



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

**MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA,
OPCIÓN EN AUTOMATIZACIÓN**

**Diseño de una red neuronal Hopfield en hardware
embebido**

T E S I S

**PRESENTADA PARA OBTENER EL TÍTULO DE:
MAESTRO EN CIENCIAS DE LA ELECTRÓNICA,
OPCIÓN EN AUTOMATIZACIÓN**

PRESENTA

ING. DIEGO RUIZ SÁNCHEZ DE LA VEGA*

DIRECTORES DE TESIS

DRA. OLGA GUADALUPE FÉLIX BELTRÁN (FCE-BUAP)

DR. JESÚS MANUEL MUÑOZ PACHECO (FCE-BUAP)

Puebla, Puebla, Mayo 2025

***Becario SECIHTI**

Agradecimientos

Me gustaría agradecer de manera especial a mi familia, es por ellos que debo la fuerza para poder concluir con mis estudios y con la realización de esta tesis. En especial quiero agradecer a mi madre Gloria Sánchez de la Vega Escalante y a mi abuela Gloria Escalante González, gracias a ellas, por su esmero y dedicación, así como entrega y sacrificio.

También agradezco en gran medida el apoyo y asesoría de la Dra Olga Guadalupe Félix Beltrán y el Dr. Jesús Manuel Muñoz Pacheco, su conocimiento y paciencia hizo posible la realización de este trabajo. Así mismo, agradezco al jurado de tesis, Dra. María Aurora Diozcora Treviño, Dr. Carlos Leopoldo Pando Lambruschini y Dr. Richard Torrealba Meléndez por sus valiosas observaciones que contribuyeron a mejorar la Tesis realizada. A su vez, agradezco a la Maestría en Ciencias de la Electrónica, a la Facultad de Ciencias de la Electrónica y en general a la Benemérita Universidad Autónoma de Puebla por acogerme durante mi estancia académica y formarme como el profesionista que soy. Finalmente, agradecer a todos mis compañeros de la facultad con quienes compartí trabajo y esfuerzo, me permitieron convertirme en mejor estudiante y persona.

El autor agradece a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SE-CIHTI), por los recursos y apoyo ofrecido para la elaboración de este trabajo.

Dedicatoria

A mi abuela y a mi madre.

Resumen

Esta tesis presenta un análisis teórico basado en el caos del modelo de red neuronal de Hopfield (HNN), seguido del desarrollo de una metodología para su implementación en circuitos analógicos programables de prototipado rápido. Para ello, primero se establecieron los fundamentos teóricos necesarios para construir una red neuronal de Hopfield. Posterior, se realizó una revisión de los antecedentes de las redes neuronales artificiales (ANN), su relación con las neuronas biológicas y el primer modelo matemático asociado. Además, se profundiza en el estudio de las redes neuronales de Hopfield (HNN), su formulación general y su implementación en ambientes electrónicos. Dado que ciertos parámetros inducen comportamiento caótico en las HNN, se analizan sistemas dinámicos caóticos y su vinculación con estas redes. Mas adelante, se realiza el análisis dinámico e implementación en hardware, en donde se presentan modelos específicos de HNN de 3 neuronas, cada uno con funciones de activación distintas, incluyendo tres nuevos sistemas caóticos no reportados en la literatura. Para cada modelo, se realiza un análisis dinámico (existencia de atractores, estabilidad de Lyapunov) y se identifican regiones caóticas mediante diagramas de bifurcación y exponentes de Lyapunov. A su vez, se introduce un modelo de HNN de 4 neuronas con comportamiento hipercaótico, validando numéricamente su acotamiento, sensibilidad a condiciones iniciales y existencia de atractores. Los parámetros que inducían el comportamiento caótico fueron encontrados de la literatura, la aportación en esta tesis es su validación en un entorno de circuitos analógicos. Adicionalmente, se adaptaron las funciones de activación para su implementación en tarjetas analógicas programables (FPAA), desarrollando las ecuaciones de diseño necesarias para su programación en AnadigmDesigner2 y registrando los resultados experimentales. Como parte de la implementación experimental, se desarrollan dos interfaces universales, la primera en Qt-Python y la segunda en C++ y MFC, estas permiten configurar parámetros y funciones de activación para HNN de 3 y 4 neuronas, facilitando su implementación en hardware. Como contribución final, se diseña una HNN con la mayor simplicidad registrada, misma que a su vez presenta una cantidad n -ésima de atractores y multiestabilidad, empleando una única no linealidad, que es implementada en las tarjetas analógicas usando la interfaz creada. Como resultado del trabajo de investigación, se obtiene un nuevo conjunto de parámetros para la matriz de pesos en el modelo HNN para los cuales existe una región más grande (inclusive infinita) para la cual el modelo HNN presenta una rica dinámica, dicho modelo se verifica experimentalmente, realizando su implementación en la tarjeta analógica embebida. Además, se propone un algoritmo para identificación de dinámicas caóticas en modelos cuya divergencia es no uniformemente conservativa o disipativa, dicho algoritmo es más eficiente en tiempo y complejidad comparado con los algoritmos de identificación de caos como el exponente máximo de Lyapunov o los diagramas de bifurcación.

Avance de Tesis I

Conceptos clave y estado del arte

Periodo de trabajo

Del 7 de agosto 2023 - 16 de diciembre de 2023

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

**Maestría en Ciencias de la Electrónica
Opción en Automatización**



Avance de Tesis I

Diseño de una red neuronal Hopfield en hardware embebido

Alumno: Diego Ruiz Sánchez de la Vega

**Directores de Tesis: Dra. Olga Guadalupe Félix Beltrán (FCE-BUAP)
Dr. Jesús Manuel Muñoz Pacheco (FCE-BUAP)**

Puebla, Puebla.

Noviembre de 2023.

Resumen

Resumen general

En esta tesis se lleva a cabo el diseño e implementación de una red neuronal Hopfield de tres y cuatro neuronas en hardware embebido programable analógico, caracterizando la dinámica no lineal que a ella describe, así como también se realiza un estudio sobre las distintas funciones no lineales de activación, evaluando los efectos en el proceso de aprendizaje de la red neuronal Hopfield. Con este fin inicialmente se desarrolla una revisión bibliográfica sobre el estado del arte de las redes neuronales artificiales, en específico sobre la red neuronal Hopfield, sus antecedentes, principales aplicaciones, implementaciones tanto en circuitos analógicos como con dispositivos programables analógicos y digitales, así como las ecuaciones que describen el modelo de la red neuronal Hopfield. Posteriormente, se desarrolla un análisis exhaustivo sobre la dinámica no lineal del sistema, este incluye la evaluación del proceso de aprendizaje de la red neuronal, así como, el análisis en base a caos para finalmente implementar el modelo de la red neuronal Hopfield con distintas funciones de activación en un dispositivo analógico programable, el cual, requiere sus propias ecuaciones de diseño, las cuales también son desarrolladas. Finalmente, con todo el trabajo realizado, se elabora una metodología para la implementación de las redes neuronales Hopfield en las tarjetas analógicas embebidas utilizadas a lo largo de esta tesis.

Resumen Avance de Tesis I

En el Avance de Tesis I estudian los antecedentes de las redes neuronales artificiales, su comparación con las neuronas biológicas y se presenta el primer modelo matemático que a ellas describe, posteriormente, se presenta el modelo de las redes neuronales Hopfield (HNN) general y específico para 3 y 4 neuronas, dado que es el tema de interés en esta tesis. Simultáneamente, se realiza un estudio sobre la implementación de las HNN, ya sea con circuito embebidos o no, su respuesta y aplicaciones principales. Otro tema de interés en esta tesis son los sistemas dinámicos caóticos esto debido a que, dados ciertos parámetros, la red neuronal Hopfield presenta caos. Con respecto a este tema, una investigación es realizada con relación a los sistemas

dinámicos caóticos y a las HNN en modo caótico. La implementación en hardware embebido de redes neuronales es un tema de recurrente estudio en la comunidad científica, por lo que una investigación a fondo en el diseño en hardware programable de las HNN, resulta evidente en el cual se incluye, un estudio e investigación sobre los dispositivos analógicos programables FPAA's, dado que este circuito será el utilizado para el desarrollo de esta tesis. Finalmente, para el análisis del modelo de la red neuronal Hopfield, es necesario desarrollar herramientas matemáticas analíticas y numéricas, con lo que respecta a este avance, se presentan dos métodos para la integración numérica por medio de los métodos de Runge-Kutta de segundo y cuarto orden.

Índice general

Resumen	II
Introducción	II
Objetivos	VII
Cronograma de Actividades	IX
Diagrama de bloques de la tesis	XI
1. Estado del arte	1
2. Marco Teórico	4
2.1. Redes Neuronales Artificiales	4
2.1.1. Redes Neuronales Hopfield	5
2.1.2. Funciones de activación	6
2.1.3. Entrenamiento de una red neuronal	8
2.2. Sistemas dinámicos	9
2.2.1. Sistemas dinámicos caóticos	10
2.2.2. Redes Neuronales Hopfield en modo caótico	10
2.3. Dispositivos programables	12
2.3.1. Hardware Analógico Embebido	12
2.4. Métodos numéricos para la solución de sistemas dinámicos	13
Conclusiones	16
Bibliografía	18

Introducción

Antecedentes

El estudio del comportamiento del cerebro humano tiene profundas raíces en la historia de la ciencia y la medicina, desde la antigüedad grandes filósofos especulaban sobre su funcionamiento, los antiguos griegos plantearon ideas sobre “maquinas pensantes” [1]. El siglo XIX marcó un avance importante con respecto al estudio de la biología; en 1839 Schleiden y Schwann formulan la Teoría Celular, que establece que todos los seres vivos están formados por células como unidades fundamentales [2], esto por supuesto incluye al cerebro.

Una de las células básicas que componen al sistema nervioso es la neurona, las neuronas procesan y transmiten información en forma de señales eléctricas, como fue propuesto por el científico español Santiago Ramón y Cajal [3], en su trabajo sobre la transmisión neuronal, en el cual se considera a la neurona como un interruptor que se “dispara” dependiendo de las señales de otras neuronas, estas señales pueden ser inhibitorias si tienden a impedir que la neurona se active o excitatorias si provocan que la neurona se active, con estas señales la neurona realiza suma ponderada y, dependiendo del valor de esta suma, se dice que la neurona está “activa” o no.

La neurona es una de las unidades fundamentales del sistema nervioso y se divide en cuatro partes importantes: el núcleo, el cuerpo de la célula, la dendrita y el axón (véase Figura 1). El cuerpo de la célula contiene al núcleo, el cual se encarga de producir la energía de la neurona, la entrada de la neurona que trata con las señales de las demás neuronas es denominada como dendrita, mientras que la salida es denominada como axón; a la conexión entre las neuronas se le conoce como sinapsis.

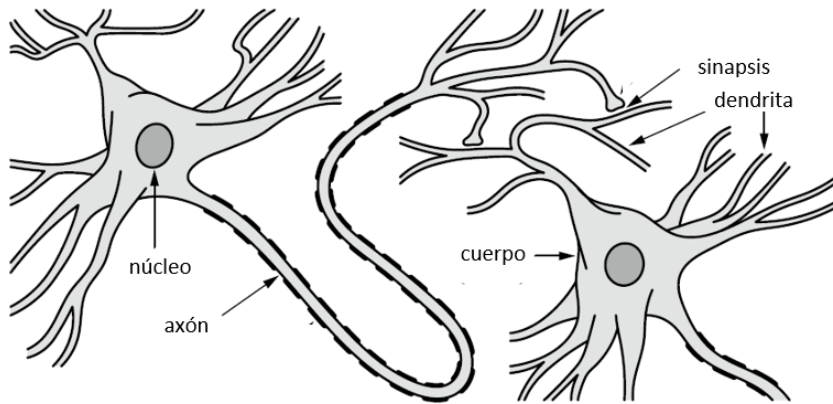


Figura 1: Neurona biológica.

Inspirados por el comportamiento de las neuronas biológicas, los investigadores y científicos han desarrollado las denominadas redes neuronales artificiales (ANN, por sus siglas en inglés), las cuales son dispositivos que simulan el comportamiento de las redes neuronales biológicas. El primer modelo de ANN fue el desarrollado por Warren McCulloch y Walter Pitts en 1943 [4], quienes proponen un modelo que toma múltiples entradas binarias ponderadas, suma estas entradas y aplica una función de umbral (véase Figura 2). Si la suma supera el umbral, la neurona emite una señal de salida. Este enfoque permitió realizar cálculos lógicos y de decisión basados en la entrada recibida, semejante al comportamiento de las neuronas biológicas.

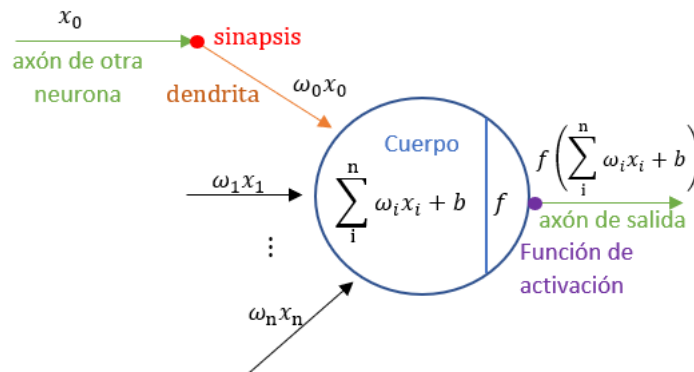


Figura 2: Modelo matemático de una neurona artificial.

En 1958, Frank Rosenblatt, padre del perceptrón [5], consiguió implementar la primera máquina capaz de aprender, distinguir y separar cartas marcadas utilizando el modelo matemático McCulloch-Pitts. El modelo del perceptrón, contiene solo una capa de entradas y una única neurona de salida, donde cada conexión entre las neuronas tiene un valor específico denominado “peso”, el cálculo del valor de la neurona de salida se realiza de la siguiente forma: se realiza el producto entre la entrada con su respectivo peso, estos valores se suman y, final-

mente, el valor calculado se introduce como argumento en una función denominada “función de activación”, para el perceptrón la función utilizada es la función *signo*, la cual es adecuada para una clasificación [6].

Si bien, el trabajo de Rosenblatt no fue bien recibido por la comunidad científica debido a sus limitantes, años después se desarrollaron modelos que solucionaban las deficiencias del perceptrón, algunos de ellos son: las redes neuronales convoluciones, redes neuronales recurrentes, perceptrón multicapa, entre otros [7]. Tiempo después al trabajo desarrollado por McCulloch-Pitts, John Hopfield propone un nuevo modelo para las ANN, denominado como red neuronal Hopfield o HNN por sus siglas en inglés, esta es una red de tipo recurrente y completamente conectada, esto es, a diferencia de los modelos de McCulloch-Pitts, las neuronas del modelo de la HNN están conectadas con todas las demás, además de tener cierta dependencia con su estado anterior, este tipo de ANN se utiliza generalmente para describir algunos procesos cerebrales, como por ejemplo, el proceso de aprendizaje y el concepto de memoria asociativa, la ecuación que generaliza cada neurona de la HNN viene dada por [8]

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^n \omega_{ij} \tanh(x_j) + I_i,$$

donde x_i es la entrada de la neurona, i es el número de neurona, $\tanh(\cdot)$ es la función de activación, w_{ij} es el peso entre las entradas de la neurona, c es un parámetro constante y finalmente la I_i es un factor de sesgo o bias.

En la Figura 3 se ilustra el diagrama a bloques de una neurona Hopfield, la cual podemos asociar a la función de propagación con la sumatoria de todas las sinapsis que posee la neurona, mientras que el bloque $f(x_i)$ representa la función de activación. Para las redes neuronales Hopfield es utilizada la función tangente hiperbólica $\tanh(\cdot)$ y, finalmente, existe un bloque de derivada, el cual nos permite solucionar la ecuación diferencial para adquirir el valor actual de la neurona [9, 10].

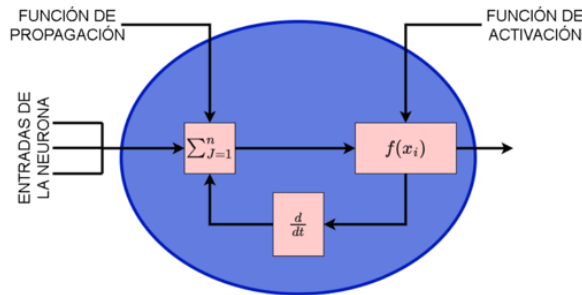


Figura 3: Diagrama en bloques de una red neuronal Hopfield.

En este escenario, debido a las capacidades de aprendizaje de las neuronas artificiales, es requerido un diseño electrónico en cual puede ser usado para aplicaciones de máquinas inteligentes, las cuales requieren implementaciones en hardware optimizadas, es decir, rápidas, baja complejidad computacional y con bajo consumo de potencia. Note que la HNN, al utilizar una menor cantidad de neuronas, requiere un menor consumo de potencia electrónica o computacional para su funcionamiento, además, las HNN son muy utilizadas debido a que, comparada con otros modelos de ANN, no existe una mejor red neuronal para la identificación de patrones.

Planteamiento del problema

En esta investigación se tiene como objetivo realizar una metodología para el desarrollo y diseño en la implementación de redes neuronales Hopfield en dispositivos programables analógicos, además de realizar un análisis exhaustivo con respecto a la dinámica del sistema y evaluar los efectos en el proceso de aprendizaje usando distintas funciones de activación.

Infraestructura

- Laboratorio de Circuitos y Sistemas Caóticos de Orden Fraccional, 1FCE4/201.
- Software especializado.
- Laptops y estaciones de trabajo.
- Tarjetas electrónicas.

Justificación

Las redes neuronales son sistemas que nos permiten dar solución a problemáticas en donde se requiere de dispositivos inteligentes y adaptables, en particular, las redes neuronales Hopfield nos permiten, cifrar datos, reconocimiento de imágenes y, principalmente, la detección de patrones, entre otros. En específico, la detección de patrones es crucial para lograr, en un futuro, la comunicación entre dispositivos inteligentes capaces de manejarse de manera autónoma sin la necesidad de la supervisión humana.

Actualmente, no hay red neuronal que sea capaz de almacenar una gran cantidad de patrones como lo hace la red neuronal Hopfield, gracias a que sus neuronas son capaces de almacenar más información que cualquier otra red neuronal existente [11].

Si bien la implementación de este tipo de red neuronal ya ha sido desarrollada por trabajos pasados, el desarrollo de una metodología para la implementación en circuitos electrónicos de

las redes neuronales Hopfield no es del todo clara y este es uno de los temas que esta tesis busca resolver.

El modelo de la red neuronal Hopfield está descrito por un sistema de ecuaciones diferenciales no lineales, su estructura, dados ciertos parámetros, presenta un comportamiento caótico, en otras palabras, bajo parámetros adecuados la implementación de una red neuronal Hopfield nos permite obtener un sistema caótico y, con él todas las propiedades que ellos nos ofrecen. Considerando lo anterior, en este trabajo se realiza un análisis y estudio de la red neuronal en modo caótico.

Dentro del modelo que a las redes neuronales describe, las funciones de activación son fundamentales, a través de los años distintas funciones de activación han sido desarrolladas, cada función de activación da distintas propiedades dinámicas a la red neuronal, así como también, afecta directamente el proceso de aprendizaje de dicha red neuronal, y este análisis es otro de los objetivos primarios de este trabajo.

Estructura del documento

El presente documento se encuentra seccionado por capítulos, donde cada uno aborda diferentes aspectos relativos al problema de investigación. Inmediatamente después de la introducción, donde se abordan los antecedentes, conceptos básicos y el estado del arte, se mencionan los objetivos de la tesis, el cronograma y el diagrama de bloques de la tesis, que se considera como la metodología de la misma. En el primer capítulo se abordan la construcción de una revisión sobre las implementaciones previas de las redes neuronales artificiales, ya sea en forma de hardware o software, es realizada, de nuevo y al ser el foco de interés, la implementación de las HNN en hardware embebido programable es de principal interés y la investigación apunta a este nodo. Finalmente, en el capítulo dos se abordan los antecedentes de las redes neuronales, haciendo un principal énfasis en las redes neuronales Hopfield, en este capítulo se hace una revisión bibliográfica sobre los conceptos de las redes neuronales artificiales. Al ser otro tema de interés en la tesis, se hace una investigación de los sistemas dinámicos caóticos, en donde se busca realizar una introducción en donde se incluyan los conceptos básicos para la comprensión de los mismos.

Objetivos

Los objetivos planteados en este tema de tesis, objetivo general y objetivos particulares, se enuncian a continuación, junto a estos últimos un desglose más específico de dicho objetivo particular.

Objetivo general

Determinar los efectos de diferentes funciones de activación en el proceso de aprendizaje de una red neuronal Hopfield en modo caótico en un entorno de circuitos electrónicos.

Objetivos particulares

- Establecer la estructura básica de una red neuronal Hopfield de tres y cuatro neuronas.
 - Determinar los modelos matemáticos para una HNN de 3 y 4 neuronas.
 - Realizar la integración numérica de los modelos de 3 y 4 neuronas de una HNN.
- Caracterizar la dinámica no lineal de la red neuronal Hopfield usando técnicas analíticas y numéricas.
 - Realizar el análisis dinámico de los modelos matemáticos de las HNN para distintas funciones de activación.
 - Realizar el análisis desde un punto de vista caótico de los modelos de la red neuronal Hopfield.
- Evaluar la respuesta en el aprendizaje de la red neuronal Hopfield con diferentes funciones no lineales de activación.
 - Evaluar los efectos en el proceso de aprendizaje de distintas funciones de activación en los modelos de las redes neuronales Hopfield de 3 y 4 neuronas.

- Proponer las ecuaciones de diseño para implementar la red neuronal Hopfield en un sistema analógico embebido programable de prototipado rápido.
 - Simulación del comportamiento de la red neuronal Hopfield en el entorno de programación de AnadigmDesigner2.
 - Desarrollar una metodología para el diseño e implementación de una red neuronal Hopfield con 3 y 4 neuronas.
- Implementar los modelos desarrollados de la red neuronal Hopfield para 3 y 4 neuronas en la tarjeta analógica embebida.
 - Evaluar la dinámica obtenida.
 - Evaluar los efectos en el proceso de aprendizaje de los modelos implementados y comparar resultados con las simulaciones desarrolladas.

Cronograma de Actividades

En la Tabla 1 se presentan las actividades propuesta para el desarrollo esta tesis, titulada *Diseño de una red neuronal Hopfield en hardware embebido*. Se remarcan en color rosa los objetivos abordados en Tesis I. Algunas de las actividades aún están en el proceso de ser realizada, por lo que, la cuantificación de las actividades considera sólo un porcentaje de la actividad realizada y es presentado en la columna de “Avance realizado” junto con un porcentaje equivalente al valor en porcentaje de la actividad del total de la tesis.

Actividades	Tesis I	Tesis II	Tesis III	Avance realizado
Revisión bibliográfica y estado del arte.	X			80 % (10 %)
Planteamiento de los modelos matemáticos de las redes neuronales Hopfield de 3 y 4 neuronas.	X			40 % (5 %)
Simulación de las redes neuronales Hopfield e integración por métodos numéricos.	X	X		40 % (10 %)
Análisis de la dinámica de los modelos de las redes neuronales Hopfield de 3 y 4 neuronas mediante técnicas numéricas y analíticas.		X	X	0 % (15 %)
Evaluación del proceso de aprendizaje de las HNN de 3 y 4 neuronas con distintas funciones de activación.		X	X	0 % (15 %)
Desarrollo de las ecuaciones para la implementación de las redes neuronales Hopfield en la tarjeta embebida analógica.		X	X	0 % (10 %)
Implementación de la red neuronal Hopfield con distintas funciones de activación.		X	X	0 % (10 %)
Desarrollo de una metodología para el diseño e implementación de la red neuronal Hopfield usando tarjetas embebidas analógicas de prototipado rápido.		X	X	0 % (10 %)
Publicación de un artículo.			X	0 % (5 %)
Análisis y comparación de resultados			X	0 % (5 %)
Escritura de la tesis.	X	X	X	0 % (5 %)
	Porcentaje acumulado			16.5 %

Tabla 1: Cronograma de actividades.

Gráfica del avance global de la tesis

A continuación, en la Figura 4 se presenta el avance global que se tiene de este trabajo de tesis, el cálculo de este está basado en la ponderación de cada una de las actividades descritas en la Tabla 1 y del porcentaje realizado de cada una de ellas. Considerando lo anterior, el porcentaje desarrollado con respecto a la actividad “Revisión bibliográfica y estado del arte” fue de un 80 %, así como para la actividad “Planteamiento de los modelos matemáticos de las redes neuronales Hopfield de 3 y 4 neuronas”, por otro lado, para la actividad descrita como “Simulación de las redes neuronales Hopfield e integración por métodos numéricos” se asigna un 40 % de avance, esto último tomando como referencia el trabajo realizado en el capítulo 2 y, finalmente, con respecto a la escritura de tesis desarrollada en este trabajo corresponde al 10 %. La suma ponderada de las actividades descritas es equivalente al **16 %**.

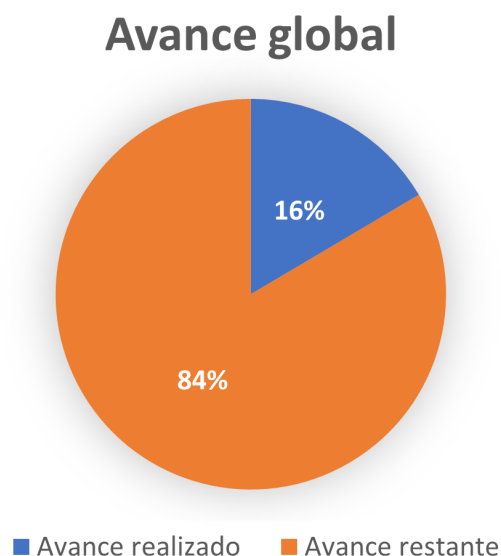


Figura 4: Gráfica con el avance global.

Diagrama de bloques de la tesis

A continuación se presenta, en la Figura 5, la propuesta general del proceso que se lleva a cabo durante el desarrollo de esta investigación. En esta se indica el periodo de inicio y desarrollo de cada actividad en un color diferente, considerando que marca el inicio y no necesariamente el final de cada actividad. El color verde indica el primer semestre, azul el segundo y morado el tercero.

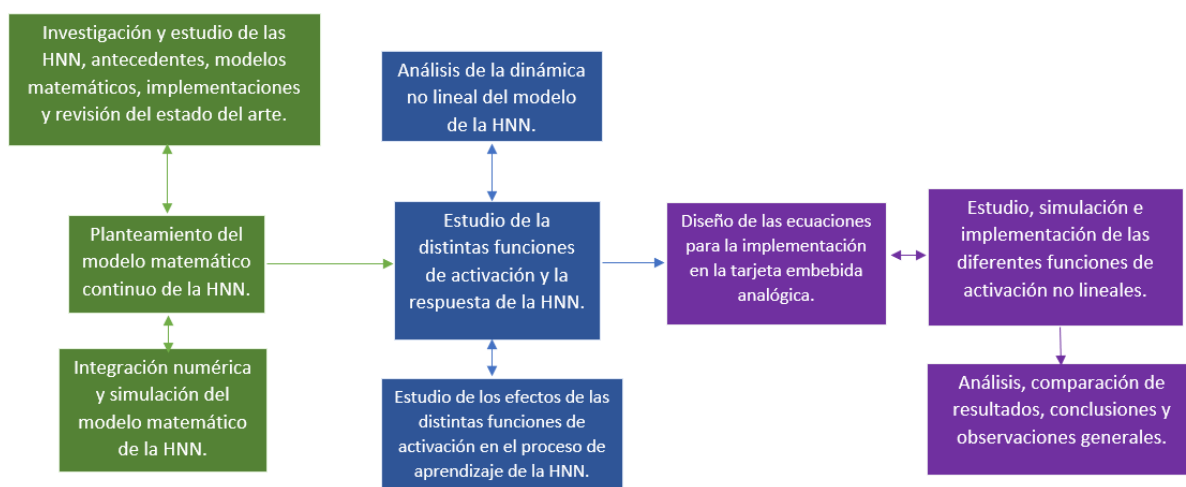


Figura 5: Resumen del trabajo por realizar en la tesis dividido por colores en diagrama de bloques.

Bloques para el primer semestre

Con lo que respecta al trabajo realizado con base en los bloques coloreados por color verde, correspondientes a las actividades desarrolladas y en desarrollo del primer semestre comprendido en agosto y diciembre del año 2023, se desglosaron de forma que sea posible describirlo más a fondo.

- **Investigación y estudio de las HNN, antecedentes, modelos matemáticos, implementaciones y revisión del estado del arte:** Esta actividad consiste en investigar y redactar los sucesos de mayor relevancia con respecto a los temas de interés de esta

tesis, es decir, de las redes neuronales artificiales, las HNN y su implementación. Posteriormente, se realiza el estado del arte de este trabajo, el cual consiste en sintetizar los avances más recientes desarrollados por la comunidad científica y hacer un pequeño repaso del impacto de estos.

- **Planteamiento del modelo matemático continuo de la HNN:** Junto con la investigación sobre las redes neuronales de Hopfield es posible hallar el modelo matemático con el cual se caracteriza en las referencias [8, 11–13], entre otras, por lo que, ayudado por esta investigación, es posible plantear un primer modelo para las HNN de 3 y 4 neuronas, en la sección 2.1.1 se presenta el modelo general en cuestión que expresa tanto el modelo de 3 como de 4 neuronas.
- **Integración numérica y simulación del modelo matemático de la HNN:** Un primer acercamiento al modelo matemático fue desarrollado al simular una de las ecuaciones que describen a la red neuronal de Hopfield, las ecuaciones y parámetros elegidos son los correspondientes al trabajo [12], cuya respuesta es caótica. La simulación fue implementada en el lenguaje de programación de Python usando el método de integración numérica de Runge-Kutta de cuarto orden.

Capítulo 1

Estado del arte

Como parte del estado del arte se realizo una revisión e investigación sobre los temas de mayor interés en este trabajo, esto es, redes neuronales Hopfield (HNN), sistemas dinámicos caóticos y hardware analógico embebido.

Revisión bibliográfica

En lo referente a las redes neuronales se realizo una investigación sobre los antecedentes, modelos matemáticos que a ellas describen e implementaciones sobre las redes neuronales artificiales, con un enfoque en las redes neuronales Hopfield, cuya implementación de 3 y 4 neuronas por medio de circuitos analógicos y digitales fue estudiada.

Por otro lado, y al ser un tema recurrente en la tesis, se realizo un estudio de los sistemas dinámicos, esto debido a que el modelo de la HNN es un sistema dinámico como es evidente por su modelo matemático [11], es importante considerar que, el ser un sistema dinámico no lineal, dados ciertos parámetros una HNN puede presentar un comportamiento caótico [12,14,15], con lo que también se realiza un estudio de los sistemas dinámicos caóticos, esto es, la investigación para definir lo que es un sistema dinámico caótico junto con sus características.

Actualidad de las HNN

Desde su creación por el Dr. Jonh J. Hopfield [8], las redes neuronales Hopfield han atraído a la comunidad científica por sus cimientos matemáticos, la aplicación de esta red neuronal está relacionada a la memoria asociativa y al reconocimiento de patrones [10]. Sin embargo, y como es demostrado por distintos trabajos de investigación, su aplicación va mas allá de ello y puede resolver problemas que usualmente son resueltos por otros modelos de redes neuronales. Además, debido a las características de su modelo matemático, su comportamiento puede ser caótico y con esto, conlleva todas las aplicaciones que los sistemas dinámicos caóticos tienen, como lo son los generadores de aleatoriedad, la criptografía, modelado de sistemas, entre otros.

Implementación de las HNN

La implementación de las redes neuronales Hopfield no es ajeno a los circuitos eléctricos, en la actualidad la implementación de un circuito no lineal es facilitada gracias a los dispositivos programables como lo son los FPGA's, múltiples trabajos han tomado esta aproximación para la implementación de una red neuronal Hopfield [12, 14, 16]. Por otro lado, también se ha desarrollado una electrónica analógica para la implementación de una red neuronal Hopfield [17–20], dicha electrónica esta basada en dispositivos activos y resistivos, la estructura presentada en estos trabajos es muy similar, sin embargo, en 2017 Bao et al. [16] desarrollo un modelo para una red neuronal Hopfield de 3 neuronas en base a memristores y partir de este trabajo múltiples desarrollos han sido llevados a cabo [15]. La implementación en un dispositivo digital implica discretizar el modelo matemático que se tiene del sistema continuo, por lo que el comportamiento obtenido será una aproximación a modelo que describe la red neuronal, mientras que la implementación en dispositivos analógicos, aunque limitada en otros aspectos, nos permite obtener el comportamiento exacto de nuestro modelo matemático.

Para gran parte de las implementaciones de las redes neuronales Hopfield, las metodologías para el diseño del circuito electrónico descritas con anterioridad son utilizadas, sin embargo, existen dispositivos analógicos programables denominados FPAA, con los cuales es posible desarrollar una metodología para la implementación de las HNN [21, 22], estos nos permiten tener la flexibilidad que nos ofrece un dispositivo programable sin la necesidad de discretizar el modelo de las HNN.

HNN en modo caótico

Una de las grandes razones por la cual las redes neuronales Hopfield resultan de interés en la comunidad científica es su riqueza dinámica a pesar de su simple estructura, su no linealidad nos permite modelar sistemas muy complejos, recientes trabajos relacionados a la construcción de una red neuronal Hopfield con base en Memristores han mostrado que estos pueden presentar caos [16, 17]. A partir de estos trabajos se han desarrollado otras estructuras y parámetros sobre las HNN en las cuales se han observado, no solo comportamientos caóticos, sino hipercaóticos [23], esto último considerando una red neuronal de cuatro neuronas.

Almacenamiento y reconocimiento de patrones

Las HNN son especialmente útiles en tareas de almacenamiento y recuperación de patrones asociativos. Pueden recordar y recuperar patrones incluso cuando la entrada está incompleta o contaminada con ruido [24]. La dinámica de las HNN se rige por una función de energía y el aprendizaje de patrones se realiza ajustando los pesos para minimizar esta energía. Sin

embargo, la estabilidad dinámica puede ser un desafío, especialmente cuando se trata de grandes conjuntos de datos o patrones complejos.

A pesar de estas fortalezas, las HNN también tienen limitaciones. No son ideales para manejar patrones complejos o grandes conjuntos de datos como lo hacen otras arquitecturas más modernas, como las redes neuronales convolucionales (CNN, por sus siglas en inglés) o las redes neuronales recurrentes (RNN, por sus siglas en inglés). Además, el modelo original de Hopfield puede tener problemas con la capacidad de almacenamiento y la recuperación de patrones en grandes dimensiones.

Modelado biológico

Las HNN han sido utilizadas para modelar la memoria asociativa, un aspecto de la memoria biológica, esta es capaz de reconocer patrones enteros usando solo partes parciales o dañadas por el ruido de ese mismo patrón [24]. La capacidad de las HNN para almacenar y recuperar patrones asociativos puede interpretarse como un intento de emular procesos de asociación en el cerebro [23], es importante tener en cuenta que estas redes son una simplificación de los sistemas biológicos. Aunque se inspiran en algunos aspectos de la neurobiología, no capturan la complejidad y la dinámica real de las redes neuronales biológicas.

Funciones de activación en las HNN

El propósito de las funciones de activación son las de determinar cuando una neurona se “dispara” o “enciende”, además, su no linealidad permite modelar comportamientos mas complejos. En la literatura se puede hallar una lista de diferentes funciones de activación [25], cada una ofrece distintos comportamientos y respuestas de la red neuronal, como es el caso de la identificación de imágenes [10]. Resulta de interés en esta tesis ser capaz de diferenciar entre qué funciones de activación son apropiadas para alguna aplicación en específico, así como la respuesta que cada una de estas tiene una vez implementada en un dispositivo programable.

Capítulo 2

Marco Teórico

2.1. Redes Neuronales Artificiales

Unas de las tecnologías de más renombres desarrolladas en los últimos años por la comunidad científica son las redes neuronales artificiales (ANN, por sus siglas en inglés). Estas toman el comportamiento de las redes neuronales biológicas observadas en el cerebro humano como base para el desarrollo de un modelo matemático que emule dicho comportamiento. El primer modelo de las redes neuronales biológicas fue el desarrollado por Warren McCulloch y Walter Pitts [4], en el cual las neuronas se describen como interruptores, cuyo estado es dependiente del de las demás neuronas. A la salida de cada neurona se le denominó como axón y a su entrada como “dendrita”, a este sistema de conexiones entre neuronas se le conoce como sinapsis. Como base de esta descripción, el psicólogo Frank Rosenblatt desarrolló el primer modelo de las ANN conocido como perceptrón, en este, cada conexión entre las neuronas, es decir, la sinapsis, tiene un valor específico denominado como peso, consideraba neuronas de entrada y una sola neurona de salida, el modelo es presentado en la Figura 2.1, en la cual se observa tanto la representación matemática como en un diagrama. Tras el trabajo de Rosenblatt, más modelos de ANN fueron desarrollados, cada uno con distintas propiedades, aún sin embargo,

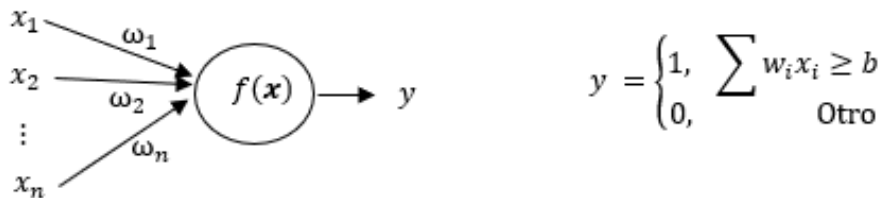


Figura 2.1: Modelo del perceptron.

los conceptos cimentados por este modelos se conservan y siguen siendo usados en modelos nuevos de las ANN. En modelos más recientes de ANN se requiere de una mayor cantidad de neuronas para la resolución de la mayoría de problemas. De acuerdo a la estructura de una red neuronal, estas se pueden separar en dos partes fundamentales, las redes neuronales, si el modelo que describe la estructura de una red neuronal no contiene ciclos o retroalimentaciones, es denominada como red neuronal prealimentada o “feed-forward network”, de lo contrario, se le denomina como red recurrente o *recurrent network* [25].

2.1.1. Redes Neuronales Hopfield

Un tipo especial de red neuronal recurrente es la red neuronal Hopfield, desarrolladas por John Hopfield en 1980 [8], estas redes destacan por su capacidad única para el almacenamiento y la recuperación de patrones asociativos, emulando de alguna manera el funcionamiento de la memoria humana [24]. Esta estructura permite a la red aprender patrones y recuperar información basándose en la activación y desactivación de sus neuronas. Cada conexión entre neuronas, ponderada por un valor sináptico, contribuye a la capacidad de la red para almacenar y recordar patrones. Por otro lado, las HNN son ideales para modelar la dinámica del cerebro [8], además de su dinámica no lineal. En general una red neuronal Hopfield de 4 neuronas puede ser representada por el diagrama de la Figura 2.2. Debido a su estructura las HNN son generalmente conocidas como sistemas dinámicos retroalimentados el cual produce múltiples patrones [24], dependiendo de la condición inicial de la HNN esta puede converger a un punto específico, estos puntos suelen ser denominados como “atractores”. El diagrama mostrado por la Figura 2.2 puede ser generalizado para n neuronas considerando la misma forma en la sinapsis. Para una HNN de n neuronas, el modelo matemático continuo es el expresado por el sistema de ecuaciones no diferenciales adimensional dado por [26]:

$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{f}(\mathbf{x}) + I, \quad (2.1)$$

donde

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}, I = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_i \\ \vdots \\ I_n \end{bmatrix}, W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nj} & \dots & w_{nn} \end{bmatrix}, C = \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix}. \quad (2.2)$$

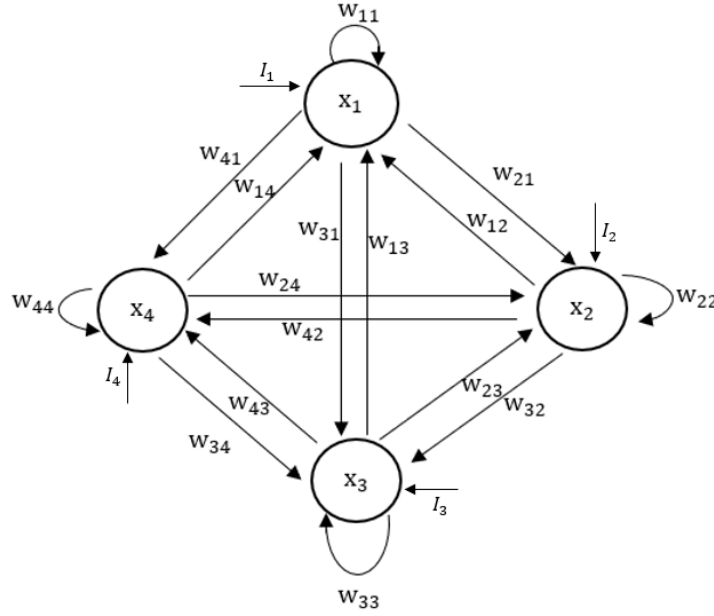


Figura 2.2: Diagrama de una red neuronal Hopfield de 4 neuronas.

x_i denota la i -ésima neurona, I_i representa un bias o un valor externo sobre la red neuronal, W es la matriz de pesos que representa la sinapsis del modelo y finalmente $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$ representa la función de activación no lineal aplicada en cada una de las variables del modelo y C una matriz diagonal. En específico, esta tesis está interesada en estudiar los modelos de 3 y 4 neuronas de una red neuronal Hopfield.

2.1.2. Funciones de activación

En cada modelo estudiado se han mencionado las denominadas funciones de activación, esta cumplen con distintos propósitos en los que se incluye, principalmente, la introducción de la no linealidad a la red neuronal permitiendo modelar sistemas mas complejas, capacidad de aprendizaje, ajuste a la salida de la red neuronal. Sin embargo, la parte de los modelos de redes neuronales no utilizan exclusivamente funciones de activación, dependiendo del modelo utilizado cada neurona utiliza una de tres tipos distintos de funciones [25]:

- Función de entrada, $f : \mathbb{R}^{k_1} \rightarrow \mathbb{R}$.
- Función de activación, $f : \mathbb{R}^{k_2} \rightarrow \mathbb{R}$.
- Función de salida, $f : \mathbb{R} \rightarrow \mathbb{R}$.

Los valores de k_1 y k_2 son números enteros positivos dependientes del tipo y número de argumentos de las funciones.

En lo que respecta a las redes neuronales Hopfield, estas son redes especiales cuyas neuronas no utilizan funciones de entrada ni de salida, solo de activación, la entrada de esta red son los valores con los que se inicia cada neurona. Por esta razón este trabajo se limita al estudio de las funciones de activación, las cuales deben cumplir las siguientes características:

- No lineal. Una función es considerada como no lineal si esta no cumple con el principio de superposición, esto es, para que una función $f(x)$ sea denominada como lineal debe cumplir

$$f(ax + by) = af(x) + bf(y),$$

para todo a, b constantes y todo x, y en el dominio de la función f . Si una función no cumple con la propiedad anterior para alguno de sus puntos es considerada como no lineal.

- Diferenciable. Se dice que una función es diferenciable si existe la derivada de la función en todos los puntos en el dominio de interés. Esto es, una función $f(x)$ es diferenciable en un dominio $[a, b]$ si el limite dado por:

$$\lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h},$$

existe para ese mismo dominio.

- Ser acotada. Una función $f(x)$ se dice acotada si existen dos constantes M y N tales que:

$$N \leq f(x) \leq M,$$

para todo el dominio de $f(x)$.

- Monótona. Una función se considera monótona en un intervalo si su comportamiento de crecimiento o decrecimiento no cambia en ese intervalo. Consideremos la función $f(x)$, si cualquier par de puntos x_1 y x_2 en su dominio cumple con:

$$x_1 \leq x_2 \Rightarrow f(x_1) \leq f(x_2),$$

se dice que la función es monótona creciente, si por lo contrario con la misma hipótesis $f(x_1) \geq f(x_2)$ se dice que la función es monótona decreciente. Por otro lado, si la desigualdad es estricta (es decir $f(x_1) < f(x_2)$ ó $f(x_1) > f(x_2)$) la función se dice estrictamente monótona.

Las principales funciones de activación [27] usadas por cualquier red neuronal se presentan la Tabla 2.1.

Tabla 2.1: Funciones de activación de las ANN más frecuentes en la literatura.

Función de activación	$\phi(x)$	$\phi'(x)$	Rango
Tangente hiperbólica	$\tanh(x)$	$1 - \phi^2(x)$	$(-1,1)$
Sigmoid	$S(x) = \frac{1}{1 + e^{-x}}$	$\phi(x)(1 - \phi(x))$	$(0,1)$
ReLU	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$[0, \infty)$
Función Headviside	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$\delta(x)$	$[0, 1]$
Función signo	$R(x) = \begin{cases} -1 & \text{para } x < 0 \\ 0 & \text{para } x = 0 \\ 1 & \text{para } x > 0 \end{cases}$	$2\delta(x)$	$[-1, 1]$
Softmax	$y_i = \frac{e^{x_j}}{\sum_j^n e^{x_j}}$	$\frac{\partial y_i}{\partial j} = y_i(\delta_{ij} - y_i)$	$(0, 1)$

La función de activación utilizada, al estar directamente implicada en el modelo matemático de la ANN, impacta directamente en la dinámica de la red neuronal.

2.1.3. Entrenamiento de una red neuronal

Una de las propiedades más atractivas de las redes neuronales (artificiales) es la posibilidad de entrenarlas para ciertas tareas con la ayuda de datos de ejemplo. En general entrenar una red neuronal consiste en adaptar los pesos de conexión y posiblemente otros parámetros (como bias) de manera que se optimice cierto criterio. Dependiendo del tipo de datos de entrenamiento y del criterio a optimizar, podemos distinguir dos tareas fundamentales de aprendizaje [25]: aprendizaje supervisado, en el cual se asigna una salida deseada a un conjunto de entrada, y aprendizaje no supervisado en el cual no se proporcionan salidas deseadas.

En lo que se refiere a una red neuronal Hopfield, su entrenamiento se refiere a diseñar un atractor o atractores para un conjunto específico de condiciones iniciales [24]. Al conjunto de estos atractores se les conoce como memoria, mientras que al conjunto de condiciones iniciales que son atraídas por un atractor se les conoce como “cuenca de atracción”, a los valores que toman las neuronas una vez que estas convergieron a su valor final se le denomina como “patrón”. Considerando lo anterior, una red neuronal Hopfield puede ser visto como un dispositivo capaz de almacenar y recuperar información relacionada entre si. Con el paso del tiempo distintos investigadores han desarrollado algoritmo para el entrenamiento de HNN y cuyo estudio sera realizado a lo largo del desarrollo de esta tesis.

2.2. Sistemas dinámicos

La gran parte de los sistemas estudiados por la física y la matemática tienen un comportamiento variante en el tiempo, es decir, no son estáticos, a este tipo de sistemas se les denomina como sistemas dinámicos y su estudio se remonta a Isaac Newton, quien intentó describir el comportamiento de los planetas alrededor del Sol usando ecuaciones diferenciales [28]. Desde aquel momento el estudio de los sistemas dinámicos se ha aplicado a diversas disciplinas dentro y fuera de la física. Gran parte de los sistemas dinámicos son representados por un sistema de ecuaciones diferenciales descrito por:

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ \dot{x}_2 &= f_2(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p).\end{aligned}\tag{2.3}$$

La ecuación (2.3) representa un sistema de n ecuaciones diferenciales, donde $u_1, u_2, \dots, u_p$ representan la entrada al sistema, $x_1, x_2, \dots, x_n$ las variables del sistema y $\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n$ de derivada con respecto al tiempo de las variables. De forma reducida la expresión anterior puede ser escrita usando vectores, de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}),\tag{2.4}$$

donde $\mathbf{f}(t, \mathbf{x}, \mathbf{u}) = [f_1(t, \mathbf{x}, \mathbf{u}), \dots, f_n(t, \mathbf{x}, \mathbf{u})]^T$, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ y $\mathbf{u} = [u_1, u_2, \dots, u_p]^T$. En las expresiones (2.3) y (2.4), el tiempo t está expresado explícitamente, en algunos sistemas dinámicos este no es el caso y no dependen explícitamente ni del tiempo ni de la entrada, a este tipo de sistemas se les denomina como sistemas dinámicos autónomos y están expresados por

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}).\tag{2.5}$$

Su estudio es de gran relevancia en esta tesis al representar el modelo matemático de las HNN.

Uno de los aspectos de relevancia en los sistemas dinámicos son los *puntos de equilibrio*. Se dice que un punto $\mathbf{x}$ es un punto de equilibrio de un sistema si siempre que este inicie en el estado $\mathbf{x}$ entonces se mantendrá en estado $\mathbf{x}$ para cualquier estado siguiente [29]. Para el sistema dado por (2.5), los puntos de equilibrio se encuentran al resolver

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = 0.\tag{2.6}$$

La expresión anterior implica que un punto de equilibrio es un punto en donde el estado del sistema se mantiene constante. El análisis de los puntos de equilibrio del sistema es muy

importante si se desea comprender el comportamiento de dicho sistema.

2.2.1. Sistemas dinámicos caóticos

En el 1961, el meteorólogo y matemático Edward Lorenz desarrolló una máquina capaz de predecir el comportamiento del clima, poco tiempo después, se dio cuenta que con mínimas variaciones en las condiciones iniciales, su sistema mostraba dos comportamientos completamente distintos, sus observaciones y resultados fueron publicadas en 1963 [30]. Su trabajo desarrolló lo que ahora se denomina como Teoría del Caos.

Un sistema dinámico caótico o sistema caótico, es un tipo especial de sistema dinámico, si bien, al día de hoy no existe un consenso general para dar una definición clara de los sistemas caóticos, si se tiene un consenso de tipo de características que estos sistemas deben cumplir [31]:

- **Determinista y no lineal.** Cuando un sistema dinámico está completamente determinado por sus condiciones iniciales se le denomina como determinista, además, cuando se adjudica el término de no lineal a un sistema, implica que este no cumple con la propiedad de la superposición para todos sus estados.
- **Extrema sensibilidad en sus condiciones iniciales.** Si bien, si un sistema caótico es determinista, su dependencia con respecto a sus condiciones iniciales es muy inestable, esto es, diminutas variaciones en la condiciones inicial resultan en cambios drásticos en el comportamiento del sistema.
- **Densidad en sus orbitas periódicas.** Aunque irregular, el comportamiento de un sistema caótico es periódico, esto es, su órbita gira alrededor de un conjunto de puntos denominado como atractores.

Es importante mencionar que conocido el sistemas de ecuaciones diferenciales que rigen el sistema caótico es posible determinar cómo evoluciona el sistema. Sin embargo, su extrema sensibilidad genera que, tras un cierto periodo de tiempo, no podamos predecir su comportamiento con absoluta certeza.

2.2.2. Redes Neuronales Hopfield en modo caótico

El modelo matemático de las redes neuronales de Hopfield se comporta como los sistemas caóticos con ciertos parámetros específicos [12, 15], en algunos de estos trabajos se ha llevado a cabo un implementación sobre algún tipo de dispositivo programable, en cuyos casos la respuesta de la HNN ha conservado sus propiedades caóticas.

La identificación de un comportamiento caótico en el modelo de la red neuronal Hopfield ha requerido de herramientas utilizadas en teoría de control, como lo son los diagramas de

bifurcación, exponentes de Lyapunov o el cálculo y análisis de los puntos de equilibrio del modelo matemático. Más específicamente, para una red neuronal Hopfield de 3 neuronas, la presencia de un comportamiento caótico fue hallado usando el modelo descrito por [14]:

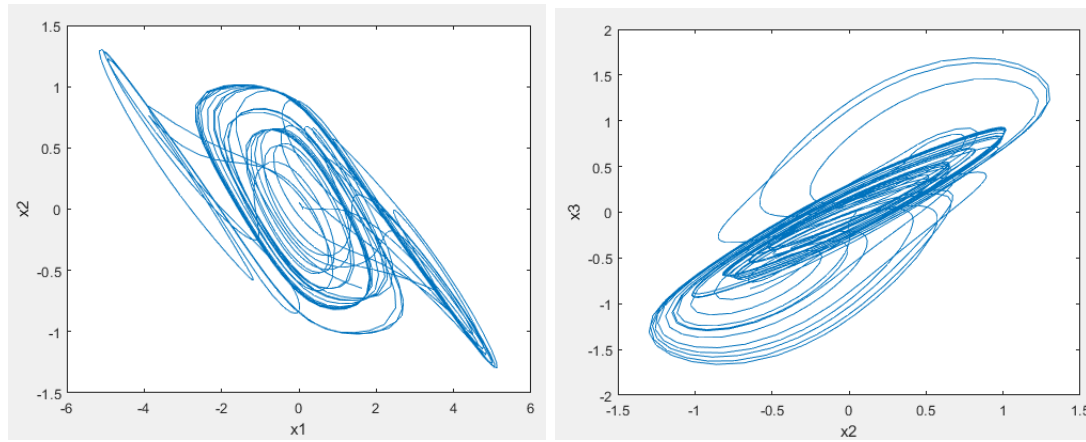
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \end{bmatrix}, \quad (2.7)$$

dada la matriz de pesos

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} 0 & 1.2 & -7 \\ 1.1 & 0 & 2.8 \\ 0.7 & -2 & 4 \end{bmatrix}.$$

Un diagrama de fase es una herramienta gráfica utilizada en el análisis de sistemas dinámicos para visualizar el comportamiento a lo largo del tiempo de un conjunto de variables relacionadas. En este tipo de diagrama se gráfica una variable del sistema contra otra distintas, el diagrama de fase proporciona una representación visual de cómo las variables evolucionan y se relacionan entre sí a medida que el sistema dinámico evoluciona.

Los diagramas de fase obtenidos del sistema (2.7) se presentan en la Figura 2.3.



(a) Diagrama de fase x_1 contra x_2 .

(b) Diagrama de fase x_2 contra x_3

Figura 2.3: Diagramas de fase 2D formados por el sistema 2.7.

Cabe mencionar que para obtener el diagrama de fase anterior fue requerida la simulación del sistema y, con esta, la solución numérica del sistema caótico, el método utilizado es presentado en la sección 2.4.

2.3. Dispositivos programables

En muchas ocasiones cuando se desarrolla un circuito electrónico es necesario ajustar o incluso rediseñar partes del circuito. Para el caso específico de este trabajo, este rediseño o ajuste resulta ser incluso más complejo en términos de estabilidad y el desarrollo del circuito requerido. Por esto, el uso de dispositivos programables es de gran utilidad en la electrónica, en lo que respecta a dispositivos digitales se utilizan los llamados “Arreglo de Puertas Programables en Campo” (FPGA, por sus siglas en inglés), su contraparte analógica es menos conocida, sin embargo, nos ofrece la misma ventaja, estos son denominado como “Arreglo de Campo de Analógico Programable” (FPAA, por sus siglas en inglés).

2.3.1. Hardware Analógico Embebido

Los FPAA's son dispositivos electrónicos programables que permiten la creación de circuitos analógicos personalizados y, como tal, nos permiten hacer operaciones como lo son la suma, resta, comparación, integración, derivación, entre otros. La arquitectura general de un FPAA es presentada en la Figura 2.5, en esta se observa que está constituido por bloques analógicos configurables (CAB's, por sus siglas en inglés), los cuales tienen una función de transferencia equivalente, estos bloques están interconectados por interruptores controlados. En lo que

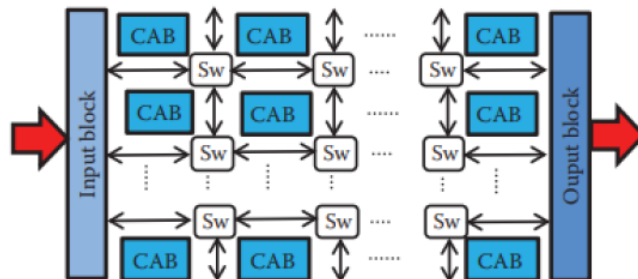


Figura 2.4: Arquitectura general de un FPAA [32].

respecta a este trabajo y su implementación, el FPAA utilizado es el modelo AN231E04 desarrollado por Anadigm, la misma distribuidora ofrece un kit que contiene una tarjeta con 4 FPAA's. Por otro lado, el desarrollador y distribuidor ofrece el software para programar la tarjeta con los AN231E04 llamado AnadigmDesigner2, el cual proporciona un entorno de diseño gráfico con el cual podemos diseñar y simular circuitos analógicos. El ambiente de trabajo de AnadigmDesigner2 (Figura 2.5) consiste en bloques, los cuales representan el FPAA sobre el cual se programará y dentro de este se pueden implementar Módulos Analógicos Configurables (CAM's, por sus siglas en inglés); cada uno de estos tiene una función de transferencia equivalente que se encuentra en la documentación dada por el desarrollador.

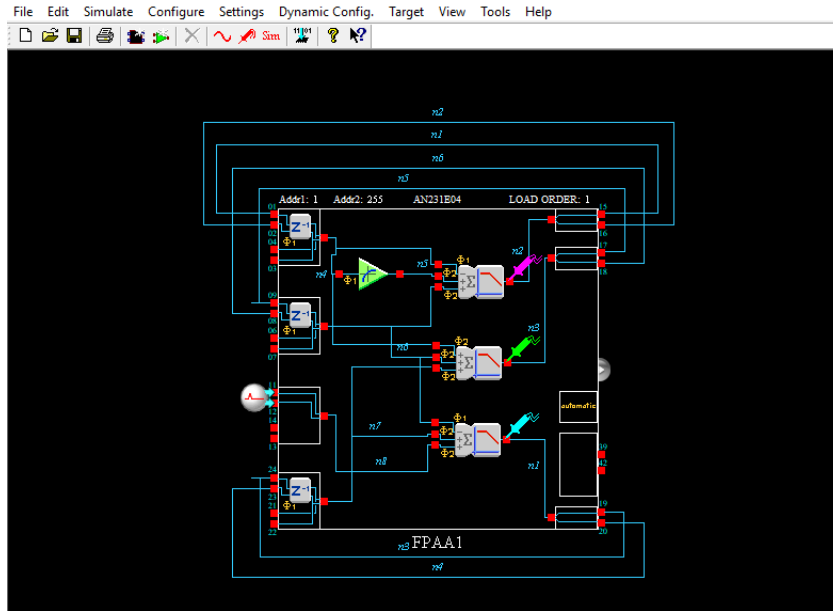


Figura 2.5: Entorno de programación en bloques de AnadigmDesigner2.

2.4. Métodos numéricos para la solución de sistemas dinámicos

En muchos problemas de la física y las matemáticas se estudian modelos descritos por ecuaciones diferenciales lineales y no lineales, es por esto que una solución para dichos sistemas es de suma relevancia. Sin embargo, no siempre es posible el desarrollo de una solución analítica del modelo matemático y, en estos casos, es necesario utilizar métodos numéricos que aproximen a la solución deseada.

Existen una gran cantidad de métodos numéricos para dar solución a un sistema de ecuaciones diferenciales; con lo que respecta a esta tesis, se utilizarán los métodos desarrollados por los matemáticos alemanes Carl David Tolmé Runge y Martin Wilhelm Kutta, a fines del siglo XIX y principios del siglo XX [33]. En 1895, Tolmé Runge publicó un artículo titulado “Über die numerische Auflösung von Differentialgleichungen”(Sobre la resolución numérica de ecuaciones diferenciales), donde propuso un método iterativo para la solución de ecuaciones diferenciales. Posteriormente, en 1901, Martin Wilhelm Kutta publicó un trabajo titulado “Beiträge zur näherungsweise Integration totaler Differentialgleichungen”(Contribuciones a la integración aproximada de ecuaciones diferenciales totales), donde desarrolló y popularizó un método más amplio y generalizado, que incluía el método de Runge y otros métodos similares. Este método se conoció como el método de Runge-Kutta, combinando los apellidos de ambos matemáticos [33].

Dentro del grupo de métodos Runge-Kutta, en este trabajo se tomarán dos metodologías,

el método de Runge-Kutta de segundo orden, también llamado método de Heun, y el método de Runge-Kutta de cuarto orden. El desarrollo para la obtención del método de Heun es el siguiente: consideremos la ecuación diferencial ordinaria (EDO) de la forma

$$\frac{dy(t)}{dt} = f(t, y). \quad (2.8)$$

Aplicamos la expansión de Taylor a la función $y(t)$ alrededor de t_n :

$$y(t) = y(t_n) + (t - t_n) \frac{dy}{dt} \Big|_{t_n} + \frac{(t - t_n)^2}{2} \frac{d^2y}{dt^2} \Big|_{t_n} + O((t - t_n)^3), \quad (2.9)$$

donde $O((t - t_n)^3)$ representa términos de orden superior, que estamos ignorando para obtener una aproximación de segundo orden.

Integrando ambos lados de la ecuación de la EDO de t_n a $t_{n+1} = t_n + h$ (donde h es el tamaño del paso) y aplicando la regla del trapecio para la integración, obtenemos:

$$y_{n+1} = y_n + \frac{h}{2} (f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})), \quad (2.10)$$

donde $\tilde{y}_{n+1}$ es una estimación de y_{n+1} que obtenemos usando la aproximación de Euler:

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n). \quad (2.11)$$

El método de Runge-Kutta de segundo orden ofrece una mejor aproximación a la solución de una ecuación diferencial, a diferencia del muy conocido método de Euler. Sin embargo, en gran parte de los trabajos de investigación se ha popularizado el método de Runge-Kutta de cuarto orden debido a su facilidad de programación y gran precisión en la solución de ecuaciones diferenciales. Consideremos nuevamente la expresión, (2.8)

$$\frac{dy(t)}{dt} = f(t, y).$$

Para la aplicación del método de Runge-Kutta de 4 orden se divide el intervalo $[t_n, t_{n+1}]$ en cuatro pasos y se calculan las pendientes intermedias k_1 , k_2 , k_3 , y k_4 :

$$\begin{aligned} k_1 &= h \cdot f(t_n, y_n), \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= h \cdot f(t_n + h, y_n + k_3). \end{aligned} \quad (2.12)$$

Se combina ponderadamente las cuatro pendientes para obtener la pendiente final y la solución se actualiza en el siguiente paso de tiempo t_{n+1} :

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (2.13)$$

La cantidad expresada por $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ es una estimación mejorada de la pendiente de la solución en el intervalo de tiempo $[t_i, t_i + h]$, k_1 es la pendiente en el extremo izquierdo del intervalo, k_2 es la pendiente utilizada en el punto medio del intervalo mientras que k_3 es la pendiente mejorada en este mismo punto, finalmente, k_4 es la pendiente en el punto extremo derecho.

Conclusiones

A lo largo de la investigación y análisis desarrollado en este Avance de Tesis I se puede decir que, las redes neuronales artificiales son unos de los temas que mas atención han recibido por parte de la comunidad científica debido a que nos permite solucionar problemas muy diversos, como lo pueden ser la identificación de patrones, el modelado, predicción, procesamiento de señales, entre otros. Las ANN son descritas a partir de modelos matemáticos que describen a las neuronas como interruptores cuyo estado depende de las demás neuronas y, en algunos casos, su propio estado pasado. En particular, esta tesis esta interesada en la redes neuronales Hopfield, la cuales son redes neuronales recurrentes completamente conectadas, es decir, una red neuronal con neuronas retroalimentadas y completamente conectadas entre si, este tipo de redes han demostrado ser de utilidad para el reconocimiento de patrones y el modelado biológico, aunque, sus aplicaciones no están solo limitadas a estas aplicaciones. El modelo de la HNN presenta un comportamiento caótico para destinos conjuntos de parámetros, considerando esto se realizo un estudio sobre sistemas dinámicos caóticos, así como una revisión sobre que modelos de HNN tienen este comportamiento. Con fines de simulación, es necesario poder solucionar los sistemas de ecuaciones diferenciales que describen al modelos de las redes neuronales, con este fin, se desarrollo el estudio de distintos métodos numéricos para realizar una integración numérica, mas específicamente, los métodos de Runge-Kutta de orden 2 y 4.

El estudio y análisis de las redes neuronales Hopfield ha llevado a los investigadores a proponer distintos modelos para su implementación en un entorno de circuitos electrónicos, desde el punto de vista de electrónica digital, la implementación de las HNN's es realizada en FPGA's debido a la complejidad de los modelos utilizados y la flexibilidad y estabilidad que un dispositivo programable nos ofrece, desde el punto de vista analógicos, distintos circuitos electrónicos activos han sido propuestos para distintas funciones de activación, dichos circuitos sacrifican la flexibilidad y estabilidad de un dispositivo programable por un menor consumo de potencia y un mejor apego al modelo de la red neuronal, sin embargo, existen dispositivos analógicos programables que serán de especial interés en esta tesis y nos permiten obtener la flexibilidad de un dispositivo programable sin tener que hacer una aproximación al modelo como podría ser en un dispositivo digital, este tipo de dispositivos son pocos explorados por

la comunidad científica aun sin embargo, nos ofrecen ventajas de relevancia, como lo es el prototipado rápido, la implementación de circuitos analógicos estables y sin la preocupación de un ruido que perturbe la señal deseada.

Bibliografía

- [1] Nathan W. Churchill, Robyn Spring, Cheryl Grady, Bernadine Cimprich, Mary K. Askren, Patricia A. Reuter-Lorenz, Mi Sook Jung, Scott Peltier, Stephen C. Strother, and Marc G. Berman. The suppression of scale-free fMRI brain dynamics across three different sources of effort: aging, task novelty and task difficulty. *Scientific Reports*, 6(1):30895, August 2016.
- [2] Bruce Alberts. *Molecular biology of the cell*. W. W. Norton & Company, New York, seventh edition edition, 2022.
- [3] Luis Alfredo Baratas Díaz. La obra neuro-embriológica de Santiago Ramón y Cajal. *Historiam Illustrandam*, 1(17):259–279, 1997.
- [4] N. B. Karayiannis and A. N. Venetsanopoulos. *Artificial Neural Networks*. Springer US, Boston, MA, 1993.
- [5] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [6] Charu C. Aggarwal. *Neural networks and deep learning: a textbook*. Springer, Cham, second edition edition, 2023. OCLC: 1390406271.
- [7] Liangqu Long and Xiangming Zeng. *Beginning deep learning with TensorFlow work with Keras, MNIST data sets, and advanced neural networks*. Apress, United States, 2022. OCLC: 1294216402.
- [8] John J. Hopfield and David W. Tank. Computing with Neural Circuits: A Model. *Science*, 233(4764):625–633, August 1986.
- [9] Anke Meyer-Baese and Volker Schmid. Transformation and Signal-Separation Neural Networks. In *Pattern Recognition and Signal Analysis in Medical Imaging*, pages 245–289. Elsevier, 2014.

- [10] Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal Hopfield Networks: A General Framework for Single-Shot Associative Memory Models. 2022. Publisher: arXiv Version Number: 2.
- [11] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need. 2020. Publisher: arXiv Version Number: 3.
- [12] Karthikeyan Rajagopal, Jesus M. Munoz-Pacheco, Viet-Thanh Pham, Duy Vo Hoang, Fawaz E. Alsaadi, and Fuad E. Alsaadi. A Hopfield neural network with multiple attractors and its FPGA design. *The European Physical Journal Special Topics*, 227(7-9):811–820, October 2018.
- [13] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, March 2021.
- [14] Daniel Clemente-López, Jesus M. Munoz-Pacheco, and Jose De Jesus Rangel-Magdaleno. A Review of the Digital Implementation of Continuous-Time Fractional-Order Chaotic Systems Using FPGAs and Embedded Hardware. *Archives of Computational Methods in Engineering*, 30(2):951–983, March 2023.
- [15] Han Bao, ZhuGuan Chen, JianMing Cai, Quan Xu, and BoCheng Bao. Memristive cyclic three-neuron-based neural network with chaos and global coexisting attractors. *Science China Technological Sciences*, 65(11):2582–2592, November 2022.
- [16] Bocheng Bao, Hui Qian, Quan Xu, Mo Chen, Jiang Wang, and Yajuan Yu. Coexisting Behaviors of Asymmetric Attractors in Hyperbolic-Type Memristor based Hopfield Neural Network. *Frontiers in Computational Neuroscience*, 11:81, August 2017.
- [17] Z. Tabekoueng Njitacke, Sami Doubla Isaac, J. Kengne, A. Nguomkam Negou, and Gervais Dolvis Leutcho. Extremely rich dynamics from hyperchaotic Hopfield neural network: Hysteretic dynamics, parallel bifurcation branches, coexistence of multiple stable states and its analog circuit implementation. *The European Physical Journal Special Topics*, 229(6-7):1133–1154, March 2020.
- [18] Chengjie Chen, Fuhong Min, Fei Hu, Jianming Cai, and Yunzhen Zhang. Analog/digital circuit simplification for Hopfield neural network. *Chaos, Solitons & Fractals*, 173:113727, August 2023.

- [19] Z.T. Njitacke and J. Kengne. Complex dynamics of a 4D Hopfield neural networks (HNNs) with a nonlinear synaptic weight: Coexistence of multiple attractors and remerging Feigenbaum trees. *AEU - International Journal of Electronics and Communications*, 93:242–252, September 2018.
- [20] Z. Tabekoueng Njitacke, J. Kengne, and H. B. Fotsin. Coexistence of Multiple Stable States and Bursting Oscillations in a 4D Hopfield Neural Network. *Circuits, Systems, and Signal Processing*, 39(7):3424–3444, July 2020.
- [21] Samuel Shapero, Christopher Rozell, and Paul Hasler. Configurable hardware integrate and fire neurons for sparse approximation. *Neural Networks*, 45:134–143, September 2013.
- [22] Pranav O. Mathews and Jennifer O. Hasler. Physical Computing for Hopfield Networks on a Reconfigurable Analog IC. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, Monterey, CA, USA, May 2023. IEEE.
- [23] Derong Liu, editor. *Advances in neural networks–ISNN 2007: 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007: proceedings*. Number 4491-4493 in Lecture notes in computer science. Springer, Berlin ; New York, 2007. Meeting Name: International Symposium on Neural Networks OCLC: ocn137334559.
- [24] Jay Kant Pratap Singh Yadav, Laxman Singh, and Zainul Abidin Jaffery. Optimization of Hopfield Neural Network (HNN) using multiconnection and Lyapunov Energy Function (LEF) for storage and recall of handwritten images. *Sāadhanā*, 48(1):26, February 2023.
- [25] Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. *Computational intelligence: a methodological introduction*. Texts in computer science. Springer, Cham, third edition edition, 2022.
- [26] Hairong Lin, Chunhua Wang, Fei Yu, Jingru Sun, Sichun Du, Zekun Deng, and Quanli Deng. A Review of Chaotic Systems Based on Memristive Hopfield Neural Networks. *Mathematics*, 11(6):1369, March 2023.
- [27] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*, 3:4, February 2020.
- [28] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: an introduction to dynamical systems*. Textbooks in mathematical sciences. Springer, New York, NY Heidelberg, repr. edition, 2010.

- [29] Hassan K. Khalil. *Nonlinear systems. Hauptbd.* Prentice Hall, Upper Saddle River, NJ, 3. ed edition, 2002.
- [30] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, March 1963.
- [31] Robert L. Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment.* CRC Press, 1 edition, May 2018.
- [32] Hernando Fernandez-Canque. *Analog electronics applications: fundamentals of design and analysis.* CRC Press, Boca Raton London New York, 2017.
- [33] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Numerical analysis.* Cengage Learning, Boston, MA, tenth edition edition, 2016. OCLC: ocn898154569.

Avance de Tesis II

Análisis dinámico e implementación electrónica

Periodo de trabajo

Del 4 de enero 2024 - 7 de junio de 2024

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

**Maestría en Ciencias de la Electrónica
Opción en Automatización**



Avance de Tesis II

Diseño de una red neuronal Hopfield en hardware embebido

Alumno: Diego Ruiz Sánchez de la Vega

**Directores de Tesis: Dra. Olga Guadalupe Félix Beltrán (FCE-BUAP)
Dr. Jesús Manuel Muñoz Pacheco (FCE-BUAP)**

Puebla, Puebla.

Mayo de 2024.

Resumen

Resumen general

En esta tesis se lleva a cabo el diseño e implementación de una red neuronal Hopfield de tres y cuatro neuronas en hardware embebido programable analógico, caracterizando la dinámica no lineal que a ella describe, así como también se realiza un estudio sobre las distintas funciones no lineales de activación, evaluando los efectos en el proceso de aprendizaje de la red neuronal Hopfield. Con este fin inicialmente se desarrolla una revisión bibliográfica sobre el estado del arte de las redes neuronales artificiales, en específico sobre la red neuronal Hopfield, sus antecedentes, principales aplicaciones, implementaciones tanto en circuitos analógicos como con dispositivos programables analógicos y digitales, así como las ecuaciones que describen el modelo de la red neuronal Hopfield. Posteriormente, se desarrolla un análisis exhaustivo sobre la dinámica no lineal del sistema, este incluye la evaluación del proceso de aprendizaje de la red neuronal, así como, el análisis en base a caos para finalmente implementar el modelo de la red neuronal Hopfield con distintas funciones de activación en un dispositivo analógico programable, el cual, requiere sus propias ecuaciones de diseño, las cuales también son desarrolladas. Finalmente, con todo el trabajo realizado, se elabora una metodología para la implementación de las redes neuronales Hopfield en las tarjetas analógicas embebidas utilizadas a lo largo de esta tesis.

Resumen Avance de Tesis I

En el Avance de Tesis I estudian los antecedentes de las redes neuronales artificiales, su comparación con las neuronas biológicas y se presenta el primer modelo matemático que a ellas describe, posteriormente, se presenta el modelo de las redes neuronales Hopfield (HNN) general y específico para 3 y 4 neuronas, dado que es el tema de interés en esta tesis. Simultáneamente, se realiza un estudio sobre la implementación de las HNN, ya sea con circuito embebidos o no, su respuesta y aplicaciones principales. Otro tema de interés en esta tesis son los sistemas dinámicos caóticos esto debido a que, dados ciertos parámetros, la red neuronal Hopfield presenta caos. Con respecto a este tema, una investigación es realizada con relación a los sistemas

dinámicos caóticos y a las HNN en modo caótico. La implementación en hardware embebido de redes neuronales es un tema de recurrente estudio en la comunidad científica, por lo que una investigación a fondo en el diseño en hardware programable de las HNN, resulta evidente en el cual se incluye, un estudio e investigación sobre los dispositivos analógicos programables FPAA's, dado que este circuito será el utilizado para el desarrollo de esta tesis. Finalmente, para el análisis del modelo de la red neuronal Hopfield, es necesario desarrollar herramientas matemáticas analíticas y numéricas, con lo que respecta a este avance, se presentan dos métodos para la integración numérica por medio de los métodos de Runge-Kutta de segundo y cuarto orden.

Resumen Avance de Tesis II

En el Avance de Tesis II se presentaron los modelos de las redes neuronales Hopfield de tres neuronas específicos que serán estudiados en este trabajo, para cada modelo se utiliza una función de activación distinta. Dos de estos modelos con comportamiento caótico no habían sido reportados en la literatura y son presentados en este trabajo, mientras que los otros tres sistemas caóticos ya habían sido reportados en [1], [2] y [3]. Para cada uno de estos modelos se realiza un análisis dinámico en donde se demuestra la existencia de atractores en los modelos y se desarrolla en análisis de estabilidad de Lyapunov para dar un criterio para la estabilidad asintótica del sistema. Posteriormente, se utilizan los diagramas de bifurcación y exponentes de Lyapunov como métricas para identificar las regiones caóticas de los modelos de redes neuronales Hopfield. Finalmente, se desarrollan las funciones de activación en base a los bloques que ofrece la distribuidora de las tarjetas analógicas para posteriormente implementar todos los modelos en las tarjetas analógicas programables, para esto se desarrollan las ecuaciones de diseño que representan los modelos de manera que se desarrollen los programas en AD2 con comportamiento caótico y registrar los resultados obtenidos.

Índice general

Resumen	II
Introducción	III
Objetivos	VIII
Cronograma de Actividades	X
Diagrama de bloques de la tesis	XII
1. Estado del arte	1
2. Marco Teórico	4
2.1. Redes Neuronales Artificiales	4
2.1.1. Redes Neuronales Hopfield	5
2.1.2. Funciones de activación	6
2.2. Sistemas dinámicos	8
2.2.1. Representación matemática de los sistemas dinámicos	8
2.2.2. Puntos de equilibrio	9
2.2.3. Sistemas dinámicos caóticos	10
2.2.4. Redes Neuronales Hopfield en modo caótico	11
2.3. Dispositivos programables	12
2.3.1. Hardware Analógico Embebido	12
2.4. Métodos numéricos para la solución de sistemas dinámicos	13
3. Análisis de estabilidad de la red neuronal Hopfield	17
3.1. Propiedades de una función de activación	18
3.2. Modelos propuestos	19
3.2.1. Modelo con función de activación tangente hiperbólica	19
3.2.2. Modelo obtenido con función de activación sigmoide	20

3.2.3. Modelo con función de activación signo	22
3.3. Modelos reportados en la literatura	23
3.3.1. Modelo con función ReLU	23
3.3.2. Modelo con función de activación PWL	24
3.4. Análisis en base a la divergencia del sistema dinámico	25
3.5. Análisis de estabilidad en el sentido de Lyapunov	36
3.6. Análisis de estabilidad alrededor de los puntos de equilibrio	39
4. Dinámica caótica	47
4.1. Diagramas de bifurcación	47
4.1.1. Diagramas de bifurcación en Python	48
4.1.2. Diagramas de bifurcación del modelo con función de activación tangente hiperbólica	51
4.1.3. Diagramas de bifurcación del modelo con función de activación sigmoide	52
4.1.4. Diagramas de bifurcación del modelo con función de activación ReLU .	52
4.1.5. Diagramas de bifurcación del modelo con función de activación PWL . .	55
4.1.6. Diagramas de bifurcación del modelo con función de activación signo . .	55
4.2. Exponentes de Lyapunov	55
5. Implementación de las redes neuronales Hopfield	68
5.1. Desarrollo de funciones de activación	68
5.1.1. Implementación de la función signo	68
5.1.2. Implementación de la función ReLU	70
5.1.3. Implementación de la función PWL	70
5.1.4. Implementación de la función tangente hiperbólica y sigmoide	72
5.2. Programación e implementación de las HNN's de 3 neuronas	72
Conclusiones	85
Anexos	87
5.3. Codigos en AnadigmDesigner2 para la implementación de las redes neuronales de 3 dimensiones	87
Bibliografía	95

Introducción

Antecedentes

El estudio del comportamiento del cerebro humano tiene profundas raíces en la historia de la ciencia y la medicina, desde la antigüedad grandes filósofos especulaban sobre su funcionamiento, los antiguos griegos plantearon ideas sobre “maquinas pensantes” [4]. El siglo XIX marcó un avance importante con respecto al estudio de la biología; en 1839 Schleiden y Schwann formulan la Teoría Celular, que establece que todos los seres vivos están formados por células como unidades fundamentales [5], esto por supuesto incluye al cerebro.

Una de las células básicas que componen al sistema nervioso es la neurona, las neuronas procesan y transmiten información en forma de señales eléctricas, como fue propuesto por el científico español Santiago Ramón y Cajal [6], en su trabajo sobre la transmisión neuronal, en el cual se considera a la neurona como un interruptor que se “dispara” dependiendo de las señales de otras neuronas, estas señales pueden ser inhibitorias si tienden a impedir que la neurona se active o excitatorias si provocan que la neurona se active, con estas señales la neurona realiza suma ponderada y, dependiendo del valor de esta suma, se dice que la neurona está “activa” o no.

La neurona es una de las unidades fundamentales del sistema nervioso y se divide en cuatro partes importantes: el núcleo, el cuerpo de la célula, la dendrita y el axón (véase Figura 1). El cuerpo de la célula contiene al núcleo, el cual se encarga de producir la energía de la neurona, la entrada de la neurona que trata con las señales de las demás neuronas es denominada como dendrita, mientras que la salida es denominada como axón; a la conexión entre las neuronas se le conoce como sinapsis.

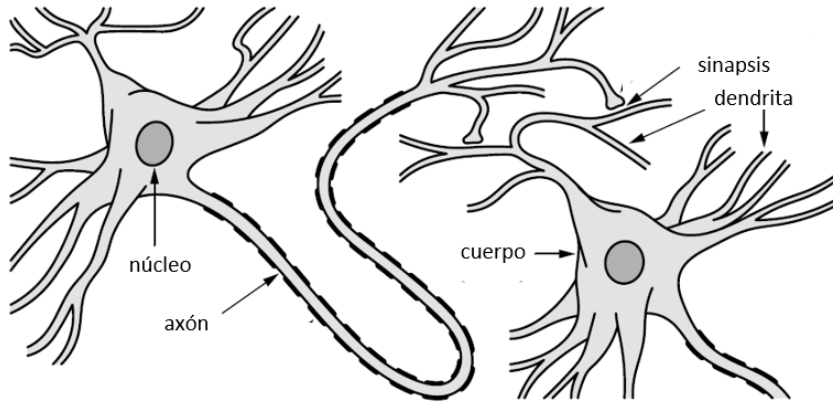


Figura 1: Neurona biológica.

Inspirados por el comportamiento de las neuronas biológicas, los investigadores y científicos han desarrollado las denominadas redes neuronales artificiales (ANN, por sus siglas en inglés), las cuales son dispositivos que simulan el comportamiento de las redes neuronales biológicas. El primer modelo de ANN fue el desarrollado por Warren McCulloch y Walter Pitts en 1943 [7], quienes proponen un modelo que toma múltiples entradas binarias ponderadas, suma estas entradas y aplica una función de umbral (véase Figura 2). Si la suma supera el umbral, la neurona emite una señal de salida. Este enfoque permitió realizar cálculos lógicos y de decisión basados en la entrada recibida, semejante al comportamiento de las neuronas biológicas.

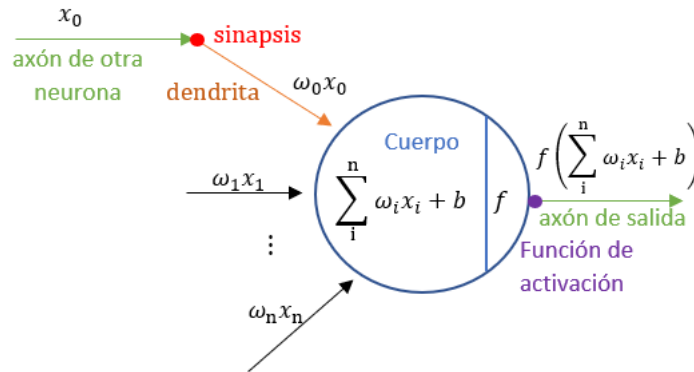


Figura 2: Modelo matemático de una neurona artificial.

En 1958, Frank Rosenblatt, padre del perceptrón [8], consiguió implementar la primera máquina capaz de aprender, distinguir y separar cartas marcadas utilizando el modelo matemático McCulloch-Pitts. El modelo del perceptrón, contiene solo una capa de entradas y una única neurona de salida, donde cada conexión entre las neuronas tiene un valor específico denominado “peso”, el cálculo del valor de la neurona de salida se realiza de la siguiente forma: se realiza el producto entre la entrada con su respectivo peso, estos valores se suman y, final-

mente, el valor calculado se introduce como argumento en una función denominada “función de activación”, para el perceptrón la función utilizada es la función *signo*, la cual es adecuada para una clasificación [9].

Si bien, el trabajo de Rosenblatt no fue bien recibido por la comunidad científica debido a sus limitantes, años después se desarrollaron modelos que solucionaban las deficiencias del perceptrón, algunos de ellos son: las redes neuronales convoluciones, redes neuronales recurrentes, perceptrón multicapa, entre otros [10]. Tiempo después al trabajo desarrollado por McCulloch-Pitts, John Hopfield propone un nuevo modelo para las ANN, denominado como red neuronal Hopfield o HNN por sus siglas en inglés, esta es una red de tipo recurrente y completamente conectada, esto es, a diferencia de los modelos de McCulloch-Pitts, las neuronas del modelo de la HNN están conectadas con todas las demás, además de tener cierta dependencia con su estado anterior, este tipo de ANN se utiliza generalmente para describir algunos procesos cerebrales, como por ejemplo, el proceso de aprendizaje y el concepto de memoria asociativa, la ecuación que generaliza cada neurona de la HNN viene dada por [11]

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^n \omega_{ij} \tanh(x_j) + I_i,$$

donde x_i es la entrada de la neurona, i es el número de neurona, $\tanh(\cdot)$ es la función de activación, w_{ij} es el peso entre las entradas de la neurona, c es un parámetro constante y finalmente la I_i es un factor de sesgo o bias.

En la Figura 3 se ilustra el diagrama a bloques de una neurona Hopfield, la cual podemos asociar a la función de propagación con la sumatoria de todas las sinapsis que posee la neurona, mientras que el bloque $f(x_i)$ representa la función de activación. Para las redes neuronales Hopfield es utilizada la función tangente hiperbólica $\tanh(\cdot)$ y, finalmente, existe un bloque de derivada, el cual nos permite solucionar la ecuación diferencial para adquirir el valor actual de la neurona [12, 13].

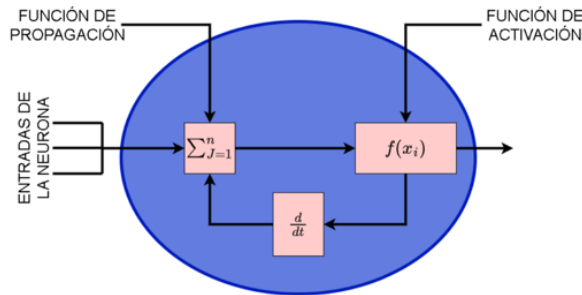


Figura 3: Diagrama en bloques de una red neuronal Hopfield.

En este escenario, debido a las capacidades de aprendizaje de las neuronas artificiales, es requerido un diseño electrónico en cual puede ser usado para aplicaciones de máquinas inteligentes, las cuales requieren implementaciones en hardware optimizadas, es decir, rápidas, baja complejidad computacional y con bajo consumo de potencia. Note que la HNN, al utilizar una menor cantidad de neuronas, requiere un menor consumo de potencia electrónica o computacional para su funcionamiento, además, las HNN son muy utilizadas debido a que, comparada con otros modelos de ANN, no existe una mejor red neuronal para la identificación de patrones.

Planteamiento del problema

En esta investigación se tiene como objetivo realizar una metodología para el desarrollo y diseño en la implementación de redes neuronales Hopfield en dispositivos programables analógicos, además de realizar un análisis exhaustivo con respecto a la dinámica del sistema y evaluar los efectos en el proceso de aprendizaje usando distintas funciones de activación.

Infraestructura

- Laboratorio de Circuitos y Sistemas Caóticos de Orden Fraccional, 1FCE4/201.
- Software especializado.
- Laptops y estaciones de trabajo.
- Tarjetas electrónicas.

Justificación

Las redes neuronales son sistemas que nos permiten dar solución a problemáticas en donde se requiere de dispositivos inteligentes y adaptables, en particular, las redes neuronales Hopfield nos permiten, cifrar datos, reconocimiento de imágenes y, principalmente, la detección de patrones, entre otros. En específico, la detección de patrones es crucial para lograr, en un futuro, la comunicación entre dispositivos inteligentes capaces de manejarse de manera autónoma sin la necesidad de la supervisión humana.

Actualmente, no hay red neuronal que sea capaz de almacenar una gran cantidad de patrones como lo hace la red neuronal Hopfield, gracias a que sus neuronas son capaces de almacenar más información que cualquier otra red neuronal existente [14].

Si bien la implementación de este tipo de red neuronal ya ha sido desarrollada por trabajos pasados, el desarrollo de una metodología para la implementación en circuitos electrónicos de

las redes neuronales Hopfield no es del todo clara y este es uno de los temas que esta tesis busca resolver.

El modelo de la red neuronal Hopfield está descrito por un sistema de ecuaciones diferenciales no lineales, su estructura, dados ciertos parámetros, presenta un comportamiento caótico, en otras palabras, bajo parámetros adecuados la implementación de una red neuronal Hopfield nos permite obtener un sistema caótico y, con él todas las propiedades que ellos nos ofrecen. Considerando lo anterior, en este trabajo se realiza un análisis y estudio de la red neuronal en modo caótico.

Dentro del modelo que a las redes neuronales describe, las funciones de activación son fundamentales, a través de los años distintas funciones de activación han sido desarrolladas, cada función de activación da distintas propiedades dinámicas a la red neuronal, así como también, afecta directamente el proceso de aprendizaje de dicha red neuronal, y este análisis es otro de los objetivos primarios de este trabajo.

Estructura del documento

El presente documento se encuentra seccionado por capítulos, donde cada uno aborda diferentes aspectos relativos al problema de investigación. En un principio se abordan los antecedentes, conceptos básicos, los objetivos de la tesis, el cronograma y el diagrama de bloques de la tesis, que se considera como la metodología de la misma. En el primer capítulo se aborda el estado del arte relacionado a las redes neuronales Hopfield, sistemas caóticos y su implementación en circuitos electrónicos. En el segundo capítulo se desarrolla en detalle los temas de interés antes mencionados. En el tercer capítulo se presentan los modelos con los que se trabajara en la Tesis, todos presentan una función de activación distinta en busca de abordar la mayor cantidad de fenómenos dinámicos. Al momento de presentar cada modelo junto con su función de activación también se dan algunas características o propiedades de interés sobre la función. En uno de los modelos se desarrolla una transformación para llevar de un modelo conocido a otro con una función de activación distinta. Así mismo se desarrolla un análisis dinámico en relación a la estabilidad del modelo de la red neuronal Hopfield. En el capítulo cuatro se aborda la dinámica caótica de los modelos de interés usando métricas para la identificación de zonas caóticas, como lo son los diagramas de bifurcación y exponentes de Lyapunov. La implementación de las redes neuronales Hopfield se muestra en el capítulo tres. Finalmente, se presentan las conclusiones del presente avance junto con un apartado de anexos donde se detallan los anexos con código utilizados para implementación de las redes neuronales Hopfield.

Objetivos

Los objetivos planteados en este tema de tesis, objetivo general y objetivos particulares, se enuncian a continuación, junto a estos últimos un desglose más específico de dicho objetivo particular.

Objetivo general

Determinar los efectos de diferentes funciones de activación en el proceso de aprendizaje de una red neuronal Hopfield en modo caótico en un entorno de circuitos electrónicos.

Objetivos particulares

- Establecer la estructura básica de una red neuronal Hopfield de tres y cuatro neuronas.
 - Determinar los modelos matemáticos para una HNN de 3 y 4 neuronas.
 - Realizar la integración numérica de los modelos de 3 y 4 neuronas de una HNN.
- Caracterizar la dinámica no lineal de la red neuronal Hopfield usando técnicas analíticas y numéricas.
 - Realizar el análisis dinámico de los modelos matemáticos de las HNN para distintas funciones de activación.
 - Realizar el análisis desde un punto de vista caótico de los modelos de la red neuronal Hopfield.
- Evaluar la respuesta en el aprendizaje de la red neuronal Hopfield con diferentes funciones no lineales de activación.
 - Evaluar los efectos en el proceso de aprendizaje de distintas funciones de activación en los modelos de las redes neuronales Hopfield de 3 y 4 neuronas.

- Proponer las ecuaciones de diseño para implementar la red neuronal Hopfield en un sistema analógico embebido programable de prototipado rápido.
 - Simulación del comportamiento de la red neuronal Hopfield en el entorno de programación de AnadigmDesigner2.
 - Desarrollar una metodología para el diseño e implementación de una red neuronal Hopfield con 3 y 4 neuronas.
- Implementar los modelos desarrollados de la red neuronal Hopfield para 3 y 4 neuronas en la tarjeta analógica embebida.
 - Evaluar la dinámica obtenida.
 - Evaluar los efectos en el proceso de aprendizaje de los modelos implementados y comparar resultados con las simulaciones desarrolladas.

Cronograma de Actividades

En la Tabla 1 se presentan las actividades propuesta para el desarrollo esta tesis, titulada *Diseño de una red neuronal Hopfield en hardware embebido*. Se remarcan en color rosa los objetivos abordados en Tesis II. Algunas de las actividades aún están en el proceso de ser realizada, por lo que, la cuantificación de las actividades considera sólo un porcentaje de la actividad realizada y es presentado en la columna de “Avance realizado” junto con un porcentaje equivalente al valor en porcentaje de la actividad del total de la tesis.

Actividades	Tesis I	Tesis II	Tesis III	Avance realizado
Revisión bibliográfica y estado del arte.	X			100 % (10 %)
Planteamiento de los modelos matemáticos de las redes neuronales Hopfield de 3 y 4 neuronas.	X	X		80 % (5 %)
Simulación de las redes neuronales Hopfield e integración por métodos numéricos.	X	X		80 % (10 %)
Análisis de la dinámica de los modelos de las redes neuronales Hopfield de 3 y 4 neuronas mediante técnicas numéricas y analíticas.		X	X	60 % (15 %)
Evaluación del proceso de aprendizaje caótico de las HNN de 3 y 4 neuronas con distintas funciones de activación.		X	X	50 % (15 %)
Desarrollo de las ecuaciones para la implementación de las redes neuronales Hopfield en la tarjeta embebida analógica.		X	X	50 % (10 %)
Implementación de la red neuronal Hopfield con distintas funciones de activación.		X	X	50 % (10 %)
Desarrollo de una metodología para el diseño e implementación de la red neuronal Hopfield usando tarjetas embebidas analógicas de prototipado rápido.		X	X	10 % (10 %)
Publicación de un artículo.			X	0 % (5 %)
Análisis y comparación de resultados			X	0 % (5 %)
Escritura de la tesis.	X	X	X	50 % (5 %)
Porcentaje acumulado				16.5 %

Tabla 1: Cronograma de actividades.

Gráfica del avance global de la tesis

A continuación, en la Figura 4 se presenta el avance global que se tiene de este trabajo de tesis, el cálculo de este está basado en la ponderación de cada una de las actividades descritas en la Tabla 1 y del porcentaje realizado de cada una de ellas. Considerando lo anterior, la suma ponderada de las actividades descritas es equivalente al **52 %**.

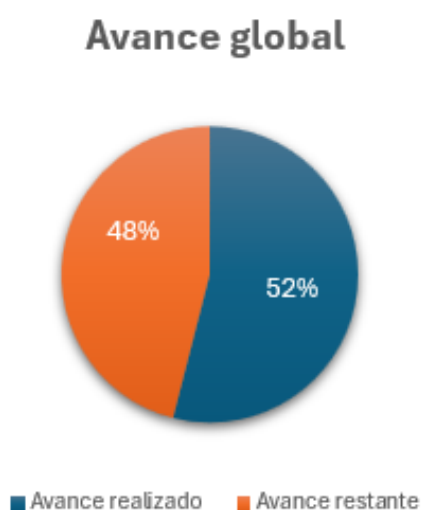


Figura 4: Gráfica con el avance global.

Diagrama de bloques de la tesis

A continuación se presenta, en la Figura 5, la propuesta general del proceso que se lleva a cabo durante el desarrollo de esta investigación. En esta se indica el periodo de inicio y desarrollo de cada actividad en un color diferente, considerando que marca el inicio y no necesariamente el final de cada actividad. El color verde indica el trabajo realizado el primer semestre, azul y morado representan los trabajos realizados el segundo y tercero.

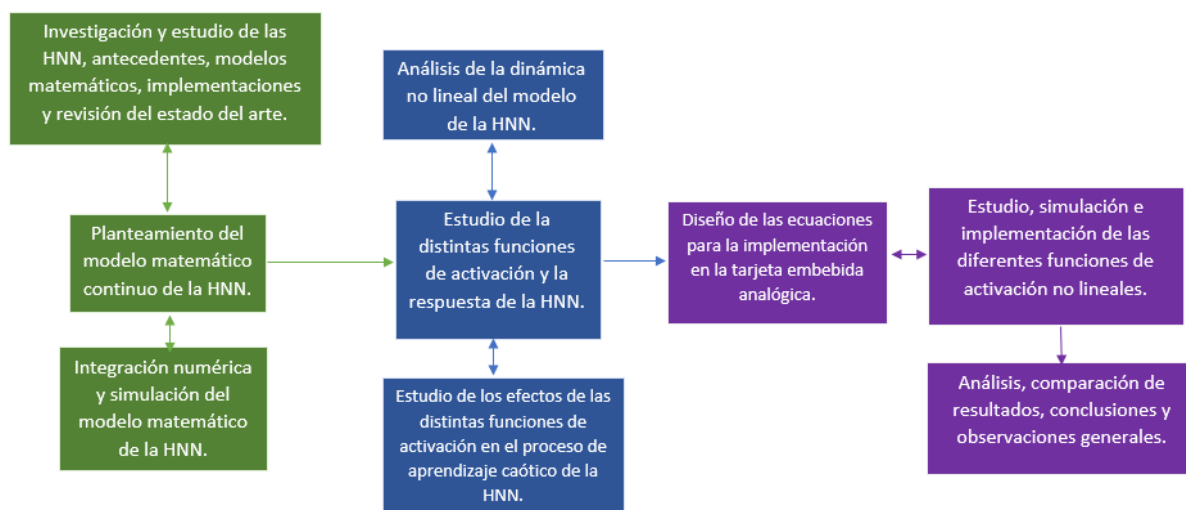


Figura 5: Resumen del trabajo por realizar en la tesis dividido por colores en diagrama de bloques.

Bloques para el primer semestre

Con lo que respecta al trabajo realizado con base en los bloques coloreados por color verde, correspondientes a las actividades desarrolladas y en desarrollo del primer semestre comprendido en agosto y diciembre del año 2023, se desglosaron de forma que sea posible describirlo más a fondo.

- **Investigación y estudio de las HNN, antecedentes, modelos matemáticos, implementaciones y revisión del estado del arte:** Esta actividad consiste en investigar y redactar los sucesos de mayor relevancia con respecto a los temas de interés de esta

tesis, es decir, de las redes neuronales artificiales, las HNN y su implementación. Posteriormente, se realiza el estado del arte de este trabajo, el cual consiste en sintetizar los avances más recientes desarrollados por la comunidad científica y hacer un pequeño repaso del impacto de estos.

- **Planteamiento del modelo matemático continuo de la HNN:** Junto con la investigación sobre las redes neuronales de Hopfield es posible hallar el modelo matemático con el cual se caracteriza en las referencias [11, 14–16], entre otras, por lo que, ayudado por esta investigación, es posible plantear un primer modelo para las HNN de 3 y 4 neuronas, en la sección 2.1.1 se presenta el modelo general en cuestión que expresa tanto el modelo de 3 como de 4 neuronas.
- **Integración numérica y simulación del modelo matemático de la HNN:** Un primer acercamiento al modelo matemático fue desarrollado al simular una de las ecuaciones que describen a la red neuronal de Hopfield, las ecuaciones y parámetros elegidos son los correspondientes al trabajo [15], cuya respuesta es caótica. La simulación fue implementada en el lenguaje de programación de Python usando el método de integración numérica de Runge-Kutta de cuarto orden.

Bloques para el segundo semestre

En el segundo semestre se trabajaron los bloques de color azul y morado, es decir, el trabajo no se realizó de manera lineal, en general se decidió trabajar con las redes neuronales Hopfield con tres neuronas, lo que incluyó su análisis e implementación con distintas funciones de activación. Un resumen del trabajo realizado con respecto a cada uno de los bloques propuestos es desglosado en los siguientes puntos.

- **Análisis de la dinámica no lineal del modelo de la HNN:** El análisis de la red neuronal Hopfield realizado se enfocó en demostrar que los modelos presentados son disipativos y por lo tanto, contienen atractores, también en la demostración del hecho que las redes neuronales presentadas están acotadas en alguna región del espacio fásico, finalmente, se desarrolla un análisis alrededor de los puntos de equilibrio (donde se encuentran los atractores), linealizando el sistema y caracterizando su comportamiento alrededor de ellos.
- **Estudio de las distintas funciones de activación y la respuesta de la HNN:** En la literatura existen una gran cantidad de funciones de activación para el modelo de las HNN, en este trabajo se exploró la aplicación de 5 distintas funciones de activación, cada una fue analizada desde un punto de vista de estabilidad, para finalmente ser implementadas en una tarjeta analógica programable.

- **Estudio de los efectos de distintas funciones de activación en el proceso de aprendizaje caótico de la HNN:** Algunas de las funciones estudiadas no tenían una propuesta para el modelo de HNN con comportamiento caótico, este el trabajo realizado a lo largo de este semestre de propusieron dos nuevos modelos así como una transformación que lleva modelos con función de activación $\tanh(\cdot)$ a modelos con funciones de activación $\sigma(\cdot)$. Para cada modelo se utilizaron dos métricas distintas para la detección de caos.
- **Diseño de las ecuaciones para la implementación en la tarjeta embebida analógica:** La programación en la tarjeta analógica es en base a un conjunto de bloques denominados CAM's, cada CAM tiene su equivalente con respuesta en frecuencia, tomando esto en consideración, se describió el modelo de redes neuronales Hopfield en su respuesta en frecuencia usando la transformada de Laplace, este resultado fue general y será utilizado para el desarrollo de las ecuaciones para implementación de las HNN de neuronas. Por su lado, los modelos para las HNN de 3 neuronas fueron implementados y coinciden con las simulaciones realizadas, este resultado es obtenido por inspección visual.
- **Estudio, simulación e implementación de las diferentes funciones de activación no lineales:** AnadigDesigner2 es el software que nos permite programar la tarjeta analógica y cuenta con una gran cantidad de bloques programables para la descripción de comportamientos lineales y no lineales, en base a uno o un conjunto de bloques es posible describir el comportamiento de las funciones de activación usadas, los resultados obtenidos en la implementación son mostrados y se realiza una comparación usando una métrica para el error.

Capítulo 1

Estado del arte

Como parte del estado del arte se realizo una revisión e investigación sobre los temas de mayor interés en este trabajo, esto es, redes neuronales Hopfield (HNN), sistemas dinámicos caóticos y hardware analógico embebido.

Revisión bibliográfica

En lo referente a las redes neuronales se realizo una investigación sobre los antecedentes, modelos matemáticos que a ellas describen e implementaciones sobre las redes neuronales artificiales, con un enfoque en las redes neuronales Hopfield, cuya implementación de 3 y 4 neuronas por medio de circuitos analógicos y digitales fue estudiada.

Por otro lado, y al ser un tema recurrente en la tesis, se realizo un estudio de los sistemas dinámicos, esto debido a que el modelo de la HNN es un sistema dinámico como es evidente por su modelo matemático [14], es importante considerar que, el ser un sistema dinámico no lineal, dados ciertos parámetros una HNN puede presentar un comportamiento caótico [15,17,18], con lo que también se realiza un estudio de los sistemas dinámicos caóticos, esto es, la investigación para definir lo que es un sistema dinámico caótico junto con sus características.

Actualidad de las HNN

Desde su creación por el Dr. Jonh J. Hopfield [11], las redes neuronales Hopfield han atraído a la comunidad científica por sus cimientos matemáticos, la aplicación de esta red neuronal está relacionada a la memoria asociativa y al reconocimiento de patrones [13]. Sin embargo, y como es demostrado por distintos trabajos de investigación, su aplicación va mas allá de ello y puede resolver problemas que usualmente son resueltos por otros modelos de redes neuronales. Además, debido a las características de su modelo matemático, su comportamiento puede ser caótico y con esto, conlleva todas las aplicaciones que los sistemas dinámicos caóticos tienen, como lo son los generadores de aleatoriedad, la criptografía, modelado de sistemas, entre otros.

Implementación de las HNN

La implementación de las redes neuronales Hopfield no es ajeno a los circuitos eléctricos, en la actualidad la implementación de un circuito no lineal es facilitada gracias a los dispositivos programables como lo son los FPGA's, múltiples trabajos han tomado esta aproximación para la implementación de una red neuronal Hopfield [15, 17, 19]. Por otro lado, también se ha desarrollado una electrónica analógica para la implementación de una red neuronal Hopfield [20–23], dicha electrónica esta basada en dispositivos activos y resistivos, la estructura presentada en estos trabajos es muy similar, sin embargo, en 2017 Bao et al. [19] desarrollo un modelo para una red neuronal Hopfield de 3 neuronas en base a memristores y partir de este trabajo múltiples desarrollos han sido llevados a cabo [18]. La implementación en un dispositivo digital implica discretizar el modelo matemático que se tiene del sistema continuo, por lo que el comportamiento obtenido será una aproximación a modelo que describe la red neuronal, mientras que la implementación en dispositivos analógicos, aunque limitada en otros aspectos, nos permite obtener el comportamiento exacto de nuestro modelo matemático.

Para gran parte de las implementaciones de las redes neuronales Hopfield, las metodologías para el diseño del circuito electrónico descritas con anterioridad son utilizadas, sin embargo, existen dispositivos analógicos programables denominados FPAA, con los cuales es posible desarrollar una metodología para la implementación de las HNN [24, 25], estos nos permiten tener la flexibilidad que nos ofrece un dispositivo programable sin la necesidad de discretizar el modelo de las HNN.

HNN en modo caótico

Una de las grandes razones por la cual las redes neuronales Hopfield resultan de interés en la comunidad científica es su riqueza dinámica a pesar de su simple estructura, su no linealidad nos permite modelar sistemas muy complejos, recientes trabajos relacionados a la construcción de una red neuronal Hopfield con base en Memristores han mostrado que estos pueden presentar caos [19, 20]. A partir de estos trabajos se han desarrollado otras estructuras y parámetros sobre las HNN en las cuales se han observado, no solo comportamientos caóticos, sino hipercaóticos [26], esto último considerando una red neuronal de cuatro neuronas.

Almacenamiento y reconocimiento de patrones

Las HNN son especialmente útiles en tareas de almacenamiento y recuperación de patrones asociativos. Pueden recordar y recuperar patrones incluso cuando la entrada está incompleta o contaminada con ruido [27]. La dinámica de las HNN se rige por una función de energía y el aprendizaje de patrones se realiza ajustando los pesos para minimizar esta energía. Sin

embargo, la estabilidad dinámica puede ser un desafío, especialmente cuando se trata de grandes conjuntos de datos o patrones complejos.

A pesar de estas fortalezas, las HNN también tienen limitaciones. No son ideales para manejar patrones complejos o grandes conjuntos de datos como lo hacen otras arquitecturas más modernas, como las redes neuronales convolucionales (CNN, por sus siglas en inglés) o las redes neuronales recurrentes (RNN, por sus siglas en inglés). Además, el modelo original de Hopfield puede tener problemas con la capacidad de almacenamiento y la recuperación de patrones en grandes dimensiones.

Modelado biológico

Las HNN han sido utilizadas para modelar la memoria asociativa, un aspecto de la memoria biológica, esta es capaz de reconocer patrones enteros usando solo partes parciales o dañadas por el ruido de ese mismo patrón [27]. La capacidad de las HNN para almacenar y recuperar patrones asociativos puede interpretarse como un intento de emular procesos de asociación en el cerebro [26], es importante tener en cuenta que estas redes son una simplificación de los sistemas biológicos. Aunque se inspiran en algunos aspectos de la neurobiología, no capturan la complejidad y la dinámica real de las redes neuronales biológicas.

Funciones de activación en las HNN

El propósito de las funciones de activación son las de determinar cuando una neurona se “dispara” o “enciende”, además, su no linealidad permite modelar comportamientos mas complejos. En la literatura se puede hallar una lista de diferentes funciones de activación [28], cada una ofrece distintos comportamientos y respuestas de la red neuronal, como es el caso de la identificación de imágenes [13]. Resulta de interés en esta tesis ser capaz de diferenciar entre qué funciones de activación son apropiadas para alguna aplicación en específico, así como la respuesta que cada una de estas tiene una vez implementada en un dispositivo programable.

Capítulo 2

Marco Teórico

Para el desarrollo de esta tesis es primero definir algunos conceptos de relevancia que se tocan en este trabajo, como lo son los sistemas dinámicos y caóticos, redes neuronales artificiales (Hopfield específicamente), hardware programable, entre otros. Con este fin se desarrollo el presente marco teórico.

2.1. Redes Neuronales Artificiales

Unas de las tecnologías de más renombres desarrolladas en los últimos años por la comunidad científica son las redes neuronales artificiales (ANN, por sus siglas en inglés). Estas toman el comportamiento de las redes neuronales biológicas observadas en el cerebro humano como base para el desarrollo de un modelo matemático que emule dicho comportamiento. El primer modelo de las redes neuronales biológicas fue el desarrollado por Warren McCulloch y Walter Pitts [7], en el cual las neuronas se describen como interruptores, cuyo estado es dependiente del de las demás neuronas. A la salida de cada neurona se denominado como axón y a su entrada como “dendrita”, a este sistema de conexiones entre neuronas se le conoce como sinapsis. Como base de esta descripción, el psicólogo Frank Rosenblatt desarrolló el primer modelo de las ANN conocido como perceptrón, en este, cada conexión entre las neuronas, es decir, la sinapsis, tiene un valor específico denominado como peso, consideraba neuronas de entrada y una sola neurona de salida, el modelo es presentado en la Figura 2.1, en la cual se observa tanto la representación matemática como en un diagrama. Tras el trabajo de Rosenblatt, más modelos de ANN fueron desarrollados, cada uno con distintas propiedades, aún sin embargo, los conceptos cimentados por este modelos se conservan y siguen siendo usados en modelos nuevos de las ANN. En modelos más recientes de ANN se requiere de una mayor cantidad de neuronas para la resolución de la mayoría de problemas. De acuerdo a la estructura de una red neuronal, estas se pueden separar en dos partes fundamentales, las redes neuronales, si el

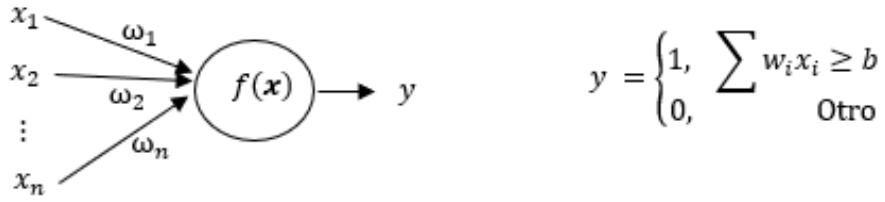


Figura 2.1: Modelo del perceptron.

modelo que describe la estructura de una red neuronal no contiene ciclos o retroalimentaciones, es denominada como red neuronal prealimentada o “feed-forward network”, de lo contrario, se le denomina como red recurrente o *recurrent network* [28].

2.1.1. Redes Neuronales Hopfield

Un tipo especial de red neuronal recurrente es la red neuronal Hopfield, desarrolladas por John Hopfield en 1980 [11], estas redes destacan por su capacidad única para el almacenamiento y la recuperación de patrones asociativos, emulando de alguna manera el funcionamiento de la memoria humana [27]. Esta estructura permite a la red aprender patrones y recuperar información basándose en la activación y desactivación de sus neuronas. Cada conexión entre neuronas, ponderada por un valor sináptico, contribuye a la capacidad de la red para almacenar y recordar patrones. Por otro lado, las HNN son ideales para modelar la dinámica del cerebro [11], además de su dinámica no lineal. En general una red neuronal Hopfield de 4 neuronas puede ser representada por el diagrama de la Figura 2.2. Debido a su estructura las HNN son generalmente conocidas como sistemas dinámicos retroalimentados el cual produce múltiples patrones [27], dependiendo de la condición inicial de la HNN esta puede converger a un punto específico, estos puntos suelen ser denominados como “atractores”. El diagrama mostrado por la Figura 2.2 puede ser generalizado para n neuronas considerando la misma forma en la sinapsis. Para una HNN de n neuronas, el modelo matemático continuo es el expresado por el sistema de ecuaciones no diferenciales adimensional dado por [29]:

$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{f}(\mathbf{x}) + \mathbf{b}, \quad (2.1)$$

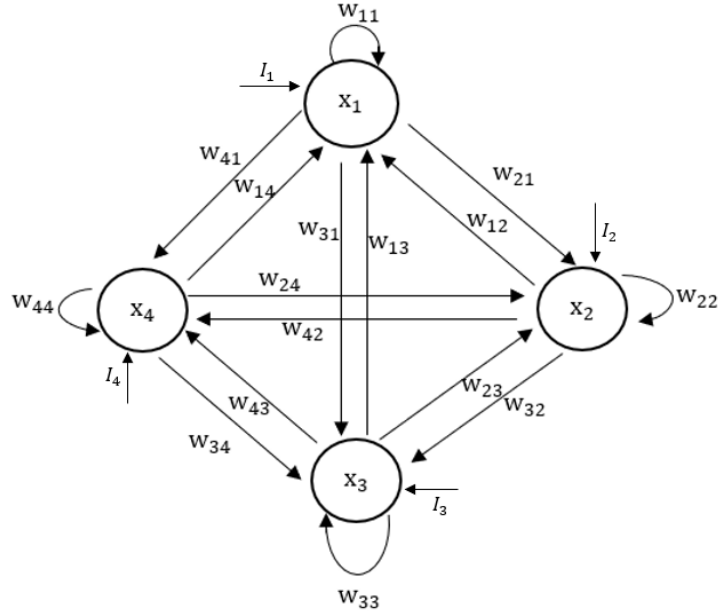


Figura 2.2: Diagrama de una red neuronal Hopfield de 4 neuronas.

donde

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}, W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nj} & \dots & w_{nn} \end{bmatrix}, C = \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix}. \quad (2.2)$$

x_i denota la i -ésima neurona, b_i representa un bias o un valor externo sobre la red neuronal, W es la matriz de pesos que representa la sinapsis del modelo y finalmente $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$ representa la función de activación no lineal aplicada en cada una de las variables del modelo y C una matriz diagonal. En específico, esta tesis está interesada en estudiar los modelos de 3 y 4 neuronas de una red neuronal Hopfield.

2.1.2. Funciones de activación

En cada modelo estudiado se han mencionado las denominadas funciones de activación, esta cumplen con distintos propósitos en los que se incluye, principalmente, la introducción de la no linealidad a la red neuronal permitiendo modelar sistemas más complejas, capacidad de aprendizaje, ajuste a la salida de la red neuronal, la continuidad que permite la interpretación

del estado de la neurona en todo momento, entre otros. Algunos modelos sin embargo, no utilizan exclusivamente funciones de activación sino otro tipo de funciones, dependiendo del modelo utilizado cada neurona utiliza una de tres tipos distintos de funciones [28]:

- Función de entrada.
- Función de activación.
- Función de salida.

En lo que respecta a las redes neuronales Hopfield, estas son redes especiales cuyas neuronas no utilizan funciones de entrada ni de salida, solo de activación, la entrada de esta red son los valores con los que se inicia cada neurona, es decir, las condiciones iniciales introducidas el sistema dinamico que a estas describe. Por esta razón este trabajo se limita al estudio de las funciones de activación. Las principales funciones de activación [30] usadas por cualquier red neuronal se presentan la Tabla 2.1.

Tabla 2.1: Funciones de activación de las ANN más frecuentes en la literatura.

Función de activación	$\phi(x)$	$\phi'(x)$	Rango
Tangente hiperbólica	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - \phi^2(x)$	$(-1, 1)$
Sigmoid	$S(x) = \frac{1}{1 + e^{-x}}$	$\phi(x)(1 - \phi(x))$	$(0, 1)$
ReLU	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$[0, \infty)$
Función Heaviside	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$\delta(x)$	$[0, 1]$
Función signo	$R(x) = \begin{cases} -1 & \text{para } x < 0 \\ 0 & \text{para } x = 0 \\ 1 & \text{para } x > 0 \end{cases}$	$2\delta(x)$	$[-1, 1]$
Softmax	$y_i = \frac{e^{x_j}}{\sum_j^n e^{x_j}}$	$\frac{\partial y_i}{\partial j} = y_i(\delta_{ij} - y_i)$	$(0, 1)$

La función de activación utilizada, al estar directamente implicada en el modelo matemático de la ANN, impacta directamente en la dinámica de la red neuronal.

Una de las propiedades más atractivas de las redes neuronales (artificiales) es la posibilidad de entrenarlas para ciertas tareas con la ayuda de datos de ejemplo. En general entrenar una red neuronal consiste en adaptar los pesos de conexión y posiblemente otros parámetros (como bias) de manera que se optimice cierto criterio. Dependiendo del tipo de datos de entrenamiento y del criterio a optimizar, podemos distinguir dos tareas fundamentales de aprendizaje [28]:

aprendizaje supervisado, en el cual se asigna una salida deseada a un conjunto de entrada, y aprendizaje no supervisado en el cual no se proporcionan salidas deseadas.

En lo que se refiere a una red neuronal Hopfield, su entrenamiento se refiere a diseñar un atractor o atractores para un conjunto específico de condiciones iniciales [27]. Al conjunto de estos atractores se les conoce como memoria, mientras que al conjunto de condiciones iniciales que son atraídas por un atractor se les conoce como “cuenca de atracción”, a los valores que toman las neuronas una vez que estas convergieron a su valor final se le denomina como “patrón”. Considerando lo anterior, una red neuronal Hopfield puede ser visto como un dispositivo capaz de almacenar y recuperar información relacionada entre si. Con el paso del tiempo distintos investigadores han desarrollado algoritmo para el entrenamiento de HNN y cuyo estudio sera realizado a lo largo del desarrollo de esta tesis.

2.2. Sistemas dinámicos

La gran parte de los sistemas estudiados por la física y la matemática tienen un comportamiento variante en el tiempo, es decir, no son estáticos, a este tipo de sistemas se les denomina como sistemas dinámicos y su estudio se remonta a la invención del calculo diferencial desarrollado por Sir Isaac Newton (1643-1727) y por Gottfried Wilhelm Freiherr von Leibniz (1646-1716). Newton, tras la invención del calculo, intentó describir el comportamiento de los planetas alrededor del Sol usando ecuaciones diferenciales [31]. Newton fue capaz de resolver el problemas de los dos cuerpos, en donde describió el movimiento de la tierra alrededor del sol, tras esta hazaña, el estudio de los sistemas dinámicos se ha aplicado a diversas disciplinas dentro y fuera de la física, permitiendo tener un sistema de modelos capaces de describir el comportamiento de un sistema a lo largo de un periodo de tiempo.

2.2.1. Representación matemática de los sistemas dinámicos

Debido a su naturaleza, los sistemas dinámicos son descritos por mapas iterativos (también conocidas como ecuaciones de diferencias) o ecuaciones diferenciales. En este trabajo estamos interesados solo en su descripción por ecuaciones diferenciales. Consideremos un sistema de n ecuaciones diferenciales, este se expresa como sigue:

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ \dot{x}_2 &= f_2(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p),\end{aligned}\tag{2.3}$$

donde t es la variable que representa el tiempo, $u_1, u_2, \dots, u_p$ representan la entrada al sistema, $x_1, x_2, \dots, x_n$ las variables del sistema y $\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n$ de derivada con respecto al tiempo de las variables, al sistema anterior se le denomina de n -ésimo orden o de n dimensiones. Consideremos ahora que se conoce el estado inicial de las variables en el sistema de ecuaciones 2.3, esto es, se tiene un vector $\mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^n$ que representa el estado del sistema para un tiempo inicial t_0 , a la solución de este tipo de problemas se les dice *problemas de valor inicial* [32], mientras que al estado inicial del sistema se le dice *condición inicial*. Por otro lado, de forma reducida la expresión 2.3 puede ser escrita usando vectores, de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (2.4)$$

donde $\mathbf{f}(t, \mathbf{x}, \mathbf{u}) = [f_1(t, \mathbf{x}, \mathbf{u}), \dots, f_n(t, \mathbf{x}, \mathbf{u})]^T$, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $\mathbf{u} = [u_1, u_2, \dots, u_p]^T$ y $\mathbf{x}(t_0) = [x_1(t_0), x_2(t_0), \dots, x_n(t_0)]^T$. En las expresiones (2.3) y (2.4), el tiempo t está expresado explícitamente, en algunos sistemas dinámicos este no es el caso y no dependen explícitamente ni del tiempo ni de la entrada, a este tipo de sistemas se les denomina como sistemas dinámicos autónomos y están expresados por

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (2.5)$$

su estudio es de gran relevancia en esta tesis al representar el modelo matemático de las HNN (véase ecuación (2.1)), para bias constantes. A su vez los sistemas dinámicos pueden ser de dos tipos: lineales o no lineales. El primero indica que las funciones contenidas por el vector $\mathbf{f}(\mathbf{x})$ cumplen con la propiedad de superposición (descrita en la sección 2.1.2), mientras que los sistemas no lineales no cumplen con dicha propiedad. Por inspección visual es posible identificar estos sistemas al estar compuestos por operaciones como multiplicaciones entre variables de estado, funciones trigonométricas, potencias, entre otros.

2.2.2. Puntos de equilibrio

La solución de la ecuación diferencial 2.4, suponiendo que exista una, puede ser conocida y expresada por un conjunto de funciones elementales o desconocida, es decir, no es posible expresarla por algún conjunto de funciones elementales. La definición de lo que es una función elemental esta fuera del alcance de esta tesis, considerese simplemente que las funciones estudiadas a lo largo de este trabajo son elementales. La solución para un sistema de ecuaciones diferenciales puede no ser única, es más, lo más común es que exista una familia de soluciones, incluso una solución son dependencia del tiempo, esto es, una solución al sistema de ecuaciones diferenciales constante. Una solución de este tipo es conocida como *punto de equilibrio* o *punto fijo*. Consideremos entonces la solución constante de un sistema dinámico autónomo como

$\mathbf{x}(t) = \eta \in \mathbb{R}^n$, entonces su derivada con respecto al tiempo da

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) = \frac{d\eta}{dt} \Rightarrow \mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad (2.6)$$

siendo $\mathbf{0} \in \mathbb{R}^n$ un vector nulo de n dimensiones. En otras palabras para hallar el punto de equilibrio de un sistema dinámico es la solución del sistema de ecuaciones descrito en la expresión 2.6. Notemos que la solución constante, en caso de que esta existe, debe ser forzosamente la condición del sistema dinámico, esto es, $\eta = \mathbf{x}_0$.

2.2.3. Sistemas dinámicos caóticos

La hazaña de Newton resolviendo el problema de los dos cuerpos inspiró a una gran cantidad de físicos y matemáticos a extender el problema de los dos cuerpos a tres cuerpos, sin embargo, este resultó en una problemática muchísima más compleja. Tras varias décadas de investigación se llegó a la conclusión que el problema de los tres cuerpos era esencialmente imposible de resolver [33]. En 1800, el matemático Henri Poincaré (1854-1912) decidió volver a analizar este sistema, en esta ocasión su análisis tuvo un enfoque más cualitativo que cuantitativo. Posteriormente en 1961, el meteorólogo y matemático Edward Lorenz desarrolló una máquina capaz de predecir el comportamiento del clima, poco tiempo después, se dio cuenta que con mínimas variaciones en las condiciones iniciales, su sistema mostraba dos comportamientos completamente distintos, sus observaciones y resultados fueron publicadas en 1963 [34]. Su trabajo desarrolló lo que ahora se denomina como Teoría del Caos.

Un sistema dinámico caótico o simplemente sistema caótico, es un tipo especial de sistema dinámico que cumple con la característica de ser extremadamente sensible a las condiciones iniciales. Si bien, al día de hoy no existe un consenso general para dar una definición clara de los sistemas caóticos, si se tiene un consenso de tipo de características que estos sistemas deben cumplir [35]:

- **Determinista y no lineal.** Cuando un sistema dinámico está completamente determinado por sus condiciones iniciales se le denomina como determinista, además, cuando se adjudica el término de no lineal a un sistema, implica que este no cumple con la propiedad de la superposición para todos sus estados.
- **Extrema sensibilidad en sus condiciones iniciales.** Si bien, si un sistema caótico es determinista, su dependencia con respecto a sus condiciones iniciales es muy inestable, esto es, diminutas variaciones en las condiciones inicial resultan en cambios drásticos en el comportamiento del sistema.
- **Densidad en sus orbitas periódicas.** Aunque irregular, el comportamiento de un

sistema caótico es periódico, esto es, su órbita gira alrededor de un conjunto de puntos denominado como atractores.

Es importante mencionar que conocido el sistemas de ecuaciones diferenciales que rigen el sistema caótico es posible determinar cómo evoluciona el sistema. Sin embargo, su extrema sensibilidad genera que, tras un cierto periodo de tiempo, no sea posible predecir su comportamiento con absoluta certeza.

2.2.4. Redes Neuronales Hopfield en modo caótico

El modelo matemático de las redes neuronales de Hopfield se comporta como los sistemas caóticos con ciertos parámetros específicos [15, 18], en algunos de estos trabajos se ha llevado a cabo una implementación sobre algún tipo de dispositivo programable, en cuyos casos la respuesta de la HNN ha conservado sus propiedades caóticas.

La identificación de un comportamiento caótico en el modelo de la red neuronal Hopfield ha requerido de herramientas utilizadas en teoría de control, como lo son los diagramas de bifurcación, exponentes de Lyapunov o el cálculo y análisis de los puntos de equilibrio del modelo matemático. Más específicamente, para una red neuronal Hopfield de 3 neuronas, la presencia de un comportamiento caótico fue hallado usando el modelo descrito por [17]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \end{bmatrix}, \quad (2.7)$$

dada la matriz de pesos

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} 0 & 1.2 & -7 \\ 1.1 & 0 & 2.8 \\ 0.7 & -2 & 4 \end{bmatrix}.$$

Un diagrama de fase es una herramienta gráfica utilizada en el análisis de sistemas dinámicos para visualizar el comportamiento a lo largo del tiempo de un conjunto de variables relacionadas. En este tipo de diagrama se gráfica una variable del sistema contra otra distintas, el diagrama de fase proporciona una representación visual de cómo las variables evolucionan y se relacionan entre sí a medida que el sistema dinámico evoluciona.

Los diagramas de fase obtenidos del sistema (2.7) se presentan en la Figura 2.3.

Cabe mencionar que para obtener el diagrama de fase anterior fue requerida la simulación del sistema y, con esta, la solución numérica del sistema caótico, el método utilizado es presentado en la sección 2.4.

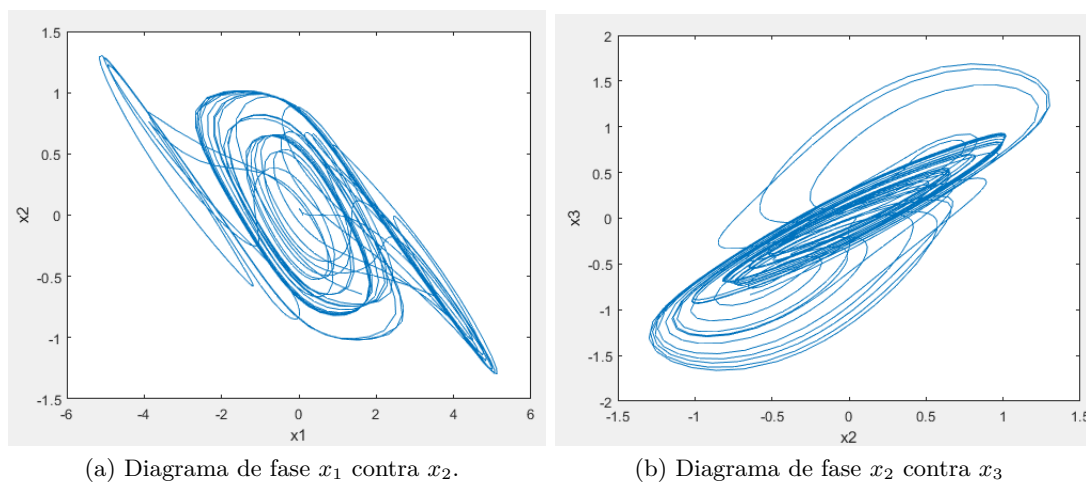


Figura 2.3: Diagramas de fase 2D formados por el sistema 2.7.

2.3. Dispositivos programables

En muchas ocasiones cuando se desarrolla un circuito electrónico es necesario ajustar o incluso rediseñar partes del circuito. Para el caso específico de este trabajo, este rediseño o ajuste resulta ser incluso más complejo en términos de estabilidad y el desarrollo del circuito requerido. Por esto, el uso de dispositivos programables es de gran utilidad en la electrónica, en lo que respecta a dispositivos digitales se utilizan los llamados “Arreglo de Puertas Programables en Campo” (FPGA, por sus siglas en inglés), su contraparte analógica es menos conocida, sin embargo, nos ofrece la misma ventaja, estos son denominados como “Arreglo de Campo de Analógico Programable” (FPAA, por sus siglas en inglés).

2.3.1. Hardware Analógico Embebido

Los FPAA's son dispositivos electrónicos programables que permiten la creación de circuitos analógicos personalizados y, como tal, nos permiten hacer operaciones como lo son la suma, resta, comparación, integración, derivación, entre otros. La arquitectura general de un FPAA es presentada en la Figura 2.5, en esta se observa que está constituido por bloques analógicos configurables (CAB's, por sus siglas en inglés), los cuales tienen una función de transferencia equivalente, estos bloques están interconectados por interruptores controlados. En lo que respecta a este trabajo y su implementación, el FPAA utilizado es el modelo AN231E04 desarrollado por Anadigm, la misma distribuidora ofrece un kit que contiene una tarjeta con 4 FPAA's. Por otro lado, el desarrollador y distribuidor ofrece el software para programar la tarjeta con los AN231E04 llamado AnadigmDesigner2, el cual proporciona un entorno de diseño gráfico con el cual podemos diseñar y simular circuitos analógicos. El ambiente de trabajo de

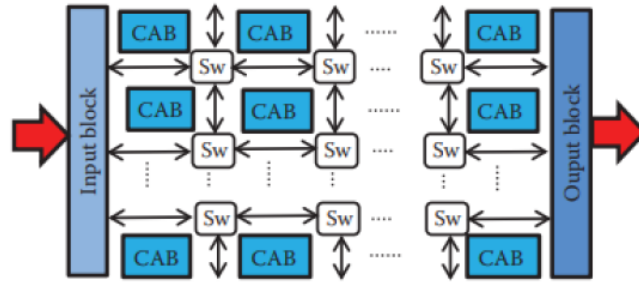


Figura 2.4: Arquitectura general de un FPAAs [36].

AnadigmDesigner2 (Figura 2.5) consiste en bloques, los cuales representan el FPAAs sobre el cual se programará y dentro de este se pueden implementar Módulos Analógicos Configurables (CAM's, por sus siglas en inglés); cada uno de estos tiene una función de transferencia equivalente que se encuentra en la documentación dada por el desarrollador.

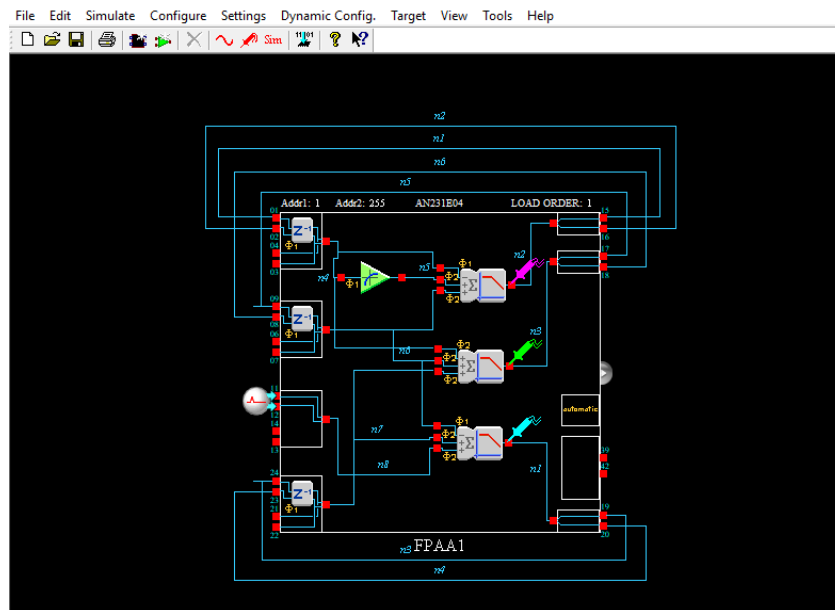


Figura 2.5: Entorno de programación en bloques de AnadigmDesigner2.

2.4. Métodos numéricos para la solución de sistemas dinámicos

En muchos problemas de la física y las matemáticas se estudian modelos descritos por ecuaciones diferenciales lineales y no lineales, es por esto que una solución para dichos sistemas es de suma relevancia. Sin embargo, no siempre es posible el desarrollo de una solución analítica del modelo matemático y, en estos casos, es necesario utilizar métodos numéricos que aproximen

a la solución deseada.

Existen una gran cantidad de métodos numéricos para dar solución a un sistema de ecuaciones diferenciales; con lo que respecta a esta tesis, se utilizarán los métodos desarrollados por los matemáticos alemanes Carl David Tolmé Runge y Martin Wilhelm Kutta, a fines del siglo XIX y principios del siglo XX [37]. En 1895, Tolmé Runge publicó un artículo titulado “Über die numerische Auflösung von Differentialgleichungen”(Sobre la resolución numérica de ecuaciones diferenciales), donde propuso un método iterativo para la solución de ecuaciones diferenciales. Posteriormente, en 1901, Martin Wilhelm Kutta publicó un trabajo titulado “Beiträge zur näherungsweise Integration totaler Differentialgleichungen”(Contribuciones a la integración aproximada de ecuaciones diferenciales totales), donde desarrolló y popularizó un método más amplio y generalizado, que incluía el método de Runge y otros métodos similares. Este método se conoció como el método de Runge-Kutta, combinando los apellidos de ambos matemáticos [37].

Dentro del grupo de métodos Runge-Kutta, en este trabajo se tomarán dos metodologías, el método de Runge-Kutta de segundo orden, también llamado método de Heun, y el método de Runge-Kutta de cuarto orden. El desarrollo para la obtención del método de Heun es el siguiente: consideremos la ecuación diferencial ordinaria (EDO) de la forma

$$\frac{dy(t)}{dt} = f(t, y). \quad (2.8)$$

Aplicamos la expansión de Taylor a la función $y(t)$ alrededor de t_n :

$$y(t) = y(t_n) + (t - t_n) \frac{dy}{dt} \Big|_{t_n} + \frac{(t - t_n)^2}{2} \frac{d^2y}{dt^2} \Big|_{t_n} + O((t - t_n)^3), \quad (2.9)$$

donde $O((t - t_n)^3)$ representa términos de orden superior, que estamos ignorando para obtener una aproximación de segundo orden.

Integrando ambos lados de la ecuación de la EDO de t_n a $t_{n+1} = t_n + h$ (donde h es el tamaño del paso) y aplicando la regla del trapecio para la integración, obtenemos:

$$y_{n+1} = y_n + \frac{h}{2} (f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})), \quad (2.10)$$

donde $\tilde{y}_{n+1}$ es una estimación de y_{n+1} que obtenemos usando la aproximación de Euler:

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n). \quad (2.11)$$

El método de Runge-Kutta de segundo orden ofrece una mejor aproximación a la solución de una ecuación diferencial, a diferencia del muy conocido método de Euler. Sin embargo, en gran parte de los trabajos de investigación se ha popularizado el método de Runge-Kutta de

cuarto orden debido a su facilidad de programación y gran precisión en la solución de ecuaciones diferenciales. Consideremos nuevamente la expresión, (2.8)

$$\frac{dy(t)}{dt} = f(t, y).$$

Para la aplicación del método de Runge-Kutta de 4 orden se divide el intervalo $[t_n, t_{n+1}]$ en cuatro pasos y se calculan las pendientes intermedias k_1 , k_2 , k_3 , y k_4 :

$$\begin{aligned} k_1 &= h \cdot f(t_n, y_n), \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= h \cdot f(t_n + h, y_n + k_3). \end{aligned} \tag{2.12}$$

Se combina ponderadamente las cuatro pendientes para obtener la pendiente final y la solución se actualiza en el siguiente paso de tiempo t_{n+1} :

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{2.13}$$

La cantidad expresada por $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ es una estimación mejorada de la pendiente de la solución en el intervalo de tiempo $[t_i, t_i + h]$, k_1 es la pendiente en el extremo izquierdo del intervalo, k_2 es la pendiente utilizada en el punto medio del intervalo mientras que k_3 es la pendiente mejorada en este mismo punto, finalmente, k_4 es la pendiente en el punto extremo derecho.

Capítulo 3

Análisis de estabilidad de la red neuronal Hopfield

En la literatura existe una diversa cantidad de modelos propuestos para las HNN con presencia de caos usando diversas funciones de activación, cada una ofrece una dinámica distinta al modelo y una estructura distinta para su implementación en circuitos electrónicos. Por lo tanto, resulta de relevancia ofrecer distintos modelos para una HNN que mantenga una compleja dinámica con distintas funciones de activación. De esta manera en esta sección se presentarán 5 modelos para la HNN con distintas funciones de activación con las cuales el sistema presenta un comportamiento caótico. De estos, dos no han sido reportados y fueron hallados por el autor, los tres restantes fueron obtenidos de la literatura, sin embargo, uno de estos fue transformado para su utilización con un nuevo modelo con otra función de activación y por lo tanto es clasificado como un nuevo propuesto en este trabajo.

Cuando se analiza un sistema dinámico se busca realizar la caracterización de las propiedades que describen el considerado sistema. El análisis puede ser dividido en dos secciones: el análisis cuantitativo y el análisis cualitativo. El primero requiere de las características y propiedades que posee el sistema analizado, mientras que el segundo se centra en los resultados estrictamente numéricos obtenidos, por ejemplo, los obtenidos por alguna simulación del sistema. La principal propiedad que se analiza en sistema dinámicos es su estabilidad, para esta existen distintas metodologías, en este trabajo se realizara el estudio de la divergencia del flujo, estabilidad de Lyapunov. Finalmente, se analiza la estabilidad de los sistemas alrededor de los puntos de equilibrio, linealizando el sistema y caracterizando el comportamiento de cada uno de los puntos de equilibrio.

3.1. Propiedades de una función de activación

Desde que Hopfield propuso su modelo de redes neuronales artificiales en 1982, diferentes modificaciones han sido propuestas incorporando nuevos factores internos o externos al modelo, una estas modificaciones es el cambio de la función de activación del modelo de las HNN. En los primeros años de investigación las funciones de activación eran acotadas, crecientes y cero evaluadas en su origen, sin embargo, diversas funciones sin estas propiedades han sido estudiadas con resultados satisfactorios para un grupo específico de problemas, como resultado no existe la definición de un grupo de propiedades para todas las funciones de activación [38]. Para este trabajo sin embargo, las funciones de activación cumplen con las propiedades a continuación enlistadas.

- **Continua.** Sea $f : A \subset \mathbb{R} \rightarrow \mathbb{R}$ y $c \in A$. Se dice que f es continua en c si dado cualquier $\epsilon > 0$, existe un $\delta > 0$ tal que si x es un punto en A se cumple

$$|x - c| < \delta \quad \Rightarrow \quad |f(x) - f(c)| < \epsilon. \quad (3.1)$$

Es importante hacer notar que si una función es diferenciable implica que es continua en el mismo dominio, sin embargo, una función de activación como continua en todo su dominio, a diferencia de su propiedad como función diferenciable. La interpretación de esta propiedad dentro del modelo de las HNN esta relacionada a los estados que toman las neuronas, debido a que estas siempre deben estar en uno la continuidad nos asegura este hecho.

- **Creciente:** Una función $f : A \subset \mathbb{R} \rightarrow \mathbb{R}$ es creciente en A si para cualquier $x_1, x_2 \in A$ se cumple que

$$x_1 \leq x_2 \quad \Rightarrow \quad f(x_1) \leq f(x_2). \quad (3.2)$$

En el caso de que en las desigualdades anteriores se cumple que $x_1 < x_2$ y $f(x_1) < f(x_2)$ entonces se dice que la función es estrictamente creciente. Si una función es creciente implica que su derivada es siempre mayor o igual a cero ($\frac{d}{dx}f(x) \geq 0$), mientras que si es estrictamente creciente es siempre mayor a cero $\frac{d}{dx}f(x) > 0$. Cabe mencionar que algunas funciones de activación consideran el caso contrario, es decir, funciones decrecientes y estrictamente decrecientes, en estos casos este comportamiento no afecta al modelo, sin embargo, se debe hacer la interpretación adecuada de los resultados. Como utilidad de esta propiedad, recordemos que una neurona biológica esta en un estado de encendido o apagado, por lo tanto, si la función de activación fuera no monótona, presentaría comportamientos irregulares no deseados.

- **No lineal:** La no linealidad de un sistema ya fue definida en la sección 2.2 de este

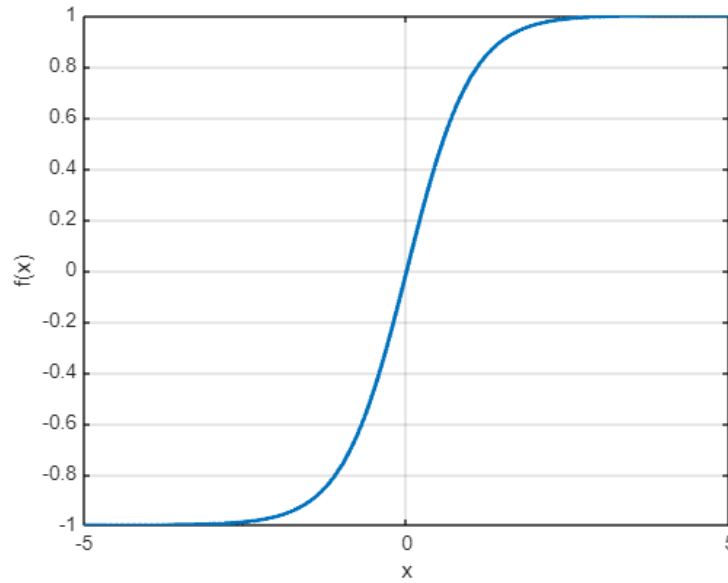


Figura 3.1: Comportamiento de la función $\tanh(x)$.

trabajo. En general, esta propiedad es relevancia ya que nos permite modelar sistemas mas complejos, a diferencia de las funciones lineales. Por otro lado y desde un punto de vista caótico, un sistema no puede presentar caos si esta descrito por funciones lineales.

3.2. Modelos propuestos

3.2.1. Modelo con función de activación tangente hiperbólica

La función tangente hiperbólica es la función de activación mas frecuente en los modelos de HNN, de comportamiento mostrado en la Figura 3.1 esta descrita por la expresión

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (3.3)$$

donde $\cosh x$ y $\sinh x$ son funciones hiperbólicas definidas por $\frac{e^x + e^{-x}}{2}$ y $\frac{e^x - e^{-x}}{2}$ respectivamente, y e^x es la función exponencial, la cual es estrictamente creciente, siempre mayor a 0, diferenciable en todo el dominio $\mathbb{R}$ y por lo tanto continua en el mismo. Además, esta función esta acotada como $|\tanh(x)| < 1$. La derivada de esta función viene dada por

$$\frac{d}{dx} \tanh(x) = 1 - \tanh^2 x \quad (3.4)$$

Como fue mencionado, distintos modelos para la HNN utilizan esta función de activación, para este trabajo se considerara **un modelo de 3 dimensiones desarrollado por el mismo**

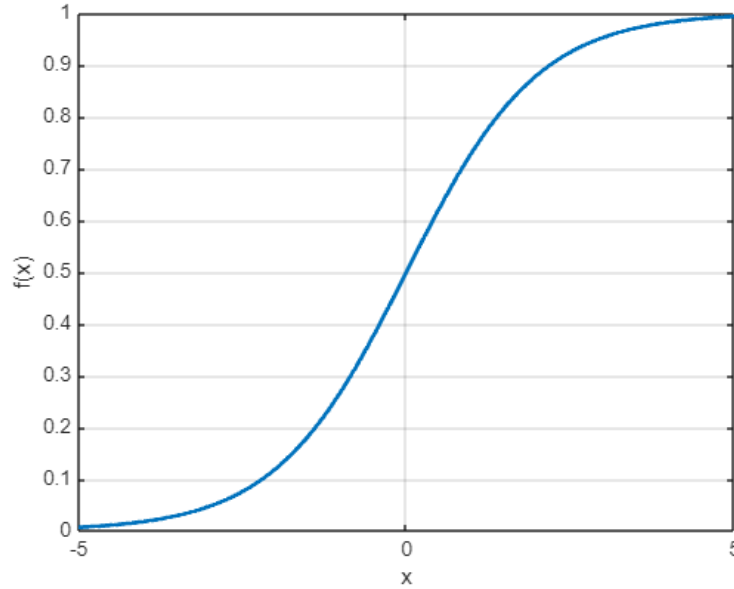


Figura 3.2: Comportamiento de la función $\sigma(x)$.

autor descrito de la siguiente manera

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & -2 & -9 \\ 1 & 3 & 3 \\ -1 & -3 & 0 \end{bmatrix} \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \\ \tanh x_3 \end{bmatrix}, \quad (3.5)$$

este modelo presenta algunas ventajas con respecto a otros modelos antes propuestos, el primero es su sencillez considerando que todos sus parámetros son enteros, y por otro lado, presenta cierta robustez con respecto a su comportamiento caótico, es decir, algunos de los pesos pueden ser variados por un rango amplio de valores y aun así el sistema mantendrá su comportamiento caótico. Esto ultimo es explorado en el análisis dinámico de las HNN.

3.2.2. Modelo obtenido con función de activación sigmoide

La función sigmoide, típicamente referida por la letra griega sigma σ y comportamiento mostrado en la Figura 3.2, es otra función de activación típicamente utilizada, esta es diferenciable en todo su dominio, es acotada por $0 < \sigma(x) < 1$ y esta definida por

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (3.6)$$

con derivada

$$\frac{d}{dx} \sigma(x) = \sigma(x)(1 - \sigma(x)). \quad (3.7)$$

Siguiendo el siguiente desarrollo es posible relacionar esta función con la tangente hiperbólica

$$\tanh x = \frac{e^x e^x - e^{-x}}{e^x e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{2}{1 + e^{2x}} - 1 \quad (3.8)$$

$$\Rightarrow \tanh x = 2\sigma(2x) - 1, \quad (3.9)$$

entonces es posible usar modelos de HNN que utilizan la función de activación $\tanh(\cdot)$ y reescribirlos usando $\sigma(\cdot)$, para esto se realizo el siguiente desarrollo. Consideremos una HNN n-dimensional, descrito por

$$\dot{\mathbf{x}} = -C\mathbf{x} + W \tanh(\mathbf{x}) + \mathbf{b}, \quad (3.10)$$

donde $\tanh(\mathbf{x})$ es un vector de n elementos dados por $\tanh(x_i)$ para todo $i = (1, 2, \dots, n)$ y por lo tanto puede ser reescrito como

$$\tanh(\mathbf{x}) = \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \\ \vdots \\ \tanh x_n \end{bmatrix} = \begin{bmatrix} 2\sigma(2x_1) - 1 \\ 2\sigma(2x_2) - 1 \\ \vdots \\ 2\sigma(2x_n) - 1 \end{bmatrix}, \quad (3.11)$$

consideremos el vector $\Sigma(\mathbf{x})$ de n elementos descritos por $\sigma(x_i)$ para todo $i = (1, 2, \dots, n)$ y un cambio de variables $2x_i \rightarrow x_i^*$. En :

$$\begin{aligned} \dot{\mathbf{x}} = -C\mathbf{x} + W \tanh(\mathbf{x}) + \mathbf{b} &\Rightarrow \dot{\mathbf{x}} = -C\mathbf{x} + W \begin{bmatrix} 2\sigma(2x_1) - 1 \\ 2\sigma(2x_2) - 1 \\ \vdots \\ 2\sigma(2x_n) - 1 \end{bmatrix} + \mathbf{b}, \\ &\Rightarrow \dot{\mathbf{x}} = -C\mathbf{x} + 2W\Sigma(2\mathbf{x}) - WI_{n \times 1} + \mathbf{b}, \\ &\Rightarrow \dot{\mathbf{x}}^* = -C\mathbf{x}^* + 4W\Sigma(\mathbf{x}^*) - 2WI_{n \times 1} + 2B. \end{aligned} \quad (3.12)$$

donde $I_{n \times 1} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^{n \times 1}$. En otras palabras, es posible crear a un nuevo modelo

$$-\mathbf{x}^* = -C_\sigma \mathbf{x} + W_\sigma \Sigma(\mathbf{x}^*) + \mathbf{b}_\sigma, \quad (3.13)$$

donde

$$\begin{aligned} \Sigma(\mathbf{x}) &= \begin{bmatrix} \sigma(x_1) & \sigma(x_2) & \dots & \sigma(x_n) \end{bmatrix}^T, \\ C_\sigma &= C, \\ W_\sigma &= 4W, \\ \mathbf{b}_\sigma &= -2WI_{n \times 1} + 2B, \end{aligned}$$

con $I_{n \times 1} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n$, en base a un modelo de HNN con función de activación $\tanh(\cdot)$ descrito por las matrices y vector W , C y $\mathbf{b}$ respectivamente. Consideremos el sistema propuesto en [1] para una HNN con la matriz de pesos

$$W = \begin{bmatrix} 1.99 & -1 & 0 \\ 2.5 & 1.7075 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix}, \quad (3.14)$$

para este sistema el vector B es un vector nulo, mientras que C es la matriz identidad de dimensión 3. Este sistema es transformado aplicando la expresión (3.12) de la siguiente manera

$$\begin{aligned} \dot{\mathbf{x}}^* &= -\mathbf{x}^* + 4 \begin{bmatrix} 2 & -1 & 0 \\ 2.5 & 1.71 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix} \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \sigma(x_3) \end{bmatrix} - 2 \begin{bmatrix} 2 & -1 & 0 \\ 2.5 & 1.71 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \\ \Rightarrow \dot{\mathbf{x}}^* &= -\mathbf{x}^* + \begin{bmatrix} 7.97 & -4 & 0 \\ 10 & 6.84 & 5 \\ -19 & 0 & 3.67 \end{bmatrix} \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \sigma(x_3) \end{bmatrix} + \begin{bmatrix} -2 \\ -10.92 \\ 7.66 \end{bmatrix}. \end{aligned} \quad (3.15)$$

El sistema anterior mantiene el comportamiento del sistema original permitiendo usar la función de activación $\sigma(\cdot)$.

3.2.3. Modelo con función de activación signo

La función signo, denotada por $\text{sign}(\cdot)$, indica el signo tiene el valor de entrada, si este es positivo la función regresa el valor 1, si es negativo la función regresa el valor de -1, para un valor de entrada la función regresa un 0 (véase Figura 3.3), de manera matemática es posible describir este comportamiento de la forma

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0, \end{cases} \quad (3.16)$$

es evidente que la derivada en el punto $x = 0$ no existe, para todos lo demás puntos esta es 0. Otro dato de interés de esta función es su relación con la función $\tanh(\cdot)$, esto debido a que es posible relacionar ambas funciones de la siguiente manera

$$\lim_{k \rightarrow \infty} \tanh(kx) \approx \text{sign}(x), \quad (3.17)$$

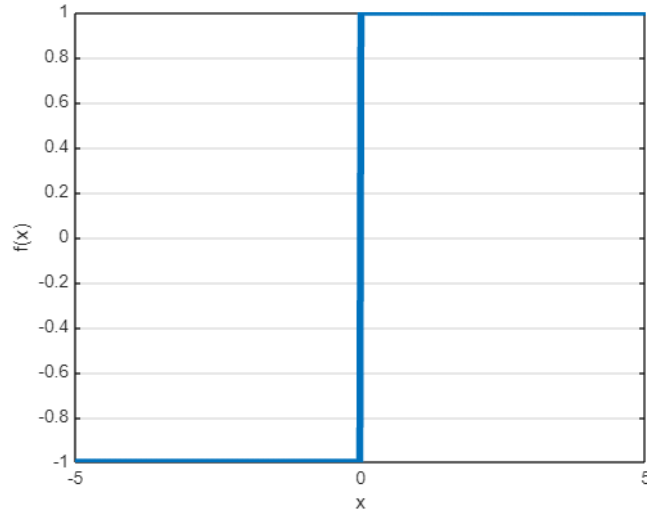


Figura 3.3: Comportamiento de la función $\text{sign}(x)$.

esta última relación podría ser utilizada, junto con un cambio de variables, para utilizar alguno de los modelos ya conocidos de la HNN con la función de activación $\tanh(\cdot)$ y obtener un comportamiento caótico, sin embargo, este acercamiento genera problemáticas en la implementación y simulación, es por esto que **el autor propuso un nuevo modelo** en donde la función $\text{sign}(\cdot)$ es sustituida en solo una de las funciones de activación del modelo, esta viene dada por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.65 & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{sign}(x_1) \\ \text{sign}(x_2) \\ \text{sign}(x_3) \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 4 & 3 \end{bmatrix} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \end{bmatrix}. \quad (3.18)$$

El modelo anterior representa, hasta donde el autor esta consciente, el modelo de red neuronal Hopfield con presencia de caos mas sencillo, con cuatro pesos del modelo en 0, lo cual simplifica en gran medida la implementación en circuitos electrónicos. Además, este modelo tiene la particularidad de tener un muy amplio espectro de posibilidades para el parámetro w_{12} , para los cuales es sistema muestra un comportamiento caótico para todo $w_{12} \in \mathbb{R}_+$, esta conclusión se obtiene tras las simulaciones y análisis posteriormente realizados en este trabajo.

3.3. Modelos reportados en la literatura

3.3.1. Modelo con función ReLU

La función $\text{ReLU}(\cdot)$ es una de las funciones de activación mas utilizadas en ANN, su nombre viene del ingles "Rectified Linear Unit" ó "Unidad Lineal Rectificada" es español, es una función

a trozos definida por

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad (3.19)$$

en donde $\max(\cdot)$ es una función que regresa en valor mas grande de los argumentos. Es evidente que esta función es siempre mayor o igual a 0 para todo x y, a diferencia de las primeras funciones de activación presentadas con anterioridad, no es diferenciable en todo su dominio ya que viene dada por

$$\frac{d}{dx}\text{ReLU}(x) = \begin{cases} 0, & x < 0, \\ 1, & x > 0, \end{cases}, \quad (3.20)$$

para el punto $x = 0$ la derivada no existe. Con lo que respecta a su implementación en HNN, existen una cantidad muy limitada de modelos con presencia de caos, hasta el momento en que esta tesis esta siendo escrita solo tres modelos han sido propuestos, el primero propuesto por C.Chen y F. Mina [39] incluye en el modelo la presencia de un Memristor, lo cual modifica el modelo dinámico de la HNN, después C. Chen, F. Min, et. al., proponen dos modelos mas, primero [21] tiene una estructura utilizada a lo largo de esta tesis, sin embargo su implementación en tarjetas embebidas resulta ser mas compleja, esta manera se utiliza el modelo presentado en [2] descrito por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1.15 & 0 \\ 0 & 0.55 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1.1 & -1.05 \\ 0.95 & 0.75 \end{bmatrix} \begin{bmatrix} \text{ReLU}(x_1) \\ \text{ReLU}(x_2) \end{bmatrix} + \begin{bmatrix} -0.5\text{sen}(0.52t) + 0.8 \\ -0.75 \end{bmatrix}, \quad (3.21)$$

este modelo es un HNN de dos dimensiones con un voltaje sinoidal de estimulación sobre la neurona x_1 .

3.3.2. Modelo con función de activación PWL

En distintas ocasiones es posible aproximar el comportamiento de distintas funciones haciendo uso de un conjunto de funciones lineales, en estos caso se suele recurrir a las llamadas "Piecewise Linear Functions" o "Función Lineal a Trozos" en español. En este caso, es posible aproximar el comportamiento de la función $\tanh(\cdot)$ por medio de tres rectas (véase Figura 3.5) propuestas [3], a esta función se denomina como $\text{satlin}(\cdot)$ y esta definida como

$$\text{satlin}(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \leq x < 1, \\ 1, & x \geq 1, \end{cases} \quad (3.22)$$

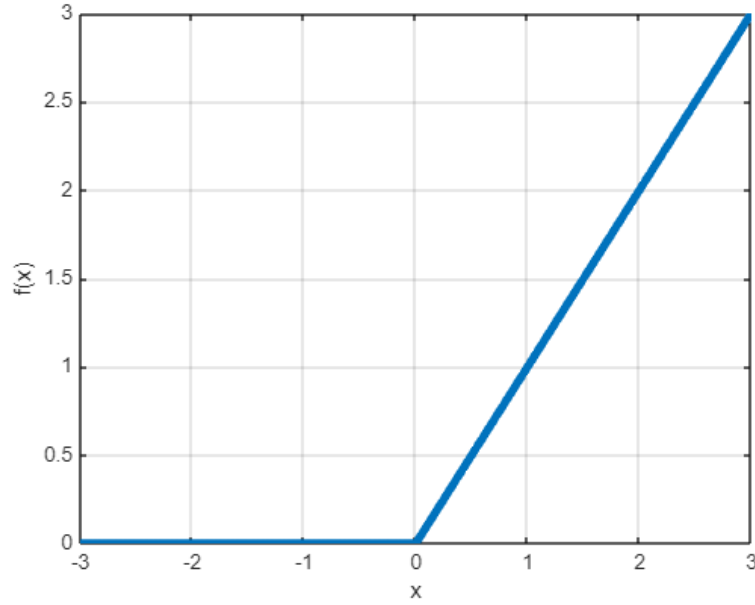


Figura 3.4: Comportamiento de la función $\text{ReLU}(x)$.

y presenta un comportamiento dinámico caótico en el modelo

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1.68 & 3.81 & -2.23 \\ -5.4 & 1.8 & -4.4 \\ -4.38 & -2.2 & -0.3 \end{bmatrix} \begin{bmatrix} \text{satlin}(x_1) \\ \text{satlin}(x_2) \\ \text{satlin}(x_3) \end{bmatrix}. \quad (3.23)$$

Para esta caso entonces se tiene una red neuronal completamente conectada. La derivada de esta función activación esta definida para todo su dominio menos en $x = 0$, y viene dada por

$$\frac{d}{dx} \text{satlin}(x) = \begin{cases} 1, & -1 < x < 1, \\ 0, & \text{otro} \end{cases} \quad \forall (x \neq -1 \wedge x \neq 1). \quad (3.24)$$

3.4. Análisis en base a la divergencia del sistema dinámico

Conforme un sistema dinámico evoluciona en el tiempo entonces el volumen del espacio fásico aumenta, disminuye o se conserva, conforme a esta característica los sistemas dinámicos pueden ser divididos en tres categorías [40]:

- Sistemas dinámicos conservativos. Estos son aquellos que conforme pasa el tiempo preservan su energía y, de acuerdo al teorema de Liouville, a pesar de la traslación y el

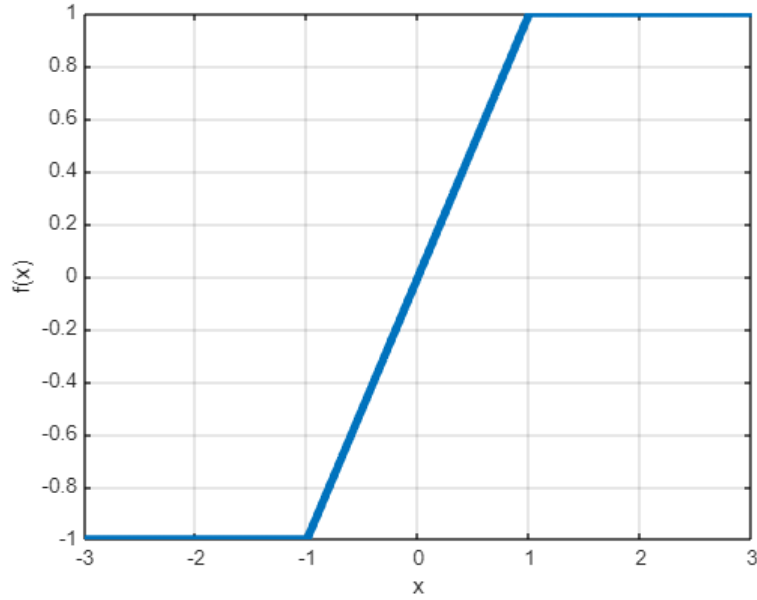


Figura 3.5: Comportamiento de la función $\text{satlin}(x)$.

cambio de forma el volumen total de dicha región permanecerá invariante. En este tipo de sistemas no pueden existir atractores.

- Sistemas dinámicos disipativos. Estos son sistemas que disipan energía conforme pasa el tiempo, es decir, el volumen del espacio físico conforme avanza el tiempo se reduce, en estos casos es posible afirmar que en este sistema existe algún o algunos atractor(es).
- Sistemas dinámicos inestables o no acotados. Estos son sistemas cuya evolución genera que el volumen del espacio físico no deje de crecer.

Esta característica es calculada a través de la operación divergente con la cual se calcula la media de la velocidad del volumen del espacio de fase generado. La operación divergente se define como sigue. Consideremos el espacio vectorial $\mathbf{f}$ definido por un conjunto abierto $\Omega \subset \mathbb{R}^n$ y consideremos sus coordenadas $\mathbf{f} = [f_1 \ f_2 \ \dots \ f_n]^T$. Ahora supongamos que $\mathbf{f}$ es diferenciable en todos los puntos de $\mathbf{x} \in \Omega$, entonces la **divergencia** del campo F es una función $\text{div}(\mathbf{f}) : \Omega \rightarrow \mathbb{R}$ calculado por

$$\text{div}(\mathbf{f}(\mathbf{x})) = \nabla \cdot \mathbf{f}(\mathbf{x}) = \sum_{i=1}^n \frac{\partial f_i(\mathbf{x})}{\partial x_i}, \quad (3.25)$$

donde ∇ es el operador nabla descrito por el vector $\left[\frac{\partial}{\partial x_1} \ \frac{\partial}{\partial x_2} \ \dots \ \frac{\partial}{\partial x_n} \right]$. La divergencia tiene una importante interpretación física, esta mide la *razón de expansión del volumen generado por*

las variables de estado [40]. En este sentido, el sistema estudiado es disipativo si se cumple que

$$\nabla \cdot \mathbf{f}(\mathbf{x}) < 0, \quad (3.26)$$

por otro lado, el sistema es conservativo si se cumple que

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = 0. \quad (3.27)$$

En otras palabras, *si la divergencia de un sistema es menor a cero, el sistema contiene atractores*. Usualmente la divergencia de un sistema suele ser denotada por la letra griega Λ .

Para el caso específico de una red neuronal Hopfield de n neuronas descrito por la expresión (2.1), se puede afirmar el siguiente teorema

Teorema 3.4.1. *Sea una función $f : \mathbb{R} \rightarrow \mathbb{R}$ que es continua y creciente, cuya derivada $f'(x)$ existe para todo $x \in A \subseteq \mathbb{R}$ y cuya cota superior mínima es algún real k . Consideremos a f como la función de activación para el modelo de las HNN, entonces, existe como mínimo un atractor en el sistema si se cumple que*

$$\sum_{i=1}^n c_i > k \sum_{i=1}^n w_{ii}, \quad (3.28)$$

en otras palabras es disipativo, mientras que el modelo es conservativo si se cumple

$$\sum_{i=1}^n c_i = k \sum_{i=1}^n w_{ii}. \quad (3.29)$$

Demostración. Consideremos el modelo n -dimensional de una HNN, en forma de sumatoria esta puede escrita por

$$\dot{x}_i = -c_i x_i + b_i + \sum_{j=1}^n w_{ij} f(x_j), \quad (3.30)$$

su divergencia viene dada por

$$\begin{aligned} \Lambda &= \nabla \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} -c_1 x_1 + b_1 + \sum_{j=1}^n w_{1j} f(x_j) \\ -c_2 x_2 + b_2 + \sum_{j=1}^n w_{2j} f(x_j) \\ \vdots \\ -c_n x_n + b_n + \sum_{j=1}^n w_{nj} f(x_j) \end{bmatrix}, \\ &= \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(-c_i x_i + b_i + \sum_{j=1}^n w_{ij} f(x_j) \right) = \sum_{i=1}^n \left(-c_i + w_{ii} \frac{\partial}{\partial x_i} f(x_i) \right). \end{aligned} \quad (3.31)$$

Por hipótesis se sabe que la derivada de $f(x_i)$ existe para todo $\mathbf{x} \in A$ y esta acotada superior-

mente por una k , i.e., $\frac{\partial f(x_i)}{\partial x_i} \leq k$, esto implica que

$$\sum_{i=1}^n \left(-c_i + w_{ii} \frac{\partial}{\partial x_i} f(x_i) \right) \leq \sum_{i=1}^n (-c_i + k w_{ii}). \quad (3.32)$$

Para que nuestro sistema sea disipativo, y por lo tanto pueda contener atractores, se debe cumplir que su divergencia sea negativa, de esta manera se tiene

$$\sum_{i=1}^n (-c_i + k w_{ii}) < 0, \quad (3.33)$$

$$\sum_{i=1}^n c_i > k \sum_{i=1}^n w_{ii}. \quad (3.34)$$

Mientras que el sistema será conservativo si la divergencia es nula, por lo tanto si el modelo de HNN cumple el criterio

$$\sum_{i=1}^n c_i = k \sum_{i=1}^n w_{ii}. \quad (3.35)$$

□

En algunos sistemas la divergencia queda en términos de alguna de las variables de estado, en este caso, es necesario realizar algún tipo de simulación del sistema para hallar el valor adecuado de k o llevar a cabo otro análisis para concluir sobre la disipación del sistema. En otros sin embargo, es suficiente hallar la cota superior mínima de la parcial $\frac{\partial f(x_i)}{\partial x_i}$ para todo $i = (1, 2, \dots, n)$.

Ejemplos ilustrativos. Apliquemos el teorema 3.4.1 las funciones de activación $\tanh(\cdot)$ y $\sigma(\cdot)$, con derivadas iguales a $1 - \tanh^2 x_i$ y $\sigma(x_i)(1 - \sigma(x_i))$. Es sencillo hallar la cota superior mínima de las derivadas antes mencionadas de la siguiente manera. Primero verifiquemos que existe una cota superior para la función $1 - \tanh^2(x)$ para todo $x \in \mathbb{R}$, denominemos a $k \in \mathbb{R}$ como la cota superior, entonces se debe cumplir que

$$\begin{aligned} 1 - \tanh^2(x) &= 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right)^2 = \frac{e^{2x} + e^{-2x} + 2 - e^{2x} - e^{-2x} + 2}{(e^x + e^{-x})^2} \leq k, \\ \Rightarrow \quad \frac{4}{(e^x + e^{-x})^2} &\leq k, \\ \Rightarrow \quad \frac{1}{((e^x + e^{-x})/2)^2} &\leq k, \end{aligned} \quad (3.36)$$

el denominador de la parte izquierda de la expresión anterior es siempre positiva y por lo tanto el valor de la fracción solo decrece, en otras palabras la cota superior k existe y su valor depende

de la cota inferior máxima de la expresión $((e^x + e^{-x})/2)^2$, para hallar esta notemos que la siguiente desigualdad se cumple para todo x real

$$\begin{aligned} (e^{\frac{1}{2}x} - e^{-\frac{1}{2}x})^2 &\geq 0, \\ \Rightarrow e^x - 2 + e^{-x} &\geq 0, \\ \Rightarrow e^x + e^{-x} &\geq 2, \\ \Rightarrow \frac{e^x + e^{-x}}{2} &\geq 1, \end{aligned}$$

sustituyendo esta cota superior en (3.36) se tiene

$$\frac{1}{1} = 1 \leq k \quad \therefore \quad 1 - \tanh^2(x) \leq 1,$$

por lo tanto, para la función de activación tangente hiperbólica $k = 1$ es el valor de la cota superior mínima, este valor es alcanzado solo cuando $x = 0$. En base al criterio dado por el teorema 3.4.1, el modelo de redes neuronales Hopfield con función de activación $\tanh(\cdot)$ es disipativo si se cumple

$$\sum_{i=1}^n c_i > \sum_{i=1}^n w_{ii}, \quad (3.37)$$

que a su vez indica que, cumpliendo el criterio, el sistema puede sostener atractores. Apliquemos este criterio para el modelo presentado con función de activación tangente hiperbólica descrito en (3.5), notemos que la suma de elementos c_i es igual a 3 mientras que en la diagonal solo hay un peso con valor igual a tres, es decir, recordemos que la cota superior mínima es alcanzada para $x = 0$, dado que el sistema evoluciona se tiene que el sistema es mayormente disipativo y conservativo en breves periodos de tiempo para los cuales x_2 es cero. A pesar de esto, el sistema efectivamente conserva sus atractores y se muestran en la Figura 3.6, en donde se observa un claro doble-scroll.

Para la función sigmoide $\sigma(x)$ se sigue el siguiente procedimiento

$$\begin{aligned} (e^{-\frac{1}{2}x} - e^{\frac{1}{2}x})^2 \geq 0 &\Rightarrow e^x + e^{-x} - 2 \geq 0 \\ &\Rightarrow e^x + e^{-x} + 2 \geq 4 \\ &\Rightarrow \frac{e^{-x}}{1+e^{-2x}+2e^{-x}} = \frac{1-1+e^{-x}}{(1+e^{-x})^2} \leq 1/4 \\ &\Rightarrow \frac{1}{1+e^{-x}} \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} \right) \leq 1/4 \\ &\Rightarrow \sigma(x)(1 - \sigma(x)) \leq 1/4. \end{aligned}$$

en otras palabras $k = \frac{1}{4}$ es la cota superior mínima de la derivada de $\sigma(x)$. Entonces podemos concluir entonces que una condición suficiente para que el sistema sea disipativo, y por lo tanto es sistema soporte atractores, en el modelo HNN con la función de activación sigmoide es que

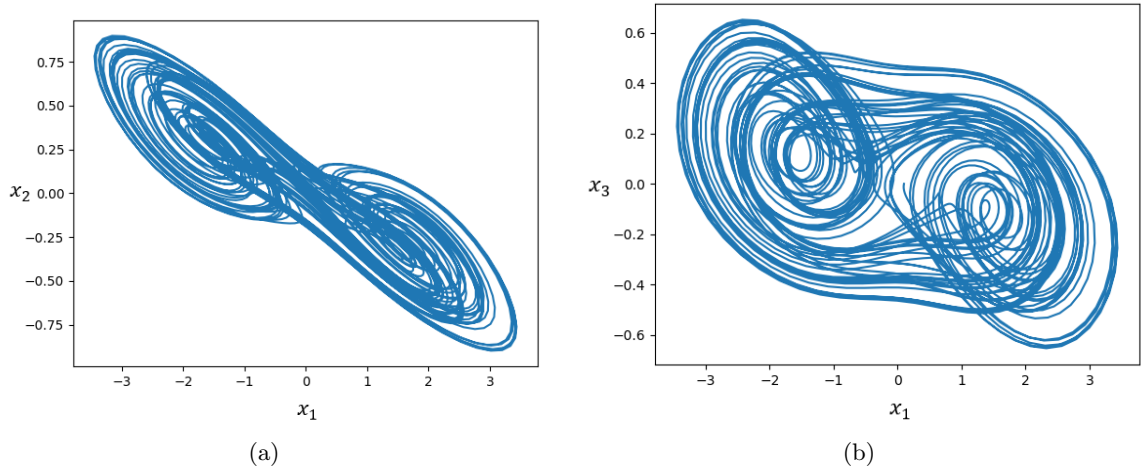


Figura 3.6: Retrato de fase formado por el modelo con función de activación tangente hiperbólica.

se cumpla

$$\sum_{i=1}^n c_i > \frac{1}{4} \sum_{i=1}^n w_{ii}. \quad (3.38)$$

Por otro lado, para las funciones de activación $\text{ReLU}(\cdot)$ y $\text{satlin}(x)$ la cota superior de su derivada tiene un valor de $k = 1$ de manera directa, por lo que es posible aplicar la expresión

$$\sum_{i=1}^n c_i > \sum_{i=1}^n w_{ii}, \quad (3.39)$$

como criterio para la existencia de atractores. Sin embargo, no es posible concluir sobre la existencia de atractores en el modelo presentado 3.15 debido que no cumple el criterio presentado. Este mismo análisis puede ser aplicado al modelo con la función $\text{sign}(\cdot)$ debido a que su divergencia esta dada por

$$\Lambda = -3 + 3(1 - \tanh^2 x_3) = -3 \tanh^2 x_3, \quad (3.40)$$

por lo que la desigualdad (3.37) como criterio también es aplicable, y entonces se puede concluir que este sistema también puede contener atractores. El retrato de fase de este modelo es presentados en la Figura 3.7 en donde se observa un doble-scroll, es decir, un sistema con dos atractores.

Desafortunadamente ninguno de los modelos faltantes utilizados pueden usar directamente el teorema desarrollado con el valor k calculado debido a su dependencia a alguna o algunas de la variables de estado, esta problemática es abordada en [40] en el cual se dividen a los sistemas conservativos y disipativos en 4 categorías:

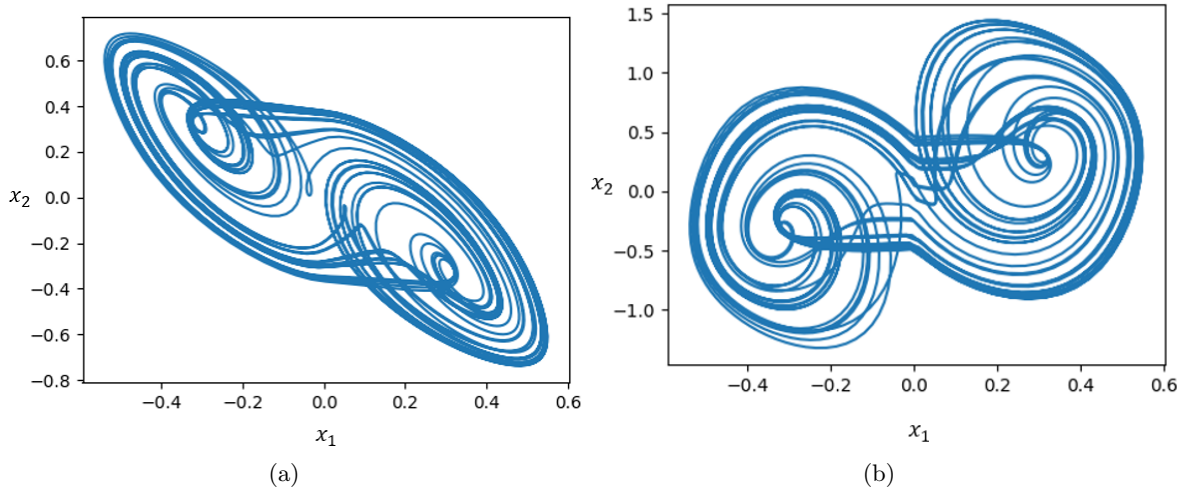


Figura 3.7: Retrato de fase formado por el modelo con función de activación signo.

- Categoría A: Órbitas en las cuales la divergencia del sistema es uniformemente cero.
- Categoría B: Órbitas cuya divergencia tienen una dependencia a alguna o algunas de las variables de estado y esta es cero en promedio como consecuencia del acotamiento del sistema.
- Categoría C: Órbitas cuya divergencia dependen de el estado del sistema, es cero en promedio pero su demostración requiere de una transformación a un sistema hamiltoniano.
- Categoría D: Sistemas en donde la disipación del sistema depende de las condiciones iniciales y no puede ser determinado simplemente por las ecuaciones que a esta describe.

Si un sistema dinámico tiene una solución acotada (es disipativo o conservativo), entonces la velocidad media de todos los estados debe ser cero, se denota el valor medio de una órbita $S(t)$ como $\langle S(t) \rangle$ y se define por

$$\langle S(t) \rangle = \lim_{t \rightarrow \infty} \frac{\int_0^t S(t) dt}{t}, \quad (3.41)$$

como se sabe, el sistema estudiado no tiene una primitiva elemental y como tal la integral en (3.41) será obtenida de manera numérica. Esta métrica será utilizada entonces para concluir si los sistemas dinámicos faltantes son capaces de contener atractores.

Consideremos el modelo descrito por (3.15) con la función de activación $\sigma(x)$, usando el operador divergencia se obtiene

$$\Lambda = -3 + 7.97\sigma(x_1)(1 - \sigma(x_1)) + 6.83\sigma(x_2)(1 - \sigma(x_2)) + 3.68\sigma(x_3)(1 - \sigma(x_3)), \quad (3.42)$$

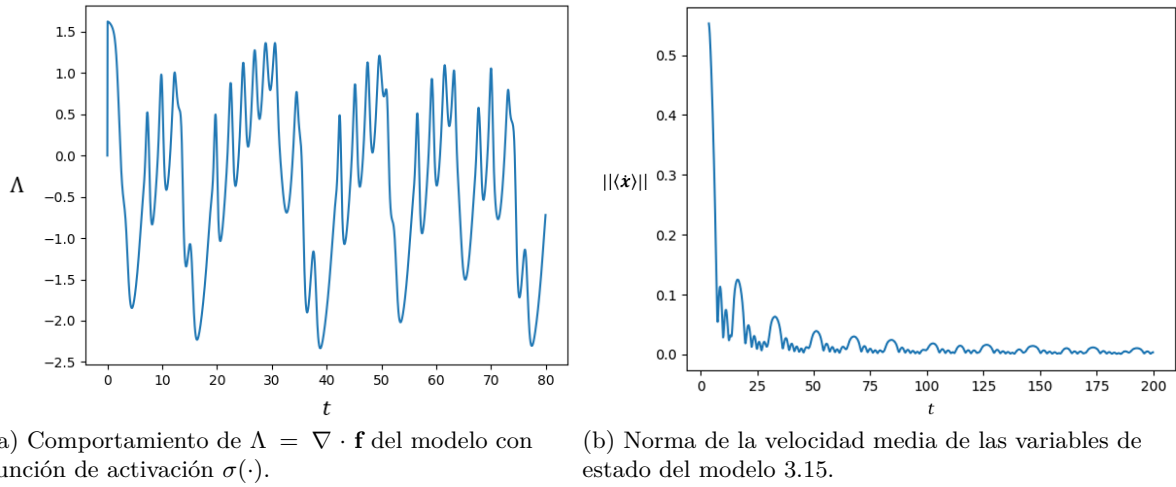


Figura 3.8: Comportamiento de la divergencia a lo largo del tiempo en el modelo (3.15).

la derivada de $\sigma(x)$ es simétrica y máxima cuando x se aproxima a 0, en base a esto podemos deducir que la divergencia sera máxima cuando las variables de estado estén cercanas a cero, este comportamiento fue graficando en la Figura 3.8 a), en el cual hay una evidente variación irregular en el tiempo en donde el sistema pasa de tener divergencia positiva a negativa y viceversa. Para evaluar este comportamiento se decidió obtener el área bajo la curva de Λ calculada tras 200 segundos, el resultado obtenido fue de -351.55, lo que indica que el sistema tiene una divergencia mayormente negativa. Además, en la Figura 3.8 b) se gráfico la norma de la media de la velocidad de la variables de estado del sistema dinámico, como es esperable en un sistema conservativo este valor se aproxima a cero conforme el sistema evoluciona.

Podemos concluir entonces que el sistema es divergente y puede soportar atractores. En la Figura 3.9 se observa el retrato de fase del sistema tras una simulación 300 segundos con un paso de integración $1e-3$ y con condiciones iniciales iguales a $(0.1, 0, 0)$.

Ahora consideremos el modelo con la función de activación $\text{ReLu}(x)$, la velocidad de expansión o contracción del volumen del espacio fásico esta dada por

$$\Lambda = -1.7 + 1.1g(x_1) + 0.75g(x_2), \quad (3.43)$$

donde $g(x)$ es la derivada de la función $\text{ReLu}(x)$ dada por la expresión (3.20), esto indica que en la Λ puede tomar 4 valores distintos dependiendo del signo que tengan las variables de estado del sistema conforme este evoluciona, este comportamiento se observa en la Figura 3.10 a), obteniendo el área bajo la curva de $\Lambda(t)$ se obtiene un valor alrededor de -56.84, lo que indica que el valor de Λ es mayoritariamente negativo. Además, la Figura 3.10 b) se obtiene la norma del valor medio de la velocidad de las variables de estado, este resulta aproximarse a

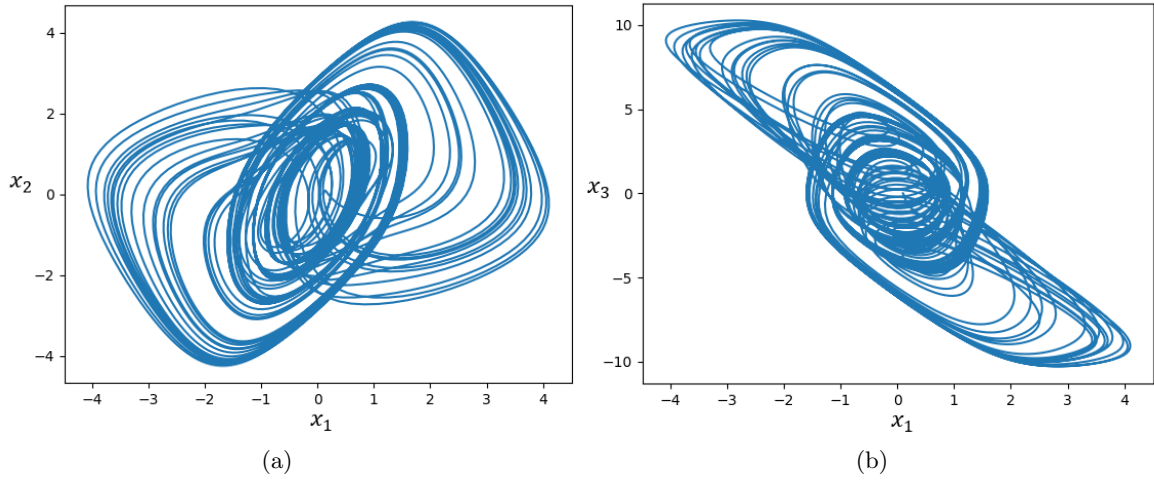


Figura 3.9: Retrato de fase formado por el modelo con función de activación sigmoide para el plano a) $x_1 - x_2$ y b) $x_1 - x_3$.

0 conforme el tiempo aumenta como es esperable en sistema disipativo. En resumen podemos concluir que este sistema tiene la capacidad de contener atractores y esto se puede confirmar al simular el sistema como fue hecho en la Figura 3.11 con un tiempo de simulación igual a 300 segundos, un paso de integración de $1e-3$ y con condiciones iniciales $(0.1, 0)$, aquí se observa un solo atractor en el sistema.

Finalmente, retomando el modelo con la función PWL denominada $\text{satlin}(x)$ se tiene que

$$\Lambda = -3 + 1.68r(x_1) + 1.8r(x_2) - 0.3r(x_3), \quad (3.44)$$

donde $r(x)$ es la derivada de la función PWL definida en (3.24), de manera similar a la función $\text{ReLU}(x)$ se tiene que Λ tiene una cantidad limitada de valores, en específico puede tomar 8 valores distintos y dependen si el valor de los estados están en el rango de $|x_i| < 1$ para $i = (1, 2, 3)$, este comportamiento se observa en la Figura 3.12 a), como métrica para evaluar este comportamiento se obtuvo el área bajo la curva, el valor obtenido fue de -131.75 tras una simulación de 300 segundos. De la misma forma, se obtuvo la curva que describe la norma de la velocidad media de las variables de estado en el sistema mostrada en la Figura 3.12, en donde queda en evidencia que esta se aproxima a cero conforme el tiempo avanza como se espera en un sistema de flujo conservativo.

Como resultados podemos afirmar la presencia de atractores en el sistema estudiado, estos son presentados en la Figura 3.13 en donde se observan dos claros scrolls. En resumen, el análisis del valor de Λ en el sistema de HNN's nos permite concluir si el sistema es conservativo, disipativo o no acotado, lo cual implica la existencia o no de atractores en el sistema, además,

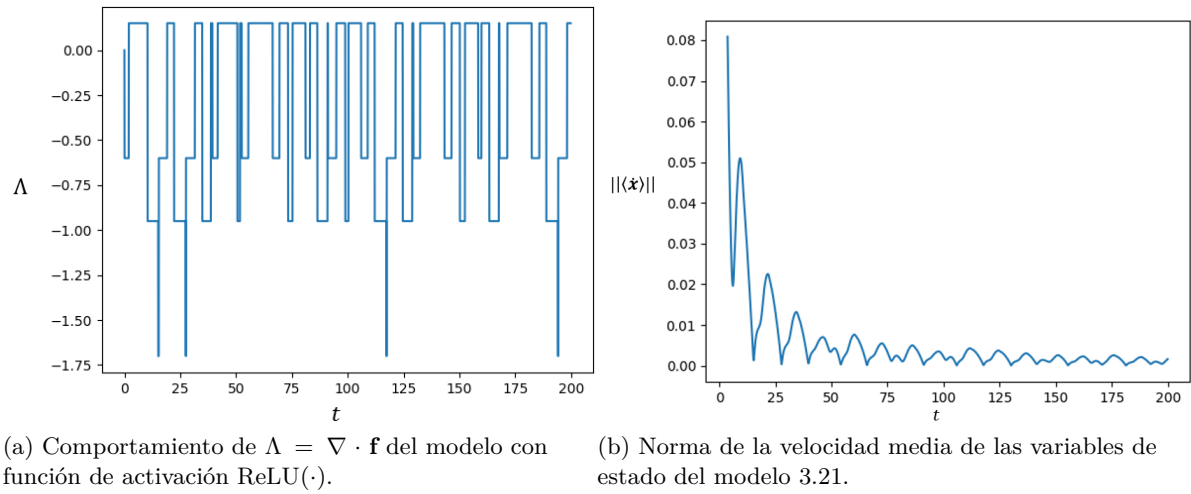


Figura 3.10: Comportamiento de la divergencia a lo largo del tiempo en el modelo con función de activación $\text{ReLU}(\cdot)$.

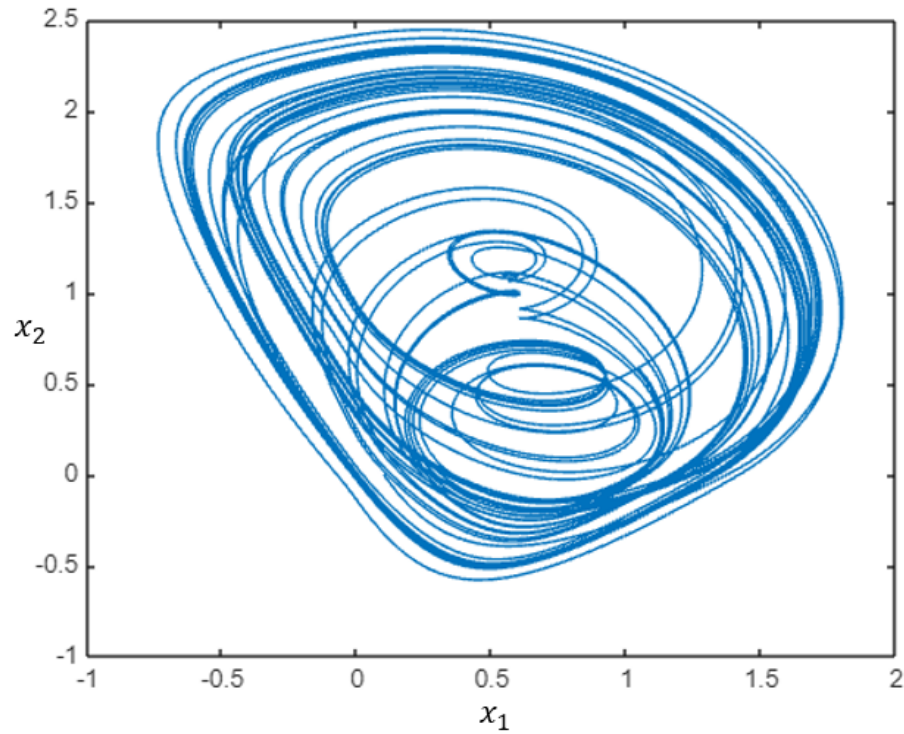
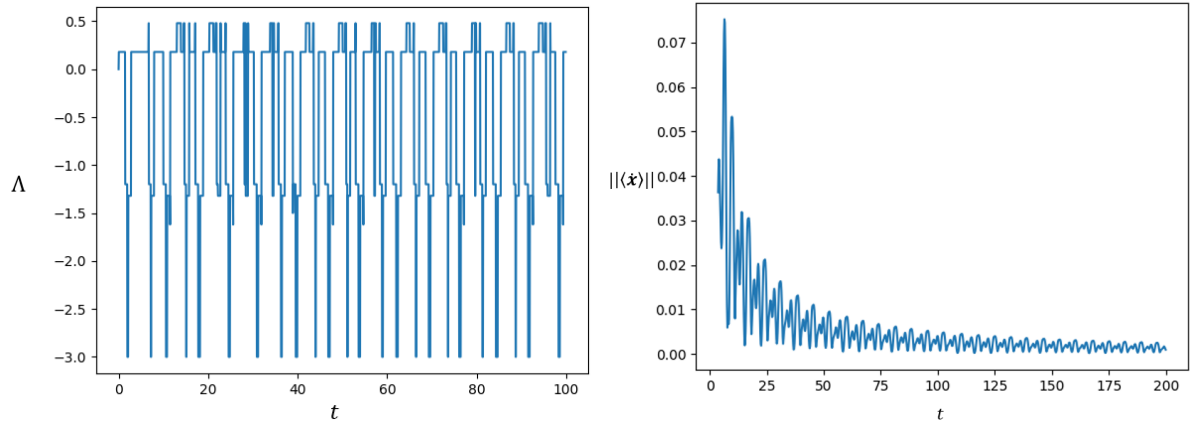


Figura 3.11: Retrato de fase del modelo con función de activación $\text{ReLU}(x)$.



(a) Comportamiento de $\Lambda = \nabla \cdot \mathbf{f}$ del modelo con función de activación PWL.

(b) Norma de la velocidad media de las variables de estado del modelo (3.23).

Figura 3.12: Valor de Λ con respecto al tiempo del modelo con función de activación PWL.

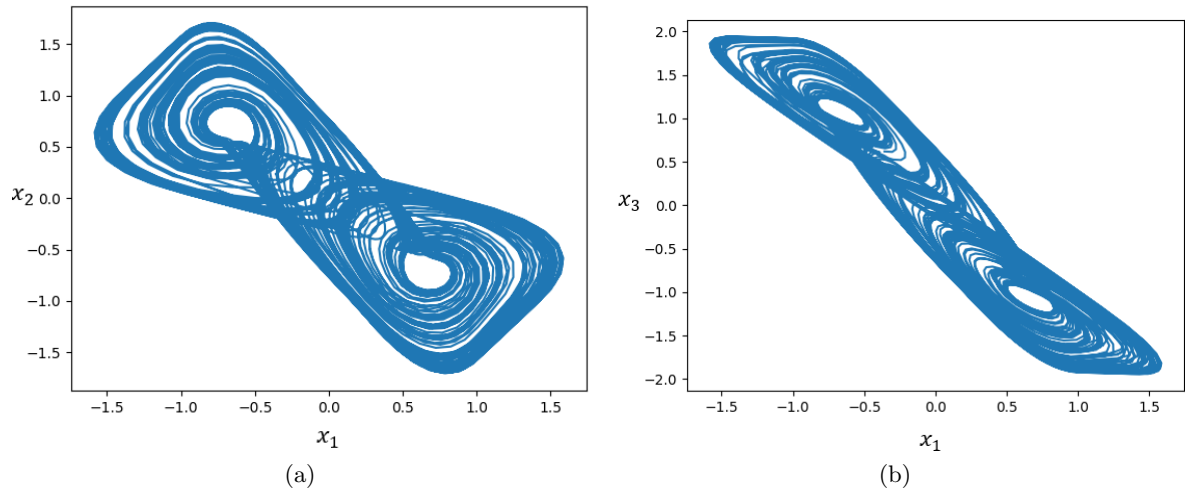


Figura 3.13: Retrato de fase del modelo con función de activación PWL para planos a) $x_1 - x_2$ y b) $x_1 - x_3$.

este esta íntimamente relacionado con el acotamiento del sistema, para este existen un método mas común que sera presentado en la siguiente sección.

3.5. Análisis de estabilidad en el sentido de Lyapunov

Cuando se habla que un sistema dinámico es estable implica que existe alguna región $0 < D < \mathbb{R}^n$ en espacio fasico la cual contiene la órbita generada por el sistema, mientras que la estabilidad asintótica implica que el sistema no solo esta acotado, sino que se aproxima a un punto en específico en el espacio. Cabe mencionar que el concepto de estabilidad también implica la existencia de al menos un atractor en el sistema, en este sentido, los resultados obtenidos en la sección 3.4 deben coincidir con la estabilidad del sistema. Como es discutido en [38], la estabilidad de la red neuronal Hopfield y extensiones ha sido un tema de investigación desde su concepción, algunos resultados obtenidos requieren del concepto de estabilidad de Lyapunov. La estabilidad de Lyapunov es de suma relevancia en el estudio de sistemas no lineales debido a que por medio de un sencillo criterio es posible definir la estabilidad de un sistema, este es descrito en [41] y requiere de la propuesta de una función candidata de Lyapunov $V : \mathbb{R}^n \rightarrow \mathbb{R}$ definida positiva cuya derivada temporal cumpla ser definida semi-negativa para sistemas estables y definida negativa para sistemas asintóticamente estables. Con el fin entonces de analizar la estabilidad de nuestro sistema se desarrollaron los siguientes dos teoremas, el primero especifica una condición para la estabilidad asintótica de la HNN, mientras que el segundo demuestra el acotamiento del sistema para alguna región en el espacio fásico.

Teorema 3.5.1. *El modelo de red neuronal Hopfield de n neuronas sin bias descrito por*

$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{f}(\mathbf{x}), \quad (3.45)$$

donde $C \in \mathbb{R}^{n \times n}$ es una matriz diagonal definida positiva, $W \in \mathbb{R}^{n \times n}$ la matriz de pesos y $\mathbf{f}(\mathbf{x})$ un vector con la función de activación del modelo la cual cumple ser al menos creciente, impar y $|\mathbf{f}(\mathbf{x})| \leq |\mathbf{x}|$, es asintóticamente estable si los elementos de la diagonal principal de la matriz de pesos son cero y a su vez esta es semi-definida negativa.

Demostración. Sea una función candidata de Lyapunov $V(\mathbf{x})$ definida como

$$V(\mathbf{x}) = \frac{1}{2} \mathbf{f}(\mathbf{x})^T \mathbf{f}(\mathbf{x}), \quad (3.46)$$

evidentemente definida positiva con derivada temporal descrita por

$$\begin{aligned}
 \dot{V}(\mathbf{x}) &= \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} (-C\mathbf{x} + W\mathbf{f}(\mathbf{x})) \\
 &= -\mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} C\mathbf{x} + \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} W\mathbf{f}(\mathbf{x}) \\
 &\leq -|\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} C|\mathbf{x}| + |\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} W|\mathbf{x}| \\
 &= -|\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} (C - W) |\mathbf{x}|,
 \end{aligned} \tag{3.47}$$

por definición sabemos que la derivada de una función estrictamente creciente es mayor a cero, de esta manera podemos concluir que la ultima expresión en 3.47 es una función definida negativa si se cumple $C - W > \mathbf{0}$, ahora consideremos a $-W$ como una matriz semi-definida positiva con los elementos en su diagonal iguales a cero, esto indica que la matriz $C + (-W)$ definida por

$$C + (-W) = \begin{bmatrix} c_1 & -w_{12} & \cdots & -w_{1n} \\ -w_{21} & c_2 & \cdots & -w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1} & -w_{n2} & \cdots & c_n \end{bmatrix} > \mathbf{0}, \tag{3.48}$$

es forzosamente definida positiva y por lo tanto el sistema de redes neuronales Hopfield, bajo estas condiciones, es asintóticamente estable. $\square$

Este resultado es comúnmente hallado en la literatura para asegurar el comportamiento asintóticamente estable de la red neuronal Hopfield, dado que nuestro interés es que la HNN presente periodicidad, por lo tanto, las matrices de pesos en nuestro sistema deben tener al menos un peso en la diagonal de la matriz de pesos distinto de cero y no ser definida negativa o, lo que es equivalente, no ser simétrica. En base a este resultado, más un trabajo numérico, fueron hallados las matrices de pesos para los modelos propuestos 3.5 y 3.18. Notemos que este teorema aplica para todos los modelos, incluyendo al modelo 3.15 con función sigmoide debido a su transformación y el modelo con función signo debido a que esta puede ser aproximada con una función tangente hiperbólica como se muestra en 3.17, con excepción del modelo con función de activación $\text{ReLU}(\cdot)$ debido al bias aplicado, en este sentido, el teorema 3.5.1 solo aplica para modelos autónomos.

Teorema 3.5.2. *El modelo de red neuronal Hopfield de n neuronas sin bias descrito por*

$$\dot{\mathbf{x}} = -\mathbf{x} + W \tanh \mathbf{x}, \tag{3.49}$$

esta acotada en una región $D \subset \mathbb{R}^n$ donde se cumple la desigualdad

$$\sum_{i=1}^n x_i^2 \geq \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \tag{3.50}$$

Demostración. Sea una función candidata de Lyapunov $V(\mathbf{x})$ definida como

$$V(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n x_i^2, \quad (3.51)$$

entonces su derivada temporal esta dada por

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \sum_{i=1}^n x_i \dot{x}_i \\ &= -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n x_i \sum_{j=1}^n w_{ij} \tanh x_j \\ &= -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i \tanh x_j \\ &\leq -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \end{aligned} \quad (3.52)$$

La última desigualdad se cumple debido a que $|x| \geq \tanh(x)$, reescribiendo se tiene

$$\dot{V}(\mathbf{x}) \leq -2V(\mathbf{x}) + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \quad (3.53)$$

Consideremos a $D_0 \geq 0$ una región suficientemente grande. Para todo punto $\mathbf{x}$ que satisface $V(\mathbf{x}) = D$ con $D > D_0$ donde existe la desigualdad

$$2V(\mathbf{x}) \geq \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|, \quad (3.54)$$

entonces en la superficie $\{\mathbf{x} | V(\mathbf{x}) = D\}$ se cumple $\dot{V}(\mathbf{x}) \leq 0$ y consecuentemente el conjunto definido por $\{\mathbf{x} | V(\mathbf{x}) \leq D\}$ es una región acotada con todas las soluciones del modelo de la HNN. $\square$

La demostración anterior requiere de la desigualdad $\tanh x \leq |x|$, considerando las funciones de activación estudiadas en este trabajo es evidente que la desigualdad $\text{satlin}(x) \leq |x|$ también se cumple y por lo tanto el modelo 3.23 cumple el mismo teorema. La función $\sigma(x)$ no cumple la misma desigualdad sin embargo puede ser expresada en términos de la tangente hiperbólica y su modelo fue obtenido usando como base otro con la misma estructura mostrada en el teorema en cuestión, por lo tanto, podemos afirmar que el modelo 3.15 esta acotado para alguna región en el espacio fásico. Por otro lado, la función de activación $\text{sign}(x)$ es siempre menor o igual a uno, en este sentido el resultado del teorema 3.5.2 es ligeramente afectado pero sigue existiendo un conjunto D para el cual el sistema esta acotado. Finalmente para la función $\text{ReLU}(x)$ se tiene la siguiente demostración desarrollada en [39], consideremos una función de Lyapunov $V(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2$, se tiene que su derivada es

$$\begin{aligned} \dot{V}(\mathbf{x}) &= -\mathbf{x}^T C |\mathbf{x}| + \mathbf{x}^T W \text{ReLU}(\mathbf{x}) + \mathbf{x}^T \mathbf{b} \leq -\mathbf{x}^T C |\mathbf{x}| + \mathbf{x}^T W |\mathbf{x}| + \|\mathbf{b}\| \cdot \|\mathbf{x}\| \\ &\leq -(\|C\| - \|W\|) \mathbf{x}^2 + \|\mathbf{b}\| \cdot \|\mathbf{x}\|, \end{aligned} \quad (3.55)$$

el lado derecho de la expresión anterior no es definida negativa, supongamos que la solución del modelo con función de activación $\text{ReLU}(x)$ es $\Gamma(t)$. La derivada $\dot{V}(\mathbf{x})$ es negativa fuera del conjunto $\|\mathbf{x} - \Gamma\| \leq \|\mathbf{b}\|/\|\mathbf{W}\| - 1$, con $\|\mathbf{b}\|^2/2(\|\mathbf{W}\| - 1)^2 < \eta$, la solución iniciando en el conjunto $V(\mathbf{x}) \leq \eta$ permanecerá en el para todo tiempo futuro, dado que la derivada de la función de Lyapunov es negativa en la frontera $V(\mathbf{x}) \leq \eta$, consecuentemente la solución esta acotada por la frontera $\sqrt{(2\eta)}$.

3.6. Análisis de estabilidad alrededor de los puntos de equilibrio

Una vez demostrada la existencia de atractores en el sistema dinámico es importante ser capaz de caracterizar el comportamiento del modelo alrededor de ellos, de esta manera es necesario realizar un análisis alrededor de los puntos de equilibrio en cada uno de los modelos presentados. Como fue mencionado en la introducción de este trabajo, los puntos de equilibrio de un sistema dinámico son hallados al resolver

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{0} , \quad (3.56)$$

que representan los puntos para los cuales el sistema se mantiene estático. Para los casos específicos del modelo de redes neuronales Hopfield, los puntos de equilibrio $\tilde{\mathbf{x}}$ se calculan resolviendo el sistema

$$c_i x_i^* = \sum_{j=1}^n w_{ij} f(x_j^*) + b_i , \quad (3.57)$$

lo cual es evidentemente un sistema de ecuaciones no lineales. Para su obtención se desarrolló un programa en Matlab y la función `solve(.)` para la solución de un sistema de ecuaciones realizando un barrido con distintas condiciones iniciales, los resultados obtenidos fueron registrados en la Tabla 3.1. Con respecto a el modelo con función de activación ReLU, este es un sistema no autónomo y su punto de equilibrio es calculado solo para $t = 0$. La caracterización de los puntos de equilibrio encontrados se realiza linealizando el modelo de las HNN's , usando el procedimiento descrito en [41], y evaluando los eigenvalores obtenidos por

$$\det \left(I\lambda - \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \right) = 0 , \quad (3.58)$$

Tabla 3.1: Puntos de equilibrio para los distintos modelos de redes neuronales.

Función de activación	Puntos de equilibrio
Tangente hiperbólica	$\begin{bmatrix} -1.7801 & 0.2753 & 0.1390 \\ 0.0000 & 0.0000 & 0.0000 \\ 1.7801 & -0.2753 & -0.1390 \end{bmatrix}$
Sigmoide	$\begin{bmatrix} -0.0002 & -0.0152 & 0.0131 \\ -0.8383 & -0.7950 & 5.5905 \\ 0.8494 & 0.7688 & -5.6351 \end{bmatrix}$
ReLU	$\begin{bmatrix} 0.6354 & 0.7316 \end{bmatrix}$
PWL	$\begin{bmatrix} 0.7093 & -0.7119 & -1.2408 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.7093 & 0.7119 & 1.2408 \end{bmatrix}$
Signo	$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$

donde $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$ es el jacobiano del sistema dinámico. De manera generalizada para el modelo de redes neuronales Hopfield los eigenvalores se obtienen por

$$\begin{vmatrix} \lambda + c_1 - w_{11}f'(\tilde{x}_1) & -w_{12}f'(\tilde{x}_2) & \cdots & -w_{1n}f'(\tilde{x}_n) \\ -w_{21}f'(\tilde{x}_1) & \lambda + c_2 - w_{22}f'(\tilde{x}_2) & \cdots & -w_{2n}f'(\tilde{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1}f'(\tilde{x}_1) & -w_{n2}f'(\tilde{x}_2) & \cdots & \lambda + c_n - w_{nn}f'(\tilde{x}_n) \end{vmatrix} = 0, \quad (3.59)$$

donde $f'(\tilde{x}_i)$ representa la derivada de la función de activación evaluada en el punto de equilibrio $\tilde{x}_i$ para todo $i = (1, 2, \dots, n)$. Consideremos el caso del modelo de HNN para 3 neuronas, desarrollando posible encontrar

$$\begin{aligned} \lambda^3 + \lambda^2 \sum_{i=1}^3 (c_i - f'(\tilde{x}_i)w_{ii}) + \lambda \left[f'(\tilde{x}_1)f'(\tilde{x}_2) \begin{vmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{vmatrix} + f'(\tilde{x}_1)f'(\tilde{x}_3) \begin{vmatrix} w_{11} & w_{13} \\ w_{31} & w_{33} \end{vmatrix} \right] \\ + \lambda \left[f'(\tilde{x}_2)f'(\tilde{x}_3) \begin{vmatrix} w_{22} & w_{23} \\ w_{32} & w_{33} \end{vmatrix} + c_1c_2 + c_1c_3 + c_2c_3 + \sum_{i=1}^3 \sum_{\substack{j=1 \\ i \neq j}}^3 c_i f'(\tilde{x}_j)w_{jj} \right] \\ + c_1c_2c_3 - |W| \prod_{i=1}^3 f'(\tilde{x}_i) = 0, \quad (3.60) \end{aligned}$$

$$\Rightarrow \lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0, \quad (3.61)$$

donde

$$\begin{aligned}
a_0 &= c_1 c_2 c_3 - |W| \prod_{i=1}^3 f'(\tilde{x}_i), \\
a_1 &= \left[f'(\tilde{x}_1) f'(\tilde{x}_2) \begin{vmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{vmatrix} + f'(\tilde{x}_1) f'(\tilde{x}_3) \begin{vmatrix} w_{11} & w_{13} \\ w_{31} & w_{33} \end{vmatrix} \right] \\
&\quad + \left[f'(\tilde{x}_2) f'(\tilde{x}_3) \begin{vmatrix} w_{22} & w_{23} \\ w_{32} & w_{33} \end{vmatrix} + c_1 c_2 + c_1 c_3 + c_2 c_3 + \sum_{i=1}^3 \sum_{j=1, i \neq j}^3 c_i f'(\tilde{x}_j) w_{jj} \right], \\
a_2 &= \sum_{i=1}^3 (c_i - f'(\tilde{x}_i) w_{ii}),
\end{aligned}$$

de este resultado se pueden obtener distintas conclusiones con respecto a la posición de los eigenvalores en el plano complejo, dependiendo de esta es posible separarlos en 8 categorías [42] las cuales se resumen la Tabla 3.2, cada uno de estos puntos tiene un comportamiento esperado como se muestra en la Figura . Es sabido que para sistemas caóticos autoexcitados, estos son, sistemas cuyos atractores están localizados en los puntos de equilibrio del sistema dinámico, como lo son los trabajados en esta tesis, los puntos equilibrio en donde se forman los scrolls son de tipo índice II, entre ellos se encuentra un punto de equilibrio de índice I, el cual conecta ambos scrolls [43]. Retomando la expresión 3.60 y de acuerdo al criterio de Routh-Hurwitz las partes reales de esta expresión son positivas, y por ende el punto de equilibrio es índice I si se cumple

$$\begin{aligned}
a_2 &> 0 \\
a_2 a_1 - a_0 &> 0 \quad . \\
a_0 &> 0
\end{aligned} \tag{3.62}$$

Los eigenvalores calculados para cada uno de los modelos con sistemas autónomos son resumidos en las tablas 3.3, 3.4, 3.5 y 3.6 junto a su clasificación en base al trabajo [42]. Con regularidad cuando se explora la dinámica de algún modelo no lineal es buena práctica analizar también los parámetros cercanos a los estudiados, en este sentido la mencionada Tabla analiza los eigenvalores variando algunos parámetros de los modelos.

En base a los resultados obtenidos se puede afirmar que los modelos con funciones de activación tangente hiperbólica, PWL y sigmoide presentan un comportamiento esperado alrededor de sus puntos de equilibrio con respecto a lo mencionado en [43]. El modelo con una función de activación signo sin embargo presenta un comportamiento interesante en su punto de equilibrio trivial, es justo en $x = 0$ donde la función tiene un cambio abrupto y como tal no tiene derivada en este punto, es por esto que se aproximó la función signo con $\tanh(400x)$, lo cual tiene como derivada $400(1 - \tanh^2(400x))$ y por lo tanto el eigenvalor obtenido es un aproximado, sin embargo, el principal interés es la posición del eigenvalor en el plano complejo, este es clasificado como un punto silla de índice-1. Finalmente para el sistema no autónomo el punto de equilibrio solo tiene sentido para $t = 0$ y para este caso los eigenvalores hallados fueron $0.075 \pm 0.9909i$,

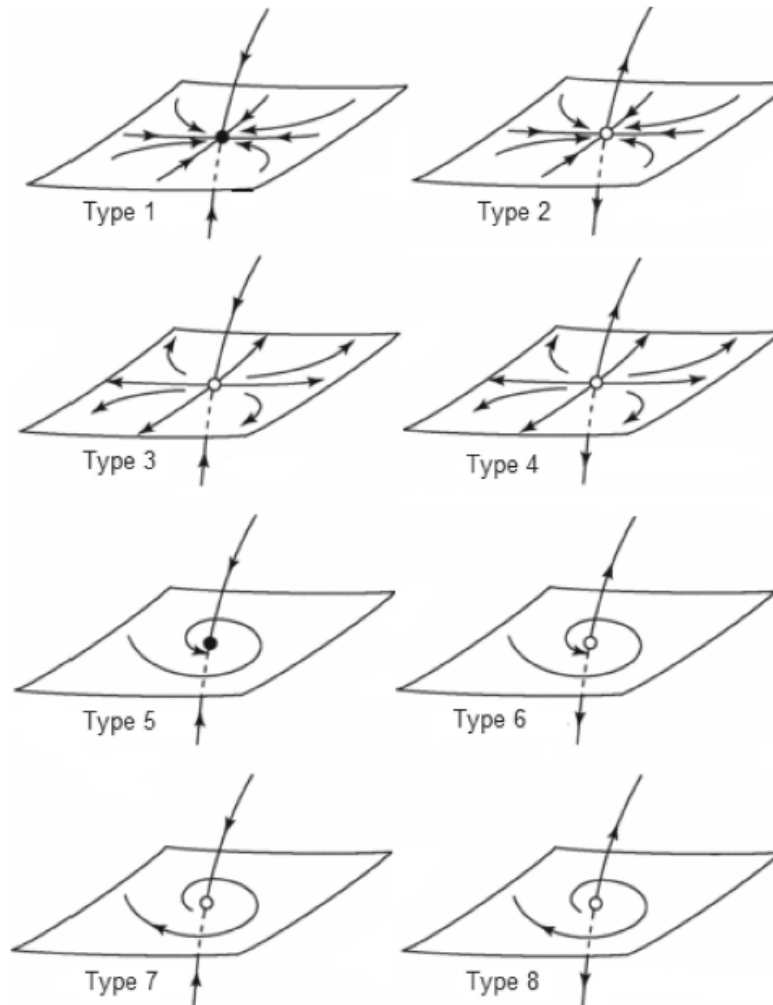


Figura 3.14: Tipos de puntos de equilibrio dependiendo de su categoría [42].

Tabla 3.2: Categorías de los punto de equilibrio en un sistema caótico en base a los eigenvalores hallados en un sistema de 3 dimensiones en base al trabajo [42].

Eigenvalores	Categoría
Tres negativos reales	Nodo índice-0
Dos negativos y un negativo todos reales	Punto silla índice-1
Dos positivos y un negativo todos reales	Punto silla índice-2
Tres positivos reales	Repulsor de índice-3
Un real negativo y un complejo conjugado con parte real negativa	Nodo espiral índice-0
Un real positivo y un complejo conjugado con parte real negativa	Punto silla espiral índice-1
Un real negativo y un complejo conjugado con parte real positiva	Punto silla espiral índice-2
Un real positivo y un complejo conjugado con parte real positiva	Repulsor espiral índice-3

Tabla 3.3: Eigenvalores hallados para el modelo con función tangente hiperbólica y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
$\begin{bmatrix} -1.780 & 0.275 & 0.139 \\ 1.780 & -0.275 & -0.139 \end{bmatrix}$	w_{12}	1.5	$-0.939 \quad 0.3613 \pm 2.317i$	Punto silla espiral de índice-2
		2.5	$-0.898 \quad 0.341 \pm 2.328i$	
		3.5	$-0.857 \quad 0.320 \pm 2.339i$	
		4.5	$-0.816 \quad 0.3003 \pm 2.350i$	
$[0, 0, 0]$	w_{12}	1.5	$2.000 \quad -1.000 \pm 1.224i$	Punto silla espiral de índice-1
		2.5	$2.000 \quad -1.000 \pm 1.581i$	
		3.5	$2.000 \quad -1.000 \pm 1.870i$	
		4.5	$2.000 \quad -1.000 \pm 2.121i$	
$\begin{bmatrix} -1.780 & 0.275 & 0.139 \\ 1.780 & -0.275 & -0.139 \end{bmatrix}$	w_{13}	6	$-0.9222 \quad 0.3529 \pm 2.3911i$	Punto silla espiral de índice-2
		7	$-0.9211 \quad 0.3524 \pm 2.3686i$	
		8	$-0.9199 \quad 0.3518 \pm 2.3460i$	
		9	$-0.9187 \quad 0.3512 \pm 2.3230i$	
		10	$-0.9174 \quad 0.3505 \pm 2.2999i$	
$[0, 0, 0]$	w_{13}	6	$1.0000 \quad -0.5000 \pm 1.6583i$	Punto silla espiral de índice-1
		7	$1.3788 \quad -0.6894 \pm 1.5575i$	
		8	$1.7100 \quad -0.8550 \pm 1.4809i$	
		9	$2.0000 \quad -1.0000 \pm 1.4142i$	
		10	$2.2583 \quad -1.1291 \pm 1.3509i$	

Tabla 3.4: Eigenvalores hallados para el modelo con función sigmoide y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
[-0.0002 -0.0152 0.0131]	w_{11}	7.77	1.7231 -0.0766 ± 1.8146i	Punto silla espiral de índice-1
		7.97	1.7373 -0.0587 ± 1.8073i	
		8.17	1.7519 -0.0410 ± 1.7998i	
[-0.8383, -0.7950, 5.5905]	w_{11}	7.77	-0.9709 0.5429 ± 1.3325i	Punto silla espiral de índice-2
		7.97	-0.9711 0.5641 ± 1.3310i	
		8.17	-0.9714 0.5853 ± 1.3291i	
[0.8494 0.7688 -5.6351]	w_{11}	7.77	-0.9722 0.5468 ± 1.3371i	Punto silla espiral de índice-2
		7.97	-0.9724 0.5679 ± 1.3358i	
		8.17	-0.9726 0.5890 ± 1.3341i	
[-0.0002 -0.0152 0.0131]	w_{22}	8.73	1.7321 -0.0686 ± 1.8100i	Punto silla espiral de índice-1
		8.78	1.7347 -0.0637 ± 1.8087i	
		8.83	1.7373 -0.0587 ± 1.8073i	
		8.88	1.7399 -0.0538 ± 1.8060i	
[-0.8383, -0.7950, 5.5905]	w_{22}	8.73	-0.9710 0.5533 ± 1.3300i	Punto silla espiral de índice-2
		8.78	-0.9711 0.5587 ± 1.3305i	
		8.83	-0.9711 0.5641 ± 1.3310i	
		8.88	-0.9712 0.5695 ± 1.3314i	
[0.8494 0.7688 -5.6351]	w_{22}	8.73	-0.9723 0.5570 ± 1.3350i	Punto silla espiral de índice-2
		8.78	-0.9723 0.5625 ± 1.3354i	
		8.83	-0.9724 0.5679 ± 1.3358i	
		8.88	-0.9725 0.5734 ± 1.3362i	

Tabla 3.5: Eigenvalores hallados para el modelo con función PWL y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
[0.7093 -0.7119 -1.2408] [-0.7093 0.7119 1.2408]	w_{11}	1.58	-1.0000 0.6900 ± 4.5345i	Punto silla espiral de índice-2
		1.68	-1.0000 0.7400 ± 4.5355i	
		1.78	-1.0000 0.7900 ± 4.5358i	
[0, 0, 0]	w_{11}	1.58	1.8973 -0.9087 ± 1.5553i	Punto silla espiral de índice-1
		1.68	1.8324 -0.8262 ± 1.4464i	
		1.78	1.7552 -0.7376 ± 1.3198i	
[0.7093 -0.7119 -1.2408] [-0.7093 0.7119 1.2408]	w_{33}	-0.2	-1.0000 0.7400 ± 4.5355i	Punto silla espiral de índice-2
		-0.3	-1.0000 0.7400 ± 4.5355i	
		-0.4	-1.0000 0.7400 ± 4.5355i	
[0, 0, 0]	w_{33}	-0.2	2.0477 -0.8838 ± 1.6532i	Punto silla espiral de índice-1
		-0.3	1.8324 -0.8262 ± 1.4464i	
		-0.4	1.5595 -0.7398 ± 1.1656i	

Tabla 3.6: Eigenvalores hallados para el modelo con función signo y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$	w_{32}	3.5	$-1.0000 \quad 0.3230 \pm 0.9996i$	Punto silla espiral de índice-2
		4	$-1.0000 \quad 0.3230 \pm 1.1799i$	
		4.5	$-1.0000 \quad 0.3230 \pm 1.3360i$	
$[0, 0, 0]$	w_{32}	3.5	$6.3440 \quad -3.1720 \pm 5.5394i$	Punto silla espiral de índice-1
		4	$6.3137 \quad -3.1569 \pm 5.5585i$	
		4.5	$6.2834 \quad -3.1417 \pm 5.5777i$	
$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$	w_{33}	2.5	$-0.8367 \quad 0.0208 \pm 1.3292i$	Punto silla espiral de índice-2
		3	$-0.8317 \quad 0.2388 \pm 1.0912i$	
		3.5	$-0.8260 \quad 0.4564 \pm 0.7170i$	
$[0, 0, 0]$	w_{33}	2.5	$6.0976 \quad -3.2988 \pm 5.5989i$	Punto silla espiral de índice-1
		3	$6.3137 \quad -3.1569 \pm 5.5585i$	
		3.5	$6.5413 \quad -3.0207 \pm 5.5131i$	

lo cual indica que es **foco inestable**.

Capítulo 4

Dinámica caótica

La dinámica de los sistemas no lineales puede presentar fenómenos más complejos en comparación con los sistemas lineales, una de estas complejidades es denominada caos. Desde su descubrimiento varias métricas para la identificación de caos en sistemas dinámicos han sido desarrolladas, las más comunes siendo los diagramas de bifurcación y exponentes de Lyapunov. En esta sección se utilizarán dichas métricas para identificar los distintos comportamientos dinámicos (no solo caóticos) que generan las redes neuronales estudiadas. Estas métricas dejan en evidencia *las ventajas que presentan las nuevas redes neuronales Hopfield propuestas* con respecto a las halladas en la literatura.

4.1. Diagramas de bifurcación

La bifurcación indica la división de una órbita en dos partes, en sistemas no lineales, este término suele ser utilizado para indicar un cambio cualitativo o cuantitativo de un sistema a partir de la variación de alguno de sus parámetros. Este parámetro es llamado como *parámetro de bifurcación*. Este comportamiento cualitativo o cuantitativo del sistema dependiente del parámetro de bifurcación es graficada en el llamado *diagrama de bifurcación*, este nos da información de la dinámica compleja que este presenta.

Gran parte de los sistemas dinámicos que representan algún fenómeno físico presentan algún tipo de periodicidad en su comportamiento, por ejemplo, la rotación y traslación de los planetas, un péndulo ideal sin fricción o un señal senoidal, todos estos fenómenos comparten necesitar solo un ciclo para repetir el mismo comportamiento y como tal se denominan de *periodo-1* o simplemente p-1, en otros casos sin embargo el comportamiento del sistema se repite tras 2 (p-2), 3 (p-3), etc., en la Figura 4.1 se graficó la respuesta de una neurona del modelo (3.5) para los cuales se registraron distintas periodicidades variando uno de sus parámetros, eventualmente si el valor de este parámetro se modifica de manera correcta es posible hallar otros tipo de

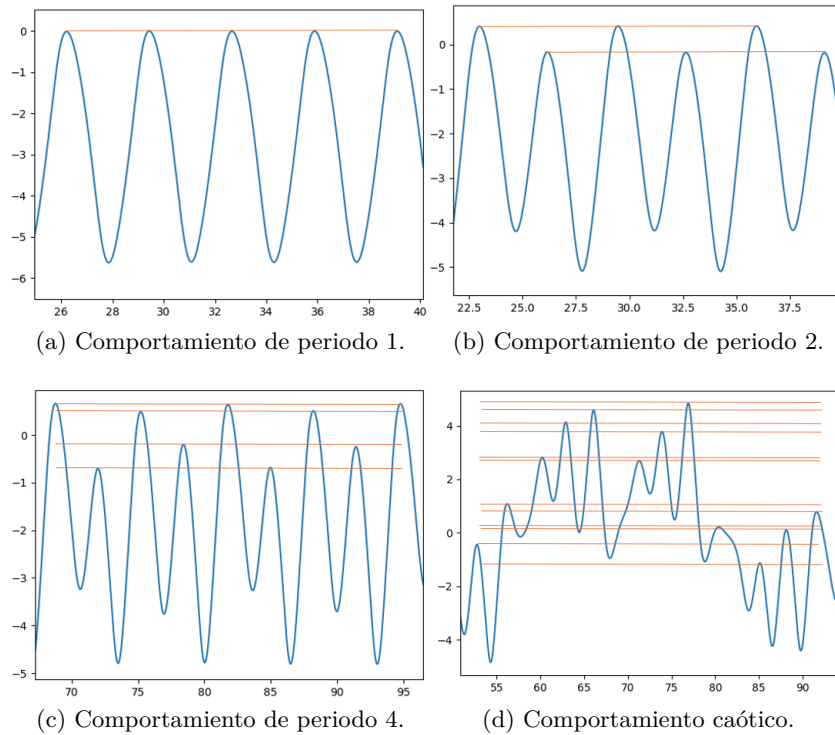


Figura 4.1: Diferentes comportamientos periódicos y caótico para el modelo 3D HNN con función de activación tangente hiperbólica.

periodicidad y eventualmente caos, esta dinámica es conocida como *ruta al caos*.

Para este trabajo se decidió tomar como cambio cualitativo al máximo local que toma una de las orbitas del sistema conforme se modifica alguno de los parámetros del sistema, esto requiere de un programa capaz simular el sistema con alguno de los métodos de integración, un detector de los máximos en la trayectoria con respecto al tiempo y un graficador, este proceso requiere de una gran potencia computacional, por lo que un lenguaje de programación veloz es recomendable, en este escenario C++ podría ser una elección idónea, sin embargo, hoy es considerado un lenguaje de mas bajo nivel además de no tener una gran cantidad de librerías, por lo tanto, se decidió desarrollar el programa en Python debido principalmente a su librería NumPy la cual nos permite utilizar un lenguaje de programación de alto nivel como Python e implementar operaciones directamente en C++.

4.1.1. Diagramas de bifurcación en Python

Para analizar el comportamiento dinámico de las redes neuronales Hopfield fue realizado un código en Python 3.12. El código requiere que primero se definan ciertos argumentos, estos son:

- `prmt`: Es el parámetro de bifurcación, debe estar descrito como un array del tipo Numpy. De preferencia utilizar el método `linspace(start, stop, num)` de Numpy que regresa un array con valores de "start" a "stop" de "num" cantidad de elementos.
- `funcion`: Función que describe el sistema dinámico en forma de un vector de NumPy, los argumentos en esta función son el tiempo y un vector NumPy con las variables de estado. Por ejemplo, para describir el modelo (3.5) se tiene:

```
def fct(x, t, a):
    return np.array([-x[0]+a*tg(x[1])-9*tg(x[2]),
                    -x[1]+tg(x[0])+3*tg(x[1])+3*tg(x[2]),
                    -x[2]-tg(x[0])-3*tg(x[1])])
```

- `initialCondition`: Condición inicial para la integración numérica del sistema en forma de matriz, se aceptan un conjunto de condiciones iniciales, por ejemplo, `initialCondition = [[0.1, 0, 0], [-0.1, 0, 0]]`, en este ejemplo se exploran dos condiciones iniciales distintas. El color de los diagramas de bifurcación cambia con respecto a que condición inicial se explora, es azul para la primer condición inicial, rojo para la segunda, verde para las demás.
- `paso`: Es el paso de integración numérica, esta definida como $1e-2$ de manera predeterminada.
- `porcentaje`: Un sistemas dinámico requiere de cierto tiempo hasta "estabilizarse" al comportamiento que este mantendrá, por lo tanto, es necesario retirar esta sección pasajera del sistema, este argumento indica que porcentaje es considerado para iniciar a graficar los máximos locales del sistema, es decir, si se desea considerar la última tercera parte de las simulaciones realizadas se colocaría un 33 como argumento. El valor predeterminado es de 50.
- `tf`: Tiempo final de las simulaciones realizadas, de manera predeterminada este valor es 100.
- `eje`: Indica el eje que se desea analizar, por ejemplo, si en se tienen tres variables de estado (x, y, z) y se requiere en diagrama de bifurcación de y con respecto a un parámetro el valor del `eje=2`. De manera predeterminada es 1.
- `rest`: La obtención de los diagramas de bifurcación requiere de una gran potencia computacional, esto genera un desgaste en el computador importante y generar un sobrecalentamiento, este argumento permite dar un descanso en segundos entre las iteraciones

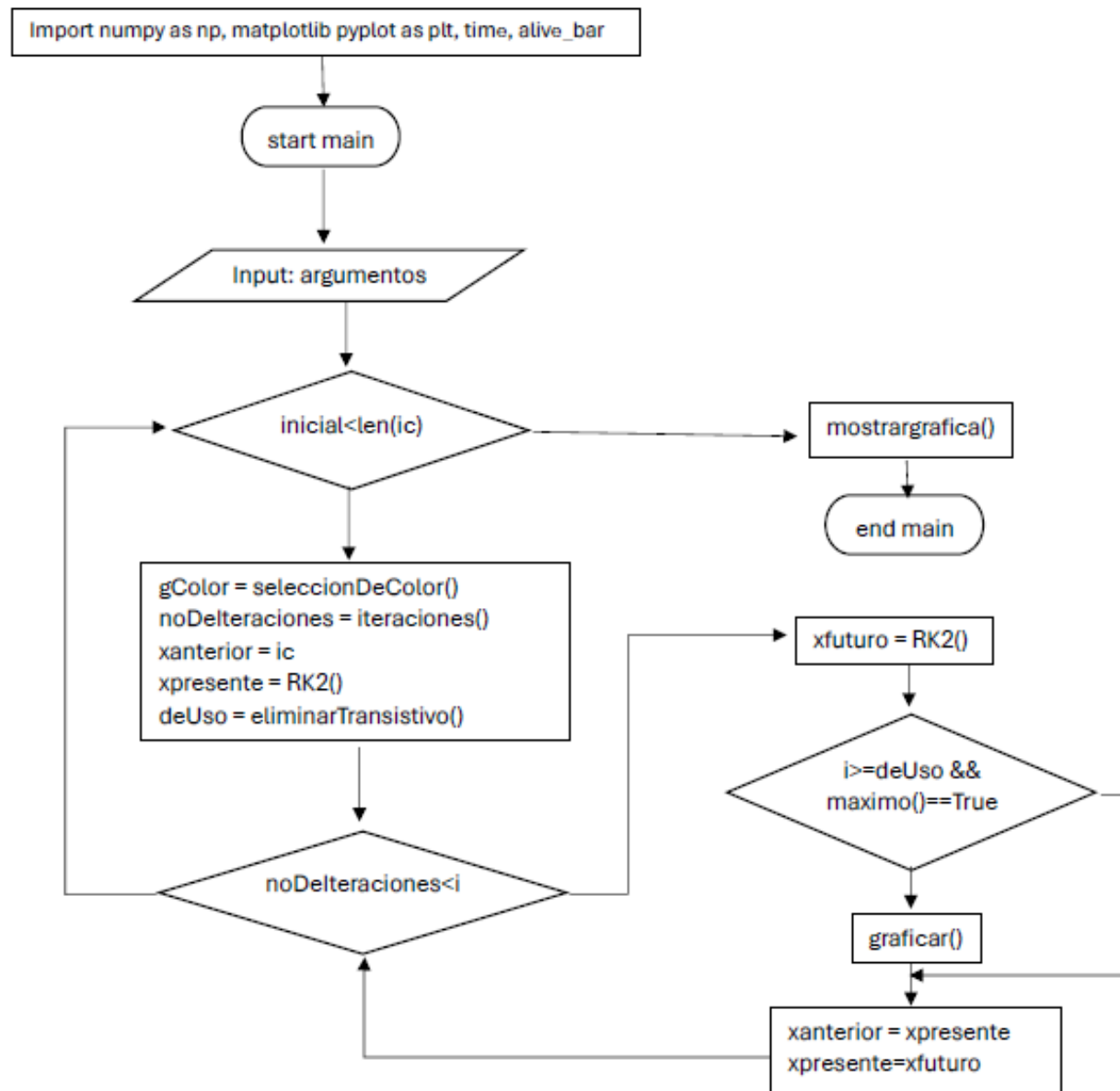


Figura 4.2: Diagrama de flujo para el desarrollo del código para obtener diagramas de bifurcación.

realizadas, por ejemplo, si se desea que se descansa un tiempo de 1ms entre iteraciones se colocaría como argumento $\text{rest} = 1\text{e-}3$.

Como base para este código considerando los argumentos antes descritos se realizó el diagrama de flujo de la Figura 4.2.

El código requiere de dos conceptos bases para su funcionamiento, la detección la simulación del sistema dinámico y la detección del máximo local. Es necesario también considerar la potencia computacional requerida, dado que es importante tomar una gran cantidad de divisiones, en donde es necesario integrar el sistema para todos estos valores del parámetro de interés, hallar y graficar el máximo. Es por esto que se utiliza un método de integración que consume baja cantidad de recursos como lo es método de integración Runge-Kutta 2 y obtener los máximos conforme el sistema evoluciona y no como un proceso posterior a la integración numérica. Para detectar el máximo de una curva el programa almacena tres estados distintos de una curva, el estado anterior x^{**} , presente x^* y futuro x , se detecta un máximo local si se cumple

$$x^* - x^{**} \geq 0 \quad \wedge \quad x - x^* < 0, \quad (4.1)$$

el programa es capaz entonces de detectar las curvas que cumplen esta condición y graficar el valor del máximo hallado.

4.1.2. Diagramas de bifurcación del modelo con función de activación tan-gente hiperbólica

El resultado de los diagramas de bifurcación para el modelo 3.5 es presentado en la Figura 4.4, en esta diversos fenómenos son denotados. Para iniciar, el sistema presenta el fenómeno de *bi-estabilidad*, esto es, con diferentes condiciones iniciales puede tener un comportamiento u otro completamente distinto, más específicamente, en azul se utiliza la condición inicial $[0.1, 0, 0]$ y en rojo la condición inicial $[-0.1, 0, 0]$. De la figura también se observan las distintas periodicidades que el sistema presenta, por ejemplo, en $w_{12} = -5$ se tiene una periodicidad 2, en $w_{12} = -3.7$ se tiene una periodicidad de 5, entre otras. En algunos casos las regiones para las cuales se presenta algún tipo de periodicidad es muy pequeña, por ejemplo, con un paso de integración de 0.0025 se encontró un periodo tres en la región $-2.296 < w_{12} < -2.267$, mientras que para un paso de integración de 0.001 se encontró esta región en $-2.247 < w_{12} < -2.225$, por lo que para regiones tan pequeñas es difícil determinar su periodicidad exacta. Además de la periodicidad del sistema, los diagramas de bifurcación nos ofrecen las zonas en donde el sistema se siente atraído por un solo atractor o múltiples, por ejemplo, variando el parámetro w_{12} en la zona $4.05 < w_{12} < 4.34$ el sistema solo muestra un scroll y se evidencia en el diagrama de bifurcación y la Figura 4.3. Por cuál de los dos scrolls el sistema se siente atraído es dependiente de la condición inicial del sistema y del valor del parámetro de bifurcación, en la Figura 4.3 se

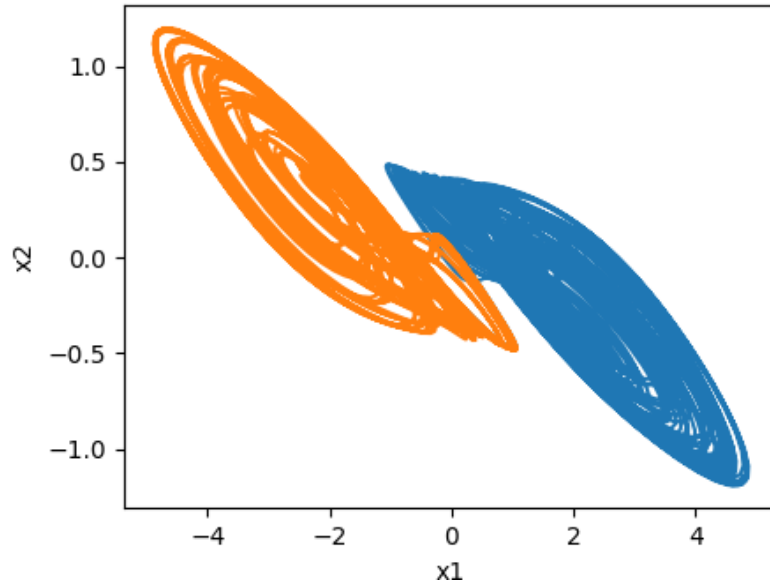


Figura 4.3: Comportamiento del sistema con un solo atractor usando $w_{12} = 4.05$ en azul y $w_{12} = 4.06$ en naranja.

simula el sistema con $w_{12} = 4.05$ en azul y $w_{12} = 4.06$ en naranja, en donde se evidencia esta descripción.

4.1.3. Diagramas de bifurcación del modelo con función de activación sigmoide

Por otro lado, la Figura 4.5 presenta el diagrama de bifurcación variando los parámetros a) w_{11} y b) w_{22} del modelo 3.15, a diferencia del modelo anterior, este sistema no presenta bi-estabilidad, así como también tiene una notoria reducción en espacio en donde el sistema presenta caos, además de presentar una menor cantidad de fenómenos en general. Nótese que existen tres zonas con caos en el sistema variando w_{11} , para la región contenida en $7.968 < w_{11} < 7.974$ existe un doble scroll, mientras que para las otras zonas el sistema solo es atraído por un atractor. Esta característica es compartida por el segundo parámetro w_{22} . Su limitada región caótica dificulta su implementación en circuitos electrónicos.

4.1.4. Diagramas de bifurcación del modelo con función de activación ReLU

Para el modelo con función de activación ReLU los parámetros de bifurcación no son los pesos sino la amplitud y frecuencia de la señal externa senoidal insertada en la primera neurona. Tras su análisis se encontró el fenómeno de bi-estabilidad en el sistema así como una amplia gama de periodicidades en el sistema. Por un lado, al variar la amplitud de la señal externa se encuentran distintas zonas periódicas y caóticas con amplitud proporcional a la señal externa.

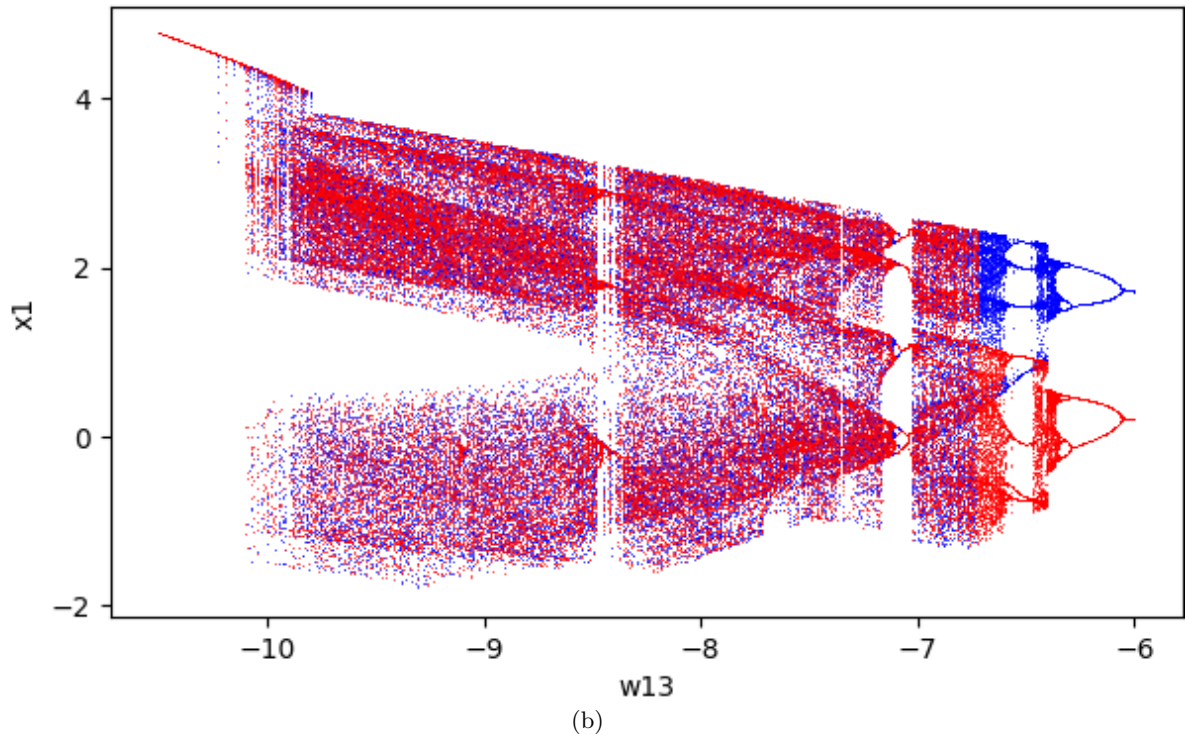
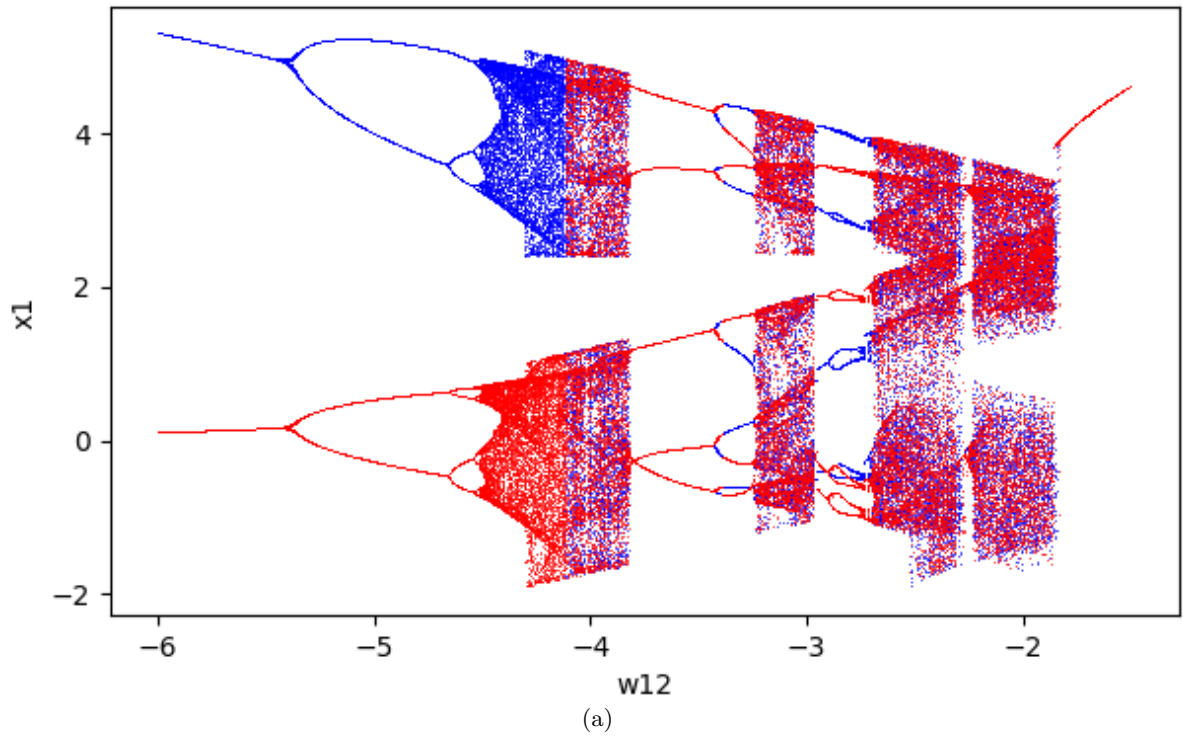


Figura 4.4: Diagramas de bifurcación para el modelo 3.5 con dos condiciones iniciales, $[0.1, 0, 0]$ y $[-0.1, 0, 0]$ en azul y rojo respectivamente, variando el parámetro en a) $\beta = w_{12}$ de 1.5 a 6, y b) el parámetro $\beta = w_{13}$ se varía de 5 a 10.

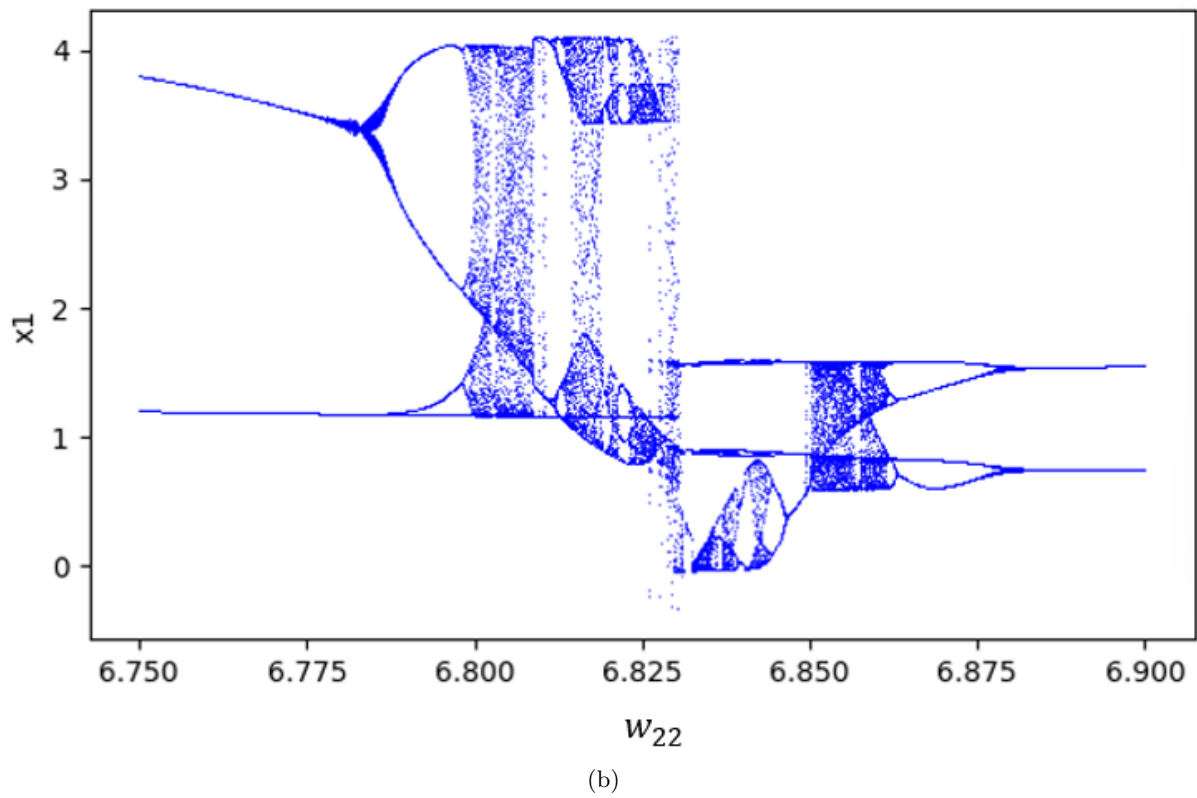
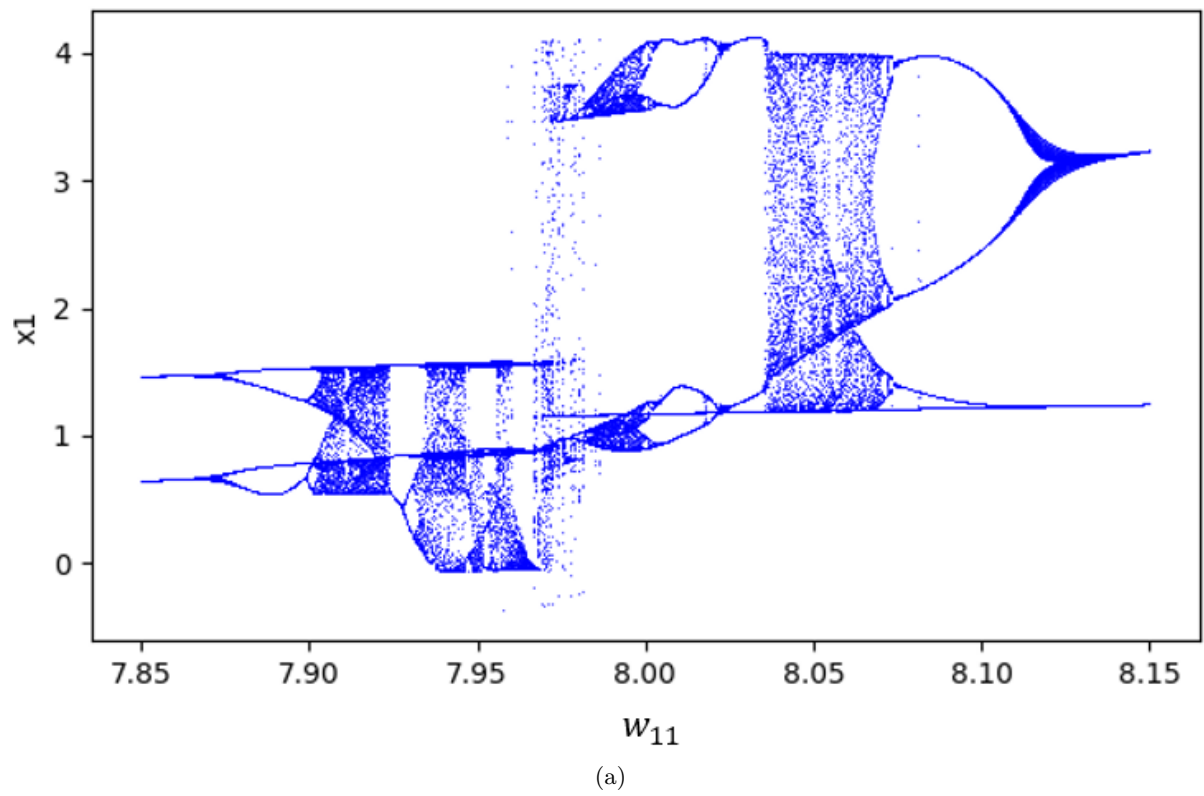


Figura 4.5: Diagramas de bifurcación para el modelo con función de activación sigmoide.

Por otro lado, al variar la frecuencia de la señal externa se presenta una gran variedad de periodicidades así como regiones caóticas, para una amplia cantidad de frecuencias existe en apariencia caos, son en realidad el resultado de la frecuencia tan alta de la señal externa aplicada al sistema, para este caso, las zonas caóticas son identificadas de manera apropiada por los exponentes de Lyapunov hallados en la sección 4.2.

4.1.5. Diagramas de bifurcación del modelo con función de activación PWL

Para el caso del modelo con función de activación $\text{satlin}(x)$ se detectó de nueva manera bi-estabilidad en el sistema y se evidencia en los diagramas de bifurcación de la Figura 4.7, de color azul se utilizó la condición inicial $(-0.1, 0, 0)$ y en rojo $(0.1, 0, 0)$, notemos de los diagramas de bifurcación que las regiones de periodicidad son muy pequeñas.

4.1.6. Diagramas de bifurcación del modelo con función de activación signo

Finalmente, para el modelo propuesto con función de activación signo se analizaron dos parámetros w_{32} y w_{33} , el diagrama de bifurcación de w_{32} es presentado en la Figura 4.8 a), en ella se introdujeron dos condiciones iniciales $(-0.1, 0, 0)$ y $(0.1, 0, 0)$ de color azul y rojo respectivamente, de aquí se evidencia que para valores de w_{32} mayores a 6.58 el sistema solo se siente atraído por uno de sus atractores. Por su parte, la variación del parámetro w_{33} con diagrama de bifurcación en la Figura 4.8 b) presenta una menor cantidad de fenómenos dinámicos en un menor rango. Un aspecto que resulta interesante sobre este modelo es el peso w_{12} , fue hallado que **el sistema presenta caos para todo $w_{12} \in \mathbb{R}_+$** , este es el comportamiento observado en la Figura 4.9 y donde también queda en evidencia que la modificación de este parámetro solo aumenta la magnitud del sistema.

4.2. Exponentes de Lyapunov

Otra de las métricas más relevantes y usadas para la identificación de caos son los exponentes de Lyapunov (LE), su uso nace de la propiedad más característica de los sistemas caóticos, su extrema sensibilidad a las condiciones iniciales, en este sentido, los exponentes de Lyapunov miden la divergencia exponencial entre dos orbitas infinitesimalmente cercanas. Para su definición imaginemos un estado $\mathbf{x}(t)$ de un sistema dinámico caótico que evoluciona en el tiempo. En el momento $t = 0$ creamos otra trayectoria perturbando la original con una pequeña perturbación $\mathbf{y}$, con $\delta = \|\mathbf{y}\|$. Si la dinámica es caótica, el original y las trayectorias perturbadas se separan cada vez más en el tiempo a medida que evolucionan. Para pequeños perturbaciones, esta separación está ocurriendo aproximadamente exponencialmente rápido. Después en algún

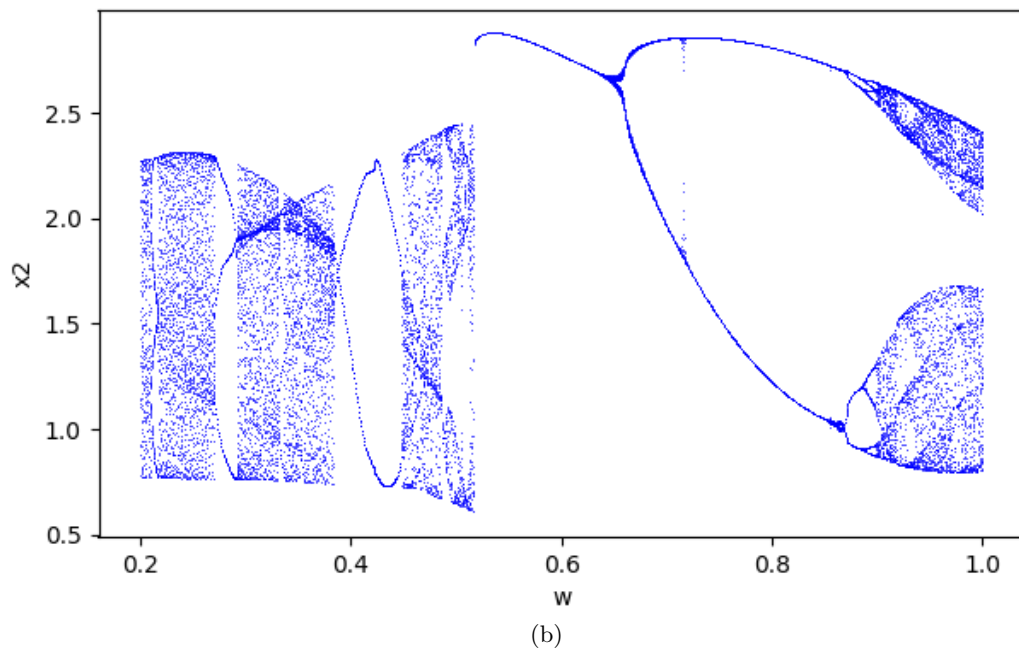
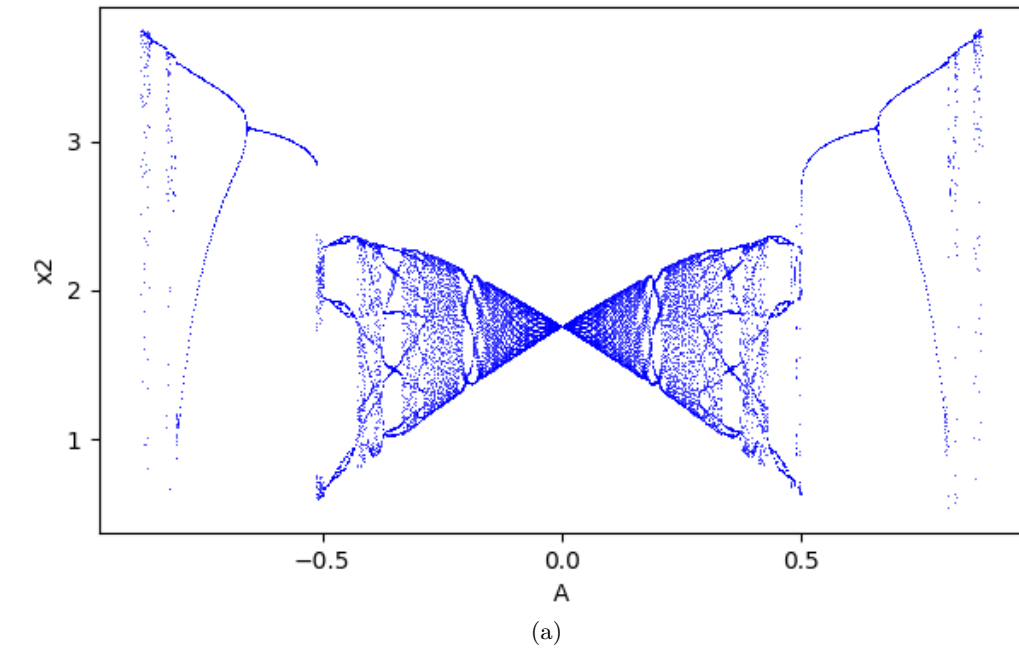


Figura 4.6: Diagramas de bifurcación para el modelo con función de activación ReLU.

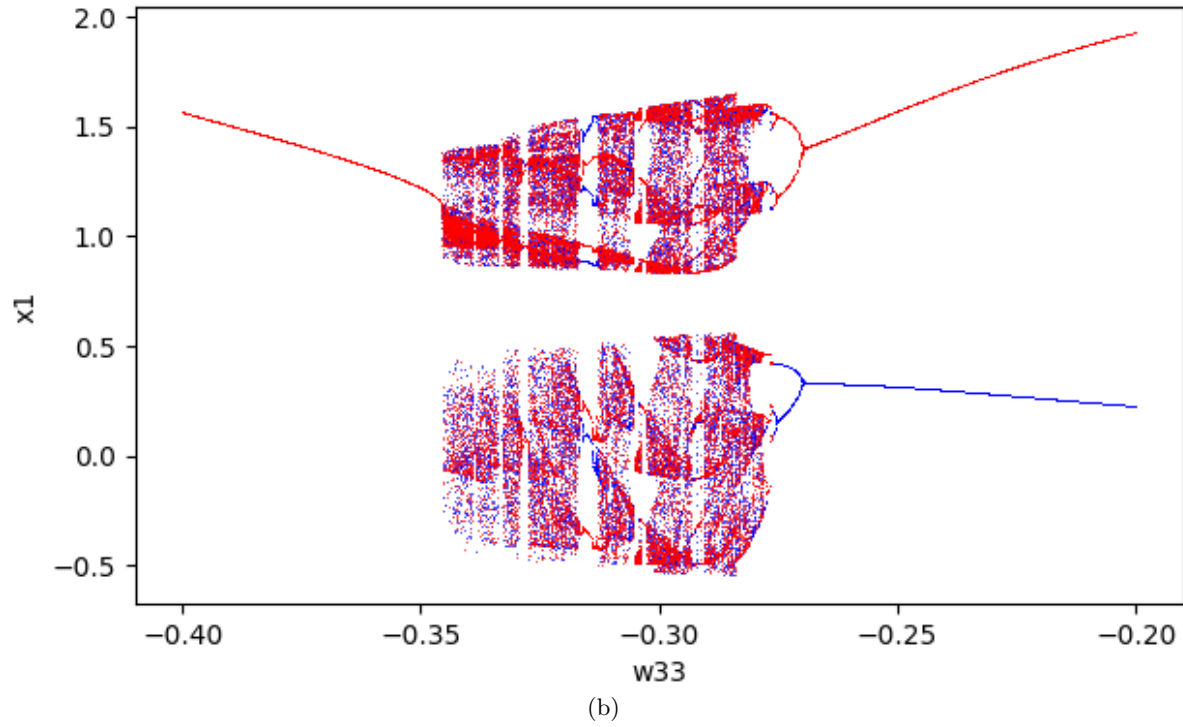
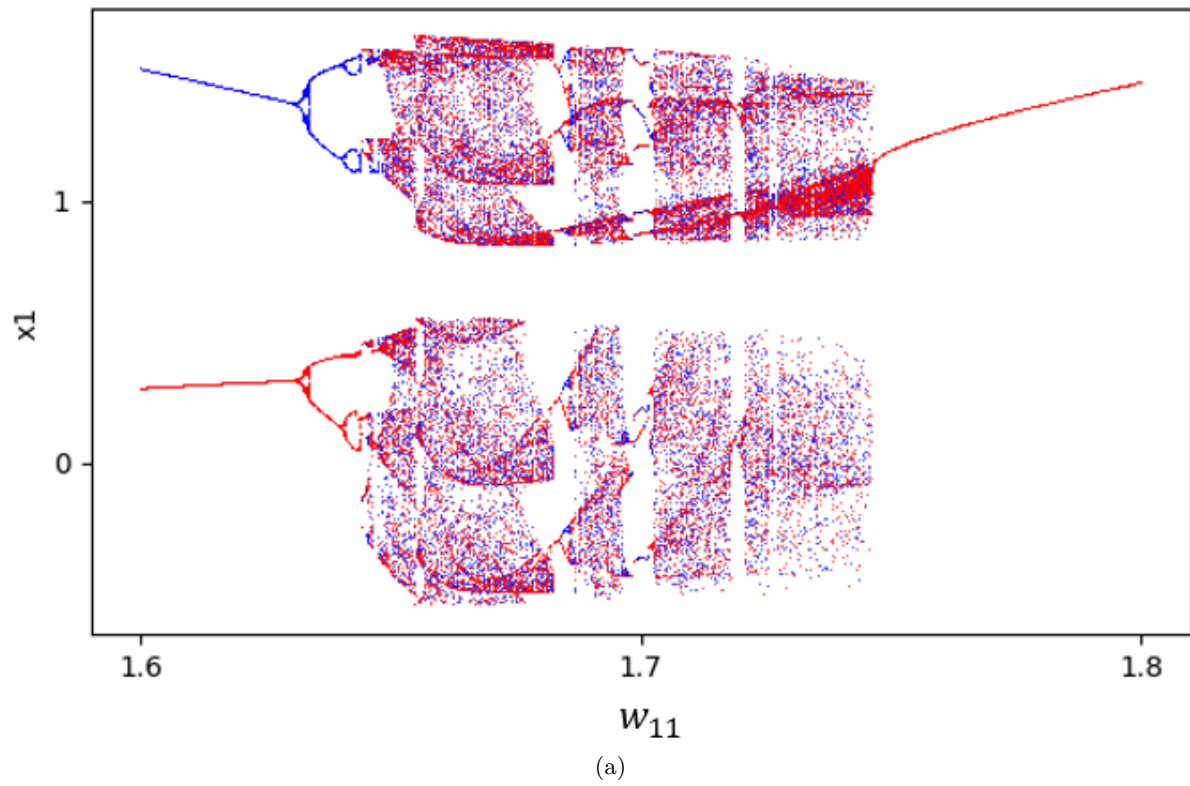
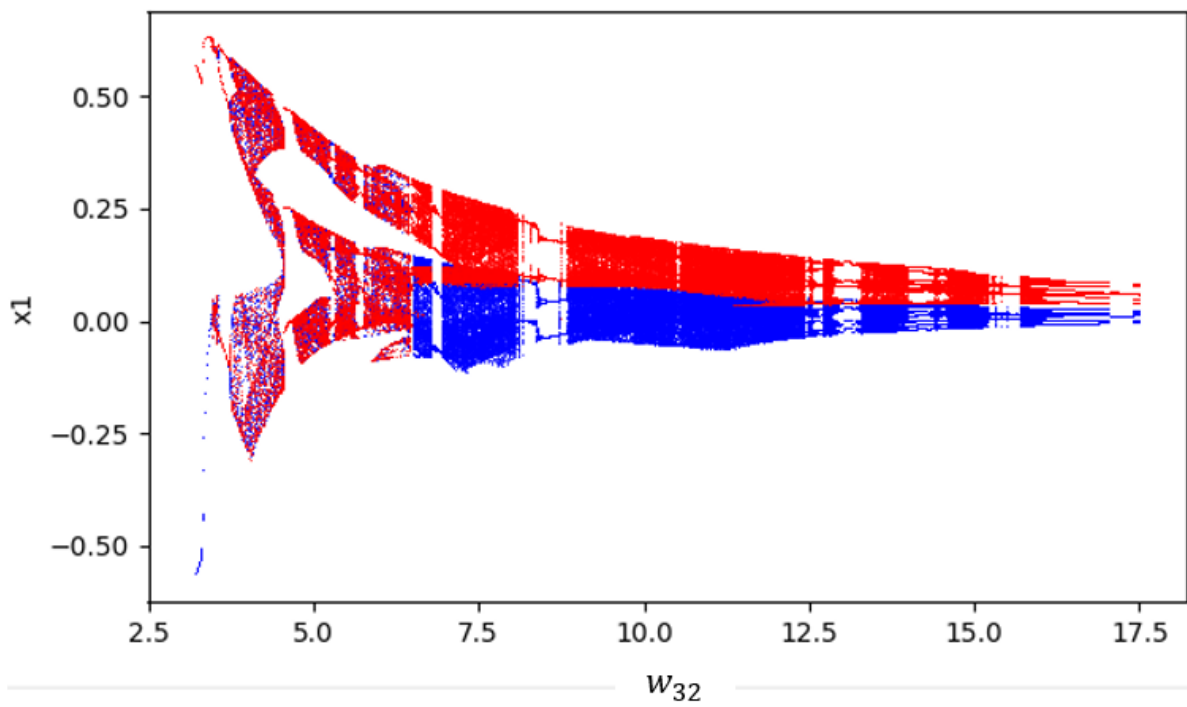
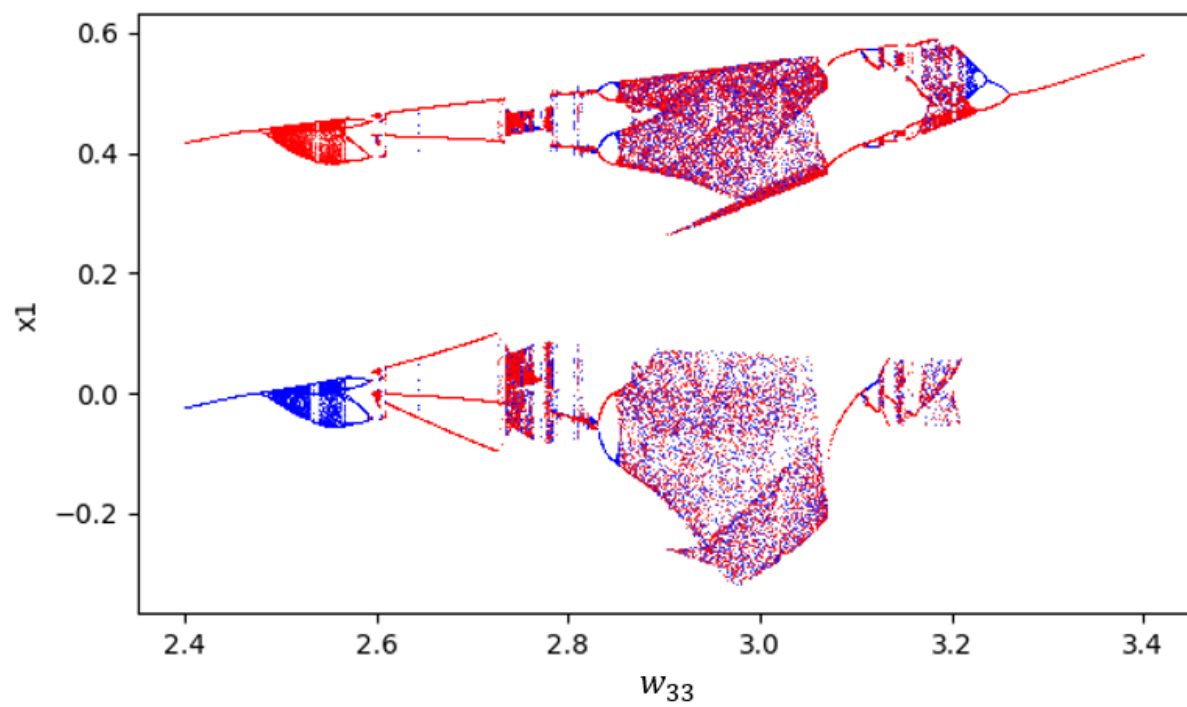


Figura 4.7: Diagramas de bifurcación para el modelo con función de activación PWL.



(a)



(b)

Figura 4.8: Diagramas de bifurcación para el modelo con función de activación signo.

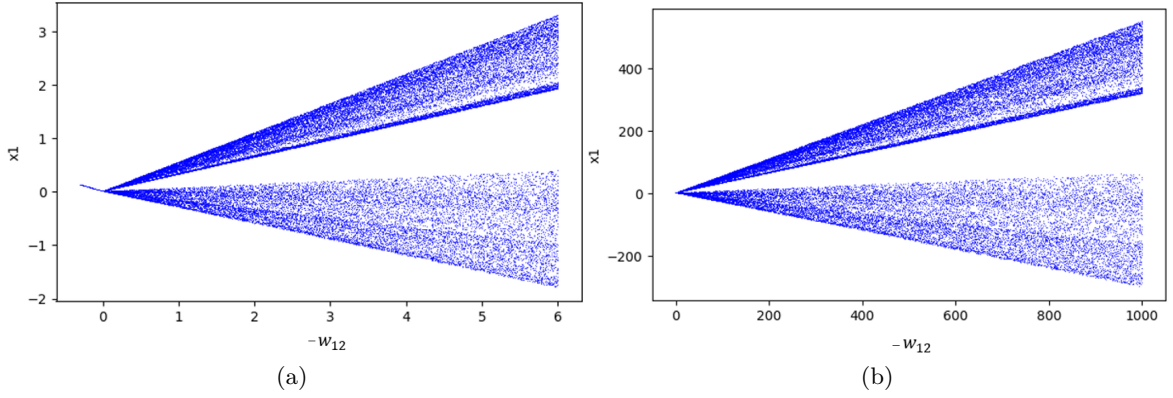


Figura 4.9: Diagrama de bifurcación variando el parametro $-w_{12}$ del modelo con función de activación signo de a) 0 hasta 6 y b) 0 hasta 1000.

momento t las dos trayectorias tendrán una distancia

$$\delta(t) \approx \delta_0 e^{\lambda t}. \quad (4.2)$$

La cantidad λ se llama el mayor exponente de Lyapunov y cuantifica la divergencia exponencial de trayectorias cercanas [44]. El exponente de Lyapunov da comportamientos cualitativos del sistema dinámico muy importantes, para empezar, si $\lambda > 0$ indica que pequeñas divergencias crecen exponencialmente conforme el sistema evoluciona, mientras que si $\lambda < 0$ indica lo contrario. Para sistemas dinámicos de tres dimensiones caóticos es sabido que se deben obtener un exponente positivo, un exponente negativo y un exponente cero, además si se está trabajando con un sistema disipativo, se espera que el valor del exponente negativo sea mayor al negativo. La obtención de los exponentes de Lyapunov resulta en una problemática computacional, las librerías de diversos lenguajes de programación actuales, sin embargo, nos permiten obtener los exponentes de Lyapunov con relativa sencillez, en este trabajo se utilizó el lenguaje JuliaProgramming y su librería JuliaDynamics, debido a que nos ofrece múltiples funciones con relación a sistemas dinámicos y caóticos, en específico la función `lyapunovspectrum()` obtiene los exponentes de Lyapunov de un sistema dinámico.

El número de exponentes de Lyapunov obtenidos depende de la dimensión del modelo dinámico, es por esta razón que con todos los modelo estudiados hasta ahora se obtendrán tres exponentes de Lyapunov con excepción del modelo con función $\text{ReLU}(x)$, para seguir la misma regla para la identificación de caos es posible expandir el sistema no autónomo a uno autónomo

si consideramos $x_3 = t$, y por lo tanto es sistema se convierte

$$\begin{aligned}\dot{x}_1 &= -1.15x_1 + 1.1\text{ReLU}(x_1) - 1.05\text{ReLU}(x_2) - 0.5\text{sen}(0.52x_3) + 0.8 \\ \dot{x}_2 &= -0.55x_2 + 0.95\text{ReLU}(x_1) + 0.75\text{ReLU}(x_2) - 0.75 \\ \dot{x}_3 &= 1\end{aligned}, \quad (4.3)$$

Los exponentes de Lyapunov dan una indicación del comportamiento del sistema dinámico en relación al signo que estos poseen, la Tabla 4.1 resume el comportamiento esperado del atractor para un sistema dinámico de tres dimensiones disipativo dado el signo de los exponentes de Lyapunov [32].

Signo de los LE's	Tipo de atractor
$(-, -, -)$	Punto fijo
$(0, -, -)$	Ciclo limite
$(0, 0, -)$	Cuasi-periódico
$(+, 0, -)$	Caótico

Tabla 4.1: Exponentes de Lyapunov y su relación con los tipos de atractores en un sistema dinámico de tres dimensiones.

Los resultados obtenidos para los LE variando los parámetros estudiados del modelo con función tangente hiperbólica se muestran en la Figura 4.10, se analiza el mismo spectrum de valores estudiados con los diagramas de bifurcación con los cuales existe una evidente similitud en los resultados. Posteriormente en la Figura 4.11 se desarrolló un mapa de calor con el exponente máximo de Lyapunov (LLE) variando los parámetros estudiados w_{12} y w_{13} , en blanco están las zonas no caóticas, en amarillo están marcadas las zonas donde el LLE es relativamente pequeño, estas zonas se marcan distinto considerando que el valor positivo del exponente podría tratarse de un ligero error numérico, mientras que en las zonas rojas y negras se afirma con seguridad la existencia de zonas caóticas. Comparado con los modelos hallados en la literatura, la zona caótica es mayor, esto se verá en los mapas de calor posteriores, en este sentido, este es un modelo cuya implementación electrónica es más sencilla debido al mayor rango de valores posibles, así como también tiene una utilidad en aplicaciones criptográficas en donde se busca variar los parámetros del modelo de manera constante y por lo tanto se requiere de modelos con una amplia zona caótica.

El mismo procedimiento para la obtención de los exponentes de Lyapunov del modelo con función tangente hiperbólica fue realizado con los demás modelos, algunas observaciones con los resultados obtenidos son los siguientes. Para el modelo con función de activación sigmoide se tienen la Figura 4.12 para los LE, mientras que la Figura 4.13 ofrece el mapa de calor con los LLE, de este último queda en evidencia su dificultad para una implementación en dispositivos electrónicos debido a la pequeña zona con comportamiento caótico del sistema. En las Figuras

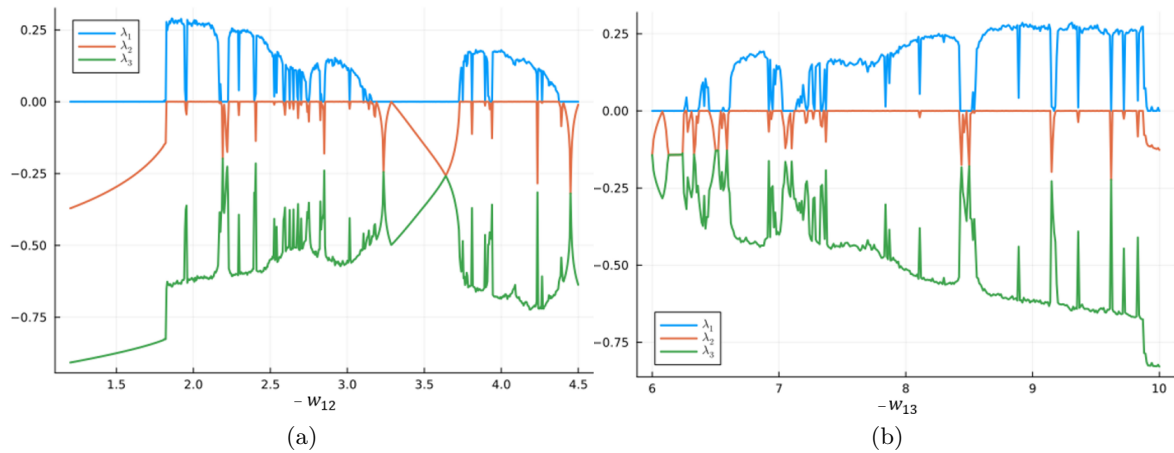


Figura 4.10: Signo de los exponentes de Lyapunov variando parametros de bifurcación a) w_{12} y b) w_{13} del modelo con función de activación tangente hiperbólica.

4.14 y 4.15 se exploran los LE y LLE del modelo 3.21, notemos la simetría con respecto al parámetro de amplitud de la señal externa además de un mayor rango de parámetros para los cuales el sistema presenta caos con respecto al sistema anterior, los exponentes máximos de Lyapunov sin embargo, son las pequeños al sistema propuesto. Para el modelo con función de activación PWL se tiene la Figura 4.16 con los LE's se observa un comportamiento frecuente con funciones de activación no completamente diferenciables, como fue mencionado, para que un sistema sea considerado caótico los LE deben ser uno positivo, otro cero y otro negativo, los resultados entonces no concuerdan con esta descripción, esto es debido a los puntos no diferenciables, considerando este hecho el modelo sigue siendo considerado como caos [45], el mapa de calor correspondiente (Figura 4.17) muestra una zona caótica relativamente pequeña pero mayor a la hallada con la función sigmoide. Finalmente, para el modelo con función signo los resultados para LE Figura 4.18 y LLE Figura 4.19 se encontró una gran área con zonas caóticas, considerando incluso que la variación del parámetro w_{12} no afecta el comportamiento cualitativo del sistema se tiene un volumen infinito con las posibilidades para la elección de los parámetros del modelo. En general podemos concluir que los modelos propuestos ofrecen una implementación electrónica más sencilla y estable con respecto a los modelos hallados en la literatura.

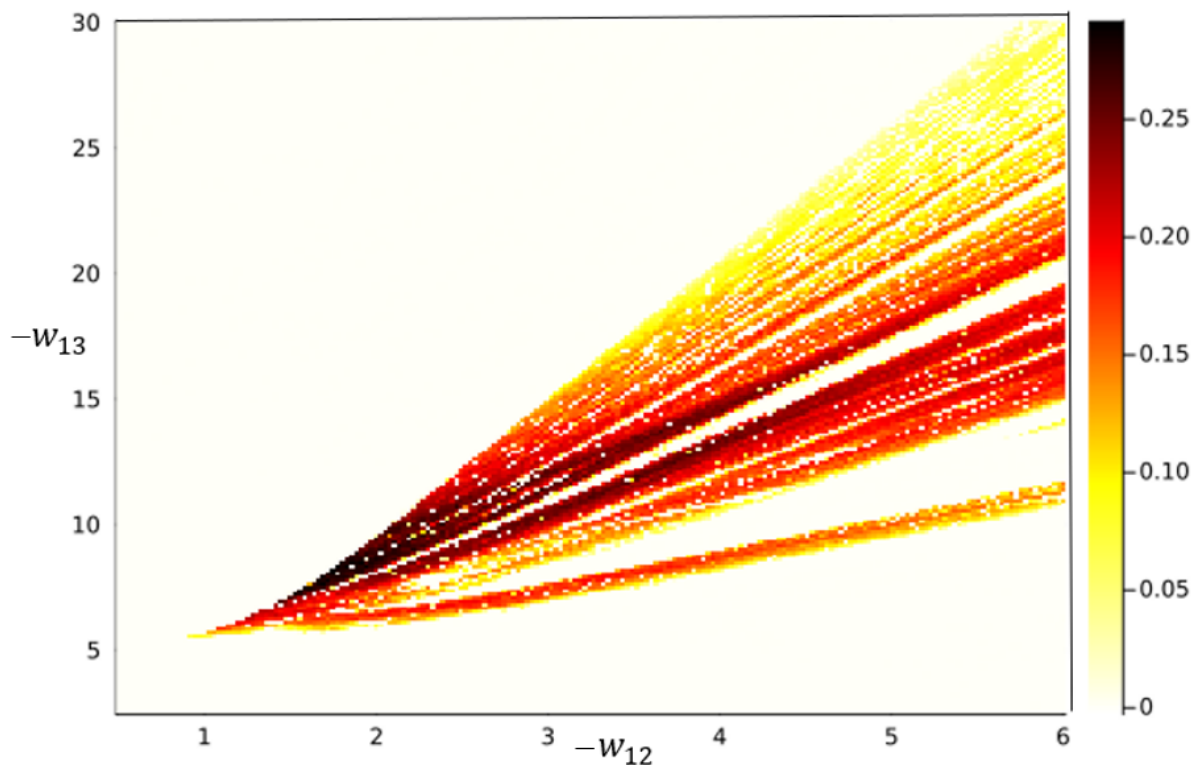


Figura 4.11: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{12} y w_{13} del modelo de HNN's con función de activación tangente hiperbólica.

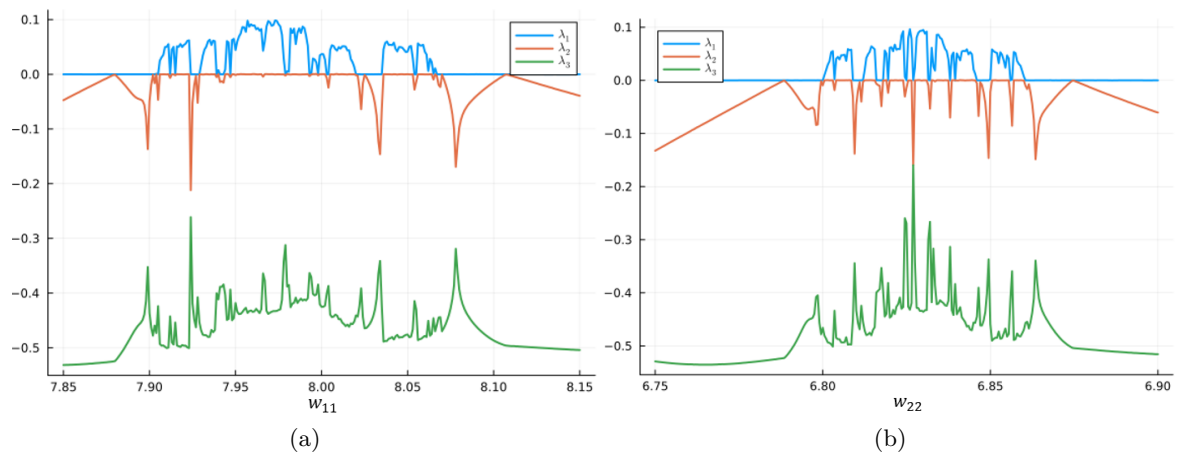


Figura 4.12: Exponentes de Lyapunov variando parametros de bifurcación a) w_{11} y b) w_{22} del modelo con función de activación sigmoide.

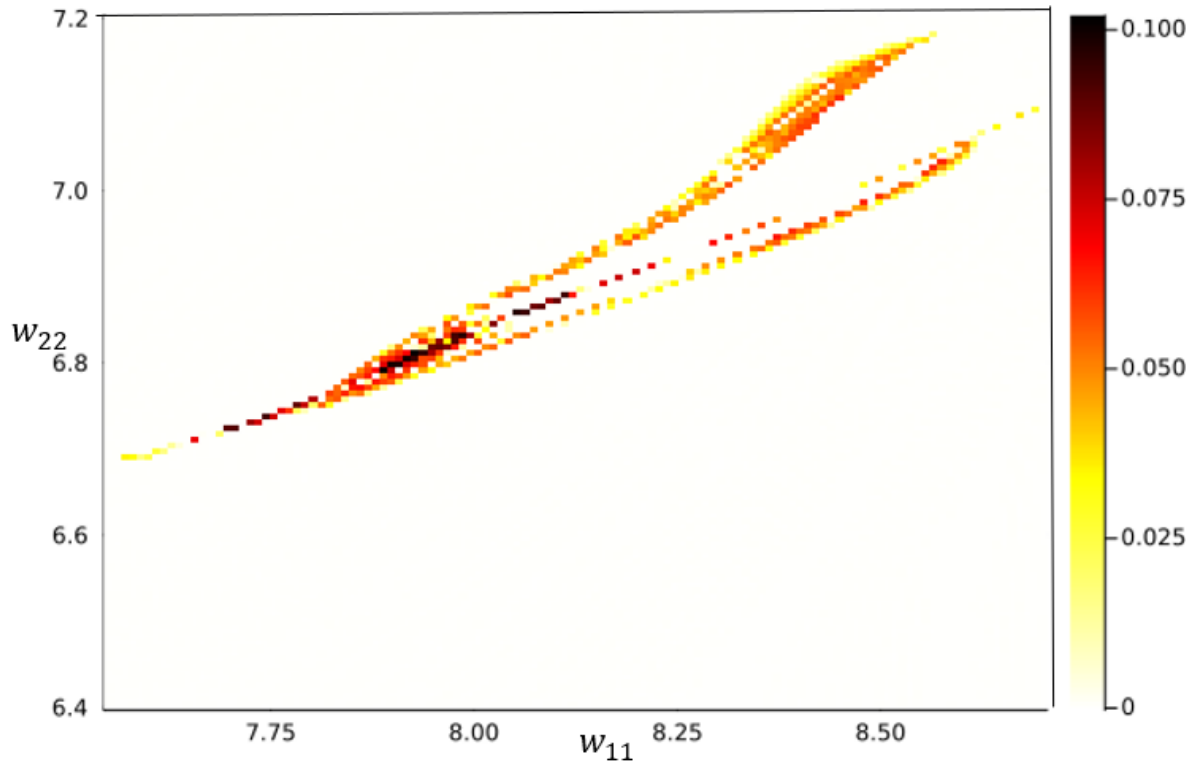


Figura 4.13: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{11} y w_{22} del modelo de HNN's con función de activación sigmoide.

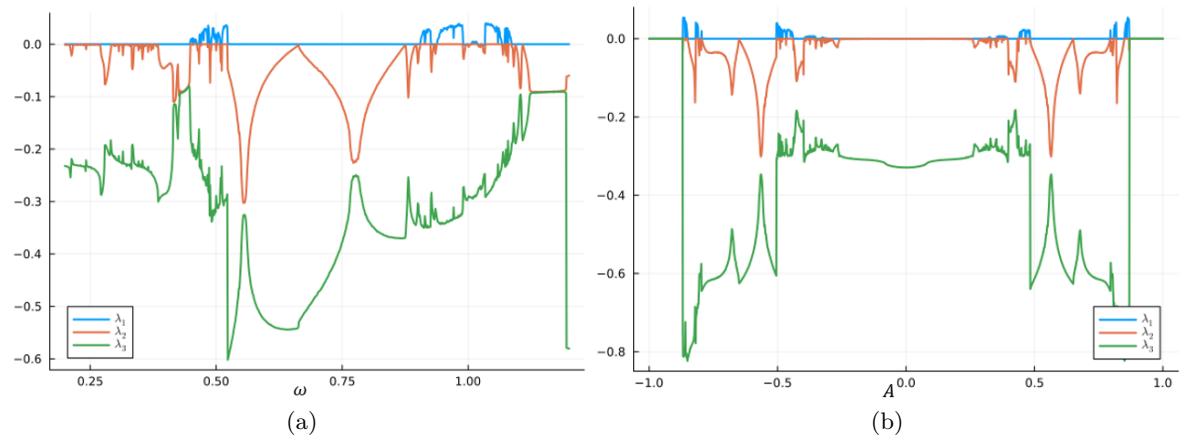


Figura 4.14: Exponentes de Lyapunov variando parametros de bifurcación a) frecuencia de la señal externa senoidal ω y b) amplitud de la señal externa senoidal A del modelo con función de activación ReLU.

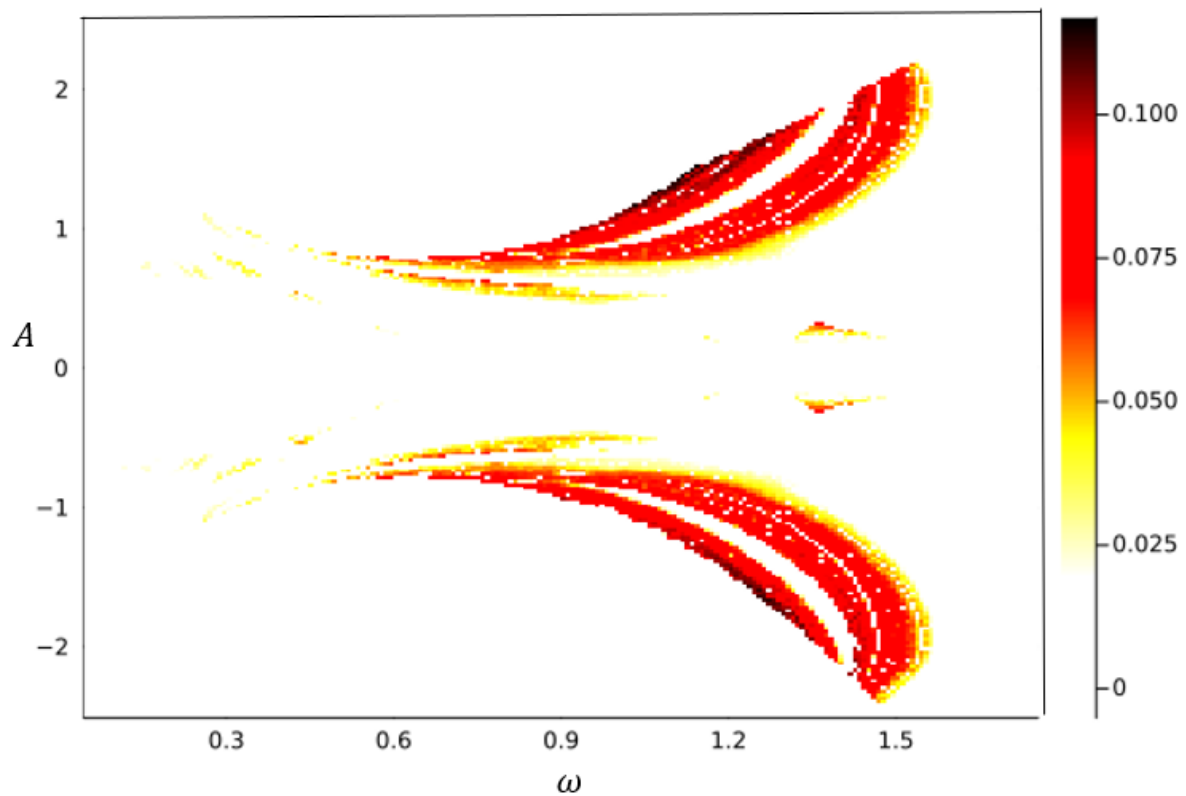


Figura 4.15: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros ω y A del modelo de HNN's con función de activación ReLU.

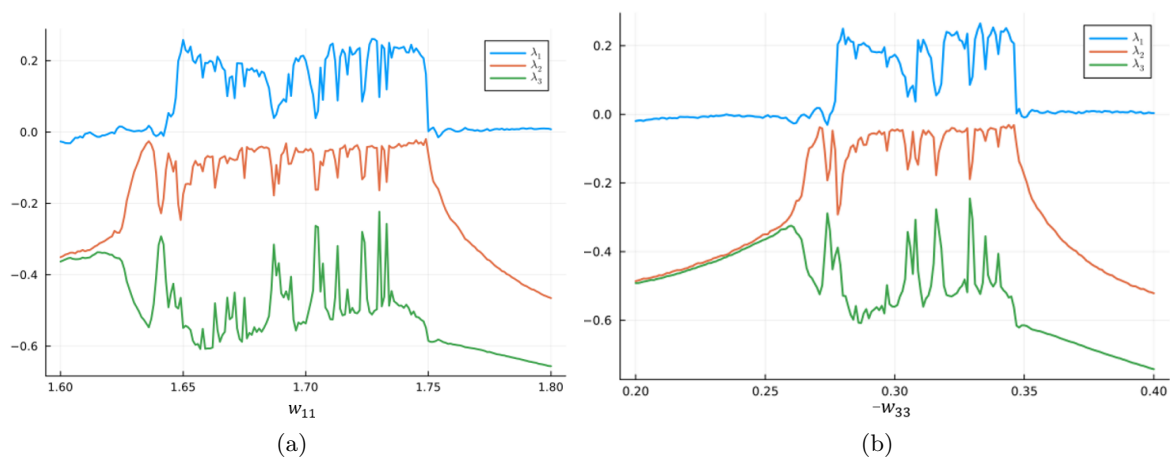


Figura 4.16: Exponentes de Lyapunov variando parámetros de bifurcación a) w_{11} y c) w_{33} del modelo con función de activación PWL.

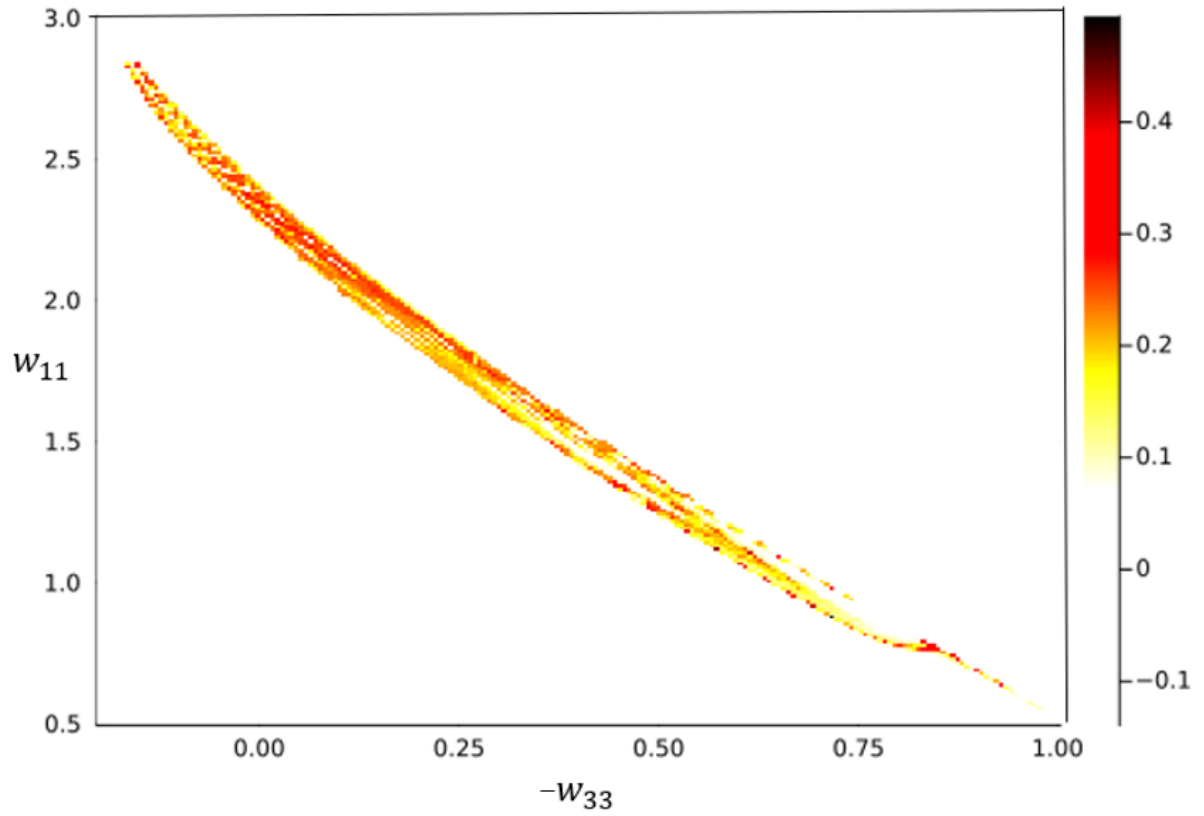


Figura 4.17: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{11} y w_{33} del modelo de HNN's con función de activación PWL.

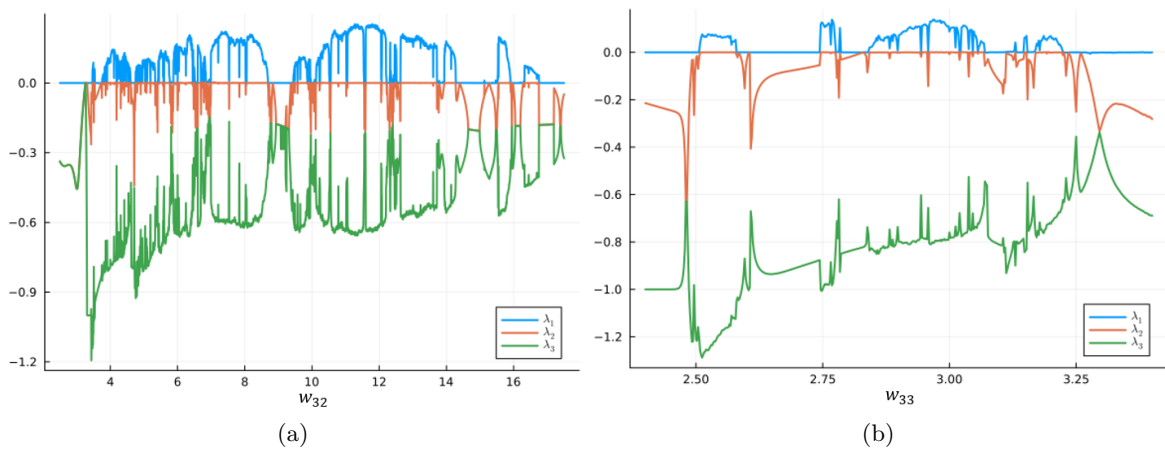


Figura 4.18: Exponentes de Lyapunov variando parametros de bifurcación a) w_{32} y b) w_{33} del modelo con función de activación signo.

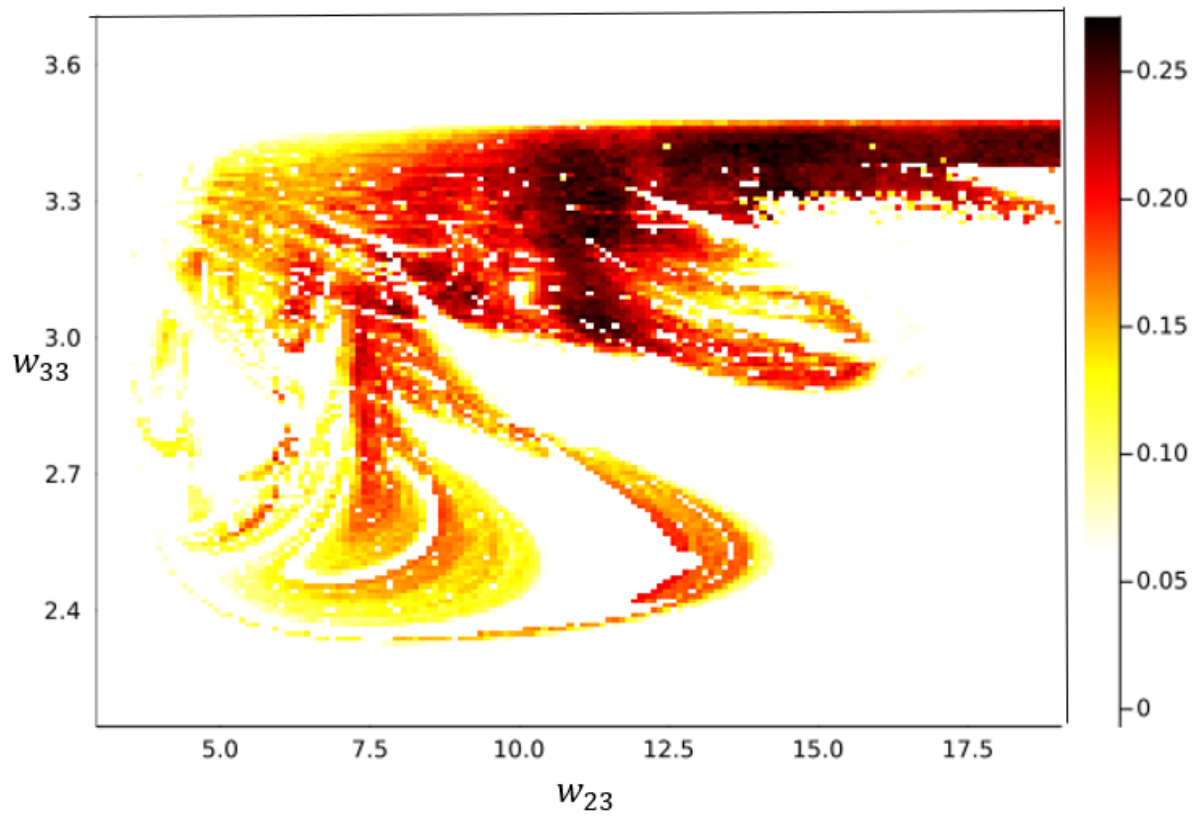


Figura 4.19: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{32} y w_{33} del modelo de HNN's con función de activación signo.

Capítulo 5

Implementación de las redes neuronales Hopfield

5.1. Desarrollo de funciones de activación

Las funciones de activación pueden ser descritas en base a circuitos electrónicos o, para nuestro caso, en base a bloques de programación para la tarjeta programable analógica. A día en el que se está redactando este trabajo la distribuidora Anadigm proporciona 54 bloques de programación en los que se encuentran ganancias (programables y saturadas), comparadores, señales periódicas, filtros (pasa-bajas, pasa-altas, pasa-banda y rechaza-banda), integradores, entre otros. A partir de algunos de estos bloques se describió el comportamiento de las funciones de activación.

5.1.1. Implementación de la función signo

La función signo puede ser vista como un comparador el cual identifica entre valores positivos y negativos, es por esto que se utilizó el bloque **comparator** para emular este comportamiento, en la Figura 5.1 se tienen los bloques utilizados para la implementación de la función signo, el primer bloque es un comparador con histéresis nula sin voltaje de referencia mientras que el segundo es un inversor de ganancia 0.5, la expresión equivalente al bloque comparador es

$$V_{out} = \begin{cases} -2, & \text{si } V_{in} > V_{Hyst} \\ 2, & \text{si } V_{in} < -V_{Hyst} \end{cases}, \quad (5.1)$$

mientras que el voltaje de salida se mantendrá sin cambio si alguna de estas condiciones no se cumplen. Multiplicando por una ganancia de $-\frac{1}{2}$ y fijando a $V_{Hyst} = 0$ se obtiene el comportamiento de la función signo.

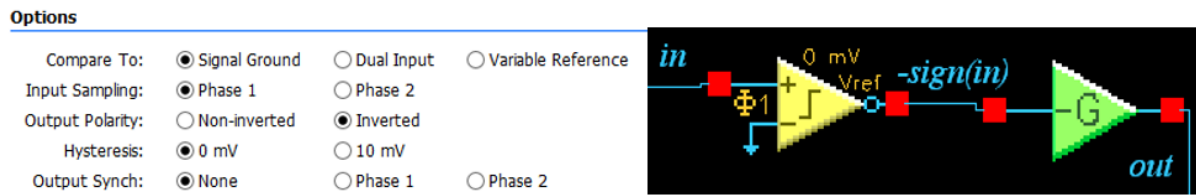
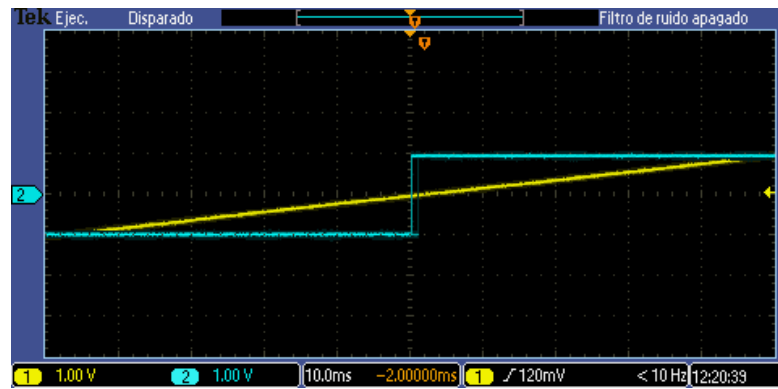
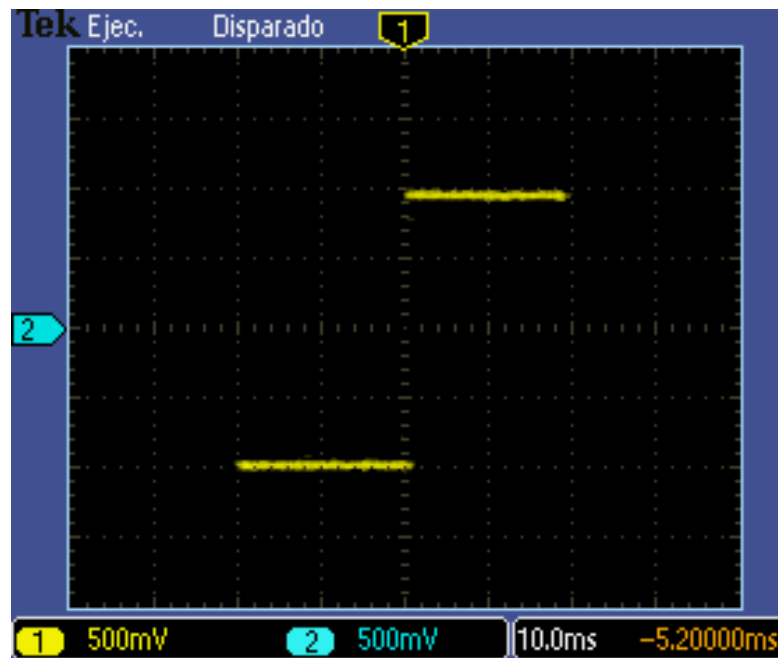


Figura 5.1: Configuración y programación en bloques de la función signo.



(a)



(b)

Figura 5.2: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida.

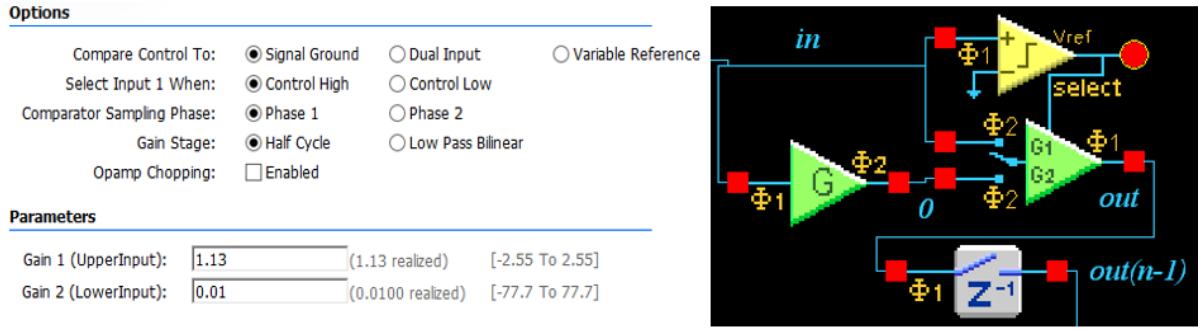


Figura 5.3: Configuración y programación en bloques del comparador analógico.

5.1.2. Implementación de la función ReLu

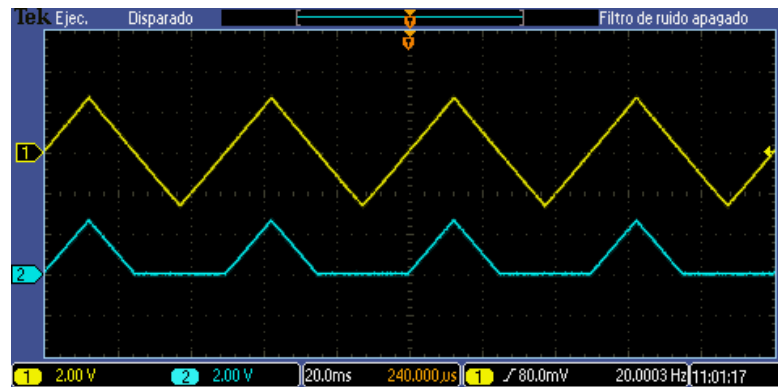
La función $\text{ReLu}(x)$ es cero cuando la entrada es menor o igual a cero, mientras que iguala la entrada cuando esta es mayor a cero, para este comportamiento entonces es necesario un comparador con respecto a cero que active y desactive la entrada, el bloque que permite este comportamiento es **Gain Stage with Switchable Inputs**, este es un bloque con ganancia dependiente a un comparador, de esta manera es posible conectar el voltaje V_{in} a una de las entradas de la ganancia intercambiable y al comparador, mientras que a la otro entrada se conecta una ganancia muy pequeña, además, configuración de la ganancia intercambiable para esta entrada también se configura muy pequeña, en específico ambas ganancias son de 0.01, es decir, el valor para la segunda entrada es como máximo 0.00027. Finalmente es necesario un bloque **Delay** debido a que la conmutación genera fenómenos no deseados en el comportamiento del circuito. La expresión que describe el comportamiento de este circuito es

$$V_{out} = \begin{cases} V_{in}, & \text{select} = V_{in} > 0, \\ 0.0001V_{in}, & \text{select} = V_{in} < 0. \end{cases} \quad (5.2)$$

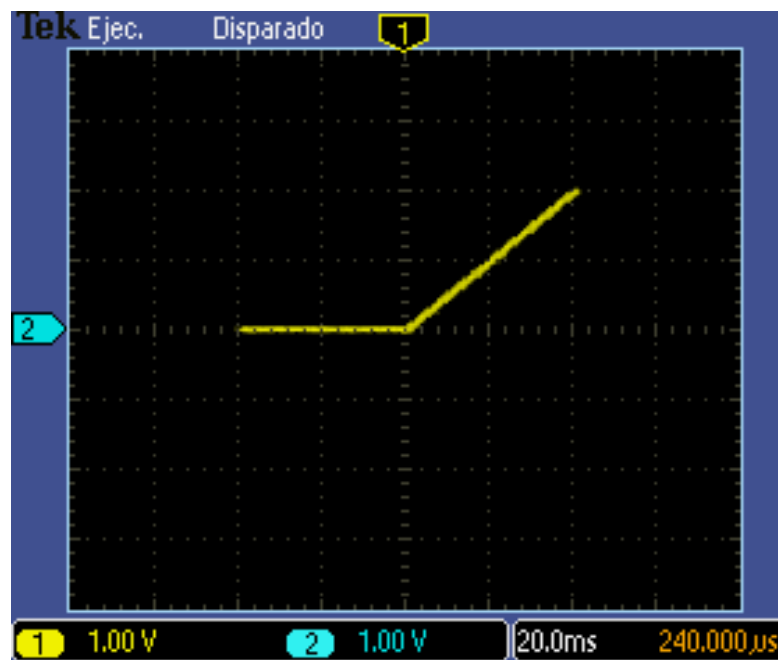
La implementación de la función de activación es presentada en la Figura 5.4 para una señal tipo rampa, notese que la entrada es multiplicada por una ganancia de 1.13, este valor fue encontrado experimentalmente para obtener un comportamiento adecuado.

5.1.3. Implementación de la función PWL

La función PWL utilizada es una función saturada de ganancia unitaria, este es un comportamiento descrito por un único bloque llamado **GainLimiter** presentado en la Figura ??, en este sentido esta es la función de activación con implementación más sencilla de las presentadas. En la implementación y registro de resultado fue encontrado que este bloque no funcionaba de manera adecuada para voltajes menores a -0.45 V, es por esto que los voltajes de satura-



(a)



(b)

Figura 5.4: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida de la función ReLu.



Figura 5.5: Configuración y programación en bloques de la ganancia saturada.

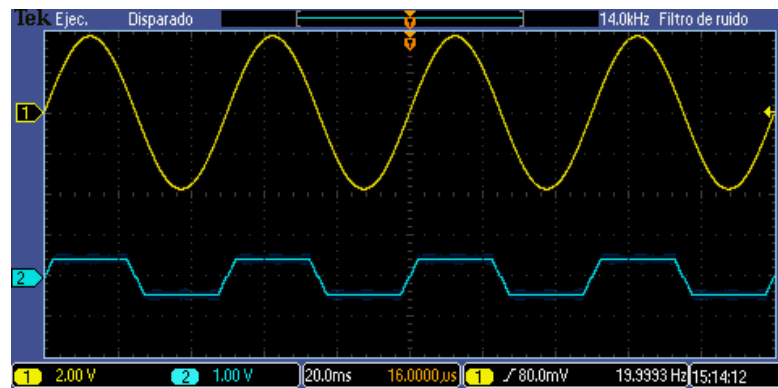
ción del bloque fueron ajustados a 0.4 V, a pesar de este cambio los parámetros en el modelo correspondiente no fueron modificados ya que el comportamiento caótico no se ve afectado, el único cambio cualitativo en el sistema es la magnitud de este. El resultado ante una entrada senoidal es presentado en la Figura 5.6 en (a) mientras que (b) es el resultado de graficar la señal de entrada contra la salida.

5.1.4. Implementación de la función tangente hiperbólica y sigmoide

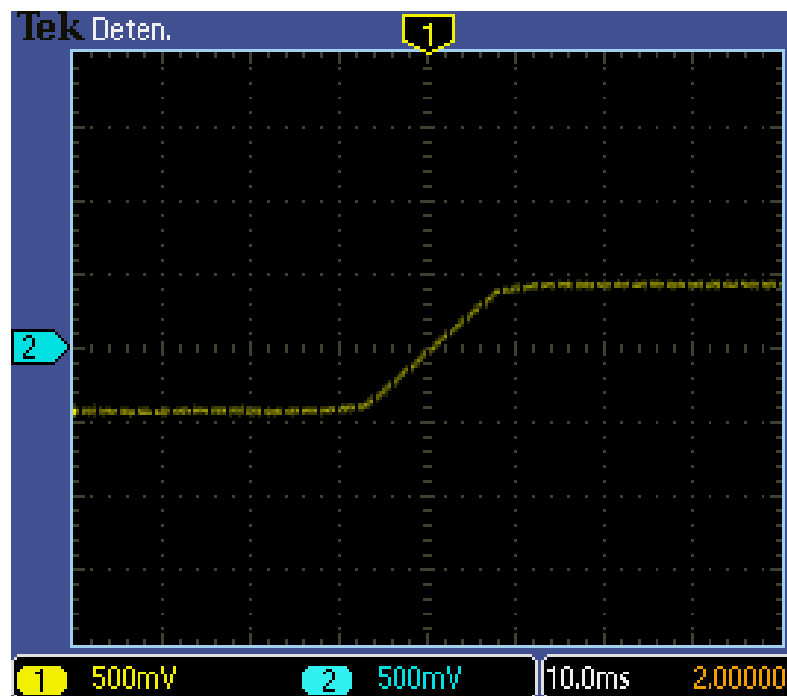
Para funciones más complejas Anadigm ofrece un bloque con memoria Look Up Table o LUT (Tabla de correspondencia en español), con la cual es posible asignar un valor de voltaje de entrada a uno de salida por medio de una SRAM de 256 x 8 bits, el diagrama de bloques de la LUT facilitado por Anadigm [46] es presentado en la Figura 5.7 en donde se observa la utilización de un ADC de 8 bits para la digitalización de la señal de entrada conectado a la memoria SRAM con 256 direcciones y su salida un DAC, en la misma figura se tiene la ventana para la programación de la LUP, a este es necesario cargar un documento .CSV con los 256 valores que se asignara a la SRAM. Por medio de la LUP es que se implementó las funciones no lineales tangente hiperbólica y sigmoide, esto implica que el resultado obtenido es un aproximado a las funciones reales. El resultado de la implementación de la función $\tanh(2V_{in})$ es presentado en la Figura 5.8 ante una función tipo rampa 6V de pico a pico con un periodo de 40ms junto a su respuesta de entrada contra salida, mientras que en la Figura 5.9 se presenta la respuesta de la función de activación $\sigma(2V_{in})$ ante la misma entrada.

5.2. Programación e implementación de las HNN's de 3 neuronas

Para la implementación de la red neuronal Hopfield en circuitos analógicos resulta conveniente transformar el sistemas de ecuaciones diferenciales que describen el sistema a su respuesta en frecuencia en base a la transformada de Laplace, consideremos la i -ésima ecuación



(a)



(b)

Figura 5.6: Resultado de la implementación electrónica a) con entrada senoidal y b) respuesta entrada contra salida de la función PWL.

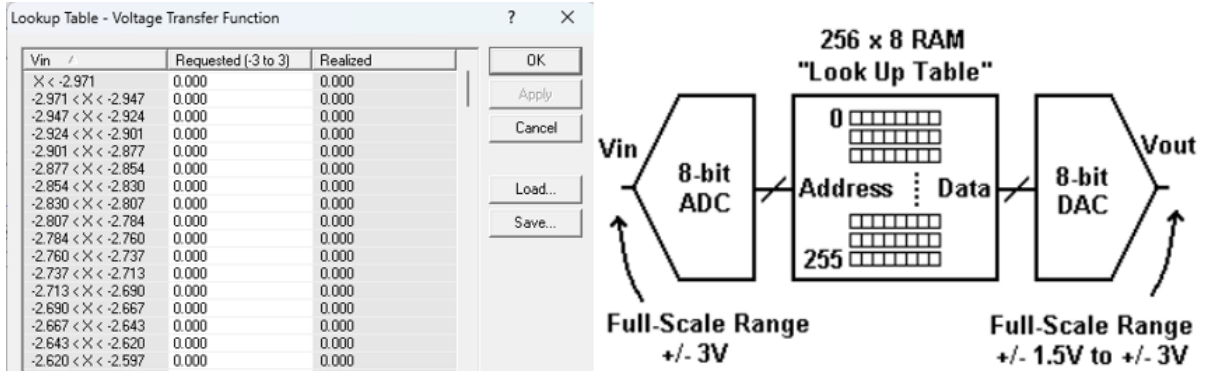


Figura 5.7: Diagrama en bloques de la Look Up Table.

que representa el voltaje que pasa por la respectiva neurona de una HNN

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^n w_{ij} f(x_j) + b_i, \quad (5.3)$$

aplicando la transformada de Laplace de la expresión anterior se tiene

$$\begin{aligned} \mathcal{L}(\dot{x}_i) &= -\mathcal{L}(c_i x_i) + \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ sX_i(s) &= -c_i X_i(s) + \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ \Rightarrow X_i(s)(s + c_i) &= \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ \Rightarrow X_i(s) &= \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s + c_i} \end{aligned} \quad (5.4)$$

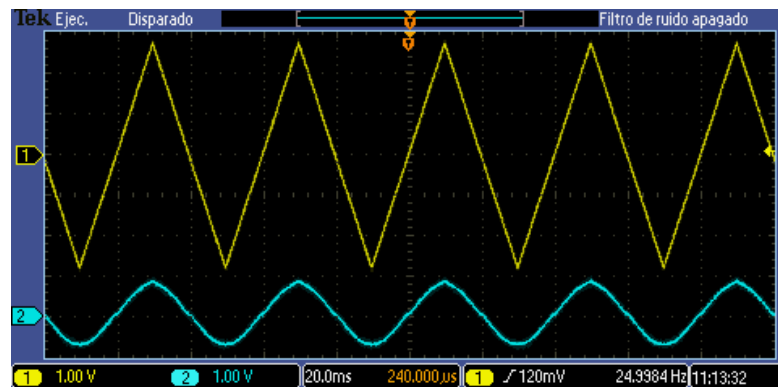
consideremos ahora un escalamiento en frecuencia de magnitud $2\pi f_0$, para el cual se realiza un cambio de variables $s \rightarrow s/(2\pi f_0)$, esto da

$$\begin{aligned} X_i(s) &= \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s/(2\pi f_0) + c_i} \\ X_i(s) &= 2\pi f_0 \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s + 2\pi f_0 c_i} \end{aligned} \quad (5.5)$$

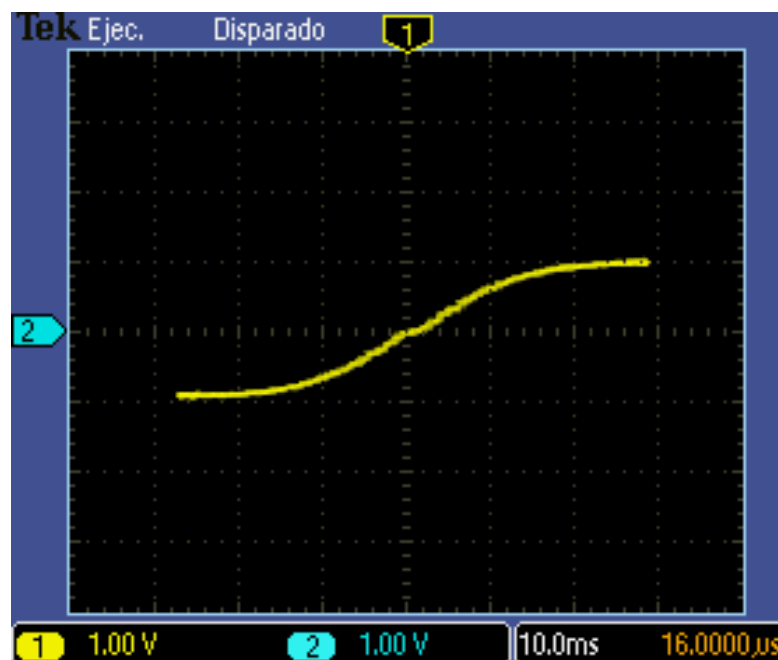
la forma de la expresión anterior es equivalente a un filtro sumador cuyo bloque esta incluido en la biblioteca del dispositivo analógico programable presentado en la Figura 5.10 junto a sus opciones y parámetros configurables. La función de transferencia equivalente a este bloque esta en la documentación de AnadigmDesigner2 y viene dada por

$$V_{out}(s) = \frac{2\pi f_0 \sum_{i=1}^3 \pm G_i V_{in-i}(s)}{s + 2\pi f_0}, \quad (5.6)$$

donde f_0 es la frecuencia de corte configurable del bloque en cuestión. En base a la expresión del filtro sumador podemos reescribir el modelo de las redes neuronales Hopfield. Para el modelo con función de activación tangente hiperbólica es importante considerar que el FPAA trabaja

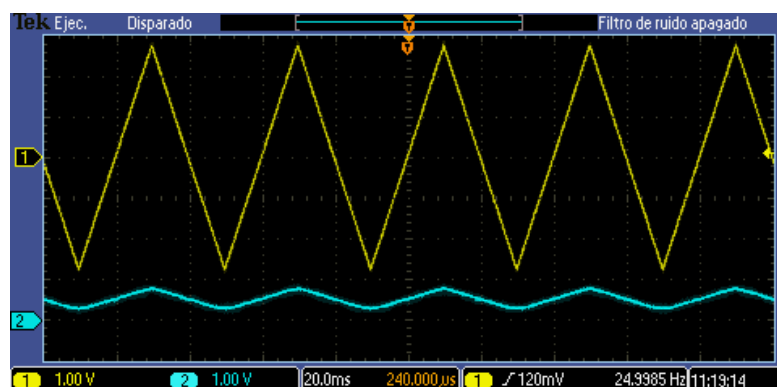


(a)

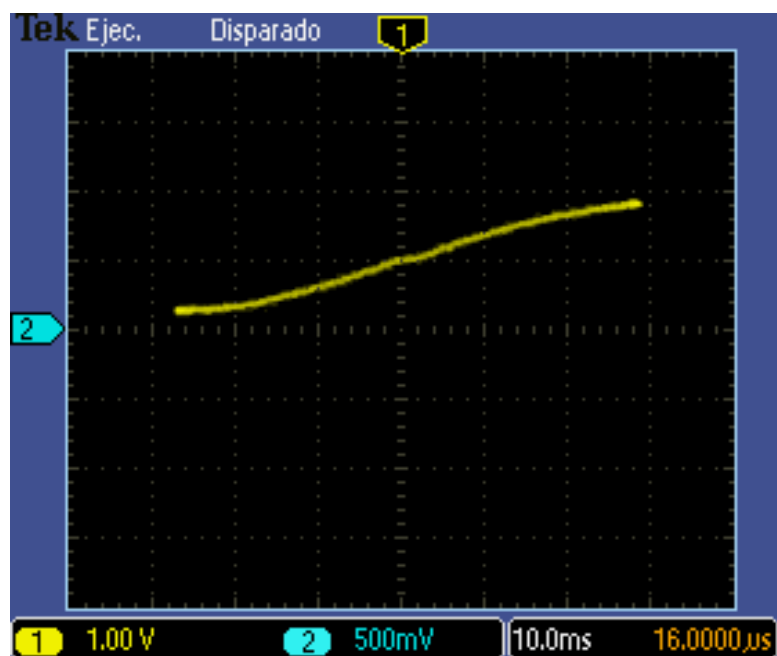


(b)

Figura 5.8: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida de la función tangente hiperbólica.



(a)



(b)

Figura 5.9: Resultado de la implementación electrónica a) con entrada sinoidal y b) respuesta entrada contra salida de la función sigmoide.

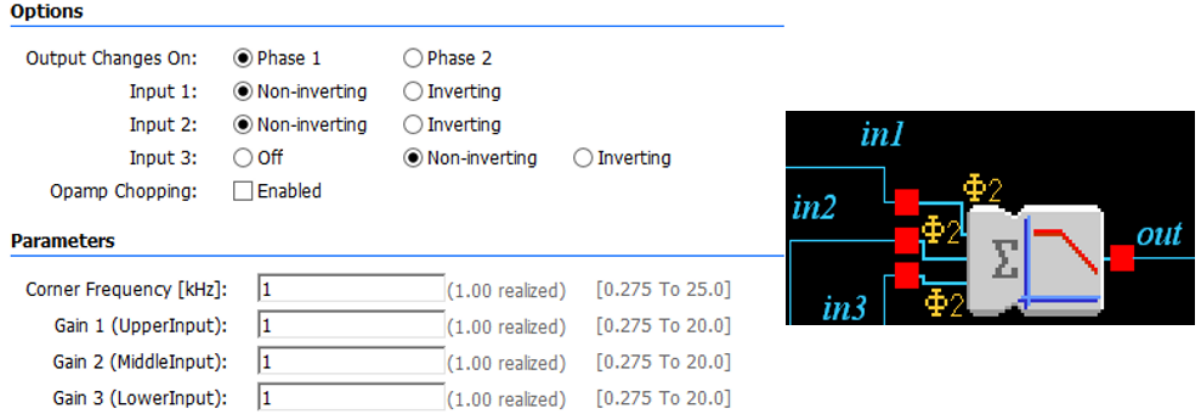


Figura 5.10: Bloque configurable de un filtro sumador.

con voltajes de ± 3 , la magnitud obtenida en la simulación del modelo en cuestión se tiene que varía de -3.2 V a 3.2 V, es entonces necesario realizar un escalamiento de magnitud de la variable de estado $x_1(t)$ realizando un cambio de variable $x_1(t) \rightarrow 2x_1(t)$, esto da

$$\begin{aligned}\dot{x}_1 &= -x_1 - \tanh x_2 - 4.5 \tanh x_3 \\ \dot{x}_2 &= -x_2 + \tanh 2x_1 + 3 \tanh x_2 + 3 \tanh x_3 \\ \dot{x}_3 &= -x_3 - \tanh 2x_1 - 3 \tanh x_2.\end{aligned}\tag{5.7}$$

Para su implementación en la tarjeta analógica se tiene

$$\begin{aligned}X_1(s) &= 2\pi f_0 \frac{-\tanh X_2(s) - 4.5 \tanh X_3(s)}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{\tanh(2X_1(s)) + 3 \tanh X_1(s) + 3 \tanh X_3(s)}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{-\tanh(2X_1(s)) - 3 \tanh X_2(s)}{s + 2\pi f_0},\end{aligned}\tag{5.8}$$

con $f_0 = 1\text{kHz}$, este valor se respeta para las restantes implementaciones. **La programación en bloques de este circuito es presentada en la sección de Anexo**, lo mismo ocurre para los circuitos implementados posteriores. El resultado de la implementación de 5.8 se muestra en la Figura 5.11 con los retratos de fase del modelo 3.5 de los planos $x_1 - x_2$ y para el plano $x_1 - x_3$ se tiene la Figura 5.12. Junto a cada implementación se coloca la simulación del sistema (escalado de ser necesario) con propósitos de comparación.

El modelo con la función de activación sigmoide requiere de un escalamiento en magnitud considerando los retratos de fase obtenidos por la simulación, los escalamientos son $x_1 \rightarrow 2x_1$,

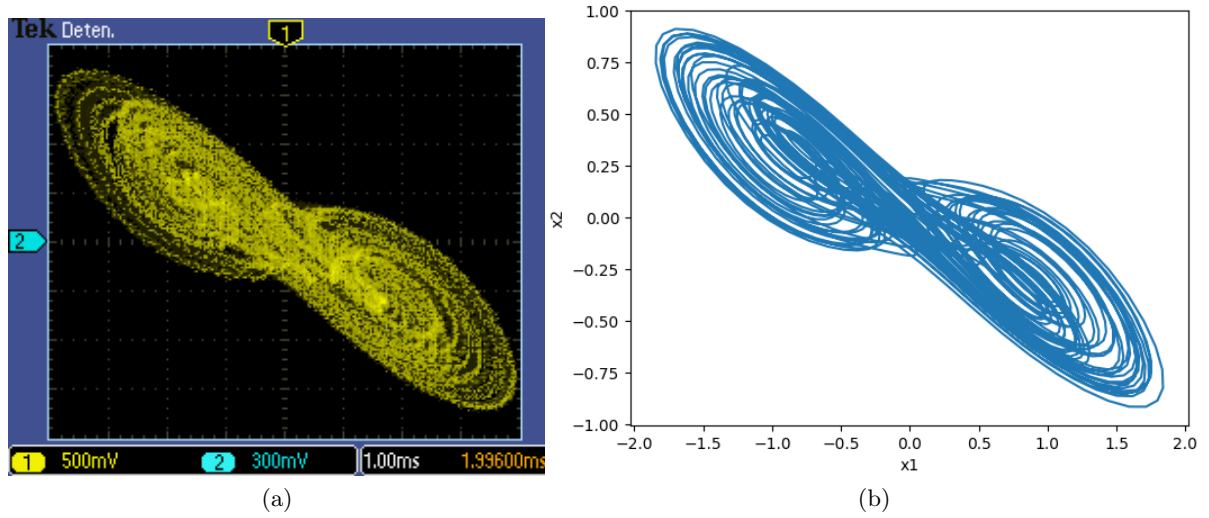


Figura 5.11: Retratos de fase del modelo con función de activación tangente hiperbólica para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

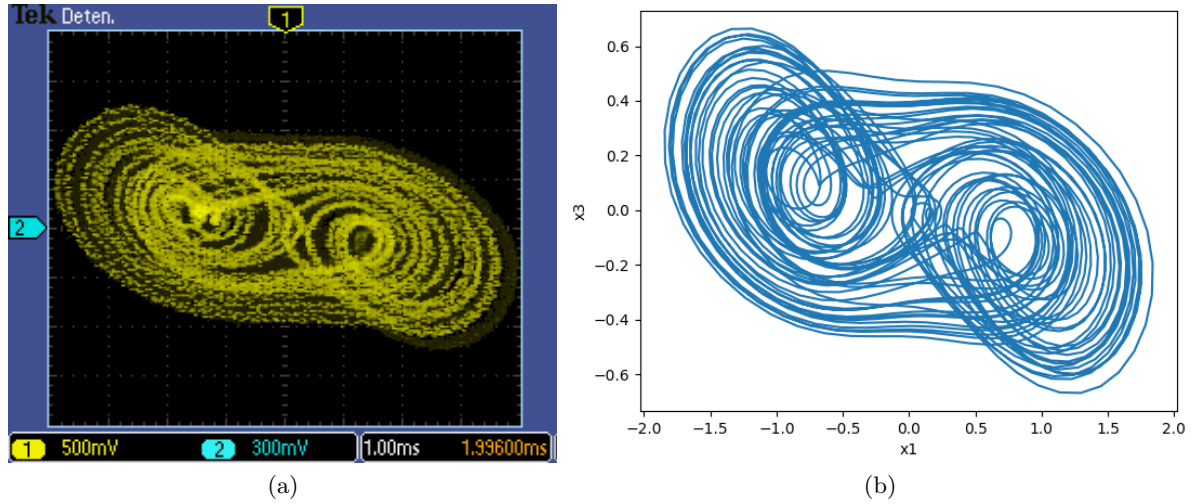


Figura 5.12: Retratos de fase del modelo con función de activación tangente hiperbólica para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

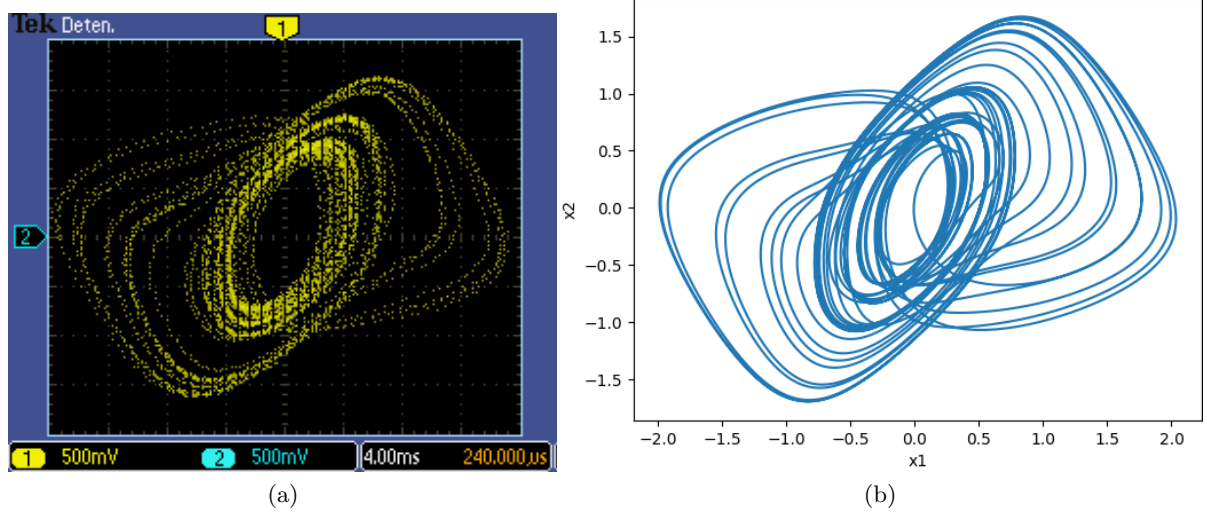


Figura 5.13: Retratos de fase del modelo con función de activación sigmoide para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

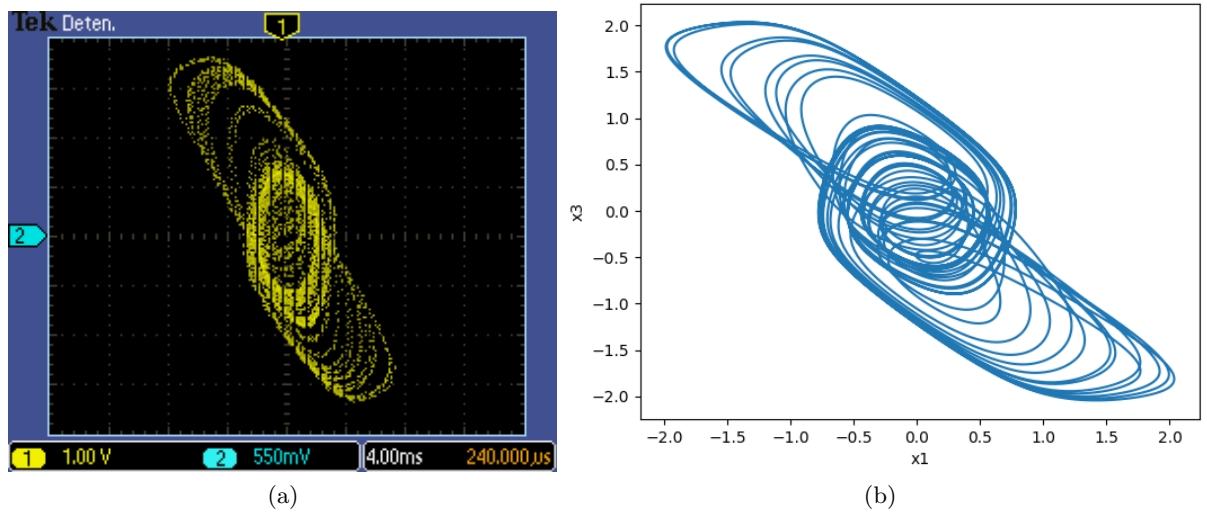


Figura 5.14: Retratos de fase del modelo con función de activación sigmoide para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

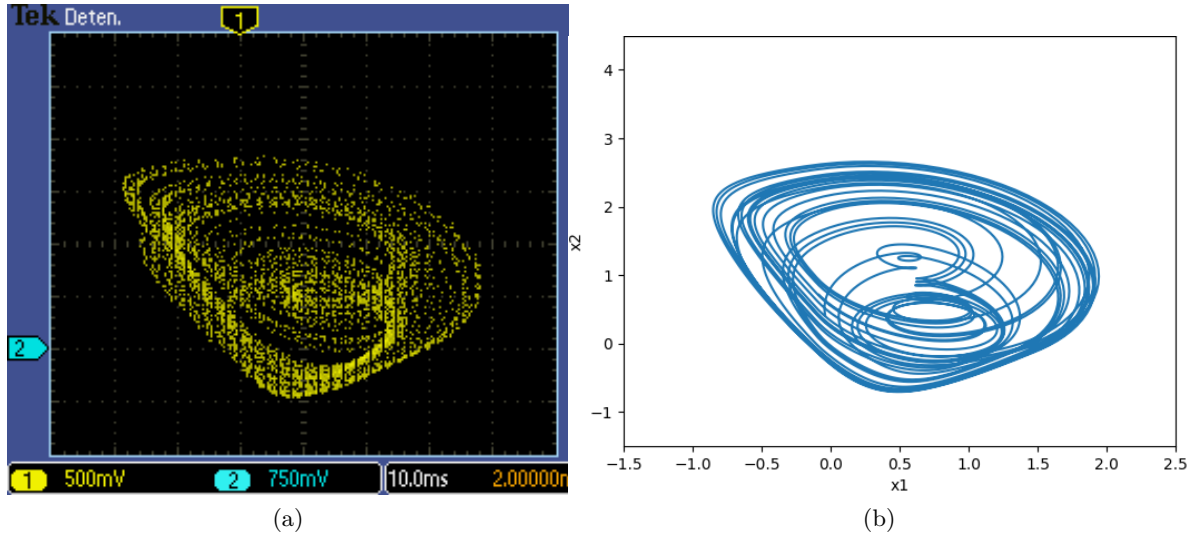


Figura 5.15: Retrato de fase del modelo con función de activación ReLU con a) la implementación electrónica y b) su simulación usando el método RK-2 con un paso de integración $1e-3$ y condición inicial $[0.1, 0]$.

$x_2 \rightarrow 2.5x_2$ y $x_3 \rightarrow 5x_3$, con lo cual se obtiene

$$\begin{aligned}\dot{x}_1 &= -x_1 + 3.985\sigma(2x_1) - 2\sigma(2.5x_2) - 1 \\ \dot{x}_2 &= -x_2 + 4\sigma(2x_1) + 2.736\sigma(2.5x_2) + 2\sigma(5x_3) - 4.365, \\ \dot{x}_3 &= -x_3 - 3.8\sigma(2x_1) + 0.734\sigma(5x_3) + 1.532\end{aligned}\quad (5.9)$$

el voltaje externo b_2 sobrepasa el voltaje máximo con el que trabajo el FPAA, esta problemática puede ser resulta escalando la frecuencia del sistema multiplicando la parte derecha del sistema por una constante, sea 0.6 esa constante para obtener

$$\begin{aligned}\dot{x}_1 &= -0.6x_1 + 2.4\sigma(2x_1) - 1.2\sigma(2.5x_2) - 0.6 \\ \dot{x}_2 &= -0.6x_2 + 2.4\sigma(2x_1) + 1.64\sigma(2.5x_2) + 1.2\sigma(5x_3) - 2.62, \\ \dot{x}_3 &= -0.6x_3 - 2.26\sigma(2x_1) + 0.44\sigma(5x_3) + 0.92\end{aligned}\quad (5.10)$$

que usando al transformada de Laplace puede ser escrito como

$$\begin{aligned}X_1(s) &= 2\pi f_0 \frac{0.4X_1(s) + 2.4\sigma(2X_1(s)) - 1.2\sigma(2.5X_2(s)) - 0.6}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{0.4X_2(s) + 2.4\sigma(2X_1(s)) + 1.64\sigma(2.5X_2(s)) + 1.2\sigma(5X_3(s)) - 2.62}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{0.4X_3(s) - 2.26\sigma(2X_1(s)) + 0.44\sigma(5X_3(s)) + 0.92}{s + 2\pi f_0},\end{aligned}\quad (5.11)$$

con el cual se formó el código presentado en el anexo y se obtuvieron los resultados mostrados en la Figura 5.13, junto a este resultado se presenta la simulación del modelo escalado para evidenciar similitud. Para la implementación del con función ReLU se utilizó una fuente de

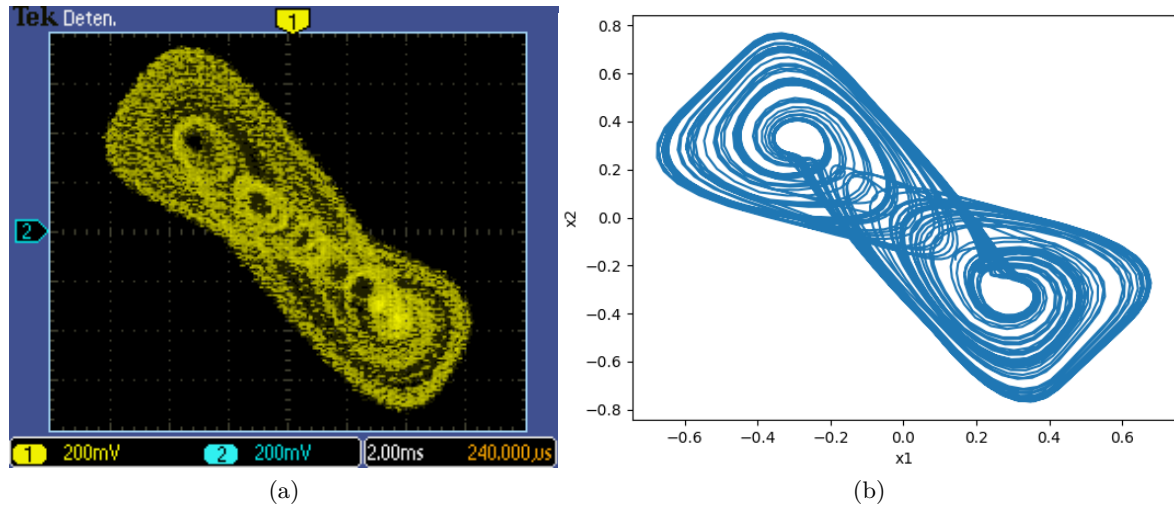


Figura 5.16: Retratos de fase del modelo con función de activación PWL para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema.

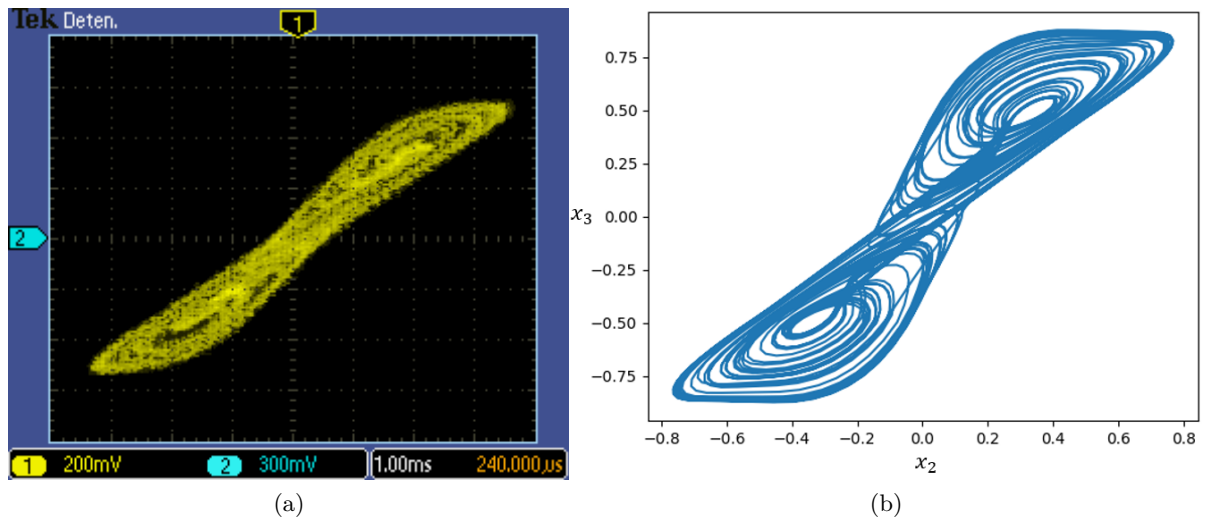


Figura 5.17: Retratos de fase del modelo con función de activación PWL para el plano $x_2 - x_3$ en a) implementación electrónica y a) simulación del sistema.

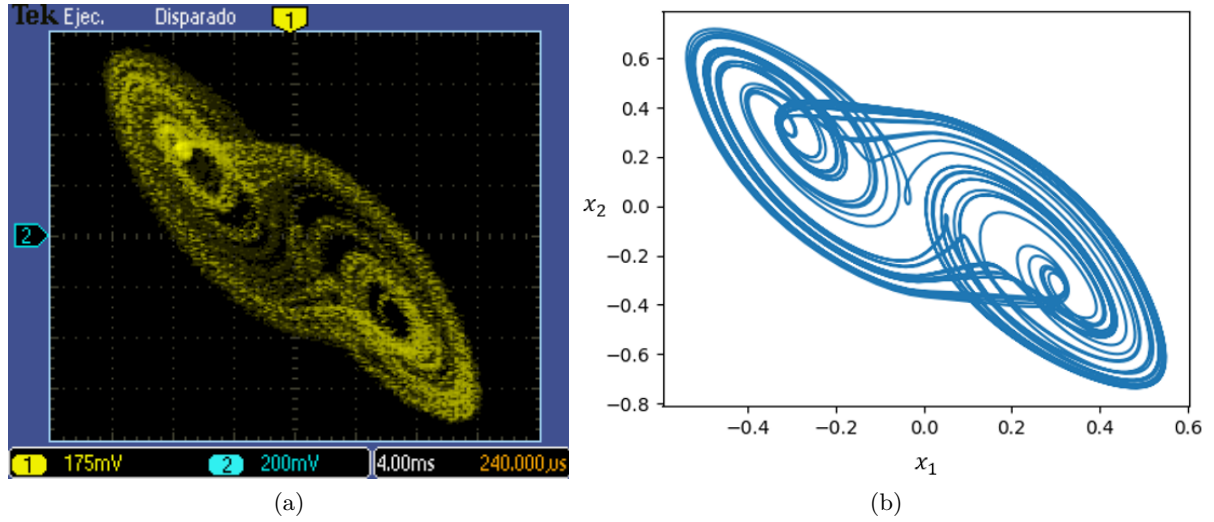


Figura 5.18: Retratos de fase del modelo con función de activación signo para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema.

voltaje externa para generar la señal bias senoidal sobre la neurona x_1 , usando la transformada de Laplace sobre el modelo se tiene

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{-0.15X_1(s) + 1.1\text{ReLU}(X_1(s)) - 1.05\text{ReLU}(X_2(s)) - 0.5\text{sen}(0.52t) + 0.8}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{0.45X_2(s) + 0.95\text{ReLU}(X_1(s)) + 0.75\text{ReLU}(X_2(s)) - 0.75}{s + 2\pi f_0}, \end{aligned} \quad (5.12)$$

con resultado en su implementación presentado en la Figura 5.15.

De la misma forma que el modelo con función ReLU, el modelo con función PWL no requiere de un escalamiento y su transformación queda como

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{1.68\text{satlin}(X_1(s)) + 3.81\text{satlin}(X_2(s)) - 2.23\text{satlin}(X_3(s))}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{-5.4\text{satlin}(X_1(s)) + 1.8\text{satlin}(X_2(s)) - 4.4\text{satlin}(X_3(s))}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{-4.38\text{satlin}(X_1(s)) - 2.2\text{satlin}(X_2(s)) - 0.3\text{satlin}(X_3(s))}{s + 2\pi f_0}, \end{aligned} \quad (5.13)$$

con resultado en su implementación dado en la Figura 5.16 y 5.17 junto con el sistema escalado. Finalmente para el modelo con función de activación signo se tiene

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{-\tanh X_2(s)}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{-\tanh X_3(s)}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{0.65 \tanh(X_1(s)) + 4 \tanh X_2(s) + 3 \tanh X_3(s)}{s + 2\pi f_0}, \end{aligned} \quad (5.14)$$

con evidencias de implementación en la Figura 5.18 para el plano $x_1 - x_2$ y 5.19 para el plano $x_1 - x_3$.

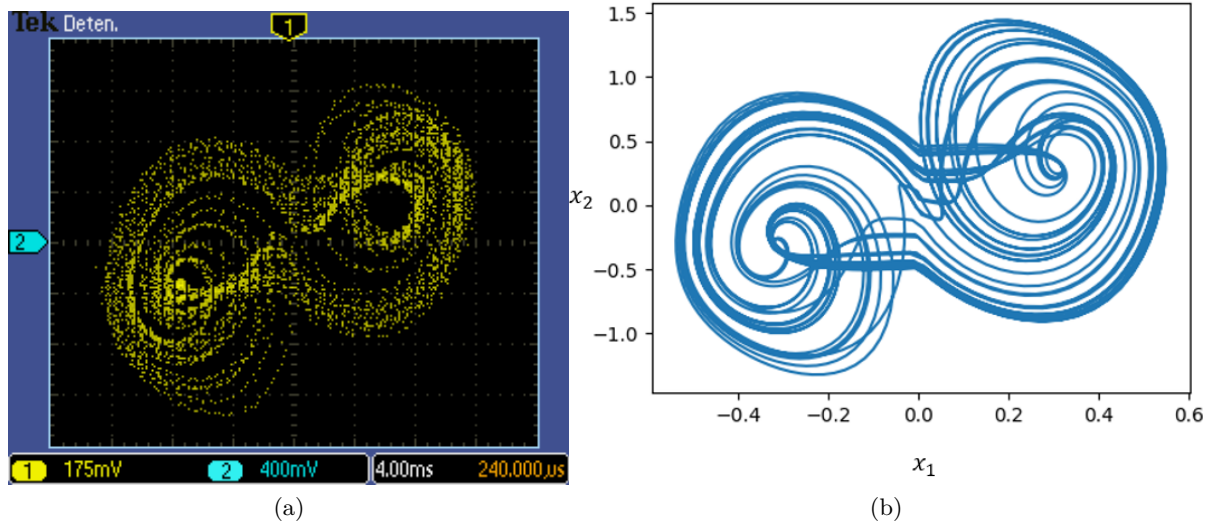


Figura 5.19: Retratos de fase del modelo con función de activación signo para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema.

Conclusiones

Conclusiones del avance II

En el Avance de Tesis II se presentaron las propuestas para los 5 modelos de la red neuronal Hopfield con 5 funciones de activación distintas, para los cuales el sistema presenta un comportamiento caótico. Los modelos con función de activación $\text{ReLu}(\cdot)$ y PWL ya habían sido propuestos en la literatura y de ahí fueron tomados. Con lo que respecta al modelo con función $\tanh(\cdot)$, existe una variedad de modelos ya propuestos, sin embargo, a través de suposiciones con base en un análisis de estabilidad y simulaciones numéricas, se encontró un nuevo modelo con rica dinámica. No existían modelos en la literatura que usarán la función sigmoide o la función signo, para la primera se fue capaz de utilizar una transformación para llevar modelos con función $\tanh(\cdot)$ a modelos con función de activación sigmoide, mientras que para la función de activación se reportó un nuevo modelo, este representa el primer modelo de red neuronal Hopfield con presencia de caos que utiliza una función con un salto abrupto.

Posteriormente, se desarrolló el análisis de los modelos presentados. Este análisis tiene un enfoque en la estabilidad del sistema y lo que esto conlleva, es decir, su acotamiento para alguna región del espacio fásico, la existencia de atractores en el sistema dinámico y el comportamiento del sistema alrededor de sus puntos de equilibrio. Para el análisis de estabilidad existen distintas metodologías, en este trabajo, se utilizó la divergencia del sistema para determinar si el sistema es disipativo y por lo tanto puede contener atractores. Esta metodología es poco común y ha sido recientemente estudiada en diversos documentos de investigación. Otra metodología utilizada es la estabilidad de Lyapunov, a partir de ella fue posible determinar condiciones para las cuales el modelo de redes neuronales Hopfield es asintóticamente estable, además de demostrar que existe una región en la cual la solución del sistema de ecuaciones diferenciales que representa el sistema dinámico esta acotada. Finalmente, en el análisis de los puntos de equilibrio se caracterizó cada uno de estos y se verificaron las observaciones encontradas en la literatura, es importante mencionar que el modelo con la función de activación signo requiere de una aproximación.

El propósito de las redes estudiadas es su adaptación en los parámetros que a ella describen

para que se comporten como sistemas caóticos, la detección de caos es un tema complejo para el cual ya existen diversas métricas, en este avance se trataron con las dos más comunes: los diagramas de bifurcación y los exponentes de Lyapunov. Para el primero se desarrolló un código en Python que grafica los máximos locales de la simulación del modelo, dicha simulación se desarrolló en base al método RK2. Para la obtención de los exponentes de Lyapunov se usó como recurso las librerías del lenguaje Julia, más específicamente, la librería JuliaDynamics permite la obtención de los exponentes de Lyapunov. Con el fin de tener un panorama más extenso sobre el comportamiento caótico de los sistemas se desarrollaron mapas de calor con los exponentes máximos de Lyapunov.

Finalmente, las redes neuronales de Hopfield con tres neuronas fueron implementadas en la tarjeta analógica embebida, para esto fue necesario transformar los modelos en magnitud y frecuencia. Para lo primero se requiere de un cambio de variables mientras que para el segundo se multiplica la parte derecha de los sistemas por una constante, esto último sólo aplicó para el modelo con función sigmoide. Algunos de los bloques en AnadigmDesigner2 están descritos por su respuesta en frecuencia, es necesario entonces llevar los modelos a su respuesta en frecuencia usando la transformada de Laplace. Usando esta transformación fue hallado que el modelo de redes neuronales Hopfield puede ser descrito por un filtro sumador analógico, que a su vez es uno de los bloques disponibles en la biblioteca de AnadigmDesigner2, en base a esta transformación se implementaron los 5 modelos de redes neuronales.

Anexos

5.3. Codigos en AnadigmDesigner2 para la implementación de las redes neuronales de 3 dimensiones

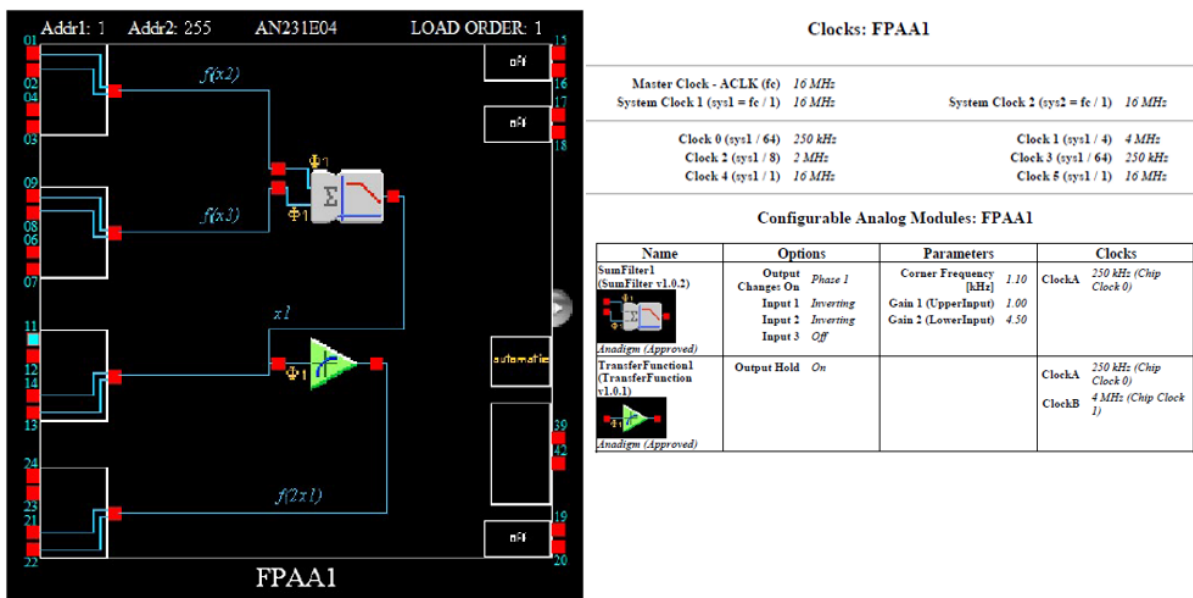


Figura 5.20: Programación y configuración del FPPA 1 para el modelo con función tangente hiperbólica.

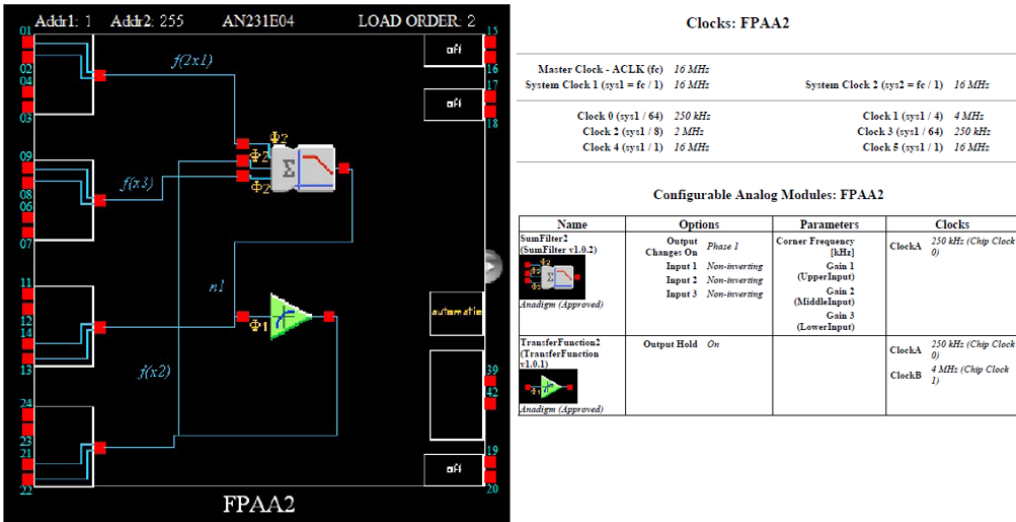


Figura 5.21: Programación y configuración del FPPA 2 para el modelo con función tangente hiperbólica.

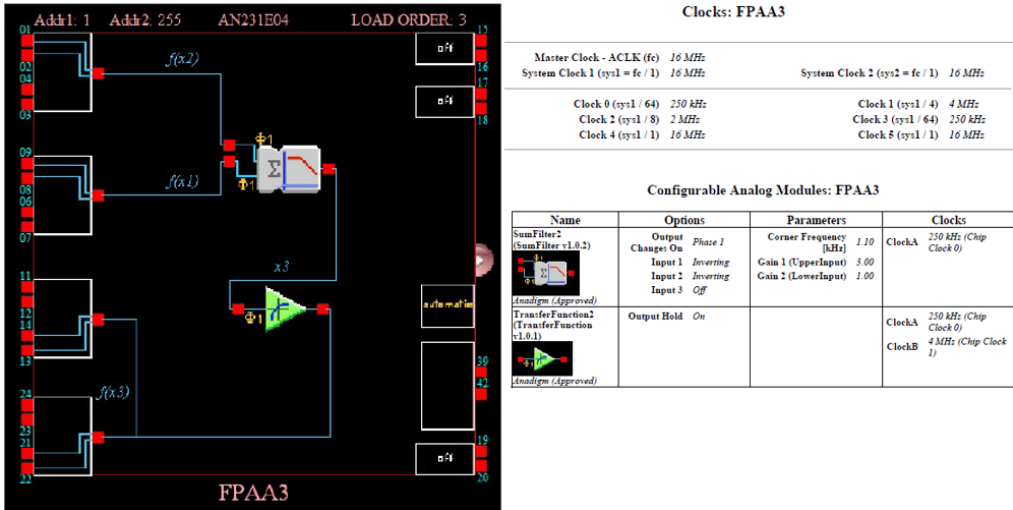


Figura 5.22: Programación y configuración del FPPA 3 para el modelo con función tangente hiperbólica.

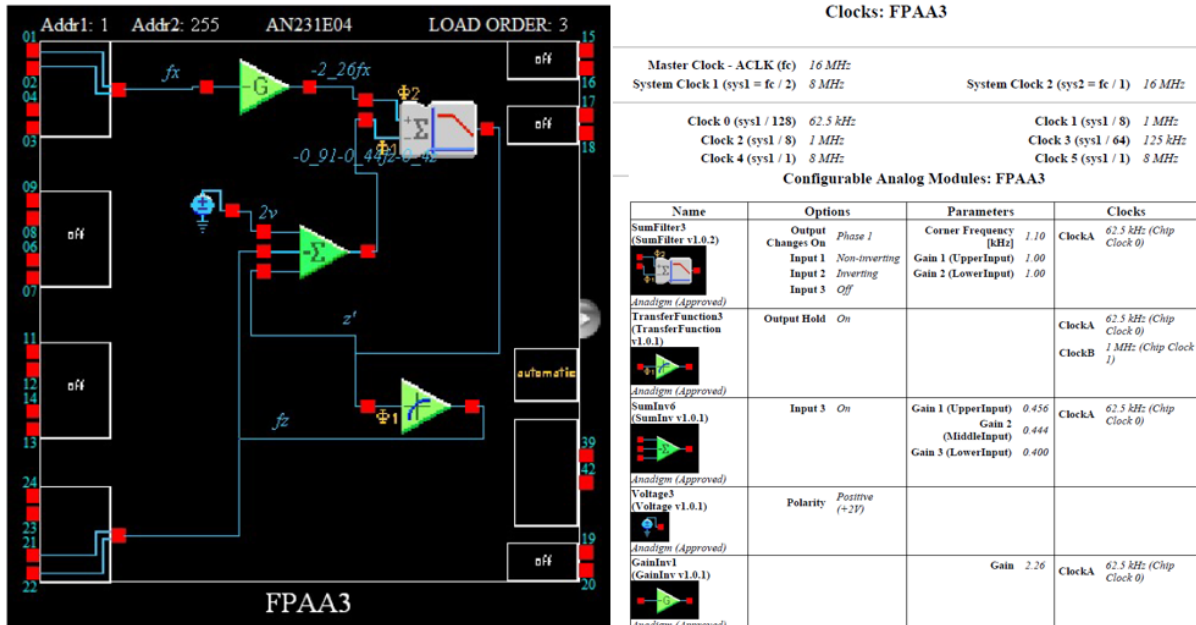


Figura 5.25: Programación y configuración del FPPA 3 para el modelo con función sigmoide.

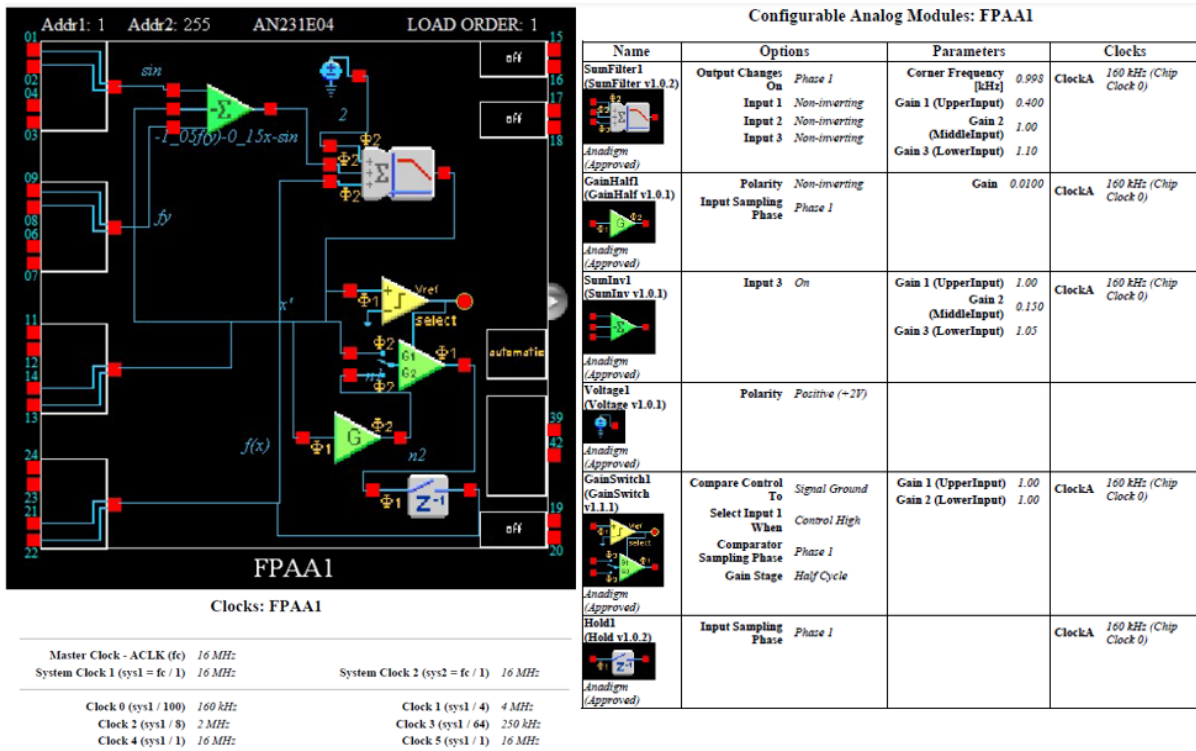


Figura 5.26: Programación y configuración del FPPA 1 para el modelo con función ReLU.

5.3. CODIGOS EN ANADIGMDESIGNER2 PARA LA IMPLEMENTACIÓN DE LAS REDES NEURONAL

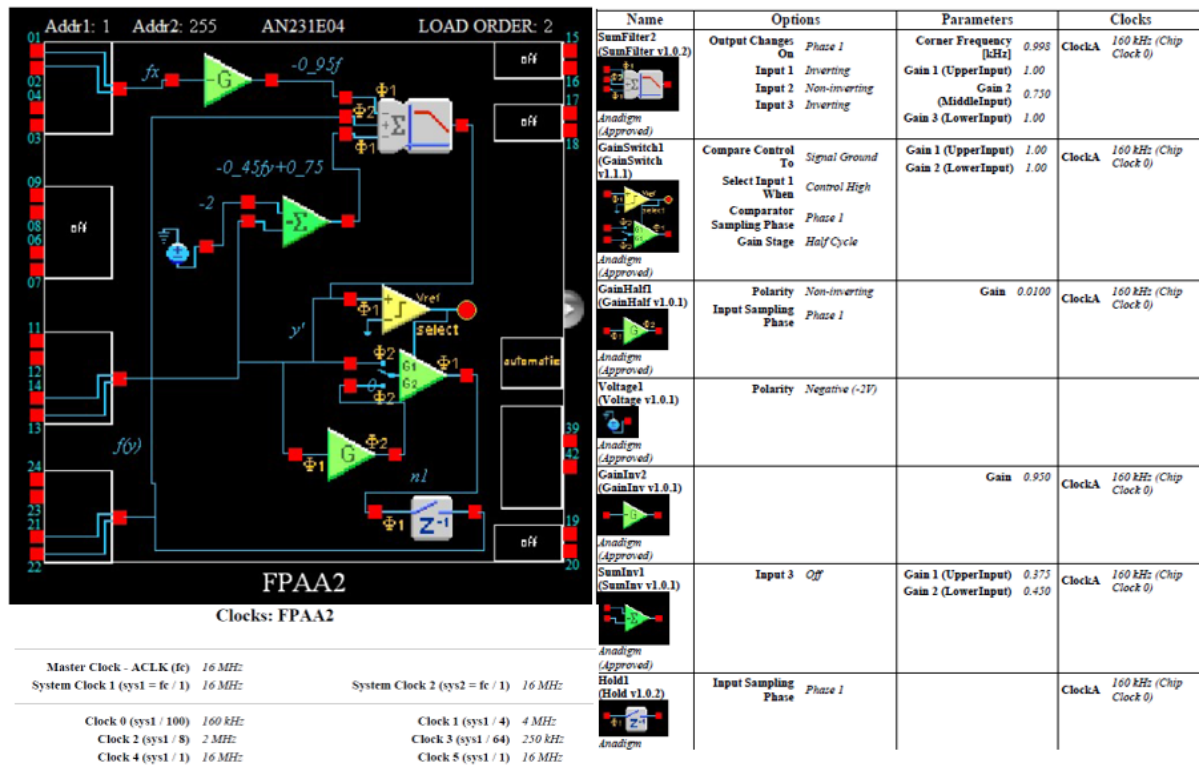


Figura 5.27: Programación y configuración del FPPA 2 para el modelo con función ReLU.

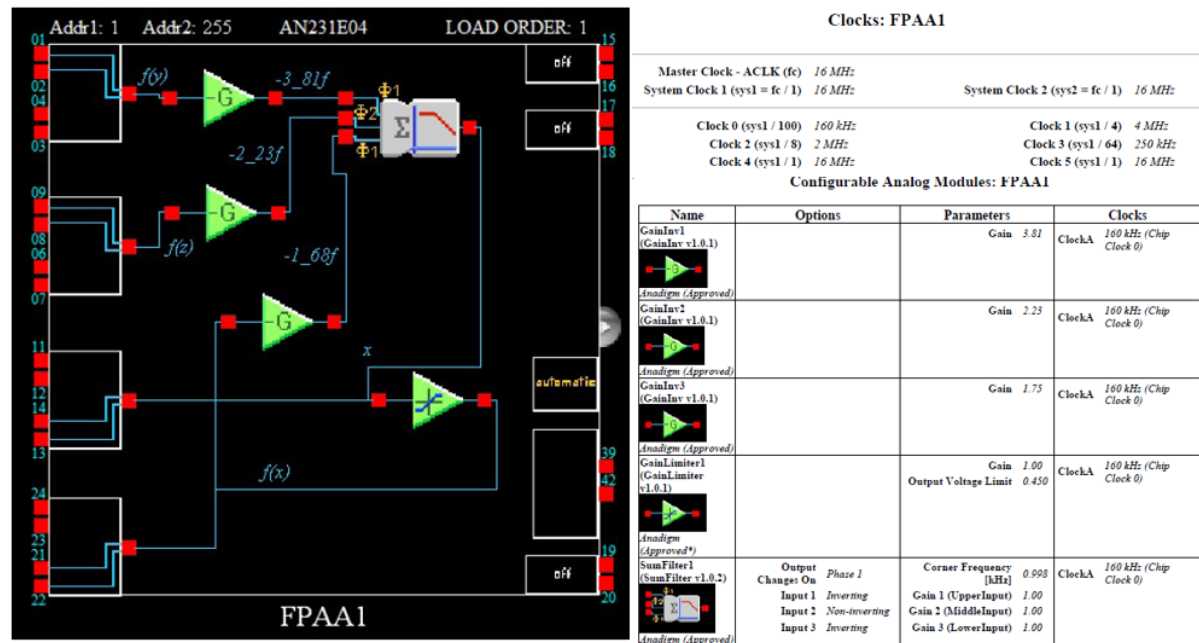


Figura 5.28: Programación y configuración del FPPA 1 para el modelo con función PWL.

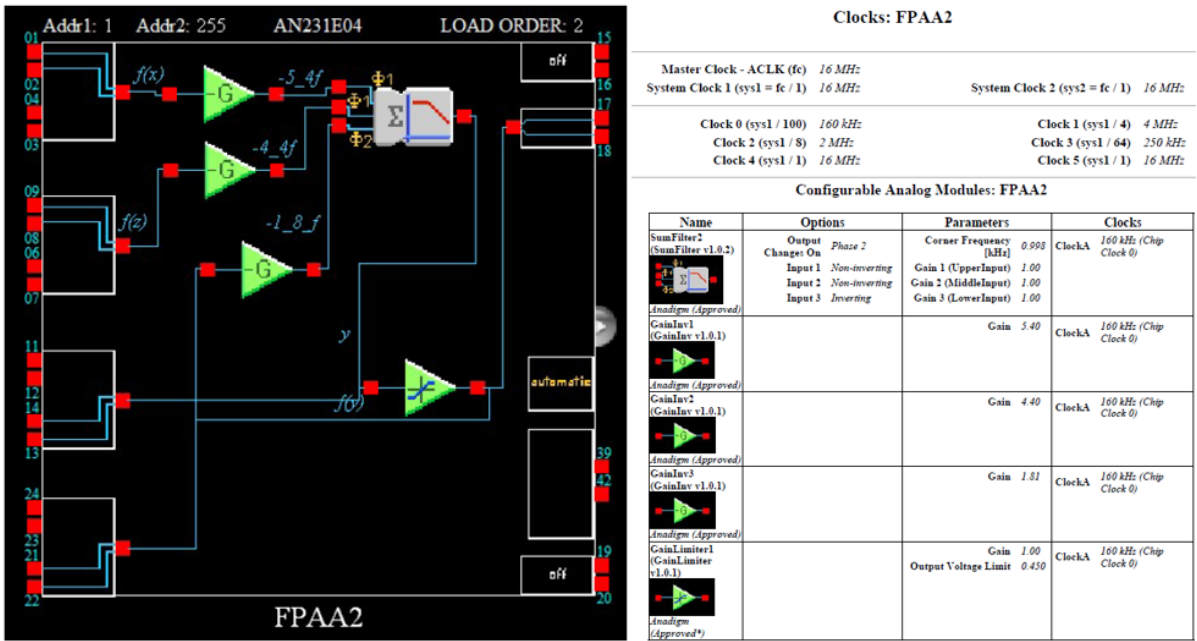


Figura 5.29: Programación y configuración del FPPA 2 para el modelo con función PWL.

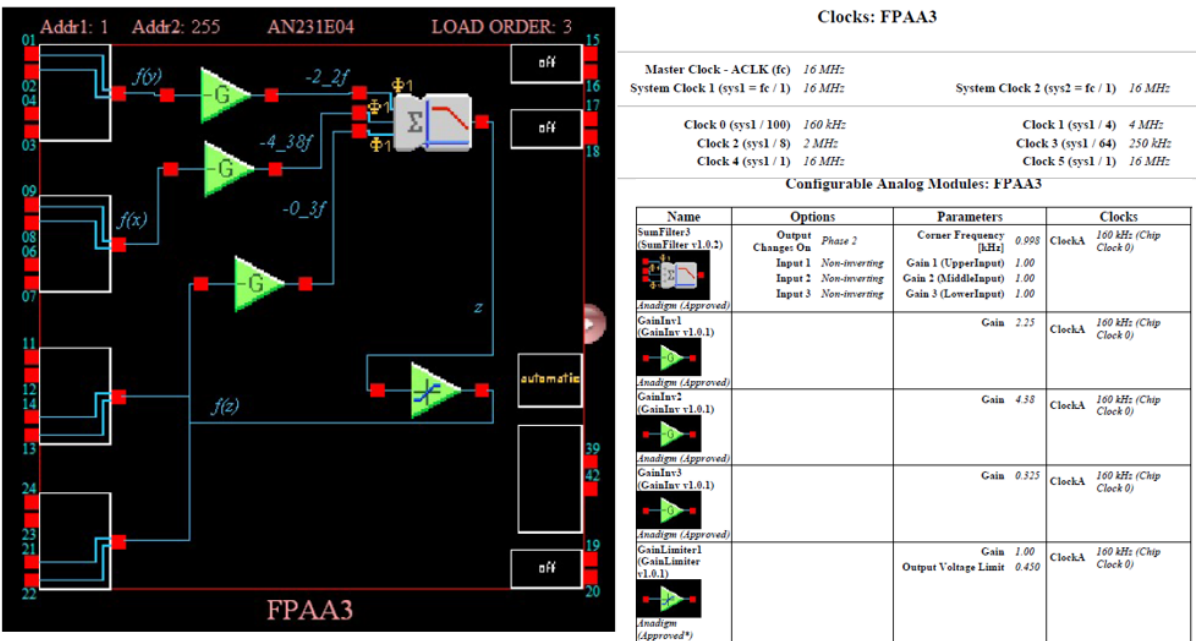


Figura 5.30: Programación y configuración del FPPA 3 para el modelo con función PWL.

5.3. CODIGOS EN ANADIGMDESIGNER2 PARA LA IMPLEMENTACIÓN DE LAS REDES NEURONAL

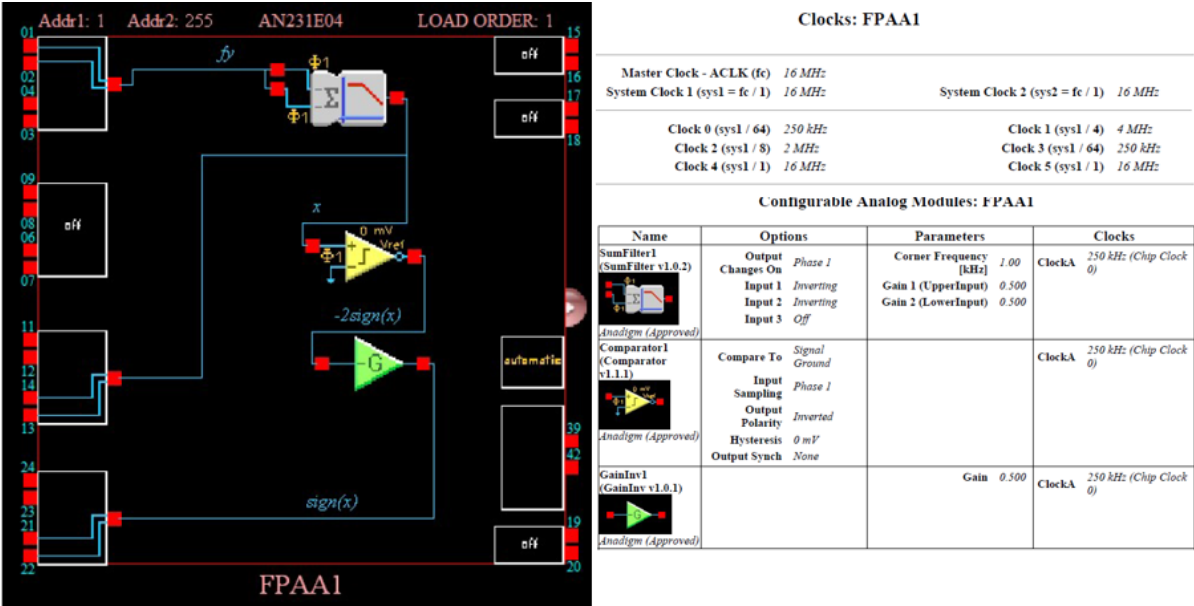


Figura 5.31: Programación y configuración del FPPA 1 para el modelo con función signo.

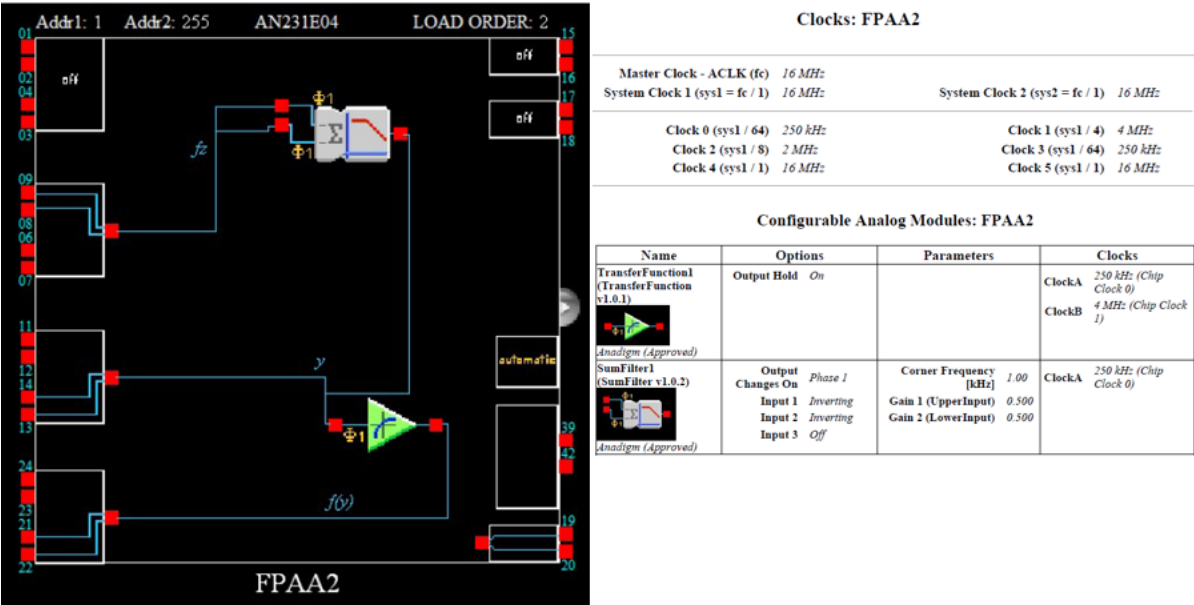


Figura 5.32: Programación y configuración del FPPA 2 para el modelo con función signo.

Bibliografía

- [1] Iqbal M. Batiha, Ramzi B. Albadarneh, Shaher Momani, and Iqbal H. Jebril. Dynamics analysis of fractional-order Hopfield neural networks. *International Journal of Biomathematics*, 13(08):2050083, December 2020.
- [2] Chengjie Chen, Fuhong Min, Yunzhen Zhang, and Han Bao. ReLU-type Hopfield neural network with analog hardware implementation. *Chaos, Solitons & Fractals*, 167:113068, February 2023.
- [3] Zheng Peng-Sheng, Tang Wan-Sheng, and Zhang Jian-Xiong. A new chaotic Hopfield network with piecewise linear activation function. *Chinese Physics B*, 19(3):030514, March 2010.
- [4] Nathan W. Churchill, Robyn Spring, Cheryl Grady, Bernadine Cimprich, Mary K. Askren, Patricia A. Reuter-Lorenz, Mi Sook Jung, Scott Peltier, Stephen C. Strother, and Marc G. Berman. The suppression of scale-free fMRI brain dynamics across three different sources of effort: aging, task novelty and task difficulty. *Scientific Reports*, 6(1):30895, August 2016.
- [5] Bruce Alberts. *Molecular biology of the cell*. W. W. Norton & Company, New York, seventh edition edition, 2022.
- [6] Luis Alfredo Baratas Díaz. La obra neuro-embriológica de Santiago Ramón y Cajal. *Historiam Illustrandam*, 1(17):259–279, 1997.
- [7] N. B. Karayiannis and A. N. Venetsanopoulos. *Artificial Neural Networks*. Springer US, Boston, MA, 1993.
- [8] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [9] Charu C. Aggarwal. *Neural networks and deep learning: a textbook*. Springer, Cham, second edition edition, 2023.

- [10] Liangqu Long and Xiangming Zeng. *Beginning deep learning with TensorFlow work with Keras, MNIST data sets, and advanced neural networks*. Apress, United States, 2022.
- [11] John J. Hopfield and David W. Tank. Computing with Neural Circuits: A Model. *Science*, 233(4764):625–633, August 1986.
- [12] Anke Meyer-Baese and Volker Schmid. Transformation and Signal-Separation Neural Networks. In *Pattern Recognition and Signal Analysis in Medical Imaging*, pages 245–289. Elsevier, 2014.
- [13] Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal Hopfield Networks: A General Framework for Single-Shot Associative Memory Models. 2022.
- [14] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need. 2020.
- [15] Karthikeyan Rajagopal, Jesus M. Munoz-Pacheco, Viet-Thanh Pham, Duy Vo Hoang, Fawaz E. Alsaadi, and Fuad E. Alsaadi. A Hopfield neural network with multiple attractors and its FPGA design. *The European Physical Journal Special Topics*, 227(7-9):811–820, October 2018.
- [16] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, March 2021.
- [17] Daniel Clemente-López, Jesus M. Munoz-Pacheco, and Jose De Jesus Rangel-Magdaleno. A Review of the Digital Implementation of Continuous-Time Fractional-Order Chaotic Systems Using FPGAs and Embedded Hardware. *Archives of Computational Methods in Engineering*, 30(2):951–983, March 2023.
- [18] Han Bao, ZhuGuan Chen, JianMing Cai, Quan Xu, and BoCheng Bao. Memristive cyclic three-neuron-based neural network with chaos and global coexisting attractors. *Science China Technological Sciences*, 65(11):2582–2592, November 2022.
- [19] Bocheng Bao, Hui Qian, Quan Xu, Mo Chen, Jiang Wang, and Yajuan Yu. Coexisting Behaviors of Asymmetric Attractors in Hyperbolic-Type Memristor based Hopfield Neural Network. *Frontiers in Computational Neuroscience*, 11:81, August 2017.

- [20] Z. Tabekoueng Njitacke, Sami Doubla Isaac, J. Kengne, A. Nguomkam Negou, and Gervais Dolvis Leutcho. Extremely rich dynamics from hyperchaotic Hopfield neural network: Hysteretic dynamics, parallel bifurcation branches, coexistence of multiple stable states and its analog circuit implementation. *The European Physical Journal Special Topics*, 229(6-7):1133–1154, March 2020.
- [21] Chengjie Chen, Fuhong Min, Fei Hu, Jianming Cai, and Yunzhen Zhang. Analog/digital circuit simplification for Hopfield neural network. *Chaos, Solitons & Fractals*, 173:113727, August 2023.
- [22] Z.T. Njitacke and J. Kengne. Complex dynamics of a 4D Hopfield neural networks (HNNs) with a nonlinear synaptic weight: Coexistence of multiple attractors and remerging Feigenbaum trees. *AEU - International Journal of Electronics and Communications*, 93:242–252, September 2018.
- [23] Z. Tabekoueng Njitacke, J. Kengne, and H. B. Fotsin. Coexistence of Multiple Stable States and Bursting Oscillations in a 4D Hopfield Neural Network. *Circuits, Systems, and Signal Processing*, 39(7):3424–3444, July 2020.
- [24] Samuel Shapero, Christopher Rozell, and Paul Hasler. Configurable hardware integrate and fire neurons for sparse approximation. *Neural Networks*, 45:134–143, September 2013.
- [25] Pranav O. Mathews and Jennifer O. Hasler. Physical Computing for Hopfield Networks on a Reconfigurable Analog IC. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, Monterey, CA, USA, May 2023. IEEE.
- [26] Derong Liu, editor. *Advances in neural networks–ISNN 2007: 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007: proceedings*. Number 4491-4493 in Lecture notes in computer science. Springer, Berlin ; New York, 2007.
- [27] Jay Kant Pratap Singh Yadav, Laxman Singh, and Zainul Abidin Jaffery. Optimization of Hopfield Neural Network (HNN) using multiconnection and Lyapunov Energy Function (LEF) for storage and recall of handwritten images. *Sādhanā*, 48(1):26, February 2023.
- [28] Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. *Computational intelligence: a methodological introduction*. Texts in computer science. Springer, Cham, third edition edition, 2022.
- [29] Hairong Lin, Chunhua Wang, Fei Yu, Jingru Sun, Sichun Du, Zekun Deng, and Quanli Deng. A Review of Chaotic Systems Based on Memristive Hopfield Neural Networks. *Mathematics*, 11(6):1369, March 2023.

- [30] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*, 3:4, February 2020.
- [31] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: an introduction to dynamical systems*. Textbooks in mathematical sciences. Springer, New York, NY Heidelberg, repr. edition, 2010.
- [32] Morris W. Hirsch, Stephen Smale, and Robert L. Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic Press, Waltham, MA, 3rd ed edition, 2013.
- [33] Steven Strogatz and Mitchal Dichter. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, Boca Raton, FL, second edition edition, 2018.
- [34] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, March 1963.
- [35] Robert L. Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. CRC Press, 1 edition, May 2018.
- [36] Hernando Fernandez-Canque. *Analog electronics applications: fundamentals of design and analysis*. CRC Press, Boca Raton London New York, 2017.
- [37] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Numerical analysis*. Cengage Learning, Boston, MA, tenth edition edition, 2016.
- [38] Huaguang Zhang, Zhanshan Wang, and Derong Liu. A Comprehensive Review of Stability Analysis of Continuous-Time Recurrent Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1229–1262, July 2014.
- [39] Chengjie Chen and Fuhong Min. ReLU-type memristor-based Hopfield neural network. *The European Physical Journal Special Topics*, 231(16-17):2979–2992, November 2022.
- [40] Sajad Jafari, Julien C. Sprott, and Soroush Dehghan. Categories of Conservative Flows. *International Journal of Bifurcation and Chaos*, 29(02):1950021, February 2019.
- [41] Hassan K. Khalil. *Nonlinear systems. Hauptbd.* Prentice Hall, Upper Saddle River, NJ, 3. ed edition, 2002.
- [42] J. C. Sprott. Strange attractors with various equilibrium types. *The European Physical Journal Special Topics*, 224(8):1409–1419, July 2015.

- [43] Mohammad Saleh Tavazoei and Mohammad Haeri. Chaotic attractors in incommensurate fractional order systems. *Physica D: Nonlinear Phenomena*, 237(20):2628–2637, October 2008.
- [44] George Datseris and Ulrich Parlitz. *Nonlinear dynamics: a concise introduction interlaced with code*. Undergraduate lecture notes in physics. Springer, Cham, 2022.
- [45] Sitabhra Sinha. Chaotic Dynamics in Iterated Map Neural Networks with Piecewise Linear Activation Function. *Fundamenta Informaticae*, 37(1,2):31–50, 1999.
- [46] Anadigm. Using the anadigm transferfunction cam, 2009. https://www.anadigm.com/_doc/DB0223009-U210.pdf [Accessed: 28-05-2024].

Avance de Tesis III

*Metodología para la implementación analógica embebida y
diseño de una red de n cantidad de enrollamientos*

Periodo de trabajo
Del 5 de agosto 2024 - 14 de diciembre de 2024

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

**Maestría en Ciencias de la Electrónica
Opción en Automatización**



Avance de Tesis III

Diseño de una red neuronal Hopfield en hardware embebido

Alumno: Diego Ruiz Sánchez de la Vega

**Directores de Tesis: Dra. Olga Guadalupe Félix Beltrán (FCE-BUAP)
Dr. Jesús Manuel Muñoz Pacheco (FCE-BUAP)**

Puebla, Puebla.

Noviembre de 2024.

Resumen

Resumen general

En esta tesis se lleva a cabo el diseño e implementación de una red neuronal Hopfield de tres y cuatro neuronas en hardware embebido programable analógico, caracterizando la dinámica no lineal que a ella describe, así como también se realiza un estudio sobre las distintas funciones no lineales de activación, evaluando los efectos en el proceso de aprendizaje de la red neuronal Hopfield. Con este fin inicialmente se desarrolla una revisión bibliográfica sobre el estado del arte de las redes neuronales artificiales, en específico sobre la red neuronal Hopfield, sus antecedentes, principales aplicaciones, implementaciones tanto en circuitos analógicos como con dispositivos programables analógicos y digitales, así como las ecuaciones que describen el modelo de la red neuronal Hopfield. Posteriormente, se desarrolla un análisis exhaustivo sobre la dinámica no lineal del sistema, este incluye la evaluación del proceso de aprendizaje de la red neuronal, así como, el análisis en base a caos para finalmente implementar el modelo de la red neuronal Hopfield con distintas funciones de activación en un dispositivo analógico programable, el cual, requiere sus propias ecuaciones de diseño, las cuales también son desarrolladas. Finalmente, con todo el trabajo realizado, se elabora una metodología para la implementación de las redes neuronales Hopfield en las tarjetas analógicas embebidas utilizadas a lo largo de esta tesis.

Resumen Avance de Tesis I

En el Avance de Tesis I estudian los antecedentes de las redes neuronales artificiales, su comparación con las neuronas biológicas y se presenta el primer modelo matemático que a ellas describe, posteriormente, se presenta el modelo de las redes neuronales Hopfield (HNN) general y específico para 3 y 4 neuronas, dado que es el tema de interés en esta tesis. Simultáneamente, se realiza un estudio sobre la implementación de las HNN, ya sea con circuito embebidos o no, su respuesta y aplicaciones principales. Otro tema de interés en esta tesis son los sistemas dinámicos caóticos esto debido a que, dados ciertos parámetros, la red neuronal Hopfield presenta caos. Con respecto a este tema, una investigación es realizada con relación a los sistemas

dinámicos caóticos y a las HNN en modo caótico. La implementación en hardware embebido de redes neuronales es un tema de recurrente estudio en la comunidad científica, por lo que una investigación a fondo en el diseño en hardware programable de las HNN, resulta evidente en el cual se incluye, un estudio e investigación sobre los dispositivos analógicos programables FPAA's, dado que este circuito será el utilizado para el desarrollo de esta tesis. Finalmente, para el análisis del modelo de la red neuronal Hopfield, es necesario desarrollar herramientas matemáticas analíticas y numéricas, con lo que respecta a este avance, se presentan dos métodos para la integración numérica por medio de los métodos de Runge-Kutta de segundo y cuarto orden.

Resumen Avance de Tesis II

En el Avance de Tesis II se presentaron los modelos de las redes neuronales Hopfield de tres neuronas específicos que serán estudiados en este trabajo, para cada modelo se utiliza una función de activación distinta. Dos de estos modelos con comportamiento caótico no habían sido reportados en la literatura y son presentados en este trabajo, mientras que los otros tres sistemas caóticos ya habían sido reportados en [1], [2] y [3]. Para cada uno de estos modelos se realiza un análisis dinámico en donde se demuestra la existencia de atractores en los modelos y se desarrolla en análisis de estabilidad de Lyapunov para dar un criterio para la estabilidad asintótica del sistema. Posteriormente, se utilizan los diagramas de bifurcación y exponentes de Lyapunov como métricas para identificar las regiones caóticas de los modelos de redes neuronales Hopfield. Finalmente, se desarrollan las funciones de activación en base a los bloques que ofrece la distribuidora de las tarjetas analógicas para posteriormente implementar todos los modelos en las tarjetas analógicas programables, para esto se desarrollan las ecuaciones de diseño que representan los modelos de manera que se desarrollen los programas en AD2 con comportamiento caótico y registrar los resultados obtenidos.

Resumen Avance de Tesis III

En el Avance de Tesis III se presenta un modelo de la red neuronal Hopfield con 4 neuronas cuyo comportamiento es hipercaótico. De dicho modelo se realiza el respectivo análisis en donde se demuestra de manera numérica el acotamiento y la existencia de atractores de la red, posteriormente, se halla las zonas densas y sensibles a las condiciones iniciales inicial del modelo estudiado. Este mismo modelo es implementado en FPAAs usando un diseño de circuitos en base a filtros analógicos. Mas adelante, se realiza una interfaz a base a distintas topologías de circuitos que permiten la implementación de cualquier red neuronal Hopfield de 3 y 4 neuronas, estos circuitos son realizados en base a integrados. La interfaz es diseñada en

Qt-Python y permite al usuario colocar los parámetros de la red neuronal, así como también, incluir cualquier tipo de función de activación en el modelo HNN. Finalmente, se desarrolla una red neuronal Hopfield con infinitos atractores usando solo una no linealidad en el sistema, entre las diversas propiedades y características de dicha red se encuentra la multiestabilidad. Dicho sistema es implementado en las tarjetas analógicas usando la interfaz desarrollada.

Índice general

Resumen	II
Introducción	IV
Objetivos	X
Cronograma de Actividades	XII
Diagrama de bloques de la tesis	XIV
1. Estado del arte	1
2. Marco Teórico	4
2.1. Redes Neuronales Artificiales	4
2.1.1. Redes Neuronales Hopfield	5
2.1.2. Funciones de activación	6
2.2. Sistemas dinámicos	8
2.2.1. Representación matemática de los sistemas dinámicos	8
2.2.2. Puntos de equilibrio	9
2.2.3. Sistemas dinámicos caóticos	10
2.2.4. Redes Neuronales Hopfield en modo caótico	11
2.3. Dispositivos programables	12
2.3.1. Hardware Analógico Embebido	12
2.4. Métodos numéricos para la solución de sistemas dinámicos	13
3. Análisis de estabilidad de la red neuronal Hopfield	17
3.1. Propiedades de una función de activación	18
3.2. Modelos propuestos	19
3.2.1. Modelo con función de activación tangente hiperbólica	19
3.2.2. Modelo obtenido con función de activación sigmoide	20

3.2.3.	Modelo con función de activación signo	22
3.3.	Modelos reportados en la literatura	23
3.3.1.	Modelo con función ReLU	23
3.3.2.	Modelo con función de activación PWL	24
3.3.3.	Modelo de red neuronal Hopfield de 4 neuronas	25
3.4.	Análisis en base a la divergencia del sistema dinámico	26
3.4.1.	Criterio numérico para la existencia de atractores en sistemas dinámicos no uniformes	30
3.4.2.	Atractores en el modelo de HNN con FA tangente hiperbólica	33
3.4.3.	Atractores en el modelo de HNN con FA signo	35
3.4.4.	Atractores en el modelo de HNN con FA sigmoide	35
3.4.5.	Atractores en el modelo de HNN con FA $\text{ReLu}(x)$	37
3.4.6.	Atractores en el modelo de HNN con FA PWL.	38
3.4.7.	Atractores en el modelo HNN de 4 dimensiones	40
3.5.	Análisis de estabilidad en el sentido de Lyapunov	41
3.6.	Análisis de estabilidad alrededor de los puntos de equilibrio	45
4.	Dinámica caótica	53
4.1.	Diagramas de bifurcación	53
4.1.1.	Diagramas de bifurcación en Python	54
4.1.2.	Diagramas de bifurcación del modelo con función de activación tangente hiperbólica	57
4.1.3.	Diagramas de bifurcación del modelo con función de activación sigmoide	58
4.1.4.	Diagramas de bifurcación del modelo con función de activación ReLU	58
4.1.5.	Diagramas de bifurcación del modelo con función de activación PWL	60
4.1.6.	Diagramas de bifurcación del modelo con función de activación signo	61
4.1.7.	Diagramas de bifurcación del modelo con 4 neuronas	61
4.2.	Exponentes de Lyapunov	61
5.	Implementación de las redes neuronales Hopfield	73
5.1.	Desarrollo de funciones de activación	73
5.1.1.	Implementación de la función signo	73
5.1.2.	Implementación de la función ReLu	75
5.1.3.	Implementación de la función PWL	75
5.1.4.	Implementación de la función tangente hiperbólica y sigmoide	77
5.2.	Programación e implementación de las HNN's de 3 neuronas	77
5.2.1.	Implementación del modelo con FA tangente hiperbólica	82
5.2.2.	Implementación del modelo con FA sigmoide	82

5.2.3. Implementación del modelo con FA $\text{ReLU}(x)$	87
5.2.4. Implementación del modelo con FA PWL	87
5.3. Implementación del modelo de hipercaótico HNN de 4 neuronas	88
6. Desarrollo de una interfaz como metodología para la implementación del modelo HNN	93
6.1. Topología de los circuitos en AN2	93
6.2. Desarrollo de la interfaz en Qt-Python	95
7. Modelo de HNN con infinitos atractores	102
7.1. El modelo propuesto	102
7.2. Retratos de fase, diagramas de bifurcación y exponentes de Lyapunov	106
7.3. Implementación en un ambiente electrónico	108
Conclusiones	112
Ponencia en congreso y publicación	113
Anexos	116
.1. Codigos en AnadigmDesigner2 para la implementación de las redes neuronales de 3 dimensiones	116
Bibliografía	124

Introducción

Antecedentes

El estudio del comportamiento del cerebro humano tiene profundas raíces en la historia de la ciencia y la medicina, desde la antigüedad grandes filósofos especulaban sobre su funcionamiento, los antiguos griegos plantearon ideas sobre “maquinas pensantes” [4]. El siglo XIX marcó un avance importante con respecto al estudio de la biología; en 1839 Schleiden y Schwann formulan la Teoría Celular, que establece que todos los seres vivos están formados por células como unidades fundamentales [5], esto por supuesto incluye al cerebro.

Una de las células básicas que componen al sistema nervioso es la neurona, las neuronas procesan y transmiten información en forma de señales eléctricas, como fue propuesto por el científico español Santiago Ramón y Cajal [6], en su trabajo sobre la transmisión neuronal, en el cual se considera a la neurona como un interruptor que se “dispara” dependiendo de las señales de otras neuronas, estas señales pueden ser inhibitorias si tienden a impedir que la neurona se active o excitatorias si provocan que la neurona se active, con estas señales la neurona realiza suma ponderada y, dependiendo del valor de esta suma, se dice que la neurona está “activa” o no.

La neurona es una de las unidades fundamentales del sistema nervioso y se divide en cuatro partes importantes: el núcleo, el cuerpo de la célula, la dendrita y el axón (véase Figura 1). El cuerpo de la célula contiene al núcleo, el cual se encarga de producir la energía de la neurona, la entrada de la neurona que trata con las señales de las demás neuronas es denominada como dendrita, mientras que la salida es denominada como axón; a la conexión entre las neuronas se le conoce como sinapsis.

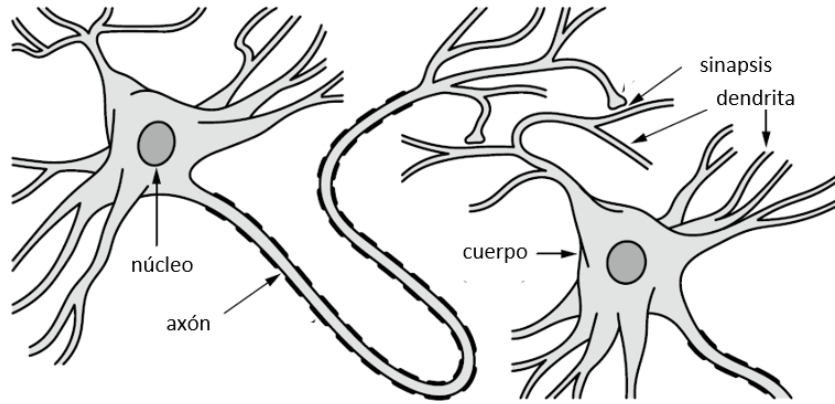


Figura 1: Neurona biológica.

Inspirados por el comportamiento de las neuronas biológicas, los investigadores y científicos han desarrollado las denominadas redes neuronales artificiales (ANN, por sus siglas en inglés), las cuales son dispositivos que simulan el comportamiento de las redes neuronales biológicas. El primer modelo de ANN fue el desarrollado por Warren McCulloch y Walter Pitts en 1943 [7], quienes proponen un modelo que toma múltiples entradas binarias ponderadas, suma estas entradas y aplica una función de umbral (véase Figura 2). Si la suma supera el umbral, la neurona emite una señal de salida. Este enfoque permitió realizar cálculos lógicos y de decisión basados en la entrada recibida, semejante al comportamiento de las neuronas biológicas.

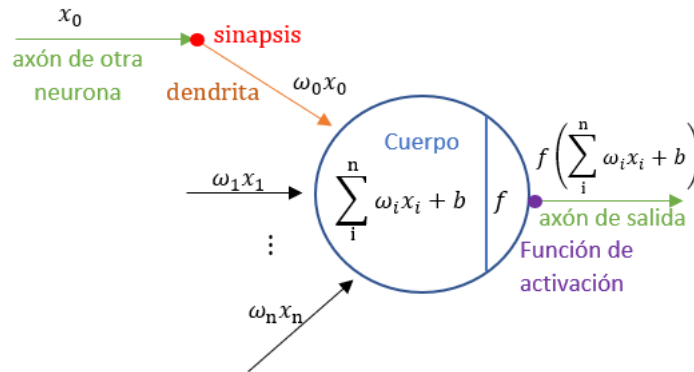


Figura 2: Modelo matemático de una neurona artificial.

En 1958, Frank Rosenblatt, padre del perceptrón [8], consiguió implementar la primera máquina capaz de aprender, distinguir y separar cartas marcadas utilizando el modelo matemático McCulloch-Pitts. El modelo del perceptrón, contiene solo una capa de entradas y una única neurona de salida, donde cada conexión entre las neuronas tiene un valor específico denominado “peso”, el cálculo del valor de la neurona de salida se realiza de la siguiente forma: se realiza el producto entre la entrada con su respectivo peso, estos valores se suman y, final-

mente, el valor calculado se introduce como argumento en una función denominada “función de activación”, para el perceptrón la función utilizada es la función *signo*, la cual es adecuada para una clasificación [9].

Si bien, el trabajo de Rosenblatt no fue bien recibido por la comunidad científica debido a sus limitantes, años después se desarrollaron modelos que solucionaban las deficiencias del perceptrón, algunos de ellos son: las redes neuronales convoluciones, redes neuronales recurrentes, perceptrón multicapa, entre otros [10]. Tiempo después al trabajo desarrollado por McCulloch-Pitts, John Hopfield propone un nuevo modelo para las ANN, denominado como red neuronal Hopfield o HNN por sus siglas en inglés, esta es una red de tipo recurrente y completamente conectada, esto es, a diferencia de los modelos de McCulloch-Pitts, las neuronas del modelo de la HNN están conectadas con todas las demás, además de tener cierta dependencia con su estado anterior, este tipo de ANN se utiliza generalmente para describir algunos procesos cerebrales, como por ejemplo, el proceso de aprendizaje y el concepto de memoria asociativa, la ecuación que generaliza cada neurona de la HNN viene dada por [11]

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^n \omega_{ij} \tanh(x_j) + I_i,$$

donde x_i es la entrada de la neurona, i es el número de neurona, $\tanh(\cdot)$ es la función de activación, w_{ij} es el peso entre las entradas de la neurona, c es un parámetro constante y finalmente la I_i es un factor de sesgo o bias.

En la Figura 3 se ilustra el diagrama a bloques de una neurona Hopfield, la cual podemos asociar a la función de propagación con la sumatoria de todas las sinapsis que posee la neurona, mientras que el bloque $f(x_i)$ representa la función de activación. Para las redes neuronales Hopfield es utilizada la función tangente hiperbólica $\tanh(\cdot)$ y, finalmente, existe un bloque de derivada, el cual nos permite solucionar la ecuación diferencial para adquirir el valor actual de la neurona [12, 13].

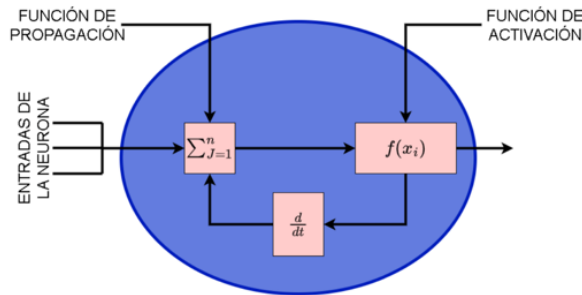


Figura 3: Diagrama en bloques de una red neuronal Hopfield.

En este escenario, debido a las capacidades de aprendizaje de las neuronas artificiales, es requerido un diseño electrónico en cual puede ser usado para aplicaciones de máquinas inteligentes, las cuales requieren implementaciones en hardware optimizadas, es decir, rápidas, baja complejidad computacional y con bajo consumo de potencia. Note que la HNN, al utilizar una menor cantidad de neuronas, requiere un menor consumo de potencia electrónica o computacional para su funcionamiento, además, las HNN son muy utilizadas debido a que, comparada con otros modelos de ANN, no existe una mejor red neuronal para la identificación de patrones.

Planteamiento del problema

En esta investigación se tiene como objetivo realizar una metodología para el desarrollo y diseño en la implementación de redes neuronales Hopfield en dispositivos programables analógicos, además de realizar un análisis exhaustivo con respecto a la dinámica del sistema y evaluar los efectos en el proceso de aprendizaje usando distintas funciones de activación.

Justificación

Las redes neuronales son sistemas que nos permiten dar solución a problemáticas en donde se requiere de dispositivos inteligentes y adaptables, en particular, las redes neuronales Hopfield nos permiten, cifrar datos, reconocimiento de imágenes y, principalmente, la detección de patrones, entre otros. En específico, la detección de patrones es crucial para lograr, en un futuro, la comunicación entre dispositivos inteligentes capaces de manejarse de manera autónoma sin la necesidad de la supervisión humana.

Actualmente, no hay red neuronal que sea capaz de almacenar una gran cantidad de patrones como lo hace la red neuronal Hopfield, gracias a que sus neuronas son capaces de almacenar más información que cualquier otra red neuronal existente [14].

Si bien la implementación de este tipo de red neuronal ya ha sido desarrollada por trabajos pasados, el desarrollo de una metodología para la implementación en circuitos electrónicos de las redes neuronales Hopfield no es del todo clara y este es uno de los temas que esta tesis busca resolver.

El modelo de la red neuronal Hopfield está descrito por un sistema de ecuaciones diferenciales no lineales, su estructura, dados ciertos parámetros, presenta un comportamiento caótico, en otras palabras, bajo parámetros adecuados la implementación de una red neuronal Hopfield nos permite obtener un sistema caótico y, con él todas las propiedades que ellos nos ofrecen. Considerando lo anterior, en este trabajo se realiza un análisis y estudio de la red neuronal en modo caótico.

Dentro del modelo que a las redes neuronales describe, las funciones de activación son

fundamentales, a través de los años distintas funciones de activación han sido desarrolladas, cada función de activación da distintas propiedades dinámicas a la red neuronal, así como también, afecta directamente el proceso de aprendizaje de dicha red neuronal, y este análisis es otro de los objetivos primarios de este trabajo.

Estructura del documento

El presente documento se encuentra seccionado por capítulos, donde cada uno aborda diferentes aspectos relativos al problema de investigación. En un principio se abordan los antecedentes, conceptos básicos, los objetivos de la tesis, el cronograma y el diagrama de bloques de la tesis, que se considera como la metodología de la misma. En el primer capítulo se aborda el estado del arte relacionado a las redes neuronales Hopfield, sistemas caóticos y su implementación en circuitos electrónicos. En el segundo capítulo se desarrolla en detalle los temas de interés antes mencionados. En el tercer capítulo se presentan los modelos con los que se trabajara en la Tesis, todos presentan una función de activación distinta en busca de abordar la mayor cantidad de fenómenos dinámicos. Al momento de presentar cada modelo junto con su función de activación también se dan algunas características o propiedades de interés sobre la función. En uno de los modelos se desarrolla una transformación para llevar de un modelo conocido a otro con una función de activación distinta. Así mismo se desarrolla un análisis dinámico en relación a la estabilidad del modelo de la red neuronal Hopfield. En el capítulo cuatro se aborda la dinámica caótica de los modelos de interés usando métricas para la identificación de zonas caóticas, como lo son los diagramas de bifurcación y exponentes de Lyapunov. La implementación de las redes neuronales Hopfield se muestra en el capítulo cinco. En el capítulo 6 se desarrolla una interfaz en Python con el fin de automatizar la implementación de los osciladores caóticos que utilizan el modelo HNN. El capítulo 7 hace uso de dicha interfaz para implementar una red neuronal con múltiples atractores, en el mismo capítulo se desarrolla el análisis requerido para el diseño de dicho sistema. Finalmente, se presentan las conclusiones del presente avance, una certificación de ponencia en el congreso EDIESCA 2024, la primer página de un artículo en revisión y un apartado de anexos donde se detallan los anexos con código utilizados para implementación de las redes neuronales Hopfield de 3 dimensiones.

Objetivos

Los objetivos planteados en este tema de tesis, objetivo general y objetivos particulares, se enuncian a continuación, junto a estos últimos un desglose más específico de dicho objetivo particular.

Objetivo general

Determinar los efectos de diferentes funciones de activación en el proceso de aprendizaje de una red neuronal Hopfield en modo caótico en un entorno de circuitos electrónicos.

Objetivos particulares

- Establecer la estructura básica de una red neuronal Hopfield de tres y cuatro neuronas.
 - Determinar los modelos matemáticos para una HNN de 3 y 4 neuronas.
 - Realizar la integración numérica de los modelos de 3 y 4 neuronas de una HNN.
- Caracterizar la dinámica no lineal de la red neuronal Hopfield usando técnicas analíticas y numéricas.
 - Realizar el análisis dinámico de los modelos matemáticos de las HNN para distintas funciones de activación.
 - Realizar el análisis desde un punto de vista caótico de los modelos de la red neuronal Hopfield.
- Evaluar la respuesta en el aprendizaje de la red neuronal Hopfield con diferentes funciones no lineales de activación.
 - Evaluar los efectos en el proceso de aprendizaje de distintas funciones de activación en los modelos de las redes neuronales Hopfield de 3 y 4 neuronas.

- Proponer las ecuaciones de diseño para implementar la red neuronal Hopfield en un sistema analógico embebido programable de prototipado rápido.
 - Simulación del comportamiento de la red neuronal Hopfield en el entorno de programación de AnadigmDesigner2.
 - Desarrollar una metodología para el diseño e implementación de una red neuronal Hopfield con 3 y 4 neuronas.
- Implementar los modelos desarrollados de la red neuronal Hopfield para 3 y 4 neuronas en la tarjeta analógica embebida.
 - Evaluar la dinámica obtenida.
 - Evaluar los efectos en el proceso de aprendizaje de los modelos implementados y comparar resultados con las simulaciones desarrolladas.

Cronograma de Actividades

En la Tabla 1 se presentan las actividades propuesta para el desarrollo esta tesis, titulada *Diseño de una red neuronal Hopfield en hardware embebido*. Se remarcan en color rosa los objetivos abordados en Tesis III. Algunas de las actividades aún están en el proceso de ser realizada, por lo que, la cuantificación de las actividades considera sólo un porcentaje de la actividad realizada y es presentado en la columna de “Avance realizado” junto con un porcentaje equivalente al valor en porcentaje de la actividad del total de la tesis.

Actividades	Tesis I	Tesis II	Tesis III	Avance realizado
Revisión bibliográfica y estado del arte.	X			100 % (10 %)
Planteamiento de los modelos matemáticos de las redes neuronales Hopfield de 3 y 4 neuronas.	X	X		100 % (5 %)
Simulación de las redes neuronales Hopfield e integración por métodos numéricos.	X	X		100 % (10 %)
Análisis de la dinámica de los modelos de las redes neuronales Hopfield de 3 y 4 neuronas mediante técnicas numéricas y analíticas.		X	X	100 % (15 %)
Evaluación del proceso de aprendizaje caótico de las HNN de 3 y 4 neuronas con distintas funciones de activación.		X	X	100 % (15 %)
Desarrollo de las ecuaciones para la implementación de las redes neuronales Hopfield en la tarjeta embebida analógica.		X	X	100 % (10 %)
Implementación de la red neuronal Hopfield con distintas funciones de activación.		X	X	70 % (10 %)
Desarrollo de una metodología para el diseño e implementación de la red neuronal Hopfield usando tarjetas embebidas analógicas de prototipado rápido.		X	X	50 % (10 %)
Publicación de un artículo.			X	50 % (5 %)
Análisis y comparación de resultados			X	70 % (5 %)
Escritura de la tesis.	X	X	X	80 % (5 %)
Porcentaje acumulado				87 %

Tabla 1: Cronograma de actividades.

Gráfica del avance global de la tesis

A continuación, en la Figura 4 se presenta el avance global que se tiene de este trabajo de tesis, el cálculo de este está basado en la ponderación de cada una de las actividades descritas en la Tabla 1 y del porcentaje realizado de cada una de ellas. Considerando lo anterior, la suma ponderada de las actividades descritas es equivalente al **87 %**.

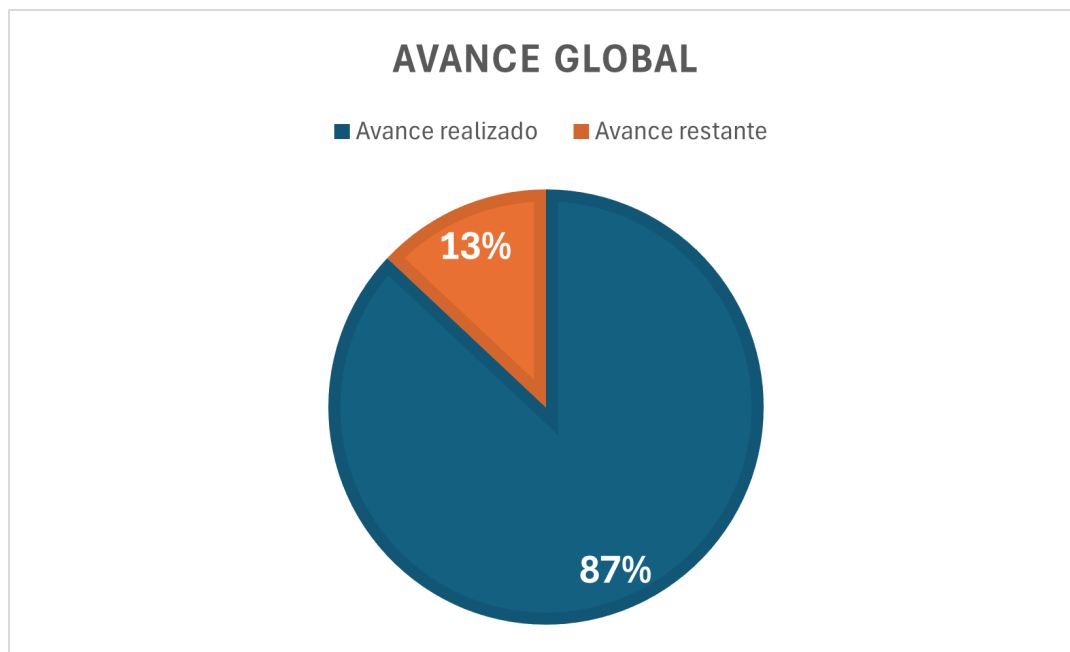


Figura 4: Gráfica con el avance global.

Diagrama de bloques de la tesis

A continuación se presenta, en la Figura 5, la propuesta general del proceso que se lleva a cabo durante el desarrollo de esta investigación. En esta se indica el periodo de inicio y desarrollo de cada actividad en un color diferente, considerando que marca el inicio y no necesariamente el final de cada actividad. El color verde indica el trabajo realizado el primer semestre, azul y morado representan los trabajos realizados el segundo y tercero.

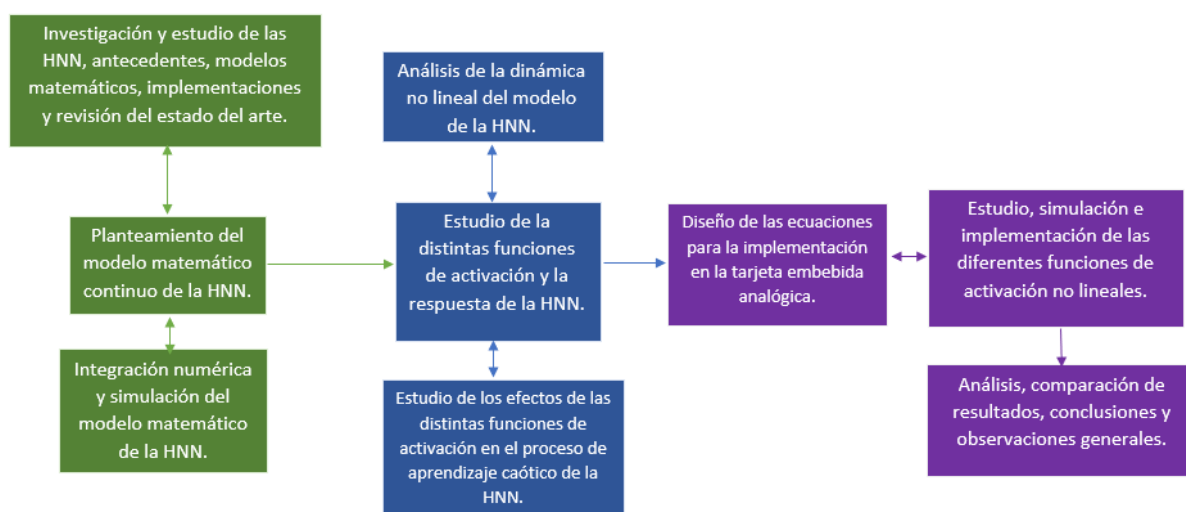


Figura 5: Resumen del trabajo por realizar en la tesis dividido por colores en diagrama de bloques.

Bloques para el primer semestre

Con lo que respecta al trabajo realizado con base en los bloques coloreados por color verde, correspondientes a las actividades desarrolladas y en desarrollo del primer semestre comprendido en agosto y diciembre del año 2023, se desglosaron de forma que sea posible describirlo más a fondo.

- **Investigación y estudio de las HNN, antecedentes, modelos matemáticos, implementaciones y revisión del estado del arte:** Esta actividad consiste en investigar y redactar los sucesos de mayor relevancia con respecto a los temas de interés de esta

tesis, es decir, de las redes neuronales artificiales, las HNN y su implementación. Posteriormente, se realiza el estado del arte de este trabajo, el cual consiste en sintetizar los avances más recientes desarrollados por la comunidad científica y hacer un pequeño repaso del impacto de estos.

- **Planteamiento del modelo matemático continuo de la HNN:** Junto con la investigación sobre las redes neuronales de Hopfield es posible hallar el modelo matemático con el cual se caracteriza en las referencias [11, 14–16], entre otras, por lo que, ayudado por esta investigación, es posible plantear un primer modelo para las HNN de 3 y 4 neuronas, en la sección 2.1.1 se presenta el modelo general en cuestión que expresa tanto el modelo de 3 como de 4 neuronas.
- **Integración numérica y simulación del modelo matemático de la HNN:** Un primer acercamiento al modelo matemático fue desarrollado al simular una de las ecuaciones que describen a la red neuronal de Hopfield, las ecuaciones y parámetros elegidos son los correspondientes al trabajo [15], cuya respuesta es caótica. La simulación fue implementada en el lenguaje de programación de Python usando el método de integración numérica de Runge-Kutta de cuarto orden.

Bloques para el segundo semestre

En el segundo semestre se trabajaron los bloques de color azul y morado, es decir, el trabajo no se realizó de manera lineal, en general se decidió trabajar con las redes neuronales Hopfield con tres neuronas, lo que incluyó su análisis e implementación con distintas funciones de activación. Un resumen del trabajo realizado con respecto a cada uno de los bloques propuestos es desglosado en los siguientes puntos.

- **Análisis de la dinámica no lineal del modelo de la HNN:** El análisis de la red neuronal Hopfield realizado se enfocó en demostrar que los modelos presentados son disipativos y por lo tanto, contienen atractores, también en la demostración del hecho que las redes neuronales presentadas están acotadas en alguna región del espacio fásico, finalmente, se desarrolla un análisis alrededor de los puntos de equilibrio (donde se encuentran los atractores), linealizando el sistema y caracterizando su comportamiento alrededor de ellos.
- **Estudio de las distintas funciones de activación y la respuesta de la HNN:** En la literatura existen una gran cantidad de funciones de activación para el modelo de las HNN, en este trabajo se exploró la aplicación de 5 distintas funciones de activación, cada una fue analizada desde un punto de vista de estabilidad, para finalmente ser implementadas en una tarjeta analógica programable.

- **Estudio de los efectos de distintas funciones de activación en el proceso de aprendizaje caótico de la HNN:** Algunas de las funciones estudiadas no tenían una propuesta para el modelo de HNN con comportamiento caótico, este el trabajo realizado a lo largo de este semestre de propusieron dos nuevos modelos así como una transformación que lleva modelos con función de activación $\tanh(\cdot)$ a modelos con funciones de activación $\sigma(\cdot)$. Para cada modelo se utilizaron dos métricas distintas para la detección de caos.
- **Diseño de las ecuaciones para la implementación en la tarjeta embebida analógica:** La programación en la tarjeta analógica es en base a un conjunto de bloques denominados CAM's, cada CAM tiene su equivalente con respuesta en frecuencia, tomando esto en consideración, se describió el modelo de redes neuronales Hopfield en su respuesta en frecuencia usando la transformada de Laplace, este resultado fue general y será utilizado para el desarrollo de las ecuaciones para implementación de las HNN de neuronas. Por su lado, los modelos para las HNN de 3 neuronas fueron implementados y coinciden con las simulaciones realizadas, este resultado es obtenido por inspección visual.
- **Estudio, simulación e implementación de las diferentes funciones de activación no lineales:** AnadigDesigner2 es el software que nos permite programar la tarjeta analógica y cuenta con una gran cantidad de bloques programables para la descripción de comportamientos lineales y no lineales, en base a uno o un conjunto de bloques es posible describir el comportamiento de las funciones de activación usadas, los resultados obtenidos en la implementación son mostrados y se realiza una comparación usando una métrica para el error.

Bloques para el tercer semestre

En el tercer semestre se trabajaron los bloques de color azul y morado, es decir, el trabajo no se realizó de manera lineal, en general se decidió trabajar con las redes neuronales Hopfield con 4 neuronas así como el diseño de un nuevo modelo de 3 neuronas con infinitos atractores, lo que incluyó su análisis e implementación de dichos modelos. También se realizó una interfaz como metodología para la implementación de las redes neuronales Hopfield. Un resumen del trabajo realizado con respecto a cada uno de los bloques propuestos es desglosado en los siguientes puntos.

- **Análisis de la dinámica no lineal del modelo de la HNN:** El análisis de la red neuronal Hopfield realizado se enfocó en utilizar las herramientas previamente utilizadas para los modelos de 3 neuronas y aplicarlo en el modelo hipercaótico, además, este análisis permitió el diseño de una red neuronal Hopfield con una infinita cantidad de atractores.

- **Diseño de las ecuaciones para la implementación en la tarjeta embebida analógica:** La implementación de la 4D-HNN hipercaótica requirió de un diseño en base a filtros pasa bajas del modelo. Posteriormente, se desarrollaron distintas topologías de circuitos en base a integradores para implementar las redes neuronales de 3 y 4 neuronas, estos circuitos son después utilizados en la interfaz que permite una simple implementación de las redes neuronales Hopfield en un entorno de circuitos electrónicos.
- **Estudio, simulación e implementación de las diferentes funciones de activación no lineales:** Se implementaron dos modelos de red neuronal, un modelo hipercaótico y un modelo de infinitos atractores. Por su parte, el modelo hipercaótico utiliza una función de activación previamente estudiada, sin embargo, la red neuronal con infinitos atractores utilizó una nueva función de activación no lineal.

Capítulo 1

Estado del arte

Como parte del estado del arte se realizo una revisión e investigación sobre los temas de mayor interés en este trabajo, esto es, redes neuronales Hopfield (HNN), sistemas dinámicos caóticos y hardware analógico embebido.

Revisión bibliográfica

En lo referente a las redes neuronales se realizo una investigación sobre los antecedentes, modelos matemáticos que a ellas describen e implementaciones sobre las redes neuronales artificiales, con un enfoque en las redes neuronales Hopfield, cuya implementación de 3 y 4 neuronas por medio de circuitos analógicos y digitales fue estudiada.

Por otro lado, y al ser un tema recurrente en la tesis, se realizo un estudio de los sistemas dinámicos, esto debido a que el modelo de la HNN es un sistema dinámico como es evidente por su modelo matemático [14], es importante considerar que, el ser un sistema dinámico no lineal, dados ciertos parámetros una HNN puede presentar un comportamiento caótico [15,17,18], con lo que también se realiza un estudio de los sistemas dinámicos caóticos, esto es, la investigación para definir lo que es un sistema dinámico caótico junto con sus características.

Actualidad de las HNN

Desde su creación por el Dr. Jonh J. Hopfield [11], las redes neuronales Hopfield han atraído a la comunidad científica por sus cimientos matemáticos, la aplicación de esta red neuronal está relacionada a la memoria asociativa y al reconocimiento de patrones [13]. Sin embargo, y como es demostrado por distintos trabajos de investigación, su aplicación va mas allá de ello y puede resolver problemas que usualmente son resueltos por otros modelos de redes neuronales. Además, debido a las características de su modelo matemático, su comportamiento puede ser caótico y con esto, conlleva todas las aplicaciones que los sistemas dinámicos caóticos tienen, como lo son los generadores de aleatoriedad, la criptografía, modelado de sistemas, entre otros.

Implementación de las HNN

La implementación de las redes neuronales Hopfield no es ajeno a los circuitos eléctricos, en la actualidad la implementación de un circuito no lineal es facilitada gracias a los dispositivos programables como lo son los FPGA's, múltiples trabajos han tomado esta aproximación para la implementación de una red neuronal Hopfield [15, 17, 19]. Por otro lado, también se ha desarrollado una electrónica analógica para la implementación de una red neuronal Hopfield [20–23], dicha electrónica esta basada en dispositivos activos y resistivos, la estructura presentada en estos trabajos es muy similar, sin embargo, en 2017 Bao et al. [19] desarrollo un modelo para una red neuronal Hopfield de 3 neuronas en base a memristores y partir de este trabajo múltiples desarrollos han sido llevados a cabo [18]. La implementación en un dispositivo digital implica discretizar el modelo matemático que se tiene del sistema continuo, por lo que el comportamiento obtenido será una aproximación a modelo que describe la red neuronal, mientras que la implementación en dispositivos analógicos, aunque limitada en otros aspectos, nos permite obtener el comportamiento exacto de nuestro modelo matemático.

Para gran parte de las implementaciones de las redes neuronales Hopfield, las metodologías para el diseño del circuito electrónico descritas con anterioridad son utilizadas, sin embargo, existen dispositivos analógicos programables denominados FPAA, con los cuales es posible desarrollar una metodología para la implementación de las HNN [24, 25], estos nos permiten tener la flexibilidad que nos ofrece un dispositivo programable sin la necesidad de discretizar el modelo de las HNN.

HNN en modo caótico

Una de las grandes razones por la cual las redes neuronales Hopfield resultan de interés en la comunidad científica es su riqueza dinámica a pesar de su simple estructura, su no linealidad nos permite modelar sistemas muy complejos, recientes trabajos relacionados a la construcción de una red neuronal Hopfield con base en Memristores han mostrado que estos pueden presentar caos [19, 20]. A partir de estos trabajos se han desarrollado otras estructuras y parámetros sobre las HNN en las cuales se han observado, no solo comportamientos caóticos, sino hipercaóticos [26], esto último considerando una red neuronal de cuatro neuronas.

Almacenamiento y reconocimiento de patrones

Las HNN son especialmente útiles en tareas de almacenamiento y recuperación de patrones asociativos. Pueden recordar y recuperar patrones incluso cuando la entrada está incompleta o contaminada con ruido [27]. La dinámica de las HNN se rige por una función de energía y el aprendizaje de patrones se realiza ajustando los pesos para minimizar esta energía. Sin

embargo, la estabilidad dinámica puede ser un desafío, especialmente cuando se trata de grandes conjuntos de datos o patrones complejos.

A pesar de estas fortalezas, las HNN también tienen limitaciones. No son ideales para manejar patrones complejos o grandes conjuntos de datos como lo hacen otras arquitecturas más modernas, como las redes neuronales convolucionales (CNN, por sus siglas en inglés) o las redes neuronales recurrentes (RNN, por sus siglas en inglés). Además, el modelo original de Hopfield puede tener problemas con la capacidad de almacenamiento y la recuperación de patrones en grandes dimensiones.

Modelado biológico

Las HNN han sido utilizadas para modelar la memoria asociativa, un aspecto de la memoria biológica, esta es capaz de reconocer patrones enteros usando solo partes parciales o dañadas por el ruido de ese mismo patrón [27]. La capacidad de las HNN para almacenar y recuperar patrones asociativos puede interpretarse como un intento de emular procesos de asociación en el cerebro [26], es importante tener en cuenta que estas redes son una simplificación de los sistemas biológicos. Aunque se inspiran en algunos aspectos de la neurobiología, no capturan la complejidad y la dinámica real de las redes neuronales biológicas.

Funciones de activación en las HNN

El propósito de las funciones de activación son las de determinar cuando una neurona se “dispara” o “enciende”, además, su no linealidad permite modelar comportamientos mas complejos. En la literatura se puede hallar una lista de diferentes funciones de activación [28], cada una ofrece distintos comportamientos y respuestas de la red neuronal, como es el caso de la identificación de imágenes [13]. Resulta de interés en esta tesis ser capaz de diferenciar entre qué funciones de activación son apropiadas para alguna aplicación en específico, así como la respuesta que cada una de estas tiene una vez implementada en un dispositivo programable.

Capítulo 2

Marco Teórico

Para el desarrollo de esta tesis es primero definir algunos conceptos de relevancia que se tocan en este trabajo, como lo son los sistemas dinámicos y caóticos, redes neuronales artificiales (Hopfield específicamente), hardware programable, entre otros. Con este fin se desarrollo el presente marco teórico.

2.1. Redes Neuronales Artificiales

Unas de las tecnologías de más renombres desarrolladas en los últimos años por la comunidad científica son las redes neuronales artificiales (ANN, por sus siglas en inglés). Estas toman el comportamiento de las redes neuronales biológicas observadas en el cerebro humano como base para el desarrollo de un modelo matemático que emule dicho comportamiento. El primer modelo de las redes neuronales biológicas fue el desarrollado por Warren McCulloch y Walter Pitts [7], en el cual las neuronas se describen como interruptores, cuyo estado es dependiente del de las demás neuronas. A la salida de cada neurona se denominado como axón y a su entrada como “dendrita”, a este sistema de conexiones entre neuronas se le conoce como sinapsis. Como base de esta descripción, el psicólogo Frank Rosenblatt desarrolló el primer modelo de las ANN conocido como perceptrón, en este, cada conexión entre las neuronas, es decir, la sinapsis, tiene un valor específico denominado como peso, consideraba neuronas de entrada y una sola neurona de salida, el modelo es presentado en la Figura 2.1, en la cual se observa tanto la representación matemática como en un diagrama. Tras el trabajo de Rosenblatt, más modelos de ANN fueron desarrollados, cada uno con distintas propiedades, aún sin embargo, los conceptos cimentados por este modelos se conservan y siguen siendo usados en modelos nuevos de las ANN. En modelos más recientes de ANN se requiere de una mayor cantidad de neuronas para la resolución de la mayoría de problemas. De acuerdo a la estructura de una red neuronal, estas se pueden separar en dos partes fundamentales, las redes neuronales, si el

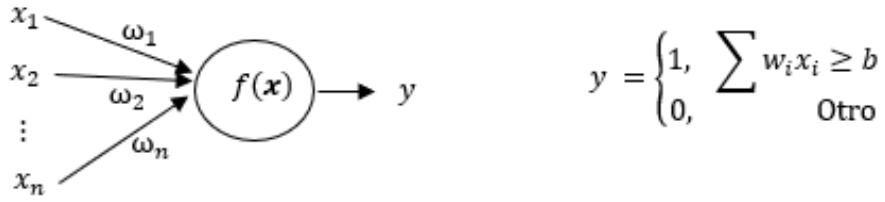


Figura 2.1: Modelo del perceptron.

modelo que describe la estructura de una red neuronal no contiene ciclos o retroalimentaciones, es denominada como red neuronal prealimentada o “feed-forward network”, de lo contrario, se le denomina como red recurrente o *recurrent network* [28].

2.1.1.1. Redes Neuronales Hopfield

Un tipo especial de red neuronal recurrente es la red neuronal Hopfield, desarrolladas por John Hopfield en 1980 [11], estas redes destacan por su capacidad única para el almacenamiento y la recuperación de patrones asociativos, emulando de alguna manera el funcionamiento de la memoria humana [27]. Esta estructura permite a la red aprender patrones y recuperar información basándose en la activación y desactivación de sus neuronas. Cada conexión entre neuronas, ponderada por un valor sináptico, contribuye a la capacidad de la red para almacenar y recordar patrones. Por otro lado, las HNN son ideales para modelar la dinámica del cerebro [11], además de su dinámica no lineal. En general una red neuronal Hopfield de 4 neuronas puede ser representada por el diagrama de la Figura 2.2. Debido a su estructura las HNN son generalmente conocidas como sistemas dinámicos retroalimentados el cual produce múltiples patrones [27], dependiendo de la condición inicial de la HNN esta puede converger a un punto específico, estos puntos suelen ser denominados como “atractores”. El diagrama mostrado por la Figura 2.2 puede ser generalizado para n neuronas considerando la misma forma en la sinapsis. Para una HNN de n neuronas, el modelo matemático continuo es el expresado por el sistema de ecuaciones no diferenciales adimensional dado por [29]:

$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{f}(\mathbf{x}) + \mathbf{b}, \quad (2.1)$$

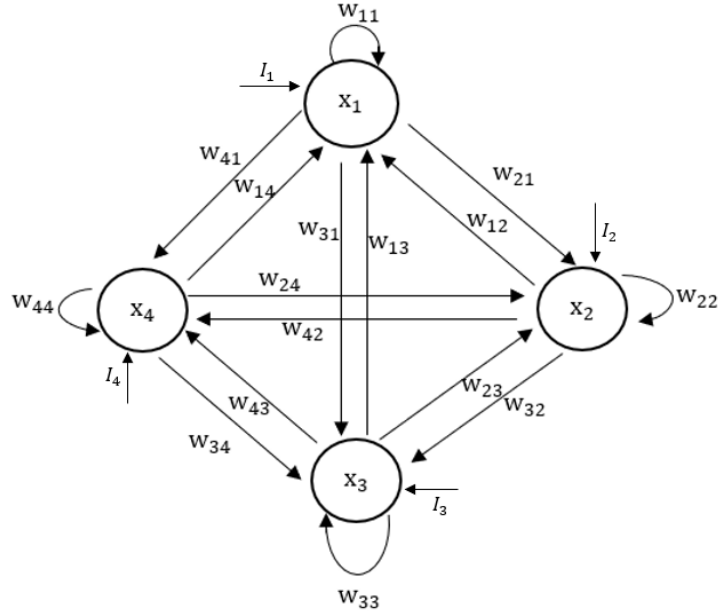


Figura 2.2: Diagrama de una red neuronal Hopfield de 4 neuronas.

donde

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}, W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ij} & \dots & w_{in} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nj} & \dots & w_{nn} \end{bmatrix}, C = \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix}. \quad (2.2)$$

x_i denota la i -ésima neurona, b_i representa un bias o un valor externo sobre la red neuronal, W es la matriz de pesos que representa la sinapsis del modelo y finalmente $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$ representa la función de activación no lineal aplicada en cada una de las variables del modelo y C una matriz diagonal. En específico, esta tesis está interesada en estudiar los modelos de 3 y 4 neuronas de una red neuronal Hopfield.

2.1.2. Funciones de activación

En cada modelo estudiado se han mencionado las denominadas funciones de activación, esta cumplen con distintos propósitos en los que se incluye, principalmente, la introducción de la no linealidad a la red neuronal permitiendo modelar sistemas más complejas, capacidad de aprendizaje, ajuste a la salida de la red neuronal, la continuidad que permite la interpretación

del estado de la neurona en todo momento, entre otros. Algunos modelos sin embargo, no utilizan exclusivamente funciones de activación sino otro tipo de funciones, dependiendo del modelo utilizado cada neurona utiliza una de tres tipos distintos de funciones [28]:

- Función de entrada.
- Función de activación.
- Función de salida.

En lo que respecta a las redes neuronales Hopfield, estas son redes especiales cuyas neuronas no utilizan funciones de entrada ni de salida, solo de activación, la entrada de esta red son los valores con los que se inicia cada neurona, es decir, las condiciones iniciales introducidas el sistema dinamico que a estas describe. Por esta razón este trabajo se limita al estudio de las funciones de activación. Las principales funciones de activación [30] usadas por cualquier red neuronal se presentan la Tabla 2.1.

Tabla 2.1: Funciones de activación de las ANN más frecuentes en la literatura.

Función de activación	$\phi(x)$	$\phi'(x)$	Rango
Tangente hiperbólica	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - \phi^2(x)$	$(-1, 1)$
Sigmoid	$S(x) = \frac{1}{1 + e^{-x}}$	$\phi(x)(1 - \phi(x))$	$(0, 1)$
ReLU	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$[0, \infty)$
Función Heaviside	$R(x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases}$	$\delta(x)$	$[0, 1]$
Función signo	$R(x) = \begin{cases} -1 & \text{para } x < 0 \\ 0 & \text{para } x = 0 \\ 1 & \text{para } x > 0 \end{cases}$	$2\delta(x)$	$[-1, 1]$
Softmax	$y_i = \frac{e^{x_j}}{\sum_j^n e^{x_j}}$	$\frac{\partial y_i}{\partial j} = y_i(\delta_{ij} - y_i)$	$(0, 1)$

La función de activación utilizada, al estar directamente implicada en el modelo matemático de la ANN, impacta directamente en la dinámica de la red neuronal.

Una de las propiedades más atractivas de las redes neuronales (artificiales) es la posibilidad de entrenarlas para ciertas tareas con la ayuda de datos de ejemplo. En general entrenar una red neuronal consiste en adaptar los pesos de conexión y posiblemente otros parámetros (como bias) de manera que se optimice cierto criterio. Dependiendo del tipo de datos de entrenamiento y del criterio a optimizar, podemos distinguir dos tareas fundamentales de aprendizaje [28]:

aprendizaje supervisado, en el cual se asigna una salida deseada a un conjunto de entrada, y aprendizaje no supervisado en el cual no se proporcionan salidas deseadas.

En lo que se refiere a una red neuronal Hopfield, su entrenamiento se refiere a diseñar un atractor o atractores para un conjunto específico de condiciones iniciales [27]. Al conjunto de estos atractores se les conoce como memoria, mientras que al conjunto de condiciones iniciales que son atraídas por un atractor se les conoce como “cuenca de atracción”, a los valores que toman las neuronas una vez que estas convergieron a su valor final se le denomina como “patrón”. Considerando lo anterior, una red neuronal Hopfield puede ser visto como un dispositivo capaz de almacenar y recuperar información relacionada entre si. Con el paso del tiempo distintos investigadores han desarrollado algoritmo para el entrenamiento de HNN y cuyo estudio sera realizado a lo largo del desarrollo de esta tesis.

2.2. Sistemas dinámicos

La gran parte de los sistemas estudiados por la física y la matemática tienen un comportamiento variante en el tiempo, es decir, no son estáticos, a este tipo de sistemas se les denomina como sistemas dinámicos y su estudio se remonta a la invención del calculo diferencial desarrollado por Sir Isaac Newton (1643-1727) y por Gottfried Wilhelm Freiherr von Leibniz (1646-1716). Newton, tras la invención del calculo, intentó describir el comportamiento de los planetas alrededor del Sol usando ecuaciones diferenciales [31]. Newton fue capaz de resolver el problemas de los dos cuerpos, en donde describió el movimiento de la tierra alrededor del sol, tras esta hazaña, el estudio de los sistemas dinámicos se ha aplicado a diversas disciplinas dentro y fuera de la física, permitiendo tener un sistema de modelos capaces de describir el comportamiento de un sistema a lo largo de un periodo de tiempo.

2.2.1. Representación matemática de los sistemas dinámicos

Debido a su naturaleza, los sistemas dinámicos son descritos por mapas iterativos (también conocidas como ecuaciones de diferencias) o ecuaciones diferenciales. En este trabajo estamos interesados solo en su descripción por ecuaciones diferenciales. Consideremos un sistema de n ecuaciones diferenciales, este se expresa como sigue:

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ \dot{x}_2 &= f_2(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p), \\ &\vdots \\ \dot{x}_n &= f_n(t, x_1, x_2, \dots, x_n, u_1, \dots, u_p),\end{aligned}\tag{2.3}$$

donde t es la variable que representa el tiempo, $u_1, u_2, \dots, u_p$ representan la entrada al sistema, $x_1, x_2, \dots, x_n$ las variables del sistema y $\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n$ de derivada con respecto al tiempo de las variables, al sistema anterior se le denomina de n -ésimo orden o de n dimensiones. Consideremos ahora que se conoce el estado inicial de las variables en el sistema de ecuaciones 2.3, esto es, se tiene un vector $\mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^n$ que representa el estado del sistema para un tiempo inicial t_0 , a la solución de este tipo de problemas se les dice *problemas de valor inicial* [32], mientras que al estado inicial del sistema se le dice *condición inicial*. Por otro lado, de forma reducida la expresión 2.3 puede ser escrita usando vectores, de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (2.4)$$

donde $\mathbf{f}(t, \mathbf{x}, \mathbf{u}) = [f_1(t, \mathbf{x}, \mathbf{u}), \dots, f_n(t, \mathbf{x}, \mathbf{u})]^T$, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $\mathbf{u} = [u_1, u_2, \dots, u_p]^T$ y $\mathbf{x}(t_0) = [x_1(t_0), x_2(t_0), \dots, x_n(t_0)]^T$. En las expresiones (2.3) y (2.4), el tiempo t está expresado explícitamente, en algunos sistemas dinámicos este no es el caso y no dependen explícitamente ni del tiempo ni de la entrada, a este tipo de sistemas se les denomina como sistemas dinámicos autónomos y están expresados por

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (2.5)$$

su estudio es de gran relevancia en esta tesis al representar el modelo matemático de las HNN (véase ecuación (2.1)), para bias constantes. A su vez los sistemas dinámicos pueden ser de dos tipos: lineales o no lineales. El primero indica que las funciones contenidas por el vector $\mathbf{f}(\mathbf{x})$ cumplen con la propiedad de superposición (descrita en la sección 2.1.2), mientras que los sistemas no lineales no cumplen con dicha propiedad. Por inspección visual es posible identificar estos sistemas al estar compuestos por operaciones como multiplicaciones entre variables de estado, funciones trigonométricas, potencias, entre otros.

2.2.2. Puntos de equilibrio

La solución de la ecuación diferencial 2.4, suponiendo que exista una, puede ser conocida y expresada por un conjunto de funciones elementales o desconocidad, es decir, no es posible expresarla por algún conjunto de funciones elementales. La definición de lo que es una función elemental esta fuera del alcance de esta tesis, considerese simplemente que las funciones estudiadas a lo largo de este trabajo son elementales. La solución para un sistema de ecuaciones diferenciales puede no ser única, es más, lo más común es que exista una familia de soluciones, incluso una solución son dependencia del tiempo, esto es, una solución al sistema de ecuaciones diferenciales constante. Una solución de este tipo es conocida como *punto de equilibrio* o *punto fijo*. Consideremos entonces la solución constante de un sistema dinámico autónomo como

$\mathbf{x}(t) = \eta \in \mathbb{R}^n$, entonces su derivada con respecto al tiempo da

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) = \frac{d\eta}{dt} \Rightarrow \mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad (2.6)$$

siendo $\mathbf{0} \in \mathbb{R}^n$ un vector nulo de n dimensiones. En otras palabras para hallar el punto de equilibrio de un sistema dinámico es la solución del sistema de ecuaciones descrito en la expresión 2.6. Notemos que la solución constante, en caso de que esta existe, debe ser forzosamente la condición del sistema dinámico, esto es, $\eta = \mathbf{x}_0$.

2.2.3. Sistemas dinámicos caóticos

La hazaña de Newton resolviendo el problema de los dos cuerpos inspiró a una gran cantidad de físicos y matemáticos a extender el problema de los dos cuerpos a tres cuerpos, sin embargo, este resultó en una problemática muchísima más compleja. Tras varias décadas de investigación se llegó a la conclusión que el problema de los tres cuerpos era esencialmente imposible de resolver [33]. En 1800, el matemático Henri Poincaré (1854-1912) decidió volver a analizar este sistema, en esta ocasión su análisis tuvo un enfoque más cualitativo que cuantitativo. Posteriormente en 1961, el meteorólogo y matemático Edward Lorenz desarrolló una máquina capaz de predecir el comportamiento del clima, poco tiempo después, se dio cuenta que con mínimas variaciones en las condiciones iniciales, su sistema mostraba dos comportamientos completamente distintos, sus observaciones y resultados fueron publicadas en 1963 [34]. Su trabajo desarrolló lo que ahora se denomina como Teoría del Caos.

Un sistema dinámico caótico o simplemente sistema caótico, es un tipo especial de sistema dinámico que cumple con la característica de ser extremadamente sensible a las condiciones iniciales. Si bien, al día de hoy no existe un consenso general para dar una definición clara de los sistemas caóticos, si se tiene un consenso de tipo de características que estos sistemas deben cumplir [35]:

- **Determinista y no lineal.** Cuando un sistema dinámico está completamente determinado por sus condiciones iniciales se le denomina como determinista, además, cuando se adjudica el término de no lineal a un sistema, implica que este no cumple con la propiedad de la superposición para todos sus estados.
- **Extrema sensibilidad en sus condiciones iniciales.** Si bien, si un sistema caótico es determinista, su dependencia con respecto a sus condiciones iniciales es muy inestable, esto es, diminutas variaciones en las condiciones inicial resultan en cambios drásticos en el comportamiento del sistema.
- **Densidad en sus orbitas periódicas.** Aunque irregular, el comportamiento de un

sistema caótico es periódico, esto es, su órbita gira alrededor de un conjunto de puntos denominado como atractores.

Es importante mencionar que conocido el sistemas de ecuaciones diferenciales que rigen el sistema caótico es posible determinar cómo evoluciona el sistema. Sin embargo, su extrema sensibilidad genera que, tras un cierto periodo de tiempo, no sea posible predecir su comportamiento con absoluta certeza.

2.2.4. Redes Neuronales Hopfield en modo caótico

El modelo matemático de las redes neuronales de Hopfield se comporta como los sistemas caóticos con ciertos parámetros específicos [15, 18], en algunos de estos trabajos se ha llevado a cabo una implementación sobre algún tipo de dispositivo programable, en cuyos casos la respuesta de la HNN ha conservado sus propiedades caóticas.

La identificación de un comportamiento caótico en el modelo de la red neuronal Hopfield ha requerido de herramientas utilizadas en teoría de control, como lo son los diagramas de bifurcación, exponentes de Lyapunov o el cálculo y análisis de los puntos de equilibrio del modelo matemático. Más específicamente, para una red neuronal Hopfield de 3 neuronas, la presencia de un comportamiento caótico fue hallado usando el modelo descrito por [17]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \end{bmatrix}, \quad (2.7)$$

dada la matriz de pesos

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} 0 & 1.2 & -7 \\ 1.1 & 0 & 2.8 \\ 0.7 & -2 & 4 \end{bmatrix}.$$

Un diagrama de fase es una herramienta gráfica utilizada en el análisis de sistemas dinámicos para visualizar el comportamiento a lo largo del tiempo de un conjunto de variables relacionadas. En este tipo de diagrama se gráfica una variable del sistema contra otra distintas, el diagrama de fase proporciona una representación visual de cómo las variables evolucionan y se relacionan entre sí a medida que el sistema dinámico evoluciona.

Los diagramas de fase obtenidos del sistema (2.7) se presentan en la Figura 2.3.

Cabe mencionar que para obtener el diagrama de fase anterior fue requerida la simulación del sistema y, con esta, la solución numérica del sistema caótico, el método utilizado es presentado en la sección 2.4.

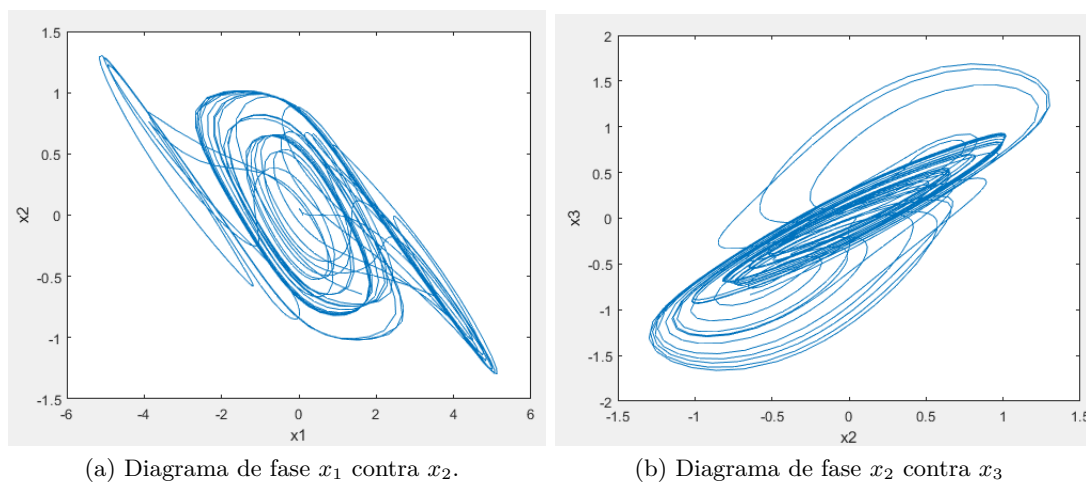


Figura 2.3: Diagramas de fase 2D formados por el sistema 2.7.

2.3. Dispositivos programables

En muchas ocasiones cuando se desarrolla un circuito electrónico es necesario ajustar o incluso rediseñar partes del circuito. Para el caso específico de este trabajo, este rediseño o ajuste resulta ser incluso más complejo en términos de estabilidad y el desarrollo del circuito requerido. Por esto, el uso de dispositivos programables es de gran utilidad en la electrónica, en lo que respecta a dispositivos digitales se utilizan los llamados “Arreglo de Puertas Programables en Campo” (FPGA, por sus siglas en inglés), su contraparte analógica es menos conocida, sin embargo, nos ofrece la misma ventaja, estos son denominado como “Arreglo de Campo de Analógico Programable” (FPAA, por sus siglas en inglés).

2.3.1. Hardware Analógico Embebido

Los FPAA’s son dispositivos electrónicos programables que permiten la creación de circuitos analógicos personalizados y, como tal, nos permiten hacer operaciones como lo son la suma, resta, comparación, integración, derivación, entre otros. La arquitectura general de un FPAA es presentada en la Figura 2.5, en esta se observa que está constituido por bloques analógicos configurables (CAB’s, por sus siglas en inglés), los cuales tienen una función de transferencia equivalente, estos bloques están interconectados por interruptores controlados. En lo que respecta a este trabajo y su implementación, el FPAA utilizado es el modelo AN231E04 desarrollado por Anadigm, la misma distribuidora ofrece un kit que contiene una tarjeta con 4 FPAA’s. Por otro lado, el desarrollador y distribuidor ofrece el software para programar la tarjeta con los AN231E04 llamado AnadigmDesigner2, el cual proporciona un entorno de diseño gráfico con el cual podemos diseñar y simular circuitos analógicos. El ambiente de trabajo de

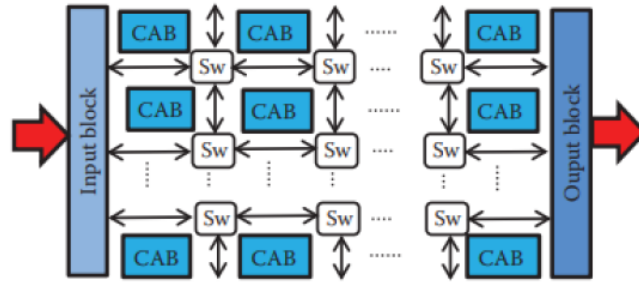


Figura 2.4: Arquitectura general de un FPAA [36].

AnadigmDesigner2 (Figura 2.5) consiste en bloques, los cuales representan el FPAA sobre el cual se programará y dentro de este se pueden implementar Módulos Analógicos Configurables (CAM's, por sus siglas en inglés); cada uno de estos tiene una función de transferencia equivalente que se encuentra en la documentación dada por el desarrollador.

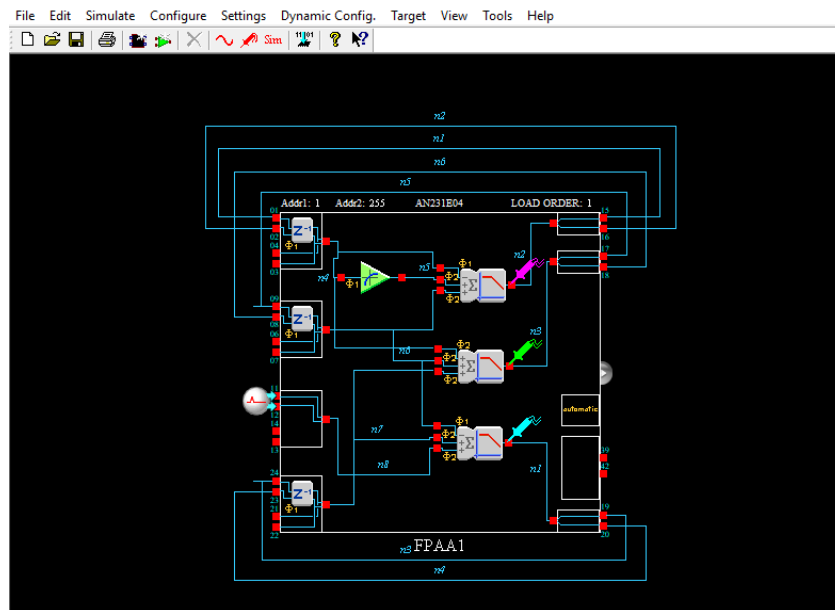


Figura 2.5: Entorno de programación en bloques de AnadigmDesigner2.

2.4. Métodos numéricos para la solución de sistemas dinámicos

En muchos problemas de la física y las matemáticas se estudian modelos descritos por ecuaciones diferenciales lineales y no lineales, es por esto que una solución para dichos sistemas es de suma relevancia. Sin embargo, no siempre es posible el desarrollo de una solución analítica del modelo matemático y, en estos casos, es necesario utilizar métodos numéricos que aproximen

a la solución deseada.

Existen una gran cantidad de métodos numéricos para dar solución a un sistema de ecuaciones diferenciales; con lo que respecta a esta tesis, se utilizarán los métodos desarrollados por los matemáticos alemanes Carl David Tolmé Runge y Martin Wilhelm Kutta, a fines del siglo XIX y principios del siglo XX [37]. En 1895, Tolmé Runge publicó un artículo titulado “Über die numerische Auflösung von Differentialgleichungen”(Sobre la resolución numérica de ecuaciones diferenciales), donde propuso un método iterativo para la solución de ecuaciones diferenciales. Posteriormente, en 1901, Martin Wilhelm Kutta publicó un trabajo titulado “Beiträge zur näherungsweise Integration totaler Differentialgleichungen”(Contribuciones a la integración aproximada de ecuaciones diferenciales totales), donde desarrolló y popularizó un método más amplio y generalizado, que incluía el método de Runge y otros métodos similares. Este método se conoció como el método de Runge-Kutta, combinando los apellidos de ambos matemáticos [37].

Dentro del grupo de métodos Runge-Kutta, en este trabajo se tomarán dos metodologías, el método de Runge-Kutta de segundo orden, también llamado método de Heun, y el método de Runge-Kutta de cuarto orden. El desarrollo para la obtención del método de Heun es el siguiente: consideremos la ecuación diferencial ordinaria (EDO) de la forma

$$\frac{dy(t)}{dt} = f(t, y). \quad (2.8)$$

Aplicamos la expansión de Taylor a la función $y(t)$ alrededor de t_n :

$$y(t) = y(t_n) + (t - t_n) \frac{dy}{dt} \Big|_{t_n} + \frac{(t - t_n)^2}{2} \frac{d^2y}{dt^2} \Big|_{t_n} + O((t - t_n)^3), \quad (2.9)$$

donde $O((t - t_n)^3)$ representa términos de orden superior, que estamos ignorando para obtener una aproximación de segundo orden.

Integrando ambos lados de la ecuación de la EDO de t_n a $t_{n+1} = t_n + h$ (donde h es el tamaño del paso) y aplicando la regla del trapecio para la integración, obtenemos:

$$y_{n+1} = y_n + \frac{h}{2} (f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})), \quad (2.10)$$

donde $\tilde{y}_{n+1}$ es una estimación de y_{n+1} que obtenemos usando la aproximación de Euler:

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n). \quad (2.11)$$

El método de Runge-Kutta de segundo orden ofrece una mejor aproximación a la solución de una ecuación diferencial, a diferencia del muy conocido método de Euler. Sin embargo, en gran parte de los trabajos de investigación se ha popularizado el método de Runge-Kutta de

cuarto orden debido a su facilidad de programación y gran precisión en la solución de ecuaciones diferenciales. Consideremos nuevamente la expresión, (2.8)

$$\frac{dy(t)}{dt} = f(t, y).$$

Para la aplicación del método de Runge-Kutta de 4 orden se divide el intervalo $[t_n, t_{n+1}]$ en cuatro pasos y se calculan las pendientes intermedias k_1 , k_2 , k_3 , y k_4 :

$$\begin{aligned} k_1 &= h \cdot f(t_n, y_n), \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= h \cdot f(t_n + h, y_n + k_3). \end{aligned} \tag{2.12}$$

Se combina ponderadamente las cuatro pendientes para obtener la pendiente final y la solución se actualiza en el siguiente paso de tiempo t_{n+1} :

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{2.13}$$

La cantidad expresada por $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ es una estimación mejorada de la pendiente de la solución en el intervalo de tiempo $[t_i, t_i + h]$, k_1 es la pendiente en el extremo izquierdo del intervalo, k_2 es la pendiente utilizada en el punto medio del intervalo mientras que k_3 es la pendiente mejorada en este mismo punto, finalmente, k_4 es la pendiente en el punto extremo derecho.

Capítulo 3

Análisis de estabilidad de la red neuronal Hopfield

En la literatura existe una diversa cantidad de modelos propuestos para las HNN con presencia de caos usando diversas funciones de activación, cada una ofrece una dinámica distinta al modelo y una estructura distinta para su implementación en circuitos electrónicos. Por lo tanto, resulta de relevancia ofrecer distintos modelos para una HNN que mantenga una compleja dinámica con distintas funciones de activación. De esta manera en esta sección se presentarán 5 modelos para la HNN con distintas funciones de activación con las cuales el sistema presenta un comportamiento caótico. De estos, dos no han sido reportados y fueron hallados por el autor, los tres restantes fueron obtenidos de la literatura, sin embargo, uno de estos fue transformado para su utilización con un nuevo modelo con otra función de activación y por lo tanto es clasificado como un nuevo propuesto en este trabajo.

Cuando se analiza un sistema dinámico se busca realizar la caracterización de las propiedades que describen el considerado sistema. El análisis puede ser dividido en dos secciones: el análisis cuantitativo y el análisis cualitativo. El primero requiere de las características y propiedades que posee el sistema analizado, mientras que el segundo se centra en los resultados estrictamente numéricos obtenidos, por ejemplo, los obtenidos por alguna simulación del sistema. La principal propiedad que se analiza en sistema dinámicos es su estabilidad, para esta existen distintas metodologías, en este trabajo se realizara el estudio de la divergencia del flujo, estabilidad de Lyapunov. Finalmente, se analiza la estabilidad de los sistemas alrededor de los puntos de equilibrio, linealizando el sistema y caracterizando el comportamiento de cada uno de los puntos de equilibrio.

3.1. Propiedades de una función de activación

Desde que Hopfield propuso su modelo de redes neuronales artificiales en 1982, diferentes modificaciones han sido propuestas incorporando nuevos factores internos o externos al modelo, una estas modificaciones es el cambio de la función de activación del modelo de las HNN. En los primeros años de investigación las funciones de activación eran acotadas, crecientes y cero evaluadas en su origen, sin embargo, diversas funciones sin estas propiedades han sido estudiadas con resultados satisfactorios para un grupo específico de problemas, como resultado no existe la definición de un grupo de propiedades para todas las funciones de activación [38]. Para este trabajo sin embargo, las funciones de activación cumplen con las propiedades a continuación enlistadas.

- **Continua.** Sea $f : A \subset \mathbb{R} \rightarrow \mathbb{R}$ y $c \in A$. Se dice que f es continua en c si dado cualquier $\epsilon > 0$, existe un $\delta > 0$ tal que si x es un punto en A se cumple

$$|x - c| < \delta \quad \Rightarrow \quad |f(x) - f(c)| < \epsilon. \quad (3.1)$$

Es importante hacer notar que si una función es diferenciable implica que es continua en el mismo dominio, sin embargo, una función de activación como continua en todo su dominio, a diferencia de su propiedad como función diferenciable. La interpretación de esta propiedad dentro del modelo de las HNN esta relacionada a los estados que toman las neuronas, debido a que estas siempre deben estar en uno la continuidad nos asegura este hecho.

- **Creciente:** Una función $f : A \subset \mathbb{R} \rightarrow \mathbb{R}$ es creciente en A si para cualquier $x_1, x_2 \in A$ se cumple que

$$x_1 \leq x_2 \quad \Rightarrow \quad f(x_1) \leq f(x_2). \quad (3.2)$$

En el caso de que en las desigualdades anteriores se cumple que $x_1 < x_2$ y $f(x_1) < f(x_2)$ entonces se dice que la función es estrictamente creciente. Si una función es creciente implica que su derivada es siempre mayor o igual a cero ($\frac{d}{dx}f(x) \geq 0$), mientras que si es estrictamente creciente es siempre mayor a cero $\frac{d}{dx}f(x) > 0$. Cabe mencionar que algunas funciones de activación consideran el caso contrario, es decir, funciones decrecientes y estrictamente decrecientes, en estos casos este comportamiento no afecta al modelo, sin embargo, se debe hacer la interpretación adecuada de los resultados. Como utilidad de esta propiedad, recordemos que una neurona biológica esta en un estado de encendido o apagado, por lo tanto, si la función de activación fuera no monótona, presentaría comportamientos irregulares no deseados.

- **No lineal:** La no linealidad de un sistema ya fue definida en la sección 2.2 de este

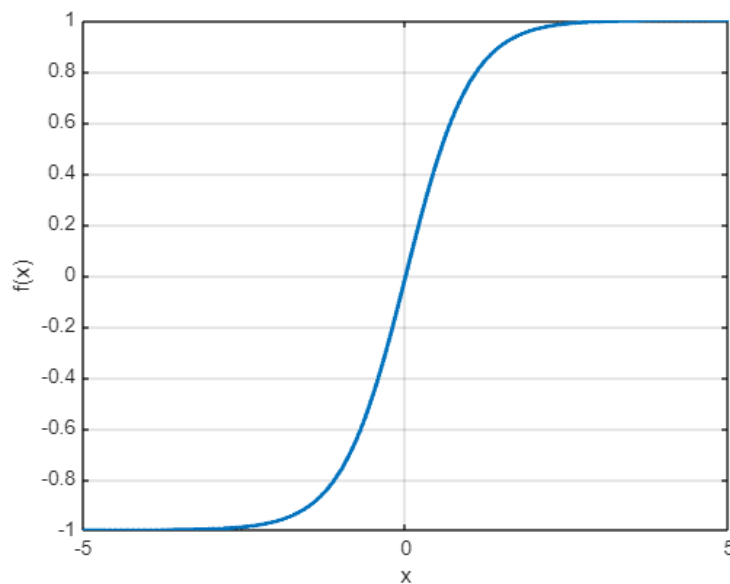


Figura 3.1: Comportamiento de la función $\tanh(x)$.

trabajo. En general, esta propiedad es relevancia ya que nos permite modelar sistemas mas complejos, a diferencia de las funciones lineales. Por otro lado y desde un punto de vista caótico, un sistema no puede presentar caos si esta descrito por funciones lineales.

3.2. Modelos propuestos

3.2.1. Modelo con función de activación tangente hiperbólica

La función tangente hiperbólica es la función de activación mas frecuente en los modelos de HNN, de comportamiento mostrado en la Figura 3.1 esta descrita por la expresión

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (3.3)$$

donde $\cosh x$ y $\sinh x$ son funciones hiperbólicas definidas por $\frac{e^x + e^{-x}}{2}$ y $\frac{e^x - e^{-x}}{2}$ respectivamente, y e^x es la función exponencial, la cual es estrictamente creciente, siempre mayor a 0, diferenciable en todo el dominio $\mathbb{R}$ y por lo tanto continua en el mismo. Además, esta función esta acotada como $|\tanh(x)| < 1$. La derivada de esta función viene dada por

$$\frac{d}{dx} \tanh(x) = 1 - \tanh^2 x \quad (3.4)$$

Como fue mencionado, distintos modelos para la HNN utilizan esta función de activación, para este trabajo se considerara **un modelo de 3 dimensiones desarrollado por el mismo**

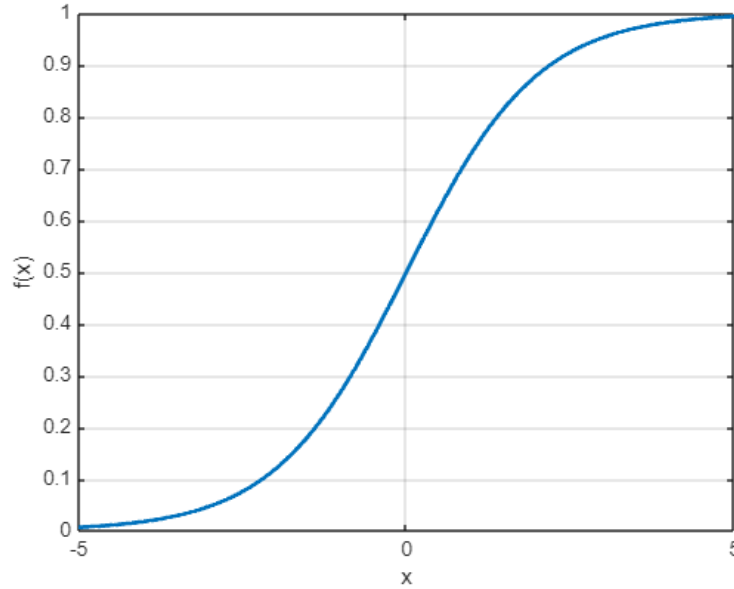


Figura 3.2: Comportamiento de la función $\sigma(x)$.

autor descrito de la siguiente manera

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & -2 & -9 \\ 1 & 3 & 3 \\ -1 & -3 & 0 \end{bmatrix} \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \\ \tanh x_3 \end{bmatrix}, \quad (3.5)$$

este modelo presenta algunas ventajas con respecto a otros modelos antes propuestos, el primero es su sencillez considerando que todos sus parámetros son enteros, y por otro lado, presenta cierta robustez con respecto a su comportamiento caótico, es decir, algunos de los pesos pueden ser variados por un rango amplio de valores y aun así el sistema mantendrá su comportamiento caótico. Esto ultimo es explorado en el análisis dinámico de las HNN.

3.2.2. Modelo obtenido con función de activación sigmoide

La función sigmoide, típicamente referida por la letra griega sigma σ y comportamiento mostrado en la Figura 3.2, es otra función de activación típicamente utilizada, esta es diferenciable en todo su dominio, es acotada por $0 < \sigma(x) < 1$ y esta definida por

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (3.6)$$

con derivada

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x)). \quad (3.7)$$

Siguiendo el siguiente desarrollo es posible relacionar esta función con la tangente hiperbólica

$$\tanh x = \frac{e^x e^x - e^{-x}}{e^x e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{2}{1 + e^{2x}} - 1 \quad (3.8)$$

$$\Rightarrow \tanh x = 2\sigma(2x) - 1, \quad (3.9)$$

entonces es posible usar modelos de HNN que utilizan la función de activación $\tanh(\cdot)$ y reescribirlos usando $\sigma(\cdot)$, para esto se realizó el siguiente desarrollo. Consideremos una HNN n -dimensional, descrito por

$$\dot{\mathbf{x}} = -C\mathbf{x} + W \tanh(\mathbf{x}) + \mathbf{b}, \quad (3.10)$$

donde $\tanh(\mathbf{x})$ es un vector de n elementos dados por $\tanh(x_i)$ para todo $i = (1, 2, \dots, n)$ y por lo tanto puede ser reescrito como

$$\tanh(\mathbf{x}) = \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \\ \vdots \\ \tanh x_n \end{bmatrix} = \begin{bmatrix} 2\sigma(2x_1) - 1 \\ 2\sigma(2x_2) - 1 \\ \vdots \\ 2\sigma(2x_n) - 1 \end{bmatrix}, \quad (3.11)$$

consideremos el vector $\Sigma(\mathbf{x})$ de n elementos descritos por $\sigma(x_i)$ para todo $i = (1, 2, \dots, n)$ y un cambio de variables $2x_i \rightarrow x_i^*$. En :

$$\begin{aligned} \dot{\mathbf{x}} = -C\mathbf{x} + W \tanh(\mathbf{x}) + \mathbf{b} &\Rightarrow \dot{\mathbf{x}} = -C\mathbf{x} + W \begin{bmatrix} 2\sigma(2x_1) - 1 \\ 2\sigma(2x_2) - 1 \\ \vdots \\ 2\sigma(2x_n) - 1 \end{bmatrix} + \mathbf{b}, \\ &\Rightarrow \dot{\mathbf{x}} = -C\mathbf{x} + 2W\Sigma(2\mathbf{x}) - WI_{n \times 1} + \mathbf{b}, \\ &\Rightarrow \dot{\mathbf{x}}^* = -C\mathbf{x}^* + 4W\Sigma(\mathbf{x}^*) - 2WI_{n \times 1} + 2\mathbf{b}. \end{aligned} \quad (3.12)$$

donde $I_{n \times 1} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^{n \times 1}$. En otras palabras, es posible crear a un nuevo modelo

$$-\dot{\mathbf{x}}^* = -C_\sigma \mathbf{x} + W_\sigma \Sigma(\mathbf{x}^*) + \mathbf{b}_\sigma, \quad (3.13)$$

donde

$$\begin{aligned} \Sigma(\mathbf{x}) &= \begin{bmatrix} \sigma(x_1) & \sigma(x_2) & \dots & \sigma(x_n) \end{bmatrix}^T, \\ C_\sigma &= C, \\ W_\sigma &= 4W, \\ \mathbf{b}_\sigma &= -2WI_{n \times 1} + 2\mathbf{b}, \end{aligned}$$

con $I_{n \times 1} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n$, en base a un modelo de HNN con función de activación $\tanh(\cdot)$ descrito por las matrices y vector W , C y $\mathbf{b}$ respectivamente. Consideremos el sistema propuesto en [1] para una HNN con la matriz de pesos

$$W = \begin{bmatrix} 1.99 & -1 & 0 \\ 2.5 & 1.7075 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix}, \quad (3.14)$$

para este sistema el vector B es un vector nulo, mientras que C es la matriz identidad de dimensión 3. Este sistema es transformado aplicando la expresión (3.12) de la siguiente manera

$$\begin{aligned} \dot{\mathbf{x}}^* &= -\mathbf{x}^* + 4 \begin{bmatrix} 2 & -1 & 0 \\ 2.5 & 1.71 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix} \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \sigma(x_3) \end{bmatrix} - 2 \begin{bmatrix} 2 & -1 & 0 \\ 2.5 & 1.71 & 1.25 \\ -4.75 & 0 & 0.92 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \\ \Rightarrow \quad \dot{\mathbf{x}}^* &= -\mathbf{x}^* + \begin{bmatrix} 7.97 & -4 & 0 \\ 10 & 6.84 & 5 \\ -19 & 0 & 3.67 \end{bmatrix} \begin{bmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \sigma(x_3) \end{bmatrix} + \begin{bmatrix} -2 \\ -10.92 \\ 7.66 \end{bmatrix}. \end{aligned} \quad (3.15)$$

El sistema anterior mantiene el comportamiento del sistema original permitiendo usar la función de activación $\sigma(\cdot)$.

3.2.3. Modelo con función de activación signo

La función signo, denotada por $\text{sign}(\cdot)$, indica el signo tiene el valor de entrada, si este es positivo la función regresa el valor 1, si es negativo la función regresa el valor de -1, para un valor de entrada la función regresa un 0 (véase Figura 3.3), de manera matemática es posible describir este comportamiento de la forma

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0, \end{cases} \quad (3.16)$$

es evidente que la derivada en el punto $x = 0$ no existe, para todos lo demás puntos esta es 0. Otro dato de interés de esta función es su relación con la función $\tanh(\cdot)$, esto debido a que es posible relacionar ambas funciones de la siguiente manera

$$\lim_{k \rightarrow \infty} \tanh(kx) \approx \text{sign}(x), \quad (3.17)$$

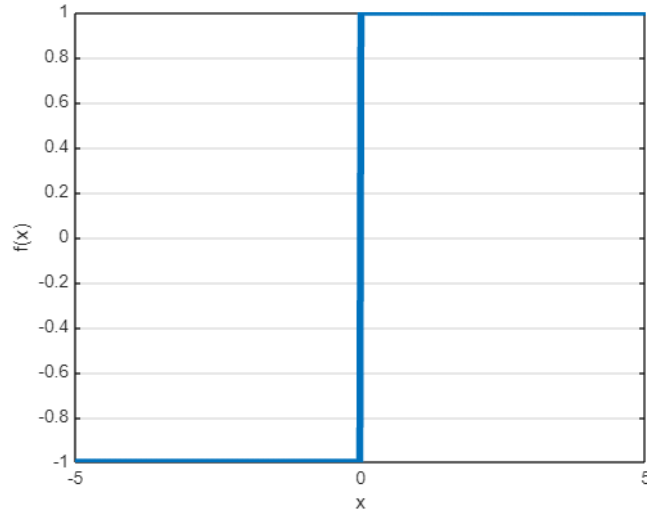


Figura 3.3: Comportamiento de la función $\text{sign}(x)$.

esta última relación podría ser utilizada, junto con un cambio de variables, para utilizar alguno de los modelos ya conocidos de la HNN con la función de activación $\tanh(\cdot)$ y obtener un comportamiento caótico, sin embargo, este acercamiento genera problemáticas en la implementación y simulación, es por esto que **el autor propuso un nuevo modelo** en donde la función $\text{sign}(\cdot)$ es sustituida en solo una de las funciones de activación del modelo, esta viene dada por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.65 & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{sign}(x_1) \\ \text{sign}(x_2) \\ \text{sign}(x_3) \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 4 & 3 \end{bmatrix} \begin{bmatrix} \tanh(x_1) \\ \tanh(x_2) \\ \tanh(x_3) \end{bmatrix}. \quad (3.18)$$

El modelo anterior representa, hasta donde el autor esta consciente, el modelo de red neuronal Hopfield con presencia de caos mas sencillo, con cuatro pesos del modelo en 0, lo cual simplifica en gran medida la implementación en circuitos electrónicos. Además, este modelo tiene la particularidad de tener un muy amplio espectro de posibilidades para el parámetro w_{12} , para los cuales es sistema muestra un comportamiento caótico para todo $w_{12} \in \mathbb{R}_+$, esta conclusión se obtiene tras las simulaciones y análisis posteriormente realizados en este trabajo.

3.3. Modelos reportados en la literatura

3.3.1. Modelo con función ReLU

La función $\text{ReLU}(\cdot)$ es una de las funciones de activación mas utilizadas en ANN, su nombre viene del ingles "Rectified Linear Unit" ó "Unidad Lineal Rectificada" es español, es una función

a trozos definida por

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad (3.19)$$

en donde $\max(\cdot)$ es una función que regresa en valor mas grande de los argumentos. Es evidente que esta función es siempre mayor o igual a 0 para todo x y, a diferencia de las primeras funciones de activación presentadas con anterioridad, no es diferenciable en todo su dominio ya que viene dada por

$$\frac{d}{dx}\text{ReLU}(x) = \begin{cases} 0, & x < 0, \\ 1, & x > 0, \end{cases}, \quad (3.20)$$

para el punto $x = 0$ la derivada no existe. Con lo que respecta a su implementación en HNN, existen una cantidad muy limitada de modelos con presencia de caos, hasta el momento en que esta tesis esta siendo escrita solo tres modelos han sido propuestos, el primero propuesto por C.Chen y F. Mina [39] incluye en el modelo la presencia de un Memristor, lo cual modifica el modelo dinámico de la HNN, después C. Chen, F. Min, et. al., proponen dos modelos mas, primero [21] tiene una estructura utilizada a lo largo de esta tesis, sin embargo su implementación en tarjetas embebidas resulta ser mas compleja, esta manera se utiliza el modelo presentado en [2] descrito por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} 1.15 & 0 \\ 0 & 0.55 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1.1 & -1.05 \\ 0.95 & 0.75 \end{bmatrix} \begin{bmatrix} \text{ReLU}(x_1) \\ \text{ReLU}(x_2) \end{bmatrix} + \begin{bmatrix} -0.5\text{sen}(0.52t) + 0.8 \\ -0.75 \end{bmatrix}, \quad (3.21)$$

este modelo es un HNN de dos dimensiones con un voltaje sinoidal de estimulación sobre la neurona x_1 .

3.3.2. Modelo con función de activación PWL

En distintas ocasiones es posible aproximar el comportamiento de distintas funciones haciendo uso de un conjunto de funciones lineales, en estos caso se suele recurrir a las llamadas "Piecewise Linear Functions" o "Función Lineal a Trozos" en español. En este caso, es posible aproximar el comportamiento de la función $\tanh(\cdot)$ por medio de tres rectas (véase Figura 3.5) propuestas [3], a esta función se denomina como $\text{satlin}(\cdot)$ y esta definida como

$$\text{satlin}(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \leq x < 1, \\ 1, & x \geq 1, \end{cases} \quad (3.22)$$

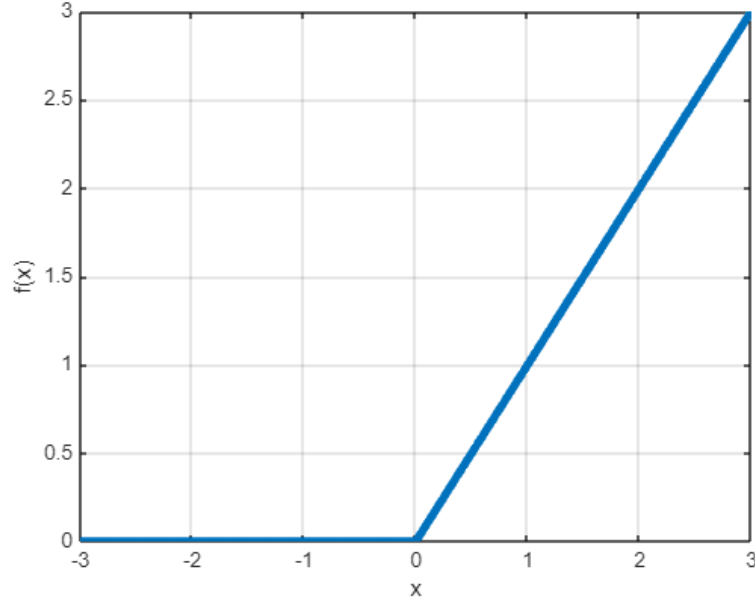


Figura 3.4: Comportamiento de la función $\text{ReLU}(x)$.

y presenta un comportamiento dinámico caótico en el modelo

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1.68 & 3.81 & -2.23 \\ -5.4 & 1.8 & -4.4 \\ -4.38 & -2.2 & -0.3 \end{bmatrix} \begin{bmatrix} \text{satlin}(x_1) \\ \text{satlin}(x_2) \\ \text{satlin}(x_3) \end{bmatrix}. \quad (3.23)$$

Para esta caso entonces se tiene una red neuronal completamente conectada. La derivada de esta función activación esta definida para todo su dominio menos en $x = 0$, y viene dada por

$$\frac{d}{dx} \text{satlin}(x) = \begin{cases} 1, & -1 < x < 1, \\ 0, & \text{otro} \end{cases} \quad \forall (x \neq -1 \wedge x \neq 1). \quad (3.24)$$

3.3.3. Modelo de red neuronal Hopfield de 4 neuronas

El trabajo realizado por Z. Njitacke et al. [22] propone un modelo HNN de 4 dimensiones con la función de activación $\tanh(\cdot)$ descrito por el sistema de ecuaciones diferenciales

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 & 0.5 & -3 & -1 \\ 0 & 2 & 3 & 0 \\ 3 & -3 & 1 & 0 \\ 100 & 0 & 0 & 170 \end{bmatrix} \begin{bmatrix} \tanh x_1 \\ \tanh x_2 \\ \tanh x_3 \\ \tanh x_4 \end{bmatrix}, \quad (3.25)$$

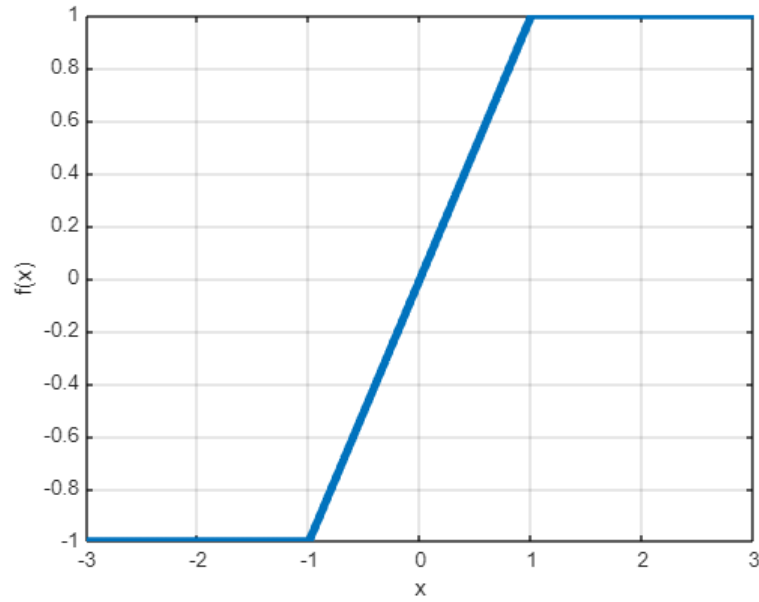


Figura 3.5: Comportamiento de la función $\text{satlin}(x)$.

entre las diversas dinámicas que se pueden encontrar en este modelo de red neuronal, las cuales serán exploradas y demostradas en posteriores secciones de esta tesis, se encuentra el hipercaos, característica que solo puede existir en sistemas dinámicos de 4 o mas dimensiones.

3.4. Análisis en base a la divergencia del sistema dinámico

Conforme un sistema dinámicos evoluciona en el tiempo entonces el volumen del espacio físico aumenta, disminuye o se conserva, conforme a esta característica los sistemas dinámicos pueden ser divididos en tres categorías [40]:

- Sistemas dinámicos conservativos. Estos son aquellos que conforme pasa el tiempo preservan su energía y, de acuerdo al teorema de Liouville, a pesar de la traslación y el cambio de forma el volumen total de dicha región permanecerá invariante. En este tipo de sistemas no pueden existir atractores.
- Sistemas dinámicos disipativos. Estos son sistemas que disipan energía conforme pasa el tiempo, es decir, el volumen del espacio físico conforme avanza el tiempo se reduce, en estos casos es posible afirmar que en este sistema existe algún o algunos atractor(es).
- Sistemas dinámicos inestables o no acotados. Estos son sistemas cuya evolución genera que el volumen del espacio físico no deje de crecer.

Esta característica es calculada a través de la operación divergente con la cual se calcula la media de la velocidad del volumen del espacio de fase generado. La operación divergente se define como sigue. Consideremos el espacio vectorial $\mathbf{f}$ definido por un conjunto abierto $\Omega \subset \mathbb{R}^n$ y consideremos sus coordenadas $\mathbf{f} = [f_1 \ f_2 \ \cdots \ f_n]^T$. Ahora supongamos que $\mathbf{f}$ es diferenciable en todos los puntos de $\mathbf{x} \in \Omega$, entonces la **divergencia** del campo F es una función $\text{div}(\mathbf{f}) : \Omega \rightarrow \mathbb{R}$ calculado por

$$\text{div}(\mathbf{f}(\mathbf{x})) = \nabla \cdot \mathbf{f}(\mathbf{x}) = \sum_{i=1}^n \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_i}, \quad (3.26)$$

donde ∇ es el operador nabla descrito por el vector $\left[\frac{\partial}{\partial x_1} \ \frac{\partial}{\partial x_2} \ \cdots \ \frac{\partial}{\partial x_n} \right]$. La divergencia tiene una importante interpretación física, esta mide *la razón de expansión del volumen generado por las variables de estado* [40]. En este sentido, el sistema estudiado es disipativo si se cumple que

$$\nabla \cdot \mathbf{f}(\mathbf{x}) < 0, \quad (3.27)$$

por otro lado, el sistema es conservativo si se cumple que

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = 0. \quad (3.28)$$

En otras palabras, *si la divergencia de un sistema es menor a cero, el sistema contiene atractores*. Usualmente la divergencia de un sistema suele ser denotada por la letra griega Λ .

Para el caso específico de una red neuronal Hopfield de n neuronas descrito por la expresión (2.1), se puede afirmar el siguiente teorema

Teorema 3.4.1. *Sea una función $f : \mathbb{R} \rightarrow \mathbb{R}$ que es continua y creciente, cuya derivada $f'(x)$ existe para todo $x \in A \subseteq \mathbb{R}$ y cuya cota superior mínima es algún real k . Consideremos a f como la función de activación para el modelo de las HNN, entonces, existe como mínimo un atractor en el sistema si se cumple que*

$$\sum_{i=1}^n c_i > k \sum_{i=1}^n w_{ii}, \quad (3.29)$$

en otras palabras es disipativo, mientras que el modelo es conservativo si se cumple

$$\sum_{i=1}^n c_i = k \sum_{i=1}^n w_{ii}. \quad (3.30)$$

Demostración. Consideremos el modelo n -dimensional de una HNN, en forma de sumatoria

esta puede escrita por

$$\dot{x}_i = -c_i x_i + b_i + \sum_{j=1}^n w_{ij} f(x_j), \quad (3.31)$$

su divergencia viene dada por

$$\begin{aligned} \Lambda &= \nabla \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} -c_1 x_1 + b_1 + \sum_{j=1}^n w_{1j} f(x_j) \\ -c_2 x_2 + b_2 + \sum_{j=1}^n w_{2j} f(x_j) \\ \vdots \\ -c_n x_n + b_n + \sum_{j=1}^n w_{nj} f(x_j) \end{bmatrix}, \\ &= \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(-c_i x_i + b_i + \sum_{j=1}^n w_{ij} f(x_j) \right) = \sum_{i=1}^n \left(-c_i + w_{ii} \frac{\partial}{\partial x_i} f(x_i) \right). \end{aligned} \quad (3.32)$$

Por hipótesis se sabe que la derivada de $f(x_i)$ existe para todo $\mathbf{x} \in A$ y esta acotada superiormente por una k , i.e., $\frac{\partial f(x_i)}{\partial x_i} \leq k$, esto implica que

$$\sum_{i=1}^n \left(-c_i + w_{ii} \frac{\partial}{\partial x_i} f(x_i) \right) \leq \sum_{i=1}^n (-c_i + k w_{ii}). \quad (3.33)$$

Para que nuestro sistema sea disipativo, y por lo tanto pueda contener atractores, se debe cumplir que su divergencia sea negativa, de esta manera se tiene

$$\sum_{i=1}^n (-c_i + k w_{ii}) < 0, \quad (3.34)$$

$$\sum_{i=1}^n c_i > k \sum_{i=1}^n w_{ii}. \quad (3.35)$$

Mientras que el sistema será conservativo si la divergencia es nula, por lo tanto si el modelo de HNN cumple el criterio

$$\sum_{i=1}^n c_i = k \sum_{i=1}^n w_{ii}. \quad (3.36)$$

□

Ejemplos ilustrativos. Apliquemos el teorema 3.4.1 las funciones de activación $\tanh(\cdot)$ y $\sigma(\cdot)$, con derivadas iguales a $1 - \tanh^2 x_i$ y $\sigma(x_i)(1 - \sigma(x_i))$. Es sencillo hallar la cota superior mínima de las derivadas antes mencionadas de la siguiente manera. Primero verifiquemos que existe una cota superior para la función $1 - \tanh^2(x)$ para todo $x \in \mathbb{R}$, denominemos a $k \in \mathbb{R}$ como la cota superior, entonces se debe cumplir que

$$1 - \tanh^2(x) = 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right)^2 = \frac{e^{2x} + e^{-2x} + 2 - e^{2x} - e^{-2x} + 2}{(e^x + e^{-x})^2} \leq k,$$

$$\begin{aligned}
&\Rightarrow \frac{4}{(e^x + e^{-x})^2} \leq k, \\
&\Rightarrow \frac{1}{((e^x + e^{-x})/2)^2} \leq k,
\end{aligned} \tag{3.37}$$

el denominador de la parte izquierda de la expresión anterior es siempre positiva y por lo tanto el valor de la fracción solo decrece, en otras palabras la cota superior k existe y su valor depende de la cota inferior máxima de la expresión $((e^x + e^{-x})/2)^2$, para hallar esta notemos que la siguiente desigualdad se cumple para todo x real

$$\begin{aligned}
&(e^{\frac{1}{2}x} - e^{-\frac{1}{2}x})^2 \geq 0, \\
&\Rightarrow e^x - 2 + e^{-x} \geq 0, \\
&\Rightarrow e^x + e^{-x} \geq 2, \\
&\Rightarrow \frac{e^x + e^{-x}}{2} \geq 1,
\end{aligned}$$

sustituyendo esta cota superior en (3.37) se tiene

$$\frac{1}{1} = 1 \leq k \quad \therefore \quad 1 - \tanh^2(x) \leq 1,$$

por lo tanto, para la función de activación tangente hiperbólica $k = 1$ es el valor de la cota superior mínima, este valor es alcanzado solo cuando $x = 0$.

Para la función sigmoide $\sigma(x)$ se sigue el siguiente procedimiento

$$\begin{aligned}
(e^{-\frac{1}{2}x} - e^{\frac{1}{2}x})^2 \geq 0 &\Rightarrow e^x + e^{-x} - 2 \geq 0 \\
&\Rightarrow e^x + e^{-x} + 2 \geq 4 \\
&\Rightarrow \frac{e^{-x}}{1 + e^{-2x} + 2e^{-x}} = \frac{1 - 1 + e^{-x}}{(1 + e^{-x})^2} \leq 1/4 \\
&\Rightarrow \frac{1}{1 + e^{-x}} \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \right) \leq 1/4 \\
&\Rightarrow \sigma(x)(1 - \sigma(x)) \leq 1/4.
\end{aligned}$$

en otras palabra $k = \frac{1}{4}$ es la cota superior mínima de la derivada de $\sigma(x)$. Entonces podemos concluir entonces que una condición suficiente para que el sistema sea disipativo, y por lo tanto es sistema soporte atractores, en el modelo HNN con la función de activación sigmoide es que se cumpla

$$\sum_{i=1}^n c_i > \frac{1}{4} \sum_{i=1}^n w_{ii}. \tag{3.38}$$

Por otro lado, para las funciones de activación $\text{ReLU}(\cdot)$ y $\text{satlin}(x)$ la cota superior de su derivada tiene un valor de $k = 1$ de manera directa, por lo que es posible aplicar la expresión

$$\sum_{i=1}^n c_i > \sum_{i=1}^n w_{ii}, \tag{3.39}$$

como criterio para la existencia de atractores.

3.4.1. Criterio numérico para la existencia de atractores en sistemas dinámicos no uniformes

En algunos sistemas la divergencia queda en términos de alguna de las variables de estado, Heidel y Zhang denominaron a los sistemas conservativos con esta característica como sistemas no uniformemente conservativos [41]. En este trabajo como veremos, tenemos sistemas disipativos con dicha característica, en este sentido nos referiremos a los sistemas dinámicos, ya sea conservativos o disipativos, como sistemas no uniformemente conservativos o disipativos.

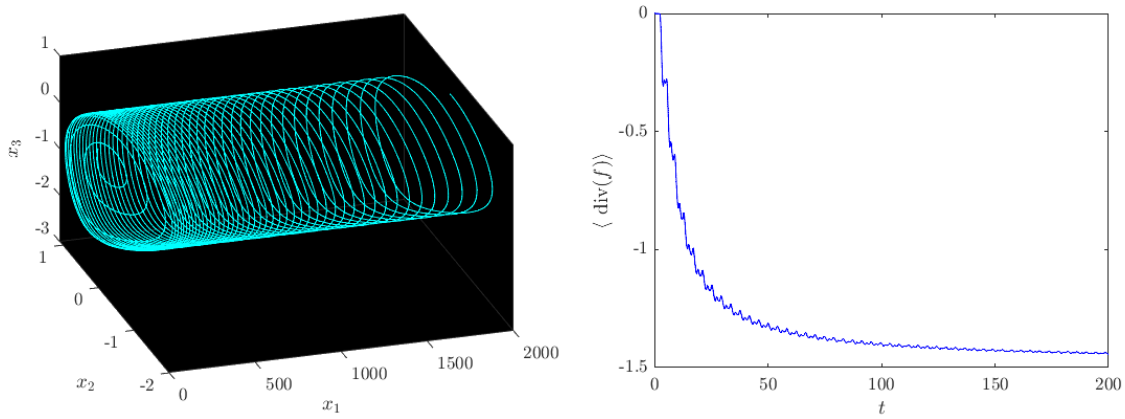
Los modelos presentados tiene una dependencia a alguna o algunas de las variables de estado, esta problemática es abordada en [40] en el cual se dividen a los sistemas conservativos en 4 categorías:

- Categoría A: Orbitas en las cuales la divergencia del sistema es uniformemente cero.
- Categoría B: Orbitas cuya divergencia tienen una dependencia a alguna o algunas de las variables de estado y su velocidad es cero en promedio como consecuencia del acotamiento del sistema.
- Categoría C: Orbitas cuya divergencia dependen de el estado del sistema, su velocidad es cero en promedio pero su demostración requiere de una transformación a un sistema hamiltoniano.
- Categoría D: Sistemas en donde la disipación del sistema depende de las condiciones iniciales y no puede ser determinado simplemente por las ecuaciones que a esta describe.

Aun cuando dicho trabajo solo considera a los sistemas conservativos, en este trabajo se tomara la esta convención y se expande para sistemas disipativos. La técnica generalmente utilizada en los trabajos especializados sobre sistemas dinámicos no uniformemente disipativos o conservativos se basa en hallar la media de la de divergencia de la forma

$$\langle \nabla \cdot \mathbf{f}(t) \rangle = \lim_{t \rightarrow \infty} \frac{\int_0^t \nabla \cdot \mathbf{f}(\tau) d\tau}{t}, \quad (3.40)$$

y aplicar el criterio antes descrito, sin embargo, Heidel y Zhang hallaron que que la divergencia no siempre ofrece un criterio correcto para la correcta identificación del tipo de sistema estudiado [41], en su trabajo hallaron en el sistema de Van Der Pol zonas donde el sistema era acotado pero la media de la divergencia era positiva. Al estudiar el modelo de HNN se encontró una zona para la cual la media de la divergencia es negativa aun cuando el sistema es desacotado. Para definir y a su vez presentar la solución que se dio a esta problemática primero definamos algunos conceptos clave.



(a) Simulación del sistema dinámico descrito en la ecuación (??) después de 200s.

(b) Comportamiento de $\langle \Lambda \rangle = \langle \nabla \cdot \mathbf{f} \rangle$ del modelo descrito en la ec. (??).

Figura 3.6: Análisis numérico de la divergencia del modelo descrito en la ec. (??).

Definición 3.4.1. Supongamos que $\mathbf{a} \in \mathbb{R}^n$ y que $r > 0$. La bola cerrada con centro $\mathbf{a}$ y radio r es el conjunto, denotado $B(\mathbf{a}, r)$, definido por

$$B(\mathbf{a}, r) := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{a}\| \leq r\}.$$

Se denota a $\mathbf{0}$ como el origen en $\mathbb{R}^n$.

Definición 3.4.2. Un conjunto $S \subset \mathbb{R}^n$ se acotado si existe un $r > 0$ tal que $S \subset B(r, \mathbf{0})$.

Definición 3.4.3. Un conjunto desacomotado es un conjunto con acotado.

Un sistema dinámico autónomo conservativo o disipativo descrito por $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ tiene una solución $\mathbf{x}$ acotada en algún conjunto $S \subset \mathbb{R}^n$ [42]. El criterio usado en la literatura usando la media de la divergencia presenta algunas inconsistencias, un ejemplo de esto fue el hallado por el modelo descrito por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0.01x_1 \\ -x_2 \\ -x_3 \end{bmatrix} + \begin{bmatrix} 0 & -2 & -9 \\ 1 & 3 & 3 \\ -1 & -3 & 0 \end{bmatrix}, \quad (3.41)$$

que es el modelo propuesto para las HNN caóticas pero suplantando $c_1 = 1$ por $c_1 = -0.01$, la simulación numérica de este sistema es presentada en la Figura 3.6a usando un tiempo de simulación de 200 segundos, notemos que x_1 se explota en magnitud y el sistema se vuelve desacomotado. En la Figura 3.6b se calcula la media de la divergencia de dicho sistema, este se aproxima a , lo cual es una inconsistencia con lo que se indica en la literatura [43].

Para tratar con esta problemática se busca desarrollar un algoritmo que sea capaz de identificar si el sistema explota en magnitud y a su vez calcule la media de la divergencia, de

esta manera se podría identificar la mencionada inconsistencia y poder definir si efectivamente el sistema dinámico estudiado es disipativo. Para poder determinar si un sistema es acotado se desarrollo en siguiente teorema.

Teorema 3.4.2. *Sea $\mathbf{x} \in \mathbb{R}^n$ la solución de un sistema dinámico descrito por $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ con condiciones iniciales $\mathbf{x}(0) = \mathbf{x}_0$. Dicha solución es acotada si en promedio su velocidad $\dot{\mathbf{x}}$ se aproxima a cero conforme $t \rightarrow \infty$.*

Demostración. La media de la velocidad de las variables de estado viene dada por

$$\langle \dot{\mathbf{x}} \rangle = \frac{\int_0^t \dot{\mathbf{x}}(\tau) d\tau}{t} \Rightarrow \langle \dot{\mathbf{x}} \rangle = \frac{\mathbf{x}(t) - \mathbf{x}_0}{t}. \quad (3.42)$$

Consideremos que la solución $\mathbf{x}(t)$ esta acotada entonces existe un radio $r > 0$ de la bola abierta centrada en el punto $\mathbf{x}_0$, entonces se cumple que

$$\|\langle \dot{\mathbf{x}} \rangle\| = \frac{\|\mathbf{x}(t) - \mathbf{x}_0\|}{t} < \frac{r}{t}, \quad (3.43)$$

para un $t \rightarrow \infty$ y sabiendo que la métrica $\|\cdot\|$ cumple la positividad tenemos

$$0 \leq \lim_{t \rightarrow \infty} \|\langle \dot{\mathbf{x}} \rangle\| \leq \lim_{t \rightarrow \infty} \frac{r}{t} = 0, \quad (3.44)$$

que en base al teorema de comparación para la desigualdad tenemos que $\lim_{t \rightarrow \infty} \|\langle \dot{\mathbf{x}} \rangle\| = 0$. $\square$

La media de la divergencia puede ser usada como una indicación del tipo de sistema dinámico estudiado [43], sin embargo como se mostró este criterio debe tomar en cuenta que el sistema es acotado, es por esta razón que para analizar el tipo de sistema que se estudia se desarrollo el algoritmo presentado en la Figura 3.7 en donde se calcula la media de la divergencia conforme el sistema evoluciona en el tiempo y a su vez se asegura que dicho sistema esta acotado en base al teorema 3.4.2.

En general podemos definir de la misma manera el criterio indicado al inicio de este capítulo como sigue: Un sistema dinámico acotado descrito por $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ se dice conservativo si $\langle \nabla \cdot \mathbf{f} \rangle$ se aproxima a cero conforme $t \rightarrow \infty$ y disipativo si tiene a un valor negativo. Para identificar el tipo de sistema dinámico de cada modelo de HNN considerado obtengamos la media de la divergencia y como consecuencia concluir si el sistema es capaz de contener atractores. Para este ejercicio se considero un tiempo de simulación de 500 segundos utilizando dos parámetros de análisis distintos para cada modelo y los cuales son indicados en la respectivamente figura solución.

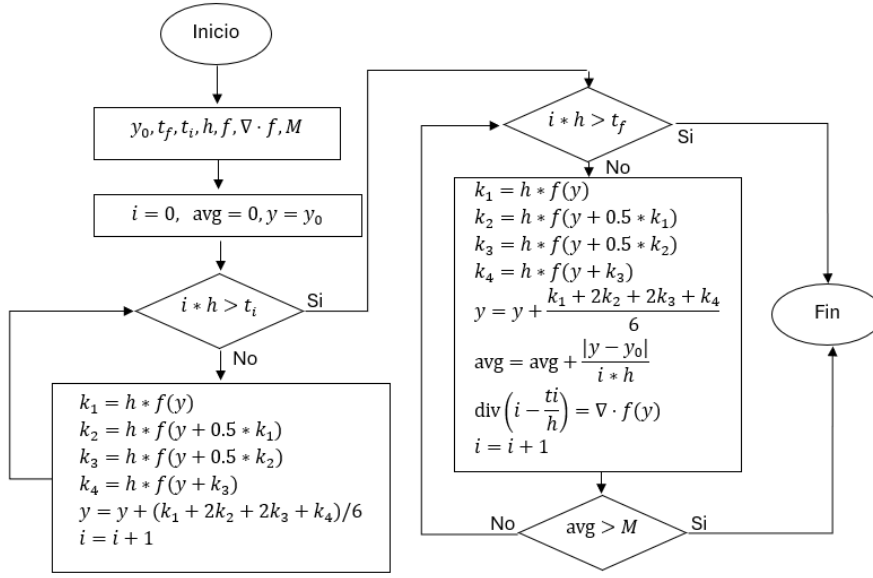


Figura 3.7: Algoritmo para la obtención de la media de la divergencia considerando el acotamiento de la solución en base a la media de las variables de estado.

3.4.2. Atractores en el modelo de HNN con FA tangente hiperbólica

Aplicando el operador divergencia sobre el modelo descrito en la ec. (3.5) se tiene

$$\Lambda = -3 \tanh^2 x_2, \quad (3.45)$$

que claramente indica que el volumen del espacio de fase tiene una dependencia en el estado del sistema. Por un lado la norma de la media de la velocidad de los estados para el caso específico con $w_{12} = -2$ y $w_{13} = -9$ es presentada en la Figura 3.8a, como es esperable para un sistema acotado tenemos que la norma se aproxima a cero y como tal se demuestra que el sistema es acotado de manera numérica para un tiempo de 200 s. Ahora por medio de algoritmo presentado con anterioridad se obtuvo la media de la divergencia del modelo con FA tangente hiperbólica, este es presentado en la Figura 3.8b con un mapa de calor explorando la zona paramétrica formada por w_{12} y w_{13} , los valores del mapa de calor indican que el sistema es no uniformemente disipativo para toda la zona paramétrica estudiada y podemos afirmar la existencia de al menos un atractor en el sistema.

El atractor extraño formado por el sistema tras 300 segundos de simulación se presenta en la Figura 3.9 en una grafica tridimensional, en esta se observan dos enrollamientos.

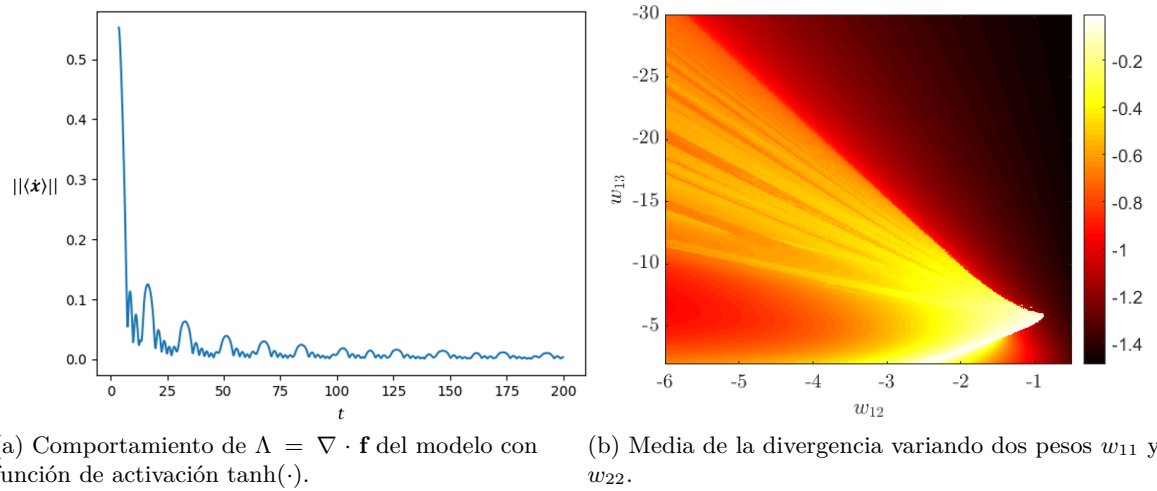


Figura 3.8: Análisis numérico de la divergencia del modelo de HNN con FA tangente hiperbólica.

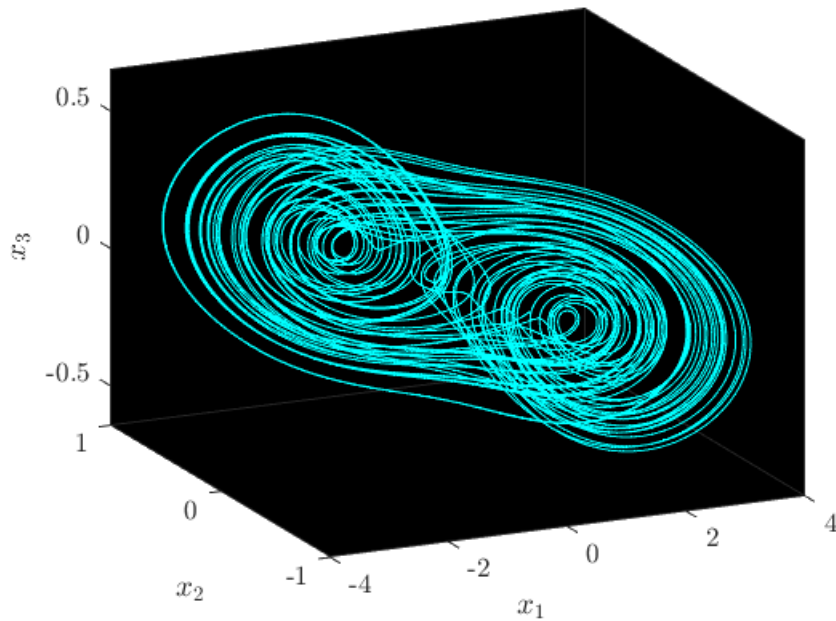
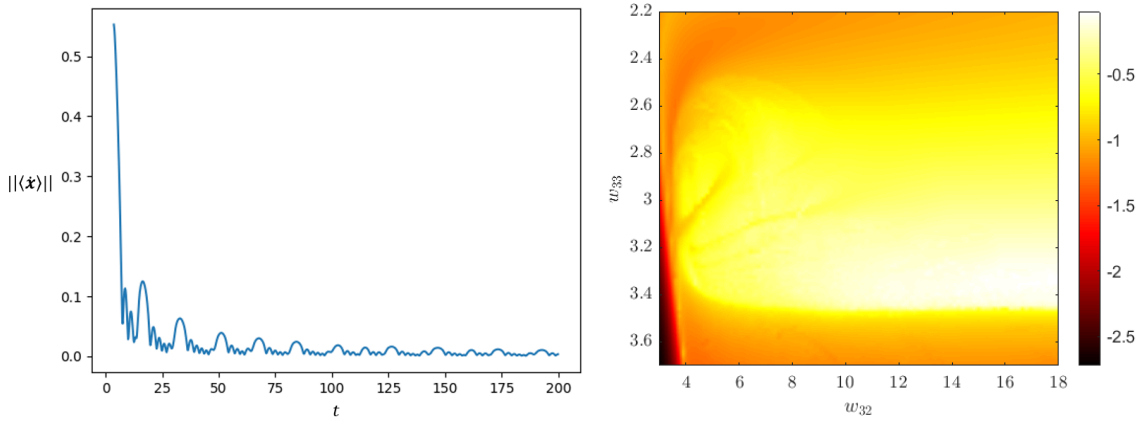


Figura 3.9: Retrato de fase tridimensional formado por el modelo con función de activación tangente hiperbólica.



(a) Comportamiento de $\Lambda = \nabla \cdot \mathbf{f}$ del modelo con función de activación $\tanh(\cdot)$. (b) Media de la divergencia variando dos pesos w_{11} y w_{32} .

Figura 3.10: Análisis numérico de la divergencia del modelo de HNN con FA tangente hiperbólica.

3.4.3. Atractores en el modelo de HNN con FA signo

El mismo análisis de la media de la divergencia aplicada en el modelo con FA $\tanh(\cdot)$ es realizado con el modelo con FA signo, el cual tiene una divergencia similar calculada por

$$\Lambda = -3 \tanh^2(x_3), \quad (3.46)$$

3.4.4. Atractores en el modelo de HNN con FA sigmoide

Consideremos el modelo descrito por (3.15) con la función de activación $\sigma(x)$, usando el operador divergencia se obtiene

$$\Lambda = -3 + 7.97\sigma(x_1)(1 - \sigma(x_1)) + 6.83\sigma(x_2)(1 - \sigma(x_2)) + 3.68\sigma(x_3)(1 - \sigma(x_3)), \quad (3.47)$$

la derivada de $\sigma(x)$ es simétrica y máxima cuando x se aproxima a 0, en base a esto podemos deducir que la divergencia sera máxima cuando las variables de estado estén cercanas a cero. La media de la velocidad de las variables de estado es presentado en la Figura 3.12a, cuyo valor tiene a cero conforme el tiempo aumenta y podemos concluir así que el sistema es acotado. Por otro lado, la media de la divergencia es representado en el mapa de calor de la Figura 3.12b usando dos parámetros de análisis w_{11} y w_{22} , del lado derecho del mapa de calor se indica el valor obtenido para $\langle \nabla \cdot \mathbf{f} \rangle$ en un tiempo de 500 s, notemos que el valor máximo encontrado fue aproximado a -3.

Podemos concluir entonces que el sistema es divergente y puede soportar atractores. En la

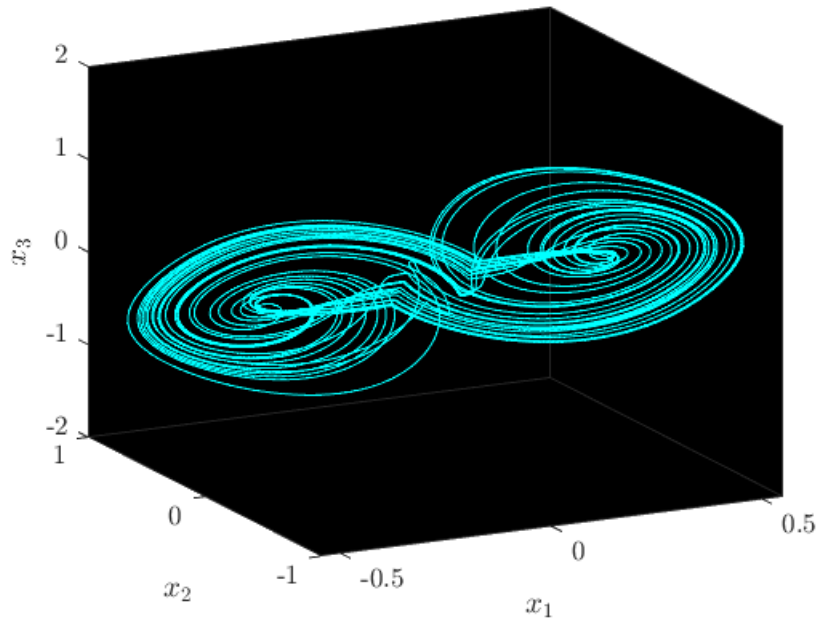
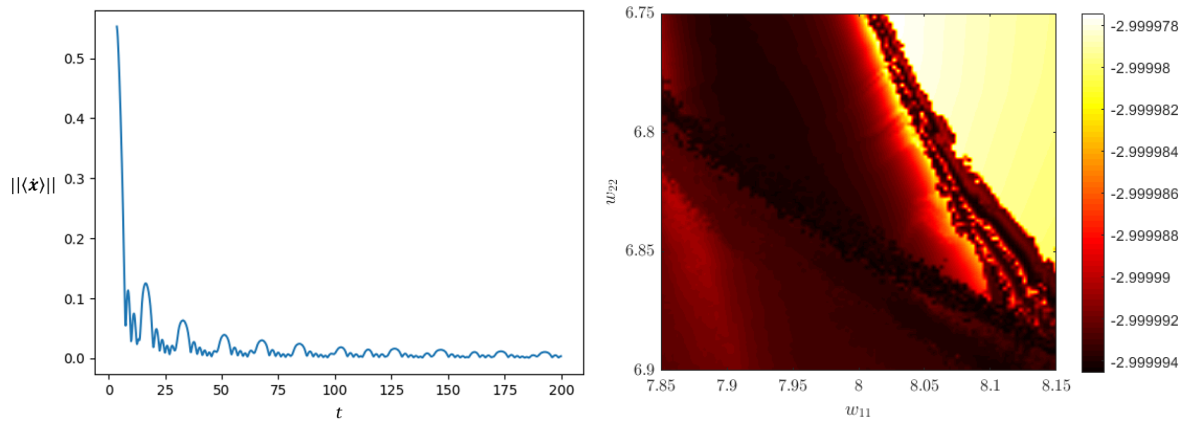


Figura 3.11: Retrato de fase tridimensional formado por el modelo con función de activación tangente hiperbólica.



(a) Comportamiento de $\Lambda = \nabla \cdot \mathbf{f}$ del modelo con función de activación $\sigma(\cdot)$.

(b) Media de la divergencia variando dos pesos w_{11} y w_{22} .

Figura 3.12: Análisis numérico de la divergencia del modelo de HNN con FA sigmoide.

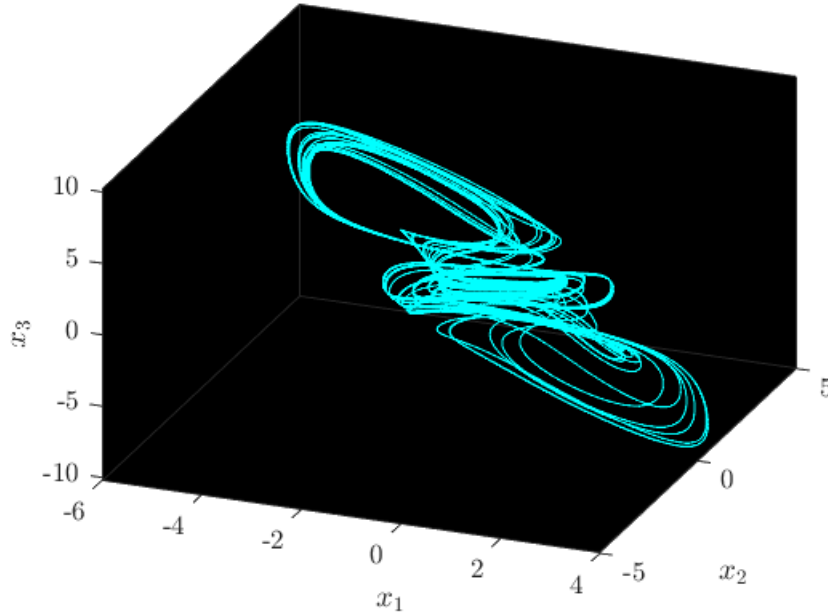


Figura 3.13: Retrato de fase formado por el modelo con función de activación sigmoide para el plano a) $x_1 - x_2$ y b) $x_1 - x_3$.

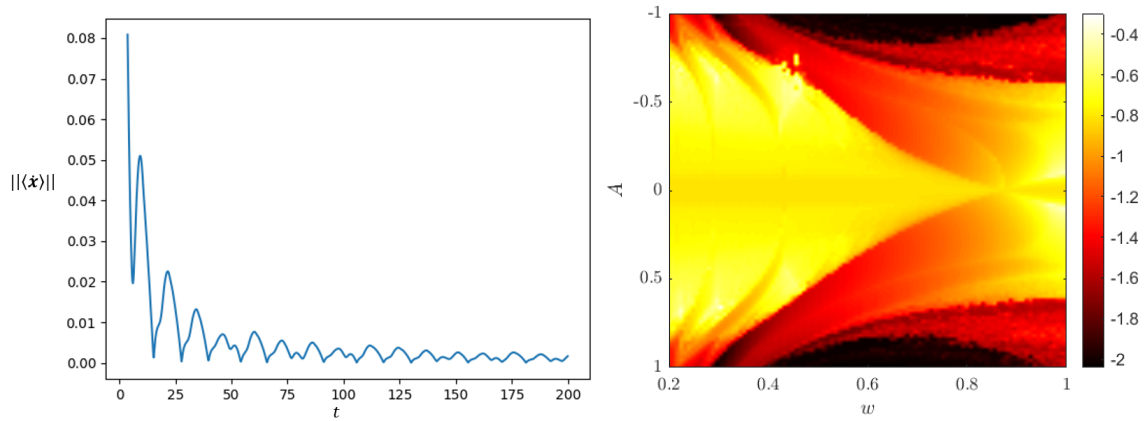
Figura 3.13 se observa el retrato de fase del sistema tras una simulación 600s, considerando una transiente de 200 segundos con un paso de integración $1e-3$ y con condiciones iniciales iguales a $(0.1, 0, 0)$.

3.4.5. Atractores en el modelo de HNN con FA ReLu(x).

Ahora consideremos el modelo con la función de activación ReLu(x), la velocidad de expansión o contracción del volumen del espacio fásico esta dada por

$$\Lambda = -1.7 + 1.1g(x_1) + 0.75g(x_2), \quad (3.48)$$

donde $g(x)$ es la derivada de la función ReLu(x) dada por la expresión (3.20), esto indica que en la Λ puede tomar 4 valores distintos -0.6, -0.95, -1.7 y 0.2; notemos que uno de estos posibles valores es positivo y por lo tanto el sistema necesita de un análisis numérico de la divergencia. En primer lugar la media de la velocidad de las variables de estado se observa en la Figura 3.14a, obteniendo el área bajo la curva de $\Lambda(t)$ se obtiene un valor alrededor de -56.84, lo que indica que el valor de Λ es mayoritariamente negativo. Además, la Figura 3.14 b) se obtiene la norma del valor medio de la velocidad de las variables de estado, este resulta aproximarse a 0 conforme el tiempo aumenta como es esperable en sistema disipativo. En resumen podemos



(a) Norma de la velocidad media de las variables de estado del modelo 3.21. (b) Media de la divergencia variando la frecuencia w y la amplitud A de la señal externa o bias.

Figura 3.14: Análisis numérico de la divergencia en el modelo HNN con FA $\text{ReLu}(\mathbf{x})$.

concluir que este sistema tiene la capacidad de contener atractores y esto se puede confirmar al simular el sistema como fue hecho en la Figura 3.15 con un tiempo de simulación igual a 300 segundos, un paso de integración de $1e-3$ y con condiciones iniciales $(0.1, 0)$, aquí se observa un solo atractor en el sistema.

3.4.6. Atractores en el modelo de HNN con FA PWL.

Finalmente, retomando el modelo con la función PWL denominada $\text{satlin}(x)$ se tiene que

$$\Lambda = -3 + 1.68r(x_1) + 1.8r(x_2) - 0.3r(x_3), \quad (3.49)$$

donde $r(x)$ es la derivada de la función PWL definida en (3.24), de manera simular a la función $\text{ReLu}(x)$ se tiene que Λ tiene una cantidad limitada de valores, en específico puede tomar 8 valores distintos y dependen si el valor de los estados están en el rango de $|x_i| < 1$ para $i = (1, 2, 3)$, estos son 0.18, -1.5, -1.2, -3.3, -3, 0.48, -1.62 y -1.32; notemos que algunos de los posibles valores de la divergencia son positivos. El resultado de la media de las velocidades de las variables de estado se presenta en la Figure 3.16a, de nuevo, la media se aproxima a cero y de esta manera es sistema es numéricamente demostrado como acotado. La media de la divergencia por su lado es presentada en la Figura 3.16b con un mapa de calor marcando un valor máximo de como se espera de un sistema dinámico no uniforme disipativo.

Como resultados podemos afirmar la presencia de atractores en el sistema estudiado, estos son presentados en la Figura 3.17 en donde se observan dos claros scrolls.

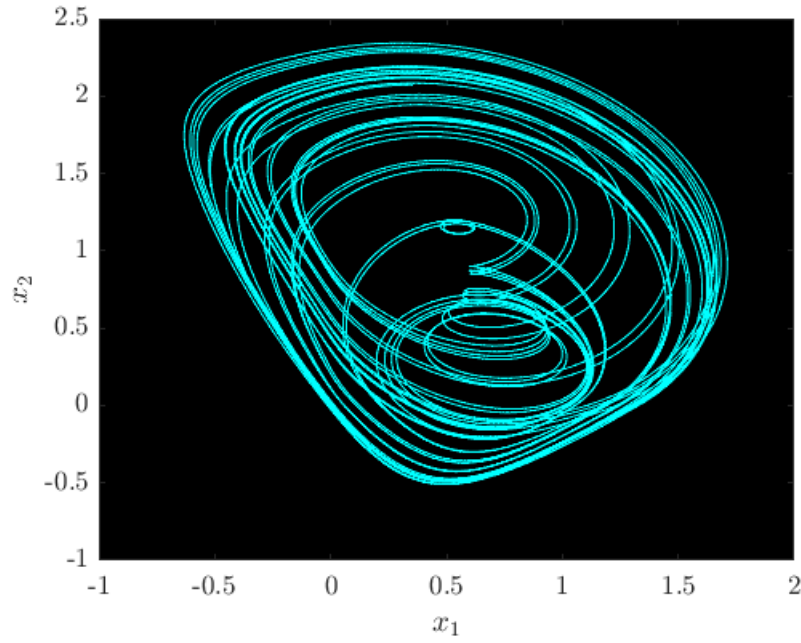
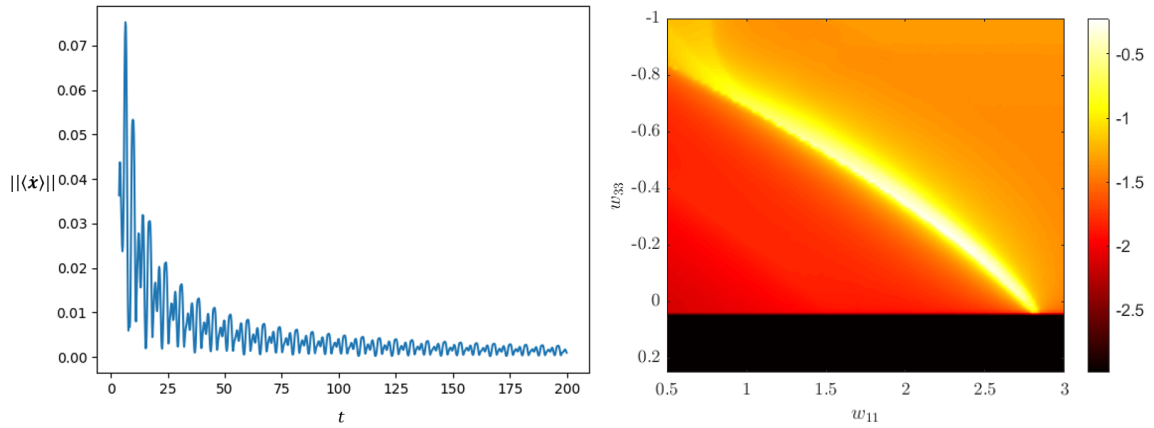


Figura 3.15: Retrato de fase del modelo con función de activación $\text{ReLU}(x)$.



(a) Norma de la velocidad media de las variables de estado del modelo (3.23).

(b) Media de la divergencia del modelo (3.23) variando dos pesos w_{11} y w_{33} .

Figura 3.16: Análisis de la divergencia del modelo con función de activación PWL .

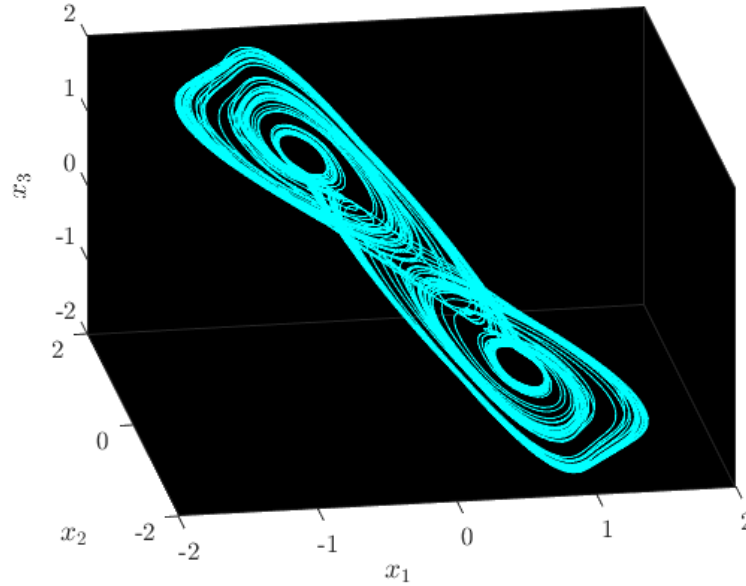


Figura 3.17: Retrato de fase del modelo con función de activación PWL.

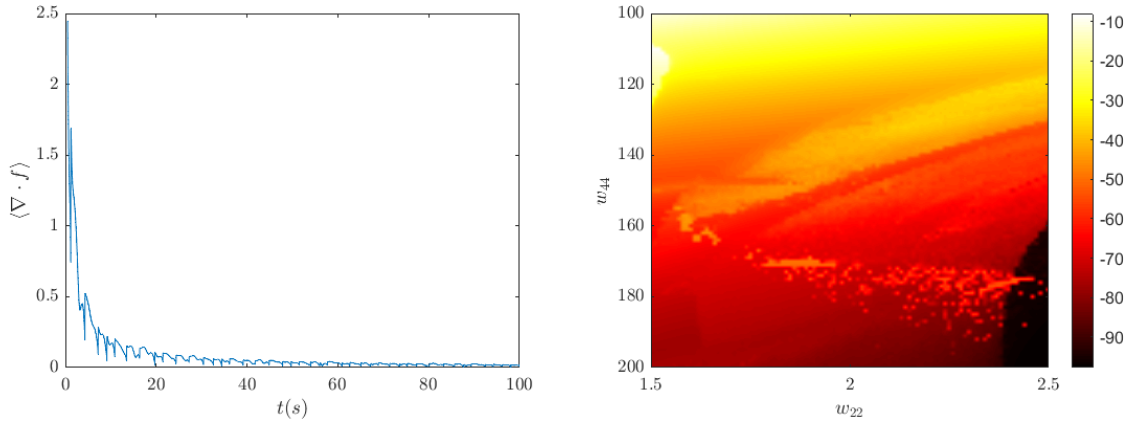
3.4.7. Atractores en el modelo HNN de 4 dimensiones

Para el modelo de 4 neuronas la divergencia media esta dada por la expresión

$$\Lambda = -101 - \tanh^2 x_1 + w_{22}(1 - \tanh^2 x_2) - \tanh^2 x_3 + w_{44}(1 - \tanh^2 x_4), \quad (3.50)$$

donde w_{22} y w_{44} son los pesos a explorar y cuyo valor en el trabajo de referencia son 2 y 170 respectivamente [22]; tomando estos mismo valores se obtuvo la norma de la velocidad de las variables de estado y se grafico en la Figura 3.16a y junto a el se explora la media de la divergencia de dicho sistema dinámico analizando una zona paramétrica mas grande.

Dado que se tiene un sistema de 4 dimensiones el numero total de retratos de fase visibles aumenta a 6 planos bidimensionales distintos y a 3 planos tridimensionales distintos, estos ultimos fueron obtenidos, graficados y son presentados en la Figura 3.19. A diferencia de los modelos previamente presentados no es posible observar enrrollamientos distintivos en el sistema. En resumen, el análisis del valor de Λ en el sistema de HNN's nos permite concluir si el sistema es conservativo, disipativo o no acotado, lo cual implica la existencia o no de atractores en el sistema, además, este esta íntimamente relacionado con el acotamiento del sistema, para este existen un método mas común que sera presentado en la siguiente sección.



(a) Norma de la velocidad media de las variables de estado del modelo HNN de 4 dimensiones descrito en la ec. (3.25). (b) Media de la divergencia del modelo (3.25) variando dos pesos w_{22} y w_{44} .

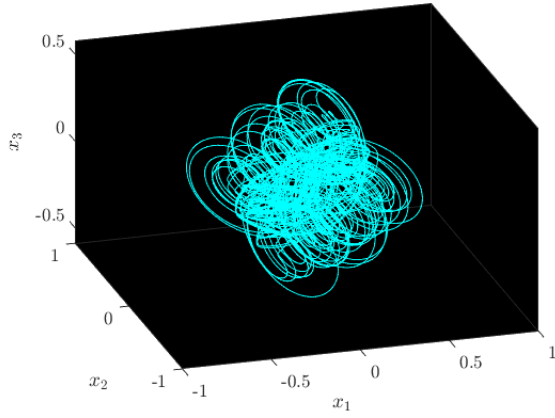
Figura 3.18: Análisis de la divergencia del modelo HNN hipercaótico.

3.5. Análisis de estabilidad en el sentido de Lyapunov

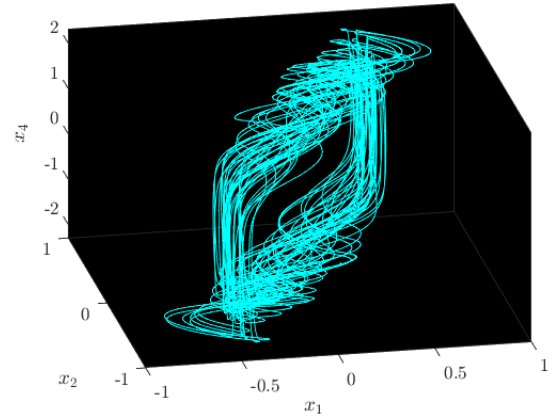
Cuando se habla que un sistema dinámico es estable implica que existe alguna región $0 < D < \mathbb{R}^n$ en espacio fásico la cual contiene la órbita generada por el sistema, mientras que la estabilidad asintótica implica que el sistema no solo está acotado, sino que se aproxima a un punto en específico en el espacio. Cabe mencionar que el concepto de estabilidad también implica la existencia de al menos un atractor en el sistema, en este sentido, los resultados obtenidos en la sección 3.4 deben coincidir con la estabilidad del sistema. Como es discutido en [38], la estabilidad de la red neuronal Hopfield y extensiones ha sido un tema de investigación desde su concepción, algunos resultados obtenidos requieren del concepto de estabilidad de Lyapunov. La estabilidad de Lyapunov es de suma relevancia en el estudio de sistemas no lineales debido a que por medio de un sencillo criterio es posible definir la estabilidad de un sistema, este es descrito en [44] y requiere de la propuesta de una función candidata de Lyapunov $V : \mathbb{R}^n \rightarrow \mathbb{R}$ definida positiva cuya derivada temporal cumpla ser definida semi-negativa para sistemas estables y definida negativa para sistemas asintóticamente estables. Con el fin entonces de analizar la estabilidad de nuestro sistema se desarrollaron los siguientes dos teoremas, el primero especifica una condición para la estabilidad asintótica de la HNN, mientras que el segundo demuestra el acotamiento del sistema para alguna región en el espacio fásico.

Teorema 3.5.1. *El modelo de red neuronal Hopfield de n neuronas sin bias descrito por*

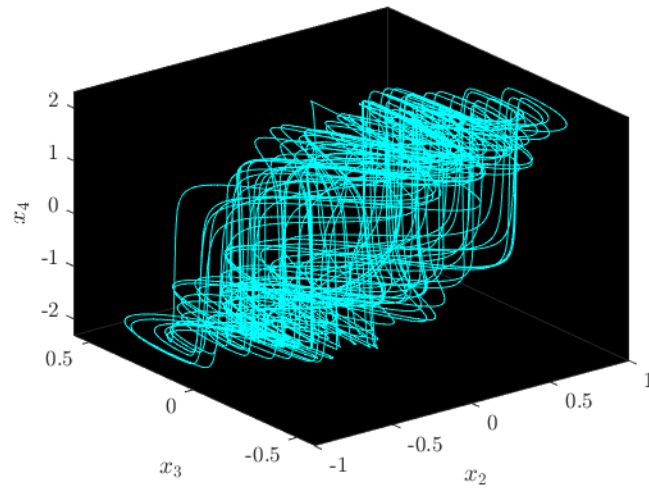
$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{f}(\mathbf{x}), \quad (3.51)$$



(a) Plano tridimensional $x_1 - x_2 - x_3$ del modelo 4D-HNN descrito en la ec. (3.25).



(b) Plano tridimensional $x_1 - x_2 - x_4$ del modelo 4D-HNN descrito en la ec. (3.25).



(c) Plano tridimensional $x_1 - x_2 - x_3$ del modelo 4D-HNN descrito en la ec. (3.25).

Figura 3.19: Retratos de fase tridimensionales del modelo 4D-HNN.

donde $C \in \mathbb{R}^{n \times n}$ es una matriz diagonal definida positiva, $W \in \mathbb{R}^{n \times n}$ la matriz de pesos y $\mathbf{f}(\mathbf{x})$ un vector con la función de activación del modelo la cual cumple ser al menos creciente, impar y $|\mathbf{f}(\mathbf{x})| \leq |\mathbf{x}|$, es asintóticamente estable si los elementos de la diagonal principal de la matriz de pesos son cero y a su vez esta es semi-definida negativa.

Demostración. Sea una función candidata de Lyapunov $V(\mathbf{x})$ definida como

$$V(\mathbf{x}) = \frac{1}{2} \mathbf{f}(\mathbf{x})^T \mathbf{f}(\mathbf{x}), \quad (3.52)$$

evidentemente definida positiva con derivada temporal descrita por

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} (-C\mathbf{x} + W\mathbf{f}(\mathbf{x})) \\ &= -\mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} C\mathbf{x} + \mathbf{f}(\mathbf{x})^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} W\mathbf{f}(\mathbf{x}) \\ &\leq -|\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} C|\mathbf{x}| + |\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} W|\mathbf{x}| \\ &= -|\mathbf{x}|^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} (C - W) |\mathbf{x}|, \end{aligned} \quad (3.53)$$

por definición sabemos que la derivada de una función estrictamente creciente es mayor a cero, de esta manera podemos concluir que la ultima expresión en 3.53 es una función definida negativa si se cumple $C - W > \mathbf{0}$, ahora consideremos a $-W$ como una matriz semi-definida positiva con los elementos en su diagonal iguales a cero, esto indica que la matriz $C + (-W)$ definida por

$$C + (-W) = \begin{bmatrix} c_1 & -w_{12} & \cdots & -w_{1n} \\ -w_{21} & c_2 & \cdots & -w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1} & -w_{n2} & \cdots & c_n \end{bmatrix} > \mathbf{0}, \quad (3.54)$$

es forzosamente definida positiva y por lo tanto el sistema de redes neuronales Hopfield, bajo estas condiciones, es asintóticamente estable. $\square$

Este resultado es comúnmente hallado en la literatura para asegurar el comportamiento asintóticamente estable de la red neuronal Hopfield, dado que nuestro interés es que la HNN presente periodicidad, por lo tanto, las matrices de pesos en nuestro sistema deben tener al menos un peso en la diagonal de la matriz de pesos distinto de cero y no ser definida negativa o, lo que es equivalente, no ser simétrica. En base a este resultado, más un trabajo numérico, fueron hallados las matrices de pesos para los modelos propuestos 3.5 y 3.18. Notemos que este teorema aplica para todos los modelos, incluyendo al modelo 3.15 con función sigmoide debido a su transformación y el modelo con función signo debido a que esta puede ser aproximada con una función tangente hiperbólica como se muestra en 3.17, con excepción del modelo con función de activación $\text{ReLU}(\cdot)$ debido al bias aplicado, en este sentido, el teorema 3.5.1 solo aplica para modelos autónomos.

Teorema 3.5.2. *El modelo de red neuronal Hopfield de n neuronas sin bias descrito por*

$$\dot{\mathbf{x}} = -\mathbf{x} + W \tanh \mathbf{x}, \quad (3.55)$$

esta acotada en una región $D \subset \mathbb{R}^n$ donde se cumple la desigualdad

$$\sum_{i=1}^n x_i^2 \geq \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \quad (3.56)$$

Demostración. Sea una función candidata de Lyapunov $V(\mathbf{x})$ definida como

$$V(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n x_i^2, \quad (3.57)$$

entonces su derivada temporal esta dada por

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \sum_{i=1}^n x_i \dot{x}_i \\ &= -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n x_i \sum_{j=1}^n w_{ij} \tanh x_j \\ &= -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i \tanh x_j \\ &\leq -\sum_{i=1}^n x_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \end{aligned} \quad (3.58)$$

La última desigualdad se cumple debido a que $|x| \geq \tanh(x)$, reescribiendo se tiene

$$\dot{V}(\mathbf{x}) \leq -2V(\mathbf{x}) + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|. \quad (3.59)$$

Consideremos a $D_0 \geq 0$ una región suficientemente grande. Para todo punto $\mathbf{x}$ que satisface $V(\mathbf{x}) = D$ con $D > D_0$ donde existe la desigualdad

$$2V(\mathbf{x}) \geq \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i |x_j|, \quad (3.60)$$

entonces en la superficie $\{\mathbf{x} | V(\mathbf{x}) = D\}$ se cumple $\dot{V}(\mathbf{x}) \leq 0$ y consecuentemente el conjunto definido por $\{\mathbf{x} | V(\mathbf{x}) \leq D\}$ es una región acotada con todas las soluciones del modelo de la HNN. $\square$

La demostración anterior requiere de la desigualdad $\tanh x \leq |x|$, considerando las funciones de activación estudiadas en este trabajo es evidente que la desigualdad $\text{satlin}(x) \leq |x|$ también se cumple y por lo tanto el modelo 3.23 cumple el mismo teorema. La función $\sigma(x)$ no cumple la misma desigualdad sin embargo puede ser expresada en términos de la tangente hiperbólica y su modelo fue obtenido usando como base otro con la misma estructura mostrada en el teorema en cuestión, por lo tanto, podemos afirmar que el modelo 3.15 esta acotado para alguna región

en el espacio fásico. Por otro lado, la función de activación $\text{sign}(x)$ es siempre menor o igual a uno, en este sentido el resultado del teorema 3.5.2 es ligeramente afectado pero sigue existiendo un conjunto D para el cual el sistema esta acotado. Finalmente para la función $\text{ReLU}(x)$ se tiene la siguiente demostración desarrollada en [39], consideremos una función de Lyapunov $V(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$, se tiene que su derivada es

$$\begin{aligned}\dot{V}(\mathbf{x}) &= -\mathbf{x}^T C \mathbf{x} + \mathbf{x}^T W \text{ReLU}(\mathbf{x}) + \mathbf{x}^T \mathbf{b} \leq -\mathbf{x}^T C \mathbf{x} + \mathbf{x}^T W \mathbf{x} + \|\mathbf{b}\| \cdot \|\mathbf{x}\| \\ &\leq -(\|C\| - \|W\|)\mathbf{x}^2 + \|\mathbf{b}\| \cdot \|\mathbf{x}\|,\end{aligned}\quad (3.61)$$

el lado derecho de la expresión anterior no es definida negativa, supongamos que la solución del modelo con función de activación $\text{ReLU}(x)$ es $\Gamma(t)$. La derivada $\dot{V}(\mathbf{x})$ es negativa fuera del conjunto $\|\mathbf{x} - \Gamma\| \leq \|\mathbf{b}\|/\|\mathbf{W}\| - 1$, con $\|\mathbf{b}\|^2/2(\|\mathbf{W}\| - 1)^2 < \eta$, la solución iniciando en el conjunto $V(\mathbf{x}) \leq \eta$ permanecerá en el para todo tiempo futuro, dado que la derivada de la función de Lyapunov es negativa en la frontera $V(\mathbf{x}) \leq \eta$, consecuentemente la solución esta acotada por la frontera $\sqrt{(2\eta)}$.

3.6. Análisis de estabilidad alrededor de los puntos de equilibrio

Una vez demostrada la existencia de atractores en el sistema dinámico es importante ser capaz de caracterizar el comportamiento del modelo alrededor de ellos, de esta manera es necesario realizar un análisis alrededor de los puntos de equilibrio en cada uno de los modelos presentados. Como fue mencionado en la introducción de este trabajo, los puntos de equilibrio de un sistema dinámico son hallados al resolver

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{0} , \quad (3.62)$$

que representan los puntos para los cuales el sistema se mantiene estático. Para los casos específicos del modelo de redes neuronales Hopfield, los puntos de equilibrio $\tilde{\mathbf{x}}$ se calculan resolviendo el sistema

$$c_i x_i^* = \sum_{j=1}^n w_{ij} f(x_j^*) + b_i , \quad (3.63)$$

lo cual es evidentemente un sistema de ecuaciones no lineales. Para su obtención se desarrolló un programa en Matlab y la función `solve(·)` para la solución de un sistema de ecuaciones realizando un barrido con distintas condiciones iniciales, los resultados obtenidos fueron registrados en la Tabla 3.1. Con respecto a el modelo con función de activación ReLU , este es un sistema no autónomo y su punto de equilibrio es calculado solo para $t = 0$. La caracterización de los puntos de equilibrio encontrados se realiza linealizando el modelo de las HNN's , usando

Tabla 3.1: Puntos de equilibrio para los distintos modelos de redes neuronales.

Función de activación	Puntos de equilibrio
Tangente hiperbólica	$\begin{bmatrix} -1.7801 & 0.2753 & 0.1390 \\ 0.0000 & 0.0000 & 0.0000 \\ 1.7801 & -0.2753 & -0.1390 \end{bmatrix}$
Sigmoide	$\begin{bmatrix} -0.0002 & -0.0152 & 0.0131 \\ -0.8383 & -0.7950 & 5.5905 \\ 0.8494 & 0.7688 & -5.6351 \end{bmatrix}$
ReLU	$\begin{bmatrix} 0.6354 & 0.7316 \end{bmatrix}$
PWL	$\begin{bmatrix} 0.7093 & -0.7119 & -1.2408 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.7093 & 0.7119 & 1.2408 \end{bmatrix}$
Signo	$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$

el procedimiento descrito en [44], y evaluando los eigenvalores obtenidos por

$$\det \left(I\lambda - \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\tilde{\mathbf{x}}} \right) = 0, \quad (3.64)$$

donde $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$ es el jacobiano del sistema dinámico. De manera generalizada para el modelo de redes neuronales Hopfield los eigenvalores se obtienen por

$$\begin{vmatrix} \lambda + c_1 - w_{11}f'(\tilde{x}_1) & -w_{12}f'(\tilde{x}_2) & \cdots & -w_{1n}f'(\tilde{x}_n) \\ -w_{21}f'(\tilde{x}_1) & \lambda + c_2 - w_{22}f'(\tilde{x}_2) & \cdots & -w_{2n}f'(\tilde{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1}f'(\tilde{x}_1) & -w_{n2}f'(\tilde{x}_2) & \cdots & \lambda + c_n - w_{nn}f'(\tilde{x}_n) \end{vmatrix} = 0, \quad (3.65)$$

donde $f'(\tilde{x}_i)$ representa la derivada de la función de activación evaluada en el punto de equilibrio $\tilde{x}_i$ para todo $i = (1, 2, \dots, n)$. Consideremos el caso del modelo de HNN para 3 neuronas, desarrollando posible encontrar

$$\begin{aligned} \lambda^3 + \lambda^2 \sum_{i=1}^3 (c_i - f'(\tilde{x}_i)w_{ii}) + \lambda \left[f'(\tilde{x}_1)f'(\tilde{x}_2) \begin{vmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{vmatrix} + f'(\tilde{x}_1)f'(\tilde{x}_3) \begin{vmatrix} w_{11} & w_{13} \\ w_{31} & w_{33} \end{vmatrix} \right] \\ + \lambda \left[f'(\tilde{x}_2)f'(\tilde{x}_3) \begin{vmatrix} w_{22} & w_{23} \\ w_{32} & w_{33} \end{vmatrix} + c_1c_2 + c_1c_3 + c_2c_3 + \sum_{i=1}^3 \sum_{\substack{j=1 \\ i \neq j}}^3 c_i f'(\tilde{x}_j)w_{jj} \right] \\ + c_1c_2c_3 - |W| \prod_{i=1}^3 f'(\tilde{x}_i) = 0, \quad (3.66) \end{aligned}$$

$$\Rightarrow \lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0, \quad (3.67)$$

donde

$$\begin{aligned} a_0 &= c_1c_2c_3 - |W| \prod_{i=1}^3 f'(\tilde{x}_i), \\ a_1 &= \left[f'(\tilde{x}_1)f'(\tilde{x}_2) \begin{vmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{vmatrix} + f'(\tilde{x}_1)f'(\tilde{x}_3) \begin{vmatrix} w_{11} & w_{13} \\ w_{31} & w_{33} \end{vmatrix} \right] \\ &\quad + \left[f'(\tilde{x}_2)f'(\tilde{x}_3) \begin{vmatrix} w_{22} & w_{23} \\ w_{32} & w_{33} \end{vmatrix} + c_1c_2 + c_1c_3 + c_2c_3 + \sum_{i=1}^3 \sum_{j=1, i \neq j}^3 c_i f'(\tilde{x}_j) w_{jj} \right], \\ a_2 &= \sum_{i=1}^3 (c_i - f'(\tilde{x}_i)w_{ii}), \end{aligned}$$

de este resultado se pueden obtener distintas conclusiones con respecto a la posición de los eigenvalores en el plano complejo, dependiendo de esta es posible separarlos en 8 categorías [45] las cuales se resumen la Tabla 3.2, cada uno de estos puntos tiene un comportamiento esperado como se muestra en la Figura ?? . Es sabido que para sistemas caóticos autoexcitados, estos son, sistemas cuyos atractores están localizados en los puntos de equilibrio del sistema dinámico, como lo son los trabajados en esta tesis, los puntos equilibrio en donde se forman los scrolls son de tipo índice II, entre ellos se encuentra un punto de equilibrio de índice I, el cual conecta ambos scrolls [46]. Retomando la expresión 3.66 y de acuerdo al criterio de Routh-Hurwitz las partes reales de esta expresión son positivas, y por ende el punto de equilibrio es índice I si se cumple

$$\begin{aligned} a_2 &> 0 \\ a_2a_1 - a_0 &> 0 \\ a_0 &> 0 \end{aligned} \quad (3.68)$$

Los eigenvalores calculados para cada uno de los modelos con sistemas autónomos son resumidos en las tablas 3.3, 3.4, 3.5 y 3.6 junto a su clasificación en base al trabajo [45]. Con regularidad cuando se explora la dinámica de algún modelo no lineal es buena práctica analizar también los parámetros cercanos a los estudiados, en este sentido la mencionada Tabla analiza los eigenvalores variando algunos parámetros de los modelos.

En base a los resultados obtenidos se puede afirmar que los modelos con funciones de activación tangente hiperbólica, PWL y sigmoide presentan un comportamiento esperado alrededor de sus puntos de equilibrio con respecto a lo mencionado en [46]. El modelo con una función de activación signo sin embargo presenta un comportamiento interesante en su punto de equilibrio trivial, es justo en $x = 0$ donde la función tiene un cambio abrupto y como tal no tiene derivada en este punto, es por esto que se aproximó la función signo con $\tanh(400x)$, lo cual tiene como derivada $400(1 - \tanh^2(400x))$ y por lo tanto el eigenvalor obtenido es un aproximado, sin embargo, el principal interés es la posición del eigenvalor en el plano complejo, este es clasificado como un punto silla de índice-1. Finalmente para el sistema no autónomo el punto de equilibrio

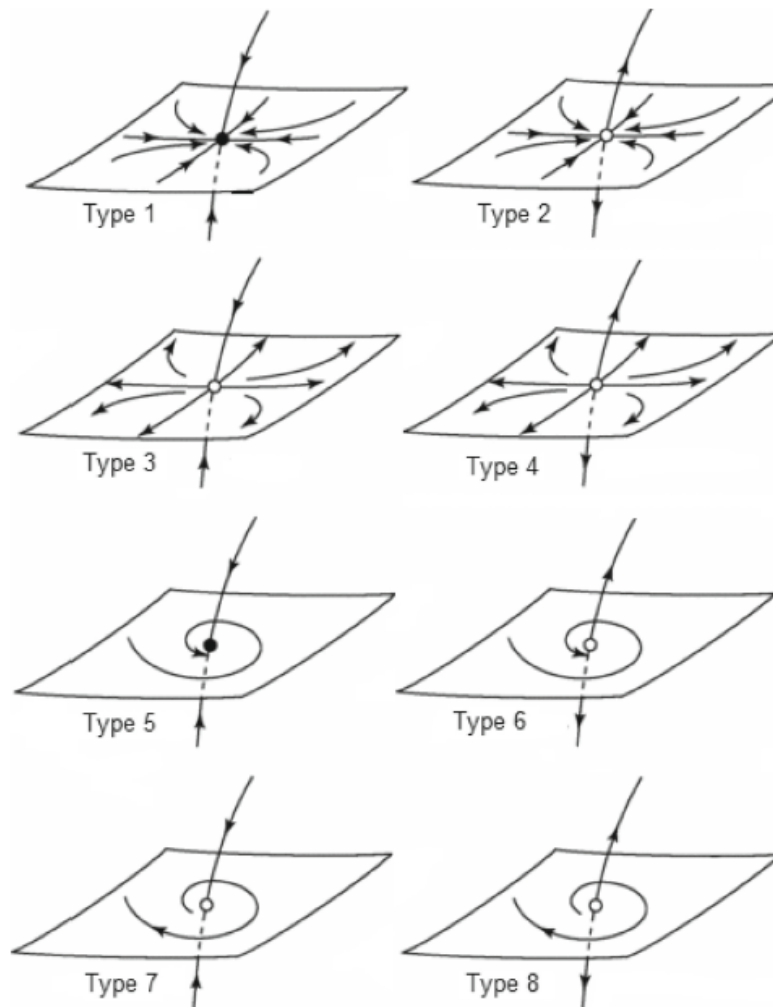


Figura 3.20: Tipos de puntos de equilibrio dependiendo de su categoría [45].

Tabla 3.2: Categorías de los punto de equilibrio en un sistema caótico en base a los eigenvalores hallados en un sistema de 3 dimensiones en base al trabajo [45].

Eigenvalores	Categoría
Tres negativos reales	Nodo índice-0
Dos negativos y un negativo todos reales	Punto silla índice-1
Dos positivos y un negativo todos reales	Punto silla índice-2
Tres positivos reales	Repulsor de índice-3
Un real negativo y un complejo conjugado con parte real negativa	Nodo espiral índice-0
Un real positivo y un complejo conjugado con parte real negativa	Punto silla espiral índice-1
Un real negativo y un complejo conjugado con parte real positiva	Punto silla espiral índice-2
Un real positivo y un complejo conjugado con parte real positiva	Repulsor espiral índice-3

Tabla 3.3: Eigenvalores hallados para el modelo con función tangente hiperbólica y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
$\begin{bmatrix} -1.780 & 0.275 & 0.139 \\ 1.780 & -0.275 & -0.139 \end{bmatrix}$	w_{12}	1.5	$-0.939 \quad 0.3613 \pm 2.317i$	Punto silla espiral de índice-2
		2.5	$-0.898 \quad 0.341 \pm 2.328i$	
		3.5	$-0.857 \quad 0.320 \pm 2.339i$	
		4.5	$-0.816 \quad 0.3003 \pm 2.350i$	
$[0, 0, 0]$	w_{12}	1.5	$2.000 \quad -1.000 \pm 1.224i$	Punto silla espiral de índice-1
		2.5	$2.000 \quad -1.000 \pm 1.581i$	
		3.5	$2.000 \quad -1.000 \pm 1.870i$	
		4.5	$2.000 \quad -1.000 \pm 2.121i$	
$\begin{bmatrix} -1.780 & 0.275 & 0.139 \\ 1.780 & -0.275 & -0.139 \end{bmatrix}$	w_{13}	6	$-0.9222 \quad 0.3529 \pm 2.3911i$	Punto silla espiral de índice-2
		7	$-0.9211 \quad 0.3524 \pm 2.3686i$	
		8	$-0.9199 \quad 0.3518 \pm 2.3460i$	
		9	$-0.9187 \quad 0.3512 \pm 2.3230i$	
		10	$-0.9174 \quad 0.3505 \pm 2.2999i$	
$[0, 0, 0]$	w_{13}	6	$1.0000 \quad -0.5000 \pm 1.6583i$	Punto silla espiral de índice-1
		7	$1.3788 \quad -0.6894 \pm 1.5575i$	
		8	$1.7100 \quad -0.8550 \pm 1.4809i$	
		9	$2.0000 \quad -1.0000 \pm 1.4142i$	
		10	$2.2583 \quad -1.1291 \pm 1.3509i$	

Tabla 3.4: Eigenvalores hallados para el modelo con función sigmoide y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
[-0.0002 -0.0152 0.0131]	w_{11}	7.77	1.7231 -0.0766 ± 1.8146i	Punto silla espiral de índice-1
		7.97	1.7373 -0.0587 ± 1.8073i	
		8.17	1.7519 -0.0410 ± 1.7998i	
[-0.8383, -0.7950, 5.5905]	w_{11}	7.77	-0.9709 0.5429 ± 1.3325i	Punto silla espiral de índice-2
		7.97	-0.9711 0.5641 ± 1.3310i	
		8.17	-0.9714 0.5853 ± 1.3291i	
[0.8494 0.7688 -5.6351]	w_{11}	7.77	-0.9722 0.5468 ± 1.3371i	Punto silla espiral de índice-2
		7.97	-0.9724 0.5679 ± 1.3358i	
		8.17	-0.9726 0.5890 ± 1.3341i	
[-0.0002 -0.0152 0.0131]	w_{22}	8.73	1.7321 -0.0686 ± 1.8100i	Punto silla espiral de índice-1
		8.78	1.7347 -0.0637 ± 1.8087i	
		8.83	1.7373 -0.0587 ± 1.8073i	
		8.88	1.7399 -0.0538 ± 1.8060i	
[-0.8383, -0.7950, 5.5905]	w_{22}	8.73	-0.9710 0.5533 ± 1.3300i	Punto silla espiral de índice-2
		8.78	-0.9711 0.5587 ± 1.3305i	
		8.83	-0.9711 0.5641 ± 1.3310i	
		8.88	-0.9712 0.5695 ± 1.3314i	
[0.8494 0.7688 -5.6351]	w_{22}	8.73	-0.9723 0.5570 ± 1.3350i	Punto silla espiral de índice-2
		8.78	-0.9723 0.5625 ± 1.3354i	
		8.83	-0.9724 0.5679 ± 1.3358i	
		8.88	-0.9725 0.5734 ± 1.3362i	

Tabla 3.5: Eigenvalores hallados para el modelo con función PWL y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
[0.7093 -0.7119 -1.2408] [-0.7093 0.7119 1.2408]	w_{11}	1.58	-1.0000 0.6900 ± 4.5345i	Punto silla espiral de índice-2
		1.68	-1.0000 0.7400 ± 4.5355i	
		1.78	-1.0000 0.7900 ± 4.5358i	
[0, 0, 0]	w_{11}	1.58	1.8973 -0.9087 ± 1.5553i	Punto silla espiral de índice-1
		1.68	1.8324 -0.8262 ± 1.4464i	
		1.78	1.7552 -0.7376 ± 1.3198i	
[0.7093 -0.7119 -1.2408] [-0.7093 0.7119 1.2408]	w_{33}	-0.2	-1.0000 0.7400 ± 4.5355i	Punto silla espiral de índice-2
		-0.3	-1.0000 0.7400 ± 4.5355i	
		-0.4	-1.0000 0.7400 ± 4.5355i	
[0, 0, 0]	w_{33}	-0.2	2.0477 -0.8838 ± 1.6532i	Punto silla espiral de índice-1
		-0.3	1.8324 -0.8262 ± 1.4464i	
		-0.4	1.5595 -0.7398 ± 1.1656i	

Tabla 3.6: Eigenvalores hallados para el modelo con función signo y su clasificación.

Puntos de equilibrio	Parámetro	Valor	Eigenvalores	Clasificación
$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$	w_{32}	3.5	$-1.0000 \quad 0.3230 \pm 0.9996i$	Punto silla espiral de índice-2
		4	$-1.0000 \quad 0.3230 \pm 1.1799i$	
		4.5	$-1.0000 \quad 0.3230 \pm 1.3360i$	
$[0, 0, 0]$	w_{32}	3.5	$6.3440 \quad -3.1720 \pm 5.5394i$	Punto silla espiral de índice-1
		4	$6.3137 \quad -3.1569 \pm 5.5585i$	
		4.5	$6.2834 \quad -3.1417 \pm 5.5777i$	
$\begin{bmatrix} 0.3306 & -0.3435 & 0.3581 \\ -0.3306 & 0.3435 & -0.3581 \end{bmatrix}$	w_{33}	2.5	$-0.8367 \quad 0.0208 \pm 1.3292i$	Punto silla espiral de índice-2
		3	$-0.8317 \quad 0.2388 \pm 1.0912i$	
		3.5	$-0.8260 \quad 0.4564 \pm 0.7170i$	
$[0, 0, 0]$	w_{33}	2.5	$6.0976 \quad -3.2988 \pm 5.5989i$	Punto silla espiral de índice-1
		3	$6.3137 \quad -3.1569 \pm 5.5585i$	
		3.5	$6.5413 \quad -3.0207 \pm 5.5131i$	

solo tiene sentido para $t = 0$ y para este caso los eigenvalores hallados fueron $0.075 \pm 0.9909i$, lo cual indica que es **foco inestable**.

Capítulo 4

Dinámica caótica

La dinámica de los sistemas no lineales puede presentar fenómenos más complejos en comparación con los sistemas lineales, una de estas complejidades es denominada caos. Desde su descubrimiento varias métricas para la identificación de caos en sistemas dinámicos han sido desarrolladas, las más comunes siendo los diagramas de bifurcación y exponentes de Lyapunov. En esta sección se utilizarán dichas métricas para identificar los distintos comportamientos dinámicos (no solo caóticos) que generan las redes neuronales estudiadas. Estas métricas dejan en evidencia *las ventajas que presentan las nuevas redes neuronales Hopfield propuestas* con respecto a las halladas en la literatura.

4.1. Diagramas de bifurcación

La bifurcación indica la división de una órbita en dos partes, en sistemas no lineales, este término suele ser utilizado para indicar un cambio cualitativo o cuantitativo de un sistema a partir de la variación de alguno de sus parámetros. Este parámetro es llamado como *parámetro de bifurcación*. Este comportamiento cualitativo o cuantitativo del sistema dependiente del parámetro de bifurcación es graficada en el llamado *diagrama de bifurcación*, este nos da información de la dinámica compleja que este presenta.

Gran parte de los sistemas dinámicos que representan algún fenómeno físico presentan algún tipo de periodicidad en su comportamiento, por ejemplo, la rotación y traslación de los planetas, un péndulo ideal sin fricción o un señal senoidal, todos estos fenómenos comparten necesitar solo un ciclo para repetir el mismo comportamiento y como tal se denominan de *periodo-1* o simplemente p-1, en otros casos sin embargo el comportamiento del sistema se repite tras 2 (p-2), 3 (p-3), etc., en la Figura 4.1 se graficó la respuesta de una neurona del modelo (3.5) para los cuales se registraron distintas periodicidades variando uno de sus parámetros, eventualmente si el valor de este parámetro se modifica de manera correcta es posible hallar otros tipo de

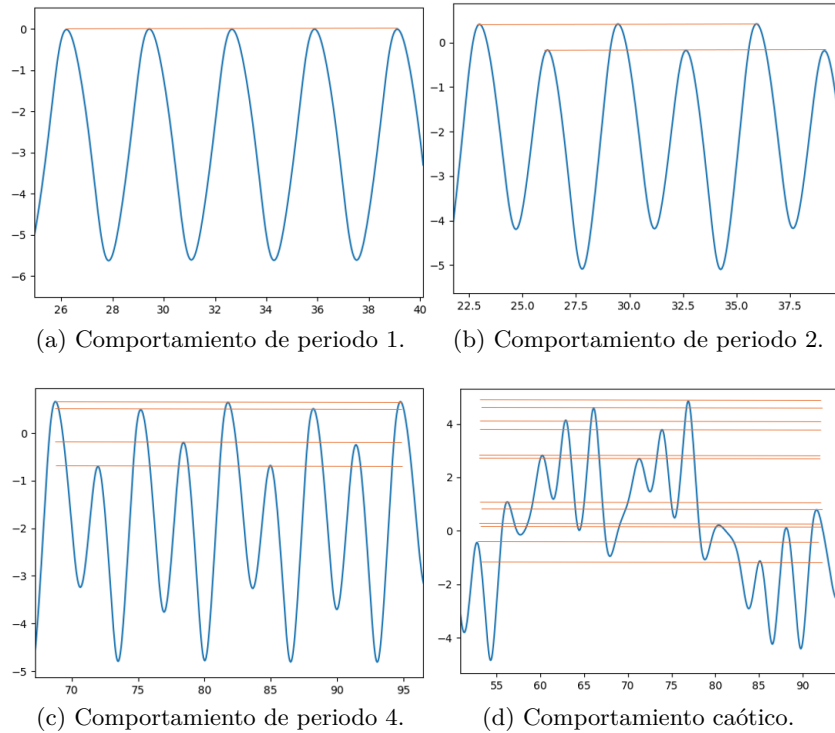


Figura 4.1: Diferentes comportamientos periódicos y caótico para el modelo 3D HNN con función de activación tangente hiperbólica.

periodicidad y eventualmente caos, esta dinámica es conocida como *ruta al caos*.

Para este trabajo se decidió tomar como cambio cualitativo al máximo local que toma una de las orbitas del sistema conforme se modifica alguno de los parámetros del sistema, esto requiere de un programa capaz simular el sistema con alguno de los métodos de integración, un detector de los máximos en la trayectoria con respecto al tiempo y un graficador, este proceso requiere de una gran potencia computacional, por lo que un lenguaje de programación veloz es recomendable, en este escenario C++ podría ser una elección idónea, sin embargo, hoy es considerado un lenguaje de mas bajo nivel además de no tener una gran cantidad de librerías, por lo tanto, se decidió desarrollar el programa en Python debido principalmente a su librería NumPy la cual nos permite utilizar un lenguaje de programación de alto nivel como Python e implementar operaciones directamente en C++.

4.1.1. Diagramas de bifurcación en Python

Para analizar el comportamiento dinámico de las redes neuronales Hopfield fue realizado un código en Python 3.12. El código requiere que primero se definan ciertos argumentos, estos son:

- `prmt`: Es el parámetro de bifurcación, debe estar descrito como un array del tipo Numpy. De preferencia utilizar el método `linspace(start, stop, num)` de Numpy que regresa un array con valores de "start" a "stop" de "num" cantidad de elementos.
- `funcion`: Función que describe el sistema dinámico en forma de un vector de NumPy, los argumentos en esta función son el tiempo y un vector NumPy con las variables de estado. Por ejemplo, para describir el modelo (3.5) se tiene:

```
def fct(x, t, a):
    return np.array([-x[0]+a*tg(x[1])-9*tg(x[2]),
                    -x[1]+tg(x[0])+3*tg(x[1])+3*tg(x[2]),
                    -x[2]-tg(x[0])-3*tg(x[1])])
```

- `initialCondition`: Condición inicial para la integración numérica del sistema en forma de matriz, se aceptan un conjunto de condiciones iniciales, por ejemplo, `initialCondition = [[0.1, 0, 0], [-0.1, 0, 0]]`, en este ejemplo se exploran dos condiciones iniciales distintas. El color de los diagramas de bifurcación cambia con respecto a que condición inicial se explora, es azul para la primer condición inicial, rojo para la segunda, verde para las demás.
- `paso`: Es el paso de integración numérica, esta definida como `1e-2` de manera predeterminada.
- `porcentaje`: Un sistemas dinámico requiere de cierto tiempo hasta "estabilizarse" al comportamiento que este mantendrá, por lo tanto, es necesario retirar esta sección pasajera del sistema, este argumento indica que porcentaje es considerado para iniciar a graficar los máximos locales del sistema, es decir, si se desea considerar la última tercera parte de las simulaciones realizadas se colocaría un `33` como argumento. El valor predeterminado es de `50`.
- `tf`: Tiempo final de las simulaciones realizadas, de manera predeterminada este valor es `100`.
- `eje`: Indica el eje que se desea analizar, por ejemplo, si en se tienen tres variables de estado (x, y, z) y se requiere en diagrama de bifurcación de y con respecto a un parámetro el valor del `eje=2`. De manera predeterminada es `1`.
- `rest`: La obtención de los diagramas de bifurcación requiere de una gran potencia computacional, esto genera un desgaste en el computador importante y generar un sobrecalentamiento, este argumento permite dar un descanso en segundos entre las iteraciones

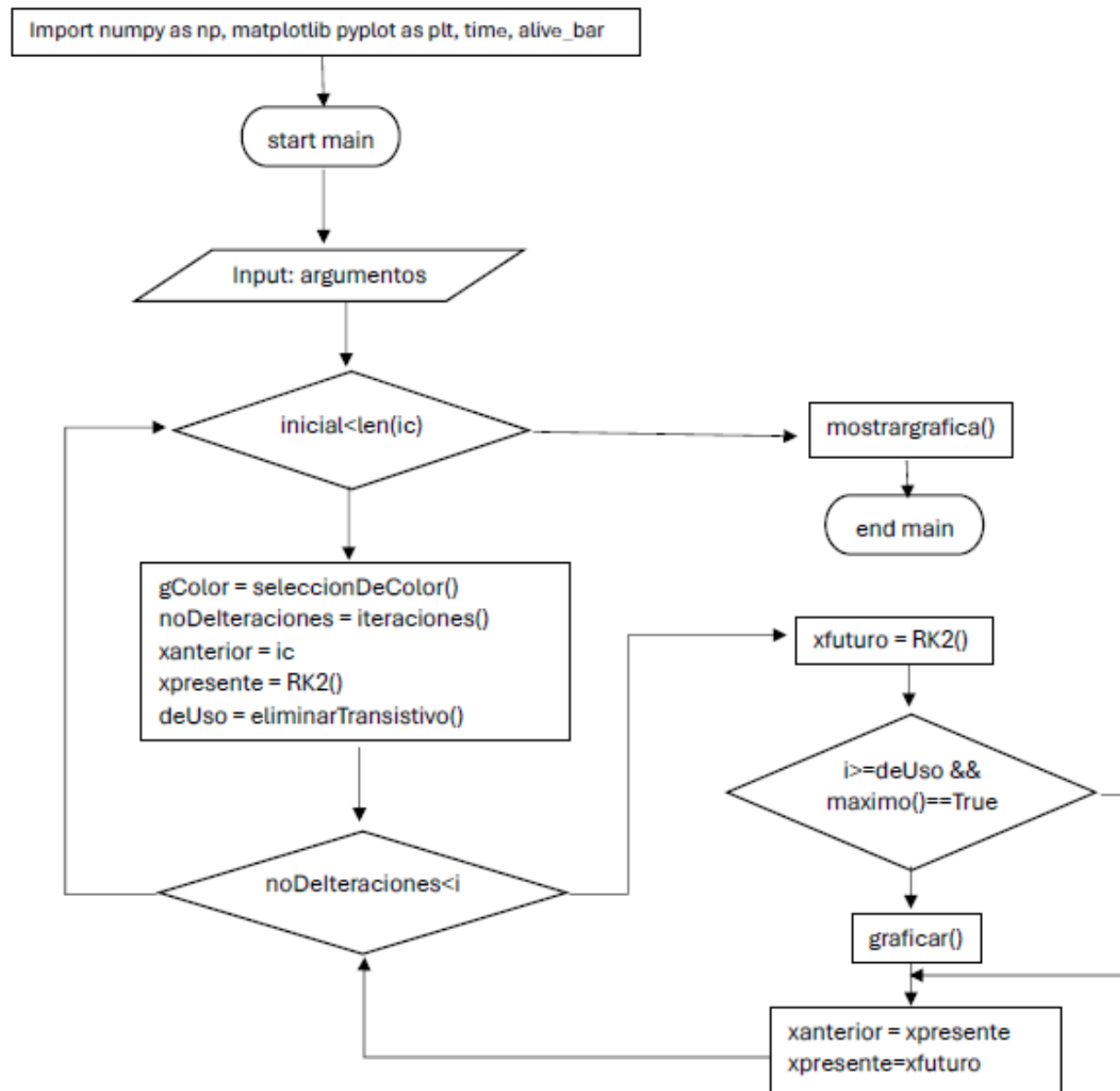


Figura 4.2: Diagrama de flujo para el desarrollo del código para obtener diagramas de bifurcación.

realizadas, por ejemplo, si se desea que se descansa un tiempo de 1ms entre iteraciones se colocaría como argumento $\text{rest} = 1\text{e-}3$.

Como base para este código considerando los argumentos antes descritos se realizó el diagrama de flujo de la Figura 4.2.

El código requiere de dos conceptos bases para su funcionamiento, la detección la simulación del sistema dinámico y la detección del máximo local. Es necesario también considerar la potencia computacional requerida, dado que es importante tomar una gran cantidad de divisiones, en donde es necesario integrar el sistema para todos estos valores del parámetro de interés, hallar y graficar el máximo. Es por esto que se utiliza un método de integración que consume baja cantidad de recursos como lo es método de integración Runge-Kutta 2 y obtener los máximos conforme el sistema evoluciona y no como un proceso posterior a la integración numérica. Para detectar el máximo de una curva el programa almacena tres estados distintos de una curva, el estado anterior x^{**} , presente x^* y futuro x , se detecta un máximo local si se cumple

$$x^* - x^{**} \geq 0 \quad \wedge \quad x - x^* < 0, \quad (4.1)$$

el programa es capaz entonces de detectar las curvas que cumplen esta condición y graficar el valor del máximo hallado.

4.1.2. Diagramas de bifurcación del modelo con función de activación tan-gente hiperbólica

El resultado de los diagramas de bifurcación para el modelo 3.5 es presentado en la Figura 4.4, en esta diversos fenómenos son denotados. Para iniciar, el sistema presenta el fenómeno de *bi-estabilidad*, esto es, con diferentes condiciones iniciales puede tener un comportamiento u otro completamente distinto, más específicamente, en azul se utiliza la condición inicial $[0.1, 0, 0]$ y en rojo la condición inicial $[-0.1, 0, 0]$. De la figura también se observan las distintas periodicidades que el sistema presenta, por ejemplo, en $w_{12} = -5$ se tiene una periodicidad 2, en $w_{12} = -3.7$ se tiene una periodicidad de 5, entre otras. En algunos casos las regiones para las cuales se presenta algún tipo de periodicidad es muy pequeña, por ejemplo, con un paso de integración de 0.0025 se encontró un periodo tres en la región $-2.296 < w_{12} < -2.267$, mientras que para un paso de integración de 0.001 se encontró esta región en $-2.247 < w_{12} < -2.225$, por lo que para regiones tan pequeñas es difícil determinar su periodicidad exacta. Además de la periodicidad del sistema, los diagramas de bifurcación nos ofrecen las zonas en donde el sistema se siente atraído por un solo atractor o múltiples, por ejemplo, variando el parámetro w_{12} en la zona $4.05 < w_{12} < 4.34$ el sistema solo muestra un scroll y se evidencia en el diagrama de bifurcación y la Figura 4.3. Por cuál de los dos scrolls el sistema se siente atraído es dependiente de la condición inicial del sistema y del valor del parámetro de bifurcación, en la Figura 4.3 se

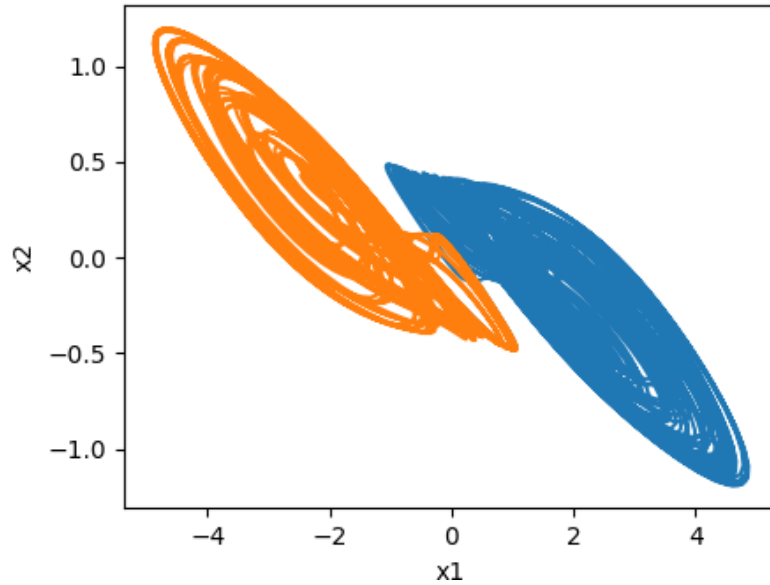


Figura 4.3: Comportamiento del sistema con un solo atractor usando $w_{12} = 4.05$ en azul y $w_{12} = 4.06$ en naranja.

simula el sistema con $w_{12} = 4.05$ en azul y $w_{12} = 4.06$ en naranja, en donde se evidencia esta descripción.

4.1.3. Diagramas de bifurcación del modelo con función de activación sigmoide

Por otro lado, la Figura 4.5 presenta el diagrama de bifurcación variando los parámetros a) w_{11} y b) w_{22} del modelo 3.15, a diferencia del modelo anterior, este sistema no presenta bi-estabilidad, así como también tiene una notoria reducción en espacio en donde el sistema presenta caos, además de presentar una menor cantidad de fenómenos en general. Nótese que existen tres zonas con caos en el sistema variando w_{11} , para la región contenida en $7.968 < w_{11} < 7.974$ existe un doble scroll, mientras que para las otras zonas el sistema solo es atraído por un atractor. Esta característica es compartida por el segundo parámetro w_{22} . Su limitada región caótica dificulta su implementación en circuitos electrónicos.

4.1.4. Diagramas de bifurcación del modelo con función de activación ReLU

Para el modelo con función de activación ReLU los parámetros de bifurcación no son los pesos sino la amplitud y frecuencia de la señal externa senoidal insertada en la primera neurona. Tras su análisis se encontró el fenómeno de bi-estabilidad en el sistema así como una amplia gama de periodicidades en el sistema. Por un lado, al variar la amplitud de la señal externa se encuentran distintas zonas periódicas y caóticas con amplitud proporcional a la señal externa.

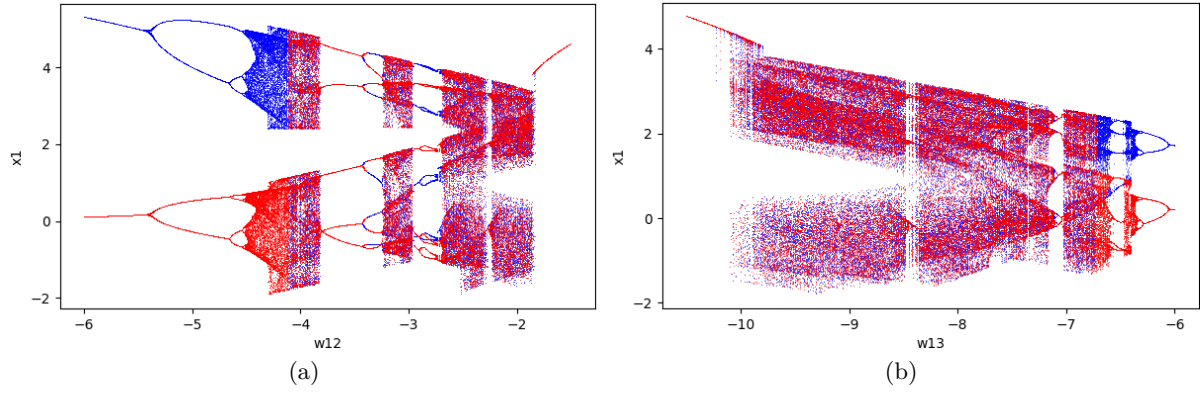


Figura 4.4: Diagramas de bifurcación para el modelo 3.5 con dos condiciones iniciales, $[0.1, 0, 0]$ y $[-0.1, 0, 0]$ en azul y rojo respectivamente, variando el parámetro en a) $\beta = w_{12}$ de 1.5 a 6, y b) el parámetro $\beta = w_{13}$ se varía de 5 a 10.

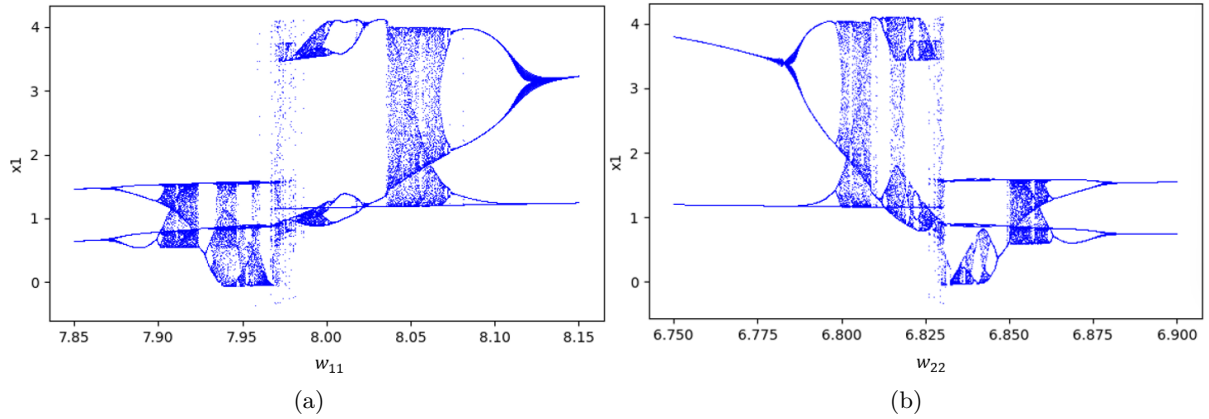


Figura 4.5: Diagramas de bifurcación para el modelo con función de activación sigmoide.

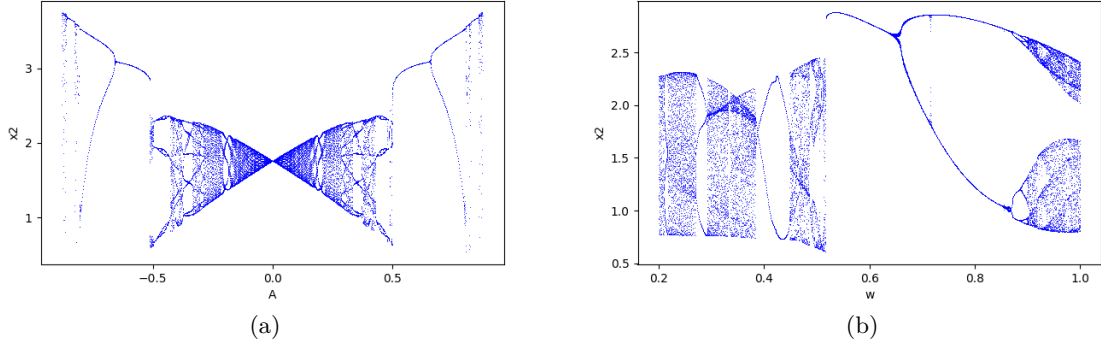


Figura 4.6: Diagramas de bifurcación para el modelo con función de activación ReLU.

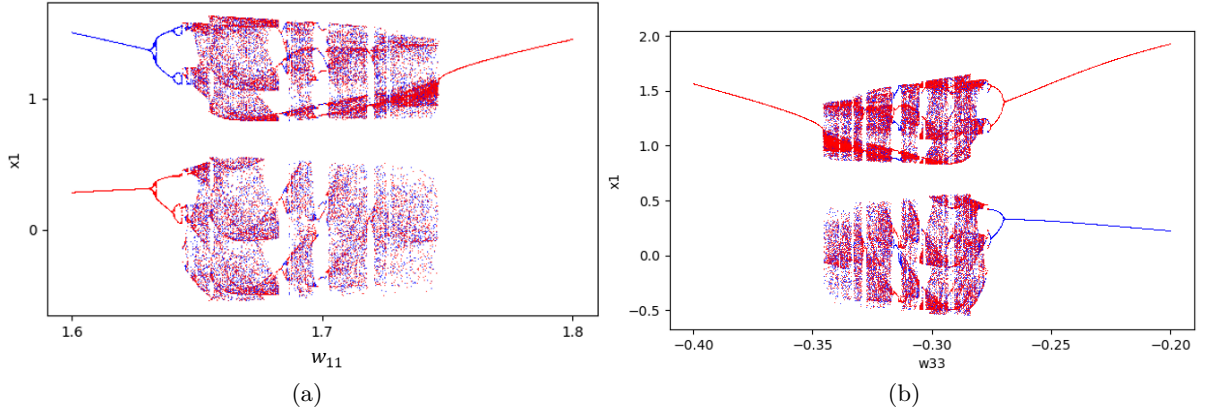


Figura 4.7: Diagramas de bifurcación para el modelo con función de activación PWL.

Por otro lado, al variar la frecuencia de la señal externa se presenta una gran variedad de periodicidades así como regiones caóticas, para una amplia cantidad de frecuencias existe en apariencia caos, son en realidad el resultado de la frecuencia tan alta de la señal externa aplicada al sistema, para este caso, las zonas caóticas son identificadas de manera apropiada por los exponentes de Lyapunov hallados en la sección 4.2.

4.1.5. Diagramas de bifurcación del modelo con función de activación PWL

Para el caso del modelo con función de activación $\text{satlin}(x)$ se detectó de nueva manera bi-estabilidad en el sistema y se evidencia en los diagramas de bifurcación de la Figura 4.7, de color azul se utilizó la condición inicial $(-0.1, 0, 0)$ y en rojo $(0.1, 0, 0)$, notemos de los diagramas de bifurcación que las regiones de periodicidad son muy pequeñas.

4.1.6. Diagramas de bifurcación del modelo con función de activación signo

Finalmente, para el modelo propuesto con función de activación signo se analizaron dos parámetros w_{32} y w_{33} , el diagrama de bifurcación de w_{32} es presentado en la Figura 4.8 a), en ella se introdujeron dos condiciones iniciales $(-0.1, 0, 0)$ y $(0.1, 0, 0)$ de color azul y rojo respectivamente, de aquí se evidencia que para valores de w_{32} mayores a 6.58 el sistema solo se siente atraído por uno de sus atractores. Por su parte, la variación del parámetro w_{33} con diagrama de bifurcación en la Figura 4.8 b) presenta una menor cantidad de fenómenos dinámicos en un menor rango. Un aspecto que resulta interesante sobre este modelo es el peso w_{12} , fue hallado que **el sistema presenta caos para todo** $w_{12} \in \mathbb{R}_+$, este es el comportamiento observado en la Figura 4.8c y en la Figura 4.8d, donde también queda en evidencia que la modificación de este parámetro solo aumenta la magnitud del sistema.

4.1.7. Diagramas de bifurcación del modelo con 4 neuronas

En el modelo de 4 neuronas se eligen como parámetros de análisis los pesos w_{22} y w_{44} , esto en base al trabajo responsable a su propuesta [22]. De este trabajo se sabe que la base de atracción del sistema es variante y el comportamiento del sistema cambia en base a la condición inicial del sistema. La Figura 4.9a donde $w_{44} = 170$ y la Figura 4.9b donde $w_{22} = 2$, muestran el respectivo diagrama de bifurcación del sistema, en este se observan algunas zonas periodicas, entre ellas periodo 3, lo cual implica caos. Para obtener este diagrama de bifurcación se utiliza la condición inicial $(0.01, -0.02, 0, 0)$. Para el caso específico de $w_{22} = 2$ y $w_{44} = 170$ se observa un sistema denso y sera estos valores para los cuales se diseñara al red neuronal Hopfield en circuitos electrónicos.

4.2. Exponentes de Lyapunov

Otra de las métricas más relevantes y usadas para la identificación de caos son los exponentes de Lyapunov (LE), su uso nace de la propiedad más característica de los sistemas caóticos, su extrema sensibilidad a las condiciones iniciales, en este sentido, los exponentes de Lyapunov miden la divergencia exponencial entre dos orbitas infinitesimalmente cercanas. Para su definición imaginemos un estado $\mathbf{x}(t)$ de un sistema dinámico caótico que evoluciona en el tiempo. En el momento $t = 0$ creamos otra trayectoria perturbando la original con una pequeña perturbación $\mathbf{y}$, con $\delta = \|\mathbf{y}\|$. Si la dinámica es caótica, el original y las trayectorias perturbadas se separan cada vez más en el tiempo a medida que evolucionan. Para pequeños perturbaciones, esta separación está ocurriendo aproximadamente exponencialmente rápido. Después en algún

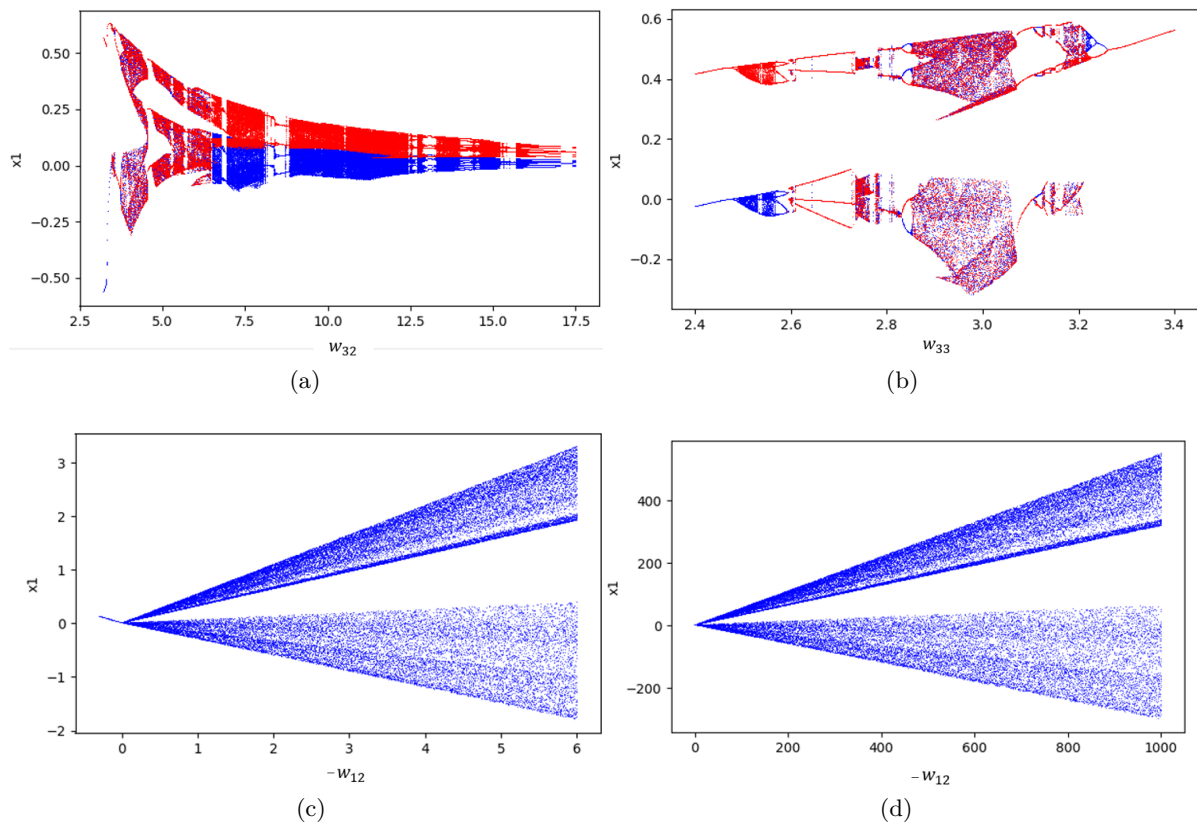


Figura 4.8: Diagramas de bifurcación para el modelo con función de activación signo.

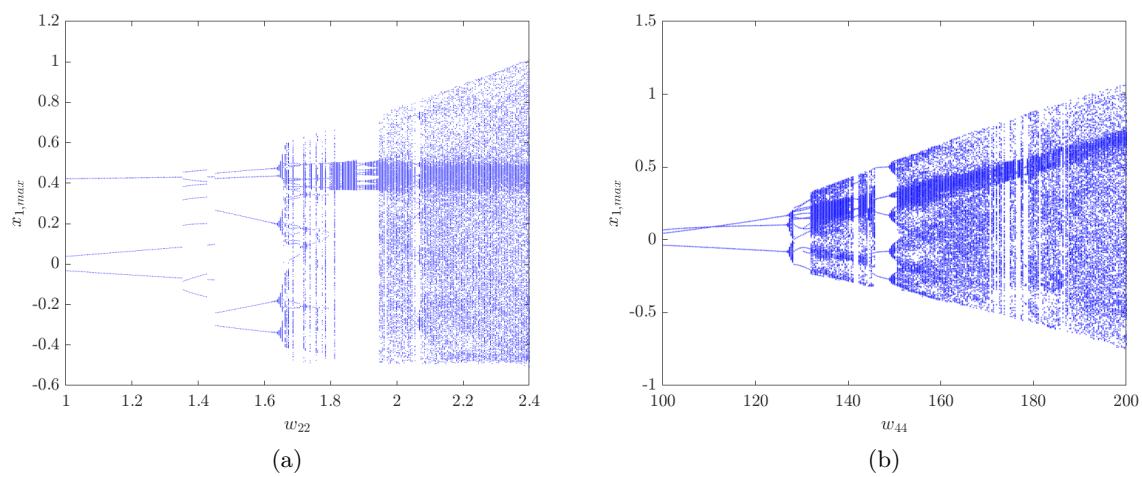


Figura 4.9: Diagramas de bifurcación para el modelo de 4 dimensiones.

momento t las dos trayectorias tendrán una distancia

$$\delta(t) \approx \delta_0 e^{\lambda t}. \quad (4.2)$$

La cantidad λ se llama el mayor exponente de Lyapunov y cuantifica la divergencia exponencial de trayectorias cercanas [47]. El exponente de Lyapunov da comportamientos cualitativos del sistema dinámico muy importantes, para empezar, si $\lambda > 0$ indica que pequeñas divergencias crecen exponencialmente conforme el sistema evoluciona, mientras que si $\lambda < 0$ indica lo contrario. Para sistemas dinámicos de tres dimensiones caóticos es sabido que se deben obtener un exponente positivo, un exponente negativo y un exponente cero, además si se está trabajando con un sistema disipativo, se espera que el valor del exponente negativo sea mayor al negativo. La obtención de los exponentes de Lyapunov resulta en una problemática computacional, las librerías de diversos lenguajes de programación actuales, sin embargo, nos permiten obtener los exponentes de Lyapunov con relativa sencillez, en este trabajo se utilizó el lenguaje JuliaProgramming y su librería JuliaDynamics, debido a que nos ofrece múltiples funciones con relación a sistemas dinámicos y caóticos, en específico la función `lyapunovspectrum()` obtiene los exponentes de Lyapunov de un sistema dinámico.

El número de exponentes de Lyapunov obtenidos depende de la dimensión del modelo dinámico, es por esta razón que con todos los modelo estudiados hasta ahora se obtendrán tres exponentes de Lyapunov con excepción del modelo con función $\text{ReLU}(x)$, para seguir la misma regla para la identificación de caos es posible expandir el sistema no autónomo a uno autónomo si consideramos $x_3 = t$, y por lo tanto es sistema se convierte

$$\begin{aligned} \dot{x}_1 &= -1.15x_1 + 1.1\text{ReLU}(x_1) - 1.05\text{ReLU}(x_2) - 0.5\text{sen}(0.52x_3) + 0.8 \\ \dot{x}_2 &= -0.55x_2 + 0.95\text{ReLU}(x_1) + 0.75\text{ReLU}(x_2) - 0.75 \\ \dot{x}_3 &= 1 \end{aligned} \quad , \quad (4.3)$$

Los exponentes de Lyapunov dan una indicación del comportamiento del sistema dinámico en relación al signo que estos poseen, la Tabla 4.1 resume el comportamiento esperado del atractor para un sistema dinámico de tres y cuatro dimensiones dado el signo de los exponentes de Lyapunov [32].

Los resultados obtenidos para los LE variando los parámetros estudiados del modelo con función tangente hiperbólica se muestran en la Figura 4.10, se analiza el mismo spectrum de valores estudiados con los diagramas de bifurcación con los cuales existe una evidente similitud en los resultados. Posteriormente en la Figura 4.11 se desarrolló un mapa de calor con el exponente máximo de Lyapunov (LLE) variando los parámetros estudiados w_{12} y w_{13} , en blanco están las zonas no caóticas, en amarillo están marcadas las zonas donde el LLE es relativamente pequeño, estas zonas se marcan distinto considerando que el valor positivo del

Signo de los LE's	Tipo de atractor
$(-, -, -)$	Punto fijo
$(0, -, -)$	Ciclo limite
$(0, 0, -)$	Cuasi-periódico
$(+, 0, -)$	Caótico
$(-, -, -, -)$	Punto fijo
$(0, -, -, -)$	Ciclo limite
$(0, 0, -, -)$	Torus 2 (Cuasi-periódico)
$(0, 0, 0, -)$	Torus 3
$(+, 0, 0, -)$	Caótico
$(+, +, 0, -)$	Hipercaótico

Tabla 4.1: Exponentes de Lyapunov y su relación con los tipos de atractores en un sistema dinámico de tres y cuatro dimensiones.

exponente podría tratarse de un ligero error numérico, mientras que en las zonas rojas y negras se afirma con seguridad la existencia de zonas caóticas. Comparado con los modelos hallados en la literatura, la zona caótica es mayor, esto se verá en los mapas de calor posteriores, en este sentido, este es un modelo cuya implementación electrónica es más sencilla debido al mayor rango de valores posibles, así como también tiene una utilidad en aplicaciones criptográficas en donde se busca variar los parámetros del modelo de manera constante y por lo tanto se requiere de modelos con una amplia zona caótica.

El mismo procedimiento para la obtención de los exponentes de Lyapunov del modelo con función tangente hiperbólica fue realizado con los demás modelos, algunas observaciones con los resultados obtenidos son los siguientes. Para el modelo con función de activación sigmoide se tienen la Figura 4.12 para los LE, mientras que la Figura 4.13 ofrece el mapa de calor con los LLE, de este último queda en evidencia su dificultad para una implementación en dispositivos electrónicos debido a la pequeña zona con comportamiento caótico del sistema. En las Figuras 4.14 y 4.15 se exploran los LE y LLE del modelo 3.21, notemos la simetría con respecto al parámetro de amplitud de la señal externa además de un mayor rango de parámetros para los cuales el sistema presenta caos con respecto al sistema anterior, los exponentes máximos de Lyapunov sin embargo, son los pequeños al sistema propuesto. Para el modelo con función de activación PWL se tiene la Figura 4.16 con los LE's se observa un comportamiento frecuente con funciones de activación no completamente diferenciables, como fue mencionado, para que un sistema sea considerado caótico los LE deben ser uno positivo, otro cero y otro negativo, los resultados entonces no concuerdan con esta descripción, esto es debido a los puntos no diferenciables, considerando este hecho el modelo sigue siendo considerado como caos [48], el mapa de calor correspondiente (Figura 4.17) muestra una zona caótica relativamente pequeña pero mayor a la hallada con la función sigmoide. Finalmente, para el modelo con función signo

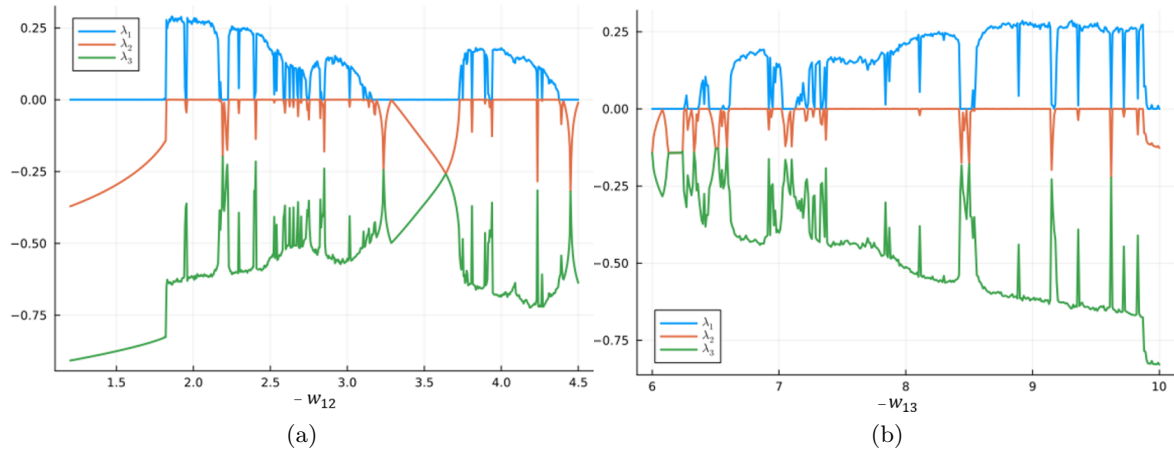


Figura 4.10: Signo de los exponentes de Lyapunov variando parametros de bifurcación a) w_{12} y b) w_{13} del modelo con función de activación tangente hiperbólica.

los resultados para LE Figura 4.18 y LLE Figura 4.19 se encontró una gran área con zonas caóticas, considerando incluso que la variación del parámetro w_{12} no afecta el comportamiento cualitativo del sistema se tiene un volumen infinito con las posibilidades para la elección de los parámetros del modelo. En general podemos concluir que los modelos propuestos ofrecen una implementación electrónica más sencilla y estable con respecto a los modelos hallados en la literatura.

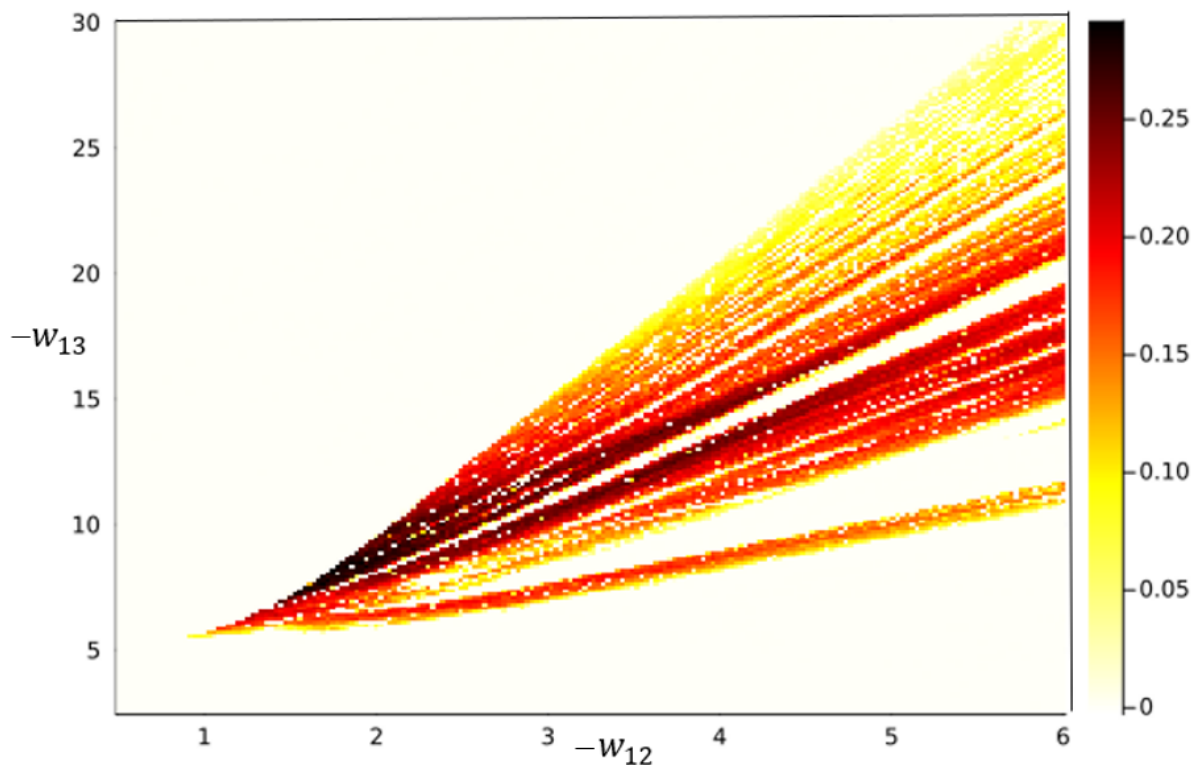


Figura 4.11: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{12} y w_{13} del modelo de HNN's con función de activación tangente hiperbólica.

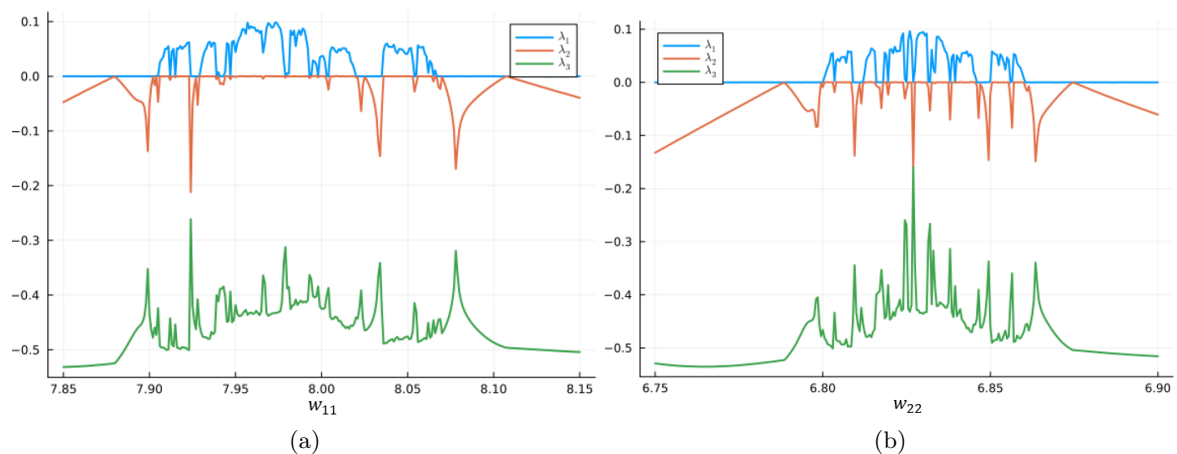


Figura 4.12: Exponentes de Lyapunov variando parametros de bifurcación a) w_{11} y b) w_{22} del modelo con función de activación sigmoide.

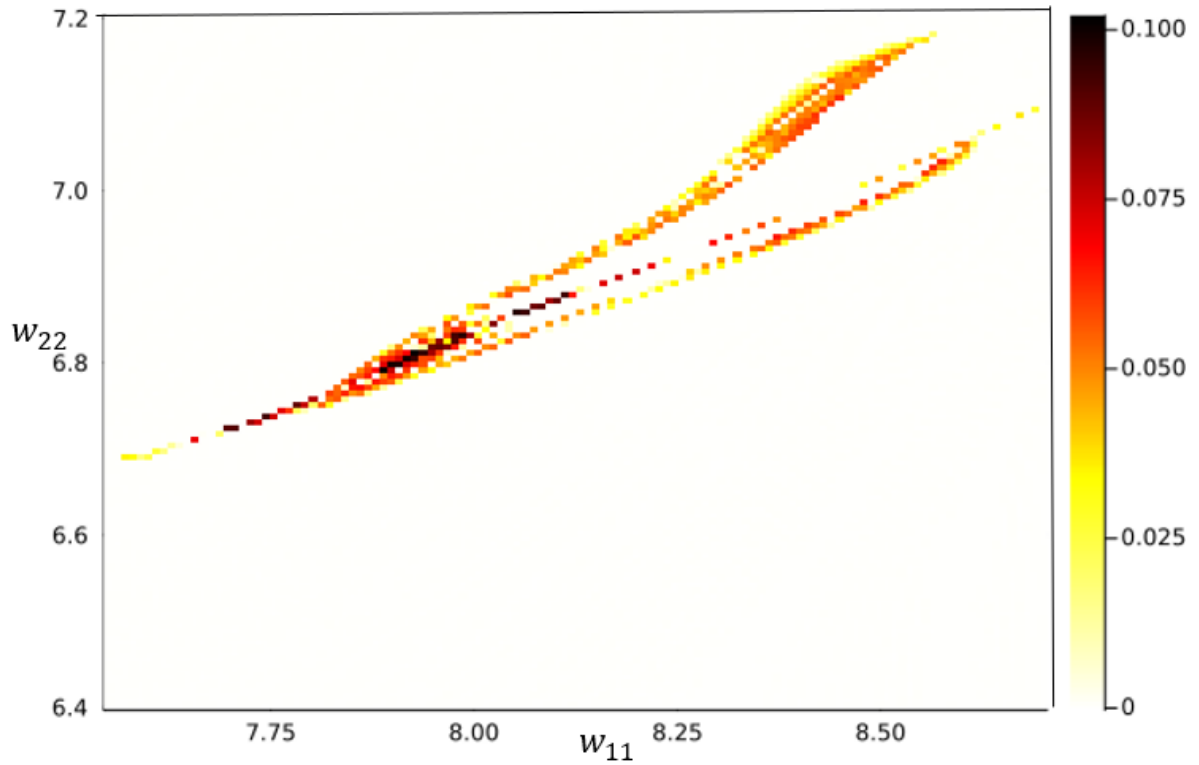


Figura 4.13: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{11} y w_{22} del modelo de HNN's con función de activación sigmoide.

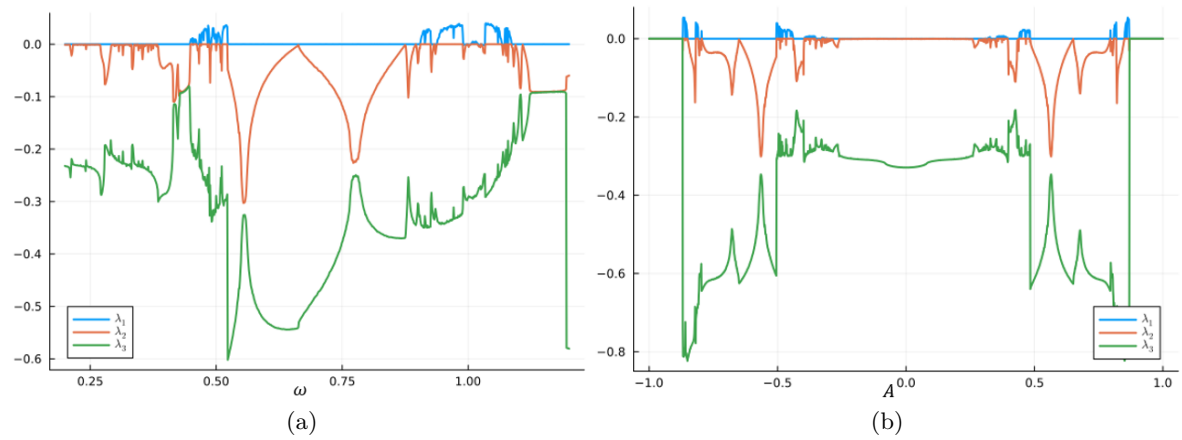


Figura 4.14: Exponentes de Lyapunov variando parametros de bifurcación a) frecuencia de la señal externa senoidal ω y b) amplitud de la señal externa senoidal A del modelo con función de activación ReLU.

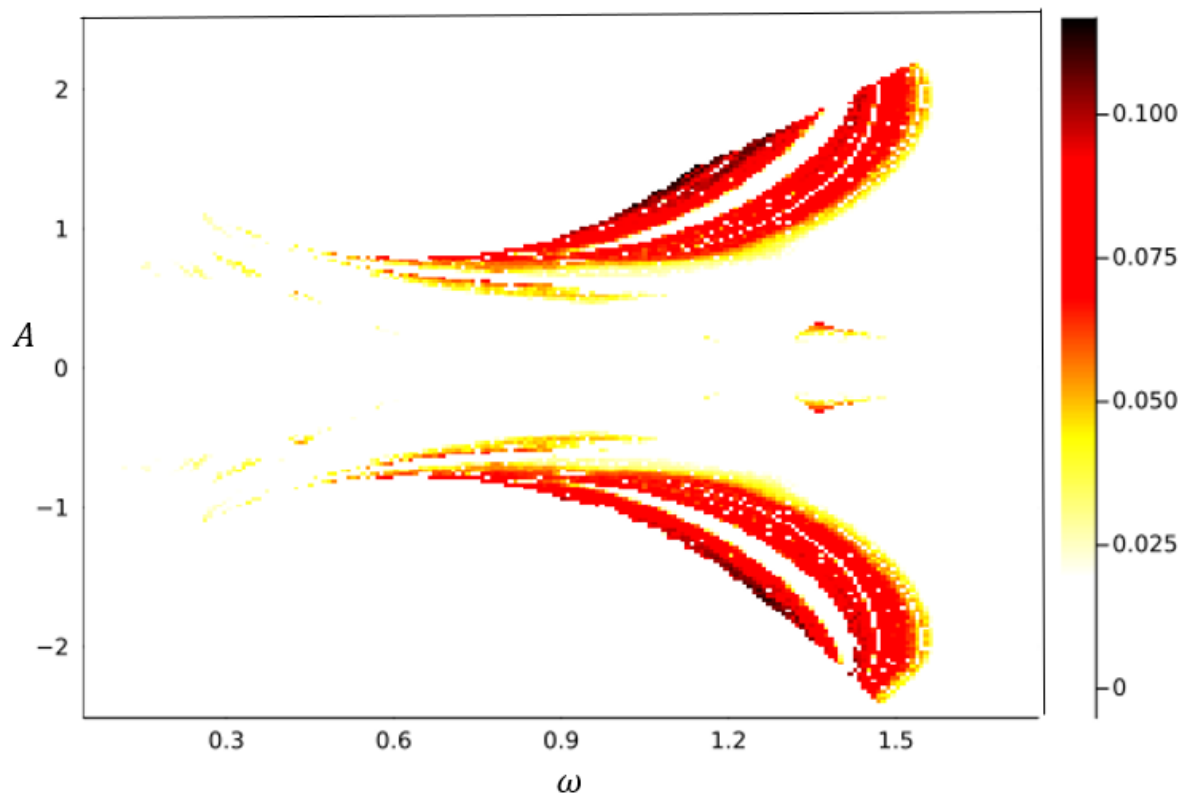


Figura 4.15: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros ω y A del modelo de HNN's con función de activación ReLU.

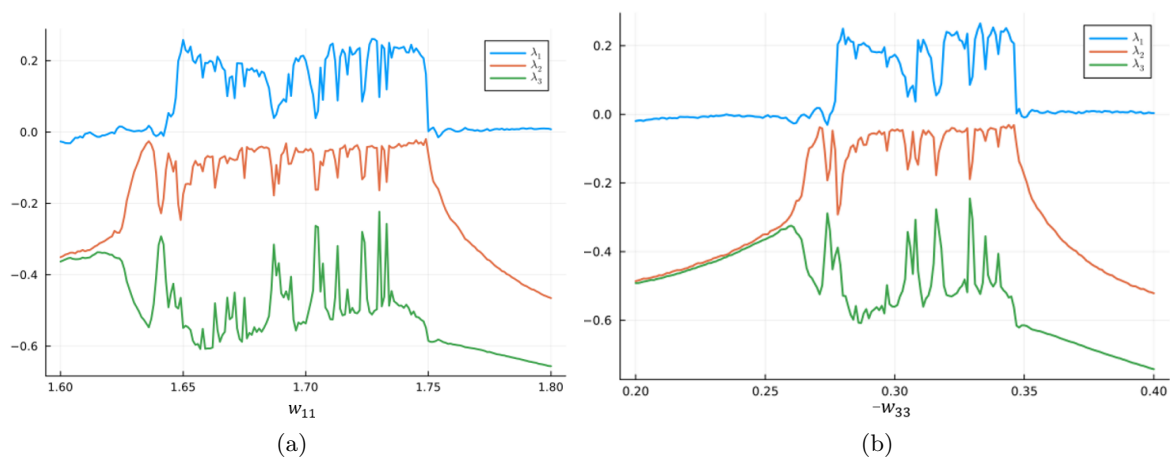


Figura 4.16: Exponentes de Lyapunov variando parámetros de bifurcación a) w_{11} y c) w_{33} del modelo con función de activación PWL.

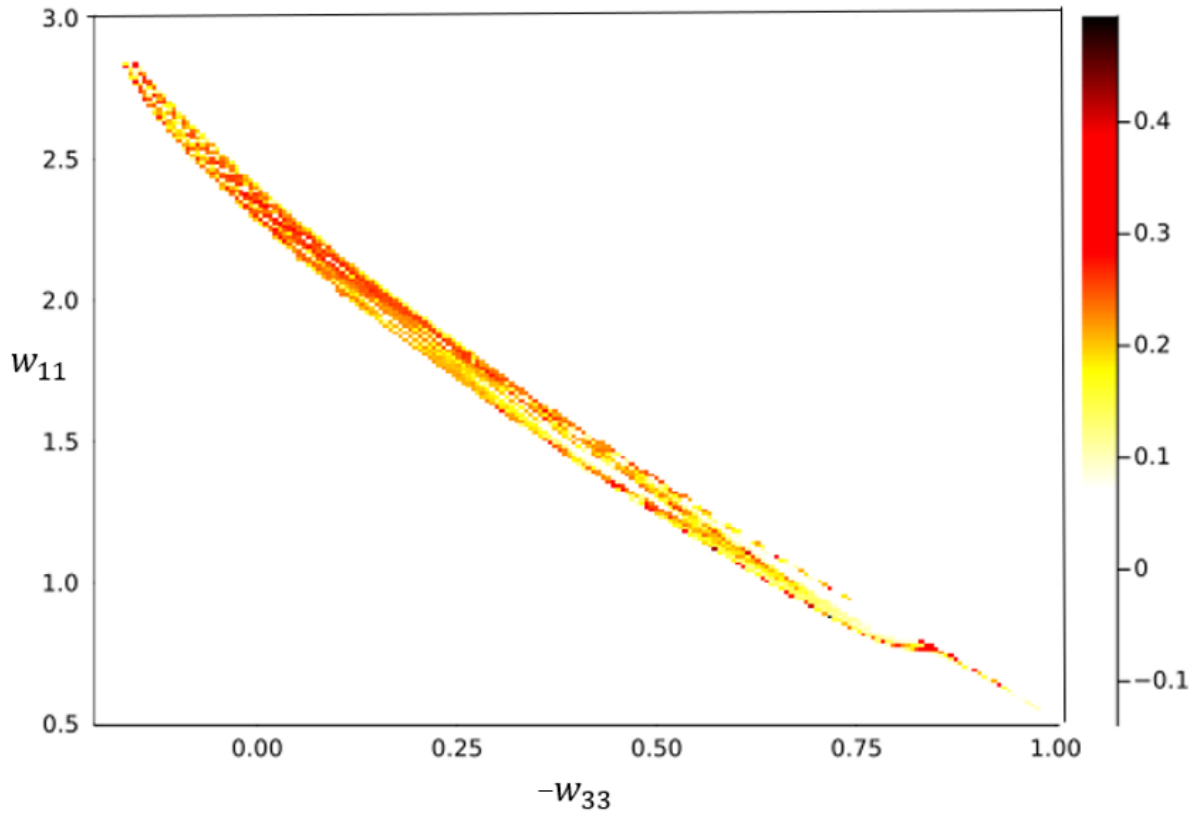


Figura 4.17: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{11} y w_{33} del modelo de HNN's con función de activación PWL.

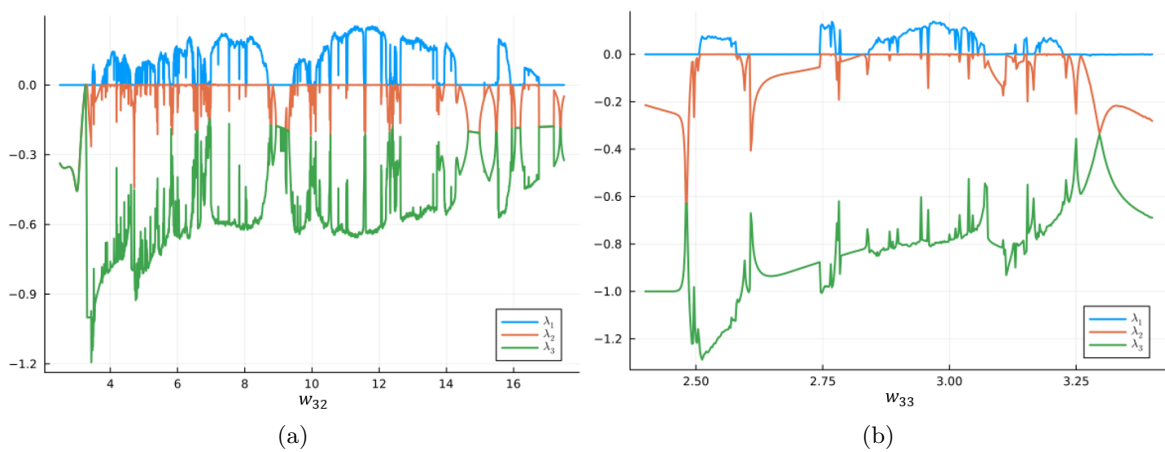


Figura 4.18: Exponentes de Lyapunov variando parámetros de bifurcación a) w_{32} y b) w_{33} del modelo con función de activación signo.

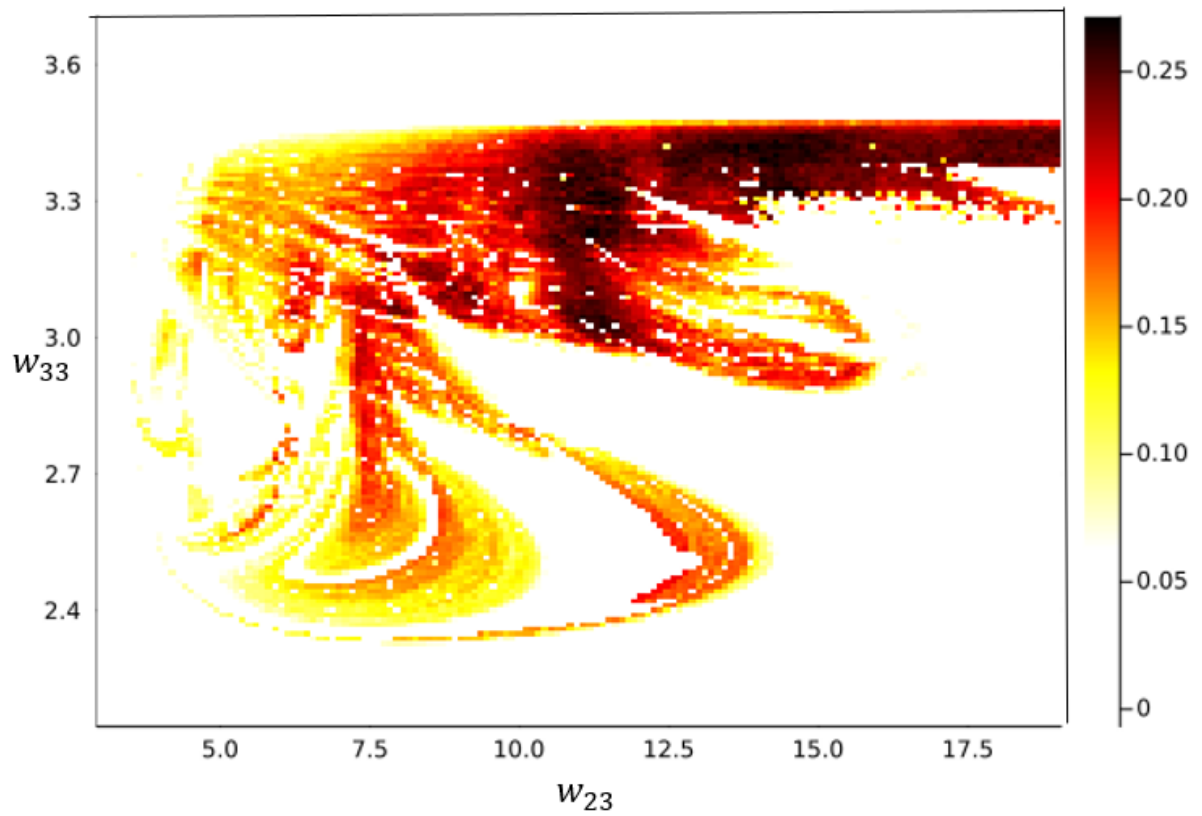


Figura 4.19: Mapa de calor con los exponentes máximos de Lyapunov variando los parámetros w_{32} y w_{33} del modelo de HNN's con función de activación signo.

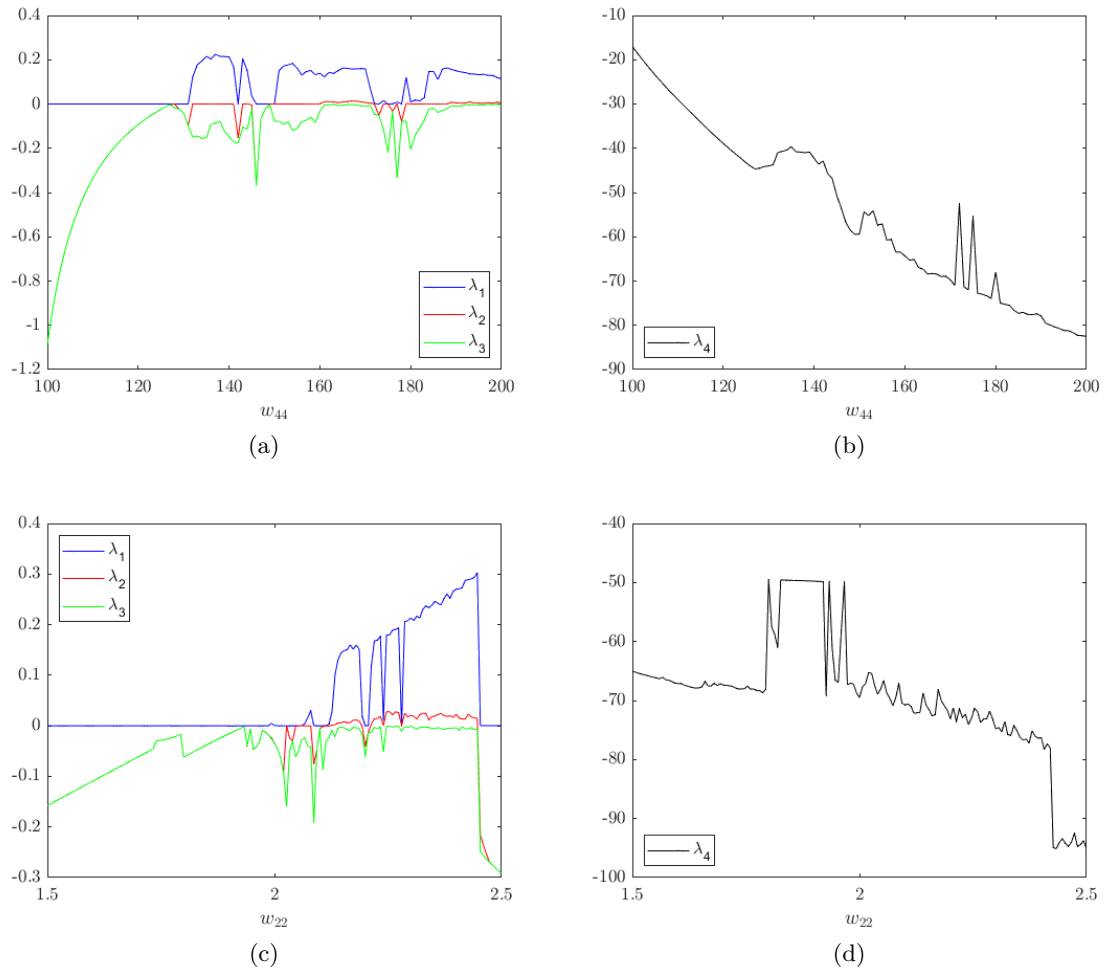


Figura 4.20: Exponentes de Lyapunov variando parámetros de bifurcación a), b) w_{44} y c), d) w_{22} del modelo HNN de 4 neuronas.

Capítulo 5

Implementación de las redes neuronales Hopfield

5.1. Desarrollo de funciones de activación

Las funciones de activación pueden ser descritas en base a circuitos electrónicos o, para nuestro caso, en base a bloques de programación para la tarjeta programable analógica. A día en el que se está redactando este trabajo la distribuidora Anadigm proporciona 54 bloques de programación en los que se encuentran ganancias (programables y saturadas), comparadores, señales periódicas, filtros (pasa-bajas, pasa-altas, pasa-banda y rechaza-banda), integradores, entre otros. A partir de algunos de estos bloques se describió el comportamiento de las funciones de activación.

5.1.1. Implementación de la función signo

La función signo puede ser vista como un comparador el cual identifica entre valores positivos y negativos, es por esto que se utilizó el bloque **comparator** para emular este comportamiento, en la Figura 5.1 se tienen los bloques utilizados para la implementación de la función signo, el primer bloque es un comparador con histéresis nula sin voltaje de referencia mientras que el segundo es un inversor de ganancia 0.5, la expresión equivalente al bloque comparador es

$$V_{out} = \begin{cases} -2, & \text{si } V_{in} > V_{Hyst} \\ 2, & \text{si } V_{in} < -V_{Hyst} \end{cases}, \quad (5.1)$$

mientras que el voltaje de salida se mantendrá sin cambio si alguna de estas condiciones no se cumplen. Multiplicando por una ganancia de $-\frac{1}{2}$ y fijando a $V_{Hyst} = 0$ se obtiene el comportamiento de la función signo.

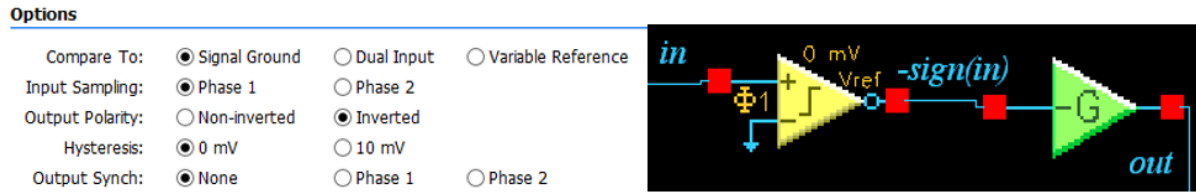
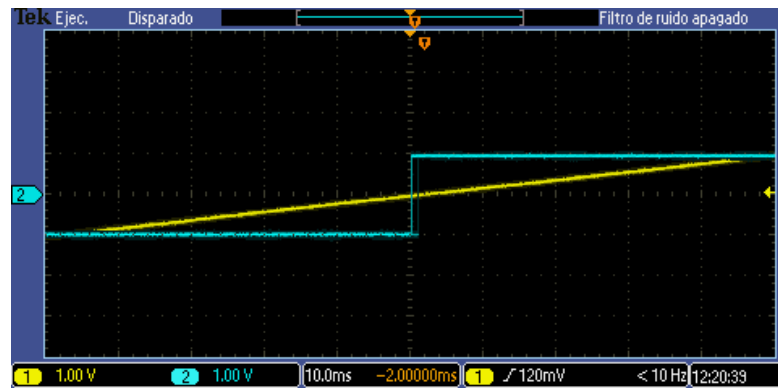
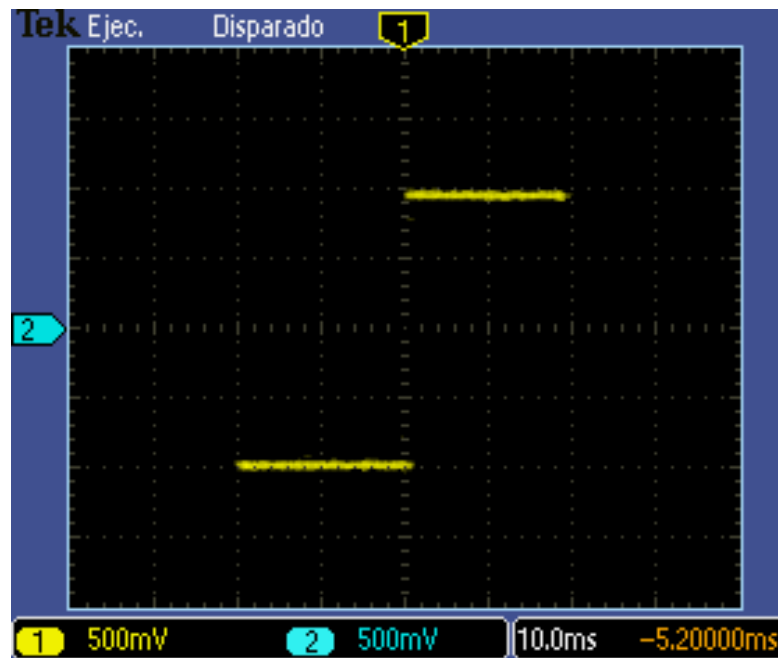


Figura 5.1: Configuración y programación en bloques de la función signo.



(a)



(b)

Figura 5.2: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida.

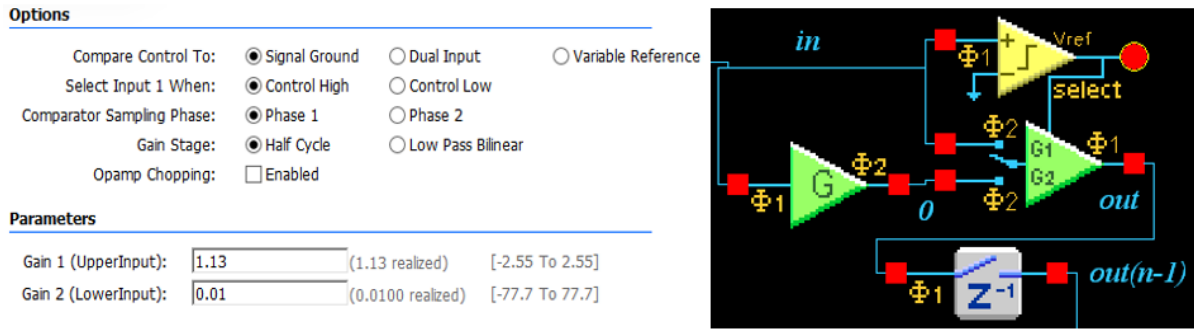


Figura 5.3: Configuración y programación en bloques del comparador analógico.

5.1.2. Implementación de la función ReLu

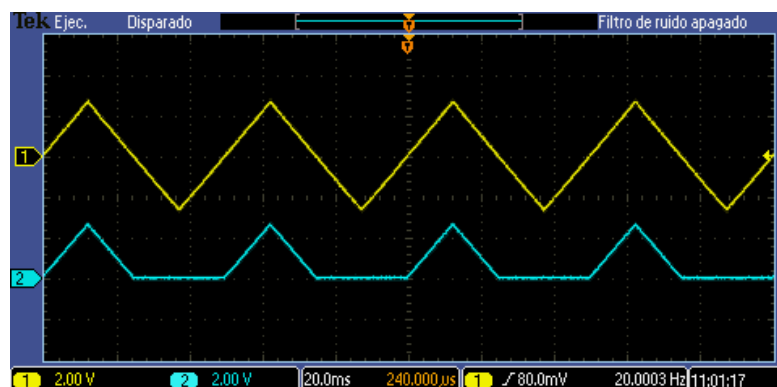
La función $\text{ReLu}(x)$ es cero cuando la entrada es menor o igual a cero, mientras que iguala la entrada cuando esta es mayor a cero, para este comportamiento entonces es necesario un comparador con respecto a cero que active y desactive la entrada, el bloque que permite este comportamiento es **Gain Stage with Switchable Inputs**, este es un bloque con ganancia dependiente a un comparador, de esta manera es posible conectar el voltaje V_{in} a una de las entradas de la ganancia intercambiable y al comparador, mientras que a la otro entrada se conecta una ganancia muy pequeña, además, configuración de la ganancia intercambiable para esta entrada también se configura muy pequeña, en específico ambas ganancias son de 0.01, es decir, el valor para la segunda entrada es como máximo 0.00027. Finalmente es necesario un bloque **Delay** debido a que la conmutación genera fenómenos no deseados en el comportamiento del circuito. La expresión que describe el comportamiento de este circuito es

$$V_{out} = \begin{cases} V_{in}, & \text{select} = V_{in} > 0, \\ 0.0001V_{in}, & \text{select} = V_{in} < 0. \end{cases} \quad (5.2)$$

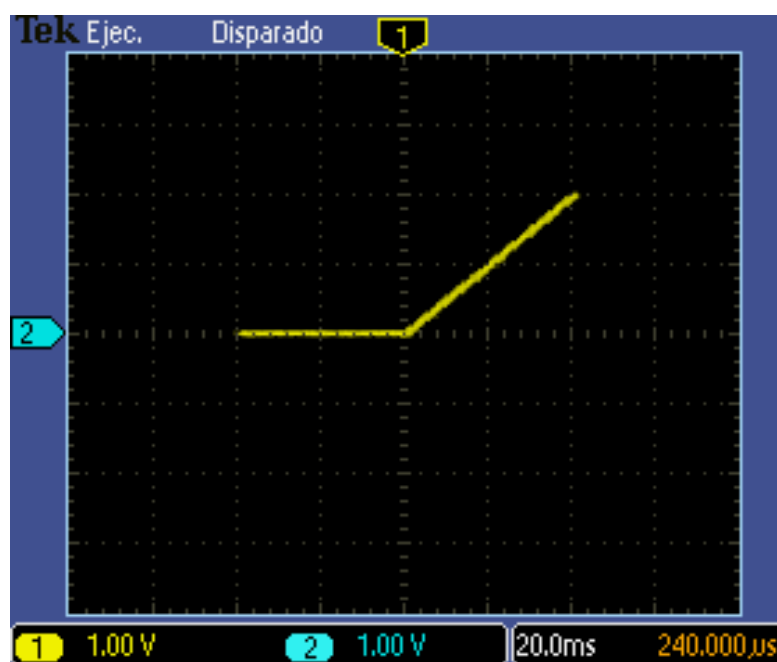
La implementación de la función de activación es presentada en la Figura 5.4 para una señal tipo rampa, notese que la entrada es multiplicada por una ganancia de 1.13, este valor fue encontrado experimentalmente para obtener un comportamiento adecuado.

5.1.3. Implementación de la función PWL

La función PWL utilizada es una función saturada de ganancia unitaria, este es un comportamiento descrito por un único bloque llamado **GainLimiter** presentado en la Figura ??, en este sentido esta es la función de activación con implementación más sencilla de las presentadas. En la implementación y registro de resultado fue encontrado que este bloque no funcionaba de manera adecuada para voltajes menores a -0.45 V, es por esto que los voltajes de satura-



(a)



(b)

Figura 5.4: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida de la función ReLu.



Figura 5.5: Configuración y programación en bloques de la ganancia saturada.

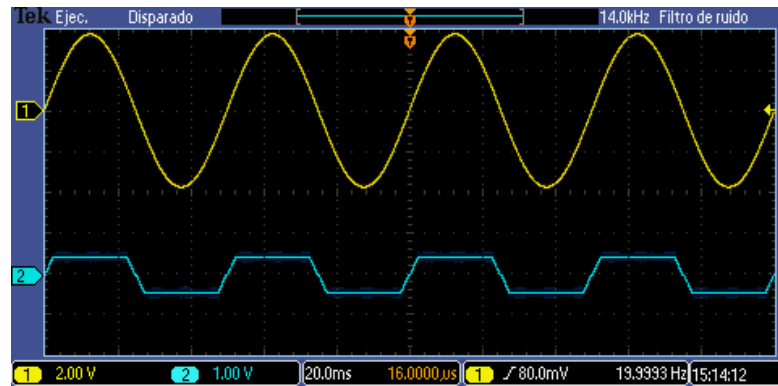
ción del bloque fueron ajustados a 0.4 V, a pesar de este cambio los parámetros en el modelo correspondiente no fueron modificados ya que el comportamiento caótico no se ve afectado, el único cambio cualitativo en el sistema es la magnitud de este. El resultado ante una entrada senoidal es presentado en la Figura 5.6 en (a) mientras que (b) es el resultado de graficar la señal de entrada contra la salida.

5.1.4. Implementación de la función tangente hiperbólica y sigmoide

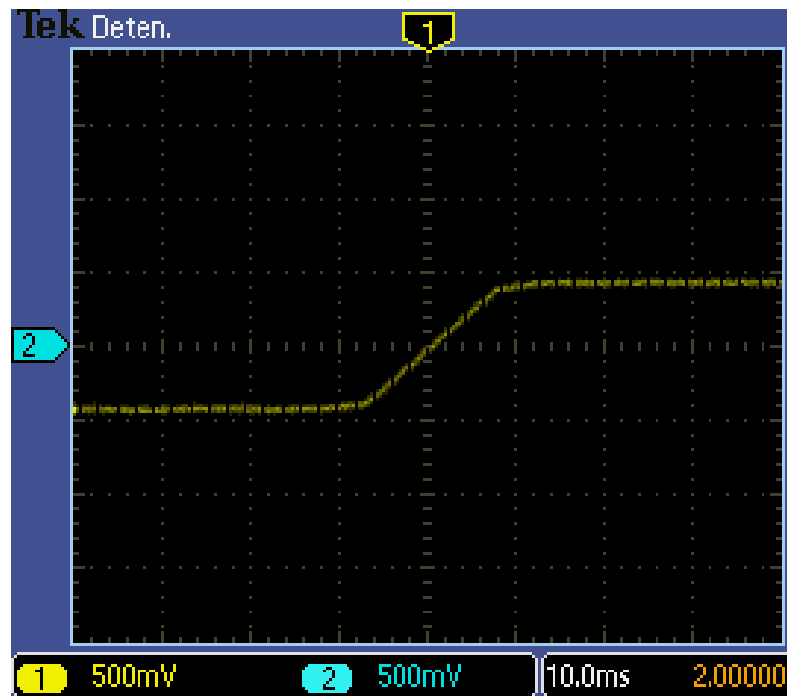
Para funciones más complejas Anadigm ofrece un bloque con memoria Look Up Table o LUT (Tabla de correspondencia en español), con la cual es posible asignar un valor de voltaje de entrada a uno de salida por medio de una SRAM de 256 x 8 bits, el diagrama de bloques de la LUT facilitado por Anadigm [49] es presentado en la Figura 5.7 en donde se observa la utilización de un ADC de 8 bits para la digitalización de la señal de entrada conectado a la memoria SRAM con 256 direcciones y su salida un DAC, en la misma figura se tiene la ventana para la programación de la LUP, a este es necesario cargar un documento .CSV con los 256 valores que se asignara a la SRAM. Por medio de la LUP es que se implementó las funciones no lineales tangente hiperbólica y sigmoide, esto implica que el resultado obtenido es un aproximado a las funciones reales. El resultado de la implementación de la función $\tanh(2V_{in})$ es presentado en la Figura 5.8 ante una función tipo rampa 6V de pico a pico con un periodo de 40ms junto a su respuesta de entrada contra salida, mientras que en la Figura 5.9 se presenta la respuesta de la función de activación $\sigma(2V_{in})$ ante la misma entrada.

5.2. Programación e implementación de las HNN's de 3 neuronas

Para la implementación de la red neuronal Hopfield en circuitos analógicos resulta conveniente transformar el sistemas de ecuaciones diferenciales que describen el sistema a su respuesta en frecuencia en base a la transformada de Laplace, consideremos la i -ésima ecuación



(a)



(b)

Figura 5.6: Resultado de la implementación electrónica a) con entrada senoidal y b) respuesta entrada contra salida de la función PWL.

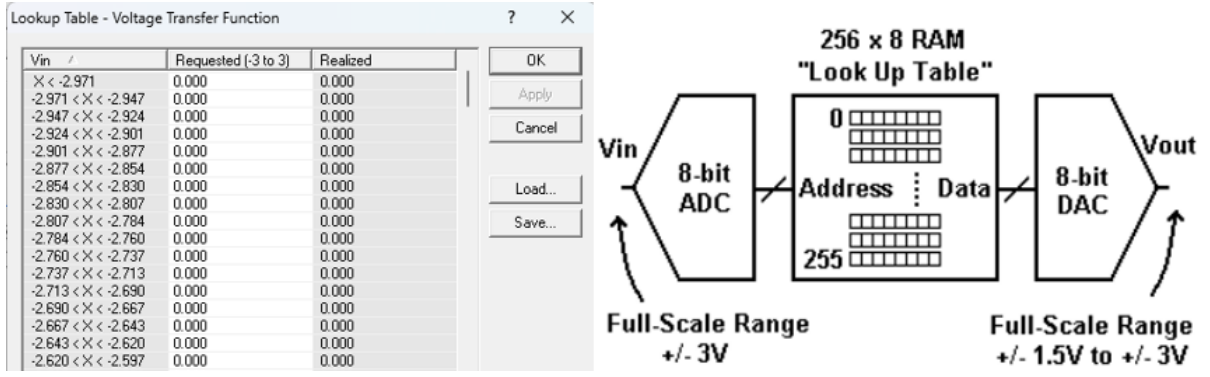


Figura 5.7: Diagrama en bloques de la Look Up Table.

que representa el voltaje que pasa por la respectiva neurona de una HNN

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^n w_{ij} f(x_j) + b_i, \quad (5.3)$$

aplicando la transformada de Laplace de la expresión anterior se tiene

$$\begin{aligned} \mathcal{L}(\dot{x}_i) &= -\mathcal{L}(c_i x_i) + \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ sX_i(s) &= -c_i X_i(s) + \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ \Rightarrow X_i(s)(s + c_i) &= \mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i) \\ \Rightarrow X_i(s) &= \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s + c_i} \end{aligned} \quad (5.4)$$

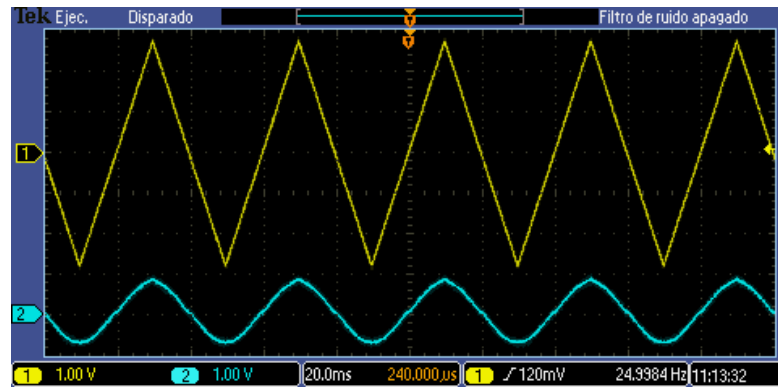
consideremos ahora un escalamiento en frecuencia de magnitud $2\pi f_0$, para el cual se realiza un cambio de variables $s \rightarrow s/(2\pi f_0)$, esto da

$$\begin{aligned} X_i(s) &= \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s/(2\pi f_0) + c_i} \\ X_i(s) &= 2\pi f_0 \frac{\mathcal{L}(\sum_{j=1}^n w_{ij} f(x_j) + b_i)}{s + 2\pi f_0 c_i} \end{aligned} \quad (5.5)$$

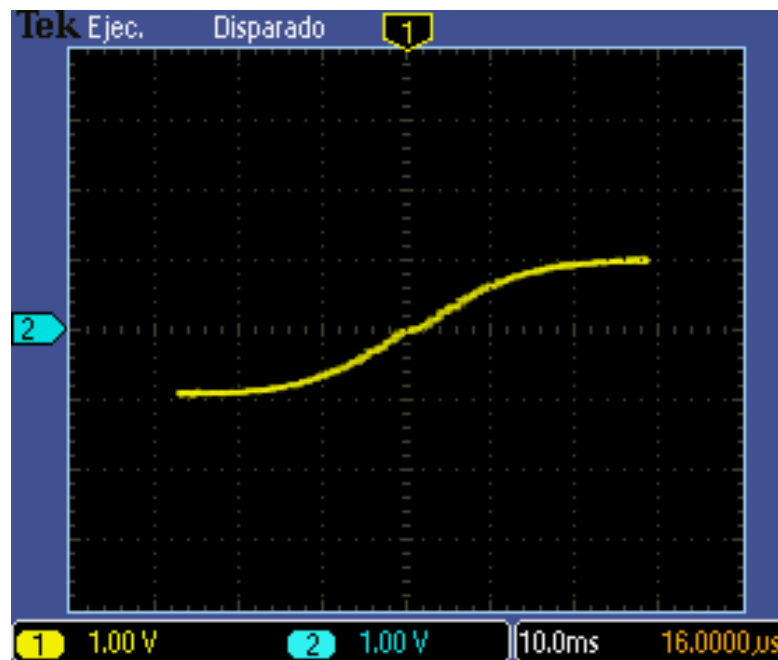
la forma de la expresión anterior es equivalente a un filtro sumador cuyo bloque esta incluido en la biblioteca del dispositivo analógico programable presentado en la Figura 5.10 junto a sus opciones y parámetros configurables. La función de transferencia equivalente a este bloque esta en la documentación de AnadigmDesigner2 y viene dada por

$$V_{out}(s) = \frac{2\pi f_0 \sum_{i=1}^3 \pm G_i V_{in-i}(s)}{s + 2\pi f_0}, \quad (5.6)$$

donde f_0 es la frecuencia de corte configurable del bloque en cuestión. En base a la expresión del filtro sumador podemos reescribir el modelo de las redes neuronales Hopfield.

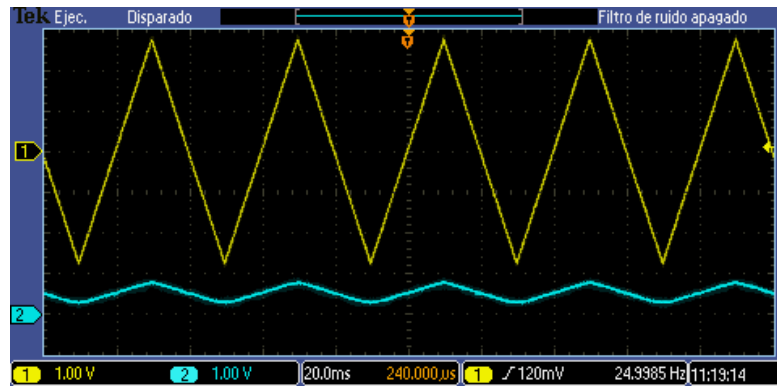


(a)

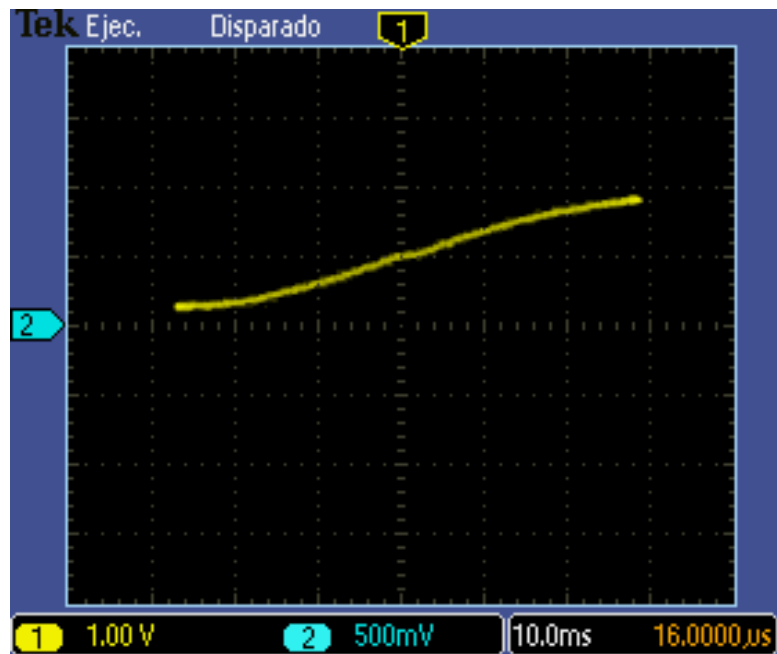


(b)

Figura 5.8: Resultado de la implementación electrónica a) con entrada rampa y b) respuesta entrada contra salida de la función tangente hiperbólica.



(a)



(b)

Figura 5.9: Resultado de la implementación electrónica a) con entrada sinoidal y b) respuesta entrada contra salida de la función sigmoide.

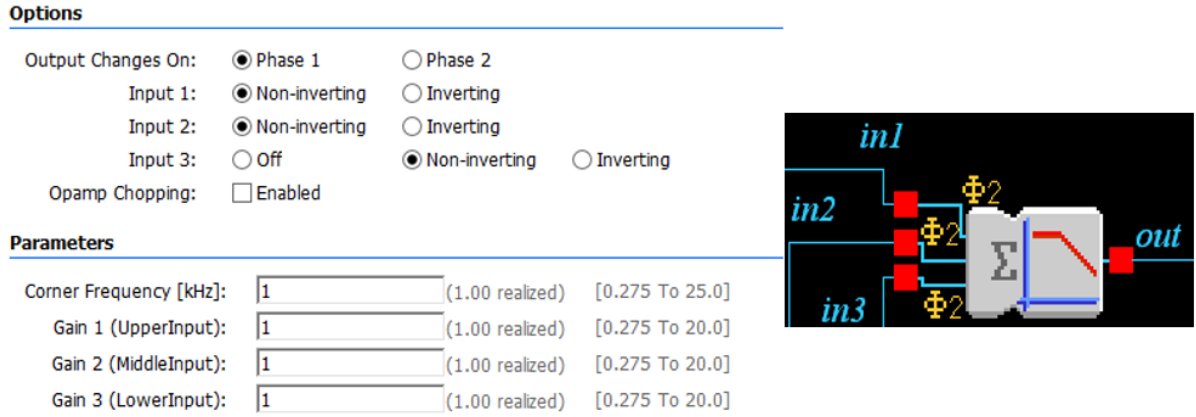


Figura 5.10: Bloque configurable de un filtro sumador.

5.2.1. Implementación del modelo con FA tangente hiperbólica

Para el modelo con función de activación tangente hiperbólica es importante considerar que el FPAA trabaja con voltajes de ± 3 , la magnitud obtenida en la simulación del modelo en cuestión se tiene que varia de -3.2 V a 3.2 V, es entonces necesario realizar un escalamiento de magnitud de la variable de estado $x_1(t)$ realizando un cambio de variable $x_1(t) \rightarrow 2x_1(t)$, esto da

$$\begin{aligned}\dot{x}_1 &= -x_1 - \tanh x_2 - 4.5 \tanh x_3 \\ \dot{x}_2 &= -x_2 + \tanh 2x_1 + 3 \tanh x_2 + 3 \tanh x_3 \\ \dot{x}_3 &= -x_3 - \tanh 2x_1 - 3 \tanh x_2.\end{aligned}\tag{5.7}$$

Para su implementación en la tarjeta analógica se tiene

$$\begin{aligned}X_1(s) &= 2\pi f_0 \frac{-\tanh X_2(s) - 4.5 \tanh X_3(s)}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{\tanh(2X_1(s)) + 3 \tanh X_1(s) + 3 \tanh X_3(s)}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{-\tanh(2X_1(s)) - 3 \tanh X_2(s)}{s + 2\pi f_0},\end{aligned}\tag{5.8}$$

con $f_0 = 1$ kHz, este valor se respeta para las restantes implementaciones. **La programación en bloques de este circuito es presentada en la sección de Anexo**, lo mismo ocurre para los circuitos implementados posteriores. El resultado de la implementación de 5.8 se muestra en la Figura 5.11 con los retratos de fase del modelo 3.5 de los planos $x_1 - x_2$ y para el plano $x_1 - x_3$ se tiene la Figura 5.12. Junto a cada implementación se coloca la simulación del sistema (escalado de ser necesario) con propósitos de comparación.

5.2.2. Implementación del modelo con FA sigmoide

El modelo con la función de activación sigmoide requiere de una escalamiento en magnitud considerando los retratos de fase obtenidos por la simulación, los escalamientos son $x_1 \rightarrow 2x_1$,

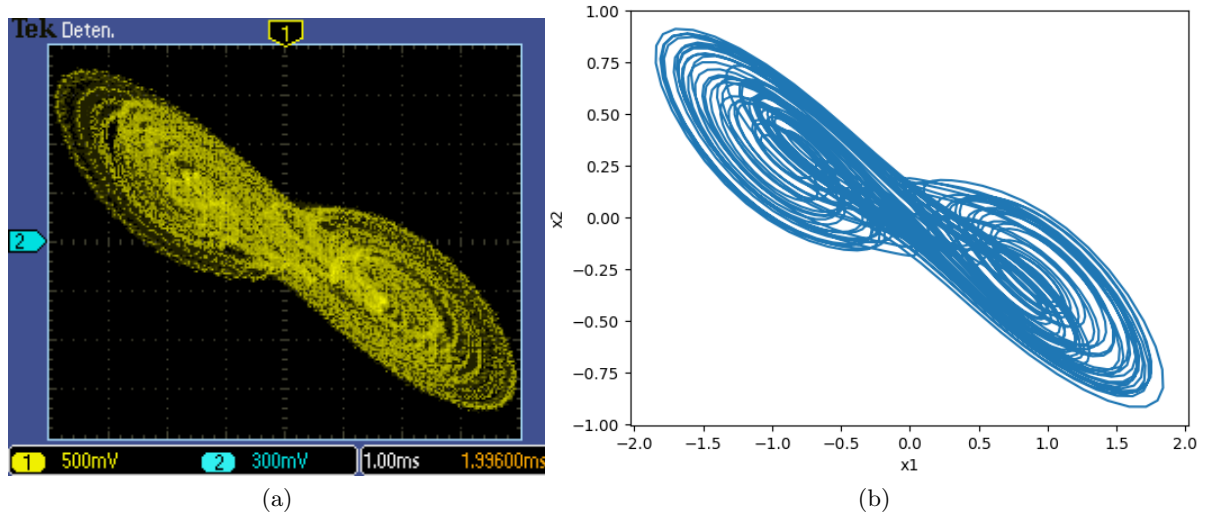


Figura 5.11: Retratos de fase del modelo con función de activación tangente hiperbólica para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

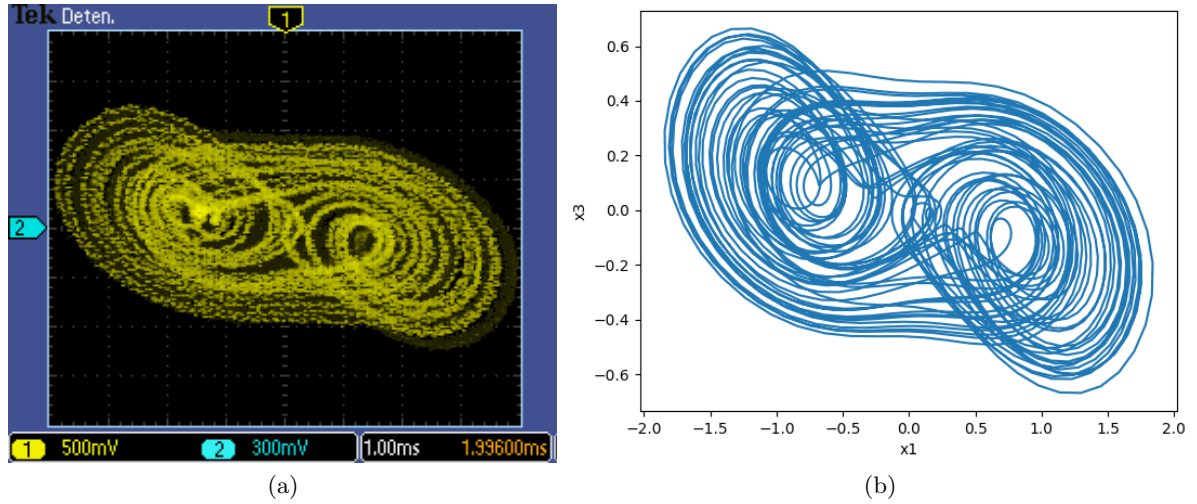
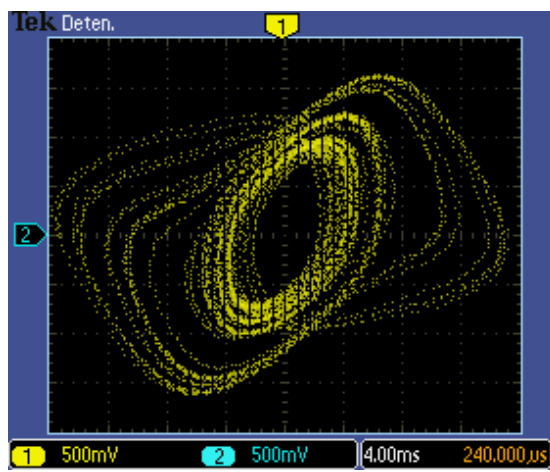
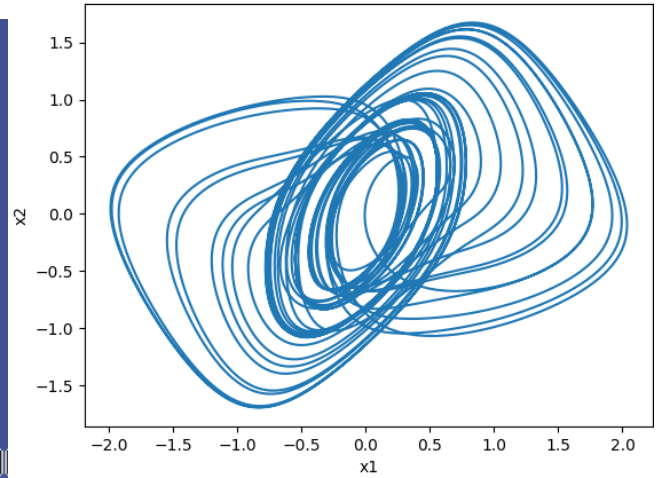


Figura 5.12: Retratos de fase del modelo con función de activación tangente hiperbólica para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

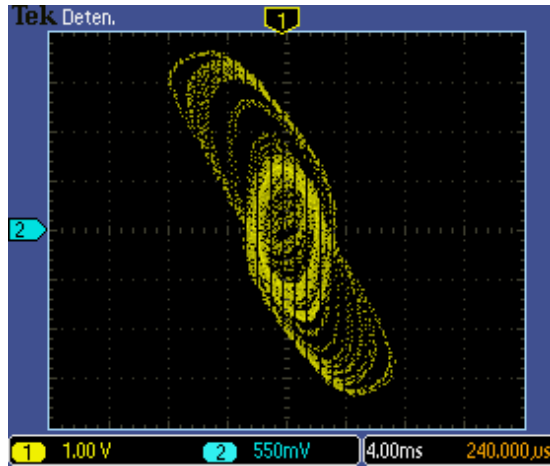


(a)

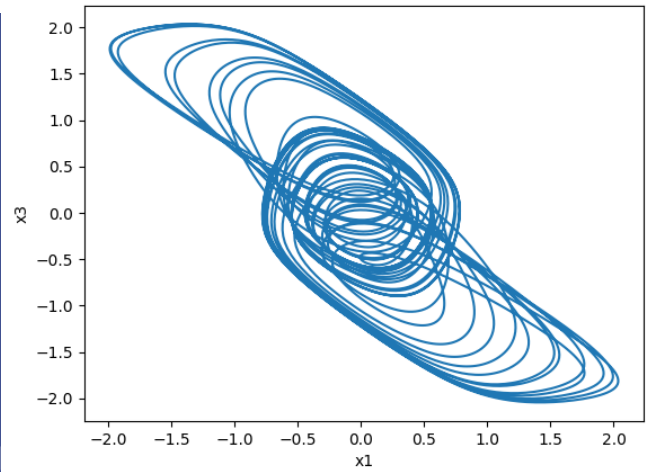


(b)

Figura 5.13: Retratos de fase del modelo con función de activación sigmoide para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.



(a)



(b)

Figura 5.14: Retratos de fase del modelo con función de activación sigmoide para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema escalado en magnitud.

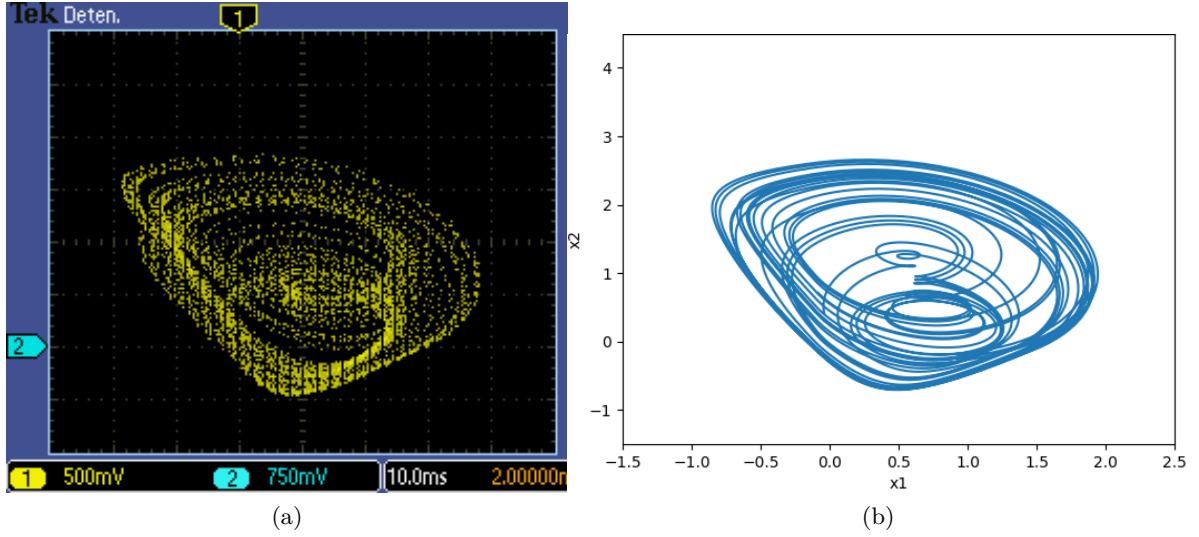


Figura 5.15: Retrato de fase del modelo con función de activación ReLU con a) la implementación electrónica y b) su simulación usando el método RK-2 con un paso de integración $1e-3$ y condición inicial $[0.1, 0]$.

$x_2 \rightarrow 2.5x_2$ y $x_3 \rightarrow 5x_3$, con lo cual se obtiene

$$\begin{aligned}\dot{x}_1 &= -x_1 + 3.985\sigma(2x_1) - 2\sigma(2.5x_2) - 1 \\ \dot{x}_2 &= -x_2 + 4\sigma(2x_1) + 2.736\sigma(2.5x_2) + 2\sigma(5x_3) - 4.365, \\ \dot{x}_3 &= -x_3 - 3.8\sigma(2x_1) + 0.734\sigma(5x_3) + 1.532\end{aligned}\quad (5.9)$$

el voltaje externo b_2 sobrepasa el voltaje máximo con el que trabajo el FPAA, esta problemática puede ser resulta escalando la frecuencia del sistema multiplicando la parte derecha del sistema por una constante, sea 0.6 esa constante para obtener

$$\begin{aligned}\dot{x}_1 &= -0.6x_1 + 2.4\sigma(2x_1) - 1.2\sigma(2.5x_2) - 0.6 \\ \dot{x}_2 &= -0.6x_2 + 2.4\sigma(2x_1) + 1.64\sigma(2.5x_2) + 1.2\sigma(5x_3) - 2.62, \\ \dot{x}_3 &= -0.6x_3 - 2.26\sigma(2x_1) + 0.44\sigma(5x_3) + 0.92\end{aligned}\quad (5.10)$$

que usando al transformada de Laplace puede ser escrito como

$$\begin{aligned}X_1(s) &= 2\pi f_0 \frac{0.4X_1(s) + 2.4\sigma(2X_1(s)) - 1.2\sigma(2.5X_2(s)) - 0.6}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{0.4X_2(s) + 2.4\sigma(2X_1(s)) + 1.64\sigma(2.5X_2(s)) + 1.2\sigma(5X_3(s)) - 2.62}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{0.4X_3(s) - 2.26\sigma(2X_1(s)) + 0.44\sigma(5X_3(s)) + 0.92}{s + 2\pi f_0},\end{aligned}\quad (5.11)$$

con el cual se formó el código presentado en el anexo y se obtuvieron los resultados mostrados en la Figura 5.13, junto a este resultado se presenta la simulación del modelo escalado para evidenciar similitud.

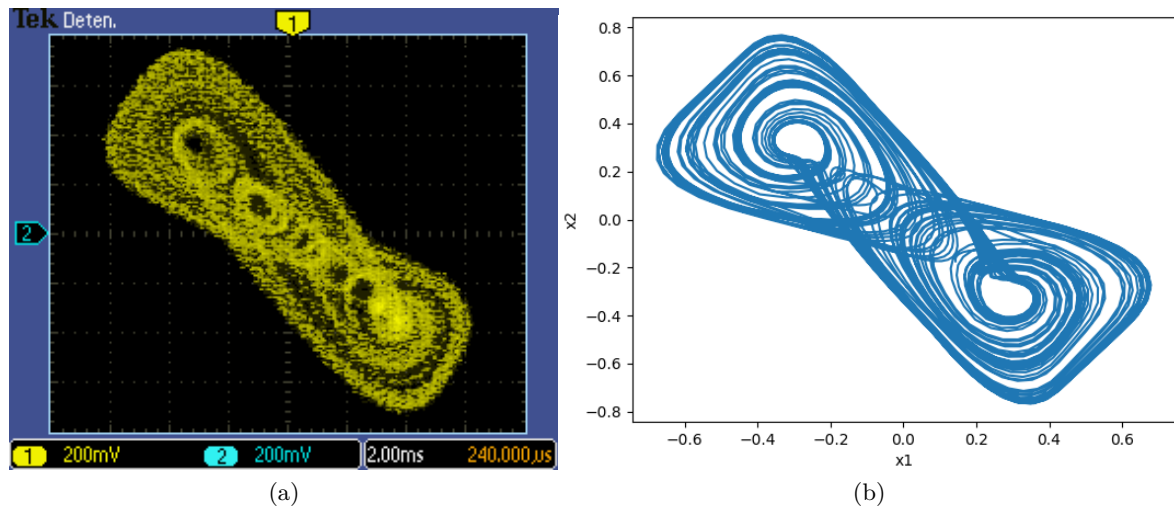


Figura 5.16: Retratos de fase del modelo con función de activación PWL para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema.

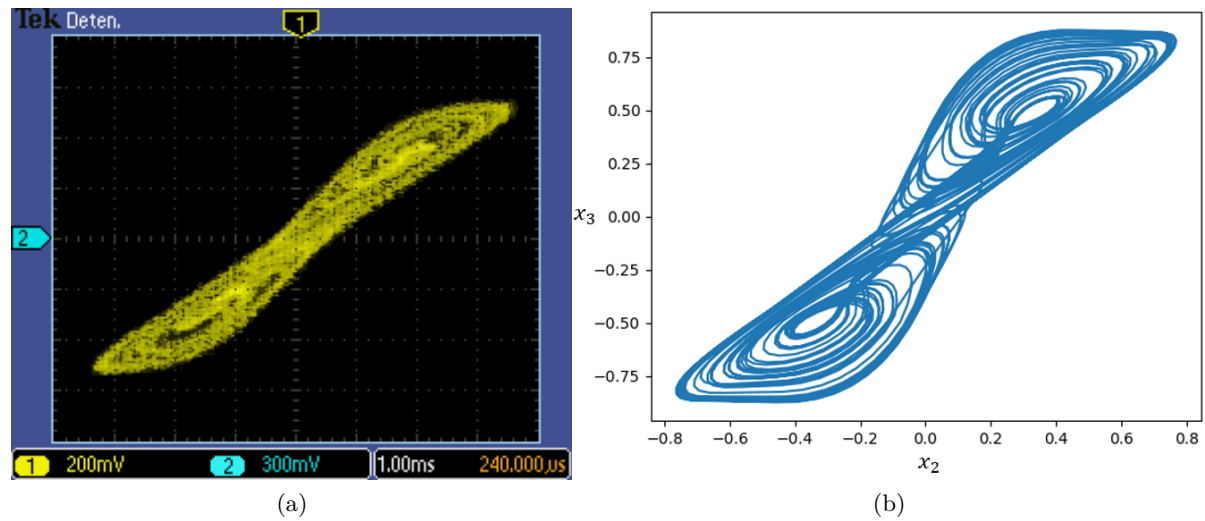


Figura 5.17: Retratos de fase del modelo con función de activación PWL para el plano $x_2 - x_3$ en a) implementación electrónica y a) simulación del sistema.

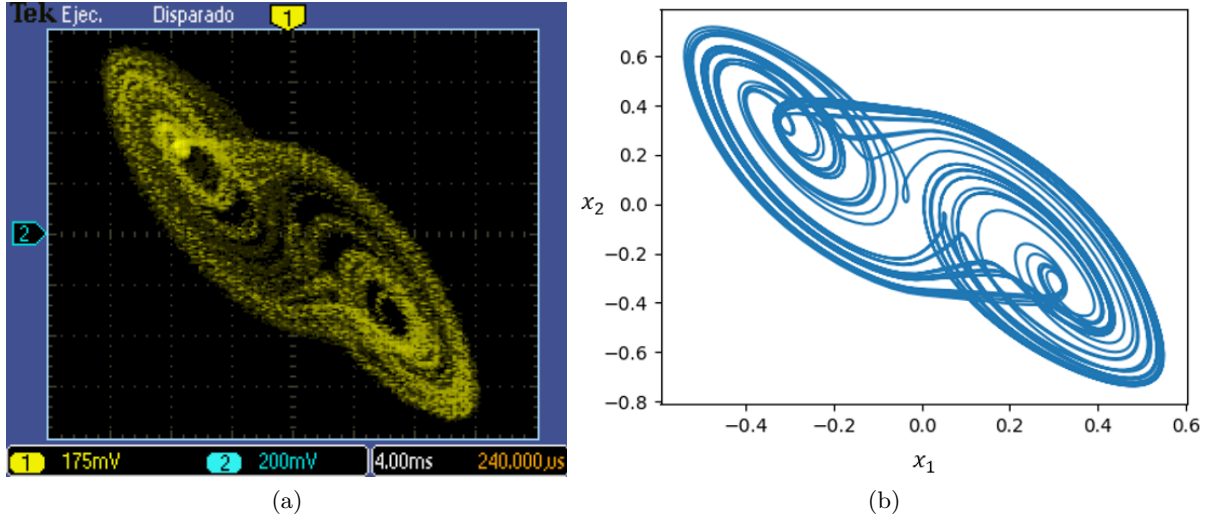


Figura 5.18: Retratos de fase del modelo con función de activación signo para el plano $x_1 - x_2$ en a) implementación electrónica y a) simulación del sistema.

5.2.3. Implementación del modelo con FA $\text{ReLU}(x)$

Para la implementación del con función ReLU se utilizó una fuente de voltaje externa para generar la señal bias senoidal sobre la neurona x_1 , usando la transformada de Laplace sobre el modelo se tiene

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{-0.15X_1(s) + 1.1\text{ReLU}(X_1(s)) - 1.05\text{ReLU}(X_2(s)) - 0.5\text{sen}(0.52t) + 0.8}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{0.45X_2(s) + 0.95\text{ReLU}(X_1(s)) + 0.75\text{ReLU}(X_2(s)) - 0.75}{s + 2\pi f_0}, \end{aligned} \quad (5.12)$$

con resultado en su implementación presentado en la Figura 5.15.

5.2.4. Implementación del modelo con FA PWL

De la misma forma que el modelo con función ReLU, el modelo con función PWL no requiere de un escalamiento y su transformación queda como

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{1.68\text{satlin}(X_1(s)) + 3.81\text{satlin}(X_2(s)) - 2.23\text{satlin}(X_3(s))}{s + 2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{-5.4\text{satlin}(X_1(s)) + 1.8\text{satlin}(X_2(s)) - 4.4\text{satlin}(X_3(s))}{s + 2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{-4.38\text{satlin}(X_1(s)) - 2.2\text{satlin}(X_2(s)) - 0.3\text{satlin}(X_3(s))}{s + 2\pi f_0}, \end{aligned} \quad (5.13)$$

con resultado en su implementación dado en la Figura 5.16 y 5.17 junto con el sistema escalado.

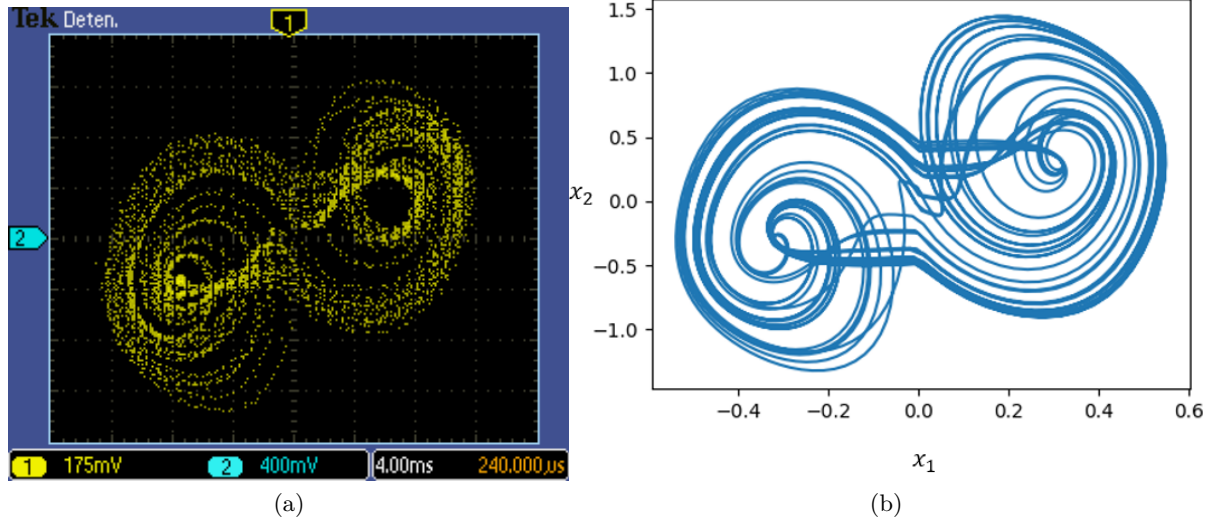


Figura 5.19: Retratos de fase del modelo con función de activación signo para el plano $x_1 - x_3$ en a) implementación electrónica y a) simulación del sistema.

Finalmente para el modelo con función de activación signo se tiene

$$\begin{aligned} X_1(s) &= 2\pi f_0 \frac{-\tanh X_2(s)}{s+2\pi f_0}, \\ X_2(s) &= 2\pi f_0 \frac{-\tanh X_3(s)}{s+2\pi f_0}, \\ X_3(s) &= 2\pi f_0 \frac{0.65 \tanh(X_1(s)) + 4 \tanh X_2(s) + 3 \tanh X_3(s)}{s+2\pi f_0}, \end{aligned} \quad (5.14)$$

con evidencias de implementación en la Figura 5.18 para el plano $x_1 - x_2$ y 5.19 para el plano $x_1 - x_3$.

5.3. Implementación del modelo de hipercaótico HNN de 4 neuronas

Al aplicar la transformada de Laplace sobre el modelo descrito en la ecuación (3.25) se obtiene

$$sX_1(s) = -X_1(s) + \mathcal{L}\{\tanh x_1 + 0.5 \tanh x_2 - 3 \tanh x_3 - \tanh x_4\}, \quad (5.15)$$

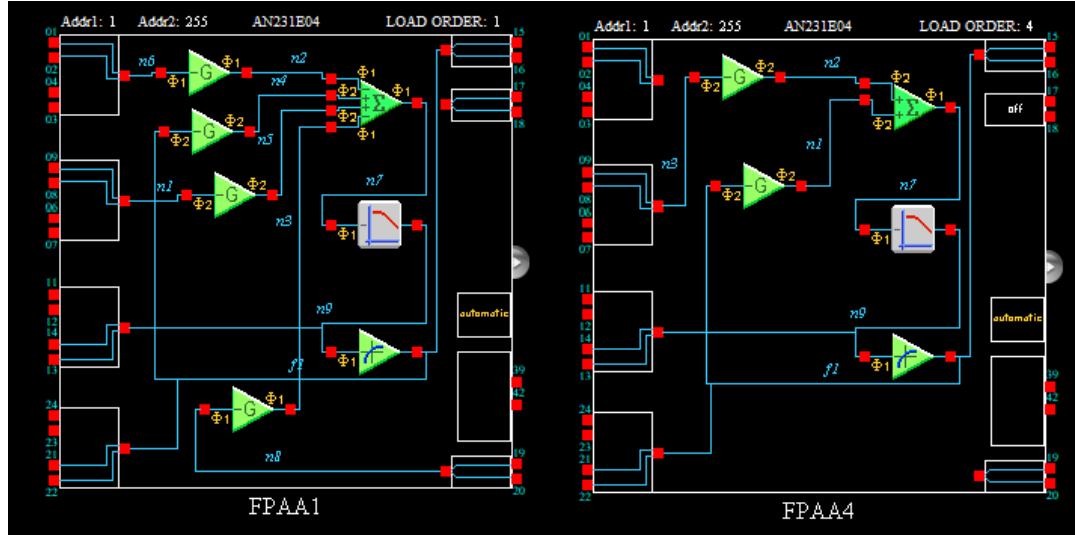
$$sX_2(s) = -X_2(s) + \mathcal{L}\{2 \tanh x_2 + 3 \tanh x_3\}, \quad (5.16)$$

$$sX_3(s) = -X_3(s) + \mathcal{L}\{3 \tanh x_1 - 3 \tanh x_2 + \tanh x_3\}, \quad (5.17)$$

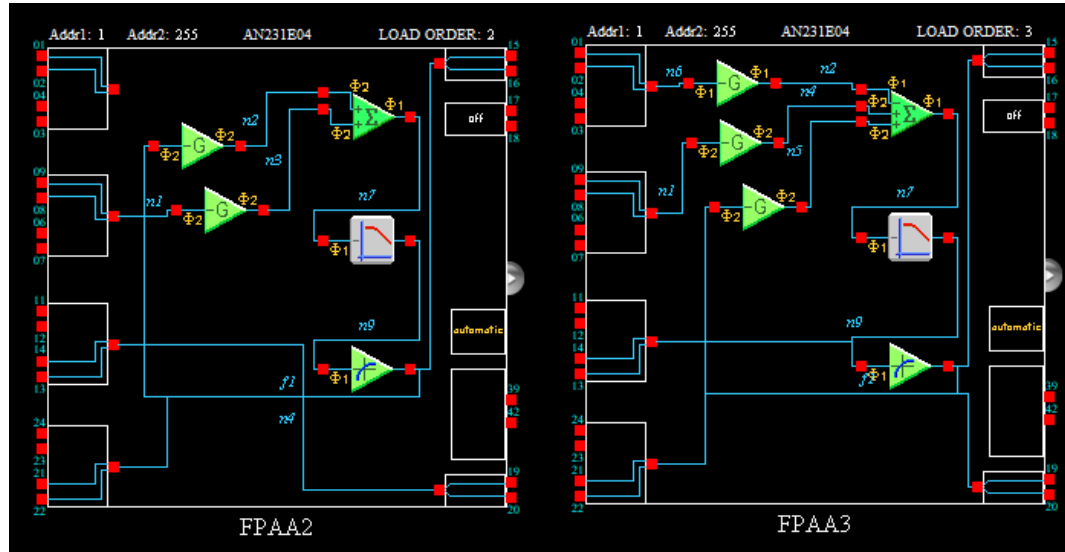
$$sX_4(s) = -100X_4(s) + \mathcal{L}\{100 \tanh x_1 + 170 \tanh x_4\}, \quad (5.18)$$

$$(5.19)$$

5.3. IMPLEMENTACIÓN DEL MODELO DE HIPERCAÓTICO HNN DE 4 NEURONAS 89



(a)



(b)

Figura 5.20: Código en bloques para la programación del modelo HNN de 4 neuronas descrito en la ec. (5.27).

que puede ser reescrito como

$$X_1(s) = \frac{\mathcal{L}\{\tanh x_1 + 0.5 \tanh x_2 - 3 \tanh x_3 - \tanh x_4\}}{s + 1}, \quad (5.20)$$

$$X_2(s) = \frac{\mathcal{L}\{2 \tanh x_2 + 3 \tanh x_3\}}{s + 1}, \quad (5.21)$$

$$X_3(s) = \frac{\mathcal{L}\{3 \tanh x_1 - 3 \tanh x_2 + \tanh x_3\}}{s + 1}, \quad (5.22)$$

$$X_4(s) = \frac{\mathcal{L}\{100 \tanh x_1 + 170 \tanh x_4\}}{s + 100}. \quad (5.23)$$

Notemos que el valor de $w_{41} = c_4 = 100$ y w_{44} , estas ganancias generan una problemática en la implementación en circuitos electrónicos. En específico, al comparar con el filtro dado por AN2 descrito en la ec. (5.6) notemos que el termino con f_0 debe ser igual, de esta manera reescribamos y escalemos en frecuencia el sistema por un factor de 0.25 kHz y 25 kHz para obtener

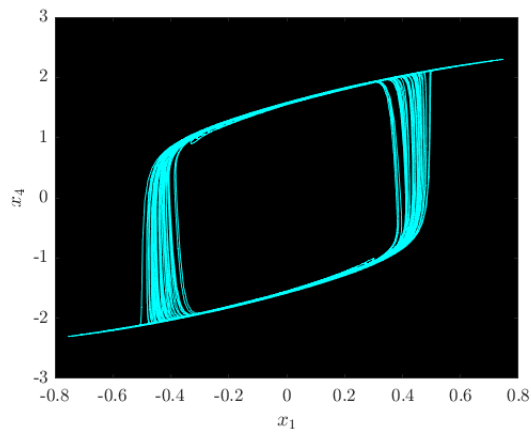
$$X_1(s) = \frac{2\pi(0.25\text{kHz})}{s + 2\pi(0.25\text{kHz})} \mathcal{L}\{\tanh x_1 + 0.5 \tanh x_2 - 3 \tanh x_3 - \tanh x_4\}, \quad (5.24)$$

$$X_2(s) = \frac{2\pi(0.25\text{kHz})}{s + 2\pi(0.25\text{kHz})} \mathcal{L}\{2 \tanh x_2 + 3 \tanh x_3\}, \quad (5.25)$$

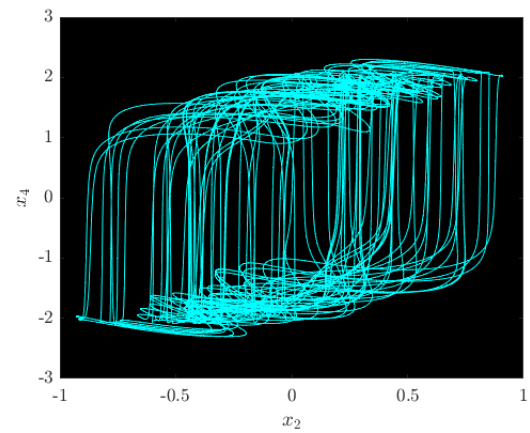
$$X_3(s) = \frac{2\pi(0.25\text{kHz})}{s + 2\pi(0.25\text{kHz})} \mathcal{L}\{3 \tanh x_1 - 3 \tanh x_2 + \tanh x_3\}, \quad (5.26)$$

$$X_4(s) = \frac{1}{25} \frac{2\pi(25\text{kHz})}{s + 2\pi(25\text{kHz})} \mathcal{L}\{100 \tanh x_1 + 170 \tanh x_4\}, \quad (5.27)$$

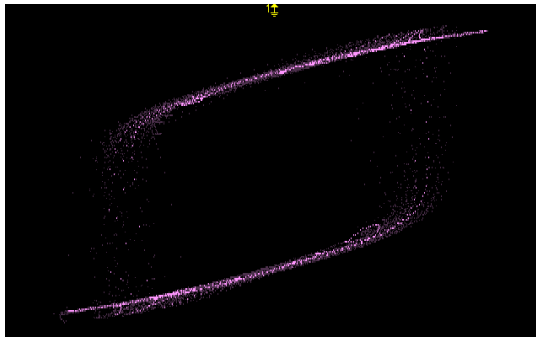
de esta manera las ganancias de w_{11} y w_{44} son escaladas a 4 y 6.8 respectivamente. De la expresión anterior podemos notar que las ganancias han sido escaladas y es posible una implementación en las tarjetas analógicas configurables como se muestra en la Figura 5.20. Al ser un sistema de 4 dimensiones se tiene un posible de 6 retratos de fase distintos. En la Figura 5.21 se presentan dos retratos de fase tanto en simulación como en su implementación $x_1 - x_4$ y $x_2 - x_4$ respectivamente.



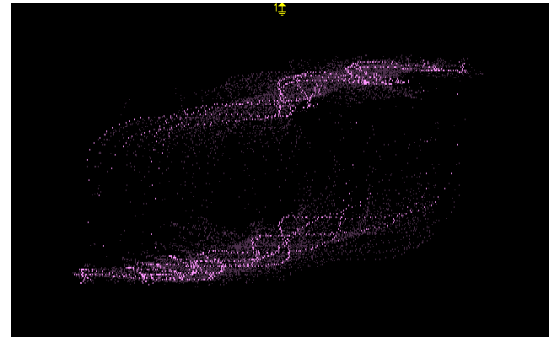
(a)



(b)



(c)



(d)

Figura 5.21: Retratos de fase del modelo con de 4 neuronas para el plano $x_1 - x_4$.

Capítulo 6

Desarrollo de una interfaz como metodología para la implementación del modelo HNN

La metodología para la implementación en un ambiente de circuitos electrónicos requiere que cualquiera que desee implementar una red neuronal Hopfield sea capaz de hacerlo al seguir una serie de pesos previamente estructurados. Como metodología entonces, se propone el desarrollo de una interfaz específica en Qt-Python la cual capture los parámetros de la red neuronal Hopfield de 3 y 4 neuronas así como también las funciones de activación de cada una de las neuronas. La interfaz en cuestión generara un archivo el es cargado en la tarjeta que automáticamente la configura para tener el comportamiento de la HNN deseada. La captura de las funciones de activación puede ser expresamente a través de su expresión matemática o también a través de un archivo de texto con 256 con su valor equivalente considerando una entrada de voltaje de -3V hasta 3V.

6.1. Topología de los circuitos en AN2

Para la generación en texto de los circuitos con equivalencia al modelo de red neuronal se desarrollaron distintas topologías dependiendo de las características del modelo que se desea implementar. En general se desarrollaron 4 topologías distintas:

- Modelo HNN de 3 neuronas sin bias.
- Modelo HNN de 3 neuronas con bias
- Modelo HNN de 4 neuronas sin bias.

Tabla 6.1: Módulos configurables analógicos usado en el diseño de las redes neuronales junto con su código en hexadecimal.

Nombre en AN2	Diseño del bloque	Expresión equivalente	Código en hexadecimal
GainInv		$V_{out} = -\frac{C_{in}}{C_{out}} V_{in}$	0xC _{in} , 0xC _{in} , 0xC _{out} , 0xC _{out}
GainHalf		$V_{out} = \pm \frac{C_{in}}{C_{out}} V_{in}$	0xC _{out} , 0xC _{in}
SumDiff		$V_{out} = \pm \frac{C_{inA}}{C_{out}} V_{in1} \pm \frac{C_{inB}}{C_{out}} V_{in2} \pm \frac{C_{inC}}{C_{out}} V_{in3}$	0xC _{out} , 0xC _{inA} , 0xC _{inB} , 0xC _{inC}
SumInt		$\frac{V_{out}(s)}{s} = \pm \frac{C_{inA}}{C_{out}} V_{in1} \pm \frac{C_{inB}}{C_{out}} V_{in2}$	0xC _{inA} , 0xC _{inB} , 0xC _{out}
Transfer Function		$V_{out} = f(V_{in})$	$0x\frac{255}{6}f\left(-3 + \frac{6*0}{255}\right),$ $0x\frac{255}{6}f\left(-3 + \frac{6*1}{255}\right),$ $\vdots$ $0x\frac{255}{6}f\left(-3 + \frac{6*255}{255}\right)$
Voltage		$V_{out} = \pm 2V$	$\begin{cases} 0x81, & V_{out} = 2V \\ 0x18, & V_{out} = -2V \end{cases}$

Para cada uno de estos se desarrollo un circuito distinto en AN2, generando con este con archivo de texto en hexadecimal con los módulos configurables analógicos y la conexión entre ellos, este mismo archivo de texto es posteriormente modificado por la interfaz en Qt-Python para ser cargado en la tarjeta. Entonces es importante dar las especificaciones de cada modulo utilizada, los distintos módulos utilizados para la topología de los circuitos para los distintos modelos se resumen en la Tabla 6.1, en donde se presenta el nombre que asigno AN2, el diseño de bloque, su ecuación equivalente y su código equivalente en hexadecimal. En base a las expresiones de la Tabla 6.1 podemos una de las variables de estado del modelo de las HNN de 3 dimensiones. Tomemos la i -ésima variable expresada de la forma

$$X_i(s) = 2\pi f_0 \frac{-c_i X_i(s) + w_{i1}g_1(X_1(s)) + w_{i2}g_2(X_2(s)) + w_{i3}g_3(X_3(s))}{s}, \quad (6.1)$$

donde g_i es la función de activación de la i -ésima neurona y $2\pi f_0$ es el escalamiento en frecuencia del sistema. Su equivalente usando los módulos analógicos configurables en el ambiente de programación AN2 es presentado en la Figura 6.1a, en este caso específico se utilizó la variable $\dot{x}_1$ y correspondientemente se colocaron etiquetas en las entradas y salidas de los bloques. En la Figura 6.4b se colocó la topología utilizada para la implementación de las 3D-HNN con bias, en este caso se agregó una señal interna de voltaje de 2 volts, para especificar el valor requerido se coloca una ganancia a su salida. La conexión externa de la tarjeta para la implementación de dichas topologías es presentada en la Figura 6.1c con un diagrama ilustrado, cada conexión entre las celdas de salida o entrada de FPAA se ilustra con un cable de cierto color, del lado derecho de la misma figura se colocó el número de FPAA junto a la celda de entrada o salida a la que va conectada. Así mismo se da la indicación de habilitar dos DIP switch de la tarjeta, en específico, el DIP switch del FPAA 1 que conecta su salida 4 con la entrada 1 del FPAA 2, y el DIP switch del FPAA 2 que conecta su salida 4 con la entrada 1 del FPAA 3.

Para la topología de los circuitos para la implementación de las 4D-HNN primero notemos que la transformada de Laplace de una de sus variables de estado tendría una forma similar a la de la ec. (6.1), es por esto que la estructura de la topología que similar, esta es presentada en la Figura 6.2 junto con la configuración externa de la tarjeta.

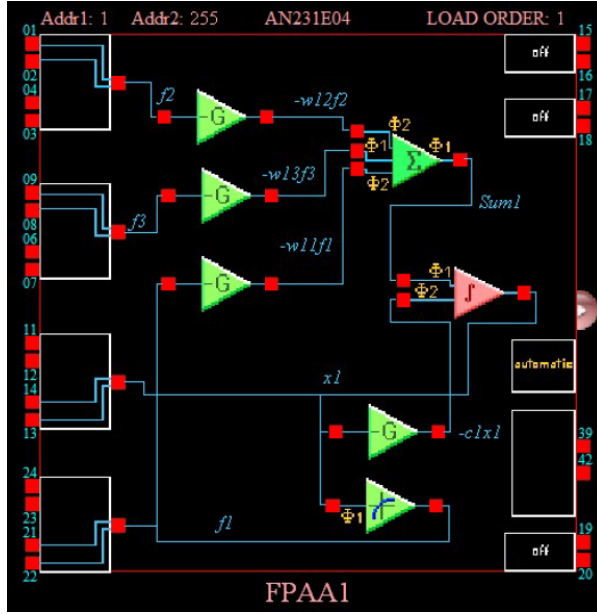
6.2. Desarrollo de la interfaz en Qt-Python

Cada uno de los circuitos realizados en la sección anterior generan un archivo de texto en hexadecimal con la configuración interna de la tarjeta, este mismo archivo puede ser cargado a la tarjeta para configurarla con alguno de los modelos de la red neuronal Hopfield. Con base en este archivo de texto es que se desarrolló una interfaz capaz de tomar los parámetros y la función de activación introducidos por un usuario e implementar el modelo descrito en las tarjetas analógicas sin la necesidad que el usuario pase por todos los pasos de diseño de la red.

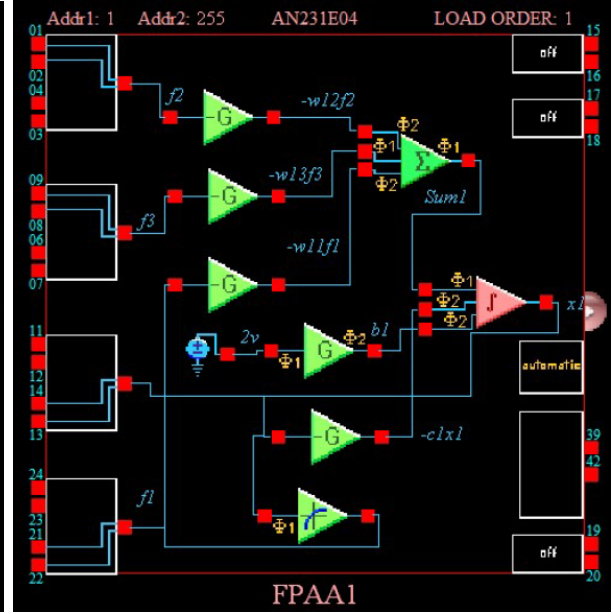
En la Figura 6.3 se desglosa el archivo de texto que configura el primer FPAA de tarjeta al implementar el modelo de red neuronal con la FA PWL, en esta imagen se resalta y describe cada uno de los módulos analógicos en su código hexadecimal. Notemos que en múltiples ocasiones las expresiones que representan cada bloque requieren de dos o más valores de capacitancia C_{in} y C_{out} , estos son valores hexadecimales entre 0 y FF que indican el valor de una ganancia de la forma:

$$k = \frac{C_{in}}{C_{out}}, \quad (6.2)$$

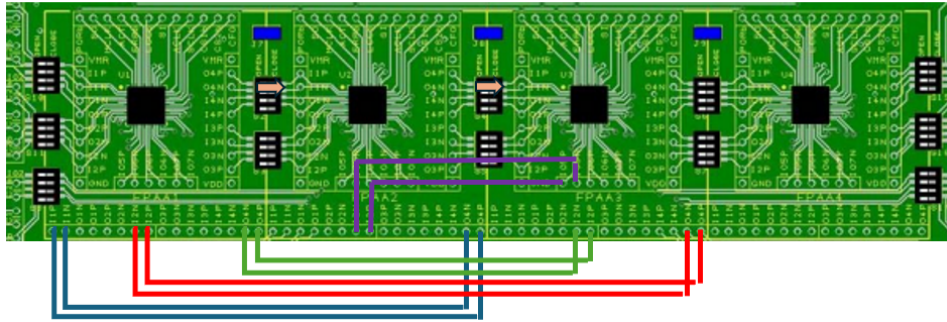
dado que el valor de estas capacitancias está acotada entre 256 valores posibles existe la posibilidad que k se describa de múltiples, una o cero formas. Para hallar un valor adecuado de



(a)

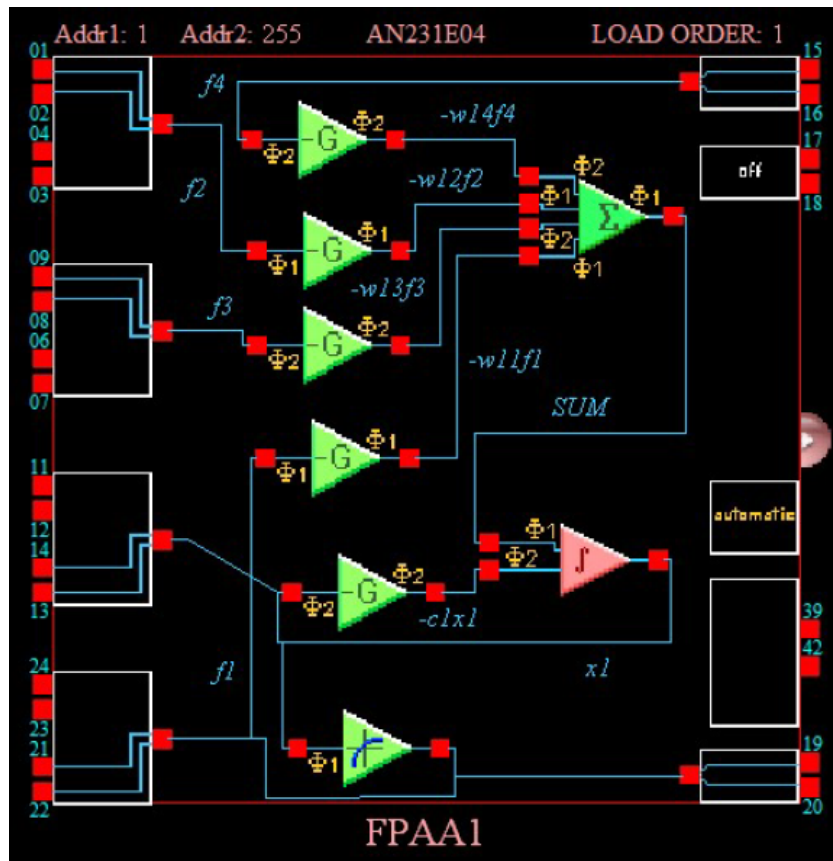


(b)

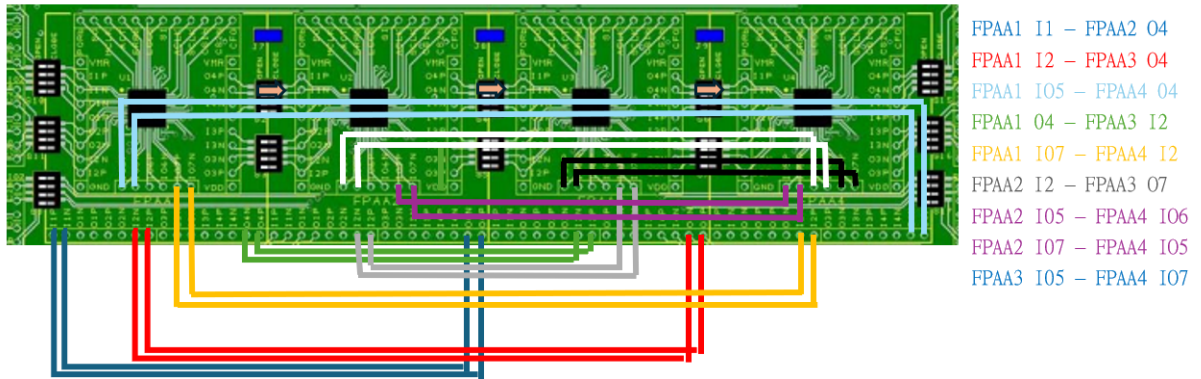


(c)

Figura 6.1: Diagrama de la conexión interna a), b) y externa en c) de la tarjeta analógica embebida para la topología de los circuitos de la implementación de las 3D-HNN.



(a)



(b)

Figura 6.2: Diagrama de la conexión interna en a) y externa en b) de la tarjeta analógica embebida para la topología de los circuitos de la implementación de las 4D-HNN sin bias.

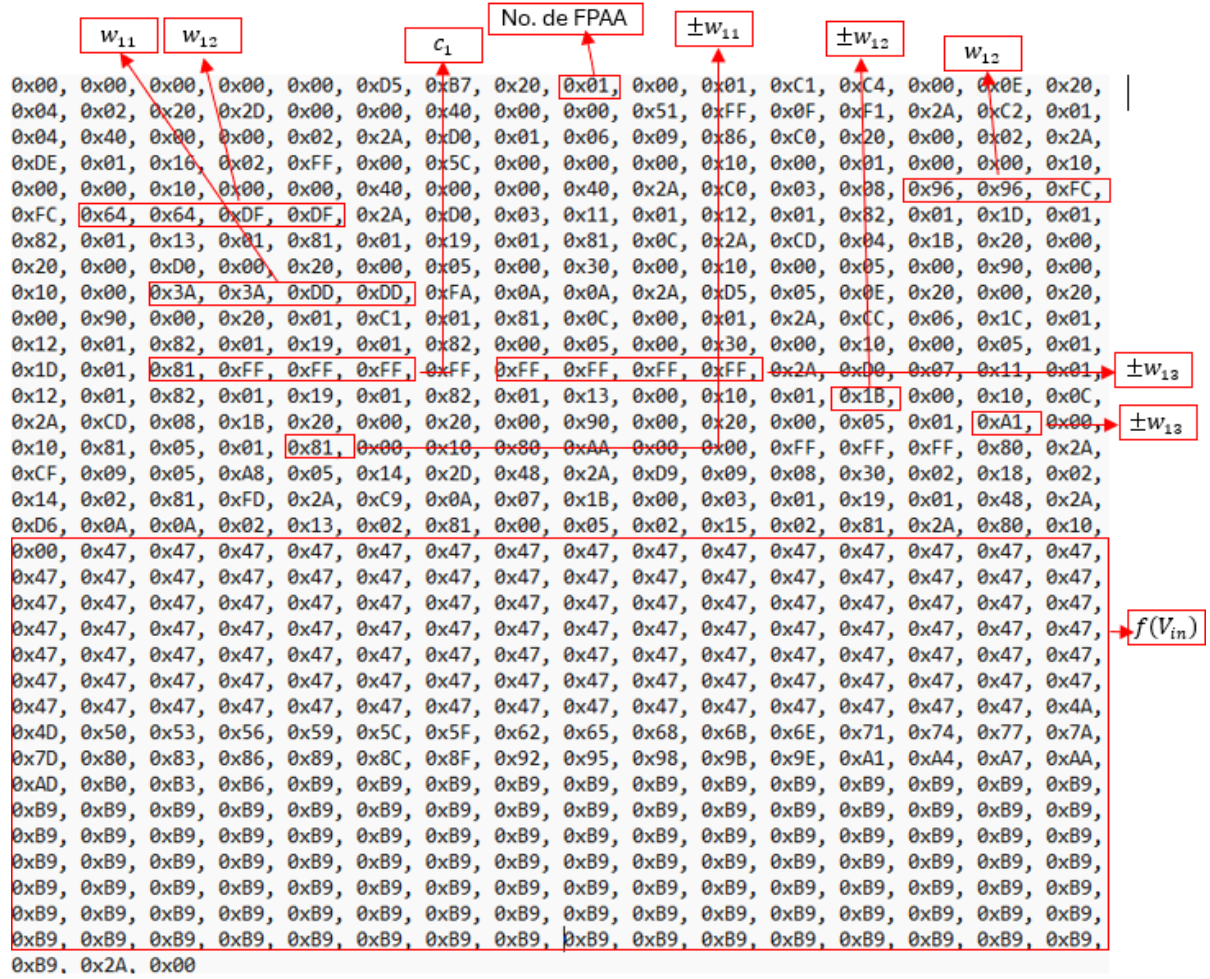


Figura 6.3: Código hexadecimal que describe la topología del circuito de las 3D-HNN.

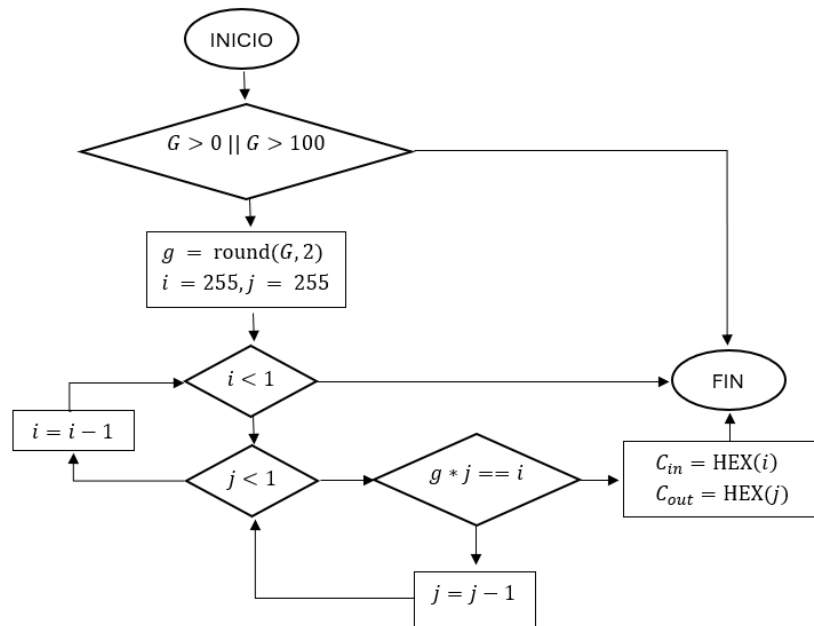


Figura 6.4: Diagrama de flujo para hallar los valores adecuados de las capacitancias C_{in} y C_{out} .

C_{in} y C_{out} se desarrollo el algoritmo presentado en la Figura 6.4, este algoritmo es llevado a un método de Python en donde se solicita un argumento G que representa la ganancia requerida. Notemos del mismo algoritmo que es necesario algún método `round(·)` que permita redondear la ganancia a dos decimales.

La interfaz desarrollada en Qt se presenta en la Figura 6.5 con cada una de sus secciones desglosadas, en esta se debe seleccionar la dimensión de la red neuronal, ingresar sus parámetros, así como también las distintas funciones de activación de la red neuronal. Como es mencionado al inicio de la sección, las funciones de activación pueden ser descritas por algunas expresiones matemáticas o cargadas por un archivo de texto con los 256 valores de la función de activación discretizada.

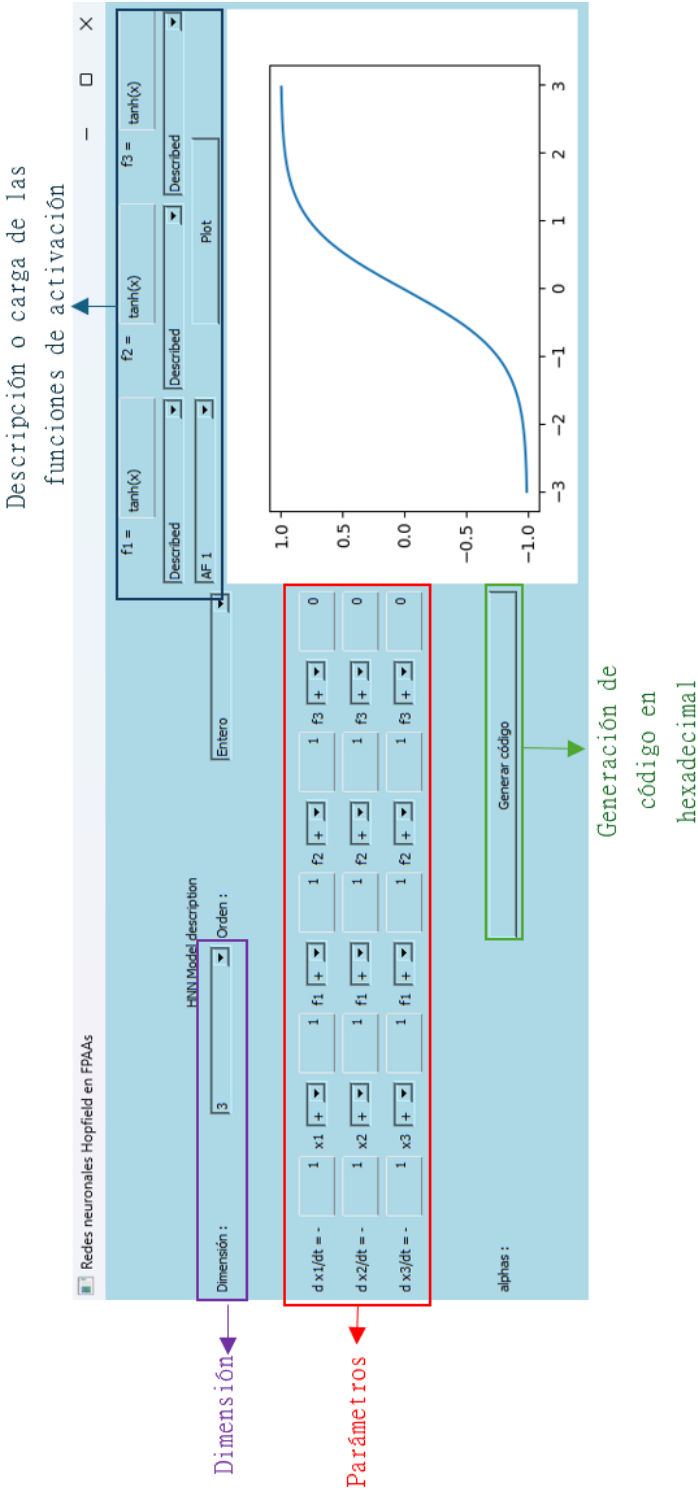


Figura 6.5: Interfaz desarrollada en Qt-Python para implementar diversos modelos de redes neuronales Hopfield.

Capítulo 7

Modelo de HNN con infinitos atractores

Esta sección da el análisis, desarrollo e implementación de un sistema caótico de infinitos atractores usando el modelo de red neuronal Hopfield de 3 neuronas. Para esto se retoman distintos tópicos explicados en capítulos pasados como son los puntos de equilibrio, eigenvalores, la representación del modelo HNN en filtros pasabajas, diagramas de bifurcación y exponentes de Lyapunov.

El desarrollo de sistemas caóticos de múltiples atractores no solo es un tema de interés debido a su significado en modelo de sistemas físicos, sino que también tienen una variedad de aplicaciones de la ingeniería y el su diseño ha sido previamente reportado [50].

Finalmente, en este capítulo se implementa el modelo propuesto en la tarjeta analógica embebida, cuya programación se genera en base a la interfaz desarrollada en el capítulo

7.1. El modelo propuesto

En la búsqueda de una simplificación en la implementación electrónica de un sistema HNN caótico se propone el siguiente modelo de red neuronal

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & w_{12} & 0 \\ 0 & 0 & w_{13} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} g(x_1) \\ x_2 \\ x_3 \end{bmatrix}, \quad (7.1)$$

donde $g(x_1)$ es una función no lineal. Notemos que las funciones de activación para x_2 y x_3 han sido sustituidas por la función identidad $f : x \mapsto x$. Esta propuesta considera los requerimientos mínimos para la existencia de caos, siendo estas la no linealidad y la irreducibilidad de un sistema dinámico de 3 a uno 2 dimensiones por medio de alguna transformación. El Jacobiano

del modelo propuesto esta dado por

$$J = \begin{bmatrix} -1 & w_{12} & 0 \\ 0 & -1 & w_{23} \\ w_{31}g'(x_1) & w_{32} & -1 + w_{33} \end{bmatrix}, \quad (7.2)$$

y alrededor de los PE podemos caracterizar el comportamiento del sistema no lineal al calcular los eigenvalores calculados por

$$|\lambda I - J| = \begin{vmatrix} \lambda + 1 & -w_{12} & 0 \\ 0 & \lambda + 1 & -w_{23} \\ -w_{31}g'(x_1) & -w_{32} & \lambda + 1 - w_{33} \end{vmatrix}, \quad (7.3)$$

que da la ecuación característica

$$\lambda^3 + (3 - w_{33})\lambda^2 + (3 - 2w_{33} - w_{23}w_{32})\lambda + \left(1 - w_{33} - w_{23}w_{32} - w_{12}w_{23}w_{31}\frac{\partial g(x_1)}{\partial x_1}\right) = 0. \quad (7.4)$$

En base a la función no lineal propuesta tenemos dos posibilidades: $g(x) = 0$ o $g(x) \gg 1$, para la primera se tiene que los eigenvalores están dados por

$$\lambda_1 = -1, \quad \lambda_{2,3} = \frac{w_{33}}{2} - 1 + \frac{\sqrt{w_{33}^2 + 4w_{23}w_{32}}}{2}, \quad (7.5)$$

de esta manera podemos decir que, para que todo PE para el cual $g(\hat{x}) = 0$ sea hiperbólico de índice 2 es necesario que se cumplan

$$w_{33} > 2, \quad w_{33}^2 + 4w_{23}w_{32} < 0. \quad (7.6)$$

Por otro lado para $\frac{\partial g(x)}{\partial x} \gg 1$ tenemos

$$\lambda_1 = \frac{w_{33}}{3} + \alpha_2 + \frac{\alpha_3}{\alpha_2} - 1, \quad (7.7)$$

$$\lambda_{2,3} = \frac{w_{33}}{3} - \frac{\alpha_2}{2} - \frac{\alpha_3}{2\alpha_2} \pm \alpha_1 - 1, \quad (7.8)$$

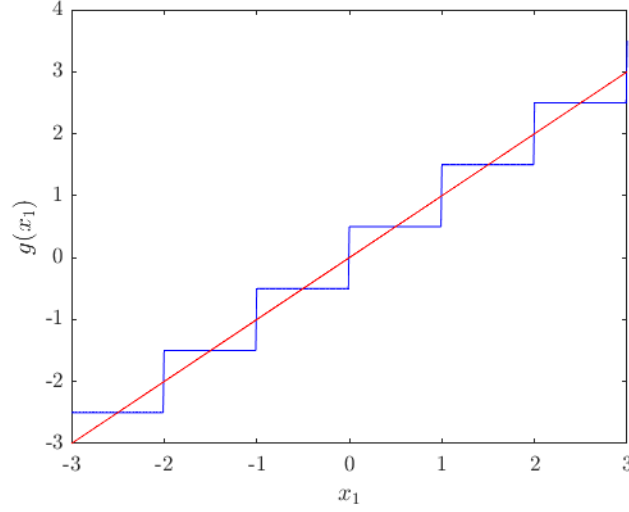


Figura 7.1: Función no lineal de activación propuesta.

donde

$$\alpha_1 = \frac{\left(\alpha_2 - \frac{\alpha_3}{\alpha_2}\right)^{1/3} i}{2}, \quad (7.9)$$

$$\alpha_2 = \left(\sqrt{\alpha_4^2 - \alpha_3^3} + \alpha_4\right)^{1/3} \quad (7.10)$$

$$\alpha_3 = \frac{w_{33}^2}{9} + \frac{w_{23}w_{32}}{3}, \quad (7.11)$$

$$\alpha_4 = \frac{w_{33}^3}{27} + \frac{w_{23}w_{32}w_{33}}{6} + \frac{w_{12}w_{23}w_{31}}{2} \frac{\partial g(x_1)}{\partial x}, \quad (7.12)$$

considerando que $\frac{\partial g(x_1)}{\partial x_1}$ es mucho mas grande que cualquiera de los pesos se tiene que $\alpha_4 \gg 1$, que a su vez implica que $\alpha_3 \ll \alpha_4$, ademas que $\alpha_2 \gg 1$, que a su vez indica que $\alpha_3 \ll \alpha_2$ y que por lo tanto implica que $\alpha_1 \gg 1i$. Sustituyendo estos resultados en ec. (7.8) tenemos

$$\lambda_1 \approx \alpha_2, \quad \lambda_{2,3} \approx -\frac{\alpha_2}{2} \pm \alpha_1, \quad (7.13)$$

y que por lo tanto indica que todo PE para el cual se cumple que $\frac{\partial g(x_1)}{\partial x_1} \gg 1$ es hiperbólico de índice 1. Es sabido en un sistema caótico que los atractores se forman cuando entre dos PE hiperbólicos de índice 2 (eigenvalor con real positivo y dos complejos conjugados con parte real negativa) hay un EP hiperbólico de índice 1 (eigenvalor con real negativo y dos complejos conjugados con parte real negativa) [50]. Es por esta razón que se propone la función no lineal

$$g(x_1) = \text{ceil}(x_1) - 0.5, \quad (7.14)$$

como la función no lineal implementada en el modelo, en este caso la función $\text{ceil}(x)$ por su parte esta definida por

$$\text{ceil}(x) = n, \quad \text{tal que } n - 1 < x \leq n, \text{ para todo } n \in \mathbb{Z}. \quad (7.15)$$

Para la ubicación de los puntos de equilibrio se debe resolver

$$-x_1 + w_{12}x_2 = 0, \quad (7.16)$$

$$-x_2 + w_{23}x_3 = 0, \quad (7.17)$$

$$-x_3 + w_{31}g(x_1) + w_{32}x_2 + w_{33}x_3 = 0, \quad (7.18)$$

por lo tanto se tiene $x_2 = -x_1/w_{12}$ y $x_3 = x_1/(w_{12}w_{23})$, sustituyendo en la última expresión de la ec. (7.18) se tiene

$$g(x_1) = \frac{1 + \frac{w_{32}}{w_{12}} - \frac{w_{33}}{w_{12}w_{23}}}{w_{31}}x_1, \quad (7.19)$$

y por lo tanto los puntos de equilibrio del sistema propuesto dependen la cantidad de veces que la función no lineal g cruce por la recta de pendiente $\frac{1 + \frac{w_{32}}{w_{12}} - \frac{w_{33}}{w_{12}w_{23}}}{w_{31}}$. Si selecciona $w_{12} = w_{23} = -1$, $w_{31} = 1$ y $w_{32} = w_{33}$ se tiene que los puntos de equilibrio del sistema son dependientes de

$$g(x_1) = x_1, \quad (7.20)$$

$$x_2 = -x_1, \quad (7.21)$$

$$x_3 = x_1. \quad (7.22)$$

Considerando esta función no lineal se tiene un punto de equilibrio cada $k \in \mathbb{Z}$ en $(\frac{k}{2}, -\frac{k}{2}, \frac{k}{2})$ como se observa en la Figura 7.1, en la cual se grafica la función no lineal $g(x)$ en azul y en rojo la recta que representa la función identidad y cuyos cruces indican el valor del EP en x_1 . Además, bajo los pesos considerados se puede retomar la segunda desigualdad de la ec. (7.6) para obtener

$$w_{33}(w_{33} - 4) > 0 \quad \Rightarrow \quad w_{33} < 4, \quad (7.23)$$

de esta manera los valores para w_{32} y w_{33} deben ser mayores a 2 pero menos a 4.

7.2. Retratos de fase, diagramas de bifurcación y exponentes de Lyapunov

Tras el análisis desarrollada el modelo de HNN con infinitos atractores queda definido por

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} g(x_1) \\ x_2 \\ x_3 \end{bmatrix}, \quad (7.24)$$

el cual al aplicar el operador divergencia da

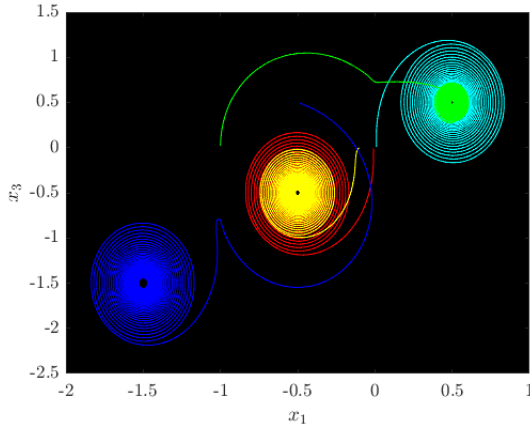
$$\nabla \cdot \mathbf{f} = -3 + w_{33}, \quad (7.25)$$

al ser una constante se tiene como criterio para la existencia de atractores que $w_{33} < 3$.

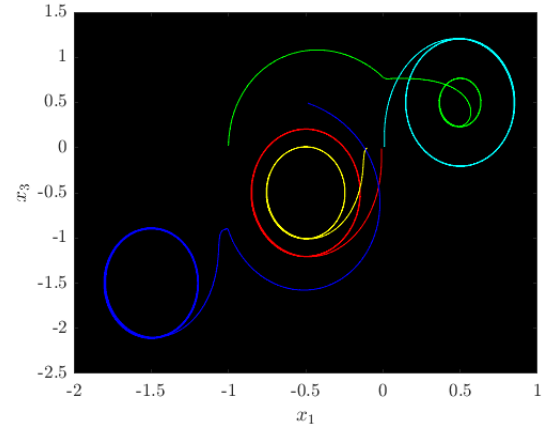
Al analizar el sistema numéricamente se encontró el fenómeno de multiestabilidad, para que este ocurra el estado de un sistema debe ser dependiente de la condición inicial aun en zonas no caóticas. Para ilustrar este comportamiento se desarrollaron los retratos de fase de la Figura 7.2, en donde se inicio el sistema con las condiciones iniciales siguientes: $(0.01, 0, 0)$ en celeste, $(-0.01, 0, 0)$ en rojo, $(-1, 1, 0)$ en verde, $(-0.1, -0.2, 0)$ en amarillo y $(-0.5, -0.2, 0.5)$ en azul, como se observa cada condición inicial lleva a un comportamiento distinto.

Por otra parte, en busca de un comportamiento denso y sabiendo que $w_{32} = w_{33}$ se desarrollo el diagrama de bifurcación de máximos locales que se muestra en la Figura 7.2a. En este diagrama de bifurcación se observa que a partir de $w_{33} > 2$ existe un pequeño periodo de periodicidad para después volverse un sistema denso. Para valores de $w_{33} < 2$ se encuentra un comportamiento de punto fijo mientras que para valores de $w_{33} > 2.75$ el sistema se vuelve desacotado. Además, todos los atractores se presentan en los puntos de equilibrio hallados y por lo tanto se dicen auto-excitados. Un comportamiento hallado muy interesante fue la extrema multiestabilidad del sistema dinámico propuesto, esto es, la respuesta del sistema es diferente dadas ciertas condiciones iniciales. El retrato de fase en tres dimensiones es presentado en la Figura 7.2 donde se forma un total de 15 atractores en un tiempo de 1500 segundos de simulación, sin embargo, esta cantidad de enrollamientos depende del propio tiempo, para simulaciones mas largas se halla una mayor cantidad de atractores.

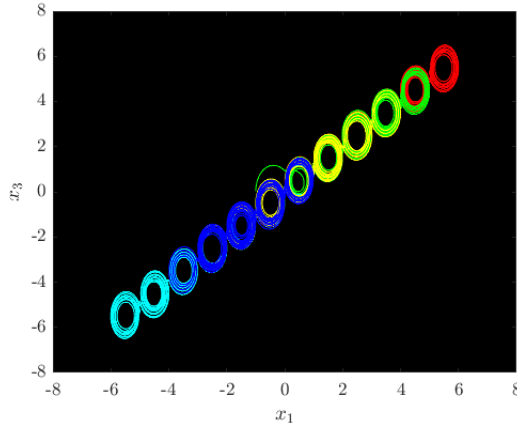
Para identificar si la zona densa identificada por el diagrama de bifurcación es caótica se hallaron los exponentes de Lyapunov variando ambos parámetros w_{23} y w_{33} . Para la obtención de los LEs la función $\text{ceil}(x)$ genera problemas en el algoritmo debido a su no discontinuidad en cada entero, es por esta razón que para la obtención de los LEs se desarrollo la función



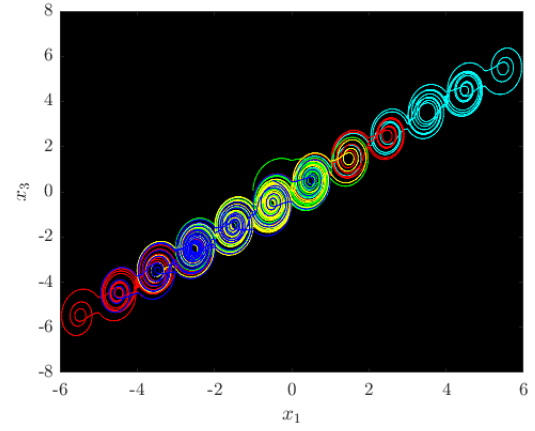
(a) Plano $x_1 - x_3$ del modelo propuesto con $w_{32} = w_{33} = 1.98$.



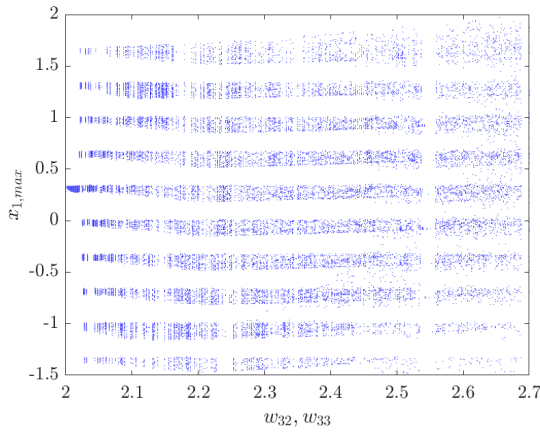
(b) Plano $x_1 - x_3$ del modelo propuesto con $w_{32} = w_{33} = 2$.



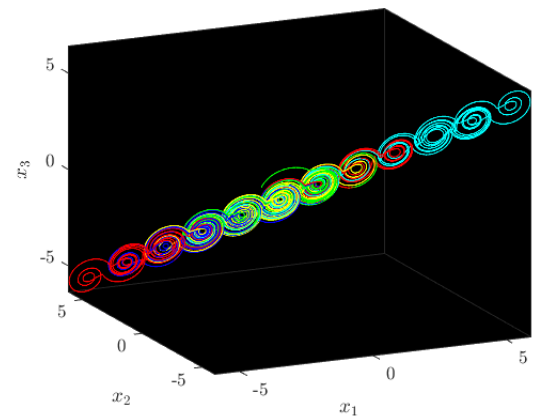
(c) Plano $x_1 - x_3$ del modelo propuesto con $w_{32} = w_{33} = 2.05$.



(d) Plano $x_1 - x_3$ del modelo propuesto con $w_{32} = w_{33} = 2.2$.



(e) Diagrama de bifurcación.



(f) Retrato de fase tridimensional del modelo propuesto.

Figura 7.2: a) - d) Retratos de fase del sistema propuesto variando los parámetros $w_{32} = w_{33}$; e) diagrama de bifurcación de máximos locales con condiciones iniciales $(0.1, 0, 0)$ y limitando el sistema a un total de 10 atractores; y e) retrato de fase tridimensional con $w_{32} = w_{33} = 2.2$.

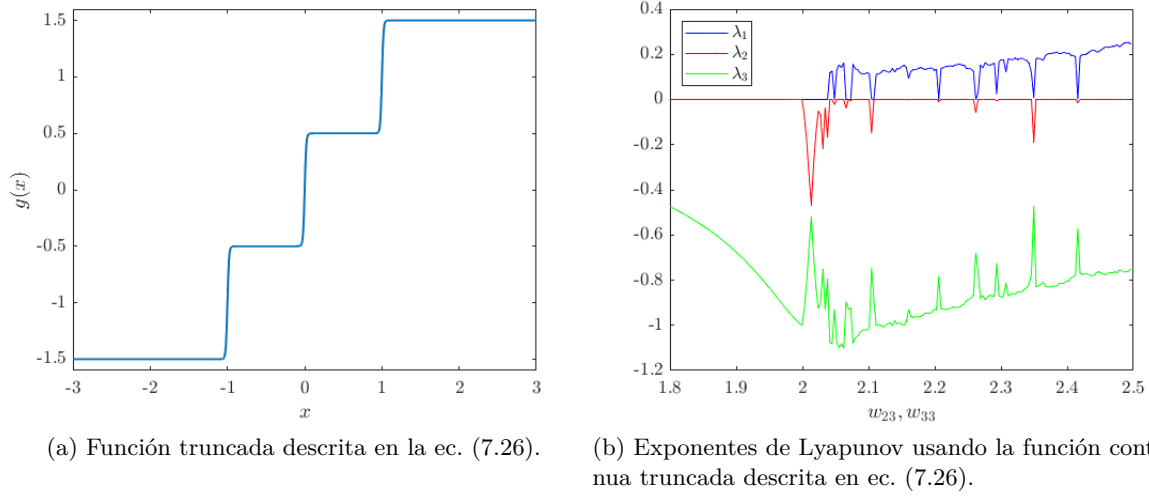


Figura 7.3: a) Función truncada utilizada para el cálculo numérico de los exponentes de Lyapunov mostrados en b).

continua descrita por

$$g(x) = -1.5 \frac{1}{1 + e^{k(x+1)}} + 0.5 \left(\frac{1}{1 + e^{-kx}} - \frac{1}{1 + e^{-k(x+1)}} \right) + 0.5 \left(-\frac{1}{1 + e^{-k(x-1)}} + \frac{1}{1 + e^{-kx}} \right) + 1.5 \frac{1}{1 + e^{-k(x-1)}}, \quad (7.26)$$

donde k es un número positivo mucho mayor a 1. La grafica de esta función es presentada en la Figura 7.3a en donde se evidencia que la función continua diseñada es un truncamiento de la función inicialmente propuesta en la ec. (7.14) a solo 4 escalones, sin embargo, esto solo afecta a la cantidad de enrollamientos del sistema. Los exponentes encontrados con la función truncada son presentados en la Figura 7.3b, el exponente máximo concuerda con la zona densa encontrada por los diagramas de bifurcación y entonces se puede afirmar que el sistema es caótico. Además, los exponentes para valores de $w_{23} = w_{33}$ concuerdan con los retratos antes mostrados en donde se haya un comportamiento de punto fijo para $w_{33} < 2$ y ciclo limite en $w_{33} = 2$.

7.3. Implementación en un ambiente electrónico

La implementación electrónica del modelo de infinitos atractores propuesto fue desarrollada utilizando la interfaz presentada en la sección anterior, la configuración es en base a los pesos y funciones de activación del modelo en la ec. (3.25). Con lo que respecta a la función no lineal, su implementación en la tarjeta analógica requirió de la opción de "Load" en la sección

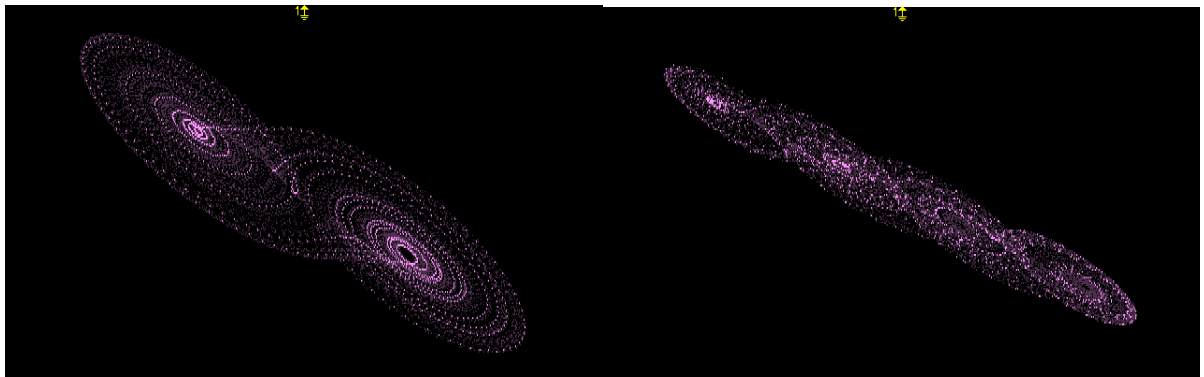
de funciones de activación de la interfaz, en la cual se debe cargar un archivo en formato de texto con 256 valores asignados para una entrada entre voltajes de -3V a 3V. En la interfaz se cargaron varios archivos con funciones de activación distintas pero basadas en la presentada con anterioridad. Para la creación de un modelo con doble enrollamiento se utilizo la función descrita por

$$g(x) = -0.5 \left(\frac{1}{1 + e^{kx}} \right) + 0.5 \left(\frac{1}{1 + e^{-kx}} \right) \quad (7.27)$$

para la creación de un modelo con 4 enrollamientos se utilizo la función truncada descrita en (7.26), mientras que para un sistema de 8 enrollamientos se utilizo la función

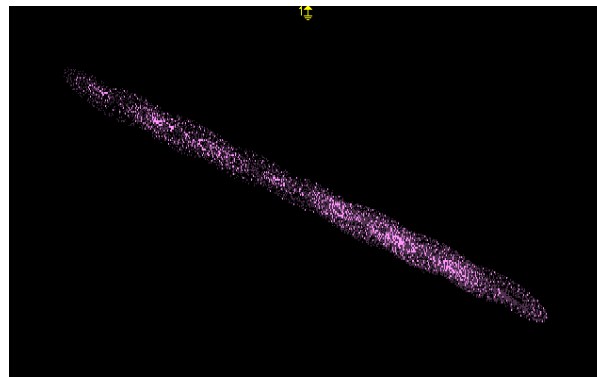
$$\begin{aligned} g(x) = & -1.5 \left(\frac{1}{1 + e^{k(x+\frac{9}{7})}} \right) - \frac{15}{14} \left(\frac{1}{1 + e^{-k(x+\frac{9}{7})}} - \frac{1}{1 + e^{-k(x+\frac{6}{7})}} \right) - \\ & \frac{9}{14} \left(\frac{1}{1 + e^{-k(x+\frac{6}{7})}} - \frac{1}{1 + e^{-k(x+\frac{3}{7})}} \right) - \frac{3}{14} \left(\frac{1}{1 + e^{-k(x+\frac{3}{7})}} - \frac{1}{1 + e^{-k(x)}} \right) + \\ & \frac{3}{14} \left(\frac{1}{1 + e^{-k(x)}} - \frac{1}{1 + e^{-k(x-\frac{3}{7})}} \right) + \frac{9}{14} \left(\frac{1}{1 + e^{-k(x-\frac{3}{7})}} - \frac{1}{1 + e^{-k(x-\frac{6}{7})}} \right) + \\ & \frac{15}{14} \left(\frac{1}{1 + e^{-k(x-\frac{6}{7})}} - \frac{1}{1 + e^{-k(x-\frac{9}{7})}} \right) + 1.5 \left(\frac{1}{1 + e^{-k(x-\frac{9}{7})}} \right) \end{aligned} \quad (7.28)$$

El desarrollo de estas múltiples funciones es debido a las limitaciones en hardware, recordando que los FPAA utilizados trabajan con voltajes de -3V a 3V. Los resultados de la implementación son presentados en la Figura 7.4 mostrando el retrato de fase para el plano $x_1 - x_2$ usando las múltiples funciones de activación propuesta.



(a) Sistema 3D-HNN con 2 enrollamientos.

(b) Sistema 3D-HNN con 4 enrollamientos.



(c) Sistema 3D-HNN con 8 enrollamientos.

Figura 7.4: Implementación en electrónica del modelo de HNN con múltiples enrollamientos.

Conclusiones

Conclusiones del avance III

En el Avance de Tesis III se desarrollo un algoritmo para la obtención de la media de la divergencia considerando la posibilidad que el sistema sea desacotado, para probar que el sistema es acotado numéricamente, se elaboró un teorema en donde se demuestra que la norma de la media de la velocidad de las variables de estado se aproxima a cero conforme el sistema evoluciona. Con este algoritmo se demuestra que los modelos HNN estudiados son acotados y disipativos, es decir, contienen atractores.

De la literatura se encontró un modelo de red neuronal Hopfield de 4 dimensiones con u comportamiento hipercaótico. De dicho modelo se verificaron los resultados de la literatura obteniendo los respectivos diagramas de bifurcación y exponentes de Lyapunov confirmando el comportamiento afirmado. En base a este modelo se desarrollaron las ecuaciones de diseño para la implementación en la tarjeta analógica embebida, esta implementación fue realizada en base a filtros analógicos. Dicho diseño fue implementado en FPAAs con resultados satisfactorios.

Se desarrolló una interfaz en Python como metodología para la implementación de la redes neuronal Hopfield de 3 neuronas con y sin señal inhibitoria externa. Para el desarrollo de esta interfaz fue necesario de la elaboración de distintas topologías de circuitos descritas en código base a bloques en el ambiente de programación de AN2, estas topologías generan un archivo de texto con los módulos, su configuración y la configuración de la tarjeta, y por lo tanto, puede ser editado para ser controlado por la interfaz.

Finalmente, se diseño un modelo HNN el cual puede contener infinitos atractores, lo cual ofrece múltiples dinámicas a la red neuronal, entre ellas la multiestabilidad. Para el diseño de esta red se utilizaron herramientas previamente establecidas y se hizo especial énfasis en diseñar tipos especiales de puntos de equilibrio. Esta misma red fue implementada usando la interfaz diseñada en Python con resultados adecuados.

Ponencia en congreso y publicación



THE FACULTY OF ENGINEERING AT UNIVERSIDAD PANAMERICANA CERTIFY THAT

Diego Ruiz Sánchez de la Vega

Has participated as speaker in a talk entitled

“Rich dynamics of a Hopfield neural network and its analog circuit realization”

During the 4th Meeting for the Dissemination and Research in the Study of Complex Systems and their Applications (EDIESCA 2024), Awarded in Aguascalientes México, October 28-30, 2024.

Mtra. Mercedes Urzúa González
Dean Faculty of Engineering
Universidad Panamericana
Campus Aguascalientes

Dr. Guillermo Huerta Cuellar
Head of Asociación Mexicana de
Sistemas Dinámicos y Complejidad

Dr. Héctor Eduardo Gilardi Velázquez
Head Chair EDIESCA2024
Faculty of Engineering
Universidad Panamericana
Campus Aguascalientes

Rich dynamics and analog implementation of a Hopfield neural network in integer and fractional order domains

Diego S. de la Vega^a, Jesus M. Munoz-Pacheco^{*a}, Olga G. Félix-Beltrán^a, Christos Volos^b

^aFaculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla, Puebla, México

^bLaboratory of Nonlinear Circuits, Systems Complexity (LaNSCom), Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract

Several synaptic weight matrices have been proposed for Hopfield neural network (HNN) models, where chaotic dynamics may arise. Contrary to those works, this manuscript aims to present a synaptic weight matrix where every entry can be set as an integer, harvesting an elegant chaotic HNN from a chaos theory point of view. Analytical and numerical analyses such as equilibrium points, bifurcation diagrams, Lyapunov exponents, and basins of attraction demonstrate that the proposed HNN exhibits complex behaviors across a wide range of parameter values. Also, we extend the study of the HNN into the fractional order domain. Moreover, the design and implementation details of the proposed neural network using field programmable analog arrays (FPAAs) are thoroughly discussed. This includes the various components and their configurations, highlighting how they contribute to the overall functionality of the neural network. As a result, we found a strong correlation between numerical simulations and SPICE circuit simulations.

Keywords: Hopfield neural network, Chaotic systems, fractional calculus, analog circuit, FPAAs.

1. Introduction

Mathematical models of nonlinear dynamical systems are present in almost every aspect of science since they can model most time-evolving phenomena, such as chaos. Chaotic dynamics manifest in many natural systems, including classical mechanical systems [1], electronic circuits [2], biological systems [3], among others. Indeed, mathematical models for artificial neural networks generating chaos have also been proposed in recent works [4]. Artificial Neural Networks (ANNs) are inspired by biological neural networks defined as a system of elements called neurons that can send, receive, and compute signals and exhibit rich and complicated dynamics like the human brain [5]. In literature, one can find different models for ANNs, such as convolutional, recurrent, feed-forward, de-convolutional, and so forth. Among them, Hopfield Neural Networks (HNNs) are a key focus of extensive research in the field of ANNs. First proposed by J. Hopfield [6], HNNs are applied for pattern recognition but with a simple architecture. In addition, the chaotic phenomena may also arise in HNNs. Recent investigations have discovered chaotic [3], hyperchaotic [2] behaviors with single-scroll attractors and double-scroll attractors [7]. For instance, Bao et al. [3] established a tri-neuron network (3D-HNN) with chaotic behavior, where the results were demonstrated theoretically and experimentally by dynamical tools and electronic devices, respectively. Apart from the network topology, the entries of the synaptic weight matrix are

fundamental for chaos generation. Therefore, chaotic behavior with different values for the weight matrix has been proposed and experimentally validated using analog or digital circuits.

From a chaos point of view, Sprott [8] proposed that new chaotic systems must have at least one of the following criteria:

- The system should model some unsolved problem in nature.
- The system should exhibit some unseen behavior.
- The system should be simpler than all other known examples exhibiting the observed behavior.

Simplicity in a system refers not only to the number of parameters and variables but also to the digits in the parameters of a chaotic system. In this framework, we have found an integer-valued synaptic weight matrix generating chaotic patterns in the HNN, yielding an elegant neural network satisfying the third Sprott criteria.

On the other hand, fractional-order calculus is a mathematical branch considered a generalization of integer-order calculus [9]. In recent years, fractional calculus has become a relevant tool for modeling and offering new dynamics to new and previously proposed systems. For instance, the study of the fractional version of Chua's [10], Lorenz [11], and Rossler [12] systems. There is a tendency in recent years to explore the fractional order version of a chaotic system to investigate its implications in chaos generation [13, 14, 15] since fractional calculus provides an extra degree of freedom in chaotic systems, resulting in undiscovered dynamics. In this manner, fractional-order features allow us to describe natural phenomena and provide

^{*}Corresponding author

Email address: jesum.pacheco@correo.buap.mx (Jesus M. Munoz-Pacheco)

29/11/24, 1:36 p.m.

Correo: DIEGO RUIZ SANCHEZ DE LA VEGA - Outlook



Confirm co-authorship of submission to Integration

Desde em.vlsij.0.8f73c3.6540c0a0@editorialmanager.com <em.vlsij.0.8f73c3.6540c0a0@editorialmanager.com>
en nombre de
 Integration <em@editorialmanager.com>

Fecha Sáb 16/11/2024 0:33

Para DIEGO RUIZ SANCHEZ DE LA VEGA <diego.ruizsa@alumno.buap.mx>

No suele recibir correo electrónico de em@editorialmanager.com. [Por qué es esto importante](#)

This is an automated message.

Journal: Integration

Title: Rich dynamics and analog implementation of a Hopfield neural network in integer and fractional order domains

Corresponding Author: Dr Jesus Munoz-Pacheco

Co-Authors: Diego S. de la Vega; Olga G. Félix-Beltrán; Christos Volos

Manuscript Number: **VLSIJ-D-24-00665**

Dear Diego S. de la Vega,

The corresponding author Dr Jesus Munoz-Pacheco has listed you as a contributing author of the following submission via Elsevier's online submission system for Integration.

Submission Title: Rich dynamics and analog implementation of a Hopfield neural network in integer and fractional order domains

Elsevier asks all authors to verify their co-authorship by confirming agreement to publish this article if it is accepted for publication.

Please read the following statement and confirm your agreement by clicking on this link: [Yes, I am affiliated.](#)

I irrevocably authorize and grant my full consent to the corresponding author of the manuscript to: (1) enter into an exclusive publishing agreement with Elsevier on my behalf (or, if the article is to be published under a CC BY license, a non-exclusive publishing agreement), in the relevant form set out at www.elsevier.com/copyright; and (2) unless I am a US government employee, to transfer my copyright or grant an exclusive license of rights (or for CC BY articles a non-exclusive license of rights) to Elsevier as part of that publishing agreement, effective on acceptance of the article for publication. If the article is a work made for hire, I am authorized to confirm this on behalf of my employer. I agree that the copyright status selected by the corresponding author for the article if it is accepted for publication shall apply and that this agreement is subject to the governing law of the country in which the journal owner is located.

If you did not co-author this submission, please contact the corresponding author directly at

<https://outlook.office.com/mail/inbox/id/AAQkADQxZmY2M2VjLTUzNDU0NDNBIZi1iN2JlTU0ZmFIZWU4NWQ4NAAQACs9dbygXCZJnUjSpC8e7Bc%3D>

1/2

Anexos

.1. Codigos en AnadigmDesigner2 para la implementación de las redes neuronales de 3 dimensiones

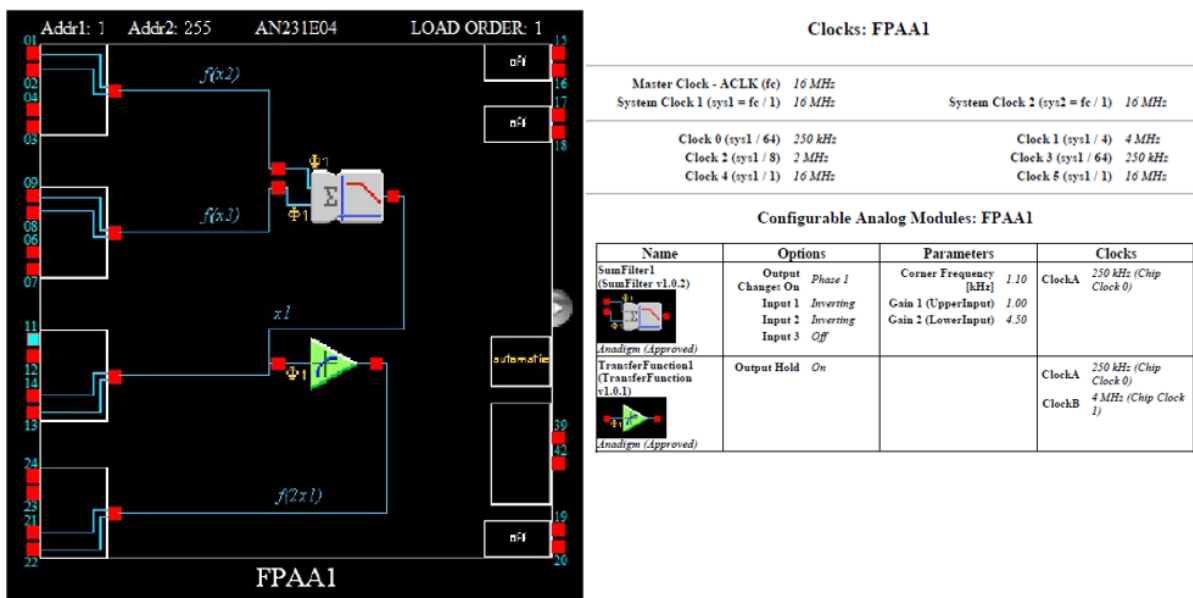


Figura 5: Programación y configuración del FPPA 1 para el modelo con función tangente hiperbólica.

1. CODIGOS EN ANADIGMDESIGNER2 PARA LA IMPLEMENTACIÓN DE LAS REDES NEURONALES

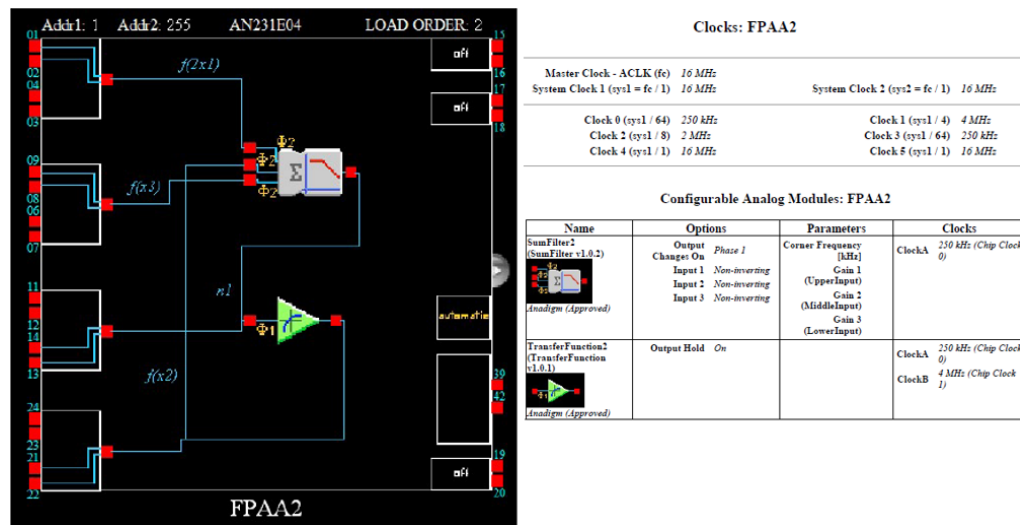


Figura 6: Programación y configuración del FPPA 2 para el modelo con función tangente hiperbólica.

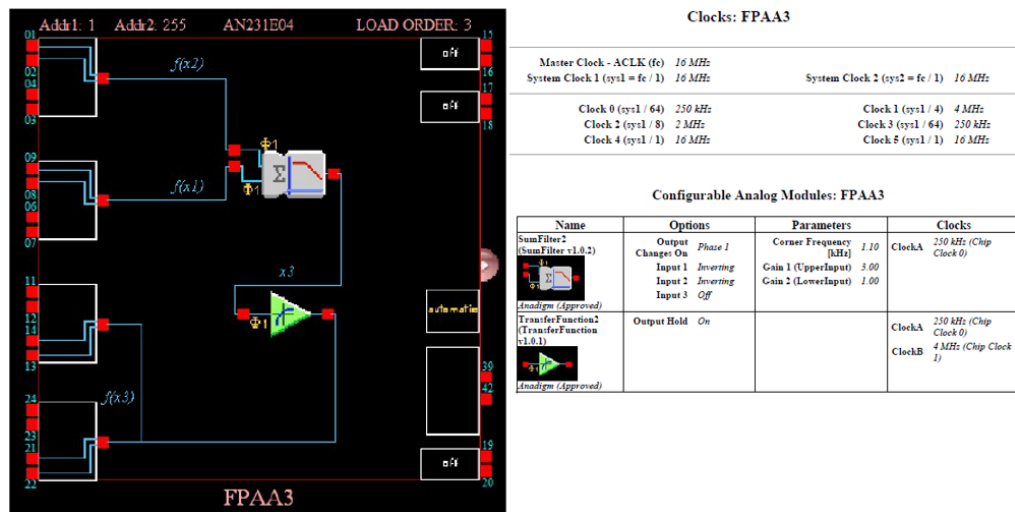


Figura 7: Programación y configuración del FPPA 3 para el modelo con función tangente hiperbólica.

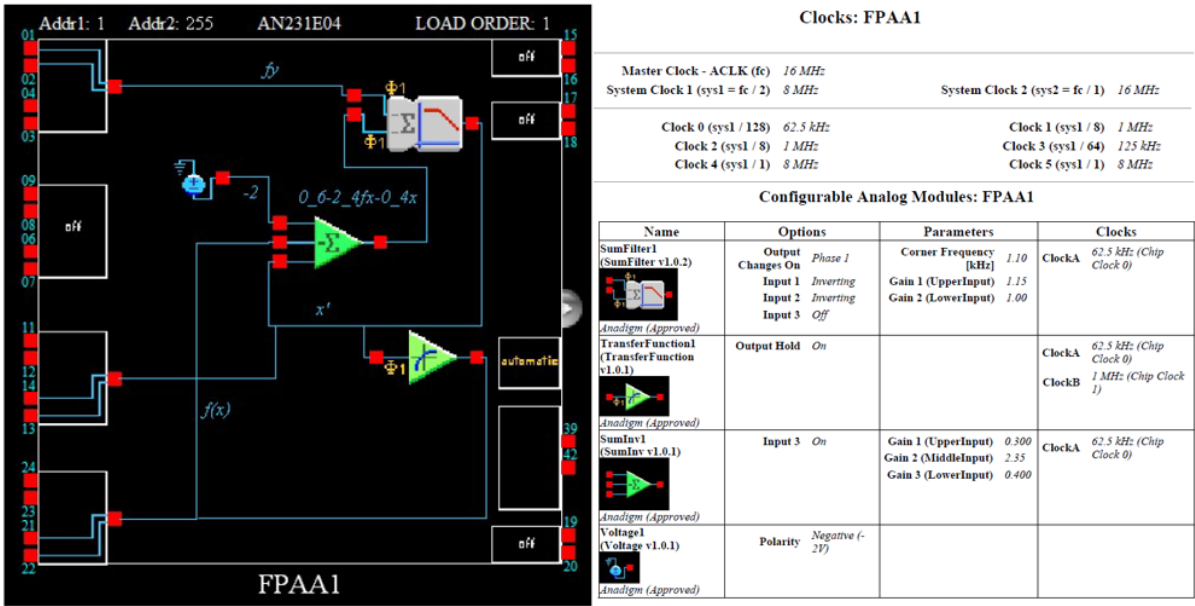


Figura 8: Programación y configuración del FPPA 1 para el modelo con función sigmoide.

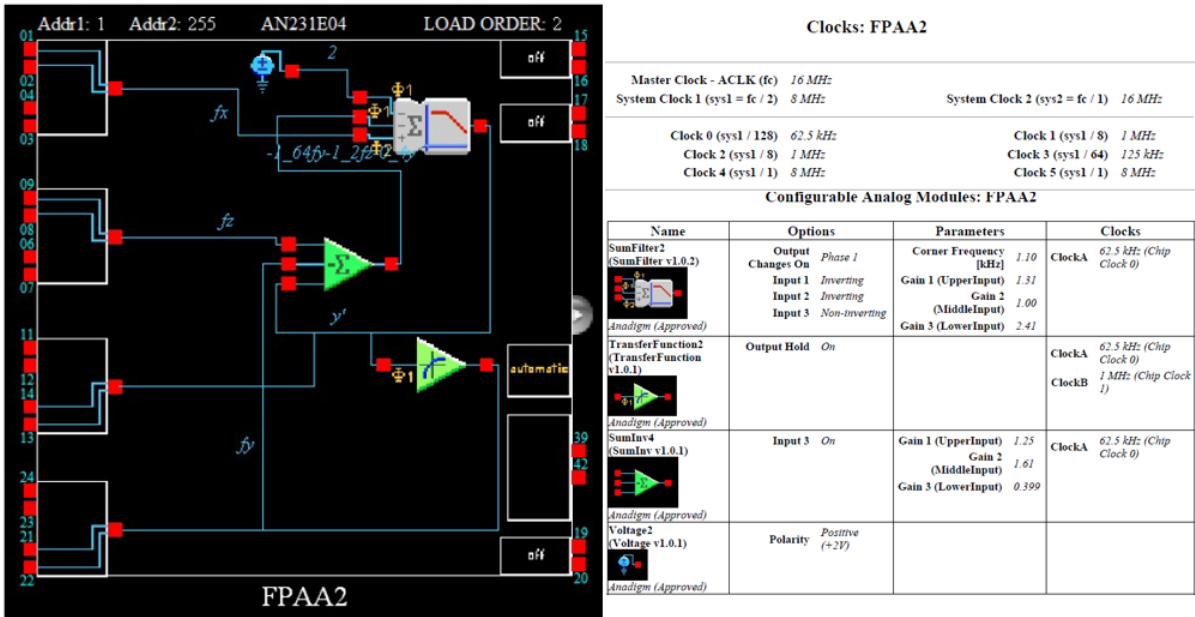


Figura 9: Programación y configuración del FPPA 2 para el modelo con función sigmoide.

1. CODIGOS EN ANADIGMDESIGNER2 PARA LA IMPLEMENTACIÓN DE LAS REDES NEURONALES

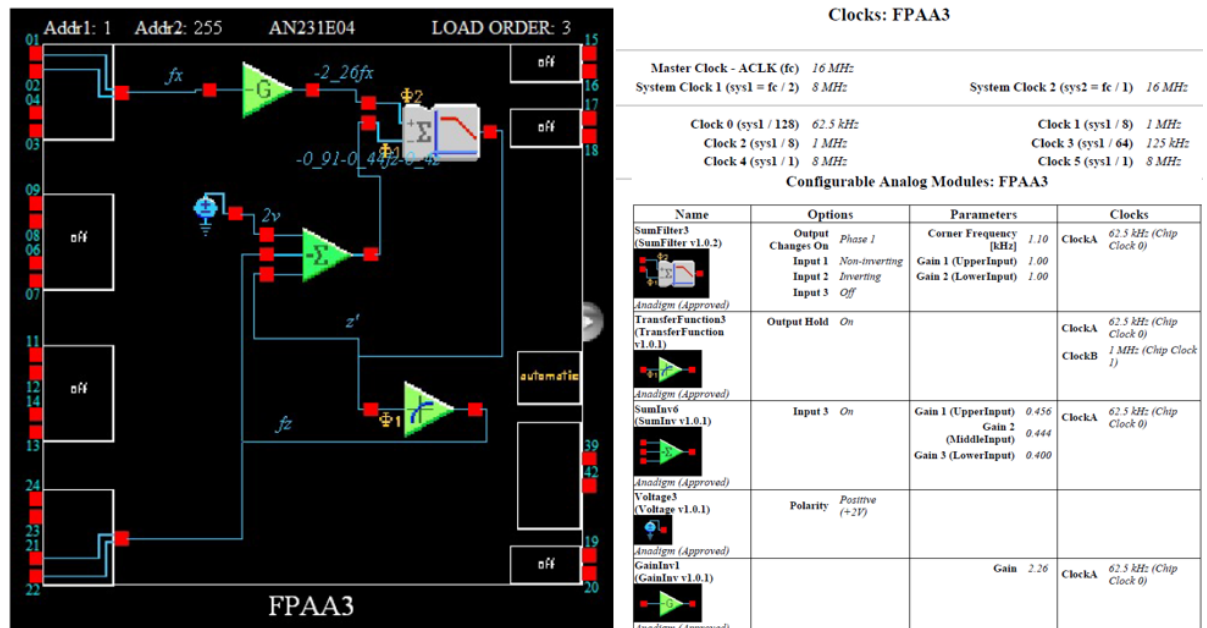


Figura 10: Programación y configuración del FPPA 3 para el modelo con función sigmoide.

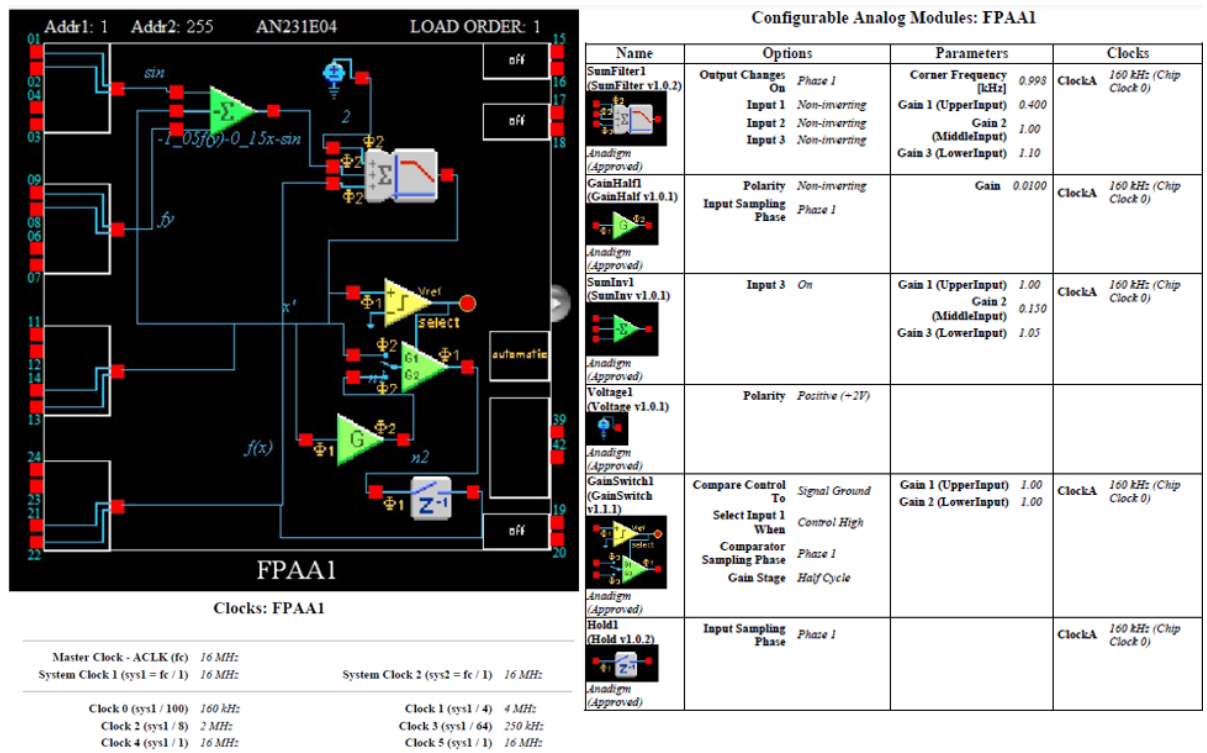


Figura 11: Programación y configuración del FPPA 1 para el modelo con función ReLU.

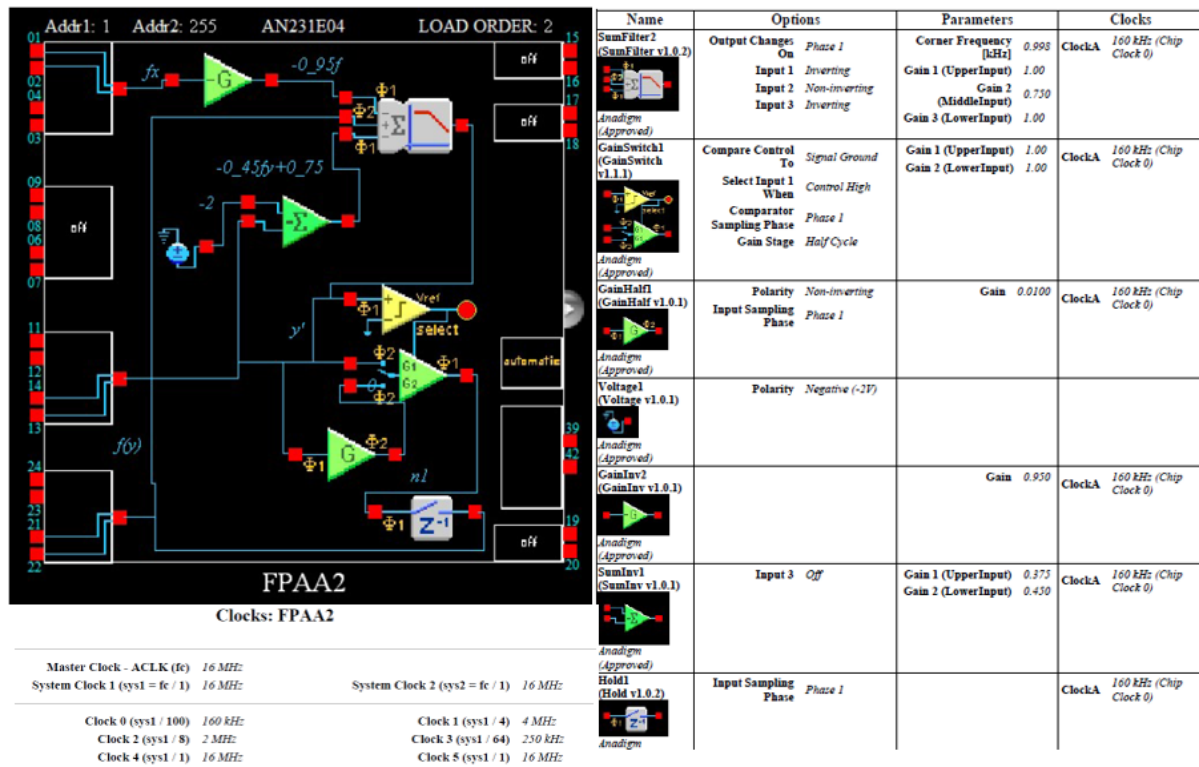


Figura 12: Programación y configuración del FPPA 2 para el modelo con función ReLU.

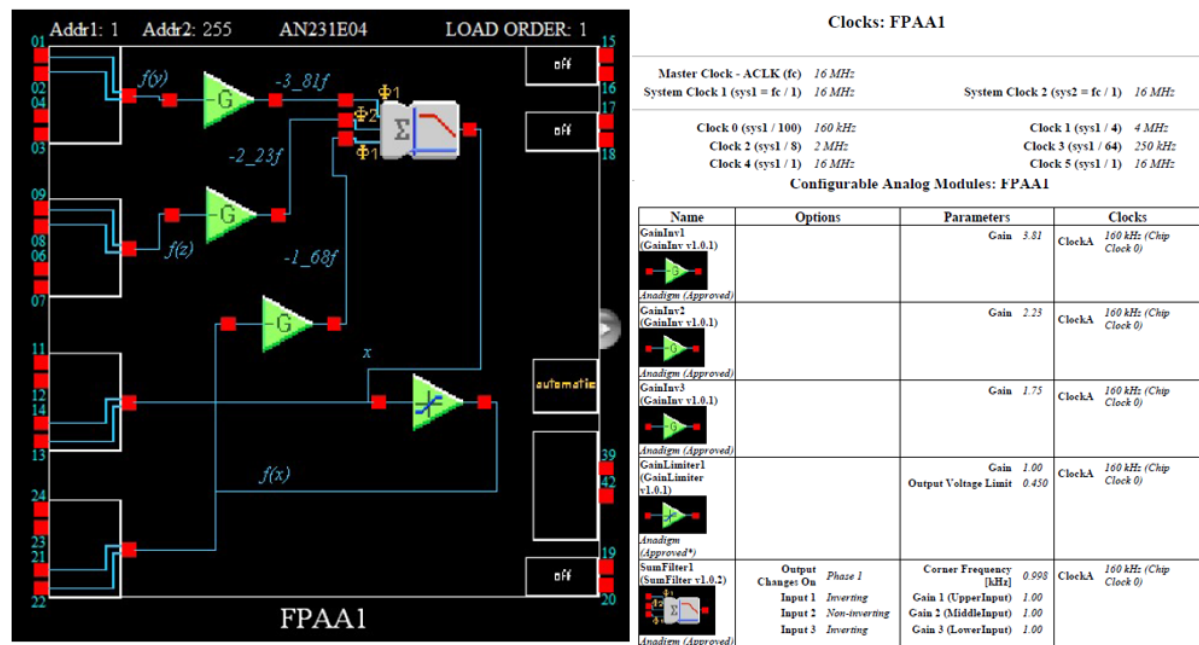


Figura 13: Programación y configuración del FPPA 1 para el modelo con función PWL.

1. CODIGOS EN ANADIGMDESIGNER2 PARA LA IMPLEMENTACIÓN DE LAS REDES NEURONALES

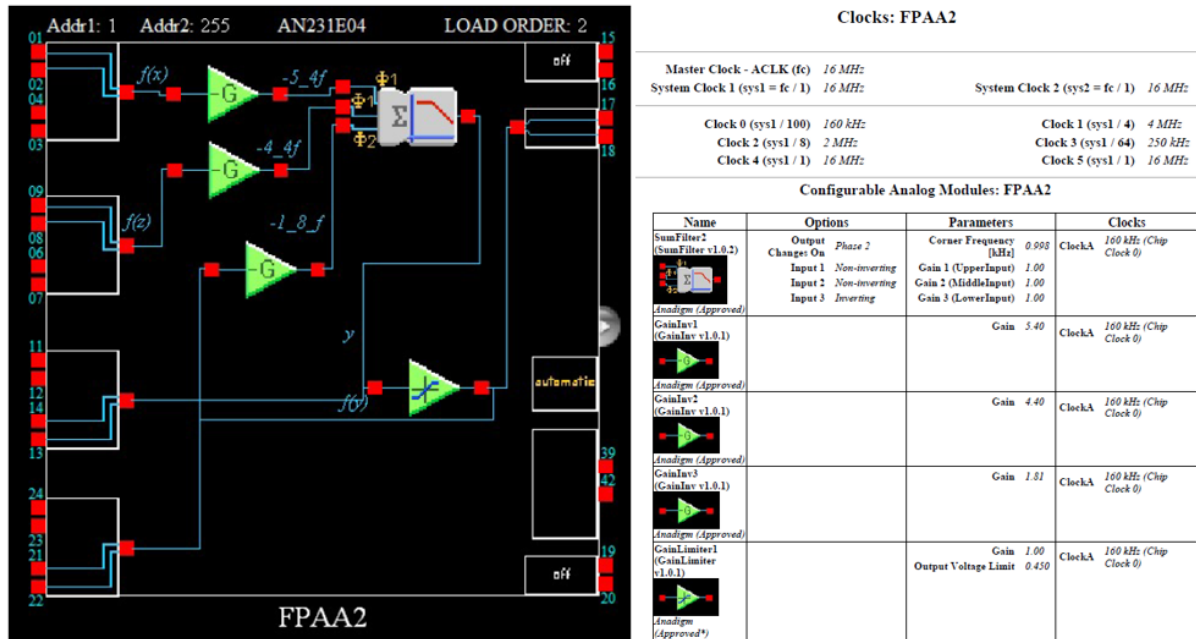


Figura 14: Programación y configuración del FPPA 2 para el modelo con función PWL.

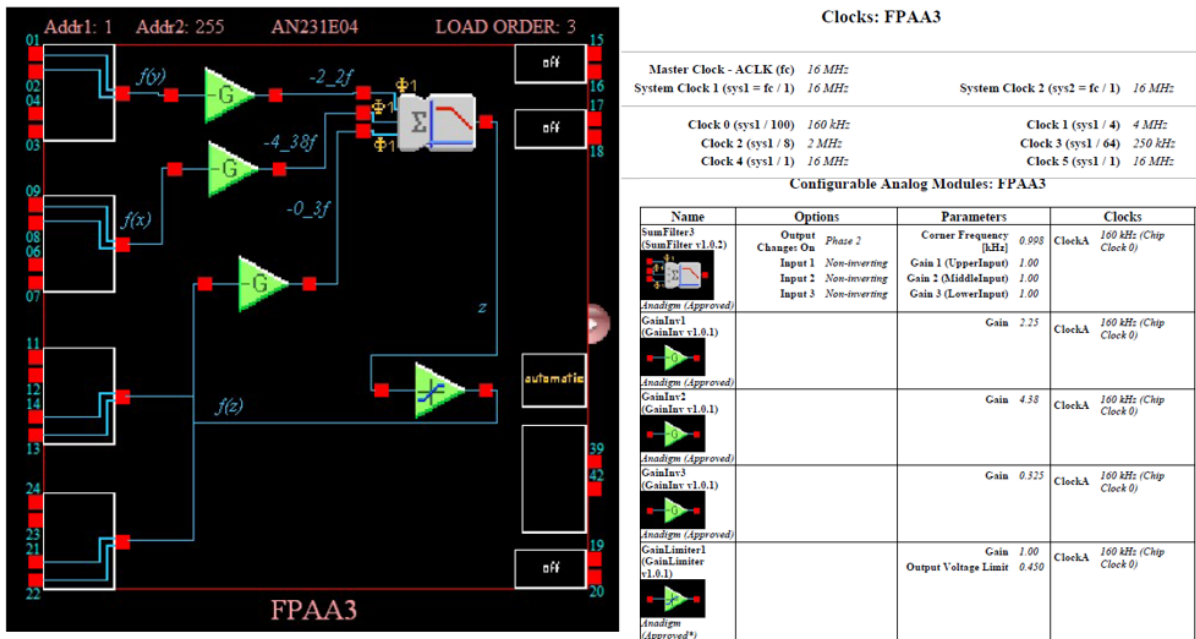


Figura 15: Programación y configuración del FPPA 3 para el modelo con función PWL.

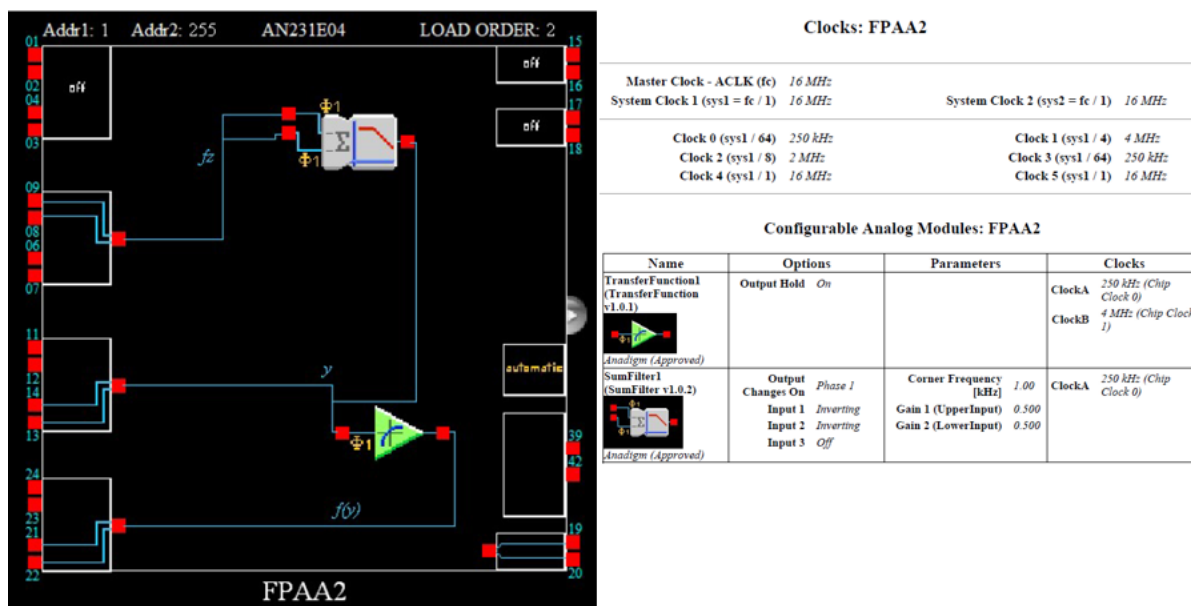
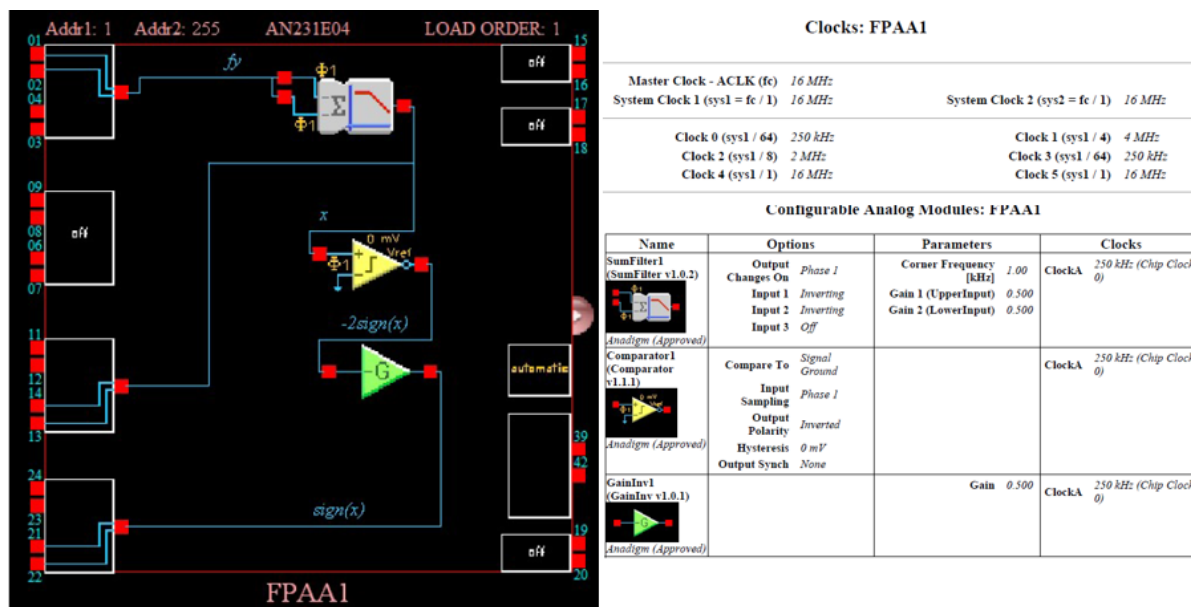


Figure 1 shows the Proteus ISIS software interface for the FPAA3 board. The board is populated with various components including a microcontroller, memory, and various analog modules. The software interface shows the component list on the left, the board layout in the center, and the component properties on the right. The board is labeled 'FPAA3' at the bottom.

Figura 18: Programación y configuración del FPPA 3 para el modelo con función signo.

Bibliografía

- [1] Iqbal M. Batiha, Ramzi B. Albadarneh, Shaher Momani, and Iqbal H. Jebril. Dynamics analysis of fractional-order Hopfield neural networks. *International Journal of Biomathematics*, 13(08):2050083, December 2020.
- [2] Chengjie Chen, Fuhong Min, Yunzhen Zhang, and Han Bao. ReLU-type Hopfield neural network with analog hardware implementation. *Chaos, Solitons & Fractals*, 167:113068, February 2023.
- [3] Zheng Peng-Sheng, Tang Wan-Sheng, and Zhang Jian-Xiong. A new chaotic Hopfield network with piecewise linear activation function. *Chinese Physics B*, 19(3):030514, March 2010.
- [4] Nathan W. Churchill, Robyn Spring, Cheryl Grady, Bernadine Cimprich, Mary K. Askren, Patricia A. Reuter-Lorenz, Mi Sook Jung, Scott Peltier, Stephen C. Strother, and Marc G. Berman. The suppression of scale-free fMRI brain dynamics across three different sources of effort: aging, task novelty and task difficulty. *Scientific Reports*, 6(1):30895, August 2016.
- [5] Bruce Alberts. *Molecular biology of the cell*. W. W. Norton & Company, New York, seventh edition edition, 2022.
- [6] Luis Alfredo Baratas Díaz. La obra neuro-embriológica de Santiago Ramón y Cajal. *Historiam Illustrandam*, 1(17):259–279, 1997.
- [7] N. B. Karayiannis and A. N. Venetsanopoulos. *Artificial Neural Networks*. Springer US, Boston, MA, 1993.
- [8] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [9] Charu C. Aggarwal. *Neural networks and deep learning: a textbook*. Springer, Cham, second edition edition, 2023.

- [10] Liangqu Long and Xiangming Zeng. *Beginning deep learning with TensorFlow work with Keras, MNIST data sets, and advanced neural networks*. Apress, United States, 2022.
- [11] John J. Hopfield and David W. Tank. Computing with Neural Circuits: A Model. *Science*, 233(4764):625–633, August 1986.
- [12] Anke Meyer-Baese and Volker Schmid. Transformation and Signal-Separation Neural Networks. In *Pattern Recognition and Signal Analysis in Medical Imaging*, pages 245–289. Elsevier, 2014.
- [13] Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal Hopfield Networks: A General Framework for Single-Shot Associative Memory Models. 2022.
- [14] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need. 2020.
- [15] Karthikeyan Rajagopal, Jesus M. Munoz-Pacheco, Viet-Thanh Pham, Duy Vo Hoang, Fawaz E. Alsaadi, and Fuad E. Alsaadi. A Hopfield neural network with multiple attractors and its FPGA design. *The European Physical Journal Special Topics*, 227(7-9):811–820, October 2018.
- [16] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, March 2021.
- [17] Daniel Clemente-López, Jesus M. Munoz-Pacheco, and Jose De Jesus Rangel-Magdaleno. A Review of the Digital Implementation of Continuous-Time Fractional-Order Chaotic Systems Using FPGAs and Embedded Hardware. *Archives of Computational Methods in Engineering*, 30(2):951–983, March 2023.
- [18] Han Bao, ZhuGuan Chen, JianMing Cai, Quan Xu, and BoCheng Bao. Memristive cyclic three-neuron-based neural network with chaos and global coexisting attractors. *Science China Technological Sciences*, 65(11):2582–2592, November 2022.
- [19] Bocheng Bao, Hui Qian, Quan Xu, Mo Chen, Jiang Wang, and Yajuan Yu. Coexisting Behaviors of Asymmetric Attractors in Hyperbolic-Type Memristor based Hopfield Neural Network. *Frontiers in Computational Neuroscience*, 11:81, August 2017.

- [20] Z. Tabekoueng Njitacke, Sami Doubla Isaac, J. Kengne, A. Nguomkam Negou, and Gervais Dolvis Leutcho. Extremely rich dynamics from hyperchaotic Hopfield neural network: Hysteretic dynamics, parallel bifurcation branches, coexistence of multiple stable states and its analog circuit implementation. *The European Physical Journal Special Topics*, 229(6):1133–1154, March 2020.
- [21] Chengjie Chen, Fuhong Min, Fei Hu, Jianming Cai, and Yunzhen Zhang. Analog/digital circuit simplification for Hopfield neural network. *Chaos, Solitons & Fractals*, 173:113727, August 2023.
- [22] Z.T. Njitacke and J. Kengne. Complex dynamics of a 4D Hopfield neural networks (HNNs) with a nonlinear synaptic weight: Coexistence of multiple attractors and remerging Feigenbaum trees. *AEU - International Journal of Electronics and Communications*, 93:242–252, September 2018.
- [23] Z. Tabekoueng Njitacke, J. Kengne, and H. B. Fotsin. Coexistence of Multiple Stable States and Bursting Oscillations in a 4D Hopfield Neural Network. *Circuits, Systems, and Signal Processing*, 39(7):3424–3444, July 2020.
- [24] Samuel Shapero, Christopher Rozell, and Paul Hasler. Configurable hardware integrate and fire neurons for sparse approximation. *Neural Networks*, 45:134–143, September 2013.
- [25] Pranav O. Mathews and Jennifer O. Hasler. Physical Computing for Hopfield Networks on a Reconfigurable Analog IC. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, Monterey, CA, USA, May 2023. IEEE.
- [26] Derong Liu, editor. *Advances in neural networks–ISNN 2007: 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007: proceedings*. Number 4491-4493 in Lecture notes in computer science. Springer, Berlin ; New York, 2007.
- [27] Jay Kant Pratap Singh Yadav, Laxman Singh, and Zainul Abidin Jaffery. Optimization of Hopfield Neural Network (HNN) using multiconnection and Lyapunov Energy Function (LEF) for storage and recall of handwritten images. *Sādhanā*, 48(1):26, February 2023.
- [28] Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. *Computational intelligence: a methodological introduction*. Texts in computer science. Springer, Cham, third edition edition, 2022.
- [29] Hairong Lin, Chunhua Wang, Fei Yu, Jingru Sun, Sichun Du, Zekun Deng, and Quanli Deng. A Review of Chaotic Systems Based on Memristive Hopfield Neural Networks. *Mathematics*, 11(6):1369, March 2023.

- [30] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*, 3:4, February 2020.
- [31] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: an introduction to dynamical systems*. Textbooks in mathematical sciences. Springer, New York, NY Heidelberg, repr. edition, 2010.
- [32] Morris W. Hirsch, Stephen Smale, and Robert L. Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic Press, Waltham, MA, 3rd ed edition, 2013.
- [33] Steven Strogatz and Mitchal Dichter. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, Boca Raton, FL, second edition edition, 2018.
- [34] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, March 1963.
- [35] Robert L. Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. CRC Press, 1 edition, May 2018.
- [36] Hernando Fernandez-Canque. *Analog electronics applications: fundamentals of design and analysis*. CRC Press, Boca Raton London New York, 2017.
- [37] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Numerical analysis*. Cengage Learning, Boston, MA, tenth edition edition, 2016.
- [38] Huaguang Zhang, Zhanshan Wang, and Derong Liu. A Comprehensive Review of Stability Analysis of Continuous-Time Recurrent Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1229–1262, July 2014.
- [39] Chengjie Chen and Fuhong Min. ReLU-type memristor-based Hopfield neural network. *The European Physical Journal Special Topics*, 231(16-17):2979–2992, November 2022.
- [40] Sajad Jafari, Julien C. Sprott, and Soroush Dehghan. Categories of Conservative Flows. *International Journal of Bifurcation and Chaos*, 29(02):1950021, February 2019.
- [41] JACK HEIDEL and FU ZHANG. Nonchaotic and chaotic behavior in three-dimensional quadratic systems: five-one conservative cases. *International Journal of Bifurcation and Chaos*, 17(06):2049–2072, 2007.

- [42] Sajad Jafari, Julien C. Sprott, and Soroush Dehghan. Categories of conservative flows. *International Journal of Bifurcation and Chaos*, 29(02):1950021, 2019.
- [43] J. C. Sprott. Some simple chaotic flows. *Phys. Rev. E*, 50:R647–R650, Aug 1994.
- [44] Hassan K. Khalil. *Nonlinear systems. Hauptbd.* Prentice Hall, Upper Saddle River, NJ, 3. ed edition, 2002.
- [45] J. C. Sprott. Strange attractors with various equilibrium types. *The European Physical Journal Special Topics*, 224(8):1409–1419, July 2015.
- [46] Mohammad Saleh Tavazoei and Mohammad Haeri. Chaotic attractors in incommensurate fractional order systems. *Physica D: Nonlinear Phenomena*, 237(20):2628–2637, October 2008.
- [47] George Datseris and Ulrich Parlitz. *Nonlinear dynamics: a concise introduction interlaced with code.* Undergraduate lecture notes in physics. Springer, Cham, 2022.
- [48] Sitabhra Sinha. Chaotic Dynamics in Iterated Map Neural Networks with Piecewise Linear Activation Function. *Fundamenta Informaticae*, 37(1,2):31–50, 1999.
- [49] Anadigm. Using the anadigm transferfunction cam, 2009. https://www.anadigm.com/_doc/DB0223009-U210.pdf [Accessed: 28-05-2024].
- [50] J. Lu, G. Chen, X. Yu, and H. Leung. Design and Analysis of Multiscroll Chaotic Attractors From Saturated Function Series. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(12):2476–2490, December 2004.

Conclusiones generales

En este trabajo de tesis se abordó el modelo de red neuronal Hopfield de 3 y 4 neuronas, desarrollando la metodología para su implementación en circuitos analógicos programables de prototipado rápido.

Se realizó el desarrollo teórico, numérico y experimental de redes neuronales Hopfield (HNN) en modo caótico, analizando el impacto de distintas funciones de activación en su dinámica y aprendizaje.

En primer lugar, se estableció la estructura básica de HNN de 3 y 4 neuronas, definiendo sus modelos matemáticos y realizando su integración numérica mediante métodos de Runge-Kutta (RK2 y RK4). Esto permitió simular y caracterizar la dinámica no lineal de los sistemas, identificando comportamientos caóticos e hipercaóticos mediante técnicas analíticas como el análisis de estabilidad de Lyapunov, puntos de equilibrio, diagramas de bifurcación y exponentes de Lyapunov. Se demostró que los modelos son disipativos y acotados, garantizando la presencia de atractores, incluso en sistemas con funciones de activación no convencionales como la función signo, para la cual se reportó el primer modelo HNN con caos.

Además, se evaluó el efecto de cinco funciones de activación (ReLU, PWL, tanh, sigmoide y signo) en el proceso de aprendizaje y dinámica caótica. Se propusieron nuevos modelos para las funciones sigmoide y signo, este último destacando por su discontinuidad abrupta, y se validó su comportamiento caótico mediante mapas de calor de exponentes de Lyapunov y simulaciones numéricas. Se confirmó que la elección de la función de activación influye directamente en la complejidad dinámica, evidenciando multiestabilidad en un modelo diseñado con n cantidad de enrollamientos.

Para la implementación práctica, se desarrollaron ecuaciones de diseño basadas en filtros sumadores analógicos, adaptando los modelos a la tarjeta FPAA mediante transformaciones de variables y frecuencia. Se implementaron con éxito tanto redes de 3 neuronas (incluyendo señales inhibitorias) como un modelo hipercaótico de 4 neuronas, validando su dinámica experimentalmente y comparándola con simulaciones. La creación de una interfaz en Python para prototipado rápido en AnadigmDesigner2 facilitó la configuración y replicabilidad de los circuitos, demostrando estabilidad y bajo ruido en el hardware. Posteriormente, se realizó una aplicación en C++ y MFC que permitió la comunicación directa con la tarjeta analógica embebida sin la necesidad de la programación en bloques y el diseño de la red en AnadigmDesigner2.

Los resultados obtenidos permitirán la implementación electrónica de una HNN con cualquier tipo de función de activación o parámetro requerido. Aún más, el análisis dinámico orientado al estudio de sistemas caóticos elaborado en este trabajo da lugar a técnicas numéricas más robustas, evitando diversas contradicciones halladas en la literatura. A su vez, se desarrolló una nueva técnica numérica para la identificación de dinámicas caóticas en sistemas dinámicos con

divergencia no uniformemente conservativa o no uniformemente disipativa, lo que contrubuye al estado del arte y ofrece una nueva herramienta de identificación de caos en sistemas dinámicos. Finalmente, se propuso una nueva función de activación que ofrece a la HNN dinámicas muy interesantes, entre las cuales se encuentran la multiestabilidad y la capacidad de contener una n cantidad de enrollamientos. A su vez, este modelo simplifica la cantidad de elementos necesarios para la implementación analógica de una HNN en modo caótico.

En conclusión, los resultados obtenidos confirman que las funciones de activación no lineales modifican significativamente la dinámica caótica y el aprendizaje de las HNN, con implicaciones en aplicaciones como el reconocimiento de patrones y sistemas multiestables. La implementación exitosa en dispositivos analógicos programables valida la metodología propuesta, combinando flexibilidad, precisión y eficiencia, y abre nuevas posibilidades para el diseño de sistemas neuromórficos basados en dinámicas complejas.

APPENDICE A

Participación en eventos académicos

Se realizó una ponencia en el congreso EDIESCA 2024, el cual se celebró los días 28, 29 y 30 de octubre del año 2024 en la universidad panamericana de Aguascalientes. La ponencia fue con respecto al análisis dinámico orientado a caos y la implementación de redes neuronales Hopfield en una tarjeta analógica embebida. El nombre del trabajo presentado fue "Rich dynamics of a Hopfield neural network and its analog circuit realization". Se anexa la constancia de participación en el evento.



THE FACULTY OF ENGINEERING AT UNIVERSIDAD PANAMERICANA CERTIFY THAT

Diego Ruiz Sánchez de la Vega

Has participated as speaker in a talk entitled

"Rich dynamics of a Hopfield neural network and its analog circuit realization"

During the 4th Meeting for the Dissemination and Research in the Study of Complex Systems and their Applications (EDIESCA 2024), Awarded in Aguascalientes México, October 28-30, 2024.

Mtra. Mercedes Urzúa González
Dean Faculty of Engineering
Universidad Panamericana
Campus Aguascalientes

Dr. Guillermo Huerta Cuellar
Head of Asociación Mexicana de
Sistemas Dinámicos y Complejidad

Dr. Héctor Eduardo Gilardi Velázquez
Head Chair EDIESCA2024
Faculty of Engineering
Universidad Panamericana
Campus Aguascalientes

Figura A.1. Constancia de participación en el congreso EDIESCA 2024.

Como resultado del trabajo de investigación se realizaron dos artículos de investigación indizados:

1. D. S. de la Vega, J. M. Munoz-Pacheco, O. G. Félix-Beltrán, and C. Volos, "Rich dynamics and analog implementation of a Hopfield neural network in integer and fractional order domains," *Integration*, vol. 103, p. 102389, 2025, doi: <https://doi.org/10.1016/j.vlsi.2025.102389>.
2. D. S. de la Vega, O. G. Félix-Beltrán, and J. M. Munoz-Pacheco, "Identifying chaotic dynamics in nonuniformly dissipative and conservative dynamical systems from the varying divergence," *Results Phys.*, p. 108259, Apr. 2025, doi: [10.1016/j.rinp.2025.108259](https://doi.org/10.1016/j.rinp.2025.108259).

El primero fue publicado en "Integration, the VLSI Journal", la cual es una revista científica internacional revisada por pares, publicada por Elsevier desde 1983, que se especializa en el campo de la Integración a Muy Gran Escala (VLSI). Su enfoque abarca el diseño, verificación, prueba y aplicación de circuitos y sistemas integrados, así como temas relacionados en tecnologías de procesos y dispositivos.

En cuanto a su impacto académico, según el Journal Citation Reports (JCR) de 2024, la revista tiene un **factor de impacto de 2.2** y su **CiteScore es de 3.8**. En términos de clasificación, se encuentra en el **cuartil Q2** en las categorías de Ingeniería Eléctrica y Electrónica y **cuartil Q3** en las categorías de Hardware y Arquitectura, y Software .

La revista está indexada en bases de datos reconocidas como Science Citation Index Expanded (SCIE) y Scopus, y publica artículos trimestralmente.

El segundo artículo fue publicado en "Results in Physics" la cual es una revista científica internacional revisada por pares, publicada por Elsevier desde 2011, que se enfoca en la publicación rápida de investigaciones originales en todas las áreas de la física y campos interdisciplinarios relacionados.

En cuanto a su impacto académico, según el Journal Citation Reports (JCR) de 2024, la revista tiene un **factor de impacto de 4.4** y su **CiteScore es de 8.7**. En términos de clasificación, se encuentra en el cuartil Q1 en las categorías de Física Aplicada y Física General.

La revista está indexada en bases de datos reconocidas como Science Citation Index Expanded (SCIE) y Scopus, y publica artículos de manera continua. Su audiencia principal incluye investigadores académicos, profesionales de la física aplicada y teórica, y estudiantes avanzados de posgrado en física y disciplinas afines.

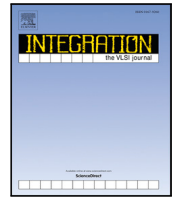
A continuación se anexan los dos artículos de investigación.



Contents lists available at ScienceDirect

Integration, the VLSI Journal

journal homepage: www.elsevier.com/locate/vlsi



Rich dynamics and analog implementation of a Hopfield neural network in integer and fractional order domains

Diego S. de la Vega^a, Jesus M. Munoz-Pacheco^a,^{*} Olga G. Félix-Beltrán^a, Christos Volos^b

^a Faculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla, Puebla, Mexico

^b Laboratory of Nonlinear Circuits, Systems & Complexity (LaNSCom), Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece

ARTICLE INFO

Keywords:

Hopfield neural network
Chaotic systems
Fractional calculus
Analog circuit
FPAA

ABSTRACT

Several synaptic weight matrices have been proposed for Hopfield neural network (HNN) models, where chaotic dynamics may arise. Contrary to those works, this manuscript aims to present a synaptic weight matrix where every entry can be set as an integer, harvesting an elegant chaotic HNN from a chaos theory point of view. Analytical and numerical analyses such as equilibrium points, bifurcation diagrams, Lyapunov exponents, and basins of attraction demonstrate that the proposed HNN exhibits complex behaviors across a wide range of parameter values. Also, we extend the study of the HNN into the fractional order domain. Moreover, the design and implementation details of the proposed neural network using field programmable analog arrays (FPAAs) are thoroughly discussed. This includes the various components and their configurations, highlighting how they contribute to the overall functionality of the neural network. As a result, we found a strong correlation between numerical simulations and SPICE circuit simulations.

1. Introduction

Mathematical models of nonlinear dynamical systems are present in almost every aspect of science since they can model most time-evolving phenomena, such as chaos. Chaotic dynamics manifest in many natural systems, including classical mechanical systems [1], electronic circuits [2], biological systems [3], among others. Indeed, mathematical models for artificial neural networks generating chaos have also been proposed in recent works [4]. Artificial Neural Networks (ANNs) are inspired by biological neural networks defined as a system of elements called neurons that can send, receive, and compute signals and exhibit rich and complicated dynamics like the human brain [5]. In literature, one can find different models for ANNs, such as convolutional, recurrent, feed-forward, de-convolutional, and so forth. Among them, Hopfield Neural Networks (HNNs) are a key focus of extensive research in the field of ANNs. First proposed by J. Hopfield [6], HNNs are applied for pattern recognition but with a simple architecture. In addition, the chaotic phenomena may also arise in HNNs. Recent investigations have discovered chaotic [3], hyperchaotic [2] behaviors with single-scroll attractors and double-scroll attractors [7]. For instance, Bao et al. [3] established a tri-neuron network (3D-HNN) with chaotic behavior, where the results were demonstrated theoretically and experimentally by dynamical tools and electronic devices, respectively. Apart from the network topology, the entries of the synaptic weight matrix are fundamental for chaos generation. Therefore, chaotic

behavior with different values for the weight matrix has been proposed and experimentally validated using analog or digital circuits.

From a chaos point of view, Sprott [8] proposed that new chaotic systems must have at least one of the following criteria:

- The system should model some unsolved problem in nature.
- The system should exhibit some unseen behavior.
- The system should be simpler than all other known examples exhibiting the observed behavior.

Simplicity in a system refers not only to the number of parameters and variables but also to the digits in the parameters of a chaotic system. In this framework, we have found an integer-valued synaptic weight matrix generating chaotic patterns in the HNN, yielding an elegant neural network satisfying the third Sprott criteria.

On the other hand, fractional-order calculus is a mathematical branch considered a generalization of integer-order calculus [9]. In recent years, fractional calculus has become a relevant tool for modeling and offering new dynamics to new and previously proposed systems. For instance, the study of the fractional version of Chua's [10], Lorenz [11], and Rossler [12] systems. There is a tendency in recent years to explore the fractional order version of a chaotic system to investigate its implications in chaos generation [13–15] since fractional calculus provides an extra degree of freedom in chaotic systems,

* Corresponding author.

E-mail address: jesusm.pacheco@correo.buap.mx (J.M. Munoz-Pacheco).

<https://doi.org/10.1016/j.vlsi.2025.102389>

Received 16 November 2024; Received in revised form 10 February 2025; Accepted 22 February 2025

Available online 5 March 2025

0167-9260/© 2025 Elsevier B.V. All rights reserved, including those for text and data mining, AI training, and similar technologies.

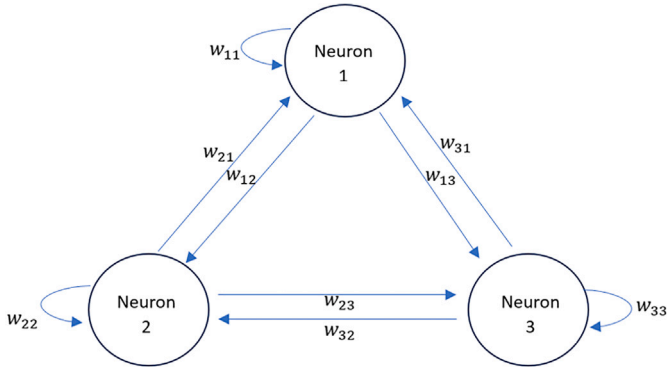


Fig. 1. A tri-neuron Hopfield neural network.

resulting in undiscovered dynamics. In this manner, fractional-order features allow us to describe natural phenomena and provide novel applications in engineering, as in fractional control [16], medicine [17], cryptography [18], and so forth.

A well-accepted approach for validating the chaotic behavior of new systems is through circuit implementation [7,19,20]. Thus, efficient circuit realizations are mandatory, e.g., fractional-order systems require the implementation of the operator s^{-q} . Unfortunately, a circuit with this behavior is not commercially available yet, and an approximation should be used instead [18]. In this work, the fractional-order integrator is performed using first-order low-pass filters implemented with field programmable analog arrays (FPAAs). Moreover, the circuit implementation of the proposed HNN model is also carried on with FPAAs.

Based on this framework, this paper aims to present a new Hopfield neural network with three neurons. The main contribution is proposing an elegant set of synaptic weight matrices (SWM) in the Sprott sense for which the HNN presents chaotic dynamics. This means that the weights can take values with minimum digits. For the proposed HNN, the parameters can be defined only as integer numbers. Moreover, we discover large chaotic regions as a function of the parameters of the Hopfield neural network, delivering a robust chaotic system that can be useful for encryption applications. Finally, we detailed the circuit implementation of the integer-order and fractional-order versions of the proposed HNN.

The manuscript is organized as follows: Section 2 presents the mathematical model of the proposed Hopfield neural network. Section 3 discusses several points about the integer-order HNN, such as equilibrium points, stability, Lyapunov exponents, bifurcation diagrams, phase portraits, and basin of attraction. It also discusses the electronic implementation in FPAAs and the design of a pseudo-random number generator. Section 4 introduces the fractional-order HNN together with the numerical and FPAA implementation. Finally, Section 5 illustrates the conclusions.

2. The proposed Hopfield neural network

The model for the artificial neural networks emulates the behavior of a biological neural network [21]. In this sense, the neural network should be able to receive, generate, and compute signals. Hopfield neural networks are fully connected networks. A diagram for a Hopfield neural network with three neurons is presented in Fig. 1. The mathematical model for an n th dimensional HNN is described by

$$\dot{\mathbf{x}} = -C\mathbf{x} + W\mathbf{g}(\mathbf{x}) + \mathbf{b}, \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is the space state variable representing the behavior of each neuron in the system, $C \in \mathbb{R}^{n \times n}$ is a diagonal matrix and represents the membrane resistance, $\mathbf{g}(\cdot) \in \mathbb{R}^n$ the activation

function, for which in this contribution is the hyperbolic tangent, $\mathbf{b} \in \mathbb{R}^n$ the input bias current, and $W = (w_{ij})_{n \times n}$ is the SWM. In this contribution for a three-dimensional HNN, we consider $n = 3$, $C = I \in \mathbb{R}^{3 \times 3}$ the identity and $\mathbf{b} = \mathbf{0}$. The SWM is proposed as follows

$$W = \begin{bmatrix} 0 & w_{12} & w_{13} \\ 1 & 3 & 3 \\ -1 & -3 & 0 \end{bmatrix}. \quad (2)$$

Note that every weight, besides w_{12} and w_{13} , of the synaptic matrix is an integer. We will see a huge parameter zone for w_{12} and w_{13} for which the HNN model can present chaos, including a diverse set of integers. Rewritten the Eq. (1) in a dimensionless form, yields

$$\begin{aligned} \dot{x}_1 = f_1(\mathbf{x}) &= -x_1 + w_{12} \tanh x_2 + w_{13} \tanh x_3, \\ \dot{x}_2 = f_2(\mathbf{x}) &= -x_2 + \tanh x_1 + 3 \tanh x_2 + 3 \tanh x_3, \\ \dot{x}_3 = f_3(\mathbf{x}) &= -x_3 - \tanh x_1 - 3 \tanh x_2, \end{aligned} \quad (3)$$

with $\mathbf{x} = (x_1, x_2, x_3)^T$ being the vector containing the space state variables. It is also worth mentioning that $\mathbf{f}(-\mathbf{x}) = -\mathbf{f}(\mathbf{x})$, which means that the model is symmetric with respect to the origin.

3. Integer order system analysis

3.1. Dynamical analysis

This section provides a general analysis of the considered dynamical system. We conclude the existence of attractors for some given parameters, the boundedness of the system, the equilibria, and the stability of the proposed system.

3.1.1. Dissipation

The volume contraction rate is used to evaluate the dissipation of a dynamical system. Let $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^T$ be a vector in $\mathbb{R}^3$ described by Eq. (3), then its divergence $\text{div}(\mathbf{f}(\mathbf{x})) = \nabla \cdot \mathbf{f}(\mathbf{x})$ usually denoted as Λ . The criterion to evaluate if a system is dissipative consists of checking the sign of Λ . In case $\Lambda < 0$ the system is dissipative, for $\Lambda = 0$ then the system is conservative, and if $\Lambda > 0$ the system is unbounded. We recall that the dissipation property of a system guarantees the presence of a bounded global attractor and an analytical indication of the global attractor in the phase space [22]. For the considered system, the divergence is obtained by

$$\Lambda = \frac{\partial f_1(\mathbf{x})}{\partial x_1} + \frac{\partial f_2(\mathbf{x})}{\partial x_2} + \frac{\partial f_3(\mathbf{x})}{\partial x_3}. \quad (4)$$

It can be easily verified that the volume contraction rate is given by

$$\Lambda = -3 \tanh^2 x_2. \quad (5)$$

Note the dependence on the state of Λ ; as such, it cannot be concluded the type of system with just a simple criterion. We recall that the average rate of the fractional volume expansion along a trajectory is equivalent to the divergence of the flow [23]. For a volume expansion $V(t)$ we have

$$\frac{1}{V(t)} \frac{dV(t)}{dt} = \Lambda \quad (6)$$

with trivial solution

$$V(t) = V(0)e^{\Lambda t}, \quad (7)$$

$$= V(0)e^{-3t \tanh^2 x_2}, \quad (8)$$

where $0 \leq \tanh^2 x_2 < 1$, then the volume reduces to zero as t tends to infinity in a rate of $-3 \tanh^2 x_2$. Let us explore the volume expanding rate further. The average of the divergence is used to verify whether the considered is a dissipative dynamical system. Consider a variable $S(t)$, then $\langle S(t) \rangle$ denotes the average of the variable and is given by

$$\langle S(t) \rangle = \frac{\int_{t_0}^{t_f} S(t) dt}{t_f - t_0}, \quad (9)$$

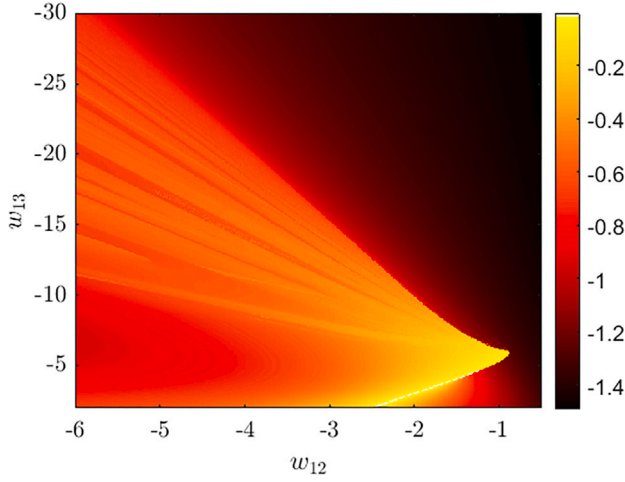


Fig. 2. Average divergence for a simulation time of 450 s of the considered model varying values of w_{12} from -6 to -0.5 and w_{13} from -30 to -2 with initial conditions $(0.5, -0.2, 0)^T$.

in a period t_0 to t_f . Then for $t_0 = 0$ the average is given by

$$\langle A \rangle = \frac{-3}{t_f} \int_0^{t_f} \tanh^2(x_2(\tau)) d\tau. \quad (10)$$

By finding the solution of x_2 and numerically solving Eq. (10) using initial conditions $\mathbf{x}(0) = (-0.5, 0.2, 0)^T$, for the parameter zone $-6 \leq w_{12} \leq -0.5$ and $-30 \leq w_{13} \leq -2$ we found the color map in Fig. 2. As a result, we notice that values of the average of the divergence are always less than zero, and as such, the proposed system is dissipative and can contain attractors. To solve the differential Eq. (3), we used the standard numerical integrator Runge–Kutta of variable order 4–5 in a period of $t_i = 0$ and $t_f = 500$ s. The result found by Fig. 2 also resembles the color map found in the next sections by finding the Lyapunov spectrum.

3.1.2. Model boundless proof

The Lyapunov method for the confinement of the 3D-HNN model can be proven as follows.

Theorem 1. *The trajectory described by the considered system (3) is confined in a bounded region $\Psi(\mathbf{x}) \in \mathbb{R}$.*

Proof. Let a candidate Lyapunov function $V(\mathbf{x})$ be defined as

$$\Psi(\mathbf{x}) = \frac{1}{2} \mathbf{x} \cdot \mathbf{x} = \frac{1}{2} \|\mathbf{x}\|^2, \quad (11)$$

then, the temporal derivative of $V(\mathbf{x})$ is given by

$$\begin{aligned} \dot{\Psi}(\mathbf{x}) &= \mathbf{x} \cdot \dot{\mathbf{x}} = \mathbf{x} \cdot (-\mathbf{x} + W\mathbf{g}(\mathbf{x})) \\ &= -\|\mathbf{x}\|^2 + \mathbf{x} \cdot (W\mathbf{g}(\mathbf{x})). \end{aligned} \quad (12)$$

Now, it taken into account the activation function $\mathbf{g}(\mathbf{x}) = (\tanh x_1, \tanh x_2, \tanh x_3)^T$ and the proposed SWM, yields

$$\begin{aligned} \dot{\Psi}(\mathbf{x}) &= -x_1^2 - x_2^2 - x_3^2 + \\ &\quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} w_{12} \tanh x_2 + w_{13} \tanh x_3 \\ \tanh x_1 + 3 \tanh x_2 + 3 \tanh x_3 \\ -\tanh x_1 - 3 \tanh x_2 \end{bmatrix}. \end{aligned} \quad (13)$$

Let a new variable $\phi(\mathbf{x})$ be

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} w_{12} \tanh x_2 + w_{13} \tanh x_3 \\ \tanh x_1 + 3 \tanh x_2 + 3 \tanh x_3 \\ -\tanh x_1 - 3 \tanh x_2 \end{bmatrix}. \quad (14)$$

Table 1

Nonzero equilibrium points and its eigenvalues for a set of weights w_{12} and w_{13} . When not specified w_{12} and w_{13} are fixed as -1 and -6 respectively.

	Value	Nonzero EP	Eigenvalue for nonzero EP
w_{12}	-1.75	$(1.093, -0.231, -0.116)$	$-0.71, 0.28 + 2.12i$
	-1.5	$(0.972, -0.217, -0.109)$	$-0.67, 0.26 + 2.02i$
	-1.25	$(0.838, -0.198, -0.010)$	$-0.63, 2.55 \pm 1.88i$
	-1	$(0.681, -0.171, -0.086)$	$-0.57, 0.24 \pm 1.68i$
	-0.75	$(0.480, -0.128, -0.064)$	$-0.47, 0.21 \pm 1.37i$
w_{13}	-9	$(1.421, -0.259, -0.131)$	$-0.91, 0.36 \pm 2.13i$
	-8	$(1.213, -0.243, -0.122)$	$-0.86, 0.34 \pm 2.02i$
	-7	$(0.977, -0.217, -0.109)$	$-0.77, 0.31 \pm 1.88i$
	-6	$(0.681, -0.171, -0.086)$	$-0.57, 0.24 \pm 1.68i$
	-5.5	$(0.478, -0.128, -0.064)$	$-0.37, 0.16 \pm 1.55i$

Recall that $|\tanh(x)| < 1$ and $|\tanh(x)| < |x|$, then $\phi(\mathbf{x})$ can be reduced as

$$\begin{aligned} |\phi(\mathbf{x})| &\leq |(x_2 - x_3) \tanh(x_1)| + |(w_{12}x_1 + 3x_2 - 3x_3) \tanh x_2| \\ &\quad + |(w_{13}x_1 + 3x_2) \tanh x_3| \\ &< |x_2 - x_3| + |(w_{12}x_1 + 3x_2 - 3x_3)| + |(w_{13}x_1 + 3x_2)| \\ &< (w_{12} + w_{13})|x_1| + 7|x_2| + 4|x_3|. \end{aligned} \quad (15)$$

Let $D = \Psi(\mathbf{x})$, then for all $\mathbf{x} = (x_1, x_2, x_3)^T$ exists a large enough region $D_0 \geq 0$ and $D > D_0$, where the following inequality holds:

$$\begin{aligned} \phi(\mathbf{x}) &< (w_{12} + w_{13})|x_1| + 7|x_2| + 4|x_3| \\ &< x_1^2 + x_2^2 + x_3^2 = 2\Psi(\mathbf{x}). \end{aligned} \quad (16)$$

Then on the surface $\{(x_1, x_2, x_3) \mid V(x_1, x_2, x_3) = D\}$ with $D > D_0$ there yields

$$\dot{\Psi}(\mathbf{x}) = -2\Psi(\mathbf{x}) + \phi(\mathbf{x}) < 0. \quad (17)$$

Consequently, the set $\{(x_1, x_2, x_3) \mid \Psi(x_1, x_2, x_3) = D_0 \leq D\}$ is a bounded region of all solutions for the HNN model with appropriate values for w_{12} and w_{13} .

3.1.3. Equilibria

The analysis of non-linear systems usually involves a linearization around possible equilibrium points (EP). This analysis is generally called equilibria and allows us to identify the behavior of a non-linear system for close trajectories of the EP. Let $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3)^T$ be an EP in the proposed model, meaning it satisfies the following equation:

$$\begin{aligned} 0 &= -\bar{x}_1 + w_{12} \tanh \bar{x}_2 + w_{13} \tanh \bar{x}_3, \\ 0 &= -\bar{x}_2 + \tanh \bar{x}_1 + 3 \tanh \bar{x}_2 + 3 \tanh \bar{x}_3, \\ 0 &= -\bar{x}_3 - \tanh \bar{x}_1 - 3 \tanh \bar{x}_2. \end{aligned} \quad (18)$$

Apart from the trivial equilibrium point EP at $(0, 0, 0)$, it is clear that the system of equations in (18) is not analytically solvable, and a numerical method is necessary. Thus, the equilibrium points are computed and recorded using Matlab software in Table 1. This summarizes the EP and eigenvalues varying parameters w_{12} and w_{13} from -1.75 up to -0.75 and -9 up to -5.5 , respectively. Given that the proposed model is symmetric only one nonzero EP is shown, the remaining equilibrium points are in the form $(\pm \bar{x}_1, \pm \bar{x}_2, \pm \bar{x}_3)$.

Also, the Jacobian matrix J of (3) around the equilibrium point $\bar{\mathbf{x}}$ can be proven to be

$$J = \begin{bmatrix} -1 & w_{12}(1 - \tanh^2 \bar{x}_2) & w_{13}(1 - \tanh^2 \bar{x}_3) \\ 1 - \tanh^2 \bar{x}_1 & -1 + 3(1 - \tanh^2 \bar{x}_2) & 3(1 - \tanh^2 \bar{x}_3) \\ -(1 - \tanh^2 \bar{x}_1) & -3(1 - \tanh^2 \bar{x}_2) & -1 \end{bmatrix}. \quad (19)$$

Then, the characteristic equation for the considered system is given by

$$|\lambda I - J| = \lambda^3 + a_1 \lambda^2 + a_2 \lambda + a_3 = 0, \quad (20)$$

Table 2

Eigenvalues for trivial equilibrium points for a set of weights w_{12} and w_{13} . When not specified w_{12} and w_{13} are fixed as -1 and -6 respectively.

	Value	Eigenvalue for trivial EP
w_{12}	-1.75	$0.94, -0.47 \pm 1.55i$
	-1.5	$0.87, -0.43 \pm 1.44i$
	-1.25	$0.79, -0.39 \pm 1.31i$
	-1	$0.68, -0.34 \pm 1.16i$
	-0.75	$0.5, -0.25 \pm 0.96i$
w_{13}	-9	$2.00, -1.00 \pm 1.00i$
	-8	$1.67, -0.83 \pm 1.05i$
	-7	$1.26, -0.63 \pm 1.09i$
	-6	$0.68, -0.34 \pm 1.16i$
	-5.5	$0.31, -0.16 \pm 1.25i$

with

$$\begin{aligned} a_1 &= 3 - 9c^2, \\ a_2 &= -6c^2 + 9c^2d^2 - w_{12}b^2c^2 + w_{13}b^2d^2 + 3, \\ a_3 &= 3c^2(w_{12}b^2d^2 + 3d^2 - 1) + w_{13}b^2(d^2 - c^2) + 1, \end{aligned} \quad (21)$$

where $b = 1 - \tanh^2 \bar{x}_1$, $c = 1 - \tanh^2 \bar{x}_2$ and $d = 1 - \tanh^2 \bar{x}_3$. Notice that the coefficients of (21) are all nonzero. Now, we recall the Routh–Hurwitz criterion that implies the real parts of the roots of (20) are positive if and only if the next set of inequalities are met

$$\begin{aligned} a_1 &> 0, \\ a_1a_2 - a_3 &> 0, \\ a_3 &> 0. \end{aligned} \quad (22)$$

The type of equilibria is obtained by solving (20) for λ , and they can be split into eight different equilibrium scenarios [24]. Table 1 displays the eigenvalues found, and from there, the specific type of equilibrium point can be described. The remaining λ conserves the corresponding value of the symmetric equilibrium points, and the attractor behaves the same way. From the results of Table 1, it is evident that all zero equilibrium points are spiral saddle points of index I and all nonzero equilibrium points are spiral saddle points of index II.

As is known for chaotic systems, scrolls are generated around saddle points of index II, while saddle points of index I are responsible for connecting the scrolls [25] as given in Table 2. It is also noticeable that for larger values of w_{13} , all real values of the eigenvalues start getting closer to zero.

3.2. Complex dynamics in the proposed system

In this section, we present the dynamical characterization of the proposed HNN (3) through standard quantitative and qualitative tools for dynamical analysis, such as phase space portrait, bifurcation diagrams, and Lyapunov exponents. In particular, we have computed a numerical solution using the variable fourth- or fifth-order Runge–Kutta algorithm of Matlab to obtain the phase portrait and bifurcation diagrams of local maxima. Besides, Lyapunov exponents are computed using the algorithm provided by V. Govorukhin. The basin of attraction is also found in this analysis. Through all this analysis, different dynamics like periodic, quasiperiodic, and chaotic zones are found.

3.2.1. Dynamics with parameters $w_{12} = -1$ and $-6.3 \leq w_{13} \leq -5.8$

System in Eq. (3) can present various dynamics by fixing $w_{12} = -1$ and varying w_{13} , where the chosen value for w_{12} is the smallest possible for which all elements in the SWM are integers, and the considered can present chaos. For starters, the system has two self-excited attractors with positions set by the EP of index II (see Table 1); bi-stability was found by initiating the system with different initial conditions. In Fig. 3, two different initial conditions $(0.5, -0.2, 0)$ (in red) and $(-0.5, 0.2, 0)$ (in blue) were used with $w_{13} = -5.8$, $w_{13} = -5.9$ and $w_{13} = -5.91$ respectively, simulating the system 1000 s with a transient of 800 s.

Table 3

Type attractor based on the sign of LEs.

LEs signs	Type of attractor
$(-, -, -)$	Stable fixed point
$(0, -, -)$	Limit cycle
$(0, 0, -)$	Quasiperiodic motion
$(+, 0, -)$	Chaos

Here, we can note three different periods in the proposed system for each given parameter as it is shown in Figs. 3(a)–3(c), it is known period 3 is an indication of chaos and as such chaotic dynamics appear for $w_{13} < 5.98$. In Fig. 3(d), we provide the phase portrait of the system for which the system is chaotic with a $w_{13} = -6$ and a single scroll attractor. Next, in Fig. 3(e), a double scroll attractor is shown; while the initial state and parameters stay the same, the time simulation was set as 5000 s, meaning the model takes long periods of time before jumping to the next scroll.

The described dynamics are summarized by the bifurcation diagram of local maxima of x_1 in Fig. 3(f), where the bifurcation diagram in blue uses the initial condition $(0.01, 0, 0)^T$ and in red $(-0.01, 0, 0)^T$. Here, we note diverse discontinuities going from period 2 to period 3 and a dense parameter for $w_{13} \in [-6.18, -5.98]$, indicating possible chaos.

3.2.2. Bifurcation diagrams in a 3D plane

To further explore the dynamics of the proposed system in the parameter space defined by w_{12} and w_{13} , we develop some three-dimensional plane bifurcation diagrams shown in Fig. 4 with initial condition $\mathbf{x}_0 = (-0.5, 0.2, 0)^T$. In Fig. 4 we plotted 4 different bifurcation diagrams with $w_{13} = -6$ (green), -9 (red), -12 (blue), -15 (black) respectively. This graph denotes that for smaller values of w_{13} , the dense regions for the parametric zone of w_{12} grow while the dynamics of the system stay relatively the same. It is evident that the dense parametric zone for w_{12} not only is expanded but also shifts. These results motivate us to explore the parametric zone for which w_{12} and w_{13} are negative. To assert this dense region as chaotic, we will use other metrics to identify chaos, such as the Lyapunov spectrum.

3.2.3. The Lyapunov spectrum

The Lyapunov spectrum (LS) measures exponential divergence between two infinitely close orbits in their initial state. It is composed of a set of n values for a system of n dimensions called the Lyapunov exponents (LEs), and are usually order from the largest to the smallest. The first method to compute the LS was published by G. Benettin et al. [26]. The LS allows us to characterize quantitatively the stochastic properties of the attractor in a system, one being chaos. For a three-dimensional system, Table 3 summarizes the type of dynamics of the attractor in a system based on the sign of the LEs.

To find the LEs first consider a dynamical system described by the set of differential equations

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \quad (23)$$

where $\dot{\mathbf{y}}$ denotes the temporal derivative of the vector field $\mathbf{y}$, $\mathbf{f}(\mathbf{y})$ is the equivalent vector field, both being of n th order. Take now the variational equation defined by

$$\dot{\mathbf{v}} = \mathbf{J}(\mathbf{y})\mathbf{v}, \quad (24)$$

where $\mathbf{J}(\mathbf{y}) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix, and $\mathbf{v} \in \mathbb{R}^n$. Eq. (24) is an initial value problem with a solution in the form of $\mathbf{v} = \mathbf{V}\mathbf{v}(0)$, where $\mathbf{V}$ meets

$$\dot{\mathbf{V}} = \mathbf{V}\mathbf{v}(0), \quad (25)$$

where $\mathbf{V}(0) = \mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix. Finally, LEs can be computed by

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{\ln \|\eta_i\|}{t}, \quad (26)$$

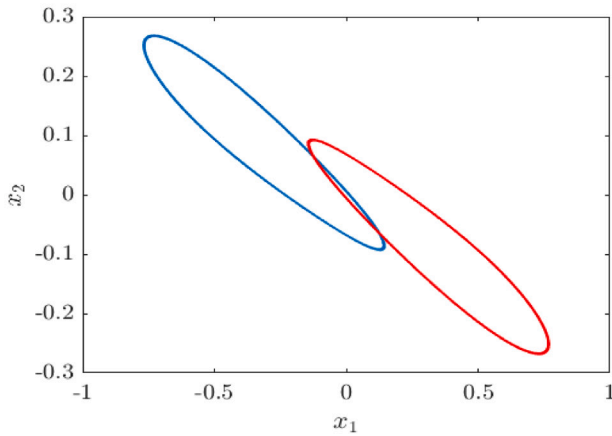
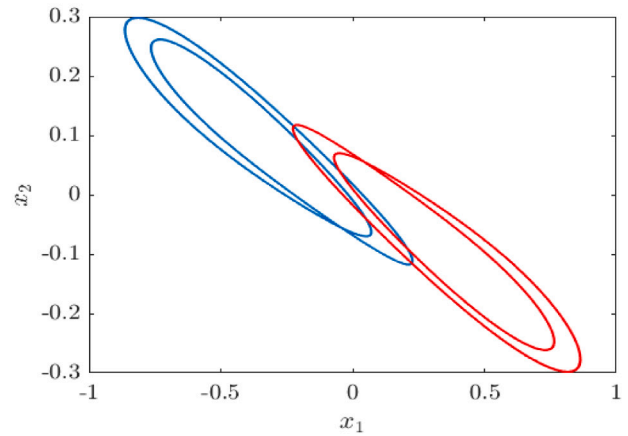
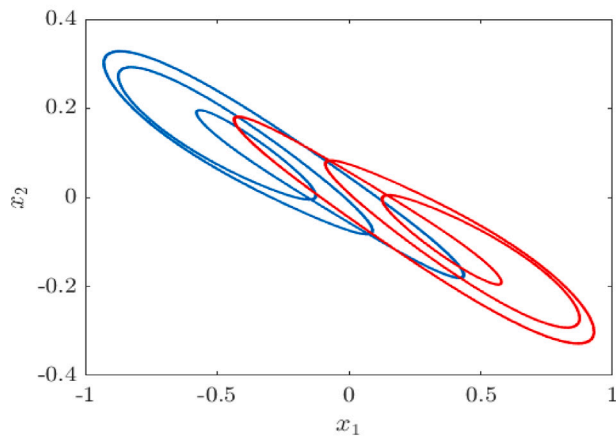
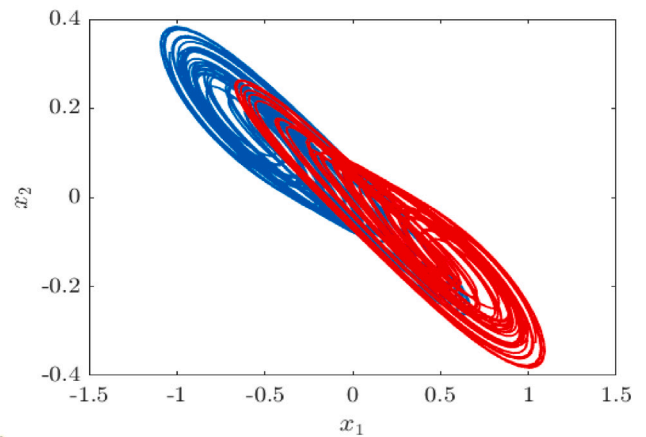
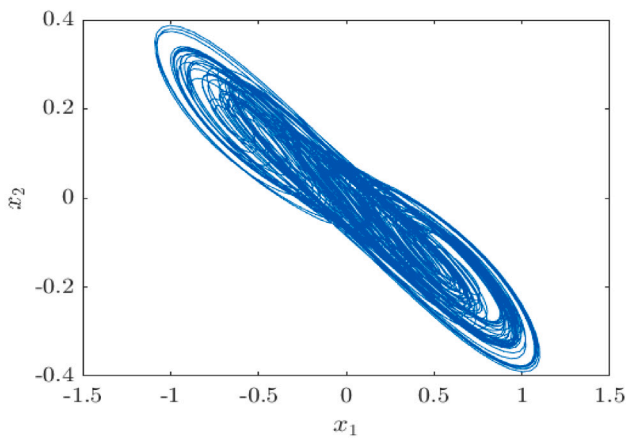
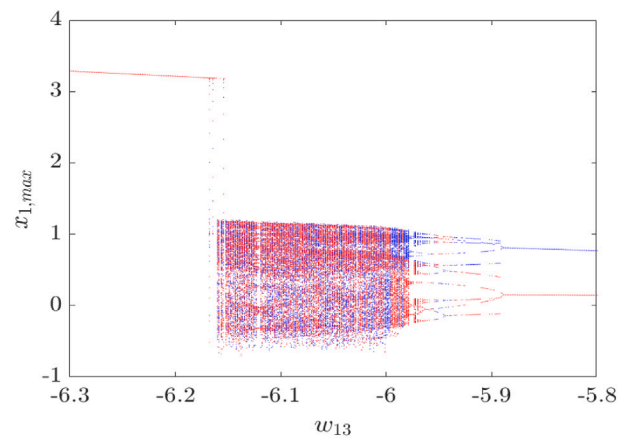
(a) Phase portrait of HNN system with period 1 behavior and $w_{13} = -5.8$.(b) Phase portrait of HNN system with period 2 behavior and $w_{13} = -5.9$.(c) Phase portrait of HNN system with period 3 behavior and $w_{13} = -5.91$.(d) Single scroll chaotic attractor with $w_{13} = -6$.(e) Double scrolls chaotic attractor with $w_{13} = -6$ and a simulation time of 5000 s.(f) Bifurcation diagram of local maxima of x_1 .

Fig. 3. Attractors and bifurcation diagram of local maxima form by the considered system in Eq. (3) with fixed $w_{12} = -1$, initial conditions $(0.5, -0.2, 0)$ in red and $(-0.5, 0.2, 0)$ in blue for phase space portraits and $(0.01, 0, 0)$ in red and $(-0.01, 0, 0)$ for bifurcation diagrams.

where η_i represent the axis length of an infinitesimal ellipsoid computed by $\eta = V\eta_i(0)$, where $\eta_i(0)$ is the orthogonal basis.

Based Eq. (26) 231361 LS were calculated, results are plotted in Figure in 5 with 481 LE's per line and 481 LE's per column. By this computation, the rich dynamics in the proposed system become more evident. Fig. 5(a) shows the various dynamics that arise in the

considered system by varying w_{12} and w_{13} in the range of -6 to -0.5 and -30 to -2 respectively, the color map is developed as follows: in blue we colored chaotic dynamics, white represents the zone with limit cycle behavior and black represents quasiperiodic motion. This figure shows the rich dynamics in the proposed HNN model with a vast parameter zone of chaotic dynamics and, as it can be seen, the more

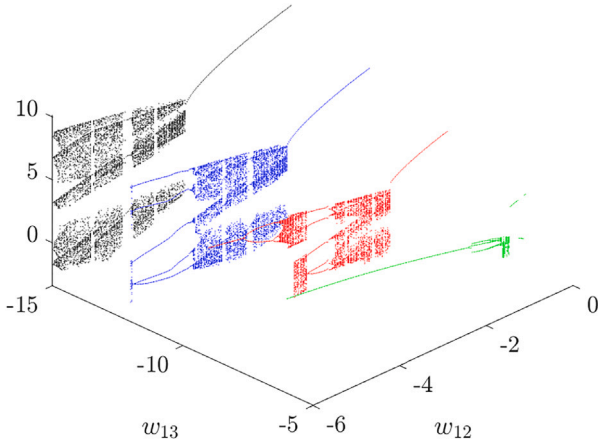


Fig. 4. Bifurcation diagram of local maxima of x_1 for the considered model in Eq. (3).

negative value for either w_{12} or w_{13} we choose the more significant are the parametric zones with chaotic dynamics, this resembles the result found by the bifurcation diagrams shown in the previous section.

To explore this last assumption further, we develop Fig. 5(c) in which we found the LEs by fixing $w_{12} = -100$ and varying $-550 \leq w_{13} \leq -85$. Then, we can see by this numerical experiment that chaotic dynamics are still present in the system with more extensive parameter zones for w_{13} to be chosen by fixing w_{12} as a more negative value.

As we know, the largest positive Lyapunov exponent (LLE) indicates chaos. The specific value for the LLE has a direct impact on the analysis of the system, mainly the Lyapunov time T_L can be computed, which is given by

$$T_L = 1/\lambda_1, \quad (27)$$

where λ_1 is the LLE. The Lyapunov time defines the predictability horizon on the HNN model, meaning the smaller T_L is, the more “unpredictable” the system is.

LE's can also be used to evidence the complexity of an attractor by finding the Kaplan–Yorke fractional dimension D_{KY} [27] defined by

$$D_{KY} = j + \frac{\sum_{i=1}^j \lambda_i}{|\lambda_{j+1}|}, \quad (28)$$

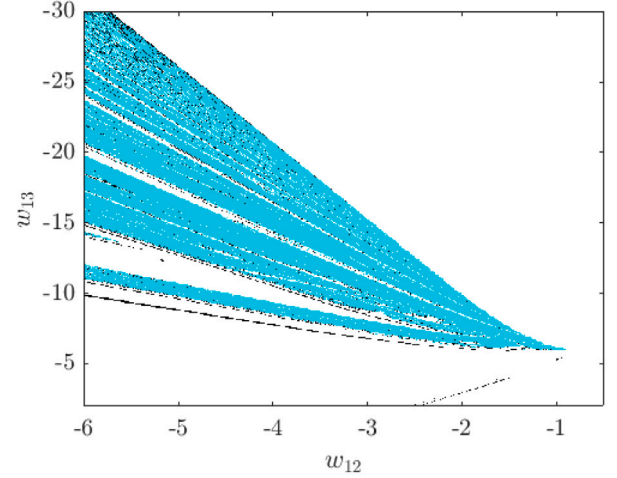
where j is the largest integer that satisfies

$$\sum_{i=1}^j \lambda_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{j+1} \lambda_i < 0. \quad (29)$$

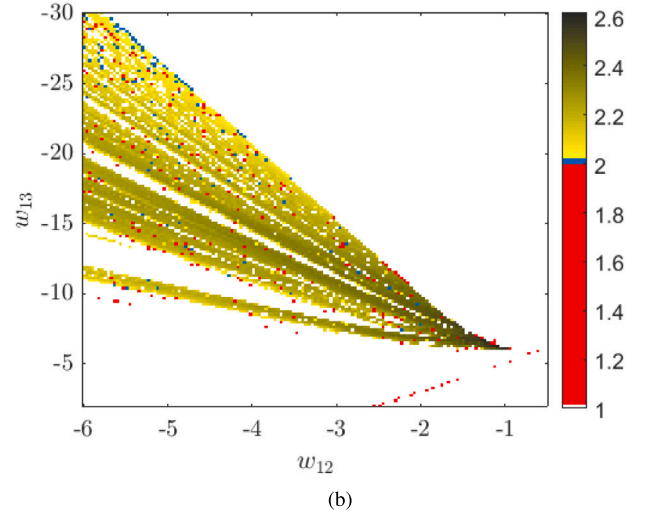
Based on this expression, we develop Fig. 5(b) with the Kaplan–Yorke fractional dimension. Due to the Kaplan–Yorke dimension being a measure of the “complexity” of the chaotic attractor, for the proposed system, the Kaplan–Yorke dimension is somewhat high considering that it is higher than the Lorenz system (2.062) or the Chen system (2.168) [28] and for the chosen integer parameters $w_{12} = -1$ and $w_{13} = -6$, equal to 2.438. Also, we found that the highest Kaplan–Yorke dimension is 2.636 for $w_{13} = -6.025$ and $w_{12} = -0.924$.

3.2.4. Basin of attraction

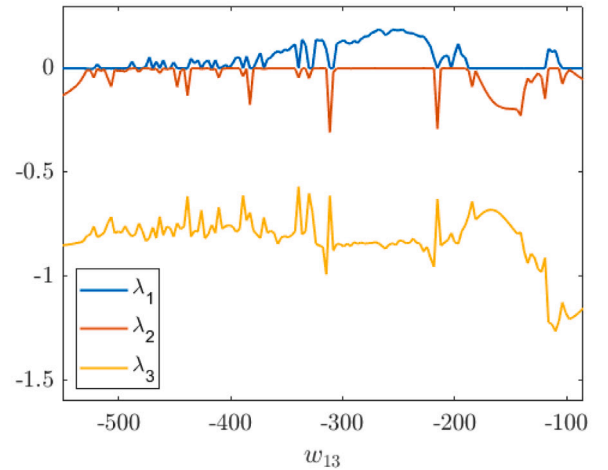
One of the peculiarities of nonlinear systems is their dependence on the behavior of the initial conditions. This characteristic is known as the basin of attraction. In this sense, the initial conditions cannot be chosen at random since some of them could lead to undesired or unexpected behavior. Thus, a numerical analysis was performed to find appropriate initial states for which the considered system with weights $w_{12} = -1$ and $w_{13} = -6$ presents the expected long-term dynamics. In Fig. 6, we present the basin of attraction for the different dynamics found given some initial state $-1.5 \leq x_1(0) \leq 1.5$, $-0.5 \leq x_2 \leq 0.5$ and fixing $x_3 = 0$



(a)



(b)



(c)

Fig. 5. Different information related to LE such as: (a) diverse dynamics in the system based on Table 3, where chaotic behavior is in red, limit cycle in white, quasiperiodic in black and in blue we find regions with all zeros LE's; (b) Kaplan–Yorke dimension and (c) Lyapunov spectrum for $w_{12} = -100$; and $-550 \leq w_{13} \leq -85$.

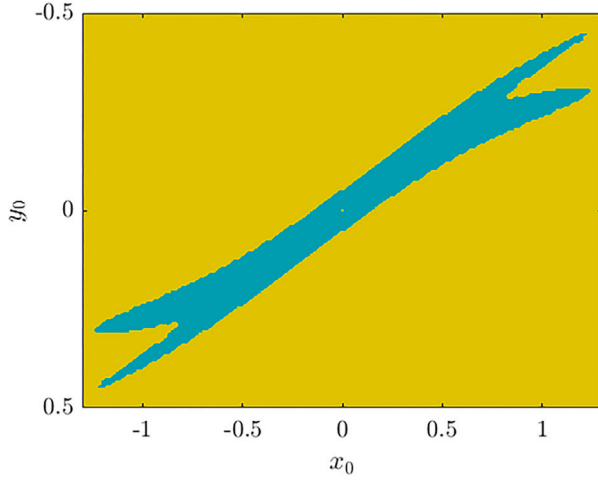


Fig. 6. Basin of attraction for $w_{12} = -1$ and $w_{13} = -6$ in the model described by Eq. (3). Blue represents a chaotic zone.

in LS basis, the chaotic dynamics in the color map is colored in blue, in yellow the attractor is a limit cycle or a fixed point. Note that only the trivial PE of the system is a fixed point.

Another basin of attraction analysis was done by choosing integer parameters $w_{12} = -2$ and $w_{13} = -7$. In this case, we explore initial conditions in the range of $-10 \leq x_1(0) \leq 10$ and $-10 \leq x_2(0) \leq 10$ and find that the system behaves chaotically for all non-PE. Next, the parameter set $w_{12} = -2$ and $w_{13} = -7$ can be used in an application for which the basin of attraction for chaotic dynamics must be bigger.

3.3. Electronic implementation on configurable embedded board

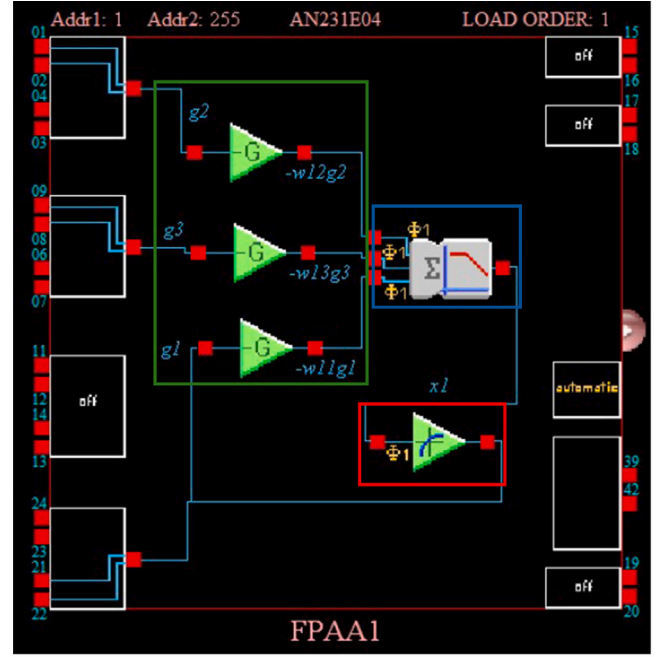
When working with dynamic systems, electronic circuits are a valuable tool for fully developing an analysis and verifying the result obtained numerically. This section presents an analog circuit implementation on a programmable analogical board with three AN231E04 FPAA's. For this, first, we find the response on the frequency of the system through the Laplace transform; we describe the topology of the system in AnadigmDesigner2 (AN2) to further implement the circuit in the configurable analogical board.

Field-Programmable Analog Array (FPAA) are integrated circuits that allow the design and implementation of linear and non-linear analog circuits through the technology of commutating capacitors. Specifically, the QUAD FPAA Development Kit was used to implement the proposed HNN model in an electronic environment. To program the board, a block-based code is developed in AN2 with the topology of the circuit. The design of the electronic implementation requires first to obtain the Laplace transform $\mathcal{L}\{\cdot\}$ of Eq. (1), with $c_i = 1$ for all $(i = 1, 2, \dots, n)$ and constant b_i , this transform yields

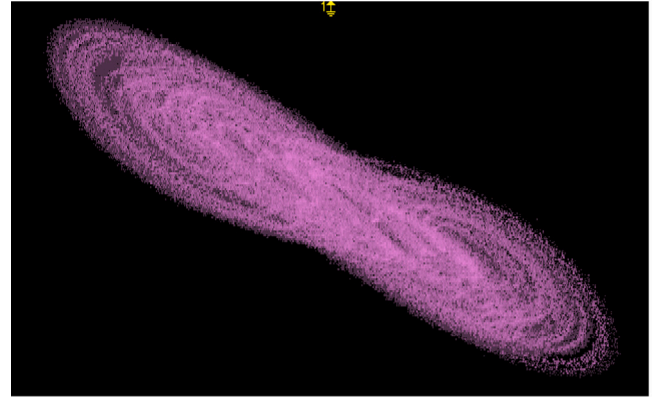
$$\begin{aligned} sX_1(s) &= -X_1(s) + \sum_{i=1}^n (w_{1i}\mathcal{L}\{g_i(x_i)\} + b_1), \\ sX_2(s) &= -X_2(s) + \sum_{i=1}^n (w_{2i}\mathcal{L}\{g_i(x_i)\} + b_2), \\ &\vdots \\ sX_n(s) &= -X_n(s) + \sum_{i=1}^n (w_{ni}\mathcal{L}\{g_i(x_i)\} + b_n). \end{aligned} \quad (30)$$

Then by solving for all $X_i(s)$ and scaling the system in frequency by a factor of $2\pi f_0$ we get

$$\begin{aligned} X_1(s) &= \frac{2\pi f_0}{s+2\pi f_0} \sum_{i=1}^n (w_{1i}\mathcal{L}\{g_i(x_i)\} + b_1), \\ X_2(s) &= \frac{2\pi f_0}{s+2\pi f_0} \sum_{i=1}^n (w_{2i}\mathcal{L}\{g_i(x_i)\} + b_2), \\ &\vdots \\ X_n(s) &= \frac{2\pi f_0}{s+2\pi f_0} \sum_{i=1}^n (w_{ni}\mathcal{L}\{g_i(x_i)\} + b_n). \end{aligned} \quad (31)$$



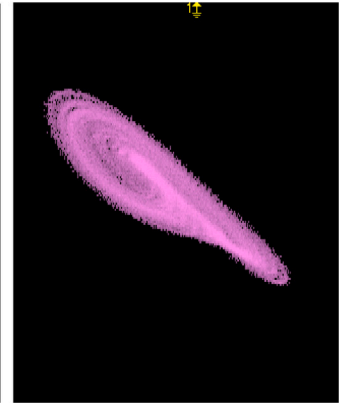
(a)



(b)



(c)



(d)

Fig. 7. (a) Topology of the first FPAA in AN2 to program the embedded analog board and physical results for phase portraits; (b) x_1 against x_2 ; (c) and (d) bi-stability of chaotic behaviors generating a single scroll.

Table 4
NIST results of the PRNG based on the proposed HNN.

Test name	P-value	Result
Frequency	0.514769	Pass
Block frequency	0.724099	Pass
Cumulative sums	0.277757	Pass
Runs	0.019260	Pass
Longest run	0.148483	Pass
Rank	0.792631	Pass
FFT	0.663355	Pass
Nonoverlapping templates	0.463600	Pass
Overlapping templates	0.570263	Pass
Universal	0.013716	Pass
Approximate entropy	0.028763	Pass
Random excursions	0.548851	Pass
Random excursions variant	0.019455	Pass
Serial	0.119676	Pass
Linear complexity	0.508662	Pass

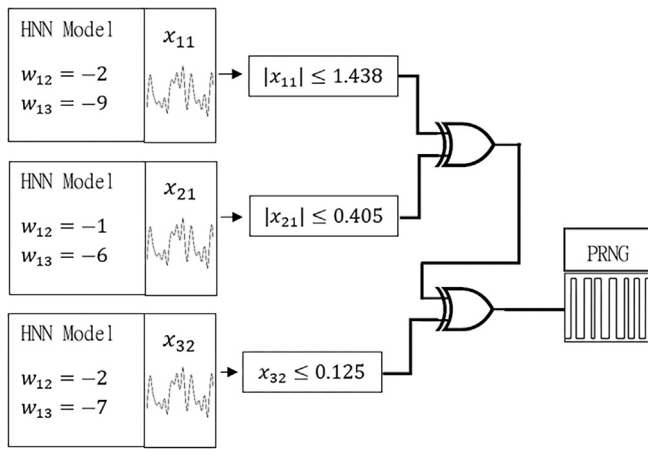


Fig. 8. Diagram for a PRNG based on the proposed HNN model.

The Eq. (31) has the form of a low-pass filter with a summing gain stage, a specific block exists in AN2 with this purpose named “SumFilter”, with equivalent transfer function given by

$$V_{out}(s) = \frac{2\pi f_0 (\pm G_1 V_{in-1}(s) \pm G_2 V_{in-2}(s) \pm G_3 V_{in-3}(s))}{s + 2\pi f_0}. \quad (32)$$

where $V_{in-1}(s)$, $V_{in-2}(s)$, and $V_{in-3}(s)$ can be set as an activation function used in the HNN model. With respect to the gains G_i , while they can be set as the weights in the SMW, their values are limited by the structure of the “SumFilter” block, then the block “GainInv” is used instead to set the appropriated weights, and allow us to implement a gain for values from 0.01 to 100.

For the non-linear activation function, an approximation can be performed using the block “TransferFunction”, which maps values from -3 V to 3 V with a look-up table of 8 bits. The topology of the circuit for the first differential equation in Eq. (31) in AN2 is shown in Fig. 7a, in green we highlight the gain module for each weight in a 3-D HNN model, in red the block with the non-linear activation function and in blue the “SumFilter” block.

Based on this topology, we describe the proposed model in Eq. (3), and then the implemented model with the application of the Laplace transform is as in Eq. (31) yields

$$\begin{aligned} X_1^*(s) &= \frac{2\pi f_0}{s+2\pi f_0} (-\mathcal{L}\{\tanh(x_2^*)\} - 4.5\mathcal{L}\{\tanh(x_3^*)\}), \\ X_2^*(s) &= \frac{2\pi f_0}{s+2\pi f_0} (\mathcal{L}\{\tanh(2x_1^*)\} + 3\mathcal{L}\{\tanh(x_2^*)\}) \\ &\quad + \frac{2\pi f_0}{s+2\pi f_0} (3\mathcal{L}\{\tanh(x_3^*)\}), \\ X_3^*(s) &= \frac{2\pi f_0}{s+2\pi f_0} (-\mathcal{L}\{\tanh(2x_1^*)\} - 3\mathcal{L}\{\tanh(x_2^*)\}). \end{aligned} \quad (33)$$

Algorithm 1 Pseudocode for generating n PRNG.

```

1: Input:  $n, h, r, T_{tr}, f_1, f_2, f_3$ 
2: Output:  $Y[1 : n]$ 
3:  $X_1 = [x_{11}, x_{12}, x_{13}] = [0.01, 0, 0]$ 
4:  $X_2 = [x_{21}, x_{22}, x_{23}] = [0.01, 0, 0]$ 
5:  $X_3 = [x_{31}, x_{32}, x_{33}] = [0.01, 0, 0]$ 
6: for  $k \leftarrow 1$  to  $T_{tr}/h$  do
7:    $X_1 = X_1 + hf_1(X_1)$ 
8:    $X_2 = X_2 + hf_2(X_2)$ 
9:    $X_3 = X_3 + hf_3(X_2)$ 
10: end for
11: for  $k \leftarrow 1$  to  $n$  do
12:   for  $i \leftarrow 1$  to  $rh$  do
13:      $X_1 = X_1 + hf_1(X_1)$ 
14:      $X_2 = X_2 + hf_2(X_2)$ 
15:      $X_3 = X_3 + hf_3(X_2)$ 
16:   end for
17:    $c_1 = |x_{11}| \leq 1.438$ 
18:    $c_2 = |x_{21}| \leq 0.405$ 
19:    $c_3 = x_{32} \leq 0.125$ 
20:    $Y[k] = (c_1 \oplus c_2) \oplus c_3$ 
21: end for
22: return  $Y[1 : n]$ 

```

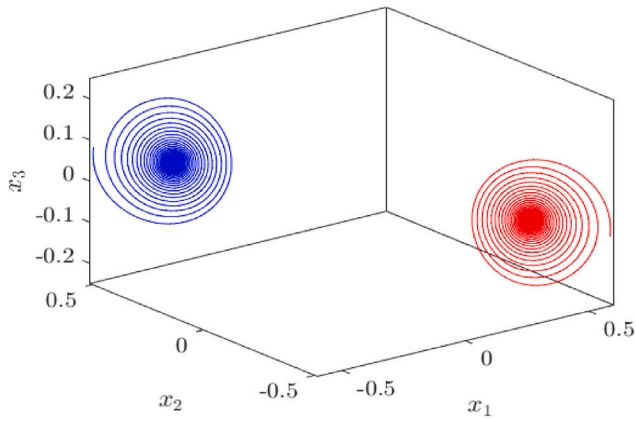
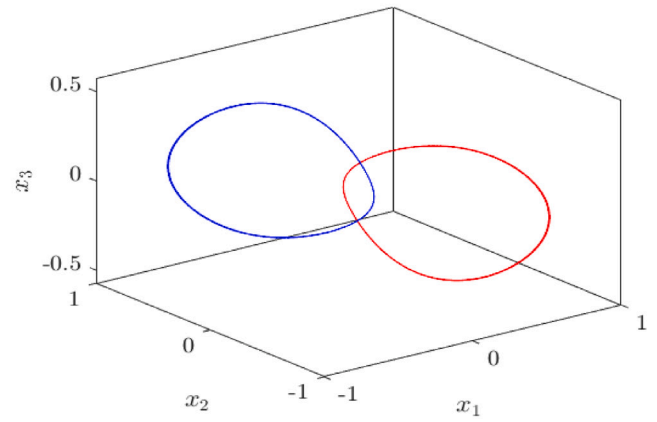
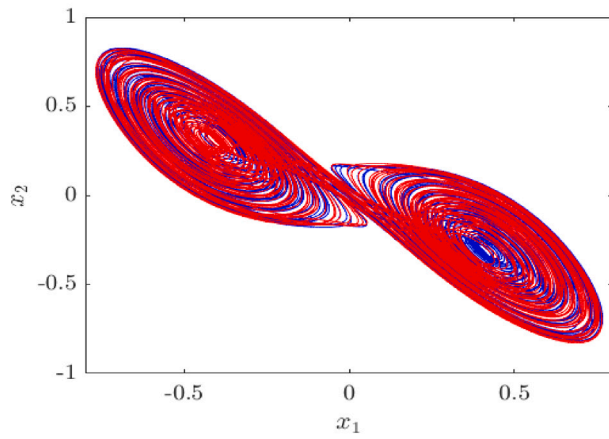
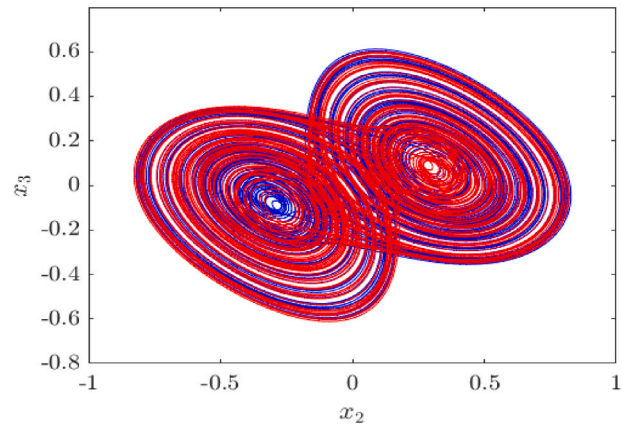
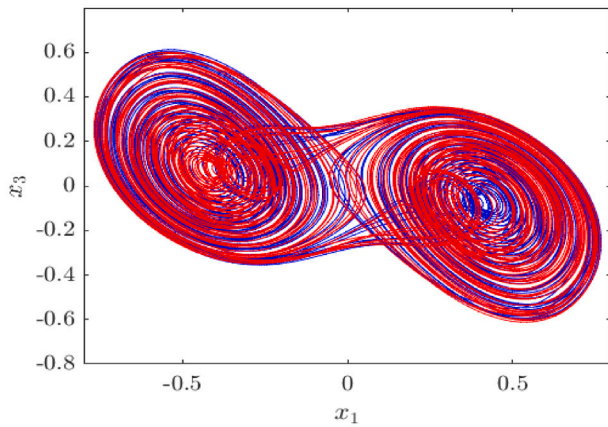
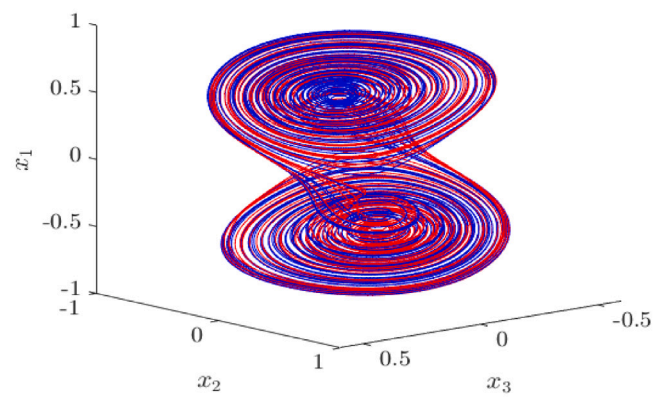
The value for f_0 can be directly set in the “SumFilter” block, in this case $f_0 = 1$ kHz. The result of the implementation in FPAAs is shown in Fig. 7 with three phase space portraits, $x_1 - x_2$, in Fig. 7(b) the HNN model present chaos behavior with two present scrolls, and 7(c), 7(c) show the dynamic of the HNN with only one scroll.

3.4. PRNG based on the proposed HNN

As was mentioned, chaotic systems have become a useful tool in engineering to solve a handful of problems, including the generation of pseudo-random patterns [29]. Such pseudo-random number generators (PRNG) are the basis of security applications. The development of chaos-based PRNG requires selecting a set of parameters called encryption keys [30]. A preliminary scheme of a PRNG based on the proposed HNN model is presented in Fig. 8. Such a proposal considers the application of XOR elements and three HNN models. A state variable is chosen from each HNN model, whereas a comparator is used to determine logical levels. The first HNN model is denoted as X_1 with encryption keys $w_{12} = -2$ and $w_{13} = -9$ with state variables x_{11} , x_{12} and x_{13} . The second HNN model is named X_2 , which has the encryption keys $w_{12} = -1$ and $w_{13} = -6$, with state variables x_{21} , x_{22} and x_{23} . Finally, the third HNN model is X_3 with encryption keys $w_{12} = -2$ and $w_{13} = -7$, and state variables x_{31} , x_{32} , x_{33} . Thus, the diagram of Fig. 8 is equivalent to the pseudocode presented in the Algorithm 1, which computes n pseudo-random numbers (PRN). To solve the set of differential equations that describes the HNN model, the Euler method is used with an integration step h . The algorithm saves a PRN every r integration step. A final parameter t_{tr} is introduced to delete the transient. To evaluate the statistical properties of PRNG, we use the NIST 800-22 test suite [31]. The results of the NIST standard using 1 Mb are shown in Table 4, proving that PRNG is feasible for security applications.

4. Fractional order HNN model

The study of fractional order dynamical systems and their applications have been one of the main topics in recent years in the field of dynamical systems. It allows for another degree of freedom to the system and provides more dynamics. There exist multiple approaches to computing the fractional order integral of a function, the most

(a) Fixed point behavior for $q = 0.78$.(b) Period 1 for $q = 0.79$.(c) Plane x_1 - x_2 with $q = 0.8$.(d) Plane x_1 - x_3 with $q = 0.8$.(e) Plane x_1 - x_3 with $q = 0.8$.(f) 3D plane of FOHNN model with $q = 0.8$.**Fig. 9.** Diverse dynamics for FOHNN model described by Eq. (37) with two initial conditions: $(0.5, -0.2, 0)$ in red and $(-0.5, 0.2, 0)$ in blue.

used being the definitions from Riemann–Liouville, Grünwald–Letnikov and Caputo [32]; this last one is the most appropriate in engineering because it uses integer order initial conditions. To provide the Caputo definition for a fractional-order integral, first, let us introduce some notation.

Definition 1. The integrodifferential operator denoted as ${}_a D_t^n$ is a combination between the derivative and integral operators as follows

$${}_a D_t^q f(t) = \begin{cases} \frac{d^n}{dt^n} f(t), & q < 0, \\ f(t), & q = 0, \\ \int_a^t f(\tau)(d\tau)^q, & q > 0. \end{cases} \quad (34)$$

Based on the last notation, we define the Caputo fractional-order derivative as follows:

Definition 2. Caputo's fractional-order derivative of a function f with dependence on time t is given by

$${}_t^C D_t^q f(t) := \frac{1}{\Gamma(m-q)} \int_{t_0}^t (t-\tau)^{m-q-1} f^{(m)}(\tau) d\tau, \quad (35)$$

where C refers to the Caputo type of fractional-order derivative used and in the next of this work will be obviated, $0 \leq t_0 < t$, $q \in \mathbb{R}$ is the order of fractional integration which meets $m-1 < q \leq m$ and $m \in \mathbb{R}^+$.

Then, we introduced the proposed HNN model of fractional order as follows:

$$\begin{aligned} {}_0 D_t^q x_1 &= -x_1 + 0.45 \tanh x_1 - 2.5 \tanh x_2 - 9 \tanh x_3, \\ {}_0 D_t^q x_2 &= -x_2 + \tanh x_1 + 3.5 \tanh x_2 + 3 \tanh x_3, \\ {}_0 D_t^q x_3 &= -x_3 - \tanh x_1 - 3 \tanh x_2 + 0.5 \tanh x_3. \end{aligned} \quad (36)$$

In this system, the x_1 space state variable is bounded by $|x_1| < 4$, since the aim of this section is to implement this fractional order model in FPAA with work limit voltages of $\pm 3V$, a scaling in amplitude is necessary, then a change of variable driven as $x_1 \rightarrow 4x_1$, giving a scaled model as

$$\begin{aligned} {}_0 D_t^q x_1 &= -x_1 + 0.1125 \tanh 4x_1 - 0.625 \tanh x_2 \\ &\quad - 2.25 \tanh x_3, \\ {}_0 D_t^q x_2 &= -x_2 + \tanh 4x_1 + 3.5 \tanh x_2 + 3 \tanh x_3, \\ {}_0 D_t^q x_3 &= -x_3 - \tanh 4x_1 - 3 \tanh x_2 + 0.5 \tanh x_3. \end{aligned} \quad (37)$$

A relevant result for fractional calculus is the result of taking the Laplace transform $\mathcal{L}\{\cdot\}$ of a fractional-order derivative; this one yields:

$$\mathcal{L}\{{}_0 D_t^q f(t); s\} = s^q F(s) - \sum_{k=0}^{n-1} s^k [{}_0 D_t^{n-k-1} f(t)]_{t=0}, \quad (38)$$

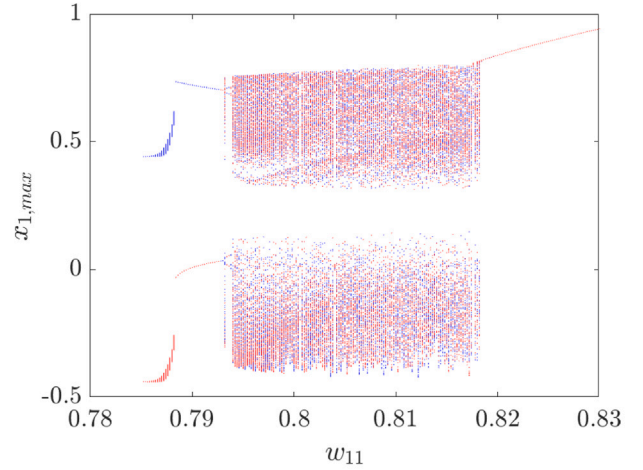
where $n \in \mathbb{Z}$ such that $n-1 < q \leq n$, and with initial conditions equal to zero we get

$$\mathcal{L}\{{}_0 D_t^q f(t); s\} = s^q F(s). \quad (39)$$

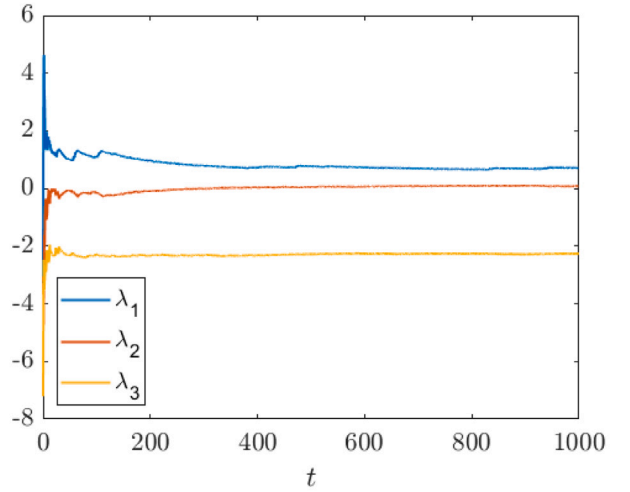
4.1. Simulation results of the fractional order HNN model

To find the solution of the fractional order differential equations described in Eq. (37), we used the work by Garrapa R. [33], based on the Adams–Bashforth–Moulton (ABM) method to estimate a numerical solution of the proposed HNN system. Fig. 9 presents the diverse behavior found while fixing $w_{12} = -2$, $w_{13} = -9$ and varying the fractional integration order $0.78 \leq q \leq 0.83$.

First, we notice that bi-stability is preserved, to denote it we used to different initial states $(-0.5, 0.2, 0)^T$ in blue and $(0.5, -0.2, 0)^T$ in red. Further, different behavior for the self excited attractors were found, by choosing $q = 0.78$ the system presents a fixed point type dynamic



(a) Bifurcation diagram of local maxima of x_1 for fractional order system in Eq. (37).

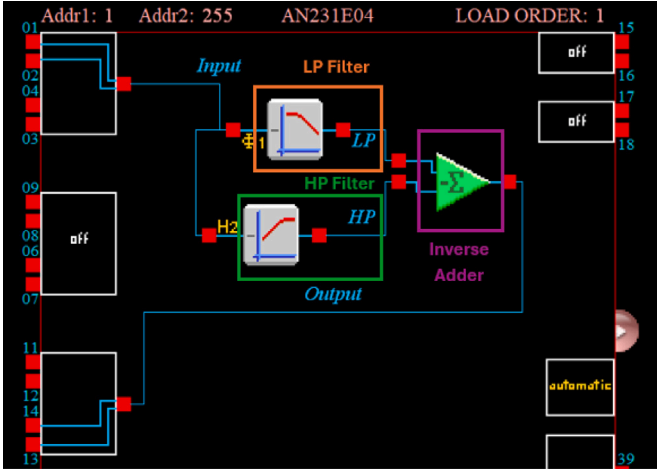
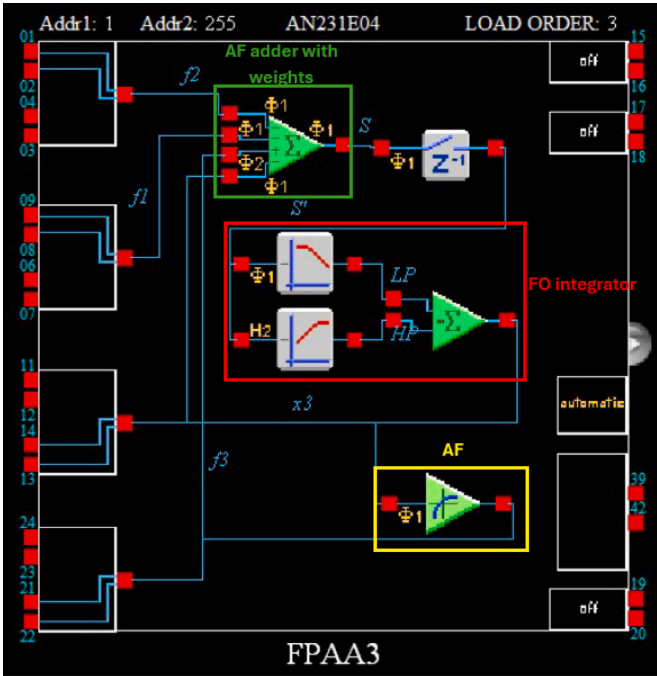


(b) LEs for fractional order system in Eq. (37) with $q = 0.8$.

Fig. 10. Two metrics to identify chaos in fractional order HNN model described in Eq. (37).

(see Fig. 9(a)), this especially interesting because this dynamic was not found in the integer order HNN model. Then, different periodic behaviors were found for the selecting values in the $0.787 \leq q \leq 0.7948$; one example is shown in Fig. 9(b). Second, for $q > 0.795$, chaos is found. Figs. 9(c)–9(e) show the different phase portraits of the system in the 2D plane, while Fig. 9(f) presents the simulation of the system in a 3D plane.

Two different metrics were used to identify chaos in fractional order systems. First, the bifurcation diagram of local maxima of x_1 was also computed and is shown in Fig. 10a, just as in the integer order system, the bifurcation diagram summarize the behavior before explained. To find such a diagram, the system is simulated during 900 s, not considering a transient time of 450 s and integration step of 0.01. Second, the LEs of the system in Eq. (37) were found with $q = 0.8$ using the work by Hang Li, Yongjun Shen et al. [34]. The results after 1000 s, an orthonormalization time of 0.2 s, and an integration step equal to 0.002 for the LEs are shown in Fig. 10b with values $\lambda_1 = 0.713$, $\lambda_2 = 0.082$ and $\lambda_3 = -2.265$, which confirms the chaotic nature of the considered fractional order system.

(a) First order approximation of s^{-q} in AnadigmDesigner2 environment.

(b) Circuit topology to program one space state variable in AnadigmDesigner2.

Fig. 11. Block-base program to implement the fractional-order HNN model.

4.2. First order filters to approximate s^{-q} and their FPAA-based design

In fractional calculus and signal processing, the expression s^{-q} , where s is a complex frequency variable in the Laplace domain, and q is a real number, represents an operation that generalizes integer-order integrals and derivatives [10]. Unlike integer powers of s , fractional powers introduce challenges in computation and interpretation, especially when q is non-integer. This term often arises in systems with memory or hereditary properties, where responses are not instantaneous and depend on past states in a non-integer order. Most of the time, an approximation is conducted to approximate the expected behavior of s^{-q} , specifically in [35] proposes an approximation based on rational functions as

$$s^{-q} \approx \frac{P(s)}{Q(s)}, \quad (40)$$

where $Q(s)$ and $P(s)$ are chosen to approximate the frequency response of s^{-q} . In this work, we use the first order of this approximation given

by

$$s^{-q} \approx \frac{(1-q)s + (1+q)}{(1+q)s + (1-q)} = \frac{As + 1}{s + A}, \quad (41)$$

where $A = \frac{1-q}{1+q}$. The expression given by Eq. (41) can be interpreted as the sum of a Low-Pass and a High-Pass filters as

$$s^{-q} \approx \frac{As}{s + A} + \frac{1}{s + A}. \quad (42)$$

Taking the Laplace transform of Eq. (37) we find

$$\begin{aligned} \mathcal{L}\{D_t^q x_1\} &= -X_1(s) + 0.45\mathcal{L}\{\tanh 4x_1\} - 0.625\mathcal{L}\{\tanh x_2\} \\ &\quad - 2.25\mathcal{L}\{\tanh x_3\}, \\ \mathcal{L}\{D_t^q x_2\} &= -X_2(s) + \mathcal{L}\{\tanh x_1\} + 3.5\mathcal{L}\{\tanh x_2\} \\ &\quad + 3\mathcal{L}\{\tanh x_3\}, \\ \mathcal{L}\{D_t^q x_3\} &= -X_3(s) - \mathcal{L}\{\tanh x_1\} - 3\mathcal{L}\{\tanh x_2\} \\ &\quad + 0.5\mathcal{L}\{\tanh x_3\}, \end{aligned} \quad (43)$$

then taking the expression in Eq. (39) and the approximation of s^{-q} given by Eq. (42) yields

$$X_1(s) = \left(\frac{As}{s + A} + \frac{1}{s + A} \right) \mathcal{L}\{-x_1 + 0.45 \tanh x_1 - 2.5 \tanh x_2 - 9 \tanh x_3\}, \quad (44)$$

$$X_2(s) = \left(\frac{As}{s + A} + \frac{1}{s + A} \right) \mathcal{L}\{-x_2 + \tanh x_1 + 3.5 \tanh x_2 + 3 \tanh x_3\}, \quad (45)$$

$$X_3(s) = \left(\frac{As}{s + A} + \frac{1}{s + A} \right) \mathcal{L}\{-x_3 - \tanh x_1 - 3 \tanh x_2 + 0.5 \tanh x_3\}. \quad (46)$$

In this work, we are mainly interested in $q = 0.8$, which gives $A \approx 0.1111$, and then, we find the first order approximation as

$$s^{-0.8} \approx \frac{0.1111s}{s + 0.1111} + \frac{1}{s + 0.1111}. \quad (47)$$

These order filters are implemented in the configurable analog board using the circuit topology presented in Fig. 11(a), in orange we highlight the block for the Low-Pass filter with a transfer function equivalent to

$$G(s) = \pm \frac{2\pi f_0 G_{LP}}{s + 2\pi f_0}, \quad (48)$$

where G_{LP} is a variable gain and f_0 is the corner frequency at which the gain is $-3 + 20 \log(G)$ dB. In green is highlighted the High-Pass filter with the transfer function

$$G(s) = -\frac{G_{HP}s}{s + 2\pi f_0}, \quad (49)$$

where G_{HP} is a variable gain and f_0 the corner frequency. By setting $G_{LP} = 1$, $G_{HP} = 0.111$ and $f_0 = 0.111$ we configure the circuit in Fig. 11(a) as an approximation for $s^{-0.8}$. In Fig. 11(b), we present the circuit topology to describe the Eqs. in (44), (45) and (46).

4.3. Implementation results of fractional order HNN model

In Fig. 12 we show the result of the implementation of the fractional order HNN model with $q = -0.8$ using the circuit designed in Section 4.2, four different phase portraits are shown. Fig. 12(a) shows the time series for x_1 and x_2 in yellow and green, respectively. Further, Figs. 12(b)–12(d) present the phase portraits of the fractional order system $x_1 - x_2$ plane, $x_1 - x_3$ plane and $x_2 - x_3$ respectively. From these results, we can see a clear resemblance with the numerical simulations previously presented.

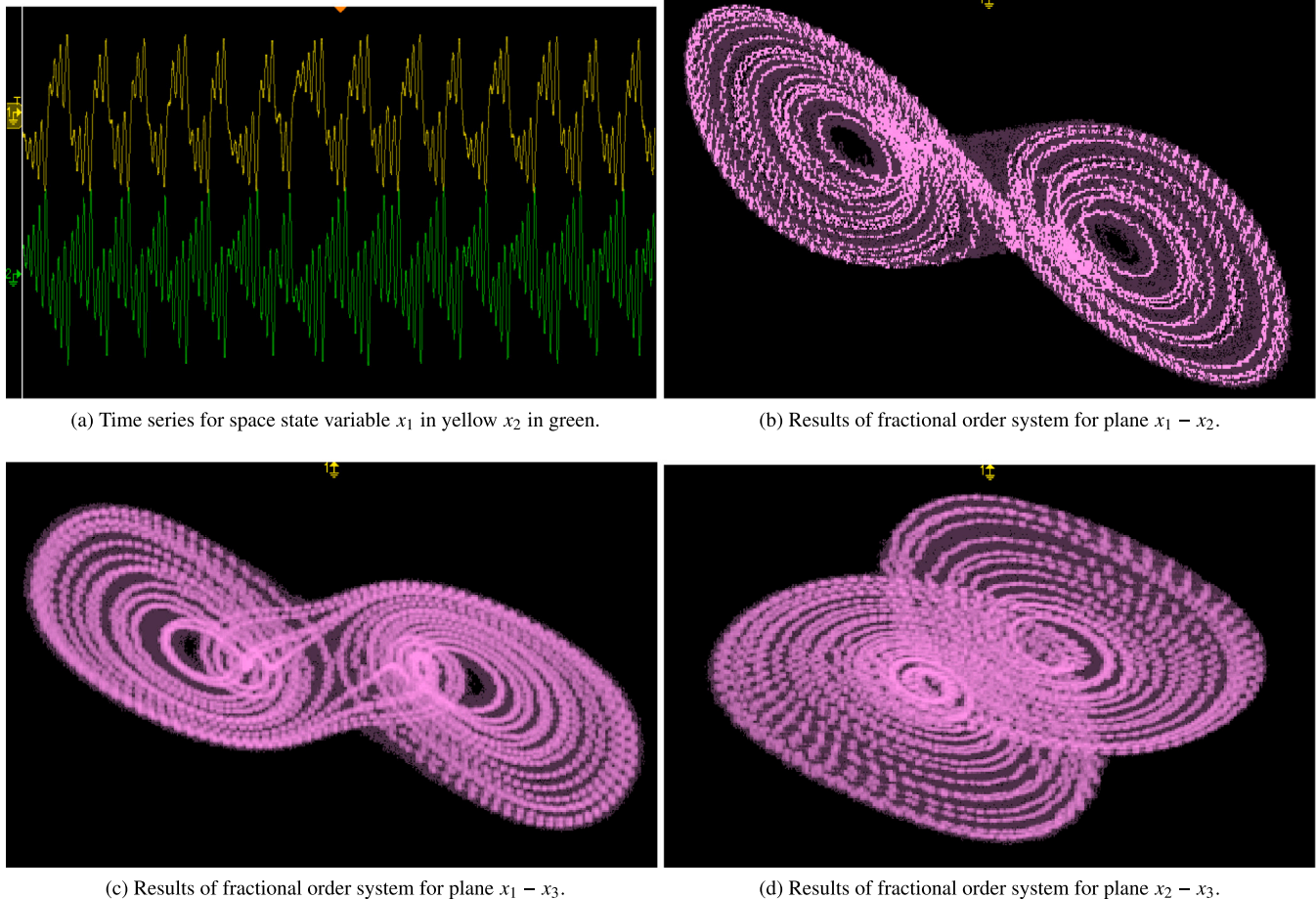


Fig. 12. FPPA configuration and circuit implementation for system in Eq. (37) with $q = 0.8$.

5. Conclusion

This paper proposed an elegant synaptic weight matrix for which the HNN model shows rich and complex dynamical behaviors such as chaos and bi-stability, even though the elements of the weight matrix are integers. To demonstrate that rich dynamics, dissipation property, stability analysis, basin of attraction, bifurcation diagrams, and Lyapunov exponents were presented. Moreover, an analog circuit was designed and implemented on FPAAs for the integer order HNN model. Finally, a fractional order analysis was conducted by applying bifurcation diagrams and Lyapunov exponents to identify the type of behavior on the fractional order side of the considered system. Also, an analog circuit was designed and correctly implemented using an approximation for the s^{-q} operator. In addition, a PRNG was designed and evaluated using the standard NIST 800–22 tests to demonstrate the effectiveness of the proposed HNN in security applications.

In future works, the presented Hopfield neural network may be used to obtain compact hardware implementations in security applications and pattern recognition, among others.

Funding

This research received no external funding.

CRedit authorship contribution statement

Diego S. de la Vega: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Jesus M. Munoz-Pacheco:** Conceptualization, Methodology, Software,

Validation, Writing – original draft, Writing – review & editing. **Olga G. Félix-Beltrán:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Christos Volos:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author thanks CONAHCYT/Mexico for the support through the scholarship CVU 1301246. The authors acknowledge BUAP for the support via project VIEP 2024.

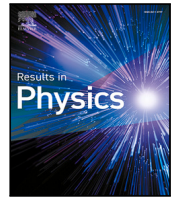
Data availability

Data will be made available on request.

References

- [1] P. Sekar, S. Narayanan, Chaos in mechanical systems — A review, *Sadhana* 20 (2–4) (1995) 529–582, <http://dx.doi.org/10.1007/BF02823207>.
- [2] Z.T. Njitacke, S.D. Isaac, J. Kengne, A.N. Negou, G.D. Leutcho, Extremely rich dynamics from hyperchaotic Hopfield neural network: Hysteretic dynamics, parallel bifurcation branches, coexistence of multiple stable states and its analog circuit implementation, *Eur. Phys. J. Spec. Top.* 229 (6–7) (2020) 1133–1154, <http://dx.doi.org/10.1140/epjst/e2020-900205-y>.

- [3] B. Bao, H. Qian, J. Wang, Q. Xu, M. Chen, H. Wu, Y. Yu, Numerical analyses and experimental validations of coexisting multiple attractors in Hopfield neural network, *Nonlinear Dynam.* 90 (4) (2017) 2359–2369, <http://dx.doi.org/10.1007/s11071-017-3808-3>.
- [4] K. Aihara, T. Takabe, M. Toyoda, Chaotic neural networks, *Phys. Lett. A* 144 (6–7) (1990) 333–340, [http://dx.doi.org/10.1016/0375-9601\(90\)90136-C](http://dx.doi.org/10.1016/0375-9601(90)90136-C).
- [5] J. Ye, P. Ji, D. Waxman, W. Lin, Y. Moreno, Impact of intra and inter-cluster coupling balance on the performance of nonlinear networked systems, 2020, <http://dx.doi.org/10.48550/ARXIV.2006.11357>, Publisher: arXiv Version Number: 1.
- [6] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci.* 79 (8) (1982) 2554–2558, <http://dx.doi.org/10.1073/pnas.79.8.2554>.
- [7] K. Rajagopal, J.M. Munoz-Pacheco, V.-T. Pham, D.V. Hoang, F.E. Alsaadi, F.E. Alsaadi, A Hopfield neural network with multiple attractors and its FPGA design, *Eur. Phys. J. Spec. Top.* 227 (7–9) (2018) 811–820, <http://dx.doi.org/10.1140/epjst/e2018-800018-7>.
- [8] J.C. Sprott, A proposed standard for the publication of new chaotic systems, *Int. J. Bifurc. Chaos* 21 (09) (2011) 2391–2394, <http://dx.doi.org/10.1142/S021812741103009X>.
- [9] R. Hilfer (Ed.), *Applications of fractional calculus in physics*, World Scientific, Singapore ; River Edge, NJ, 2000.
- [10] D. Cafagna, G. Grassi, Fractional-order chua's circuit: time-domain analysis, bifurcation, chaotic behavior and test for chaos, *Int. J. Bifurc. Chaos* 18 (03) (2008) 615–639, <http://dx.doi.org/10.1142/S0218127408020550>.
- [11] Y. Yu, H.-X. Li, S. Wang, J. Yu, Dynamic analysis of a fractional-order Lorenz chaotic system, *Chaos Solitons Fractals* 42 (2) (2009) 1181–1189, <http://dx.doi.org/10.1016/j.chaos.2009.03.016>.
- [12] C. Li, G. Chen, Chaos and hyperchaos in the fractional-order Rössler equations, *Phys. A* 341 (2004) 55–61, <http://dx.doi.org/10.1016/j.physa.2004.04.113>.
- [13] P. Kumar, T.H. Lee, V.S. Erturk, A novel fractional-order cascade tri-neuron hopfield neural network: Stability, bifurcations, and chaos, *Qual. Theory Dyn. Syst.* 23 (5) (2024) 236, <http://dx.doi.org/10.1007/s12346-024-01096-8>.
- [14] A. Yousefpour, H. Jahanshahi, J.M. Munoz-Pacheco, S. Bekiros, Z. Wei, A fractional-order hyper-chaotic economic system with transient chaos, *Chaos Solitons Fractals* 130 (2020) 109400, <http://dx.doi.org/10.1016/j.chaos.2019.109400>.
- [15] L.-J. Sheu, H.-K. Chen, J.-H. Chen, L.-M. Tam, W.-C. Chen, K.-T. Lin, Y. Kang, Chaos in the Newton–Leipnik system with fractional order, *Chaos Solitons Fractals* 36 (1) (2008) 98–103.
- [16] A. Hegazi, E. Ahmed, A. Matouk, On chaos control and synchronization of the commensurate fractional order Liu system, *Commun. Nonlinear Sci. Numer. Simul.* 18 (5) (2013) 1193–1202, <http://dx.doi.org/10.1016/j.cnsns.2012.09.026>.
- [17] S. Kumar, A. Kumar, B. Samet, J. Gómez-Aguilar, M. Osman, A chaos study of tumor and effector cells in fractional tumor-immune model for cancer treatment, *Chaos Solitons Fractals* 141 (2020) 110321, <http://dx.doi.org/10.1016/j.chaos.2020.110321>.
- [18] E. Tlelo-Cuautle, A. Dalia Pano-Azucena, O. Guillén-Fernández, A. Silva-Juárez, Analog/digital implementation of fractional order chaotic circuits and applications, first ed., Springer eBooks Engineering, Springer, Cham, 2020, <http://dx.doi.org/10.1007/978-3-030-31250-3>.
- [19] F. Yang, X. Wang, Dynamic characteristic of a new fractional-order chaotic system based on the Hopfield Neural Network and its digital circuit implementation, *Phys. Scr.* 96 (3) (2021) 035218, <http://dx.doi.org/10.1088/1402-4896/abd904>.
- [20] P.O. Mathews, J.O. Hasler, Hopfield vs Ising: A comparison on the SoC FPAA, *IEEE Trans. Circuits Syst. I. Regul. Pap.* 71 (9) (2024) 3999–4008, <http://dx.doi.org/10.1109/TCSI.2024.3411407>.
- [21] S.J. Russell, P. Norvig, *Artificial intelligence: a modern approach*, fourth ed., Pearson Series in Artificial Intelligence, Pearson, Hoboken, NJ, 2021.
- [22] R.C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*, second ed., Oxford University Press, Oxford ; New York, 2000.
- [23] J.C. Sprott, Some simple chaotic flows, *Phys. Rev. E* 50 (1994) R647–R650, <http://dx.doi.org/10.1103/PhysRevE.50.R647>, URL <https://link.aps.org/doi/10.1103/PhysRevE.50.R647>.
- [24] J.C. Sprott, Strange attractors with various equilibrium types, *Eur. Phys. J. Spec. Top.* 224 (8) (2015) 1409–1419, <http://dx.doi.org/10.1140/epjst/e2015-02469-8>.
- [25] M.S. Tavazoei, M. Haeri, Chaotic attractors in incommensurate fractional order systems, *Phys. D: Nonlinear Phenom.* 237 (20) (2008) 2628–2637, <http://dx.doi.org/10.1016/j.physd.2008.03.037>.
- [26] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory, *Meccanica* 15 (1) (1980) 9–20, <http://dx.doi.org/10.1007/BF02128236>.
- [27] P. Frederickson, J.L. Kaplan, E.D. Yorke, J.A. Yorke, The liapunov dimension of strange attractors, *J. Differential Equations* 49 (2) (1983) 185–207, [http://dx.doi.org/10.1016/0022-0396\(83\)90011-6](http://dx.doi.org/10.1016/0022-0396(83)90011-6).
- [28] Z. Wang, J. Ma, Z. Chen, Q. Zhang, A new chaotic system with positive topological entropy, *Entropy* 17 (12) (2015) 5561–5579, <http://dx.doi.org/10.3390/e17085561>.
- [29] K. Demir, S. Ergün, An analysis of deterministic chaos as an entropy source for random number generators, *Entropy* 20 (12) (2018) <http://dx.doi.org/10.3390/e20120957>.
- [30] D. Clemente-Lopez, J.J. Rangel-Magdaleno, J.M. Munoz-Pacheco, L. Morales-Velazquez, A comparison of embedded and non-embedded FPGA implementations for fractional chaos-based random number generators, *J. Ambient. Intell. Humaniz. Comput.* 14 (8) (2023) 11023–11037, <http://dx.doi.org/10.1007/s12652-022-04382-y>.
- [31] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, L. Bassham, NIST Special Publication 800-22: A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications, NIST Special Publication 800-22, 2010.
- [32] K. Diethelm, *The analysis of fractional differential equations: an application-oriented exposition using differential operators of Caputo type*, Lecture notes in mathematics, Springer-Verlag, Heidelberg ; New York, 2010, no. 2004, OCLC: ocn648933597.
- [33] R. Garrappa, Numerical solution of fractional differential equations: A survey and a software tutorial, *Math.* 6 (2) (2018) 16, <http://dx.doi.org/10.3390/math6020016>.
- [34] H. Li, Y. Shen, Y. Han, J. Dong, J. Li, Determining Lyapunov exponents of fractional-order systems: A general method based on memory principle, *Chaos Solitons Fractals* 168 (2023) 113167, <http://dx.doi.org/10.1016/j.chaos.2023.113167>.
- [35] B. Krishna, Studies on fractional order differentiators and integrators: A survey, *Signal Process.* 91 (3) (2011) 386–426, <http://dx.doi.org/10.1016/j.sigpro.2010.06.022>.



Identifying chaotic dynamics in nonuniformly dissipative and conservative dynamical systems from the varying divergence

Diego S. de la Vega, Olga G. Félix-Beltrán, Jesus M. Munoz-Pacheco [✉]*

Faculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 Sur, Puebla, 72570, Puebla, Mexico

ARTICLE INFO

Dataset link: <https://github.com/pacheco008/AEADV>

Keywords:

Chaos
Chaotic systems
Nonuniform divergence
Nonuniformly dynamical systems
Numerical simulation
Neural Networks

ABSTRACT

As well-known, chaotic dynamics can appear in both living and artificial systems, resulting in several applications in science and engineering. Thus, the persistent question is whether a determined time series represents truly chaotic behavior. Although many familiar tools exist to respond to that question, new efficient algorithms in time and complexity must be developed to cope with the striking characteristics of novel chaotic systems, e.g., systems with non-uniform divergence.

In this framework, a new metric to evaluate the chaotic behavior in nonuniformly dynamical systems using the divergence operator of the vector field is proposed. Such systems are characterized by presenting a non-constant divergence, i.e., the divergence changes as time evolves since it depends on the system states. The proposed metric can be applied in dissipative and conservative systems with non-uniform divergence. By expanding the dynamical system with the time derivative of the divergence operator, the proposed approach can identify periodic and chaotic regions from the averages of the derivative of the divergence for the first orbit (ADDFO) and second orbit (ADDFO). We evaluate the performance of the algorithm in various systems: the typical Rossler system, a three-neuron Hopfield neural network, the Pernarowski model of pancreatic beta-cells, and the Sprott D conservative system. From the numerical results, we explicitly demonstrate that the proposed metric provides an efficient algorithm regarding simulation time and complexity with the best performance compared to Lyapunov exponents and bifurcation diagrams.

Introduction

The study of dynamical systems has seen significant growth in recent years, as it allows for the analysis of complex systems. One special type of dynamical system is a chaotic system, which is a deterministic system with aperiodic behavior and extreme sensitivity to initial conditions [1]. Chaotic systems were first reported by Lorenz in 1963 [2] and have been a major focus of study within the scientific community since then. There are multiple definitions for chaos, with the most common being the one proposed by Devaney [3], which defines a chaotic system as a dynamical system $f : X \rightarrow X$ of a metric space X satisfying f is transitive, f is dense on X , and f has a sensitive dependence on initial conditions. Based on these properties, different metrics have been regularly proposed and used to identify chaotic dynamics. For instance, Lyapunov spectrum [4], bifurcation diagrams [5], Kolmogorov–Sinai entropy [6], Poincaré map [7], 0–1 test [8,9], to mention only a few. Among them, bifurcation diagrams and Lyapunov exponents are the most frequently utilized. On the one hand, bifurcation diagrams depict the qualitative transformation of a system as its parameters are altered [10]. In the context of continuous-time dynamical systems, this

transformation typically pertains to the amplitude of one of the system's state variables. Consequently, depicting local maxima or minima on the bifurcation graph facilitates the identification of diverse dynamics, such as periodicity, quasi-periodicity, and chaos. On the other hand, the Lyapunov spectrum measures the exponential separation between two infinitesimal close orbits and consists of n values called Lyapunov exponents (LEs) for an n -dimensional dynamical system. Depending on the signs of the LEs, different dynamics can be identified [4,10]. For a dynamical system with bounded solutions, a positive Lyapunov exponent indicates chaotic behavior, which is defined as the Largest Lyapunov Exponent (LLE).

In recent years, new investigation and application branches on chaotic dynamics have been opened, such as chaos-based cryptography [11,12], chaos control [13], quantum chaos [14,15], chaotic neural networks [16,17], biological systems exhibiting chaos [18], fractional-order chaotic systems [19,20], and so forth. Therefore, new algorithms for chaos detection in those new research areas are continuously being developed and refined to broaden the collection of dynamical tools available for chaos exploration in the parameter space [21–23], as

* Corresponding author.

E-mail address: jesusm.pacheco@correo.buap.mx (J.M. Munoz-Pacheco).

well as improved detection [24,25] These algorithms aim to not only expand the repertoire of techniques but also to improve computational efficiency, with a focus on reducing both time and memory usage.

One of the most relevant aspects in chaotic systems is the evolution of their volume in the phase space, called divergence. If the volume of the phase space is reduced, the system is said to be dissipative while if it is maintained, the system is called conservative. Otherwise, the system is unbounded [26]. Based on this criterion, it is possible to conclude if a system can contain attractors [27]. Let us consider a dynamical system described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is a vector containing the system variables and $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$ is the vector field. To characterize how the volume of phase space behaves, the divergence operator of the vector field $\nabla \cdot \mathbf{f}(\mathbf{x})$ is used, which is given as

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = \frac{\partial f_1(\mathbf{x})}{\partial x_1} + \frac{\partial f_2(\mathbf{x})}{\partial x_2} + \dots + \frac{\partial f_n(\mathbf{x})}{\partial x_n}. \quad (2)$$

The sign of $\nabla \cdot \mathbf{f}(\mathbf{x})$ is a well-known criterion to identify if the system is conservative ($\nabla \cdot \mathbf{f}(\mathbf{x}) = 0$), dissipative ($\nabla \cdot \mathbf{f}(\mathbf{x}) < 0$) or unbounded ($\nabla \cdot \mathbf{f}(\mathbf{x}) > 0$). However, for specific systems, the divergence depends on the states of the system. Under such circumstances, the average time of divergence along the trajectory now determines whether the system is dissipative or conservative, specifically if this is zero or nonzero. Such systems can even have characteristics that exhibit dissipation and anti-dissipation in specific time periods.

Definition 1 (Uniformly Dynamical System). A dynamical system expressed by Eq. (1) whose divergence is constant, i.e., $\nabla \cdot \mathbf{f}(\mathbf{x}) = k$, with $k \in \mathbb{R}$ being a constant value, is called Uniformly Dynamical System where the dissipative or conservative nature depends on if k is non-positive or zero, respectively [28,29].

Definition 2 (Nonuniformly Dynamical System). A dynamical system expressed by Eq. (1) whose divergence is a function of the states of the system, i.e., $\nabla \cdot \mathbf{f}(\mathbf{x}) = \Lambda(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^n$ being the set of system variables and $\Lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ a non-constant function, is called Nonuniformly Dynamical System where the dissipative or conservative nature depends on the average of states Λ as time evolves [28,29].

Apart from the classic chaotic systems such as Rossler system [30], double pendulum [31], Sprott A, D, K, O and P, etc., many others also possess a divergence that depends on states of the system [16,32,33]. In this manner, the novel contributions of this manuscript are:

- The proposed metric employs the divergence operator on dissipative and conservative nonuniformly dynamical systems (Definition 2) to determine the parametric zones with chaotic dynamics, producing an optimized algorithm in time simulation.
- The proposed metric aims to identify periodic and chaotic regions based on the time derivative of the divergence operator and the averages of the derivative of the divergence for the first orbit (ADDFO) and second orbit (ADDFO), respectively, reducing the number of data required for classification.

The basic idea works as follows. Considering that the divergence is not constant, we include the divergence operator on the right-hand side of the dynamical system (1). Then, we find the mean value of the time derivative of the divergence while the system is numerically integrated. Finally, based on the absolute error between the average values of the temporal derivative of a phase-space variable, we classify chaotic zones with less computational effort than other approaches.

The article is organized as follows. Section “Computational method” introduces the proposed metric for detecting chaos using the divergence operator. Section “Computational results” presents the new metric when it is used in four nonuniformly dynamical systems with

well-documented chaotic phenomena. The given results are compared with those obtained by the bifurcation diagrams and LLE. Section “Discussion” discusses the obtained outcomes as well as the analysis of computational cost. Finally, Section “Conclusion” summarizes the conclusions.

Computational method

Consider a dynamical system described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \beta), \quad (3)$$

where $\beta \in \mathbb{R}$ is the parameter to analyze. Now assume that system (3) is a nonuniformly dynamical system satisfying Definition 2. So, there exists a non-zero function with dependence on the state $\Lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ given by

$$\Lambda \equiv \nabla \cdot \mathbf{f}(\mathbf{x}, \beta) = \sum_{i=1}^n \frac{\partial f_i(\mathbf{x}, \beta)}{\partial x_i}. \quad (4)$$

As is well known, the divergence measures the rate at which the phase space expands or shrinks. For nonuniformly chaotic systems, two trajectories with close initial states are expected to diverge at some point because their divergence is also sensitive to initial conditions.

Then, while the time series given by Λ maintains its chaotic behavior, it can be used to analyze the dynamics of the system. Let us compute

$$\frac{d}{dt} \nabla \cdot \mathbf{f} = \dot{\Lambda}, \quad (5)$$

$$= (\nabla \Lambda)^T \dot{\mathbf{x}}, \quad (6)$$

$$= (\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta), \quad (7)$$

with $\nabla \Lambda = \left[\frac{\partial \Lambda}{\partial x_1} \quad \frac{\partial \Lambda}{\partial x_2} \quad \dots \quad \frac{\partial \Lambda}{\partial x_n} \right]^T$ being the gradient. Using the expression in Eq. (7), the solution of Λ can be introduced on the right-hand side of system (3), yielding

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\Lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}, \beta) \\ (\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta) \end{bmatrix}. \quad (8)$$

We can consider an expanded system by introducing $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \Lambda \end{bmatrix}$ and

$$\mathbf{F}(\mathbf{y}, \beta) = \begin{bmatrix} \mathbf{f}(\mathbf{x}, \beta) \\ (\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta) \end{bmatrix}. \quad (9)$$

$$\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}, \beta), \quad (9)$$

where $\mathbf{F}(\mathbf{y}, \beta) \in \mathbb{R}^{n+1}$. Suppose now that the expanded system (9) has a numerical solution given by the Euler rule expressed by

$$\mathbf{y}_N = \mathbf{y}_{N-1} + h \sum_{i=0}^{N-1} \mathbf{F}(\mathbf{y}_{N-1}, \beta), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad (10)$$

where h is the integration step and $\mathbf{y}_0$ the initial state of the system. The next step is to compute a simultaneous solution of the expanded system using a $\epsilon \rightarrow 0$ that slightly changes the initial state by $\mathbf{z} = \mathbf{y}_0 + \epsilon$. So, we have

$$\mathbf{z}_N = \mathbf{z}_{N-1} + h \sum_{i=0}^{N-1} \mathbf{F}(\mathbf{z}_{N-1}, \beta), \quad \mathbf{z}(0) = \mathbf{z}_0 = \mathbf{y}_0 + \epsilon. \quad (11)$$

The solutions y_{n+1} and z_{n+1} represent the divergence of both systems in Eqs. (10) and (11). In a chaotic regimen, both solutions are expected to diverge over time. This is because the divergence is also sensitive to initial conditions. In fact, the solutions are identical in nonchaotic parametric regions while they separate in the chaotic windows.

Then, the proposed approach identifies chaotic dynamics by taking the absolute difference of average solutions y_{n+1} and z_{n+1} in period $[hN_i, hN_f]$ computed by

$$\mu_\beta = |\langle y_{n+1} \rangle - \langle z_{n+1} \rangle|, \quad (12)$$

$$= \left| \frac{\sum_{i=N_i}^{N_f} y_{n+1,i}}{N_f - N_i} - \frac{\sum_{i=N_i}^{N_f} z_{n+1,i}}{N_f - N_i} \right|, \quad (13)$$

$$= \frac{|\sum_{i=N_f}^{N_f} (y_{n+1,i} - z_{n+1,i})|}{N_f - N_i}, \quad (14)$$

where $y_{n+1,i}$ and $z_{n+1,i}$ are the i th iteration of the n th plus one solution (the solution for the divergence) from the expressions in Eqs. (10) and (11), respectively.

It is important to remark that $\langle y_{n+1} \rangle$ refers to the average of the derivative of the divergence for the first orbit (ADDFO), whereas $\langle z_{n+1} \rangle$ means the average of the derivative of the divergence for the second orbit (ADDFO). Finally, μ_β is the absolute error of the average of the derivative of a state-variable (AEADV). In the specific case of this investigation, the state variable denotes the derivative of divergence included as an extra variable in the expanded system (8).

A relevant aspect of the proposed approach is the simulation time required to identify chaotic dynamics is optimized. While most metrics need a significant amount of time to give a proper result, the proposed method only demands a short finite time period where the value for N_f is not much bigger than N_i . This time optimization is because the average for a bounded trajectory will approximate zero as time goes to infinity. To select the value for N_i , we must consider enough simulation time so that the orbits are separated from each other in the case of chaotic systems. The pseudo-code of the presented approach is given in Algorithm 1. This can be downloaded from repository [ABSOLUTE ERROR OF THE AVERAGE DIVERGENCE VARIABLE](https://github.com/pacheco008/AEADV).¹

Algorithm 1 Pseudo-code of the new method to identify chaotic dynamics in nonuniformly dynamical systems.

```

1: Input: Dynamical System  $\Phi(\mathbf{x}, \beta)$ ,  $\beta$ ,  $\mathbf{y}_0$ ,  $N_i$ ,  $N_f$ ,  $h$ ,  $\epsilon$ 
2: Output:  $\mu_\beta$ 
3: for  $k \leftarrow 1$  to  $\text{size}(\beta)$  do
4:    $\mathbf{y} = \mathbf{y}_0$ 
5:    $\mathbf{z} = \mathbf{z}_0 + \epsilon$ 
6:    $\text{avg} = 0$ 
7:   for  $i \leftarrow 1$  to  $N_i - 1$  do
8:      $\mathbf{y} = \mathbf{y} + hF(\mathbf{y}, \beta[k])$ 
9:      $\mathbf{z} = \mathbf{z} + hF(\mathbf{z}, \beta[k])$ 
10:  end for
11:  for  $i \leftarrow N_i$  to  $N_f$  do
12:     $\mathbf{y} = \mathbf{y} + hF(\mathbf{y}, \beta[k])$ 
13:     $\mathbf{z} = \mathbf{z} + hF(\mathbf{z}, \beta[k])$ 
14:     $\text{avg} = \text{avg} + y_j - z_j$ 
15:  end for
16:   $\mu_\beta = \ln \left( \frac{\|\text{avg}\|}{N_f - N_i} + 1 \right)$ 
17: end for
18: return  $\mu_\beta$ 

```

Stability analysis

An expansion of a dynamical system may have an impact on the original behavior of the system. We will prove in this section that the introduction of divergence as a variable has no influence on the stability of the Euler numerical integration method.

Theorem 1 (Eigenvalues of the Expanded System). *The eigenvalues of the Jacobian $\mathbf{J}_{\text{exp}}$ of expanded system described in Eq. (7) are:*

- The eigenvalues of the Jacobian $\mathbf{J}(\mathbf{x}^*) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ of original system.
- An extra eigenvalue $\lambda_{n+1} = 0$.

Proof. The Jacobian of the expanded system is found by

$$\mathbf{J}_{\text{exp}}(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial \mathbf{f}(\mathbf{x}^*, \beta)}{\partial \mathbf{x}} & \frac{\partial \mathbf{f}(\mathbf{x}^*, \beta)}{\partial \Lambda} \\ \frac{\partial}{\partial \mathbf{x}} \dot{\Lambda} \Big|_{\mathbf{x}=\mathbf{x}^*} & \frac{\partial}{\partial \Lambda} \dot{\Lambda} \Big|_{\mathbf{x}=\mathbf{x}^*} \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}, \quad (15)$$

where $\mathbf{x}^*$ is the equilibrium point of the system. From the original system in Eq. (3) we have $\mathbf{J}(\mathbf{x}^*) = \frac{\partial \mathbf{f}(\mathbf{x}^*, \beta)}{\partial \mathbf{x}}$. Moreover, the states of the system have no dependence on the divergence and, given the system is a nonuniformly dynamical system, the divergence has a dependence on the states, as such we can write

$$\frac{\partial \mathbf{f}(\mathbf{x}^*, \beta)}{\partial \Lambda} = \mathbf{0} \in \mathbb{R}^n, \quad \frac{\partial}{\partial \Lambda} \dot{\Lambda} = 0 \in \mathbb{R}. \quad (16)$$

Finding the gradient of the temporal derivative of the divergence yields

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \dot{\Lambda} &= \frac{\partial}{\partial \mathbf{x}} ((\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta)), \\ &= \mathbf{f}(\mathbf{x}, \beta)^T \frac{\partial}{\partial \mathbf{x}} (\nabla \Lambda) + \nabla \Lambda^T \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \beta), \end{aligned} \quad (17)$$

recalling that $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$, then evaluating in the equilibrium point we have

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \dot{\Lambda} \Big|_{\mathbf{x}=\mathbf{x}^*} &= \mathbf{f}(\mathbf{x}^*, \beta)^T \left(\frac{\partial}{\partial \mathbf{x}} (\nabla \Lambda) \right)_{\mathbf{x}=\mathbf{x}^*} + \left(\nabla \Lambda^T \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \beta) \right)_{\mathbf{x}=\mathbf{x}^*} \\ &= \left(\nabla \Lambda^T \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \beta) \right)_{\mathbf{x}=\mathbf{x}^*}. \end{aligned} \quad (18)$$

Then, the eigenvalues of the expanded Jacobian can be computed by

$$\det(\mathbf{J}_{\text{exp}}(\mathbf{x}^*) - \lambda I_{n+1}) = \det \left(\begin{bmatrix} \mathbf{J}(\mathbf{x}^*) - \lambda I_n & \mathbf{0} \\ \left(\nabla \Lambda^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right)_{\mathbf{x}=\mathbf{x}^*} & -\lambda \end{bmatrix} \right), \quad (19)$$

$$= \det(\mathbf{J}(\mathbf{x}^*) - \lambda I_n) (-\lambda), \quad (20)$$

where $I_n \in \mathbb{R}^{n \times n}$ and $I_{n+1} \in \mathbb{R}^{(n+1) \times (n+1)}$ are the identity matrix of n and $n+1$ dimension, respectively. Based on Eq. (20) we conclude that the eigenvalues of $\mathbf{J}_{\text{exp}} = \{\lambda_1, \lambda_2, \dots, \lambda_n, \lambda_{n+1}\}$, where $\lambda_{n+1} = 0$ is the result of the expansion of the system. $\square$

Theorem 2 (Euler Integration Stability). *The expansion of the system in Eq. (1) to the system in Eq. (7) has no impact on the stability condition of the Euler numerical integration method.*

Proof. We recall the condition of stability for the Euler numerical integration method is given by [34]

$$|1 + h\lambda_i| < 1, \quad \forall i = 1, 2, \dots, n, \quad (21)$$

meaning the stability of the Euler method depends on the eigenvalues of the system. Based on Eq. (20), the expanded system (7) inherits all eigenvalues of the original system (3) plus an additional eigenvalue $\lambda_{n+1} = 0$ associated with the divergence. This additional eigenvalue does not alter the stability condition because it is the result of an additional non-independent variable of the other states. Furthermore, it lies on the boundary of the stability region, causing no exponential growth (marginal stability). As a result, the stability criterion given by Eq. (21) remains identical for both the original and the expanded system. $\square$

Computational results

In the following subsections, we demonstrate the usefulness of the proposed metric. The four studied nonuniformly chaotic systems are (i) the Rossler system, (ii) a three-neuron Hopfield neural network, (iii) the conservative Sprott D system, and (iv) the Pernarowski biological system for modeling pancreatic beta-cells. All simulations are completed on a MATLAB environment using a laptop Acer i5-1135G7 2.40 GHz with 8 GB of RAM.

¹ <https://github.com/pacheco008/AEADV>.

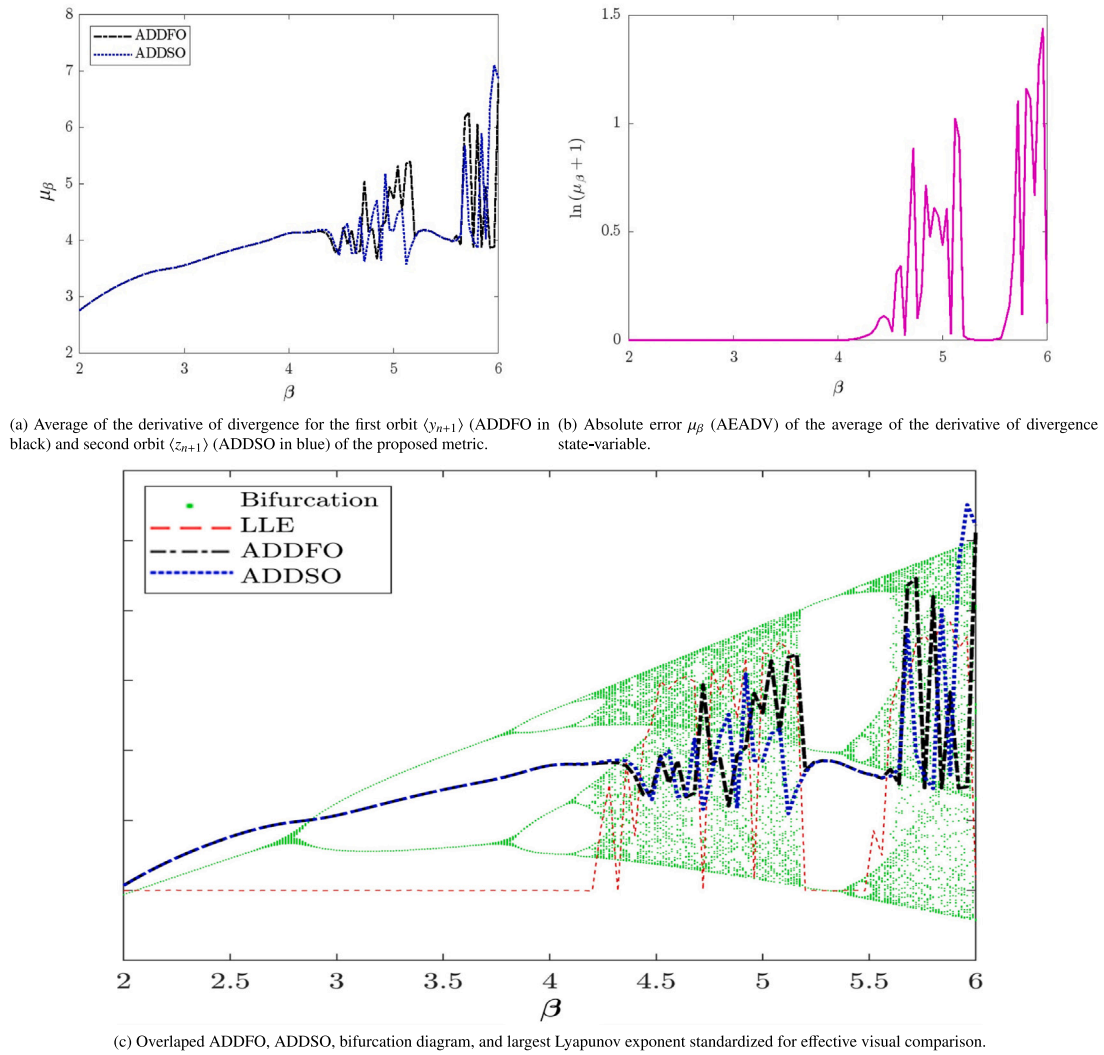


Fig. 1. Chaotic dynamics of the expanded Rossler system (27) using the proposed metric compared to bifurcation diagrams and largest Lyapunov exponent.

Nonuniformly dissipative Rossler system

One of the most well-known chaotic systems is the Rossler system described by [30]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -y - z \\ x + ay \\ b + z(x - \beta) \end{bmatrix}, \quad (22)$$

where the chaotic dynamics is present when $a = 0.2$, $b = 0.2$. In this case, β is the studied parameter which is varied in the range $4.2 \leq \beta \leq 50$. For $\beta = 5$, the Rossler system has two hyperbolic saddle equilibrium points of type II located in $(0.008, -0.04, 0.04)$ and $(4.992, -24.959, 24.959)$, respectively.

The Rossler system satisfies Definition 2 of a nonuniformly dissipative dynamical system with divergence Λ given by:

$$\Lambda = \nabla \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z}, \quad (23)$$

$$= 0.2 + x - \beta, \quad (24)$$

and after to apply Eq. (7), we have

$$(\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} -y - z \\ x + 0.2y \\ 0.2 + z(x - \beta) \end{bmatrix}, \quad (25)$$

$$= -y - z. \quad (26)$$

Then rewriting the Rossler system as in Eq. (8) yields:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\Lambda} \end{bmatrix} = \begin{bmatrix} -y - z \\ x + 0.2y \\ 0.2 + z(x - \beta) \\ -y - z \end{bmatrix}, \quad (27)$$

which is herein termed the expanded dynamical system where the additional state variable is the derivative of divergence.

Figs. 1(a) and 1(b) show the average of the derivative of divergence for the first orbit $\langle y_{n+1} \rangle$ (ADDFO) and the second orbit $\langle z_{n+1} \rangle$ (ADDSO), respectively, and the absolute error μ_β (AEADV). Choosing the initial conditions of the original states as $\mathbf{x}_0 = [0.1 \ 1.0 \ 1.0]^T$ we can compute the initial state of divergence variable as $\Lambda(0) = 0.2 + x(0) - \beta = 0.3 - \beta$. Also, we select $\epsilon = [0.001 \ 0.001 \ 0.001 \ 0.001]^T$, integration step $h = 0.001$, $N_i = 100000$, and $N_f = 105000$ where the parameter β is uniformly swept with 100 divisions for values between 2 and 6. As can be seen, ADDFO and ADDSO evolve almost identically when the parameter β is in the non-chaotic region. On the other hand, ADDFO and ADDSO split significantly if β belongs to the parametric region of chaotic dynamics. So, the proposed metric can distinguish chaotic and non-chaotic areas in the chosen parameter space. For instance, in the Rossler system, the chaotic oscillations appear for $\beta > 4.2$ and vanish for $5.19 \leq \beta \leq 5.49$.

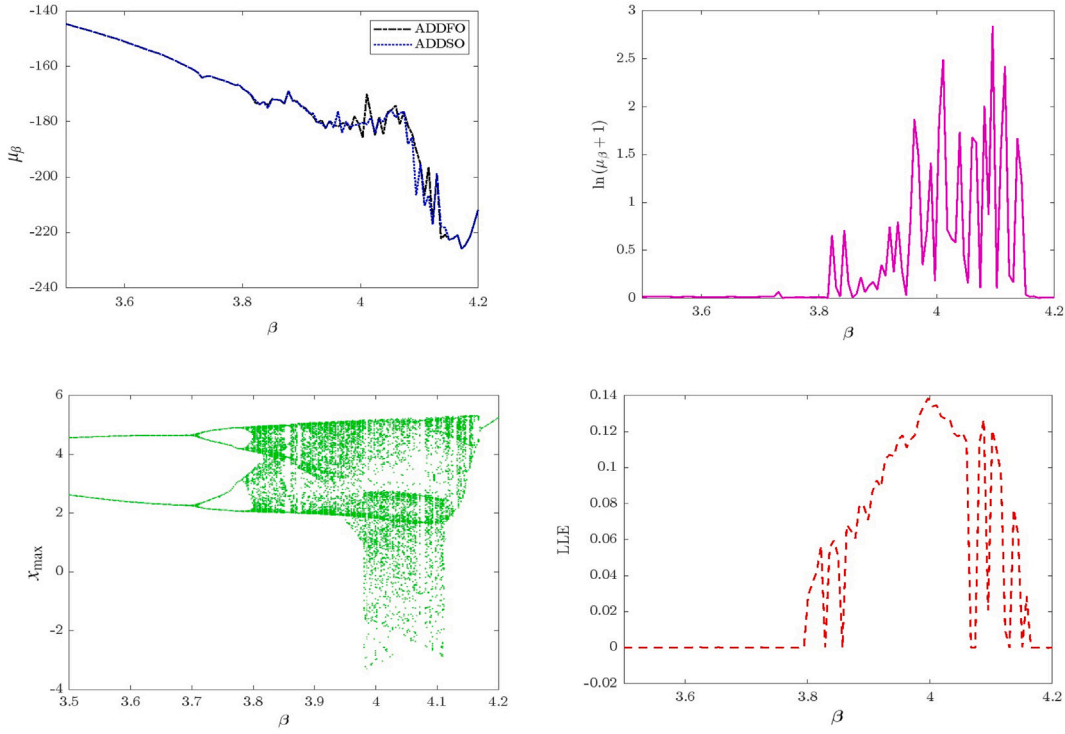


Fig. 2. Analysis of the chaotic dynamics in the nonuniformly conservative Sprott D system (39) using the proposed metric. (a) Average of the derivative of divergence for the first orbit $\langle y_{n+1} \rangle$ (ADDFO in black) and second orbit $\langle z_{n+1} \rangle$ (ADDSO in blue). (b) Absolute error μ_β (AEADV) of the average of the derivative of divergence. (c) Bifurcation diagram of local maxima of state variable x . (d) Largest Lyapunov exponent. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Elapsed time after transient response for each chaos metric of Fig. 1 for the nonuniformly dissipative Rossler system.

	Proposed metric	Bifurcation diagram	LLE
Time after transient (s)	5	400	1900

Transient time: 100 s.

To demonstrate the effectiveness of the proposed approach, we compare the given results with those of the bifurcation diagram and the largest Lyapunov exponent, as shown in Fig. 1(c). All metrics were standardized by normalizing their respective axes to arbitrary units, ensuring uniform scaling across figures for effective visual comparison. Such results reveal how the method based on ADDFO and ADDSO successfully classifies chaotic behavior comparable to classical chaos metrics but requires much less data after transient time, optimizing the elapsed simulation time (Table 1).

Nonuniformly dissipative Hopfield neural network

In literature, various artificial neural networks have reported chaotic behavior. More particularly, a Hopfield neural network is described by the following dimensionless dynamical system

$$\dot{\mathbf{x}} = -\mathbf{x} + W \Gamma(\mathbf{x}) + \mathbf{b}, \quad (28)$$

where $\mathbf{x} \in \mathbb{R}^n$ describes the behavior of each neuron, $W \in \mathbb{R}^{n \times n}$ is a synaptic weight matrix, $\mathbf{b} \in \mathbb{R}^n$ is an input bias and $\Gamma(\mathbf{x}) \in \mathbb{R}^n$ is known as the activation function for each neuron. The specific model considered in this contribution is described by [16]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = -\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 & 1.2 & -7 \\ 1.1 & 0 & 2.8 \\ 0.7 & -2 & \beta \end{bmatrix} \begin{bmatrix} \tanh x \\ \tanh y \\ \tanh z \end{bmatrix}. \quad (29)$$

Computing the divergence operator of the Hopfield neural network produces

$$\Lambda = -3 + \beta(1 - \tanh^2 z). \quad (30)$$

We can note that the divergence depends on the state variable z . Therefore, the Hopfield neural network (29) belongs to the classification of nonuniformly dissipative systems in Definition 2, with $\beta = 4$ the system has three equilibrium points in $(-3.4666, 0.5170, 0.6575)$, $(0, 0, 0)$ and $(3.4666, -0.5170, -0.6575)$.

Then, we can find the derivative of Λ as

$$\nabla \Lambda = \begin{bmatrix} 0 \\ 0 \\ -2\beta \tanh(z)(1 - \tanh^2(z)) \end{bmatrix}, \quad (31)$$

and following as in (7), we find

$$(\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta) = -2\beta \tanh(z)(1 - \tanh^2(z))\dot{z}. \quad (32)$$

As a result, the expanded system is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\Lambda} \end{bmatrix} = \begin{bmatrix} -x + 1.2 \tanh(y) - 7 \tanh(z) \\ -y + 1.1 \tanh(x) + 2.8 \tanh(z) \\ g_1(\mathbf{x}, \beta) \\ g_2(\mathbf{x}, \beta) \end{bmatrix}, \quad (33)$$

where the functions $g_1(\mathbf{x}, \beta)$ and $g_2(\mathbf{x}, \beta)$ are equivalent to

$$\begin{aligned} g_1(\mathbf{x}, \beta) &= -z + 0.7 \tanh(x) - 2 \tanh(y) + \beta \tanh(z), \\ g_2(\mathbf{x}, \beta) &= -2\beta \tanh(z)(1 - \tanh^2(z))g_1(\mathbf{x}, \beta). \end{aligned} \quad (34)$$

The expanded system (33) is now solved using Eqs. (10) and (11) with initial conditions for the original system $\mathbf{x}_0 = [0.1 \ 0 \ 0]^T$, initial divergence $\Lambda(0) = -3 + \beta$, $\epsilon = [0.001 \ 0.001 \ 0.001 \ 0.001]^T$, integration step $h = 0.001$, $N_i = 220000$, and $N_f = 240000$ for the parametric range $3.5 \leq \beta \leq 4.2$

The results for the ADDFO and ADDSO are shown in Fig. 3(a), while the AEADV is displayed in Fig. 3(b). Again, we evaluate the obtained

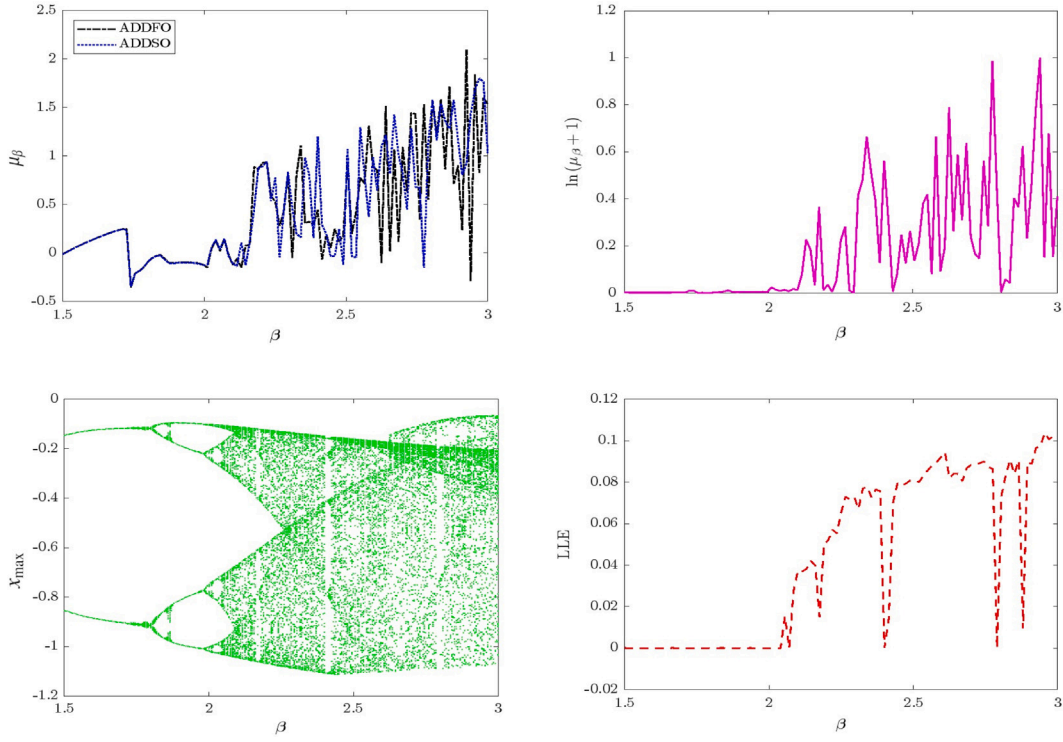


Fig. 3. Analysis of the chaotic dynamics in the nonuniformly dissipative Hopfield neural network (33) using the proposed metric. (a) Average of the derivative of divergence for the first orbit $\langle y_{n+1} \rangle$ (ADDFO in black) and second orbit $\langle z_{n+1} \rangle$ (ADDSO in blue). (b) Absolute error μ_β (AEADV) of the average of the derivative of divergence. (c) Bifurcation diagram of local maxima of state variable x . (d) Largest Lyapunov exponent. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

results with the proposed approach against the ones provided by the bifurcation diagram (Fig. 3(c)) and LLE (Fig. 3(d)). Note that the metric denotes chaotic dynamics for $\beta > 3.8$ and $\beta < 4.17$, which is the expected result based on LLE and the bifurcation diagram.

Nonuniformly conservative Sprott D system

The Sprott D dynamical system is one of the many proposed systems by Prof. J.C. Sprott in his contribution “Some simple chaotic flow” [35], and is described by the set of differential equations

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -y \\ x + z \\ xz + \beta y^2 \end{bmatrix}, \quad (35)$$

where β is the parameter under study in the interval $1.5 \leq \beta \leq 3.0$. Following the same steps as previous systems, we applied the divergence operator to find

$$\Lambda = x. \quad (36)$$

Then, it is clear that the vector field of the Sprott D system satisfies Definition 2 being a nonuniformly conservative dynamical system with only one nonhyperbolic equilibrium point located on the origin. So, the derivative of divergence is given by

$$(\nabla \Lambda)^T \mathbf{f}(\mathbf{x}, \beta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} -y \\ x + z \\ xz + \beta y^2 \end{bmatrix}, \quad (37)$$

$$= -y, \quad (38)$$

with the expanded system described by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\Lambda} \end{bmatrix} = \begin{bmatrix} -y \\ x + z \\ xz + \beta y^2 \\ -y \end{bmatrix}. \quad (39)$$

By computing a solution with initial conditions of the original system $\mathbf{x}_0 = [0.1 \ 0.1 \ 0.1]^T$, initial divergence $\Lambda(0) = x(0) = 0.1$, $\epsilon = [0.0001 \ 0.0001 \ 0.0001 \ 0.0001]^T$, $N_i = 400000$, $N_f = 409000$, and integration step $h = 0.00025$, we obtain the results for ADDFO and ADDSO shown in Fig. 2(a) and the AEADV in Fig. 2(b). Additionally, the bifurcation diagram and LLE are presented in Fig. 2(c) and Fig. 2(d), respectively. For the conservative Sprott D system, chaotic dynamics are expected to appear for $\beta > 2.04$. One can observe that the proposed approach identifies clearly chaotic and nonchaotic regions.

Nonuniformly dissipative Pernarowski biological model

Pancreatic beta-cells produce and secrete insulin to normalize the glucose level of the blood. Multiple models exist to study the behavior of a single beta-cell within the pancreatic islets, such as the Pernarowski mathematical model [36]. It is described by the set of differential equations

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{c} \end{bmatrix} = \begin{bmatrix} v \\ -\epsilon H(u, c) - vF(u) - G(u, c) \\ \epsilon H(u, c) \end{bmatrix}, \quad (40)$$

where the functions $H(u, c)$, $F(u)$ and $G(u, c)$ are defined as

$$\begin{aligned} H(u, c) &= \beta(u - \bar{u}) - c, \\ F(u) &= a[(u - \hat{u})^2 - \eta^2], \\ G(u, c) &= c + u^3 - 3(u + 1), \end{aligned} \quad (41)$$

where u is the membrane potential, v is the voltage-gated potassium channel, and c is the substance concentration that regulates the bursting electrical activity. The value for the system parameters are $a = -0.02$, $\beta = 2$, $\hat{u} = 0$, $\bar{u} = -1.5$ and $\epsilon = 0.178$. We select the parameter η to be analyzed in the range $0.5 \leq \eta \leq 3.0$. By applying our method, we find the divergence as

$$\Lambda = -F(u) + \epsilon \frac{\partial H(u, c)}{\partial c}, \quad (42)$$

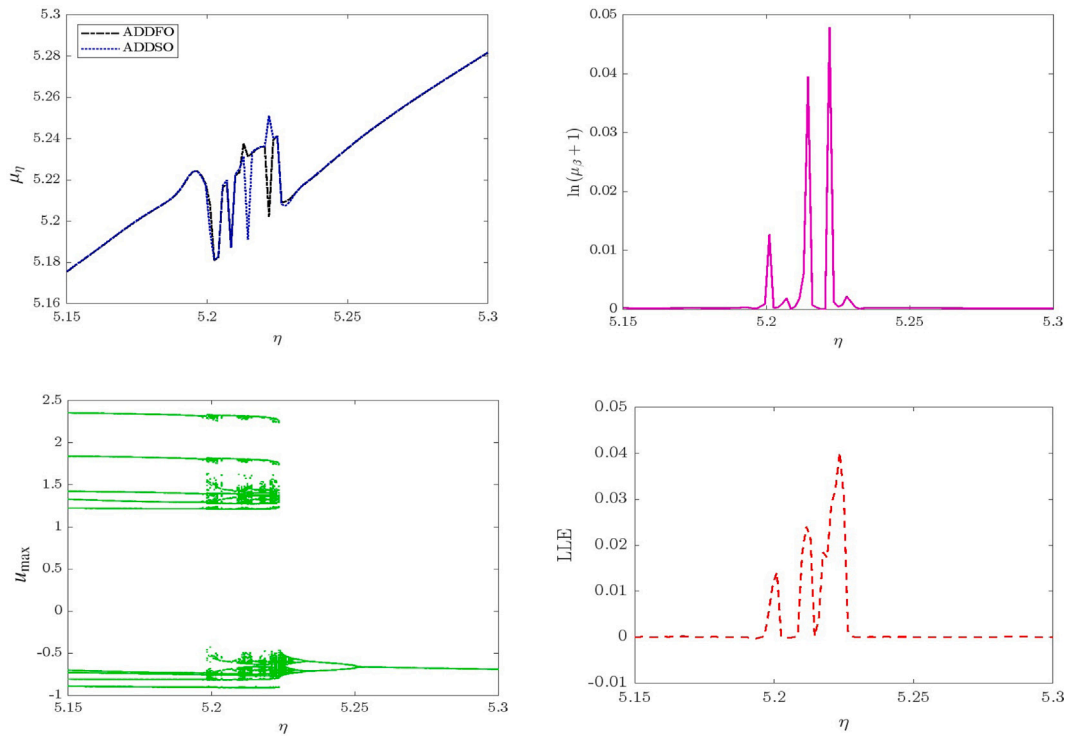


Fig. 4. Analysis of the chaotic dynamics in the nonuniformly dissipative Parnarowski model (46) using the proposed metric. (a) Average of the derivative of divergence for the first orbit $\langle v_{n+1} \rangle$ (ADDFO in black) and second orbit $\langle z_{n+1} \rangle$ (ADDSO in blue). (b) Absolute error μ_p (AEADV) of the average of the derivative of divergence. (c) Bifurcation diagram of local maxima of state variable u . (d) Largest Lyapunov exponent. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

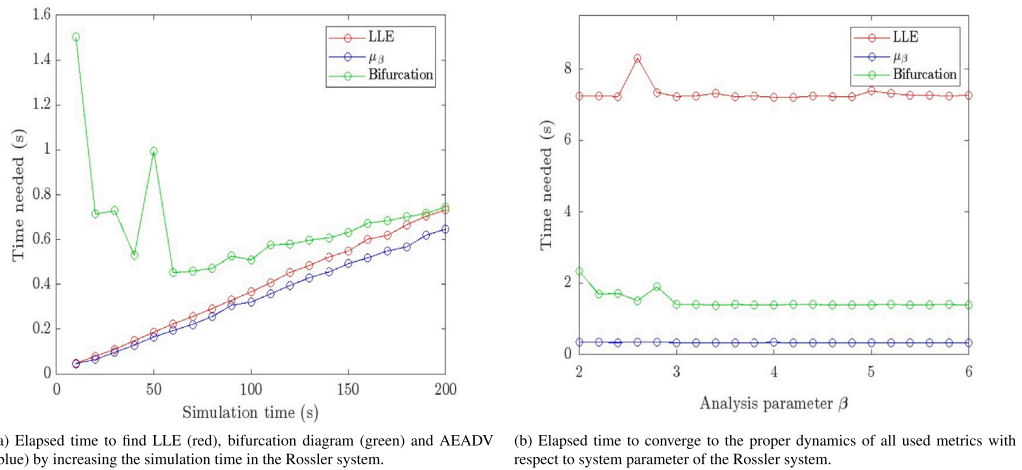


Fig. 5. Computational performance of the proposed metric Absolute Error of the Average Divergence Variable (AEADV). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$= 0.02[u^2 - \eta^2] - 0.178, \quad (43)$$

and following Eq. (7), we get

$$(\nabla A)^T \mathbf{f}(\mathbf{x}, \eta) = \begin{bmatrix} 0.02 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} v \\ \dot{v} \\ \epsilon H(u, c) \end{bmatrix}, \quad (44)$$

$$= 0.02v. \quad (45)$$

As a result, we obtain the following expanded system

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{c} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} v \\ -\epsilon H(u, c) - vF(u) - G(u, c) \\ \epsilon H(u, c) \\ 0.02v \end{bmatrix}. \quad (46)$$

Analogous to previous cases, the proposed approach can discover the nonchaotic and chaotic regions in the expanded Parnarowski biological model (46) as shown in Fig. 4. The analysis was conducted with initial conditions of the original system $\mathbf{x}_0 = [-1.5 \ 0.1 \ 1.5]^T$,

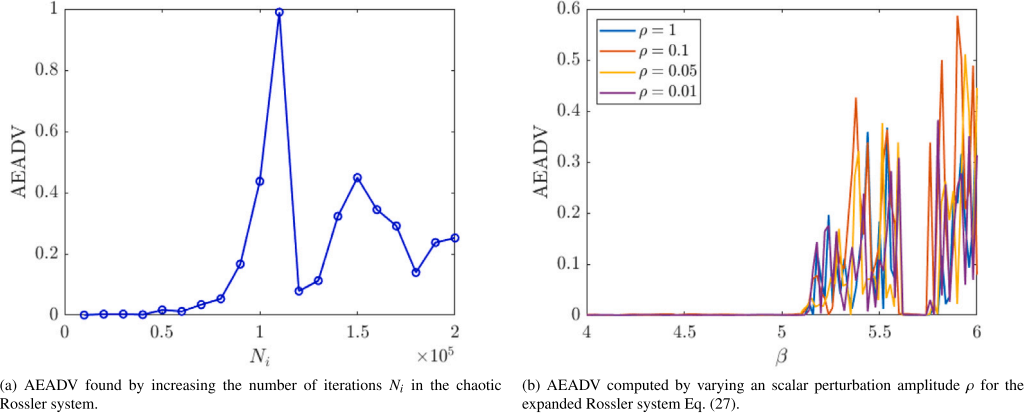


Fig. 6. (a) Transient time and (b) initial states effects on the AEADV metric.

resulting on an initial divergence $\Lambda(0) = 0.02(u(0)^2 - \eta^2) - 0.178 = \eta^2 - 0.133$, $\epsilon = [0.0002 \ 0.0002 \ 0.0002 \ 0.0002]^T$, integration step $h = 0.001$, $N_i = 120000$, and $N_f = 125000$. The results of the proposed approach agree well with the bifurcation diagrams and LLE, which demonstrate that AEADV can effectively identify chaos.

Discussion

In this section, we evaluate the computational cost, simulation time among the proposed algorithm and traditional methods. The pseudocode of the presented approach is given in Algorithm 1. This can be downloaded from <https://github.com/pacheco008/AEADV>. Also, the impact of different transient times as well as the phenomena of Critical Slowing Down is discussed.

Time response analysis

It is important to note that the first inner *for* cycle is the simulation time from $t = 0$ to $t = N_i$, i.e., the transient response. So, such a time interval is identical for the three considered metrics: proposed AEADV, LLE, and bifurcation diagram. The second inner *for* cycle integrates the expanded system from N_i to N_f , which is the simulation time considered for comparison purposes.

In this work, the LLEs are calculated with $\lambda = \langle \ln \epsilon_1 / \epsilon_0 \rangle / \Delta t$ [35], while the bifurcation diagrams are computed using the algorithm in [5]. In Fig. 5(a), the time required to compute the LLE (red), bifurcation diagrams (green), and proposed approach (blue) is plotted by varying the simulation time for each case. This shows that the algorithm based on the Absolute Error of the Average Divergence Variable (AEADV) is faster than the other two algorithms. For the particular case of the Rossler system, we find that the algorithm for LLE, bifurcation diagram, and proposed approach consumes 2000 s, 500 s, and 105 s, respectively, as given in Table 1. As a consequence of such results, we compute the computational effort needed to obtain the final result in the Rossler system with each metric as a function of the system parameter (see Fig. 5(b)). Note that the proposed metric requires less time to identify the chaotic and nonchaotic regions compared with classic metrics. Finally, the transient time should be removed to find appropriate dynamics. In the case of LLE, this may be optional, but it is not for the bifurcation diagrams and AEADV. In Table 1, we show the time required after the transient is about 100 s for the Rossler system.

Transient time and critical slowing down phenomena effects on the AEADV metric

An unquestionable feature of chaotic systems is their sensitive dependence on initial conditions; a phenomenon only observed after sufficient temporal evolution. To be found, such a phenomenon requires careful selection of the transient time, which is the interval required for the trajectories to diverge meaningfully. However, existing methodologies lack rigorous criteria for determining this critical parameter, introducing uncertainties in dynamical diagnostics such as bifurcation diagrams [37], Lyapunov exponent calculations [26], and the proposed AEADV metric. The consequences of an insufficient transient time are exemplified in Fig. 6(a), which illustrates the AEADV computed for the chaotic Rossler system (Eq. (27)) with $\beta = 5$ and parameters matching those in Section “Computational results”. Here, the metric fails to reliably distinguish chaotic dynamics until $N_i \geq 70000$, since transient trajectories require extended evolution to achieve measurable separation, where the AEADV value is 0.02 or ten times the initial separation ϵ . So, the challenge is a trade-off between a proper transient time selection with a robust chaos identification.

Furthermore, the Critical Slowing Down (CSD) [38] phenomenon also plays a relevant role in chaotic metrics, where the recovery time from small perturbations increases dramatically as the system approaches a critical threshold. In the AEADV metric, the CSD may occur while choosing initial states close to the equilibrium point of the system. In studies of chaotic systems, the CSD concepts may intersect with transient dynamics, provoking a complication in selecting an adequate transient time. Also, systems near bifurcation points may exhibit both CSD (slow recovery) and sensitive dependence on initial conditions (chaos), complicating metric-based analyses like Lyapunov exponents or the AEADV.

To investigate the critical slowing down (CSD) phenomenon, we computed the AEADV metric for the expanded Rössler system using perturbed initial conditions of the form: $\mathbf{y}_0 = \mathbf{y}^* + \rho [1, 1, 1, 0]^T$, where $\rho \in \mathbb{R}$ is a scalar perturbation amplitude affecting the equilibrium point $\mathbf{y}^*$. Fig. 6(b) shows the AEADV values for different values of ρ , demonstrating robust identification of chaotic regimes within the selected parameter space. Due to the proximity of initial conditions to the equilibrium point, a configuration amplifying CSD effects, we set an extended transient time for smaller values of ρ to guarantee trajectories diverged sufficiently before computing the proposed metric. In specific, for ρ values of 1, 0.1, 0.05, and 0.01, transient times of 250 s, 300 s, 400 s, and 600 s were selected, respectively.

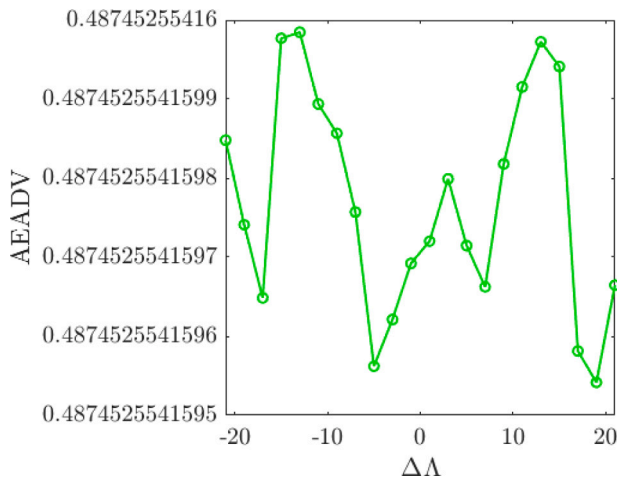


Fig. 7. Invariance on the AEADV value by introducing a perturbation $\Delta\Lambda$ in the initial state of the divergence in the Rossler system computed with $\Lambda(0) = -4.7 + \Delta\Lambda$.

Invariance under expanded variable initialization

On the other hand, the selection of initial conditions in dynamical systems can lead to other dynamic behaviors, such as basin boundaries and multistability, which also complicate chaos identification. In this work, Theorems 1 and 2 establish that the proposed system expansion preserves the dynamical invariants of the original system, including invariance to the initial states of the augmented variables. To validate such theoretical findings, we perform numerical experiments on the Rössler system with $\beta = 5$, perturbing the initial divergence variable $\Lambda(0)$ by $\Delta\Lambda \in [-21, 21]$, i.e., $\Lambda(0) = -4.7 + \Delta\Lambda$. As evidenced in Fig. 7, the AEADV metric exhibits negligible sensitivity to those perturbations, with a maximum deviation of $< 5 \times 10^{-13}$ for all tests. This demonstrates the robustness of the proposed method in terms of variations in the initial conditions.

Conclusion

The main goal of the proposed approach is to present a new algorithm to identify chaotic and nonchaotic parametric regions devoted to nonuniformly dissipative and conservative dynamical systems. The proposed metric is based on the fact that the divergence for such systems depends on state variables, and therefore, it is also sensitive to small changes to initial conditions. Then, by determining the derivative of the varying divergence, it can be added to the dynamical system as an extra state variable. As a result of this process, chaotic dynamics can be found using the absolute error of the average of divergence variable, termed herein AEADV. In nonchaotic zones, the divergence does not present sensitivity to initial conditions, so it converges to the same path, resulting in an absolute error around zero. On the other hand, when the divergence is sensitive to initial conditions, it leads to different orbits, resulting in an existing absolute error, which is used as an indicator of chaotic behavior. In addition, the proposed AEADV identifies the chaotic and nonchaotic parametric regions almost identical to the ones obtained with traditional metrics like the largest Lyapunov exponent and bifurcation diagrams but requires less computational effort. In particular, the proposed metric requires much less data and simulation time after transient time to converge to reliable results. Considering the Rossler systems, a Hopfield neural network, the Sprott-D system, and the Pearnarowski biological model as examples, we established that the AEADV method is suitable for identifying the chaotic dynamics in nonuniformly dissipative and conservative systems. In this manner, the proposed algorithm can increase the available portfolio of analysis software for nonlinear dynamics.

CRediT authorship contribution statement

Diego S. de la Vega: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization. **Olga G. Félix-Beltrán:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization. **Jesus M. Munoz-Pacheco:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization.

Funding

This research received no external funding.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author thanks CONAHCYT/Mexico for the support through the scholarship CVU 1301246. The authors acknowledge BUAP for the support via project VIEP 2025.

Data availability

The data are located in GitHub: <https://github.com/pacheco008/AEADV>.

References

- [1] Strogatz S. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. 2nd ed., first issued in hardback. A Chapman & Hall book, Boca Raton London New York: CRC Press; 2019.
- [2] Lorenz EN. Deterministic nonperiodic flow. J Atmos Sci 1963;20(2):130–41.
- [3] Devaney RL. An introduction to chaotic dynamical systems. Addison-wesley studies in nonlinearity, 2nd ed.. Redwood City, Calif: Addison-Wesley; 1989.
- [4] Wolf A, Swift JB, Swinney HL, Vastano JA. Determining Lyapunov exponents from a time series. Phys D: Nonlinear Phenom 1985;16(3):285–317. [http://dx.doi.org/10.1016/0167-2789\(85\)90011-9](http://dx.doi.org/10.1016/0167-2789(85)90011-9).
- [5] Moysis L, Lawnik M, Volos C. Density-colored bifurcation diagrams — A complementary tool for chaotic map analysis. Int J Bifurc Chaos 2023;33(15):2330036.
- [6] Shiozawa K, Tokuda IT. Estimating Kolmogorov–Sinai entropy from time series of high-dimensional complex systems. Phys Lett A 2024;510:129531.
- [7] Holmes P. Poincaré, celestial mechanics, dynamical-systems theory and “chaos”. Phys Rep 1990;193(3):137–63. [http://dx.doi.org/10.1016/0370-1573\(90\)90012-Q](http://dx.doi.org/10.1016/0370-1573(90)90012-Q).
- [8] Gottwald GA, Melbourne I. The 0-1 test for chaos: A review. Chaos Detect Predict 2016;221–47.
- [9] Skokos C. Chaos detection and predictability. New York, NY: Springer Berlin Heidelberg; 2016.
- [10] Hilborn RC. Chaos and nonlinear dynamics: an introduction for scientists and engineers. 2nd ed.. Oxford ; New York: Oxford University Press; 2000.
- [11] Venkatesh J, Pchelintsev AN, Karthikeyan A, Parastesh F, Jafari S. A fractional-order memristive two-neuron-based Hopfield neuron network: Dynamical analysis and application for image encryption. Mathematics 2023;11(21):4470.
- [12] Clemente-López D, Munoz-Pacheco JM, de Jesus Rangel-Magdaleno J. Experimental validation of IoT image encryption scheme based on a 5-D fractional hyperchaotic system and numba JIT compiler. Internet Things 2024;25:101116.
- [13] Yan S, Wang J, Li L. Analysis of a new three-dimensional jerk chaotic system with transient chaos and its adaptive backstepping synchronous control. Integration 2024;98:102210.
- [14] Pilatowsky-Cameo S, Dag CB, Ho WW, Choi S. Complete Hilbert-space ergodicity in quantum dynamics of generalized fibonacci drives. Phys Rev Lett 2023;131(25):250401.
- [15] ur Rahman M, Karaca Y, Sun M, Baleanu D, Alfuzan WF. Complex behaviors and various soliton profiles of (2+ 1)-dimensional complex modified Korteweg-de-Vries equation. Opt Quantum Electron 2024;56(5):878.
- [16] Rajagopal K, Munoz-Pacheco JM, Pham V-T, Hoang DV, Alsaadi FE, Alsaadi FE. A Hopfield neural network with multiple attractors and its FPGA design. Eur Phys J Spec Top 2018;227(7–9):811–20.

- [17] Lin H, Wang C, Yu F, Sun J, Du S, Deng Z, Deng Q. A review of chaotic systems based on memristive Hopfield neural networks. *Mathematics* 2023;11(6):1369.
- [18] Jones A, Strigul N. Is spread of COVID-19 a chaotic epidemic? *Chaos Solitons Fractals* 2021;142:110376.
- [19] Muñoz-Pacheco JM. Infinitely many hidden attractors in a new fractional-order chaotic system based on a fracmemristor. *Eur Phys J Spec Top* 2019;228(10):2185–96.
- [20] Tavazoei MS. Fractional order chaotic systems: history, achievements, applications, and future challenges. *Eur Phys J Spec Top* 2020;229:887–904.
- [21] Butusov DN, Pesterev DO, Tutueva AV, Kaplun DI, Nepomuceno EG. New technique to quantify chaotic dynamics based on differences between semi-implicit integration schemes. *Commun Nonlinear Sci Numer Simul* 2021;92:105467.
- [22] Liao H. Novel gradient calculation method for the largest Lyapunov exponent of chaotic systems. *Nonlinear Dynam* 2016;85(3):1377–92.
- [23] Jafari A, Hussain I, Nazarimehr F, Golpayegani SMRH, Jafari S. A simple guide for plotting a proper bifurcation diagram. *Int J Bifurc Chaos* 2021;31(01):2150011. <http://dx.doi.org/10.1142/S0218127421500115>.
- [24] Căliman A, Daquin J, Libert A-S. Improved detection of chaos with Lagrangian descriptors using differential algebra. *Phys D: Nonlinear Phenom* 2025;134506.
- [25] Zimper S, Ngapasare A, Hillebrand M, Katsanikas M, Wiggins SR, Skokos C. Performance of chaos diagnostics based on Lagrangian descriptors. Application to the 4D standard map. *Phys D: Nonlinear Phenom* 2023;453:133833.
- [26] Datseris G, Parlitz U. *Nonlinear dynamics: A concise introduction interlaced with code. Undergraduate lecture notes in physics*, Cham: Springer International Publishing; 2022. <http://dx.doi.org/10.1007/978-3-030-91032-7>, URL: <https://link.springer.com/10.1007/978-3-030-91032-7>.
- [27] Gan X, Wang H, Yuan R, Ao P. A new criterion beyond divergence for determining the dissipation of a system: Dissipative power. *Front Phys* 2021;9:695489.
- [28] Heidel J, Zhang F. Nonchaotic and chaotic behavior in three-dimensional quadratic systems: five-one conservative cases. *Int J Bifurc Chaos* 2007;17(06):2049–72.
- [29] Jafari S, Sprott JC, Dehghan S. Categories of conservative flows. *Int J Bifurc Chaos* 2019;29(02):1950021.
- [30] Rössler O. An equation for continuous chaos. *Phys Lett A* 1976;57(5):397–8.
- [31] Shinbrot T, Grebogi C, Wisdom J, Yorke JA. Chaos in a double pendulum. *Am J Phys* 1992;60(6):491–9. <http://dx.doi.org/10.1119/1.16860>.
- [32] Eshaghi S, Kadkhoda N, Inc M. Chaos control and synchronization of a new fractional laser chaotic system. *Qual Theory Dyn Syst* 2024;23(5):241.
- [33] Sprott J. A dynamical system with a strange attractor and invariant tori. *Phys Lett A* 2014;378(20):1361–3.
- [34] Butcher JC. *Numerical methods for ordinary differential equations. 2. Aufl ed.*. Chichester: Wiley; 2008, p. 74–9.
- [35] Sprott JC. Some simple chaotic flows. *Phys Rev E* 1994;50(2):R647–50.
- [36] Fallah H. Symmetric fold/super-Hopf bursting, chaos and mixed-mode oscillations in pernarowski model of pancreatic beta-cells. *Int J Bifurc Chaos* 2016;26(09):1630022.
- [37] Jafari A, Hussain I, Nazarimehr F, Golpayegani SMRH, Jafari S. A simple guide for plotting a proper bifurcation diagram. *Int J Bifurc Chaos* 2021;31(01):2150011. <http://dx.doi.org/10.1142/S0218127421500115>, URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218127421500115>.
- [38] Djurberg C, Svedlindh P, Nordblad P, Hansen MF, Bødker F, Mørup S. Dynamics of an interacting particle system: Evidence of critical slowing down. *Phys Rev Lett* 1997;79(25):5154–7. <http://dx.doi.org/10.1103/PhysRevLett.79.5154>, URL: <https://link.aps.org/doi/10.1103/PhysRevLett.79.5154>.