (3.1) tiene disponible el estado a la salida y es un escalar acotado, en otras palabras,  $y(t) = x(t) \in \mathcal{C} \subset \Omega$ . La estructura neuronal recibe las señales de entrada y salida de forma paralela al sistema que se desea aproximar.

**Observación 2.** Nótese que el término neuronal  $\tanh(v_1(t) - v_2(t))$  en (3.1) plantea una diferencia del vector de argumento  $v(t)$ , esto con el fin de darle más holgura a los espacios nulos, en otros palabras, el argumento de la tangente hiperbólica sólo se anulará cuando  $v_1(t) - v_2(t) = 0$ .

### 3.1.1. Puntos de equilibrio de la neurona artificial dinámica

Para calcular los puntos de equilibrio de la neurona artificial dinámica presentada en (3.1), se procede a igualar la derivada de la neurona a cero, como se desarrolla a continuación. Dicho desarrollo da origen a la observación 3.

$$\dot{x}(t) = -\frac{1}{\tau_p}x(t) + \frac{1}{\tau_p}\alpha \cdot \tanh(v_1(t) - v_2(t)),$$

$$0 = -\frac{1}{\tau_p}x + \frac{1}{\tau_p}\alpha \cdot \tanh((v_1(t) - v_2(t)))|_{(v_1(t)-v_2(t))=v_o},$$

despejando  $\tau_p$  se tiene que

$$-x + \alpha \cdot \tanh(v_o) = 0,$$

$$x_0 = \alpha \cdot \tanh(v_0).$$

**Observación 3.** El punto de equilibrio para la neurona artificial propuesta se expresa como

$$x_0 = \alpha \cdot \tanh(v_0),$$

en otras palabras, el valor final para la neurona artificial dinámica depende directamente del valor  $v_0$  evaluado dentro de la función de activación, multiplicado por la amplitud.

### 3.1.2. Constante de tiempo $\tau_p$

La constante de tiempo  $\tau_p$  es un parámetro importante en la estructura neuronal propuesta en (3.1), ya que es un indicador de la velocidad de reacción de la neurona artificial dinámica ante una señal de excitación. Explicado de otra forma quiere decir que, entre más pequeño sea este parámetro, la neurona artificial dinámica alcanzará su valor final (punto de equilibrio) más rápido. Para aplicar la teoría de control clásico, se procede a analizar la neurona artificial dinámica (3.1) en la región lineal, donde  $\tanh(v) = v$ , quedando un sistema de la forma

$$\dot{x} = -\frac{1}{\tau_p}x + \frac{1}{\tau_p}\alpha \cdot v, \quad (3.2)$$

entonces por retroalimentación del estado  $v = kx$ , se analiza si es posible modificar  $\tau_p$ , dicho de otra manera, obtener un sistema de la forma (3.3)

$$\dot{x} \cong -\frac{1}{\tau_d}x, \quad (3.3)$$

Sustituyendo la ley de control  $v$  en (3.2) se tiene que

$$\dot{x} \cong -\frac{1}{\tau_p}x + \frac{1}{\tau_p}\alpha \cdot kx \cong \frac{-x + \alpha \cdot kx}{\tau_p}, \quad (3.4)$$

desarrollando la ecuación (3.4) y agrupando términos queda de la forma:

$$\dot{x} \cong \frac{(-1 + \alpha \cdot k)x}{\tau_p}.$$

Como el propósito de esta ley de control es llegar a un sistema de la forma expresado en (3.3), es decir, un sistema autónomo. El sistema queda de la forma expresado en (3.5)

$$\dot{x} \cong -\frac{(1 - \alpha \cdot k)}{\tau_p}x, \quad (3.5)$$

ya que el objetivo es analizar la constante de tiempo deseada  $\tau_d$  del sistema, se compara la ecuación (3.3) con la ecuación (3.5), y realizando la analogía del recíproco de  $\tau_d$  queda expresado como se muestra en (3.6)

$$\frac{1}{\tau_d} = \frac{1 - \alpha \cdot k}{\tau_p} > 0, \quad (3.6)$$

despejando  $\tau_d$  queda de la siguiente forma expresado en (3.7)

$$\tau_d = \frac{\tau_p}{1 - \alpha \cdot k} > 0. \quad (3.7)$$

La primer restricción que se genera al estar tratando con la constante de tiempo del sistema es que  $\tau_p > 0$ , de igual manera para que se siga cumpliendo que  $\tau_d > 0$  surge la restricción de que  $\alpha \cdot k < 1$ . Realizando las manipulaciones matemáticas correspondientes de esta desigualdad, se puede obtener la forma de la ganancia  $k$ , como se expresa en la ecuación (3.8)

$$\alpha \cdot k < 1,$$

$$k < \frac{1}{\alpha}. \quad (3.8)$$



El propósito de este análisis es definir cómo es la constante de tiempo deseada del sistema, o sea, describir el intervalo en el que se encuentra el valor mínimo y el valor máximo que puede tomar, dicho de otra manera,  $\underline{\tau}_d < \tau_d < \bar{\tau}_d$ . Así que se define el intervalo de la ganancia  $k$  como se expresa en (3.9)

$$-\infty < k < \frac{1}{\alpha}, \quad (3.9)$$

de (3.7) se evalúa el límite cuando la ganancia  $k$  tiende a los valores del intervalo de (3.9) como se expresa en (3.10) y (3.11)

$$\lim_{k \rightarrow -\infty} \tau_d = \lim_{k \rightarrow -\infty} \frac{\tau_p}{1 - \alpha k} = \frac{\tau_p}{\infty} = 0, \quad (3.10)$$

$$\lim_{k \rightarrow \frac{1}{\alpha}} \tau_d = \lim_{k \rightarrow \frac{1}{\alpha}} \frac{\tau_p}{1 - \alpha k} = \frac{\tau_p}{0} = \infty. \quad (3.11)$$

El denominador obtiene su valor más pequeño cuando la ganancia  $k$  tiende a su valor más grande. El denominador obtiene su valor más grande cuando la ganancia  $k$  tiende a su valor más pequeño, es decir la constante de tiempo deseada del sistema puede ser tan grande como se quiera y tan chica como se necesite, pero sin llegar a ser cero, como se expresa en (3.12)

$$0 < \tau_d < \infty. \quad (3.12)$$

Se puede concluir que la constante de tiempo del sistema puede ser tan grande y tan chica como se quiera, sin llegar a ser cero o un valor negativo. Realizando la analogía con la neurona artificial dinámica propuesta, el peso sináptico que retroalimenta el estado se encargará de modificar la velocidad de respuesta de la neurona, sin la necesidad de ajustar la constante de tiempo  $\tau_p$  durante el entrenamiento.

### 3.2. Teorema para aproximación de sistemas

Con el uso de la estructura neuronal propuesta en (3.1) y las herramientas de base que se citaron en el marco teórico se fórmula el teorema 3.1 con  $m = 1$ , esto es, con entrada escalar, dicho así, el problema consiste en obtener un modelo entrada salida que describa el comportamiento de distintos tipos de sistemas. En la figura 3.2.1 se muestra un esquema general del trabajo de tesis.

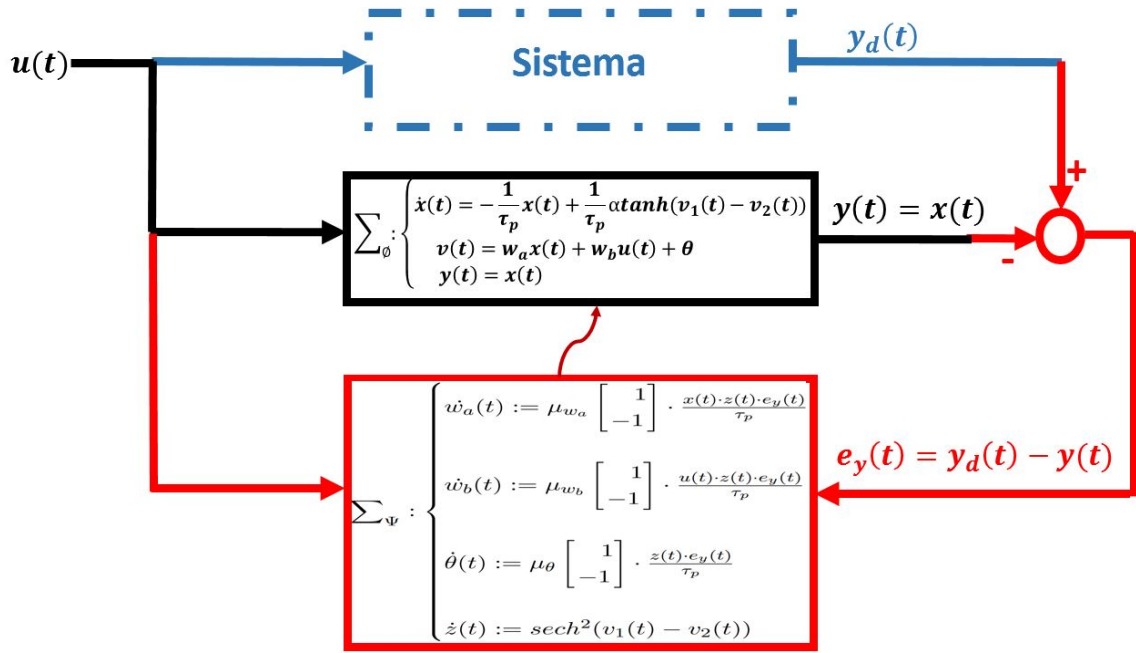


Figura 3.2.1: Esquema General

**Teorema 3.1.** Considere un sistema en el que se tiene acceso al par de datos entrada-salida, el cual es aproximado por  $\sum_\phi$  (3.1) y a su vez la suposición 1 se satisface, si se selecciona la ley de aprendizaje dinámica  $\sum_\Psi$  (3.13) como

$$\sum_\Psi : \begin{cases} \dot{w}_a(t) := \mu_{w_a} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \\ \dot{w}_b(t) := \mu_{w_b} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{u(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \\ \dot{\theta}(t) := \mu_\theta \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{z(t) \cdot e_y(t)}{\tau_p} \\ \dot{z}(t) := \text{sech}^2(v_1(t) - v_2(t)) \end{cases}, \quad (3.13)$$

donde las matrices de coeficientes de convergencia son simétricas y positivas definidas, estos es,

$$\mu_{w_a} = \mu_{w_a}^T > 0,$$

$$\mu_{w_b} = \mu_{w_b}^T > 0,$$

$$\mu_\theta = \mu_\theta^T > 0,$$

entonces el error de aproximación  $e_y(t) = y_d(t) - y(t)$  es estable.

### 3.3. Demostración del teorema para aproximación de sistemas

De igual forma para demostrar el teorema 3.1 se emplea un análisis estilo Lyapunov, en donde como primer paso se selecciona a  $J(w_a, w_b, \theta, t)$  la función candidata de Lyapunov siguiente

$$J = \frac{1}{2}e_y^2(t), \quad (3.14)$$

la función candidata presentada en (3.14) cumple las 2 primeras condiciones de estabilidad de Lyapunov que son (3.15)

$$\begin{aligned} J(0) &= 0, \\ J &> 0. \end{aligned} \quad (3.15)$$

Para calcular la derivada con respecto al tiempo de la función candidata de Lyapunov  $J(w_a, w_b, \theta, t)$  que depende de los parámetros a minimizar y el tiempo, se le aplica la regla de la cadena dando como resultado (3.16)

$$\dot{J} = \frac{\partial J}{\partial w_a} \dot{w}_a + \frac{\partial J}{\partial w_b} \dot{w}_b + \frac{\partial J}{\partial \theta} \dot{\theta}, \quad (3.16)$$

posteriormente se procede a calcular el primer término del lado derecho de la igualdad presentada en (3.16), que es

$$\frac{\partial J}{\partial w_a} \dot{w}_a,$$

para lo que se necesita desarrollar la siguiente regla de la cadena (3.17)

$$\frac{\partial J}{\partial w_a} = \frac{\partial J}{\partial e_y} \cdot \frac{\partial e_y}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial w_a}, \quad (3.17)$$

así que desarrollando cada derivada parcial de (3.17) se obtiene el siguiente resultado

$$\frac{\partial J}{\partial e_y} = \frac{\partial}{\partial e_y} \left( \frac{1}{2} e_y^2 \right) = e_y,$$

$$\frac{\partial e_y}{\partial y} = \frac{\partial}{\partial y} (y_d - y) = -1,$$

invocando la regla de Leibniz de derivación bajo la integral, integral y derivada se intercambian convenientemente para resolver la derivada parcial de la salida  $y(t)$  con respecto al vector  $v(t)$  como se desarrolla a continuación

$$\begin{aligned} \frac{\partial y}{\partial v(t)} &= \frac{\partial}{\partial v(t)} \left[ \int_0^t \left( -\frac{1}{\tau_p} x(\beta) + \frac{1}{\tau_p} \tanh(v_1(\beta) - v_2(\beta)) \right) d\beta \right] = \\ &= \int_0^t \left[ \frac{\partial}{\partial v(t)} \left( -\frac{1}{\tau_p} x(t) + \frac{1}{\tau_p} \tanh(v_1(t) - v_2(t)) \right) \Big|_{t=\beta} \right] d\beta = \\ &= \frac{1}{\tau_p} \int_0^t (\text{sech}^2(v_1(\beta) - v_2(\beta)) \cdot [1, \quad -1]) d\beta = \frac{1}{\tau_p} \int_0^t \text{sech}^2(v_1(\beta) - v_2(\beta)) d\beta \cdot [1, \quad -1]. \end{aligned}$$

La siguiente derivada a resolver en la regla de la cadena es la derivada parcial del vector  $v(t)$  con respecto al vector de pesos sinápticos  $w_a(t)$ , la solución es un campo vectorial mejor conocido en el área de control automático como Jacobiano

$$\frac{\partial v}{\partial w_a} = \frac{\partial(w_a \cdot x(t) + w_b \cdot u(t) + \theta)}{\partial w_a} = \frac{\partial(w_a \cdot x(t))}{\partial w_a} = \begin{bmatrix} \frac{\partial v_1}{\partial w_{a1}} & \frac{\partial v_1}{\partial w_{a2}} \\ \frac{\partial v_2}{\partial w_{a1}} & \frac{\partial v_2}{\partial w_{a2}} \end{bmatrix} = \begin{bmatrix} x(t) & 0 \\ 0 & x(t) \end{bmatrix} = I \cdot x(t) = \text{diag}_2\{x(t)\}.$$

Posteriormente se sustituyen los resultados de cada derivada en la ecuación (3.17) quedando como sigue

$$\frac{\partial J}{\partial w_a} = -\frac{e_y}{\tau_p} \int_0^t \text{sech}^2(v_1(\beta) - v_2(\beta)) d\beta \cdot [1, \quad -1] \cdot \text{diag}_2\{x(t)\}.$$

Nótese que la variable  $z$  se define en la ley de aprendizaje (3.13) como

$$z := \int_0^t \text{sech}^2(v_1(\beta) - v_2(\beta)) d\beta,$$

de modo que, se sustituye en  $\frac{\partial J}{\partial w_a}$  y resulta (3.18)

$$\frac{\partial J}{\partial w_a} = -\frac{e_y}{\tau_p} \cdot z \cdot [1, \quad -1] \cdot \text{diag}_2\{x(t)\}. \quad (3.18)$$

Retomando el primer término del lado derecho de la igualdad presentada en (3.16) y sustituyendo el resultado obtenido en (3.18) se consigue que

$$\frac{\partial J}{\partial w_a} \dot{w}_a = -\left( \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} [1, \quad -1] \right) \cdot \dot{w}_a$$

a)  $\dot{w}_a$  se toma de la ley de aprendizaje presentada en (3.13),

b)  $\mu_{w_a} = \mu_{w_a}^T > 0$ ,

por lo tanto resulta en un término cuadrático ponderado por una matriz simétrica, positiva definida y además la ecuación presenta un signo negativo, lo que implica que

$$\frac{\partial J}{\partial w_a} \dot{w}_a = -\left( \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} [1, \quad -1] \right) \mu_{w_a} \left( \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \right) \leq 0,$$

es decir,

$$\frac{\partial J}{\partial w_a} \dot{w}_a \leq 0$$

de tal forma que la ecuación presentada en (3.16) se puede expresar como (3.19)

$$\dot{J} \leq \frac{\partial J}{\partial w_b} \dot{w}_b + \frac{\partial J}{\partial \theta} \dot{\theta} \quad (3.19)$$

Continuando con el primer término de la inecuación (3.19) se arranca a calcular la regla de la cadena, dicho de otro modo (3.20)

$$\frac{\partial J}{\partial w_b} = \frac{\partial J}{\partial e_y} \cdot \frac{\partial e_y}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial w_b}. \quad (3.20)$$

Partiendo de las derivadas parciales resueltas en (3.17) se sigue a calcular la derivada parcial restante, dicho de otra manera

$$\frac{\partial v}{\partial w_b} = \frac{\partial(w_a \cdot x(t) + w_b \cdot u(t) + \theta)}{\partial w_b} = \frac{\partial}{\partial w_b}(w_b \cdot u(t)) = \begin{bmatrix} \frac{\partial v_1}{\partial w_{b1}} & \frac{\partial v_1}{\partial w_{b2}} \\ \frac{\partial v_2}{\partial w_{b1}} & \frac{\partial v_2}{\partial w_{b2}} \end{bmatrix} = \begin{bmatrix} u(t) & 0 \\ 0 & u(t) \end{bmatrix} = I \cdot u(t) = \text{diag}_2\{u(t)\},$$

al realizar las sustituciones correspondientes en (3.20) resulta (3.21)

$$\frac{\partial J}{\partial w_b} = -\frac{e_y}{\tau_p} \cdot z \cdot [1, -1] \cdot \text{diag}_2\{u(t)\}. \quad (3.21)$$

Así que abordando el primer término de lado derecho de la inecuación (3.19) y sustituyendo los resultados obtenidos en (3.21) se tiene que

$$\frac{\partial J}{\partial w_b} \dot{w}_b = -\left( \frac{u(t) \cdot z(t) \cdot e_y(t)}{\tau_p} [1, -1] \right) \cdot \dot{w}_b$$

a)  $\dot{w}_b$  se toma de la ley de aprendizaje presentada en (3.13),

b)  $\mu_{w_b} = \mu_{w_b}^T > 0$ ,

por lo tanto resulta en un término cuadrático ponderado por una matriz simétrica, positiva definida y además la ecuación presenta un signo negativo, lo que implica que

$$\frac{\partial J}{\partial w_b} \dot{w}_b = -\left( \frac{u(t) \cdot z(t) \cdot e_y(t)}{\tau_p} [1, -1] \right) \mu_{w_b} \left( \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{u(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \right) \leq 0,$$

en otras palabras

$$\frac{\partial J}{\partial w_b} \dot{w}_b \leq 0,$$

en consecuencia la inecuación (3.19) se puede reescribir como (3.22)

$$\dot{J} \leq \frac{\partial J}{\partial \theta} \dot{\theta}. \quad (3.22)$$

Finalmente se calcula el último gradiente para completar la siguiente regla de la cadena concierne al término restante del lado derecho de la inecuación (3.22) que es (3.23)

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial e_y} \cdot \frac{\partial e_y}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial \theta}, \quad (3.23)$$

o sea,

$$\frac{\partial v}{\partial \theta} = \frac{\partial(w_a \cdot x(t) + w_b \cdot u(t) + \theta)}{\partial \theta} = \frac{\partial}{\partial \theta}(\theta) = \begin{bmatrix} \frac{\partial v_1}{\partial \theta_1} & \frac{\partial v_1}{\partial \theta_2} \\ \frac{\partial v_2}{\partial \theta_1} & \frac{\partial v_2}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I = \text{diag}_2\{1\},$$

que al sustituir en (3.23) resulta (3.24)

$$\frac{\partial J}{\partial \theta} = -\frac{e_y}{\tau_p} \cdot z \cdot [1, -1] \cdot \text{diag}_2\{1\}. \quad (3.24)$$

Retomando la inecuación (3.22) se sustituyen los valores de los gradientes previamente calculados en (3.24) y se obtiene que

$$\dot{J} \leq -\left(\frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} [1, -1]\right) \cdot \dot{\theta}$$

a)  $\dot{\theta}$  se toma de la ley de aprendizaje presentada en (3.13),

b)  $\mu_\theta = \mu_\theta^T > 0$ ,

por lo tanto resulta en un término cuadrático ponderado por una matriz simétrica, positiva definida y además la inecuación presenta un signo negativo, lo que implica que

$$\dot{J} \leq -\left(\frac{z(t) \cdot e_y(t)}{\tau_p} [1, -1]\right) \mu_\theta \left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{z(t) \cdot e_y(t)}{\tau_p}\right) \leq 0.$$

Con lo que se garantiza que la derivada con respecto al tiempo de la función candidata de Lyapunov es semidefinida negativa, dicho de otra forma (3.25)

$$\dot{J} \leq 0, \quad (3.25)$$

en consecuencia se concluye que el error de aproximación  $e_y(t) = y_d(t) - y(t)$  es estable. Por lo que queda demostrado el teorema. ■

### 3.4. Observaciones del Teorema 3.1

A continuación se enlistan una serie de observaciones derivadas del teorema 3.1 y su demostración.

**Observación 4.** Nótese que en el caso más simple las matrices de coeficientes de convergencia  $\mu_{w_a}, \mu_{w_b}, \mu_\theta$  pueden ser reducidas a una constante positiva  $\mu_0$  postmultiplicado por una matriz identidad, dando como resultado un sistema simplificado.

**Observación 5.** Nótese que la estructura neuronal (3.1) analizada con múltiples entradas, esto quiere decir que  $m \neq 1$ , se modifica de la siguiente manera, donde la combinación lineal de las entradas  $u(t) \in \Omega^m$  con el peso sináptico  $W_b(t) \in \mathfrak{R}^{2 \times m}$  ahora pasa a ser una matriz de pesos sinápticos, dando como resultado la ley de aprendizaje presentada en (3.26).

$$\sum_{\Psi_2} : \begin{cases} \dot{w}_a(t) := \mu_{w_a} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \\ \dot{W}_b(t) := \mu_{w_b} \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(t)^T \cdot \frac{z(t) \cdot e_y(t)}{\tau_p} \\ \dot{\theta}(t) := \mu_{\theta} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{z(t) \cdot e_y(t)}{\tau_p} \\ \dot{z}(t) := \text{sech}^2(v_1(t) - v_2(t)) \end{cases} \quad (3.26)$$

**Observación 6.** Nótese que mediante una manipulación algebraica (Anexo B) se puede concluir que el error es estable y además acotado, esto quiere decir que

$$\lim_{t \rightarrow \infty} |e_y(t)| < \infty. \quad (3.27)$$

### 3.5. Metodología propuesta para aproximación de sistemas

En esta sección se anexa una breve descripción de la metodología propuesta para la aproximación de sistemas utilizando una neurona artificial dinámica.

#### 3.5.1. Pasos para aproximación de sistemas

A continuación se presentan una serie de pasos a seguir para realizar la aproximación de sistemas estáticos y dinámicos, explicadas a detalle con sus respectivas bifurcaciones para decisión.

- **Paso 1.- Inicio.**
- **Paso 2.- Datos de entrada y salida:** Generar la señal de entrada y salida acotados mediante un experimento directo con el sistema que se desea aproximar.
- **Paso 3.- Etapa de aprendizaje:** Se basa en encontrar los parámetros entrenados de la estructura neuronal propuesta, se propone una matriz de coeficientes de convergencia simétrica y positiva definida, y con condiciones iniciales propuestas por el usuario.
- **Paso 4.- Evaluación del desempeño:** Se basa en realizar mediciones del error generado entre el sistema que se está aproximando y la neurona artificial dinámica. El criterio de desempeño se encuentra relajado, ya que es particular para cada sistema que se desee aproximar.

¿Se presentan oscilaciones relevantes en el error de aproximación?

Sí: Ir al paso 3.

No: Ir al paso 5.

- **Paso 5.- Obtención del modelo:** Si el criterio de desempeño es bueno, en otras palabras, que se obtenga un error de aproximación estable y acotado. Se obtiene el modelo con la neurona artificial dinámica, es decir, con los parámetros entrenados

$$J(w_a^*) = \min_{w_a} \{J(w_a)\},$$

$$J(w_b^*) = \min_{w_b} \{J(w_b)\},$$

$$J(\theta^*) = \min_{\theta} \{J(\theta)\}.$$

Dicho de otra forma, el modelo que aproxima la salida de la neurona artificial dinámica a la salida deseada del sistema (3.28).

$$\sum_{\phi_e} : \begin{cases} \dot{x}(t) = -\frac{1}{\tau_p}x(t) + \frac{1}{\tau_p}\alpha \tanh(v_1(t) - v_2(t)) \\ v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = w_a^* \cdot x(t) + w_b^* \cdot u(t) + \theta^* \\ y(t) = x(t) \end{cases} \implies y(t) \cong y_d(t) \quad (3.28)$$

■ Paso 6.-Fin.



# Capítulo 4

---

## Simulaciones

---

En este capítulo se presenta, como primer caso de estudio, la aproximación de un sistema estático lineal con la neurona artificial dinámica. Como segundo caso, la aproximación de un sistema estático no lineal. Como tercer y cuarto caso, la aproximación de un sistema dinámico lineal y no lineal respectivamente. Como quinto caso de estudio se realizó la aproximación de un sistema estático no lineal con múltiples entradas, que corresponde a una compuerta lógica AND. Por último se presenta la aproximación de la dinámica de un biorreactor, con datos experimentales obtenidos directamente de un experimento con la planta. Esto con el fin de verificar el correcto funcionamiento en la aplicación del teorema 3.1, para realizar el modelado entrada-salida de sistemas estáticos y dinámicos.

### 4.1. Simulación 1: Aproximación de un sistema estático lineal

#### 4.1.1. Objetivo de la simulación

Aproximar un sistema estático lineal con la neurona artificial dinámica propuesta para este trabajo de tesis y analizar la convergencia de la ley de aprendizaje presentada en (3.13).

#### 4.1.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.13) para aproximar (4.1)

$$y_d(t) = -0.5 \cdot u(t). \quad (4.1)$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001, \\ \mu_{w_a} &= \text{diag}_2\{1 \times 10^{-7}\}, \\ \mu_{w_b} &= \text{diag}_2\{2 \times 10^{-7}\}, \\ \mu_{\theta} &= \text{diag}_2\{3 \times 10^{-7}\}, \\ \alpha &= 100. \end{aligned}$$

En la figura 3.2.1 se presenta el esquema general utilizado para la simulación con la variante de la ecuación (4.1) que sustituye el bloque de "Sistema". La simulación se llevó a cabo mediante el software Simulink de MatLab.

#### 4.1.3. Desarrollo de la simulación

Las condiciones iniciales para la simulación se inicializan en cero, posteriormente en la figura 4.1.1 se presenta la señal utilizada para entrenar la neurona artificial dinámica.

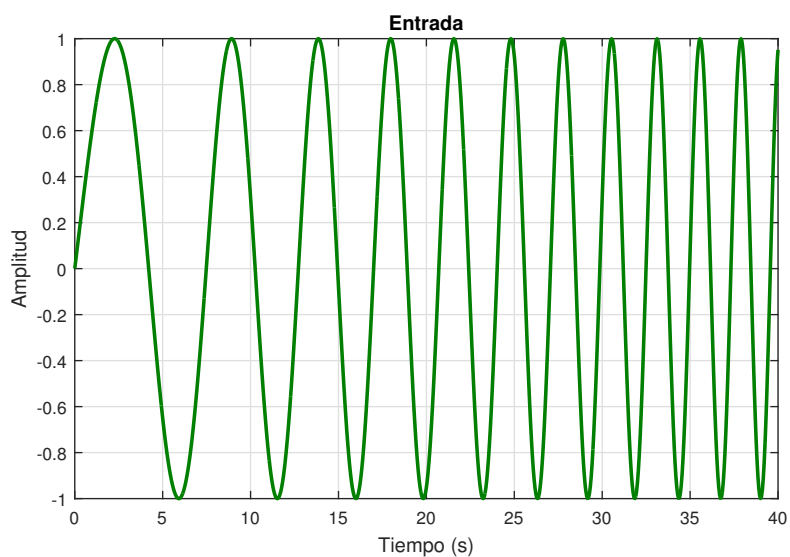


Figura 4.1.1: Señal de entrenamiento

Se realizó el entrenamiento de la NAD para aproximar la función estática de la ecuación (4.1). En la figura 4.1.2, el perfil rojo corresponde a la función estática la cual es la salida deseada del sistema que se busca aproximar, y el perfil negro es la neurona artificial dinámica durante su etapa de entrenamiento, donde se puede observar que las dos señales se empalman en 30 segundos aproximadamente, que es cuando se logra el ajuste de los pesos sinápticos.

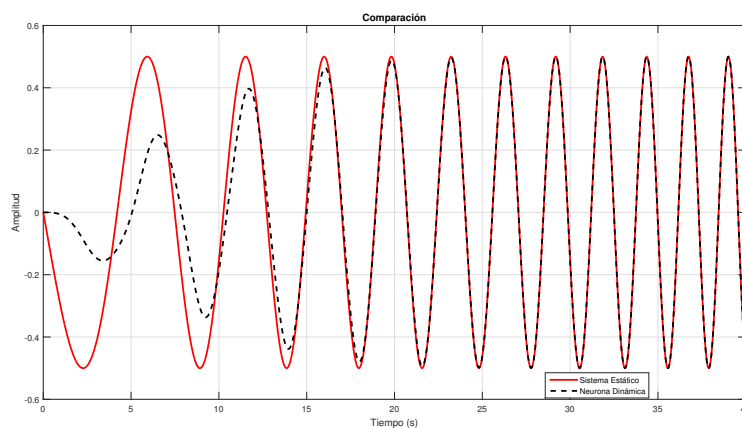


Figura 4.1.2: Aprendizaje de la neurona dinámica

En la figura 4.1.3 se muestra el entrenamiento y ajuste de los pesos sinápticos de la NAD, donde se puede notar que aproximadamente en 30s los pesos sinápticos convergen a un valor constante, en otras palabras, a los parámetros entrenados de la NAD.

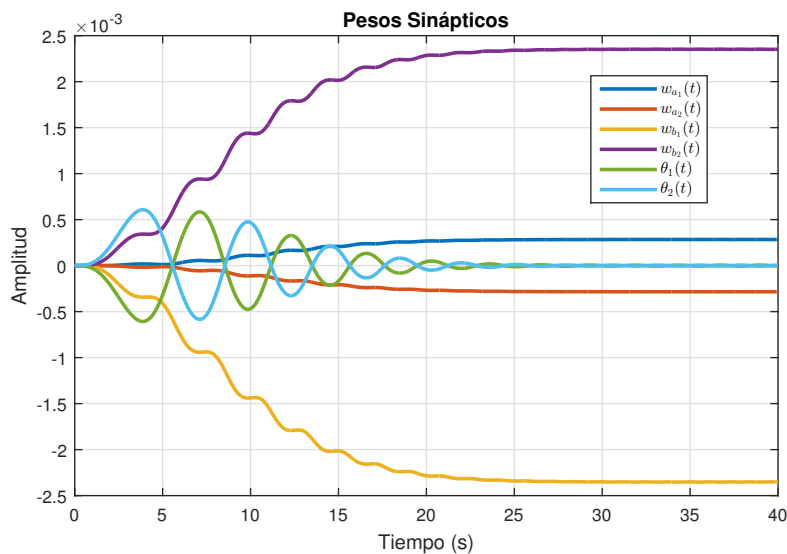


Figura 4.1.3: Convergencia de los pesos sinápticos

Por último en la figura 4.1.4 se observa el error de aproximación, el cual presenta la característica de ser asintótico y estable.

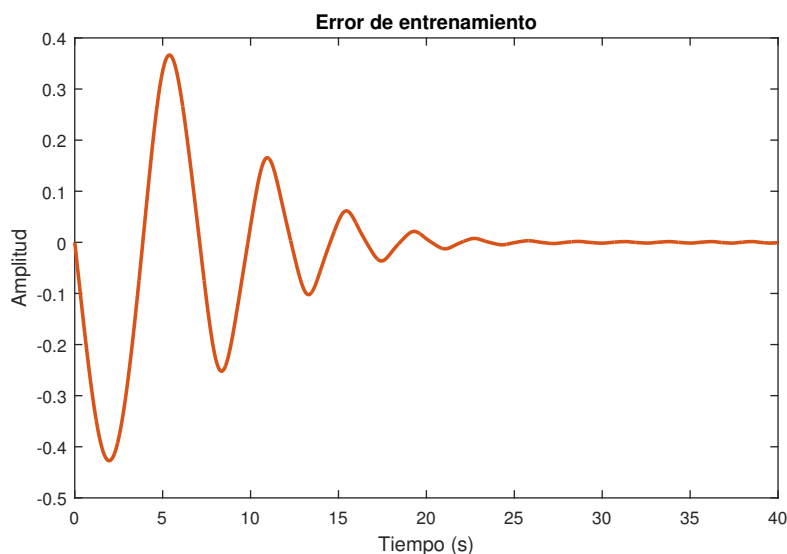


Figura 4.1.4: Error de entrenamiento

A continuación se presentan los valores de los pesos sinápticos entrenados que lograron la aproximación de la función estática lineal

$$w_a^* = \begin{bmatrix} w_{a1}^* \\ w_{a2}^* \end{bmatrix} = \begin{bmatrix} 2.842123699599547 \times 10^{-4} \\ -2.842123699599547 \times 10^{-4} \end{bmatrix}$$

$$w_b^* = \begin{bmatrix} w_{b1}^* \\ w_{b2}^* \end{bmatrix} = \begin{bmatrix} -0.002352397648752 \\ 0.002352397648752 \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} -5.974726358311638 \times 10^{-6} \\ 5.974726358311638 \times 10^{-6} \end{bmatrix}$$

En la figura 4.1.5 y la figura 4.1.6 se presentan señales de validación, para evaluar el desempeño de la neurona artificial dinámica. Como se puede apreciar en la figura 4.1.7 se realizó una comparación en lazo abierto de la neurona artificial dinámica y la función estática lineal, esto con la señal de entrada 1 (4.1.1) utilizada para el entrenamiento, con lo que se observa el ajuste preciso de la neurona dinámica a la trayectoria de la función estática lineal.

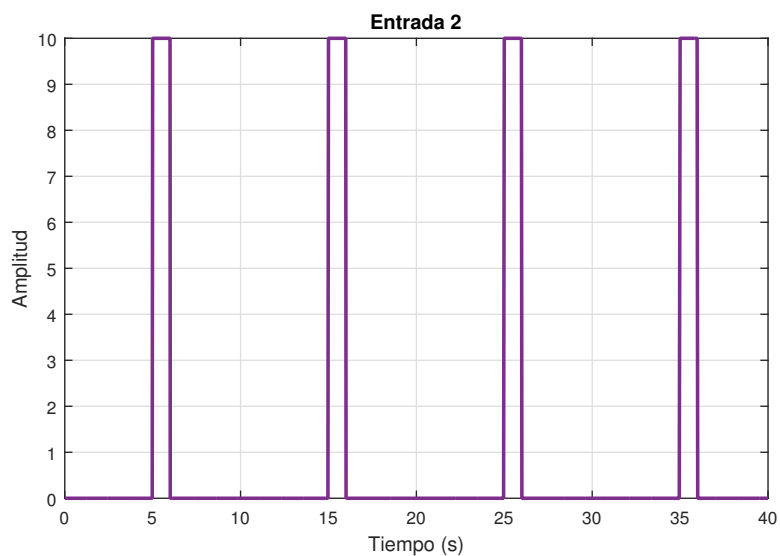


Figura 4.1.5: Señal de validación 1

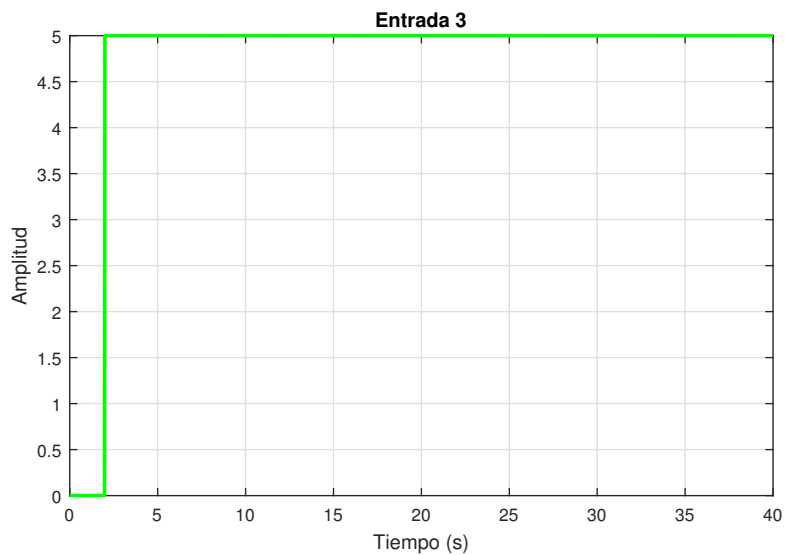


Figura 4.1.6: Señal de validación 2

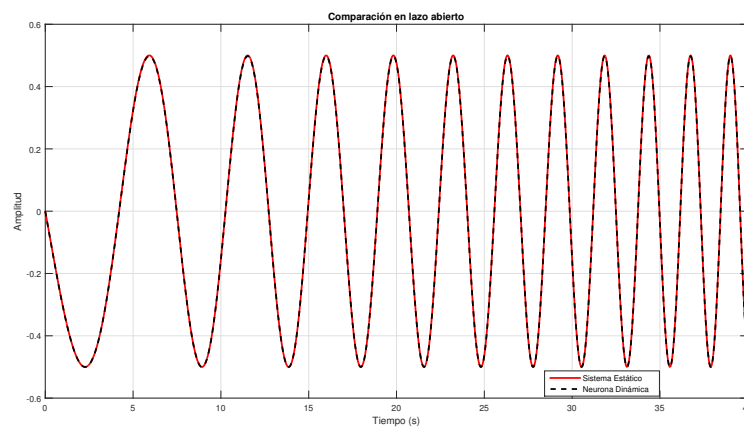


Figura 4.1.7: Respuesta a la entrada de entrenamiento

En la figura 4.1.8 se realiza la segunda prueba en lazo abierto, donde se utilizó la señal de entrada 2 (4.1.5) de validación, con lo que se reafirma el ajuste de la neurona artificial dinámica al tren de pulsos cuadrado.

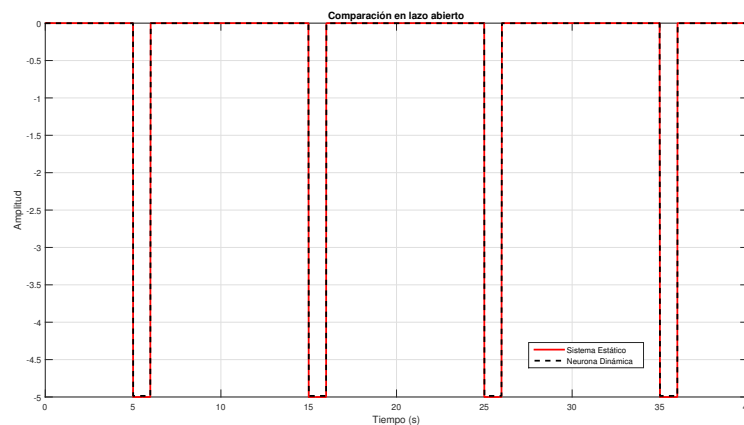


Figura 4.1.8: Respuesta a la entrada 2

Por último en la figura 4.1.9 se visualiza la última prueba en lazo abierto, donde la NAD ajusta la trayectoria de la función estática lineal regido por la señal de entrada 3 (4.1.6).

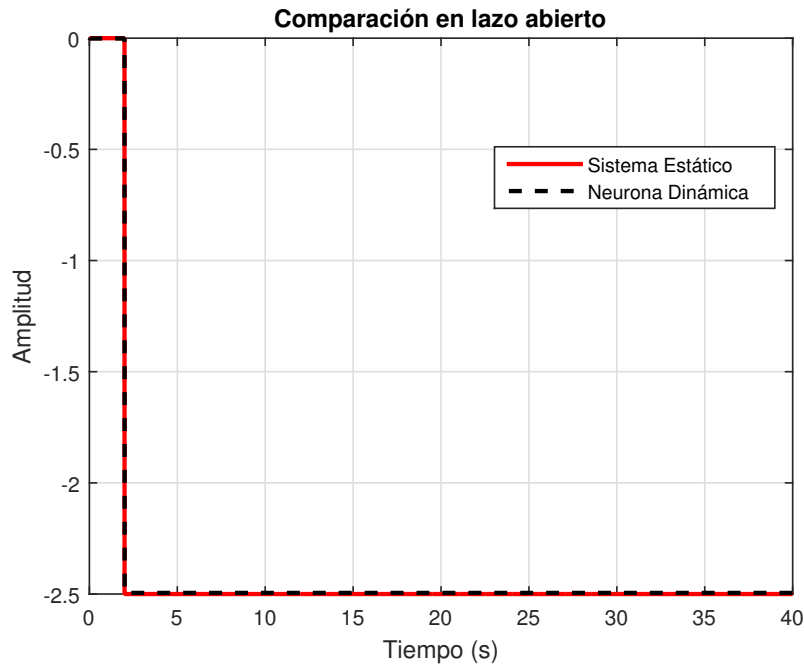


Figura 4.1.9: Respuesta a la entrada 3

#### 4.1.4. Conclusión de la simulación

Se realizó un análisis de la convergencia del error, el cual es asintóticamente estable, con lo que se demuestra que la neurona artificial dinámica puede aproximar una función estática lineal. El sistema entrenado  $\sum_{\phi_{e1}}$  es el modelo matemático de una función estática lineal como la que se muestra en la figura 4.1.10.

$$\sum_{\phi_{e1}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}100 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 2.84 \times 10^{-4} \\ -2.84 \times 10^{-4} \end{bmatrix} \cdot x(t) + \begin{bmatrix} -0.0023 \\ 0.0023 \end{bmatrix} \cdot u(t) + \begin{bmatrix} -5.97 \times 10^{-6} \\ 5.97 \times 10^{-6} \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.2)$$

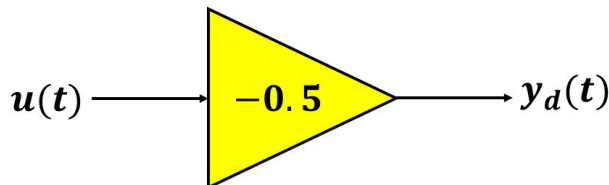


Figura 4.1.10: Función estática lineal.

Por lo tanto se puede concluir que  $y(t) \cong y_d(t)$  para una entrada  $u(t)$  como la que se muestra en 4.1.1, 4.1.5 y 4.1.6.

## 4.2. Simulación 2: Aproximación de un sistema estático no lineal

### 4.2.1. Objetivo de la simulación

Aproximar un sistema estático no lineal con la neurona artificial dinámica propuesta para este trabajo de tesis y analizar la convergencia de la ley de aprendizaje presentada en (3.13).

### 4.2.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.13), para aproximar el comparador de la ecuación (4.3)

$$y_d(t) = \begin{cases} 0.8, & \text{Sen}(t) > 0.2 \\ 0.0, & \text{Sen}(t) \leq 0.2 \end{cases} . \quad (4.3)$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001, \\ \mu_{w_a} &= \text{diag}_2\{1 \times 10^{-9}\}, \\ \mu_{w_b} &= \text{diag}_2\{2 \times 10^{-9}\}, \\ \mu_{\theta} &= \text{diag}_2\{3 \times 10^{-9}\}, \\ \alpha &= 100. \end{aligned}$$

En la figura 3.2.1 se presenta el esquema general utilizado para la simulación con la diferencia de la ecuación (4.3) que sustituye el bloque de "Sistema". La simulación se llevó a cabo mediante el software Simulink de MatLab.

### 4.2.3. Desarrollo de la simulación

Se inicializó con condiciones iniciales igual a cero, y se utilizó una señal de excitación  $u(t) = \text{Sen}(t)$  como la mostrada en la figura 4.2.1.

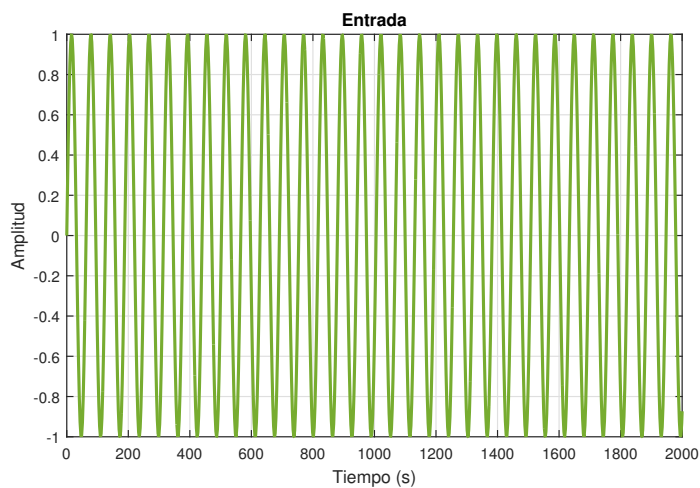


Figura 4.2.1: Señal de entrada

En la figura 4.2.2 se presenta el entrenamiento de la neurona artificial dinámica tratando de aproximar el comparador expresado en (4.3), ya que el criterio de aproximación se encuentra

relajado, entonces la neurona artificial dinámica logra aproximar el comparador debido a que la respuesta en alto se puede interpretar como un 0.8 y la respuesta en 0 se puede interpretar como un bajo.

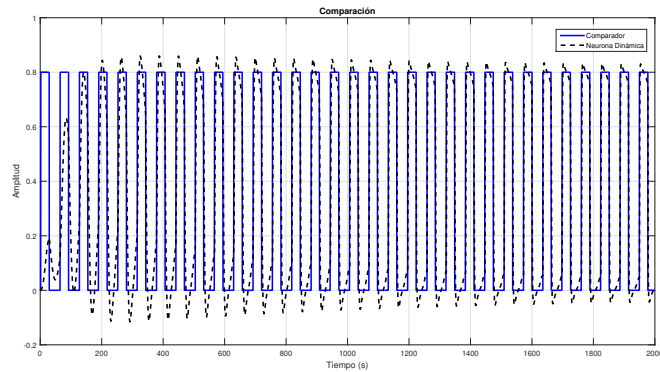


Figura 4.2.2: Aprendizaje de la neurona dinámica 2

En la figura 4.2.3 se muestra como los pesos sinápticos evolucionan en el tiempo para ajustar a la salida deseada del comparador, los cuales permanecen oscilando acotados en una región, por lo que se puede tomar el último valor de pesos sinápticos como los valores entrenados para la NAD.

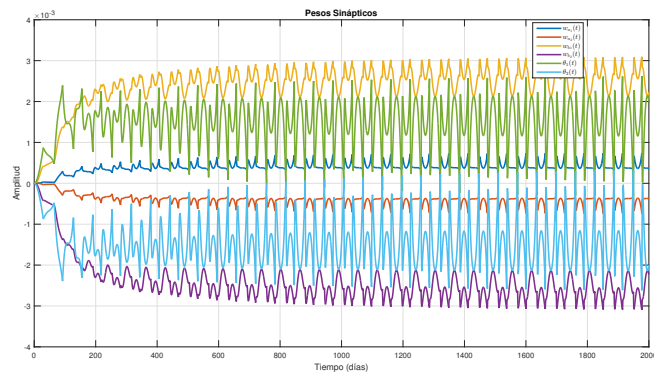


Figura 4.2.3: Convergencia de los pesos sinápticos 2

Por último en la figura 4.2.4 se presenta el error de entrenamiento, donde se visualiza una convergencia estable y acotada.



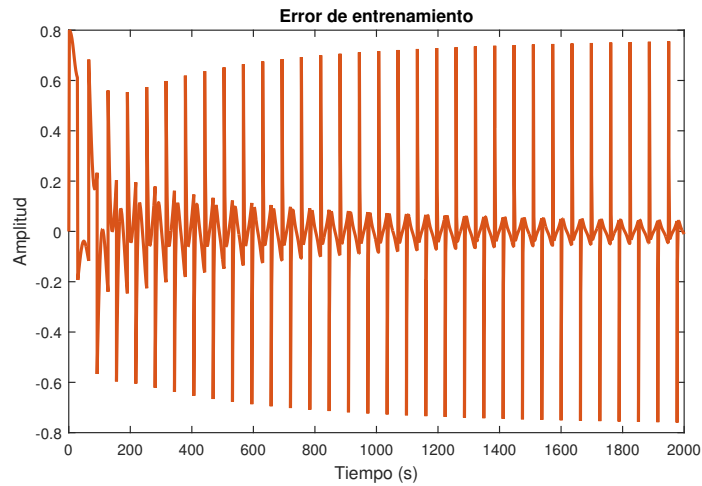


Figura 4.2.4: Error de entrenamiento 2

Los pesos sinápticos entrenados de la neurona artificial dinámica que lograron el ajuste a la trayectoria deseada, que es la salida del comparador son los siguientes:

$$w_a^* = \begin{bmatrix} w_{a1}^* \\ w_{a2}^* \end{bmatrix} = \begin{bmatrix} 3.706241917995522 \times 10^{-4} \\ -3.706241917995522 \times 10^{-4} \end{bmatrix}$$

$$w_b^* = \begin{bmatrix} w_{b1}^* \\ w_{b2}^* \end{bmatrix} = \begin{bmatrix} 0.002228946314657 \\ -0.002228946314657 \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} 0.001995854155572 \\ -0.001995854155572 \end{bmatrix}$$

Como se puede apreciar en la figura 4.2.5 se realizó una comparación en lazo abierto de la neurona artificial dinámica contra la función estática no lineal, esto con la señal de entrada (4.2.1) utilizada para el entrenamiento, con lo que se observa el error de la NAD en el perfil negro, a diferencia de la trayectoria de la función estática no lineal en el perfil azul.

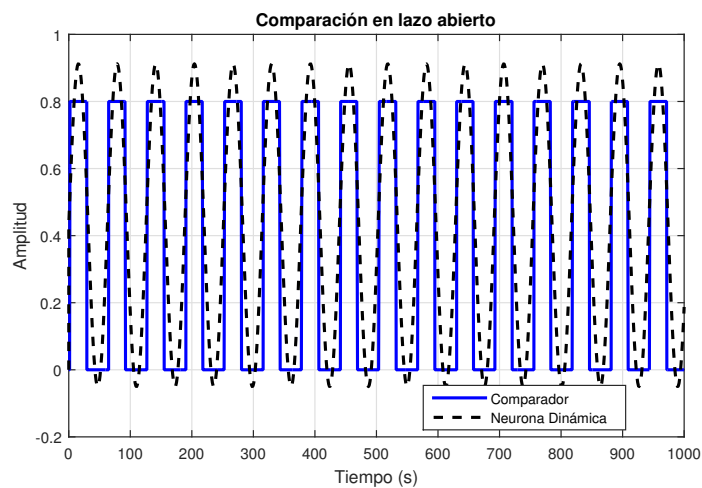


Figura 4.2.5: NAD vs Función Estática No Lineal

#### 4.2.4. Conclusión de la simulación

Se realizó un análisis de la convergencia del error, donde se puede concluir que es estable y acotado tal como lo indica el teorema 3.1, con lo que se demuestra que la neurona artificial dinámica puede aproximar un sistema estático altamente no lineal, es decir, los pesos sinápticos evolucionan en el tiempo para ajustar la trayectoria deseada. En el sistema (4.4) se presenta el modelo matemático, para la señal de entrada  $u(t) = Sen(t)$  de un comparador como el presentado en la figura 4.2.6.

$$\sum_{\phi_{e2}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}100 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 3.70 \times 10^{-4} \\ -3.70 \times 10^{-4} \end{bmatrix} \cdot x(t) + \begin{bmatrix} 0.0022 \\ -0.0022 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 0.0019 \\ -0.0019 \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.4)$$

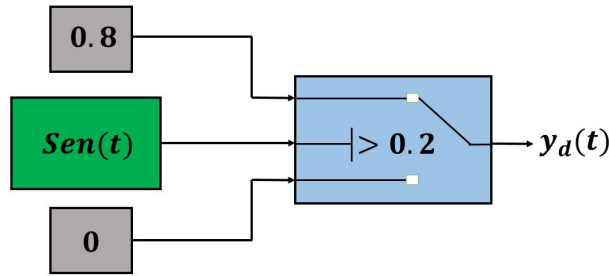


Figura 4.2.6: Función estática no lineal.

Por lo tanto se puede concluir que  $y(t) \cong y_d(t)$ , para una entrada  $u(t) = Sen(t)$ .

### 4.3. Simulación 3: Aproximación de un sistema dinámico lineal

#### 4.3.1. Objetivo de la simulación

Aproximar un circuito RC con la neurona artificial dinámica (3.1) propuesta para este trabajo de tesis y analizar la convergencia de la ley de aprendizaje calculada en (3.13).

#### 4.3.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.13) para aproximar el sistema dinámico de la ecuación (4.5)

$$\sum_{LTI} : \begin{cases} \dot{v}_c(t) = -\frac{1}{RC}v_c(t) + \frac{1}{RC}v_{in}(t) \\ y_d(t) = v_c(t) \end{cases} \quad (4.5)$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001, \\ \mu_{w_a} &= \text{diag}_2\{1 \times 10^{-9}\}, \\ \mu_{w_b} &= \text{diag}_2\{2 \times 10^{-9}\}, \end{aligned}$$

$$\begin{aligned}\mu_\theta &= \text{diag}_2\{3 \times 10^{-9}\}, \\ RC &= \frac{1}{100}, \\ \alpha &= 100.\end{aligned}$$

En la figura 3.2.1 se presenta el esquema general utilizado para la simulación con la diferencia de la sustitución del bloque "Sistema" por la ecuación (4.5). La simulación se llevó a cabo mediante el software Simulink de MatLab.

### 4.3.3. Desarrollo de la simulación

Las condiciones iniciales para la simulación son cero, posteriormente en la figura 4.3.1 se presenta la señal utilizada para entrenar la neurona dinámica, es decir,  $u(t) = v_{in}(t) = \sin(t)$ .

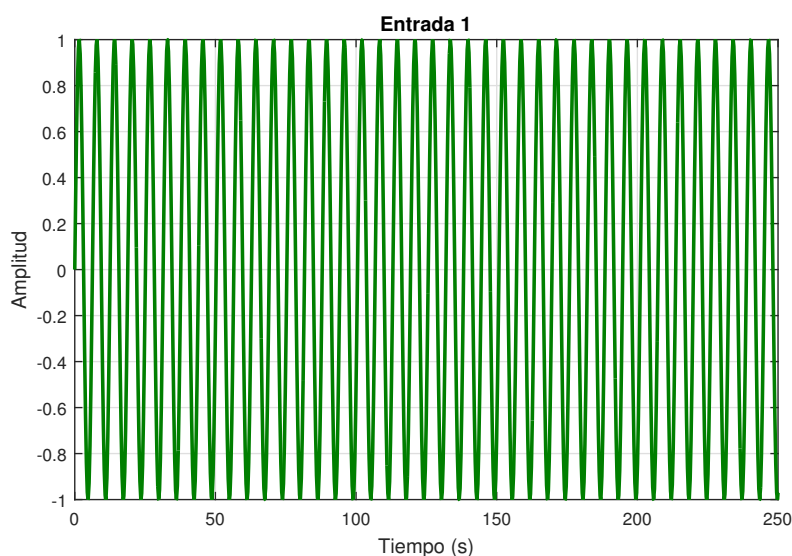


Figura 4.3.1: Señal senoidal de entrenamiento

Se realizó el entrenamiento de la neurona para aproximar el circuito RC de la ecuación (4.5). En la figura 4.3.2, el perfil magenta corresponde al voltaje del capacitor el cual es la salida deseada del sistema y el perfil negro es la neurona artificial dinámica durante su etapa de entrenamiento, donde se aprecia que en 200 segundos aproximadamente logra el ajuste a la salida deseada.

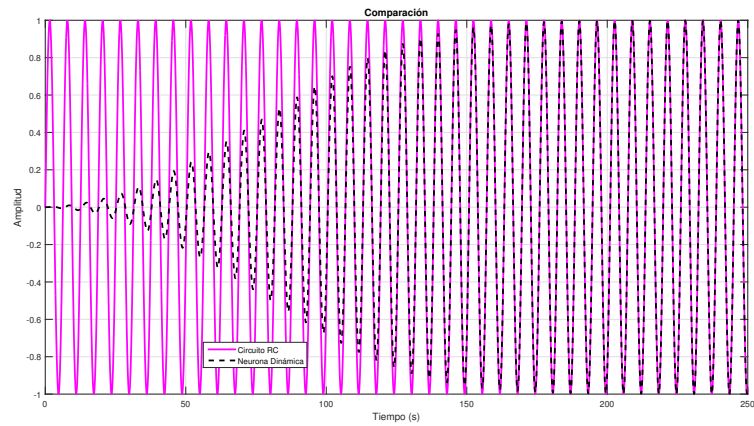


Figura 4.3.2: Aprendizaje de la neurona dinámica 3

En la figura 4.3.3 se muestra el entrenamiento y ajuste de los pesos sinápticos de la neurona artificial dinámica, donde se puede notar que aproximadamente en 200s los pesos convergen a un valor constante, lo que significa que la NAD está entrenada. Por último en la figura 4.3.4 se observa el error de entrenamiento, donde se demuestra que converge a cero en el mismo tiempo.

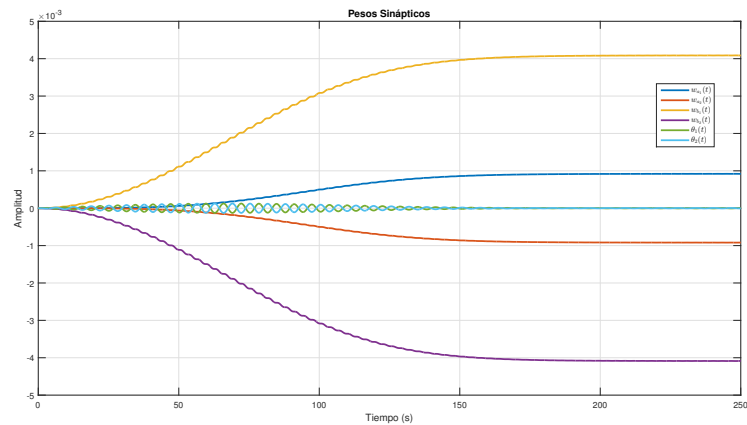


Figura 4.3.3: Convergencia de los pesos sinápticos 3

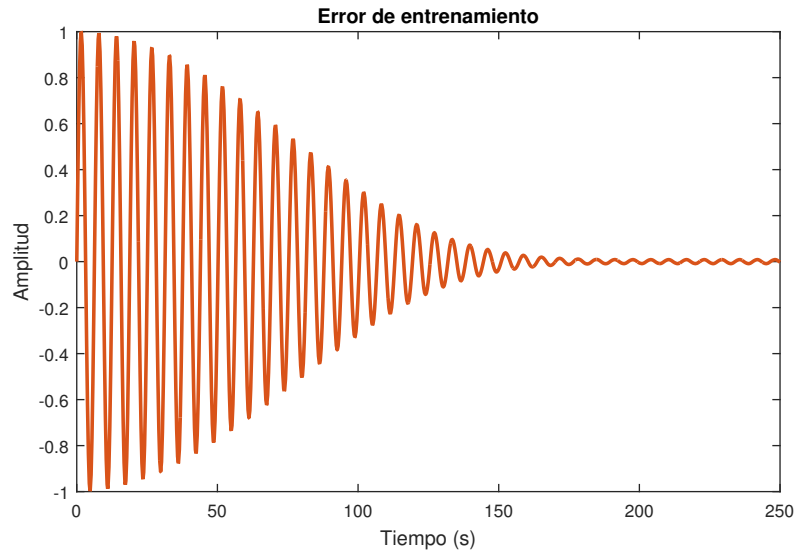


Figura 4.3.4: Error de entrenamiento 3

Los parámetros de la neurona artificial dinámica que lograron el ajuste de la trayectoria deseada son los siguientes:

$$w_a^* = \begin{bmatrix} w_{a_1}^* \\ w_{a_2}^* \end{bmatrix} = \begin{bmatrix} 9.192341377903742 \times 10^{-4} \\ -9.192341377903742 \times 10^{-4} \end{bmatrix}$$

$$w_b^* = \begin{bmatrix} w_{b_1}^* \\ w_{b_2}^* \end{bmatrix} = \begin{bmatrix} 0.004086028310422 \\ -0.004086028310422 \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} 6.299211058579984 \times 10^{-6} \\ -6.299211058579984 \times 10^{-6} \end{bmatrix}$$

En la figura 4.3.5 se presenta la entrada 2 la cual se utilizó como señal de validación, a su vez, en la figura 4.3.6 se observa la entrada 3 que también es usada como señal de validación, todo esto con el fin de evaluar el desempeño de la neurona artificial dinámica entrenada ante diferentes señales de excitación.

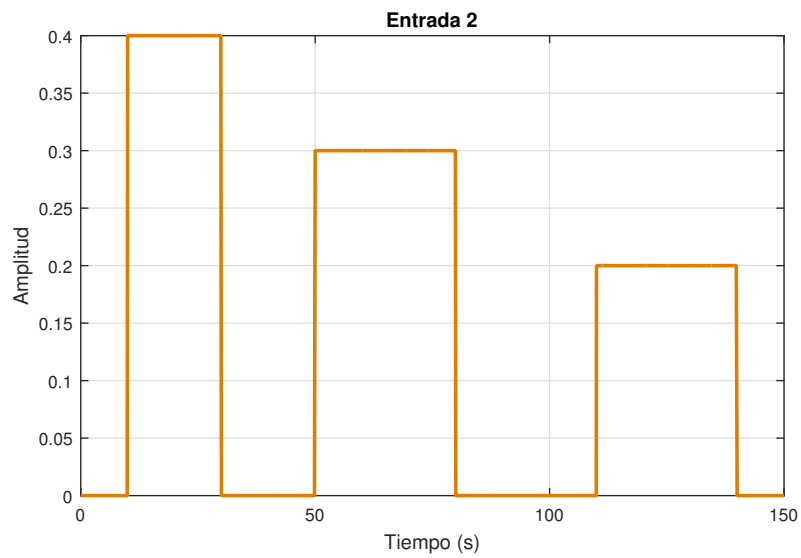


Figura 4.3.5: Tren de pulsos cuadrado para validar

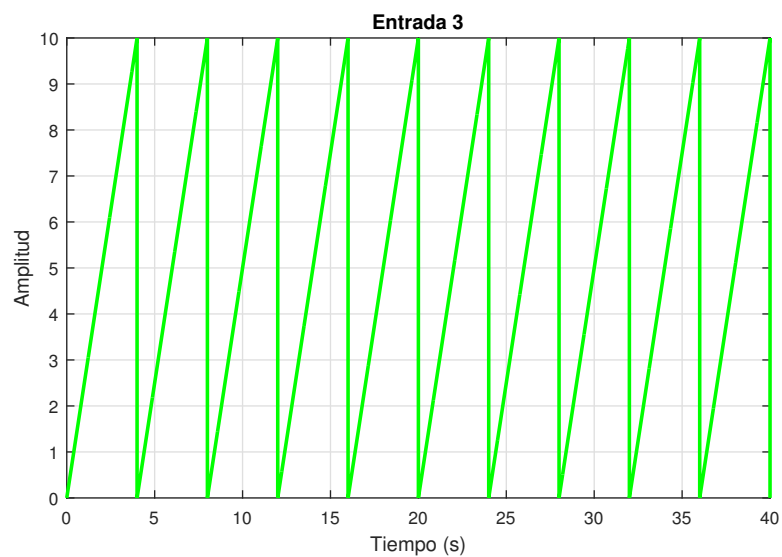


Figura 4.3.6: Cresta de sierra para validar

Como se puede apreciar en la figura 4.3.7 se realizó una comparación en lazo abierto de la neurona dinámica contra el circuito RC, esto con la señal de entrada 1 (4.3.1) utilizada para el entrenamiento, con lo que se observa el ajuste preciso de la neurona artificial dinámica en el perfil negro, a la trayectoria del circuito RC en el perfil magenta.

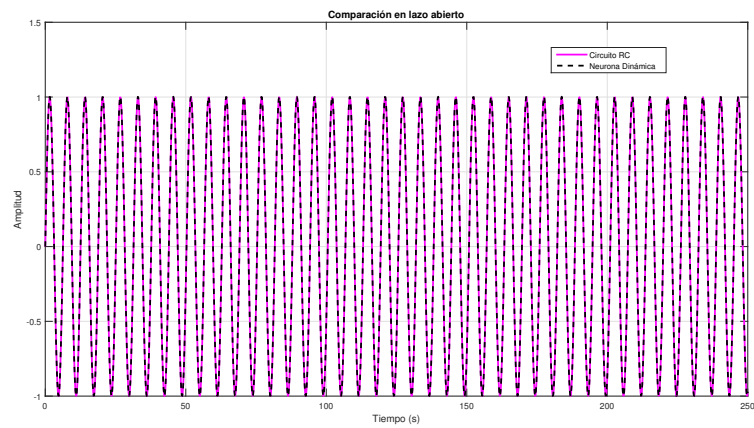


Figura 4.3.7: Respuesta a la señal senoidal

En la figura 4.3.8 se realiza la segunda prueba en lazo abierto, donde se utilizó la señal de entrada 2 (4.3.5) de validación, que es un tren de pulsos cuadrado con distinta amplitud, con lo que se reafirma el ajuste de la neurona artificial dinámica al tren de pulsos cuadrado.

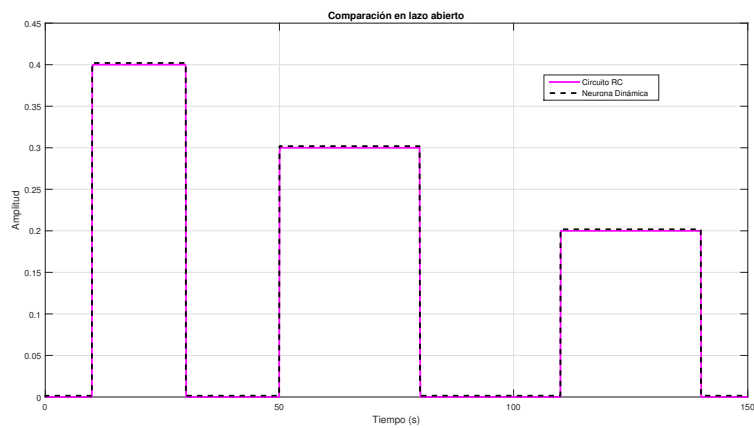


Figura 4.3.8: Respuesta al tren de pulsos cuadrado

Por último en la figura 4.3.9 se visualiza la última prueba en lazo abierto, donde la neurona artificial dinámica ajusta la trayectoria del circuito RC regido por la señal de entrada 3 (4.3.6).

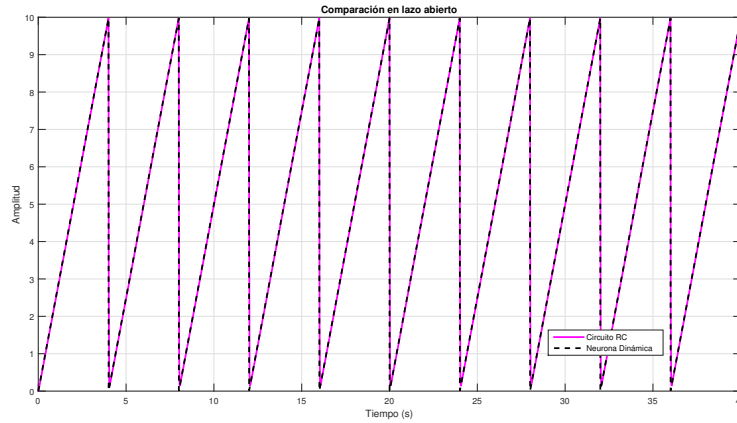


Figura 4.3.9: Respuesta a la cresta de sierra

#### 4.3.4. Conclusión de la simulación

Se realizó un análisis de la convergencia del error, el cual resulta siendo estable y asintótico, con lo que se demuestra que la neurona artificial dinámica puede aproximar un circuito RC. En el sistema (4.6) se presenta el modelo matemático de un circuito RC como el que se muestra en la figura 4.3.10.

$$\sum_{\phi_{e3}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}100 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 9.19 \times 10^{-4} \\ -9.19 \times 10^{-4} \end{bmatrix} \cdot x(t) + \begin{bmatrix} 0.004 \\ -0.004 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 6.29 \times 10^{-6} \\ -6.29 \times 10^{-6} \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.6)$$

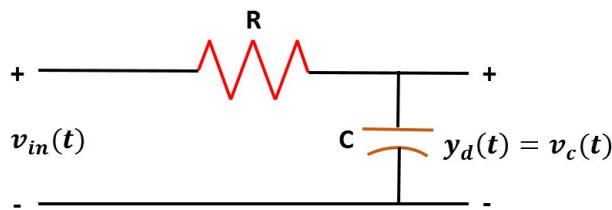


Figura 4.3.10: Circuito RC

Por lo tanto se puede concluir que  $y(t) \cong y_d(t)$  para una entrada  $u(t)$  como la que se muestra en 4.3.1, 4.3.5 y 4.3.6.

### 4.4. Simulación 4: Aproximación de un sistema dinámico no lineal

#### 4.4.1. Objetivo de la simulación

Aproximar un sistema dinámico no lineal con la neurona artificial dinámica propuesta para este trabajo de tesis y analizar la convergencia de la ley de aprendizaje presentada en (3.13).



#### 4.4.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.13) para aproximar el sistema dinámico no lineal de la ecuación (4.7)

$$\sum_{SNL} : \begin{cases} \dot{w}(t) = -k \cdot w^2(t) + ku(t) \\ y_d(t) = w(t) \end{cases} . \quad (4.7)$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001, \\ \mu_{w_a} &= \text{diag}_2\{1 \times 10^{-8}\}, \\ \mu_{w_b} &= \text{diag}_2\{2 \times 10^{-8}\}, \\ \mu_{\theta} &= \text{diag}_2\{3 \times 10^{-8}\}, \\ \alpha &= 10, \\ k &= \frac{1}{0.1}. \end{aligned}$$

En la figura 3.2.1 se presenta el esquema general utilizado para la simulación con la diferencia de la sustitución del bloque "Sistema" por la ecuación (4.7). La simulación se llevó a cabo mediante el software Simulink de MatLab.

#### 4.4.3. Desarrollo de la simulación

Las condiciones iniciales para la simulación son cero, posteriormente en la figura 4.4.1 se presenta la onda cuadrada construida para  $u(t)$ .

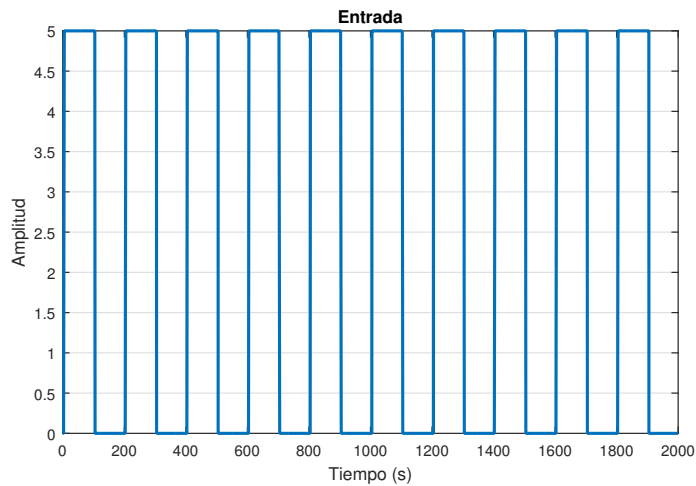


Figura 4.4.1: Tren de pulsos cuadrado

En la figura 4.4.2 se presenta la comparación de la neurona artificial dinámica en el perfil negro, y el sistema dinámico no lineal en el perfil magenta durante la etapa de entrenamiento, con lo que se observa que en 600 segundos aproximadamente se aproxima a la salida deseada.

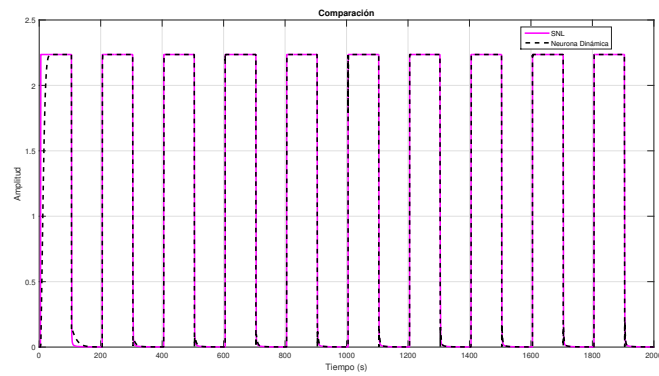


Figura 4.4.2: Aprendizaje de la neurona dinámica 4

En la figura 4.4.3 se muestra la convergencia de los pesos sinápticos y su evolución en el tiempo hasta converger a un valor constante, lo que significa que la neurona artificial dinámica se ha entrenado.

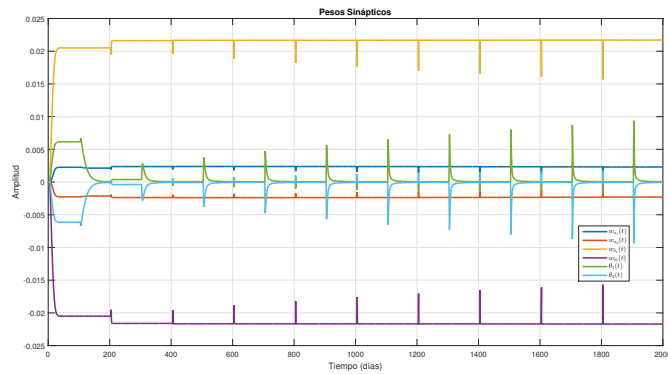


Figura 4.4.3: Convergencia de los pesos sinápticos 4

En la figura 4.4.4 se puede notar el error de aproximación, el cual es prácticamente cero a partir del segundo 400.

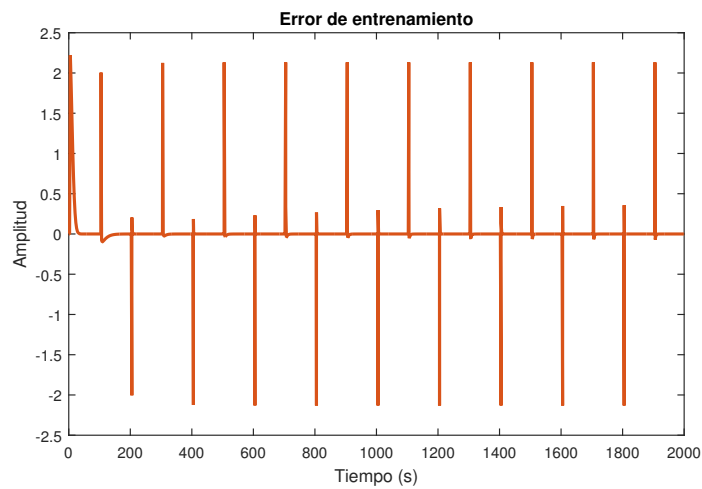


Figura 4.4.4: Error de entrenamiento 4

Los pesos sinápticos entrenados de la neurona artificial dinámica que lograron el ajuste de la trayectoria deseada son los siguientes:

$$w_a^* = \begin{bmatrix} w_{a1}^* \\ w_{a2}^* \end{bmatrix} = \begin{bmatrix} 0.002298042908200 \\ -0.002298042908200 \end{bmatrix}$$

$$w_b^* = \begin{bmatrix} w_{b1}^* \\ w_{b2}^* \end{bmatrix} = \begin{bmatrix} 0.021706467074134 \\ -0.021706467074134 \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} 5.062248931664592 \times 10^{-5} \\ -5.062248931664592 \times 10^{-5} \end{bmatrix}$$

Como se puede apreciar en la figura 4.4.5 se realizó una comparación en lazo abierto de la neurona dinámica entrenada en el perfil negro, contra el sistema dinámico no lineal en el perfil magenta, esto con la señal de entrada (4.4.1) utilizada para el entrenamiento, con lo que se observa el ajuste preciso de la neurona artificial dinámica a la trayectoria del sistema dinámico no lineal.

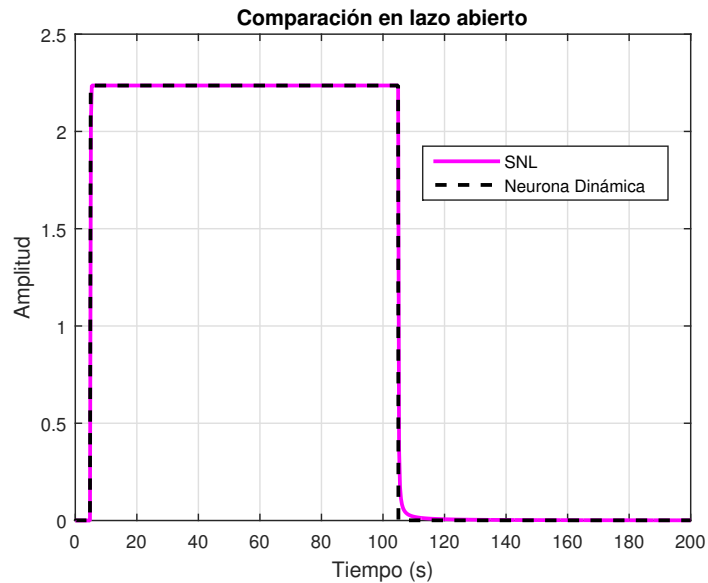


Figura 4.4.5: Neurona entrenada

#### 4.4.4. Conclusión de la simulación

Se realizó un análisis de la convergencia del error, donde se verifica su estabilidad tal como se demostró en el teorema 3.1, lo que significa que la neurona artificial dinámica puede aproximar un sistema dinámico no lineal afín con la entrada (4.7). En el sistema (4.8) se presenta el modelo matemático con la señal de entrada que se muestra en la figura 4.4.1, que representa la dinámica de un sistema no lineal afín con la entrada como el que se muestra en la figura 4.4.6.

$$\sum_{\phi_{\epsilon_4}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}10 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 0.0022 \\ -0.0022 \end{bmatrix} \cdot x(t) + \begin{bmatrix} 0.0217 \\ -0.0217 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 5.06 \times 10^{-5} \\ -5.06 \times 10^{-5} \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.8)$$

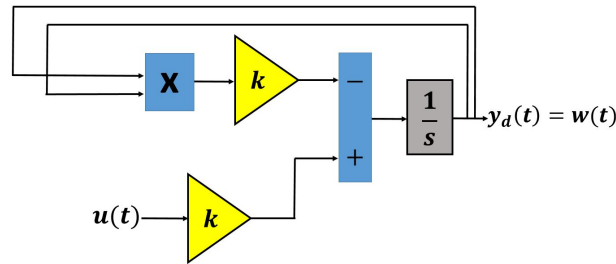


Figura 4.4.6: Sistema no lineal

Por lo tanto se puede concluir que  $y(t) \cong y_d(t)$  para una entrada  $u(t)$  como la que se muestra en (4.4.1).

#### 4.5. Simulación 5: Aproximación de un sistema estático no lineal con múltiples entradas

##### 4.5.1. Objetivo de la simulación

Aproximar una compuerta AND con la neurona artificial dinámica propuesta para este trabajo de tesis y analizar la convergencia de la ley de aprendizaje presentada en (3.26).

##### 4.5.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.26) para aproximar la compuerta lógica que se muestra en la tabla 4.5.1.

$u_1$	$u_2$	$y_d$
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 4.5.1: Compuerta AND

Los valores utilizados para los parámetros en la simulación son los siguientes:  
 $\tau_p = 0.001$ ,  
 $\mu_{w_a} = \mu_{w_b} = \mu_{\theta} = \text{diag}_2\{1 \times 10^{-11}\}$ ,  
 $\alpha = 1000$ .

En la figura 3.2.1 se presenta el esquema general utilizado para la simulación con la diferencia de la sustitución del bloque "Sistema" por la compuerta lógica de la tabla 4.5.1. La simulación se llevó a cabo mediante el software Simulink de MatLab.

### 4.5.3. Desarrollo de la simulación

Las condiciones iniciales para la simulación son cero, posteriormente en la figura 4.5.1 y 4.5.2 se muestran las señales de entrenamiento 1 y 2 para entrenar la neurona artificial dinámica.

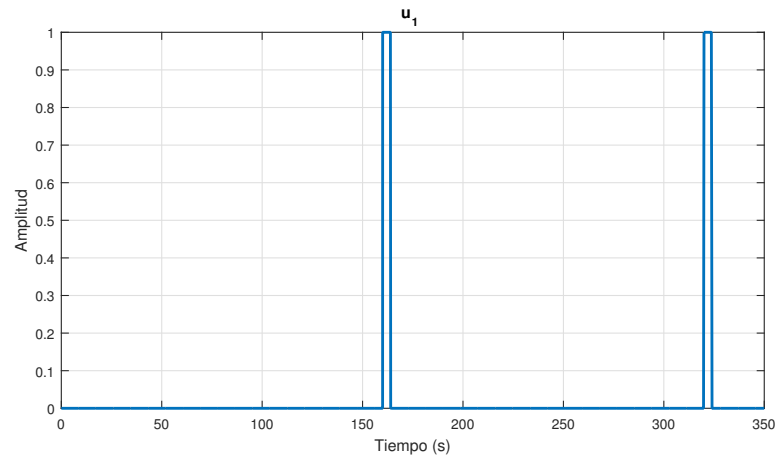


Figura 4.5.1: Señal de entrada 1

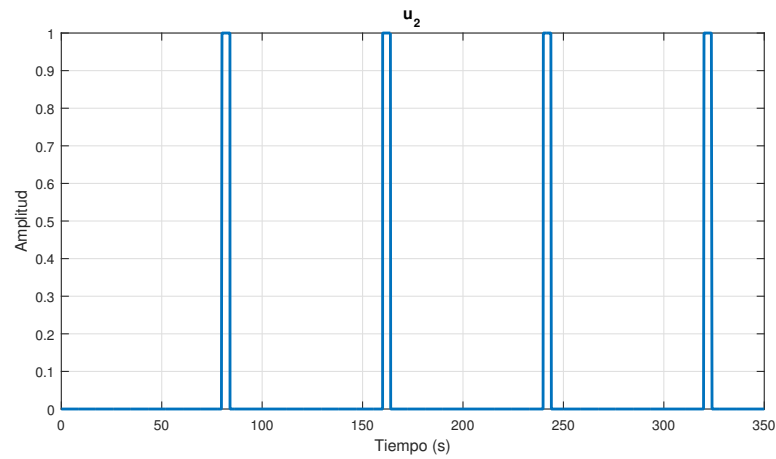


Figura 4.5.2: Señal de entrada 2

En la figura 4.5.3 se observa el entrenamiento de la neurona artificial dinámica aproximando la compuerta AND en un tiempo de 15000 segundos.

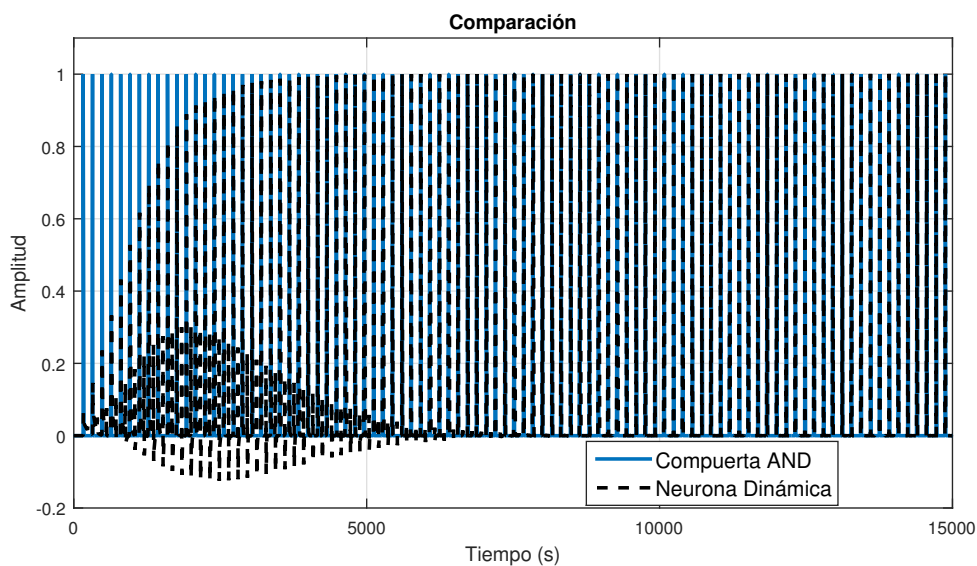


Figura 4.5.3: Aprendizaje de la neurona dinámica 5

Por consiguiente en la figura 4.5.4 se presenta su error de entrenamiento, el cual se aproxima a cero.

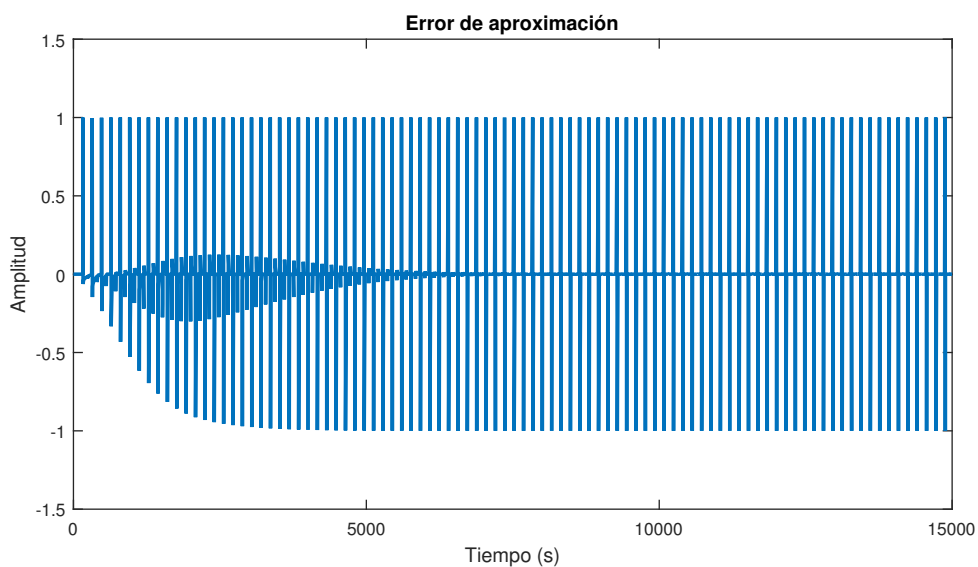


Figura 4.5.4: Error de entrenamiento 5

A su vez, en la figura 4.5.5 se muestra el ajuste de los parámetros de la neurona artificial dinámica para el vector de pesos sinápticos  $w_a$ , donde se verifica que llegan a un valor constante en 10000 segundos aproximadamente.

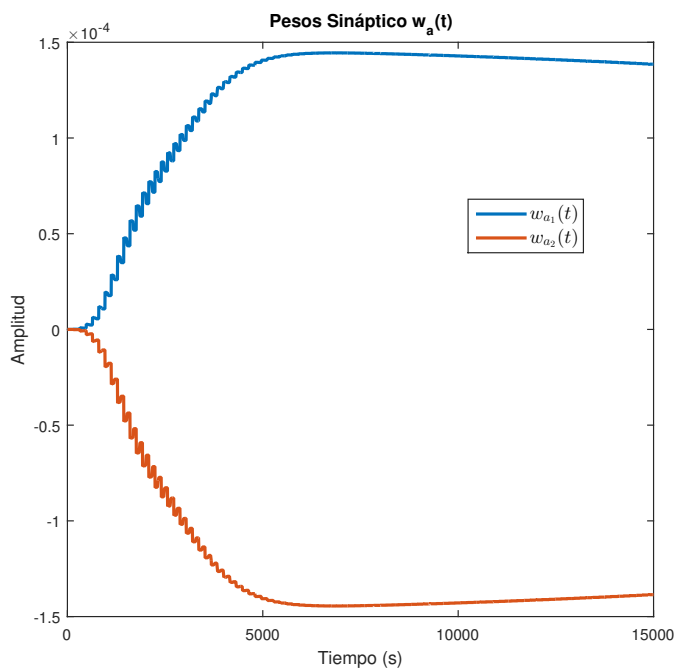


Figura 4.5.5: Entrenamiento de  $w_a$

En la figura 4.5.6 se visualiza la matriz de pesos sinápticos  $W_b$ , ya que se trata de un sistema con múltiples entradas donde  $m = 2$ , por consiguiente la matriz  $W_b$  es de dimensiones  $2 \times 2$ .

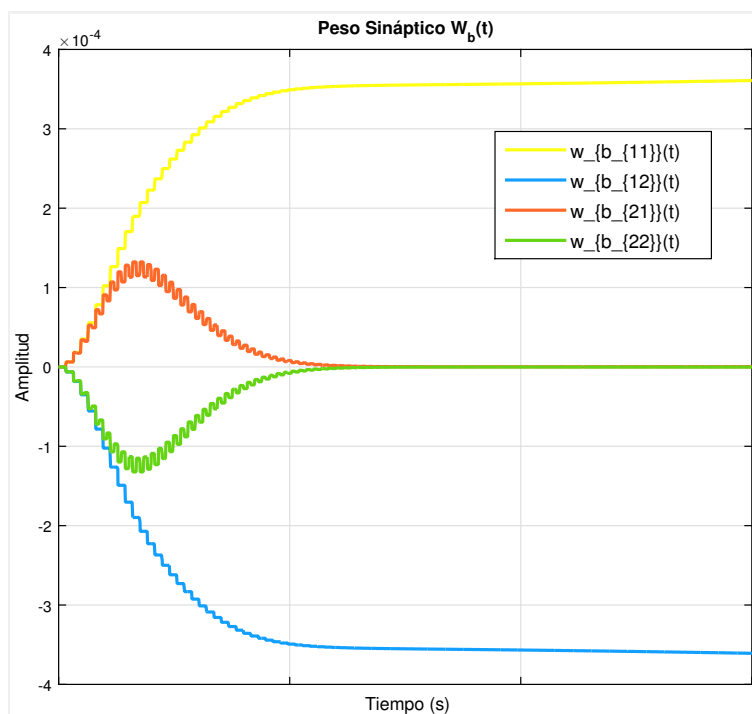


Figura 4.5.6: Entrenamiento de  $w_b$

En la figura 4.5.7 se observa el comportamiento del vector de umbrales  $\theta$ , los cuales dejan de moverse en 10000 segundos aproximadamente.

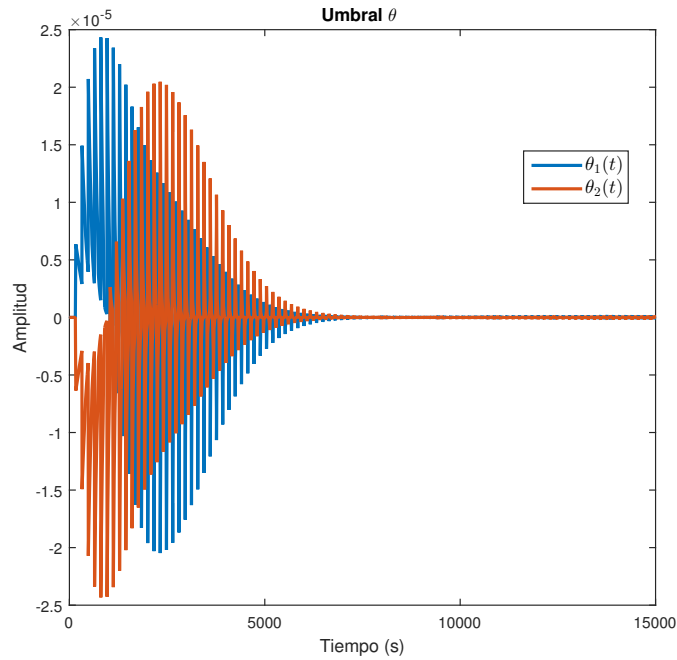


Figura 4.5.7: Entrenamiento de  $\theta$

Los parámetros de la neurona artificial dinámica que lograron el ajuste de la trayectoria deseada, dicho de otra forma, los valores constantes a los que equivalen los pesos sinápticos entrenados son los siguientes:

$$w_a^* = \begin{bmatrix} w_{a_1}^* \\ w_{a_2}^* \end{bmatrix} = \begin{bmatrix} 1.385620714398153 \times 10^{-4} \\ -1.385620714398153 \times 10^{-4} \end{bmatrix}$$

$$W_b^* = \begin{bmatrix} w_{b_{11}}^* & w_{b_{12}}^* \\ w_{b_{21}}^* & w_{b_{22}}^* \end{bmatrix} = \begin{bmatrix} 3.607937174417795 \times 10^{-4} & 1.684856444443993 \times 10^{-7} \\ -3.607937174417795 \times 10^{-4} & -1.684856444443993 \times 10^{-7} \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} -5.122709863361608 \times 10^{-14} \\ 5.122709863361608 \times 10^{-14} \end{bmatrix}$$

Como se puede apreciar en la figura 4.5.8 se realizó una comparación en lazo abierto de la neurona artificial dinámica entrenada contra la compuerta AND, esto con la señal de entrada 1 (4.5.1) y la señal de entrada 2 (4.5.2) utilizadas para el entrenamiento, con lo que se observa el ajuste preciso de la neurona dinámica a la trayectoria de la compuerta AND.



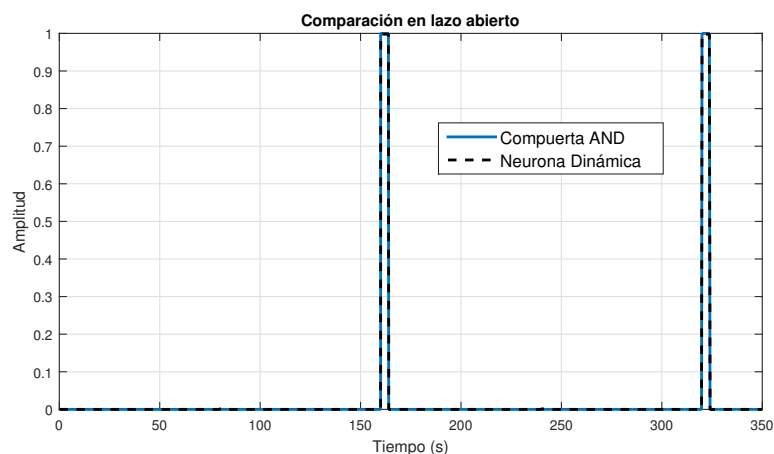


Figura 4.5.8: ND vs Compuerta AND

En la figura 4.5.11 se realiza una segunda prueba en lazo abierto, donde se utilizó la señal de entrada 1 (4.5.9) y la señal de entrada 2 (4.5.10) de validación, a pesar de ser señales distintas a las de entrenamiento, la neurona artificial dinámica entrenada realiza la compuerta lógica AND sin problemas.

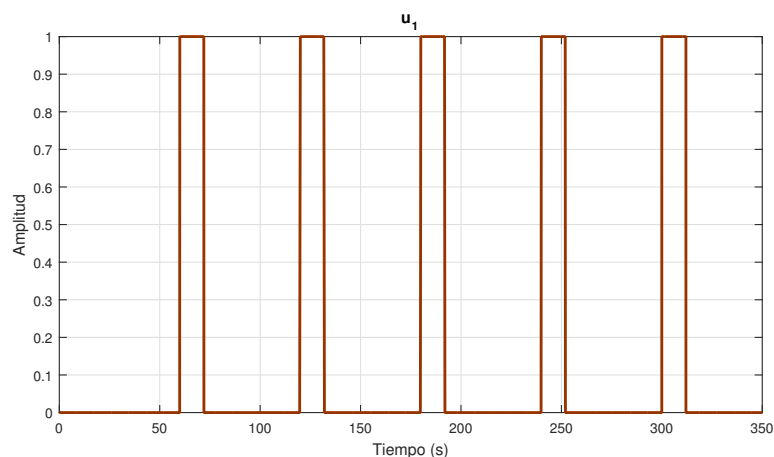


Figura 4.5.9: Señal de entrada para validar 1

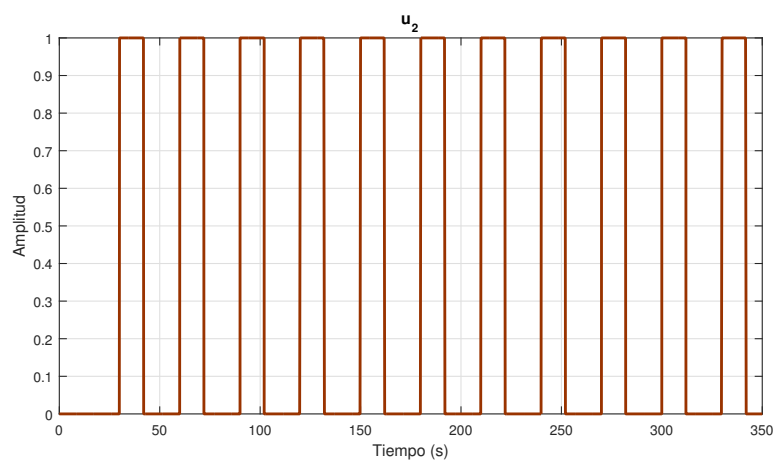


Figura 4.5.10: Señal de entrada para validar 2

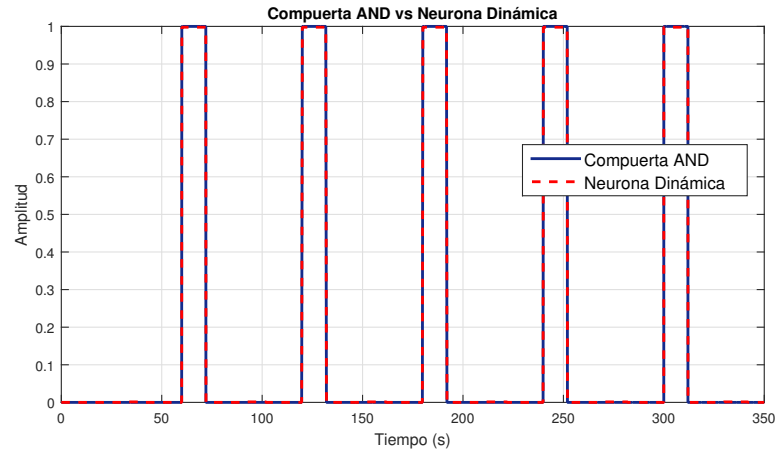


Figura 4.5.11: ND vs Compuerta AND con señales de validación

#### 4.5.4. Conclusión de la simulación

Se realizó un análisis de la convergencia del error, que resulta ser estable y acotado, como lo enuncia el teorema 3.1. En el sistema (4.9) se presenta el modelo matemático de una compuerta AND (Figura 4.5.12).

$$\sum_{\phi_{e5}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}1000 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 1.38 \times 10^{-4} \\ -1.38 \times 10^{-4} \end{bmatrix} \cdot x(t) + \begin{bmatrix} 3.60 \times 10^{-4} & 1.68 \times 10^{-7} \\ -3.60 \times 10^{-4} & -1.68 \times 10^{-7} \end{bmatrix} \cdot \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \begin{bmatrix} -5.12 \times 10^{-14} \\ 5.12 \times 10^{-14} \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.9)$$

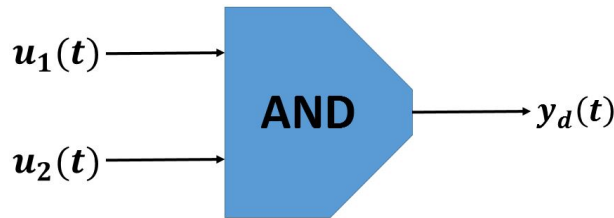


Figura 4.5.12: Compuerta AND

Por lo tanto se puede concluir que  $y(t) \cong y_d(t)$  para una entrada  $u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$  como la que se muestra en 4.5.1, 4.5.2, 4.5.9 y 4.5.10.

### 4.6. Simulación 6: Aproximación de la dinámica de un biorreactor híbrido lecho fijo-fluidizante inverso con datos experimentales

#### 4.6.1. Objetivo de la simulación

Realizar la aproximación con datos experimentales de un sistema no lineal con la neurona artificial dinámica mostrada en (3.1).

#### 4.6.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ . Los sistemas que se utilizaron para la simulación son los expresados en (3.1) y (3.13) para

aproximar el sistema dinámico correspondiente a los datos experimentales del biorreactor que se encuentra en el Instituto Tecnológico de Orizaba. Se busca aproximar la dinámica de la demanda química de oxígeno soluble (DQOs) a la salida, el cual esta expresado en gramos sobre litros (g/L). Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\tau_p = 0.001,$$

$$\mu_{w_a} = \mu_{w_b} = \mu_\theta = \text{diag}_2\{1 \times 10^{-7}\},$$

$$\alpha = 100.$$

### 4.6.3. Desarrollo de la simulación

La entrada del sistema es la carga volumétrica aplicada (CVA) y sus unidades son gramos de demanda química de oxígeno sobre litros por día (gDQO/L\*d) y se presenta en la figura 4.6.1. Las condiciones iniciales son cero, posteriormente se realizó el entrenamiento de la neurona para aproximar la demanda química de oxígeno soluble (DQOs) a la salida.

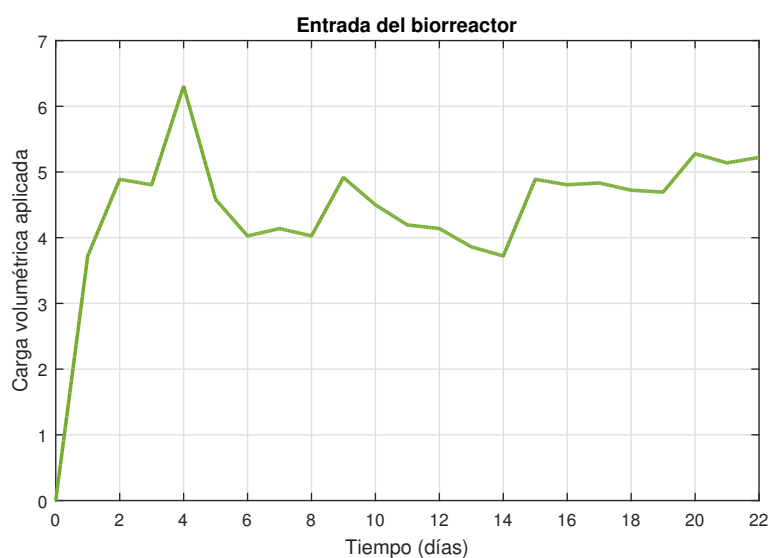


Figura 4.6.1: Entrada del sistema

En la figura 4.6.2, el perfil magenta corresponde a la demanda química de oxígeno soluble (DQOs) a la salida de los datos experimentales obtenidos de la planta, el cual es la salida deseada del sistema y el perfil negro es la neurona artificial dinámica durante su etapa de entrenamiento.

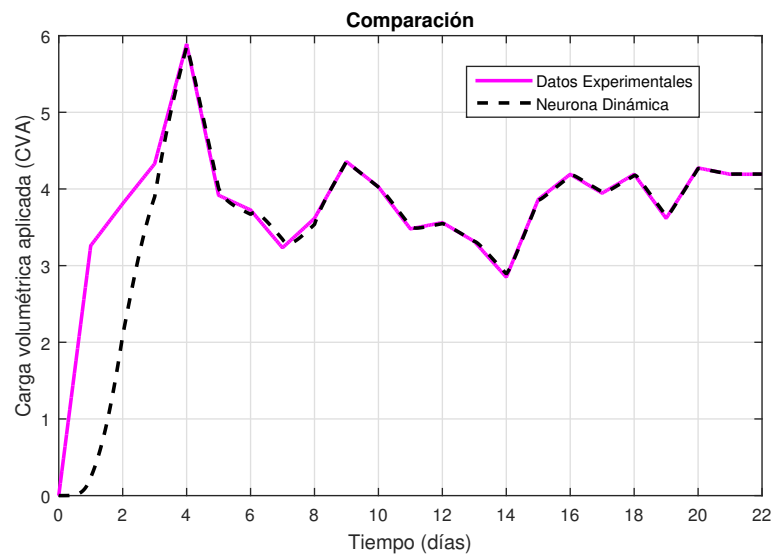


Figura 4.6.2: Aprendizaje de la neurona dinámica 6

En la figura 4.6.3 se muestra el entrenamiento y ajuste de los parámetros de la neurona artificial dinámica, los cuales tienden a un valor constante que son los pesos sinápticos entrenados.

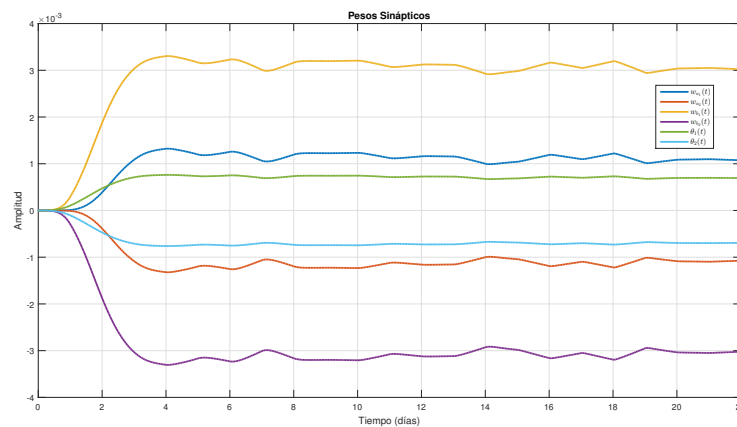


Figura 4.6.3: Convergencia de los pesos sinápticos 5

Por último en la figura 4.6.4 se observa el error de aproximación, donde se demuestra que converge a cero en 4 días aproximadamente.

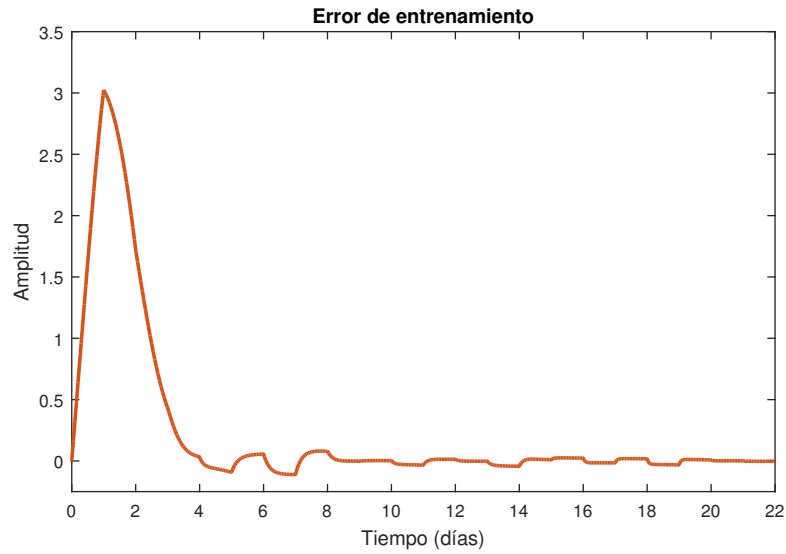


Figura 4.6.4: Error de entrenamiento 6

Los parámetros entrenados de la neurona artificial dinámica que lograron el ajuste de la trayectoria deseada son los siguientes:

$$w_a^* = \begin{bmatrix} w_{a_1}^* \\ w_{a_2}^* \end{bmatrix} = \begin{bmatrix} 0.001074598613058 \\ -0.001074598613058 \end{bmatrix}$$

$$w_b^* = \begin{bmatrix} w_{b_1}^* \\ w_{b_2}^* \end{bmatrix} = \begin{bmatrix} 0.003022366019101 \\ -0.00302236601910071 \end{bmatrix}$$

$$\theta^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \end{bmatrix} = \begin{bmatrix} 6.931119455254878 \times 10^{-4} \\ -6.931119455254878 \times 10^{-4} \end{bmatrix}$$

En la figura 4.6.5 se presenta la comparación en lazo abierto de la neurona artificial dinámica entrenada, contra los datos experimentales de la planta.

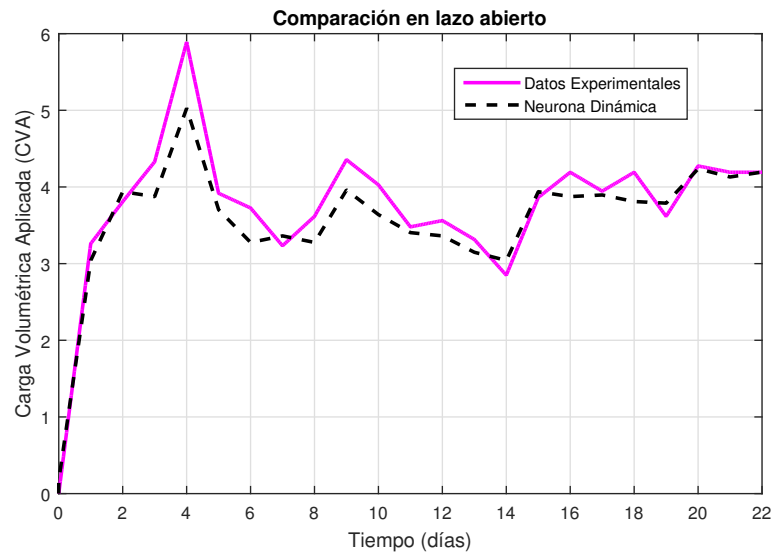


Figura 4.6.5: Neurona dinámica entrenada

#### 4.6.4. Conclusión de la simulación

Con la NAD es posible aproximar sistemas no lineales, ya que la ley de aprendizaje se encarga de ajustar los parámetros de la neurona artificial dinámica conforme transcurre el tiempo.

En la ecuación 4.10 se muestra el modelo matemático de los datos experimentales que corresponden a la dinámica del biorreactor, con lo que se hace a  $y(t) \cong y_d(t)$ , donde la salida deseada es la demanda química de oxígeno soluble.

$$\sum_{\phi_{e6}} : \begin{cases} \dot{x}(t) = -\frac{1}{0.001}x(t) + \frac{1}{0.001}100 \cdot \tanh(v_1(t) - v_2(t)) \\ \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 0.0010 \\ -0.0010 \end{bmatrix} \cdot x(t) + \begin{bmatrix} 0.0030 \\ -0.0030 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 6.93 \times 10^{-04} \\ -6.93 \times 10^{-04} \end{bmatrix} \\ y(t) = x(t) \end{cases} \quad (4.10)$$

# Capítulo 5

## Neurona Artificial Dinámica vs Red Neuronal Artificial

En este capítulo se realizó una breve comparación de la neurona artificial dinámica propuesta en este trabajo de investigación con una arquitectura de red neuronal artificial. Esto con el fin de verificar el desempeño de la metodología propuesta en esta tesis, contra las metodologías de redes neuronales existentes en la literatura.

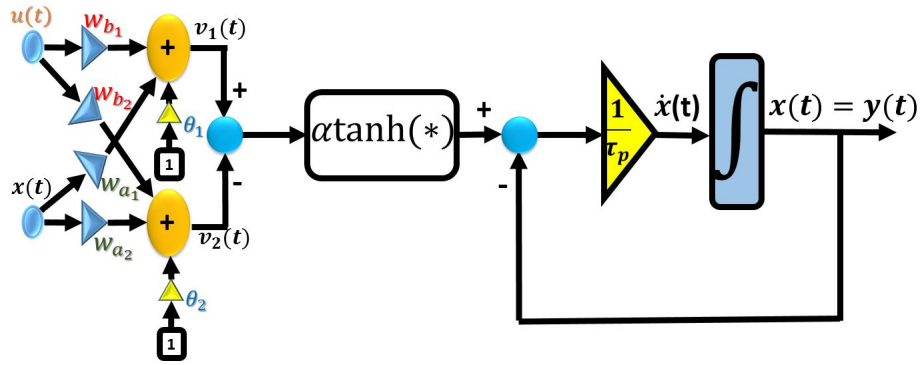


Figura 5.0.1: Neurona artificial dinámica

En la figura 5.0.1 se presenta la estructura neuronal dinámica de este trabajo de tesis, y el modelo matemático que la representa (5.1).

$$\sum_{\phi} : \begin{cases} \dot{x}(t) = -\frac{1}{\tau_p} x(t) + \frac{1}{\tau_p} \alpha \tanh(v_1(t) - v_2(t)) \\ v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = w_a(t) \cdot x(t) + w_b(t) \cdot u(t) + \theta(t) \\ y(t) = x(t) \end{cases} \quad (5.1)$$

La ley de aprendizaje dinámica de la NAD se presenta en  $\sum_{\Psi}$  (5.2).

$$\sum_{\Psi} : \begin{cases} \dot{w}_a(t) := \mu_{w_a} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{x(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \\ \dot{w}_b(t) := \mu_{w_b} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{u(t) \cdot z(t) \cdot e_y(t)}{\tau_p} \\ \dot{\theta}(t) := \mu_{\theta} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot \frac{z(t) \cdot e_y(t)}{\tau_p} \\ \dot{z}(t) := \text{sech}^2(v_1(t) - v_2(t)) \end{cases}, \quad (5.2)$$

La cual se comparó con una arquitectura estática de red neuronal artificial 2-1 de una entrada, como la mostrada en la figura 5.0.2.

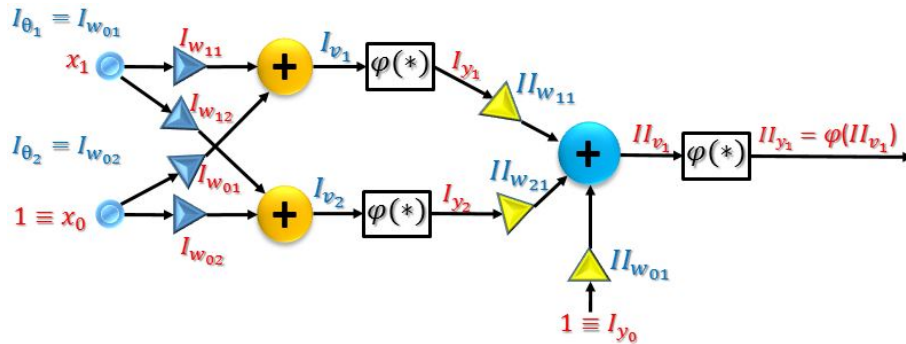


Figura 5.0.2: RNA 2-1 de una entrada

El modelo matemático de la RNA se muestra en la ecuación 5.3

$$II_{y_1}(t) = \varphi(II_{w_{11}}(t) - \varphi(I_{w_{11}}(t) \cdot x_1(t) + I_{w_{01}}(t)) + II_{w_{21}}(t) \cdot \varphi(I_{w_{12}}(t) \cdot x_1(t) + I_{w_{02}}(t)) + II_{w_{01}}(t)) \quad (5.3)$$

y su ley de aprendizaje expresada en 5.4.

$$\sum_{AA} : \begin{cases} I \dot{w}_{ij}(t) = -I \mu_{ij} \cdot \frac{\partial J}{\partial I w_{ij}(t)}, i = 1, 0; j = 1, 2. \\ II \dot{w}_{i1}(t) = -II \mu_{i1} \cdot \frac{\partial J}{\partial II w_{i1}(t)}, i = 0, 1, 2. \end{cases} \quad (5.4)$$

## 5.1. Simulación 7. RNA y NAD aproximando una función estática lineal

### 5.1.1. Objetivo de la simulación

Comparar la RNA (5.3) contra la NAD (5.1) propuesta en este trabajo de tesis en la aproximación de una función estática lineal (5.5).

### 5.1.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración \$T\_e = 0.001\$.

Los sistemas que se utilizaron para la simulación son los expresados en (5.1) y (5.3), con sus respectivas



leyes de aprendizaje, para aproximar la función estática lineal presentada en 5.5

$$y_d(t) = -0.3u(t) \tag{5.5}$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001 \\ \mu_{w_a} = \mu_{w_b} = \mu_{\theta} &= \text{diag}_2\{1 \times 10^{-7}\} \\ \alpha &= 100 \\ \varphi &= \tanh \\ {}^{II}\mu_{i1} &= \text{diag}_3\{0.01\} \\ {}^I\mu_{i1} &= \text{diag}_2\{0.02\} \\ {}^I\mu_{i2} &= \text{diag}_2\{0.03\} \end{aligned}$$

### 5.1.3. Desarrollo de la simulación

En la figura 5.1.1 se muestra la señal de excitación para entrenar la NAD y la RNA, la cual corresponde a una señal senoidal con amplitud de 1.

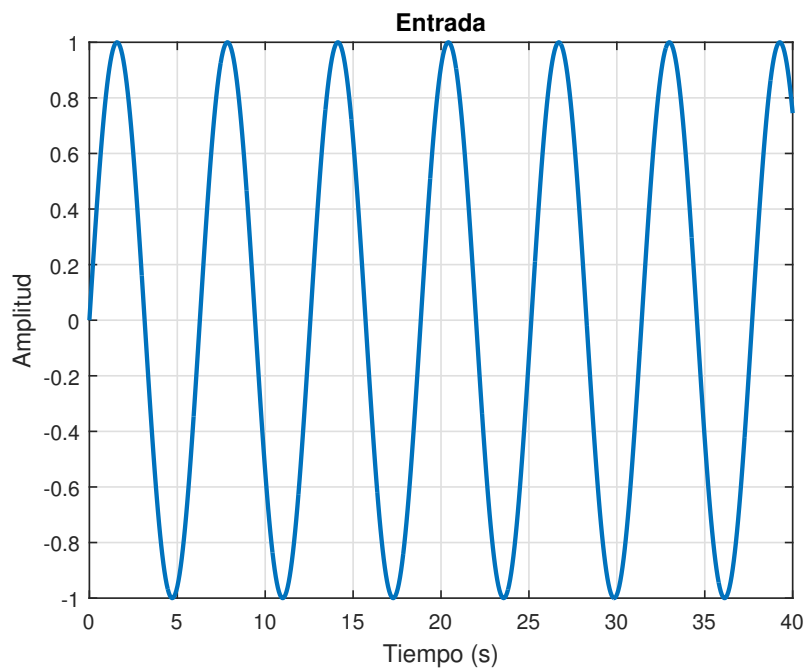


Figura 5.1.1: Señal de excitación

En la figura 5.1.2 se muestra la comparación de la RNA y la NAD aproximando la función estática lineal (5.5), en la cual se verifica que la RNA al ser estática su salida es instantánea, por lo que aproxima en menor tiempo la función estática, por otro lado, la NAD se tarda aproximadamente 20 segundos pero logra aproximar la función estática sin problemas.

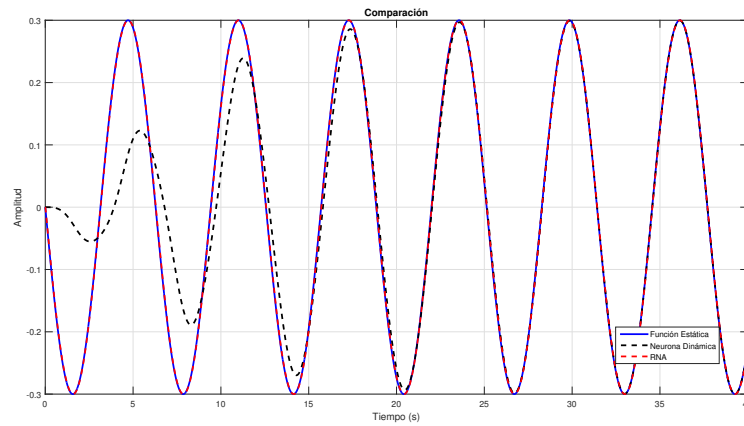


Figura 5.1.2: comparación

En la figura 5.1.3 se muestra el error de entrenamiento, en el que la RNA tiene un error muy cercano a cero, y el error de la NAD oscila los primeros 20 segundos pero al final su convergencia es asintótica.

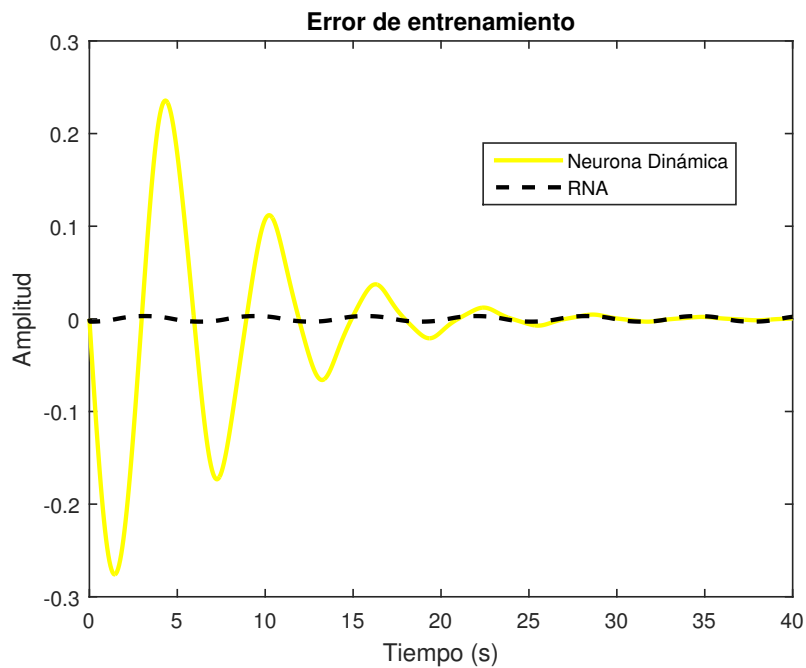


Figura 5.1.3: Error de entrenamiento 7

#### 5.1.4. Conclusión de la simulación

Se puede concluir que una RNA 2-1 de una entrada aproxima en menor tiempo una función estática lineal, que la NAD propuesta en este trabajo de tesis, sin embargo como resultado final se obtiene un error de entrenamiento asintótico tanto de la RNA como de la NAD.

## 5.2. Simulación 8. RNA y NAD aproximando una función dinámica (Circuito RC)

### 5.2.1. Objetivo de la simulación

Comparar la RNA (5.3) contra la NAD (5.1) propuesta en este trabajo de tesis en la aproximación de un circuito RC (5.6).

### 5.2.2. Configuración de la simulación

En la siguiente simulación se usó el método de integración de Euler, con un paso de integración  $T_e = 0.001$ .

Los sistemas que se utilizaron para la simulación son los expresados en (5.1), su ley de aprendizaje (5.2), y (5.3), con su ley de aprendizaje (5.4), para aproximar el circuito RC presentado en 5.6

$$\sum_{LTI} : \begin{cases} \dot{v}_c(t) = -\frac{1}{RC}v_c(t) + \frac{1}{RC}v_{in}(t) \\ y_d(t) = v_c(t) \end{cases} \quad (5.6)$$

Los valores utilizados para los parámetros en la simulación son los siguientes:

$$\begin{aligned} \tau_p &= 0.001 \\ \mu_{w_a} &= \text{diag}_2\{1 \times 10^{-9}\} \\ \mu_{w_b} &= \text{diag}_2\{2 \times 10^{-9}\} \\ \mu_{\theta} &= \text{diag}_2\{3 \times 10^{-9}\} \\ \alpha &= 1 \\ \varphi &= \tanh \\ {}^{II}\mu_{i1} &= \text{diag}_3\{0.01\} \\ {}^I\mu_{i1} &= \text{diag}_2\{0.02\} \\ {}^I\mu_{i2} &= \text{diag}_2\{0.03\} \\ RC &= 10 \end{aligned}$$

### 5.2.3. Desarrollo de la simulación

En la figura 5.2.1 se muestra la señal de excitación para entrenar la NAD y la RNA, la cual corresponde a un tren de pulsos cuadrado con amplitud de 0.5.

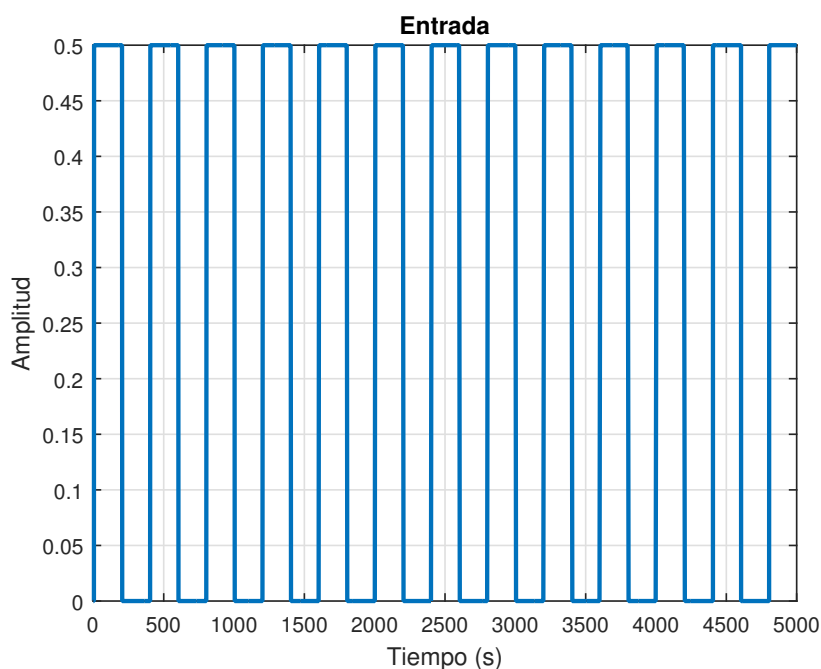


Figura 5.2.1: Señal de excitación 2

En la figura 5.2.2 se muestra la comparación de la RNA y la NAD aproximando el sistema dinámico lineal (5.6), en la cual se verifica que la RNA al ser estática su salida es instantánea, por lo que no logra aproximar el circuito RC, por otro lado, la NAD se tarda aproximadamente 1000 segundos pero logra aproximar el circuito RC.

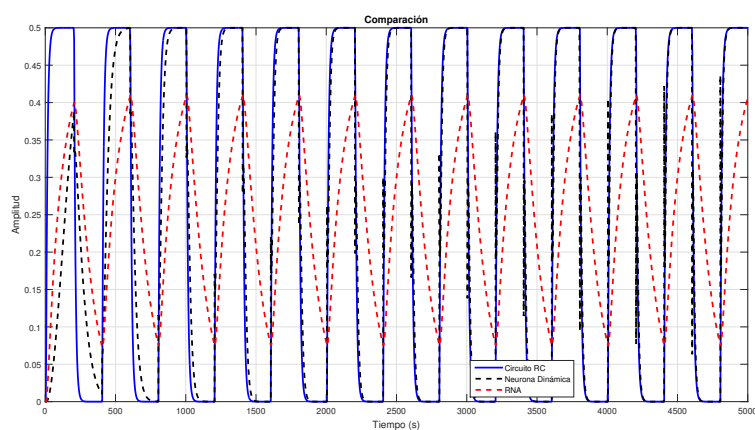


Figura 5.2.2: comparación 2

En la figura 5.2.3 se muestra el error de entrenamiento, en el que la RNA tiene un error de 0.3 aproximadamente, y el error de la NAD oscila los primeros 1000 segundos pero al final su convergencia es estable y cercana a cero.

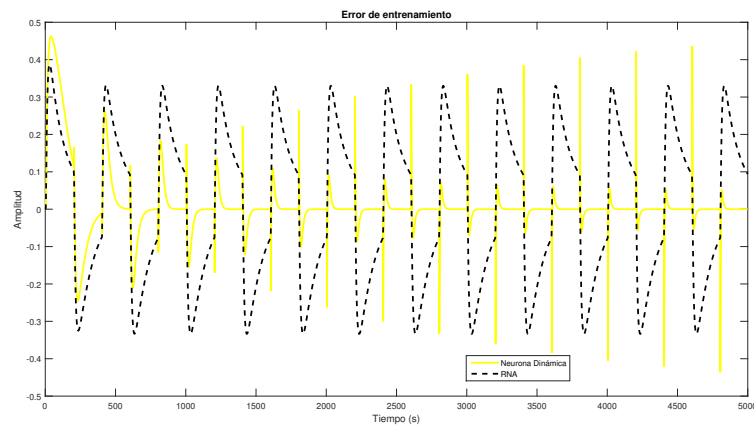


Figura 5.2.3: Error de entrenamiento 8

### 5.2.4. Conclusión de la simulación

Se puede concluir que una RNA 2-1 de una entrada no puede aproximar un sistema dinámico, ya que su respuesta es instantánea, por lo tanto no puede presentar comportamientos exponenciales suaves como el que puede presentar un circuito RC. La solución es cambiar la RNA estática por una RNA dinámica con su respectiva ley de aprendizaje. Dicho esto, la NAD propuesta en este trabajo de investigación logra aproximar el circuito RC, y así se corrobora que no es necesario cambiar la estructura ni la ley de aprendizaje para aproximar un sistema dinámico.

# Capítulo 6

---

## Conclusiones

---

El modelado entrada-salida que se desarrolló en la metodología propuesta, supone una opción factible en la descripción del comportamiento dinámico y estático de sistemas, pues permite considerar el desconocimiento total o parcial del sistema, y el costo computacional requerido para integrar las ecuaciones diferenciales que la componen, es bajo.

La principal diferencia de las metodologías existentes en la literatura es que, se utiliza una neurona artificial dinámica, y una misma ley de aprendizaje para el modelado entrada-salida de sistemas dinámicos lineales y no lineales, sistemas estáticos lineales y no lineales, con la libertad de que puede ser una entrada o múltiples entradas. Por otro lado, las metodologías existentes utilizan redes neuronales artificiales estáticas para la aproximación de sistemas estáticos, con su respectiva ley de aprendizaje, y redes neuronales artificiales dinámicas para la aproximación de sistemas dinámicos, de igual forma con su respectiva ley de aprendizaje.

En el modelo  $\sum_{\phi_{e1}}$  se aproximó una función estática lineal con una entrada, con el que resulta en un error asintóticamente estable, y para el cual se verificó el desempeño del modelo ante diferentes señales de excitación.

En el modelo  $\sum_{\phi_{e2}}$  se aproximó un comparador que es un sistema altamente no lineal, el cual se excita con una señal senoidal y la salida switchea entre un cero y un 0.8, no se obtuvo un error asintótico, pero sí es estable y acotado tal como lo asegura el teorema 3.1.

En el modelo  $\sum_{\phi_{e3}}$  se aproximó un circuito RC, correspondiente a un sistema dinámico LTI. La convergencia del error de aproximación para esta simulación fue asintóticamente estable, y se excitó el modelo entrenado  $\sum_{\phi_{e3}}$  con distintas señales de validación para verificar su funcionamiento.

En el modelo  $\sum_{\phi_{e4}}$  se aproximó un sistema dinámico no lineal afín con la entrada. Para este caso se obtuvo un error estable y acotado.

En el modelo  $\sum_{\phi_{e5}}$  se aproximó un sistema estático con 2 entradas correspondiente a una compuerta lógica AND, donde se asegura que la convergencia del error es estable y acotada. Se verifica que el modelo es válido ante diferentes señales de excitación.

En el modelo  $\sum_{\phi_{e6}}$  se aproximó un sistema dinámico no lineal correspondiente a un biorreactor híbrido lecho fijo-fluidizante inverso, donde la neurona artificial dinámica va aproximando el comportamiento del biorreactor conforme se le van ingresando los datos experimentales, con lo que resulta un modelo entrenado con un error estable y acotado.

No hay una razón específica para la elección de los valores de las matrices de coeficientes de aprendizaje  $\mu$ , solo se debe cumplir la condición de que sean simétricas y positivas definidas, tal como lo enuncia el teorema 3.1. En las simulaciones presentadas se utilizaron valores muy pequeños para dichas matrices, ya que se propuso una constante de tiempo  $\tau_p$  muy pequeña, lo cual genera una respuesta rápida de la neurona artificial dinámica. Dicho de otra forma, para darle el tiempo suficiente a la ley de aprendizaje encontrar el mínimo de la función de energía  $J$  se proponen matrices de coeficientes de aprendizaje con

valores pequeños.

La neurona artificial dinámica entrenada es el modelo matemático de un sistema en específico, y si este se prueba con condiciones iniciales diferentes al sistema que se aproximó, es evidente que tendrá distinto comportamiento, ya que no se trata de un observador que en base a la medición del error estime el estado.

El aprendizaje en las simulaciones presentadas se realizó en línea, es decir, se analizó la amplitud del error para observar su convergencia y evolución a lo largo del tiempo, en lugar de analizar el error medio cuadrático contra épocas de entrenamiento.

Se presentó un póster (Anexo C) respecto a este trabajo de investigación durante la primera jornada de ciencia y tecnología aplicada, organizada por el Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET).

Finalmente, la aproximación de sistemas con distinta naturaleza obtenidos en éste trabajo de investigación, produce errores pequeños, por lo que se puede afirmar que el teorema 3.1 siempre se cumple, es decir, el error de aproximación es estable y además acotado, demostrando así la hipótesis. La programación de dicho teorema, en cualquier software de simulación no es difícil, y el esfuerzo computacional es, desde luego, muy bajo, ya que se trata de una neurona artificial dinámica, en lugar de una red neuronal artificial.

## 6.1. Trabajos Futuros

Realizar una ley de control para la neurona artificial dinámica propuesta en este trabajo de tesis.

Implementar la neurona artificial dinámica con un sistema embebido que adquiera los datos en tiempo real de un sistema físico y realice la aproximación de la dinámica de dicho sistema.

Analizar y estudiar la estructura neuronal dinámica, pero ahora en tiempo discreto y realizar ejemplos de simulación para modelar sistemas en tiempo discreto.

Estudiar el estado interno  $z$  el cual resulta de la regla de la cadena para el cálculo de la ley de aprendizaje, es decir, que pasa si  $\lim_{t \rightarrow \infty} |z(t)| = \infty$ . También qué sucede con la convergencia del error si  $\lim_{t \rightarrow \infty} |z(t)| < \infty$ .

Por último, ampliar la estructura neuronal dinámica a una arquitectura de más de una neurona, dicho de otra manera, una red neuronal artificial dinámica.

---

# Bibliografía

---

- [1] J. R. Reyes. Pasividad robusta de sistemas no lineales vía redes neuronales dinámicas. *Tesis de doctorado, Departamento de Ingeniería Eléctrica CINVESTAV-IPN*, 2002.
- [2] Wen Yu. Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. *Information sciences*, 158:131–147, 2004.
- [3] José de Jesús Rubio and Wen Yu. Nonlinear system identification with recurrent neural networks and dead-zone kalman filter algorithm. *Neurocomputing*, 70(13-15):2460–2466, 2007.
- [4] Hong-Wei Ge, Wen-Li Du, Feng Qian, and Yan-Chun Liang. Identification and control of nonlinear systems by a time-delay recurrent neural network. *Neurocomputing*, 72(13-15):2857–2864, 2009.
- [5] Xuan Han, Wen-Fang Xie, Zhijun Fu, and Weidong Luo. Nonlinear systems identification using dynamic multi-time scale neural networks. *Neurocomputing*, 74(17):3428–3439, 2011.
- [6] José AR Vargas, Kevin HM Gularte, and Elder M Hemerly. An improved on-line neuro-identification scheme. In *Proceedings of 2012 UKACC International Conference on Control*, pages 1088–1093. IEEE, 2012.
- [7] Michel Fliess, Cédric Join, and Hebertt Sira-Ramirez. Non-linear estimation is easy. *arXiv preprint arXiv:0710.4486*, 2007.
- [8] Xin Xu, Lei Zuo, and Zhenhua Huang. Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences*, 261:1–31, 2014.
- [9] Jacek Bialowas, Beata Grzyb, and Pawel Poszumski. Firing cell: An artificial neuron with a simulation of long-term-potential-related memory. *arXiv preprint arXiv:1704.06593*, 2017.
- [10] Xumeng Zhang, Wei Wang, Qi Liu, Xiaolong Zhao, Jinsong Wei, Rongrong Cao, Zhihong Yao, Xiaoli Zhu, Feng Zhang, Hangbing Lv, et al. An artificial neuron based on a threshold switching memristor. *IEEE Electron Device Letters*, 39(2):308–311, 2018.
- [11] Fernando Mejía, Byron Vaca, Eduardo Villa, Washington Luna, and Raul Rosero. Programming languages that minimize response times of an artificial neuron. *KnE Engineering*, 1(2):16–43, 2018.
- [12] BL Ho and Rudolf E Kálmán. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.
- [13] Karl-Johan Åström and Bohlin Torsten. Numerical identification of linear dynamic systems from normal operating records. *IFAC Proceedings Volumes*, 2(2):96–111, 1965.
- [14] Brian DO Anderson, John B Moore, and RM Hawkes. Model approximations via prediction error identification. *Automatica*, 14(6):615–622, 1978.
- [15] Ljung Lennart and Peter E Caines. Asymptotic normality of prediction error estimators for approximate system models. *Stochastics*, 3(1-4):29–46, 1980.
- [16] Lennart Ljung et al. Theory for the user. In *System Identification*. Prentice-hall, Inc., 1987.

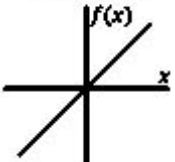
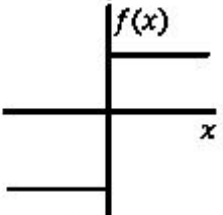
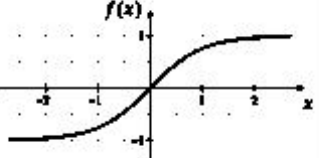
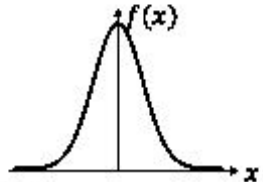
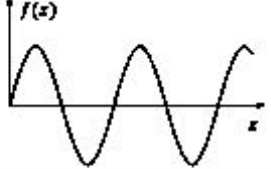


- [17] A. Mohamed Heikki Koivo Faisal. Modelling of induction motor using non-linear neural network system identification. *SICE Annual Conference in Sapporo, Control Eng. Lab.*, 2004.
- [18] Emmanuel G. Collins Wassim M. Haddad Douglas L. Hunter, David C. Hyland. Neural network system identification for improved noise rejection. *American Control Conference Seattle*, 1995.
- [19] Olivera Jovanović. Identification of dynamic system using neural network. *The Scientific Journal FACTA UNIVERSITATIS Series: Architecture and Civil Engineering*, 31:525–532, 1997.
- [20] EI Gaura, N Steele, and RJ Rider. A neural network approach for the identification of micromachined accelerometers. In *Proceedings of the Second International Conference on Modeling and Simulation of Microsystems*, pages 245–248, 1999.
- [21] Yong-Geun Lee. Three phase active rectifier power conditioning using neural network system identification. In *1998 International Conference on Power Electronic Drives and Energy Systems for Industrial Growth, 1998. Proceedings.*, volume 1, pages 219–224. IEEE, 1998.
- [22] Wen Yu and América Morales. Gasoline blending system modelling via static and dynamic neural networks. *International Journal of Modelling and Simulation*, 24(3):151–160, 2004.
- [23] Wen Yu, Alexander S Poznyak, and Edgar N Sanchez. Dynamic multilayer neural networks for nonlinear system on-line identification. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No. 00CH37147)*, pages 25–30. IEEE, 2000.
- [24] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [25] Paul J Werbos et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [26] F. Rosenblatt. Principles of neurodynamics. east lansing. 1962.
- [27] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [28] JA Anderson, E Rosenfeld, et al. Neurocomputing. foundations of research, vol. 1, 1988.
- [29] Bernard Widrow and Michael A Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.
- [30] A Cichocki and Rolf Unbehauen. *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc., 1993.
- [31] J.J.Hopfield. Neurons with grade response have collective computational properties like those of a two-state neurons. *Proc. of the National Academy of Science*, 81:3088–3092, 1984.
- [32] Hassan. K. Khali. Nonlinear systems,. *New Jersey, Prentice Hall*, 2002.
- [33] Anzures M Juan, Padilla G José, and Cuevas S Omar. Estabilidad de sistemas no-lineales: Sistema de nivel de líquidos de dos tanques interconectados. *RIEE&C*, 2008.
- [34] M. Vidyasagar. Nonlinear systems analysis,. *Second Edition, Prentice Hall*, New Jersey, 1995.
- [35] W. Li J.J.E. Slotine. Applied nonlinear control,. *Prentice Hall*, New Jersey, 1991.
- [36] Nikolaĭ Piskunov, KP Medkov, et al. *Cálculo diferencial e integral*, volume 1. Mir, 1983.
- [37] Erik Talvila. Necessary and sufficient conditions for differentiating under the integral sign. *Amer. Math. Monthly* 108, no. 6, 544–548, 2001.

- 
- [38] Rosa Barbolla, Paloma Sanz, et al. *Álgebra lineal y teoría de matrices*. 1998.
- [39] Fernando Lara and Fundación Universitaria Konrad Lorenz. *Derivación matricial*. Bogotá: *Fundación Universitaria Konrad Lorenz*, 2007.
- [40] Antonio Javier Barragán Piña. *Síntesis de Sistemas de Control Borroso Estables por Diseño: Tesis Doctoral*. A. Javier Barragán Piña, 2010.

# Anexo A

## Funciones de activación

	Función	Rango	Gráfica
<b>Identidad</b>	$y = x$	$[-\infty, \infty+]$	
<b>Escalón</b>	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
<b>Sigmoidea</b>	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
<b>Gaussiana</b>	$y = Ae^{-\beta x^2}$	$[0, +1]$	
<b>Sinusoidal</b>	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

# Anexo B

---

## Manipulación algebraica para acotar el error de aproximación

---

Para llegar a la expresión (3.27), se toma como punto de partida (3.16) a la cual se le añade un 0, dicho de otra manera,

$$\dot{J} = \frac{\partial J}{\partial w_a} \dot{w}_a + \frac{\partial J}{\partial w_b} \dot{w}_b + \frac{\partial J}{\partial \theta} \dot{\theta} + e_y^2 - e_y^2. \quad (\text{B.1})$$

Para los primeros 3 términos del lado derecho de la igualdad (B.1) se demostró que son semidefinidos negativos, por tanto se supone que el cuarto término del lado derecho de la igualdad también lo es, ya que todos se encuentran acoplados directamente al error de aproximación  $e_y$ , dicho de otra forma,

$$\frac{\partial J}{\partial w_a} \dot{w}_a + \frac{\partial J}{\partial w_b} \dot{w}_b + \frac{\partial J}{\partial \theta} \dot{\theta} + e_y^2 \leq 0, \quad (\text{B.2})$$

esto implica que

$$\dot{J} \leq -e_y^2. \quad (\text{B.3})$$

Luego se continúa a realizar una serie de manipulaciones algebraicas:

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} \dot{J}(\tau) d\tau \leq \lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} -e_y^2(\tau) d\tau, \quad (\text{B.4})$$

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} [J(t_f) - J(0)] \leq - \lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} e_y^2(\tau) d\tau, \quad (\text{B.5})$$

Despejando el término del lado derecho a la izquierda se cambia de signo, y despejando el término del lado izquierdo a la derecha de igual forma, esto es

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} e_y^2(\tau) d\tau \leq \lim_{t_f \rightarrow \infty} \frac{1}{t_f} [J(0) - J(t_f)], \quad (\text{B.6})$$

sí se toma el lado derecho de la inecuación (B.6), y se toma como cota superior el mayor de los términos, es decir,  $J(0)$ , entonces

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} e_y^2(\tau) d\tau \leq \lim_{t_f \rightarrow \infty} \frac{1}{t_f} J(0), \quad (\text{B.7})$$

Matemáticamente hablando en el límite un número entre infinito es igual a cero, pero filosóficamente hablando el error de aproximación nunca será cero, y en este trabajo de tesis no se define un valor específico de la cota para el error, dicho de esta manera, solo se define como un error estable y acotado. Por eso es que se formula como sigue

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} J(0) < \infty, \quad (\text{B.8})$$


entonces

$$\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} e_y^2(\tau) d\tau < \infty, \quad (\text{B.9})$$


por lo tanto si el límite del valor promedio del área bajo la curva del error cuadrático de aproximación cuando el tiempo tiende a infinito es acotado (B.9), por consiguiente

$$\lim_{t \rightarrow \infty} |e_y(t)| < \infty. \quad (\text{B.10})$$

## Jornada de Ciencia y Tecnología Aplicada



TECNOLÓGICO NACIONAL DE MÉXICO



cenidet  
Centro Nacional de Investigaciones y Desarrollo Tecnológico

### PROPUESTA DE UNA NEURONA ARTIFICIAL DINÁMICA PARA PROPÓSITOS DE MODELADO ENTRADA-SALIDA.

Henry Magdaleno Luna Pliego\*, Juan Reyes Reyes\*  
\*Departamento de Ingeniería Electrónica - Tecnológico Nacional de México - CENIDET.

#### Antecedentes/ Problemática

- En la actualidad se cuentan con múltiples estudios e investigaciones sobre aplicaciones de neuronas artificiales, en especial con estructuras más complejas como redes neuronales artificiales, donde se ve involucrada más de una neurona artificial en los procesos.
- Invocando a Cichocki el cual propone que se puede realizar identificación de sistemas utilizando funciones acotadas.
- Se propone realizar aproximación de un conjunto de funciones estáticas y dinámicas con una sola neurona artificial.
- Una neurona artificial impactará significativamente en menor costo computacional, con respecto a lo que demanda una red neuronal artificial.
- Se pretende que la neurona artificial pueda realizar la aproximación de funciones estáticas y dinámicas.
- Se pretende que una misma estructura neuronal con un mismo algoritmo de aprendizaje logre estas aproximaciones.

#### Propuesta de solución

Como parte del planteamiento de base se propone el sistema expresado en (1) como la estructura general de la neurona artificial dinámica.

$$\begin{cases} \dot{x}(t) = -\frac{1}{\tau_p}x(t) + \frac{1}{\tau_p}v(v_1(t) - v_2(t)) \\ y(t) = x(t) \end{cases} \quad (1)$$

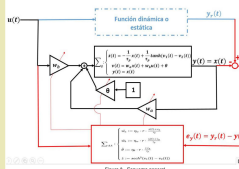
Donde:

$$y(t) = x(t) \in \Omega \subset \mathbb{R}$$
$$0 < \tau_p \ll 1$$
$$v(t) = w_1 \cdot x(t) + w_2 \cdot u(t) + \theta$$
$$u(t) \in \Omega^m$$

En el sistema expresado en (2) y (3) se muestra la estructura de la neurona artificial dinámica con entrada escalar y el sistema que representa el aprendizaje respectivamente.

$$\begin{cases} \dot{x}(t) = -\frac{1}{\tau_p}x(t) + \frac{1}{\tau_p} \tanh(v_1(t) - v_2(t)) \\ v(t) = w_1 v(t) \cdot x(t) + w_2 v(t) \cdot u(t) + \theta(t) \\ y(t) = x(t) \end{cases} \quad (2)$$
$$\sum_{i=1}^m \begin{cases} \dot{w}_1(t) := \eta w_1 \cdot r \cdot \frac{x(t)(1-x(t))}{\tau_p} \\ \dot{w}_2(t) := \eta w_2 \cdot r \cdot \frac{u(t)(1-x(t))}{\tau_p} \\ \dot{\theta}(t) := \eta \theta \cdot r \cdot \frac{y(t)(1-x(t))}{\tau_p} \\ \dot{v}_1(t) := \text{sech}^2(v_1(t) - v_2(t)) \\ \dot{v}_2(t) := -\text{sech}^2(v_1(t) - v_2(t)) \end{cases} \quad (3)$$
$$r := \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

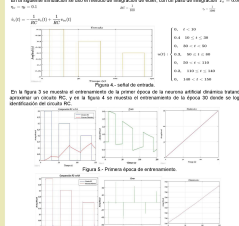
En la figura 1 se presenta el esquema general del trabajo de investigación.



#### Resultados de simulación


**Identificación de un sistema:**

El objetivo de esta simulación es aproximar un sistema RC con la neurona artificial dinámica propuesta como el sistema de más adelante. El algoritmo de aprendizaje es el algoritmo de aprendizaje basado en el método de mínimos cuadrados. Se usó un modelo de la planta de un sistema de segundo orden con la siguiente estructura:  $G(s) = \frac{1}{s^2 + 2s + 1}$ . Se usó un modelo de la planta de un sistema de segundo orden con la siguiente estructura:  $G(s) = \frac{1}{s^2 + 2s + 1}$ . Se usó un modelo de la planta de un sistema de segundo orden con la siguiente estructura:  $G(s) = \frac{1}{s^2 + 2s + 1}$ .



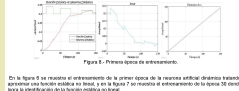
En la figura 3 se muestra el entrenamiento de la primer etapa de la neurona artificial dinámica basada en aprendizaje por mínimos cuadrados. En la figura 4 se muestra el entrenamiento de la etapa 20 desde el inicio de la identificación del sistema RC.

**Pruebas de estabilidad:**



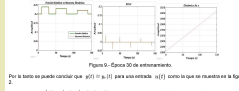
En la figura 5 se muestra el entrenamiento de la primer etapa de la neurona artificial dinámica basada en aprendizaje por mínimos cuadrados. En la figura 6 se muestra el entrenamiento de la etapa 20 desde el inicio de la identificación del sistema RC.

**Pruebas de estabilidad:**




En la figura 7 se muestra el entrenamiento de la primer etapa de la neurona artificial dinámica basada en aprendizaje por mínimos cuadrados. En la figura 8 se muestra el entrenamiento de la etapa 20 desde el inicio de la identificación del sistema RC.

**Pruebas de estabilidad:**



En la figura 9 se muestra el entrenamiento de la primer etapa de la neurona artificial dinámica basada en aprendizaje por mínimos cuadrados. En la figura 10 se muestra el entrenamiento de la etapa 20 desde el inicio de la identificación del sistema RC.

**Pruebas de estabilidad:**



#### Motivación/ Justificación

Función estática  $y_d = f(x)$   
Donde  $y_d \in \Omega \subset \mathbb{R}$   
 $x \in \Omega^m$




Figura 1. Función estática.

**Sistema dinámico no lineal**

$$\sum_{i=1}^m \dot{x}_i = F(x, u)$$
$$y_d = x$$

Filtro pasa bajas, Frecuencia de corte, Frecuencia de corte, Frecuencia de corte.




Figura 2. Sistema dinámico.

#### Objetivos


- General:**
  - Desarrollar una propuesta de arquitectura neuronal para el modelado entrada-salida de un grupo de funciones estáticas y dinámicas, dentro de un subconjunto real, con una función de activación dinámica y acotada, y los pesos sinápticos realizando una retroalimentación de la salida y una prealimentación de las entradas.
- Específicos:**
  - Definir la estructura general de la neurona artificial dinámica.
  - Definir el algoritmo de aprendizaje para los dos pesos sinápticos y el umbral.
  - Definir el algoritmo de aprendizaje del peso sináptico  $w_1$  y el umbral  $\theta$  con múltiples entradas.
  - Definir el algoritmo de aprendizaje del peso sináptico  $w_2$  y el umbral  $\theta$  con múltiples entradas.
  - Realizar la comparación con distintos métodos de optimización.
  - Comparar el esquema de una sola neurona con redes neuronales artificiales.


#### Alcances

- Formalizar el resultado indicando las condiciones necesarias para lograr la aproximación.
- Formulación matemática de los algoritmos de aprendizaje propuestos y la estructura general de la neurona.
- Generación de un teorema producto de la investigación, con su respectiva demostración matemática.
- Validación de resultados obtenidos únicamente a simulación.

#### Bibliografía

- [1] Protter, Murray H.; Morrey, Charles B., Jr. (1985). "Diferenciación bajo el signo integral". *Cálculo intermedio* (Segunda ed.). Nueva York: Springer. pp. 421-426.
- [2] Flinders, Harry (junio-julio de 1973). "Diferenciación bajo el signo integral" (PDF). *American Mathematical Monthly*. **80** (6): 615-627.
- [3] Bialowas, J., Grzyb, B., & Poszumny, P. (2017). Firing Cell: An Artificial neuron with a simulation of Long-Term-Potentialiation-Related Memory. *arXiv preprint arXiv:1704.06593*.
- [4] Zhang, X., Wang, W., Liu, Q., Zhao, X., Wei, J., Cao, R., ... & Long, S. (2016). An Artificial Neuron Based on a Threshold Switching Memistor. *IEEE Electron Device Letters*, *36*(2), 308-311.
- [5] Mejía, F., Vaca, B., Villa, E., Luna, W., & Rosero, R. (2018). Programming Languages that minimize response times of an Artificial Neuron. *KiE Engineering*, *12*(1), 16-43.







**TECNOLÓGICO  
NACIONAL DE MÉXICO**

---

***cenidet***<sup>®</sup>  
*Centro Nacional de Investigación  
y Desarrollo Tecnológico*