



INAOE

**Optimización de OTAs para mejorar el
diseño CMOS de una neurona LIF basada
en Memristor**

por

Carlos Alejandro Velázquez Morales

Tesis sometida como requisito parcial
para obtener el grado de

**MAESTRO EN CIENCIAS EN LA
ESPECIALIDAD DE ELECTRÓNICA**

en el

**INSTITUTO NACIONAL DE ASTROFÍSICA,
ÓPTICA Y ELECTRÓNICA**

Agosto 2025
Tonantzintla, Puebla

Supervisada por:

Dr. Luis Hernández Martínez
Investigador titular del INAOE

y

Dr. Esteban Tlelo Cuautle
Investigador titular del INAOE

©INAOE 2025

Derechos Reservados

El autor otorga al INAOE el permiso de
reproducir y distribuir copias de esta tesis en su
totalidad o en partes mencionando la fuente.



*A mi familia, amigos y mentores.
Esta tesis es un reflejo de todo lo que me han enseñado.*

AGRADECIMIENTOS

Ante todo, quiero expresar mi más profundo agradecimiento a mi madre, cuyo amor incondicional ha sido mi refugio y mi motor; a mi hermano, por su apoyo inquebrantable y su complicidad en cada etapa; y a mi novia, por ser mi compañera, mi alegría y mi fuerza en momentos difíciles. A mis abuelos y a toda mi familia, gracias por creer en mí incluso cuando las metas parecían lejanas, y por celebrar cada pequeño paso que me llevó a alcanzar este posgrado.

A mis asesores, cuya paciencia y sabiduría transformaron mis dudas en aprendizajes y mis ideas en resultados. Cada observación, cada corrección y cada palabra de aliento fueron fundamentales para culminar esta tesis. Igualmente, agradezco a todos mis profesores del INAOE por compartir no solo su conocimiento, sino también su pasión por la ciencia, moldeando así el profesional que soy hoy.

Al CONAHCYT México, por la beca que me permitió dedicarme de lleno a mis estudios y convertir este sueño en una realidad. Sin ese apoyo, este logro no habría sido posible.

Finalmente, quiero agradecer a mis amigos, compañeros y colegas por los momentos compartidos: las risas que aliviaron el estrés, las largas discusiones que enriquecieron mi perspectiva y la amistad que hizo de este camino una experiencia inolvidable. Este logro no es solo mío, sino de todos los que caminaron a mi lado.

RESUMEN

El diseño de sistemas neuromórficos en hardware ha surgido como una solución prometedora para superar las limitaciones de eficiencia energética y escalabilidad de las redes neuronales tradicionales implementadas en plataformas digitales. En particular, las redes neuronales de picos (SNNs, por sus siglas en inglés), que emulan el comportamiento neuronal biológico mediante impulsos eléctricos (*spikes*), ofrecen un enfoque más eficiente para el procesamiento de datos en tiempo real. Sin embargo, la implementación de SNNs en tecnologías de fabricación de circuitos integrados convencionales, como la complementaria de metal-óxido-semiconductor (CMOS), presenta desafíos significativos en términos de consumo energético, velocidad y densidad de integración. La presente Tesis propone el diseño de una neurona de picos en tecnología CMOS, utilizando un modelo memristivo, aprovechando su capacidad de almacenar y procesar información en la misma unidad física. El dispositivo memristivo se diseña usando amplificadores de transconductancia (OTAs), los cuales son optimizados aplicando la metodología gm/I_D , combinada con el algoritmo de optimización multiobjetivo NSGA-II, garantizando una región de transconductancia lineal con un valor deseado. El diseño incluye análisis de variaciones de proceso, voltaje y temperatura. Los resultados demuestran que la optimización del OTA mejora las características de un memristor, cuya aplicación se orienta al diseño CMOS de una neurona tipo LIF (*leaky integrate and fire*), lo cual permite implementar sinapsis programables de bajo consumo de potencia.

ÍNDICE GENERAL

Agradecimientos	IV
RESUMEN	v
1. INTRODUCCIÓN	1
1.1. Antecedentes	4
1.2. Justificación	6
1.3. Objetivos	7
1.3.1. Objetivo General	7
1.3.2. Objetivos Específicos	7
1.4. Contribuciones	8
1.5. Organización de la Tesis	9
2. Redes Neuronales de Picos	11
2.1. Fundamentos de las redes neuronales de picos	12
2.2. Modelos de neuronas de picos	13
2.3. Dinámica de las neuronas LIF	16

2.4. Conclusiones	18
3. El Memristor	19
3.1. Fundamentos del memristor	19
3.2. Memristores en neuronas tipo LIF	21
3.3. Memristor basado en OTA	24
3.4. Conclusiones	27
4. Optimización gm/I_D y NSGA-II	28
4.1. Diseño CMOS del OTA	28
4.2. Método de diseño gm/I_D y codificación entera	31
4.3. Algoritmo evolutivo NSGA-II	35
4.3.1. Algoritmo NSGA-II	36
4.3.2. Dimensionamiento de OTAs usando NSGA-II	39
4.4. Conclusiones	41
5. Simulaciones y Resultados	43
5.1. OTAs diseñados con primera iteración gm/I_D	43
5.1.1. Memristor basado en OTAs con gm/I_D	43
5.1.2. Neurona de picos basada en memristor con gm/I_D	45
5.2. OTAs optimizados con NSGA-II	46
5.2.1. Esquinas PVT de OTAs con NSGA-II	46

ÍNDICE GENERAL	VIII
5.2.2. Memristor basado en OTA con NSGA-II	48
5.2.3. Neurona de picos basada en memristor con NSGA-II	49
5.3. Conclusiones	55
6. Conclusiones Generales y Trabajo Futuro	56

ÍNDICE DE FIGURAS

1.1. Respuesta en el tiempo de las tres generaciones de redes neuronales artificiales.	3
2.1. Modelo de la neurona LIF: (a) Circuito esquemático, (b) Gráfica de corriente de entrada vs <i>spikes</i> de salida.	17
3.1. Elementos de circuito fundamentales de dos terminales: resistencia, capacitor, inductor y memristor [1].	20
3.2. Modelo de la neurona LIF basada en memristor volátil: (a) Circuito esquemático, (b) Gráfica de voltaje de entrada vs <i>spikes</i> de salida. . .	22
3.3. Neurona LIF basada en memristor dividido en cuatro bloques [2]: Espejo de corriente (verde). Integrador (rojo). Schmitt trigger (azul). Módulo de reset memristivo (morado).	23
3.4. Emulador de memristor aterrizado [2].	25
3.5. Análisis DC del inversor CMOS: (a) Transconductancia (g_m). (b) Voltaje de salida (V_B).	26
3.6. Curva característica del Memristor[2].	27
4.1. Topología <i>Current Mirror OTA</i> [3].	29

4.2. Topología <i>Folded Cascode OTA</i> [4].	30
4.3. Topología <i>Low Voltage Current Mirror OTA</i> [5].	30
4.4. Comparación entre la transconductancia ideal y real en el plano $I - V$ para una topología OTA CMOS.	31
4.5. Simulación de la característica $I - V$ de las tres topologías OTAs utilizando los tamaños de la primera iteración aplicando el método g_m/I_D : (a) <i>Current Mirror OTA</i> . (b) <i>Folded Cascode OTA</i> . (c) <i>Low Voltage Current Mirror OTA</i>	36
4.6. Enfoque propuesto para optimizar el dimensionamiento del circuito.	40
4.7. Resultados de dimensionamiento con NSGA-II para las topologías: (a) <i>Current Mirror OTA</i> , (b) <i>Folded Cascode OTA</i> , (c) <i>Low Voltage Current Mirror OTA</i>	42
5.1. Lazos de histéresis del emulador de memristor CMOS pre-diseñado con g_m/I_D para las topologías: (a) <i>Current Mirror OTA</i> . (b) <i>Folded Cascode OTA</i> . (c) <i>Low Voltage Current Mirror OTA</i>	44
5.2. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) pre-diseñada con g_m/I_D (señal de entrada: $T = 1\mu s$, azul) para las topologías ($C_{int} = 10pF$ y $C_m = 1000pF$): (a) <i>Current Mirror OTA</i> , (b) <i>Folded Cascode OTA</i> . (c) <i>Low Voltage Current Mirror OTA</i>	45
5.3. Esquinas PVT para las topologías: <i>Current Mirror OTA</i> : (a) Análisis AC y (b) Análisis DC. <i>Folded Cascode OTA</i> : (c) Análisis AC y (d) Análisis DC. <i>Low Voltage Current Mirror OTA</i> : (e) Análisis AC y (f) Análisis DC.	47

5.4. Lazos de histéresis del emulador de memristor CMOS optimizados con NSGA-II para las topologías: **(a)** *Current Mirror OTA*. **(b)** *Folded Cascode OTA*. **(c)** *Low Voltage Current Mirror OTA*. 48

5.5. Respuesta temporal (rojo) de la neurona de picos (Figura 3.3) con *Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 49

5.6. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Folded Cascode OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 50

5.7. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Low Voltage Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 51

5.8. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 52

5.9. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Folded Cascode OTA* para diferentes capacitancias (Señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 53

5.10. Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Low Voltage Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$ 54

ÍNDICE DE TABLAS

4.1. Especificaciones objetivo para el dimensionamiento de los OTAs CMOS que se muestran en las figuras 4.1, 4.2 y 4.3.	33
4.2. Variables de codificación de transistores para las topologías: (a) Current Mirror OTA. (b) Folded Cascode OTA. (c) Low Voltage Current Mirror OTA.	34
4.3. Primera iteración con el método g_m/I_D para dimensionar los OTAs. . .	35
4.4. Última generación con NSGA-II para dimensionamiento de OTAs. . .	41
5.1. Dimensiones de transistores para la neurona de picos basada en memristor de la figura 3.3.	44
5.2. Parámetros para el análisis PVT.	46

INTRODUCCIÓN

Durante las últimas décadas, las redes neuronales han experimentado una evolución notable, convirtiéndose en una de las herramientas más importantes de la inteligencia artificial (IA) y el aprendizaje automático. Tomando inspiración en las redes neuronales biológicas del cerebro —desde la perspectiva del procesamiento de información—, estas redes representan un modelo simplificado y se implementan mediante distintas arquitecturas según sus conexiones [6]. Las redes neuronales están compuestas por unidades básicas (neuronas artificiales) que emulan el comportamiento de sus análogas biológicas [7].

Su capacidad para modelar problemas complejos ha impulsado aplicaciones como el procesamiento de señales de comunicación realizado por Wu et al. en 2017 para estudios de contaminación atmosférica [8], el diagnóstico médico asistido por Dey et al. en 2022 para citologías [9] o la identificación de especies de *Aspergillus* mediante reconocimiento de imágenes por Tsang et al. en 2025 [10]. Sin embargo, la creciente demanda de eficiencia energética y velocidad de procesamiento —especialmente en dispositivos móviles— ha llevado a la necesidad de explorar nuevas arquitecturas que superen las limitaciones de las redes neuronales tradicionales [11]. La evolución de estos modelos puede clasificarse en tres generaciones computacionales, ilustradas en la figura 1.1, cada una con características distintivas [12]:

- **Redes Neuronales Digitales:** La primera generación se basan en el modelo de McCulloch-Pitts, que utiliza neuronas binarias (perceptrones o puertas de umbral). Son computacionalmente universales para operaciones booleanas e incluyen arquitecturas como las redes de Hopfield y las máquinas de Boltzmann. Sin embargo, su dependencia de valores discretos limita su capacidad para procesar datos continuos, lo que restringe su aplicación en problemas complejos. Además, carecen de mecanismos eficientes de aprendizaje para entornos analógicos, lo que las hace poco versátiles en contextos modernos.
- **Redes con Funciones de Activación Continua:** La segunda generación supera las limitaciones de las redes digitales mediante el uso de funciones de activación no lineales (p.ej., sigmoide, ReLU), permitiendo salidas en un rango continuo. Esto facilita el modelado de relaciones complejas entre datos y la aproximación de funciones continuas (teorema de aproximación universal). Arquitecturas como las Redes Neuronales Profundas (DNNs, por sus siglas en inglés) y las redes de funciones de base radial aprovechan algoritmos de aprendizaje basados en gradientes (p.ej., *backpropagation*), interpretando las salidas como tasas de disparo para emular parcialmente el comportamiento biológico. No obstante, su alto costo computacional y consumo energético —especialmente en modelos profundos— siguen siendo desafíos críticos.
- **Redes Neuronales de Picos:** Las redes neuronales de picos (SNNs, por sus siglas en inglés) representan a la tercera generación y un avance en la inteligencia bioinspirada, al procesar información mediante picos (*spikes*) que emulan el comportamiento de las neuronas biológicas. A diferencia de modelos anteriores, estas redes codifican datos en el momento exacto de cada activación neuronal, lo que permite un procesamiento más eficiente y biológicamente realista. Su arquitectura de activación por eventos (*event-driven*) las hace notablemente eficientes: solo consumen energía al generar los picos (*spikes*), a diferencia de las redes tradicionales que procesan continuamente. Esta ventaja las hace ideales para hardware neuromórfico diseñado para

operar con bajo consumo energético. Además, las SNNs sobresalen en el procesamiento de datos temporales gracias a su capacidad de operar en tiempo continuo. Esto las ha posicionado como herramientas importantes en robótica, interfaces cerebro-máquina y neurociencia computacional [13].

Como resultado de las demandas en la eficiencia, las limitaciones de las primeras generaciones han impulsado el desarrollo de las SNNs [11]. Estas redes, pertenecientes a la tercera generación, procesan información mediante picos en el tiempo (*spikes*), lo que reduce el consumo energético y permite un procesamiento más similar al biológico [13]. Su capacidad para operar en hardware neuromórfico las hace ideales para aplicaciones donde la eficiencia es crítica.

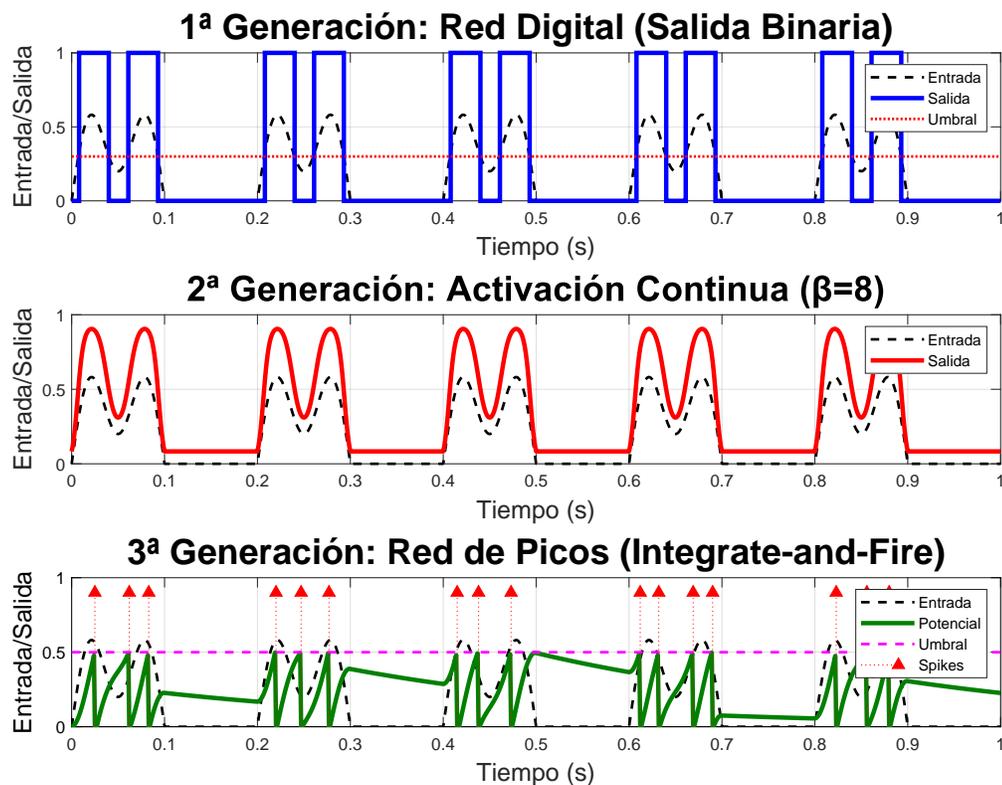


Figura 1.1: Respuesta en el tiempo de las tres generaciones de redes neuronales artificiales.

En paralelo a los avances en modelos de redes neuronales, los memristores han surgido como componentes esenciales gracias a su memoria no volátil.

Sus aplicaciones abarcan desde la implementación de sistemas caóticos [14] y el desarrollo de sistemas de comunicación seguros y confiables [15], hasta el almacenamiento multivalor en arquitecturas neuromórficas [16]. Estos dispositivos ofrecen una estructura de dos terminales con alta densidad de almacenamiento y bajo consumo energético [17].

El comportamiento no lineal de los memristores los hace adecuados para implementar matrices cruzadas —estructuras de circuitos basadas en una red de filas y columnas perpendiculares, donde cada intersección puede programarse de manera individual—, lo que no solo incrementa la velocidad de procesamiento, sino que además optimiza las operaciones de recuperación en redes SNNs [16]. Esta sinergia entre memristores y SNNs representa un avance significativo hacia sistemas de inteligencia artificial más eficientes, capaces de emular con mayor fidelidad los mecanismos biológicos de procesamiento de información [18].

Esta Tesis propone la optimización en el diseño de una topología de memristor basada en Amplificadores Operacionales de Transconductancia (OTAs, por sus siglas en inglés). Mediante algoritmos evolutivos combinados con el método gm/I_D , se optimizarán los anchos de canal de los transistores contenidos en los OTAs para lograr una transconductancia constante en una región lineal específica, utilizando tecnología CMOS 180nm TSMC. Este enfoque dual propone mejorar el rendimiento de las SNNs y avanzar en el desarrollo de hardware neuromórfico más eficiente y escalable.

1.1 ANTECEDENTES

La evolución de las redes neuronales artificiales se divide en tres etapas fundamentales de desarrollo. Inicialmente surgieron los modelos digitales básicos representados por los perceptrones, seguidos por las arquitecturas profundas con funciones de activación continua (DNNs), hasta llegar al paradigma actual de las

redes neuronales de picos [12]. Estas redes de tercera generación, inspiradas en los procesos biológicos, procesan la información mediante pulsos temporales (*spikes*), lo que les da ventajas significativas en eficiencia energética y capacidad para el procesamiento de señales temporales [13].

Un avance notable en este campo ha sido la incorporación de memristores en las arquitecturas SNN, aprovechando sus propiedades únicas como la memoria no volátil [19], y su comportamiento análogo al de las sinapsis neuronales [17]. Si bien estos dispositivos han demostrado potencial para acelerar las operaciones en SNNs [16], los memristores comerciales disponibles presentan limitaciones importantes en cuanto a precisión y estabilidad operacional.

En la implementación circuital, los memristores suelen emularse mediante configuraciones electrónicas que combinan dispositivos activos como VDTAs [20], CFTAs [21], CCIIIs [22], CCTAs [23], DVCCTAs [24], VDIBAs [25], OTAs [26], CDTAs [27], VDBAs [28], VDCCs [29] y CFOAs [30] con elementos pasivos tradicionales como resistores y capacitores. Estas configuraciones permiten replicar fielmente el comportamiento no lineal característico de los memristores, incluyendo sus distintivos lazos de histéresis pinchados en el origen cuando se analizan en el plano corriente-voltaje ($I - V$).

La presente investigación tiene fundamento en el diseño del emulador de memristor aterrizado propuesto en [2], cuya arquitectura integra un amplificador de transconductancia (OTA) junto con un inversor CMOS y un capacitor. El enfoque principal del trabajo consiste en optimizar la linealidad de la transconductancia de tres topologías de OTAs para emular el comportamiento del memristor. Para alcanzar este objetivo, se ha desarrollado una metodología que combina los principios del método de diseño gm/I_D [31] —que permite el dimensionamiento de transistores basado exclusivamente en sus puntos de polarización [32]— con la potencia del algoritmo evolutivo NSGA-II [33], reconocido por su eficiencia en la resolución de problemas complejos de optimización multiobjetivo en el ámbito del diseño de circuitos integrados analógicos [34].

1.2 JUSTIFICACIÓN

El rápido desarrollo de la inteligencia artificial (IA) ha revelado las limitaciones fundamentales de los paradigmas computacionales tradicionales, particularmente en lo referente a eficiencia energética y capacidad de procesamiento temporal. En este contexto, las redes neuronales de picos emergen como un enfoque revolucionario que emula con mayor fidelidad los mecanismos de procesamiento de información del cerebro humano. Sin embargo, su implementación práctica se ve significativamente limitada por las restricciones del hardware convencional, lo que demanda soluciones innovadoras para cerrar esta brecha tecnológica.

Esta investigación aborda la necesidad de desarrollar sistemas neuromórficos energéticamente eficientes y más similares al comportamiento biológico, centrándose en el uso del memristor como elemento clave para emular dinámicas neuronales. No obstante, la implementación práctica de estos dispositivos enfrenta desafíos considerables, incluyendo la variabilidad en los procesos de fabricación y la inherente no linealidad de su respuesta, problemas que requieren enfoques de diseño novedosos. Para superar estos obstáculos, el trabajo propone una estrategia de optimización de amplificadores de transconductancia (OTAs) que combina el método gm/I_D con algoritmos evolutivos, integrando técnicas de diseño analógico de precisión con métodos de optimización automática.

Los resultados esperados podrían conducir a una reducción sustancial del consumo energético en sistemas neuromórficos, una mejora significativa en el procesamiento de señales biológicas complejas, y el establecimiento de metodologías de diseño reproducibles. Estas contribuciones tendrían aplicaciones directas en el desarrollo de interfaces cerebro-máquina más eficientes y en la creación de plataformas autónomas con capacidades avanzadas de aprendizaje.

Desde la perspectiva metodológica, la investigación aporta un marco integrado para el co-diseño de OTAs y memristores optimizados para SNNs, validado expe-

rimentalmente en tecnología CMOS 180nm TSMC. Su relevancia social se hace evidente al considerar la creciente demanda global de soluciones de IA sostenible, democratizando el acceso a tecnologías neuromórficas avanzadas con aplicaciones prometedoras en áreas como la salud, la robótica autónoma y la computación.

La originalidad fundamental de este trabajo reside en su enfoque holístico, sintetizando conocimientos de neurociencia computacional, diseño analógico de precisión y técnicas de optimización automática. Esta convergencia interdisciplinaria no solo permite abordar problemas complejos que tienen dificultades en los enfoques tradicionales, sino también establecer un puente sólido entre los avances teóricos en redes neuronales y sus aplicaciones prácticas en hardware especializado, contribuyendo a cerrar la brecha entre la investigación fundamental y las soluciones tecnológicas concretas.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar y optimizar una neurona de picos tipo LIF basada en memristor en tecnología CMOS 180nm TSMC, mediante una metodología híbrida que integra el método gm/I_D con algoritmos evolutivos (NSGA-II), evaluando su desempeño mediante simulación Ngspice y Cadence-Virtuoso para garantizar su compatibilidad con arquitecturas neuromórficas.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Analizar y clasificar los modelos de redes neuronales de picos.
2. Modelar matemáticamente la dinámica de una neurona LIF con memristores, incorporando las no linealidades del dispositivo y las restricciones de la tec-

nología CMOS 180nm TSMC.

3. Implementar tres topologías circuitales de OTAs para emular el comportamiento memristivo, considerando parámetros clave como transconductancia (gm), ancho de banda y ganancia.
4. Adaptar el método gm/I_D para el dimensionamiento inicial de transistores en las topologías de OTAs, definiendo rangos óptimos de polarización.
5. Integrar el algoritmo evolutivo NSGA-II para optimización multiobjetivo de los parámetros clave de los circuitos OTAs.
6. Simular los OTAs optimizados en Ngspice y Cadence-Virtuoso, verificando la linealidad en la transconductancia (gm) y la robustez ante variaciones de proceso, voltaje y temperatura.
7. Simular los memristores y neuronas optimizadas en Cadence-Virtuoso, verificando los lazos de histéresis en el plano $I - V$ y la respuesta temporal a estímulos más similares a los biológicos.
8. Comparar la neurona memristiva optimizada con el dispositivo sin optimizar.

1.4 CONTRIBUCIONES

- Metodología híbrida del método gm/I_D con algoritmos evolutivos (NSGA-II) para optimizar la linealidad en la transconductancia de OTAs, reduciendo el tiempo de diseño manual y mejora la eficiencia energética frente a métodos unimodales, demostrado en tres topologías.
- Modelo matemático de neurona LIF memristiva que incorpora no linealidades y restricciones de fabricación en tecnología CMOS 180nm TSMC, validado mediante correlación MATLAB - Ngspice - Cadence-Virtuoso.

- Integración de los dispositivos optimizados en una neurona de picos, mediante simulaciones y prototipos de hardware.

1.5 ORGANIZACIÓN DE LA TESIS

Este trabajo de tesis se organiza de la siguiente manera.

En el capítulo 2 se describen las redes neuronales de picos (SNNs), se presentan modelos computacionales que permiten reproducir el mecanismo de integración y disparo, que implica la integración de corrientes sinápticas y la generación de potenciales de acción (*spikes*), característico de éstas neuronas. Se detallan sus ecuaciones diferenciales, destacando la neurona tipo LIF (*leaky integrate and fire*), debido a su simplicidad y su eficiencia reproduciendo aspectos del comportamiento neuronal básico.

En el capítulo 3 se describe al memristor y su aplicación en las SNNs. Se describen dos topologías de SNNs usando memristores para la generación de los picos y sus respectivas ecuaciones. Se propone el diseño de un emulador de memristor basado en un amplificador operacional de transconductancia (OTA) del estado del arte, y se asegura su dinámica.

En el capítulo 4 se realiza la optimización de tres topologías de OTA para implementarlas en sistemas complejos, como el memristor descrito en el capítulo 3. Se codifican los valores de los anchos del canal en los transistores que conforman al OTA, y se implementa una combinación del método gm/I_D con el algoritmo evolutivo NSGA-II (Non-dominated Sorting Genetic Algorithm II), optimizando el valor de transconductancia del OTA (G_m), la ganancia en lazo abierto (A_v) y el ancho de banda (BW).

En el capítulo 5 se obtienen simulaciones y resultados del comportamiento dinámico de los dispositivos OTA optimizados con la combinación del método gm/I_D

y el algoritmo evolutivo NSGA-II. Se observa la comparación entre los valores optimizados y los obtenidos únicamente con el método gm/I_D . Finalmente se hacen análisis PVT a los OTA optimizados y se verifica su comportamiento dinámico con el memristor del capítulo 3 y la neurona LIF del capítulo 2.

En el capítulo 6 se describen las conclusiones generales y trabajos futuros.

CAPÍTULO 2

REDES NEURONALES DE PICOS

Las redes neuronales de picos (SNNs) representan un paradigma de computación neuromórfica, constituyendo la tercera generación de redes neuronales artificiales [12]. A diferencia de los modelos tradicionales basados en activaciones continuas, las SNNs replican con fidelidad los mecanismos de procesamiento de información del sistema nervioso biológico mediante el uso de pulsos temporales (*spikes*) [13]. Esta aproximación se fundamenta en tres principios neurofisiológicos clave: la codificación temporal de la información, el umbral de activación dinámico y la plasticidad dependiente del tiempo de disparo (Spike-Timing-Dependent Plasticity, STDP) [35].

El mecanismo operativo de las SNNs se basa estrictamente en eventos, donde cada neurona artificial mantiene un potencial de membrana que se modifica ante las entradas sinápticas. Solo cuando este potencial supera un umbral crítico se genera un *spike* eléctrico, análogo a los potenciales de acción biológicos [36]. Esta característica permite una eficiencia energética superior, lo que las convierte en una herramienta prometedora para aplicaciones en neurociencia, robótica y sistemas embebidos [37].

2.1 FUNDAMENTOS DE LAS REDES NEURONALES DE PICOS

El principio operativo de las SNNs se basa en la propagación de información mediante pulsos (*spikes*), cuya generación está determinada por el potencial de membrana de cada neurona artificial. Estos *spikes* representan una analogía computacional de los potenciales de acción en neuronas biológicas, dotando a las SNNs de una mayor eficiencia energética y una similitud más biológica en comparación con otros modelos de redes neuronales [37].

El mecanismo de integración y disparo que caracteriza a las SNNs sigue un proceso secuencial bien definido. En la fase de integración, la neurona recibe entradas sinápticas en forma de *spikes* procedentes de otras neuronas conectadas, lo que produce una acumulación gradual del potencial de membrana. Cuando este potencial supera un umbral específico (*threshold*), se activa la fase de disparo, donde la neurona genera su propio *spike* que se propaga a través de sus conexiones sinápticas. Posteriormente, el sistema entra en una fase de reinicio donde el potencial de membrana retorna a su valor de reposo, seguido de un período refractario durante el cual la neurona permanece inactiva, imitando el comportamiento observado en sistemas neuronales biológicos [36].

Desde el punto de vista matemático, las SNNs se representan mediante ecuaciones diferenciales ordinarias no lineales que capturan la dinámica temporal del potencial de membrana neuronal. Estas ecuaciones describen con precisión cómo evoluciona el estado interno de cada neurona en función de las entradas sinápticas recibidas de otras neuronas, las constantes de tiempo características del sistema y los mecanismos de adaptación celular [13]. La naturaleza no lineal de estas ecuaciones permite que las SNNs exhiban comportamientos que incluyen oscilaciones autosostenidas, fenómenos de sincronización a escala de red neuronal, respuestas resonantes dependientes de la frecuencia de estimulación, y transiciones críticas

entre diferentes modos operativos [36].

El procesamiento de información en las SNNs operan bajo principios diferentes a los de las redes neuronales tradicionales, basándose en esquemas de codificación temporal donde la información se representa mediante patrones de disparo neural. Estos patrones incluyen relaciones temporales entre *spikes* individuales, correlaciones estadísticas en secuencias de activación neuronal, y relaciones de fase específicas en oscilaciones poblacionales. Este paradigma de codificación contrasta con los modelos clásicos basados en tasa de disparo, donde la información se reduce a simples promedios de frecuencia en ventanas temporales fijas [37].

2.2 MODELOS DE NEURONAS DE PICOS

Los modelos computacionales de neuronas de picos capturan la dinámica esencial del potencial de membrana en neuronas biológicas mediante formulaciones matemáticas de complejidad variable. Estos modelos simulan dos procesos fundamentales: la integración de corrientes sinápticas y la generación de potenciales de acción (*spikes*). A continuación se presentan los principales enfoques de modelado:

- **Modelo de Integración y Disparo (IF, por sus siglas en inglés):** El modelo IF, desarrollado originalmente por Louis Lapicque en 1907, representa la aproximación más básica de la dinámica neuronal. En este esquema, la membrana neuronal se modela con un simple capacitor, donde el potencial $V_m(t)$ se acumula en respuesta a corrientes de entrada $I_m(t)$ hasta superar un umbral fijo V_{th} momento en el cual se genera un *spike* y el potencial se reinicia [37]. La dinámica se describe mediante:

$$\frac{dV_m(t)}{dt} = \frac{I_m(t)}{C_m} \quad (2.1)$$

donde C_m representa la capacitancia de membrana. A pesar de su simplicidad, este modelo captura la esencia del proceso de integración de señales

neuronales.

■ **Modelo de Integración y Disparo con Fugas (LIF, por sus siglas en inglés):**

El modelo LIF extiende el modelo básico IF incorporando un mecanismo de fuga pasiva que simula la tendencia natural de la membrana a regresar a su potencial de reposo u_{rest} [37]. Esta mejora se implementa mediante una resistencia de fuga R_{leak} en paralelo con la capacitancia, dando lugar a la ecuación:

$$\tau_m \frac{dV_m(t)}{dt} = -(V_m(t) - u_{rest}) + R_{leak} I_m(t) \quad (2.2)$$

donde $\tau_m = R_{leak} C_m$ es la constante de tiempo característica del sistema. El modelo LIF, aunque sencillo, reproduce satisfactoriamente muchos aspectos del comportamiento neuronal básico.

- **Modelo Adaptativo Exponencial de Integración y Disparo (AdEx):** Como extensión más sofisticada, el modelo AdEx incorpora dos mejoras fundamentales al esquema LIF: un término exponencial que modela la activación rápida de los canales de sodio durante el disparo, y una variable de adaptación que captura los efectos de los canales iónicos dependientes de voltaje [36]. El sistema de ecuaciones resultante es:

$$\tau_m \frac{dV_m(t)}{dt} = -(V_m(t) - u_{rest}) + \Delta_T \exp\left(\frac{V_m(t) - u_T}{\Delta_T}\right) + R_{leak} (I_m(t) - \omega(t)) \quad (2.3)$$

$$\tau_w \frac{d\omega(t)}{dt} = a (V_m(t) - u_{rest}) - \omega(t) \quad (2.4)$$

donde $\omega(t)$ representa la corriente de adaptación, u_T es el potencial de umbral efectivo, y Δ_T controla la pendiente de la activación exponencial. Este modelo puede reproducir una amplia variedad de patrones de disparo observados experimentalmente, manteniendo una complejidad computacional manejable.

Los modelos básicos de neuronas de picos (IF, LIF, AdEx) capturan los aspectos esenciales de la generación de potenciales de acción, pero existen aproximaciones más sofisticadas que describen con mayor detalle la dinámica interna neuronal. Estos modelos avanzados incorporan mecanismos iónicos específicos y

pueden reproducir comportamientos complejos como oscilaciones, ráfagas de *spikes* (bursting) y transiciones entre modos dinámicos. Algunos de estos modelos son descritos brevemente:

- **Modelo de Hodgkin-Huxley (HH):** El modelo HH, desarrollado en 1952 a partir de experimentos con axones gigantes de calamar, estableció los fundamentos matemáticos para entender la generación de potenciales de acción [13]. Este modelo describe minuciosamente la dinámica de los canales iónicos de sodio (Na^+) y potasio (K^+) mediante un sistema de ecuaciones diferenciales no lineales:

$$C_m \frac{dV_m(t)}{dt} = I_{ion}(t) + I_m(t) \quad (2.5)$$

$$I_{ion}(t) = G_K n^4 (V_m(t) - u_K) + G_{Na} m^3 h (V_m(t) - u_{Na}) + G_{leak} (V_m(t) - u_{rest}) \quad (2.6)$$

donde n , m y h representan las probabilidades de activación/inactivación de los canales iónicos, G_K , G_{Na} y G_{leak} son las conductancias máximas; u_K , u_{Na} y u_{leak} son los potenciales de reversión. Aunque computacionalmente costoso, el modelo HH sigue siendo referencia en neurociencia computacional para estudios que requieren alta fidelidad biológica.

- **Modelo de Izhikevich:** El modelo Izhikevich combina la eficiencia de los modelos de integración y disparo con la capacidad de reproducir diversos patrones de disparo observados experimentalmente del modelo HH. La simplicidad del modelo, junto con su capacidad para emular comportamientos complejos como disparos regulares y ráfagas, lo hace ideal para simulaciones a gran escala [13].
- **Modelo Reducidos:** Para aplicaciones que requieren menos detalle iónico pero mayor eficiencia, existen varias aproximaciones [36]:
 - **FitzHugh-Nagumo (FHN):** El modelo FHN es una simplificación del modelo HH, con solo dos variables que captura la excitabilidad básica y las oscilaciones con menos complejidad, lo que lo hace útil para estudios de dinámicas neuronales básicas.

- **Hindmarsh-Rose (HR):** El modelo HR extiende el modelo FHN que modela comportamientos más complejos, como las ráfagas de picos (*bursting*). Este modelo es útil para estudiar dinámicas neuronales avanzadas.
- **Morris-Lecar (ML):** El modelo ML, propuesto en 1981, describe la dinámica de células excitables, como las neuronas o células musculares. Es una aproximación de segundo orden del modelo HH. El modelo esencialmente describe el voltaje de membrana y la variable de recuperación que da la probabilidad de que el canal de potasio conduzca.

Estos modelos ofrecen diferentes compromisos entre precisión biológica y eficiencia computacional, permitiendo seleccionar el nivel de detalle apropiado para cada aplicación en SNNs.

2.3 DINÁMICA DE LAS NEURONAS LIF

El modelo de neurona de integración y disparo con fugas (LIF) es una de las aproximaciones más utilizadas en neurociencia computacional y hardware neuromórfico debido a su simplicidad y capacidad para emular comportamientos neuronales biológicos [38]. Este modelo es computacionalmente eficiente y se implementa fácilmente en circuitos analógicos y digitales, lo que lo hace ideal para sistemas de SNNs, sin incurrir en la complejidad computacional de modelos más detallados, como Hodgkin-Huxley [13].

La neurona LIF puede modelarse mediante un circuito eléctrico simple compuesto por un capacitor de membrana (C_m) y una resistencia de fuga (R_{leak}), como se describe en la ecuación (2.2). El capacitor acumula carga en respuesta a la corriente de entrada ($I_m(t)$), mientras que la resistencia representa la conductancia pasiva de la membrana, responsable de la fuga de iones en reposo. Cuando las corrientes sinápticas excitan la neurona, el potencial de membrana $V_m(t)$ aumenta gradualmente. Si este potencial alcanza un umbral predeterminado V_{th} , la neurona

genera un pico y se reinicia inmediatamente a un potencial de reposo, simulando el comportamiento de una neurona biológica después de un potencial de acción [39].

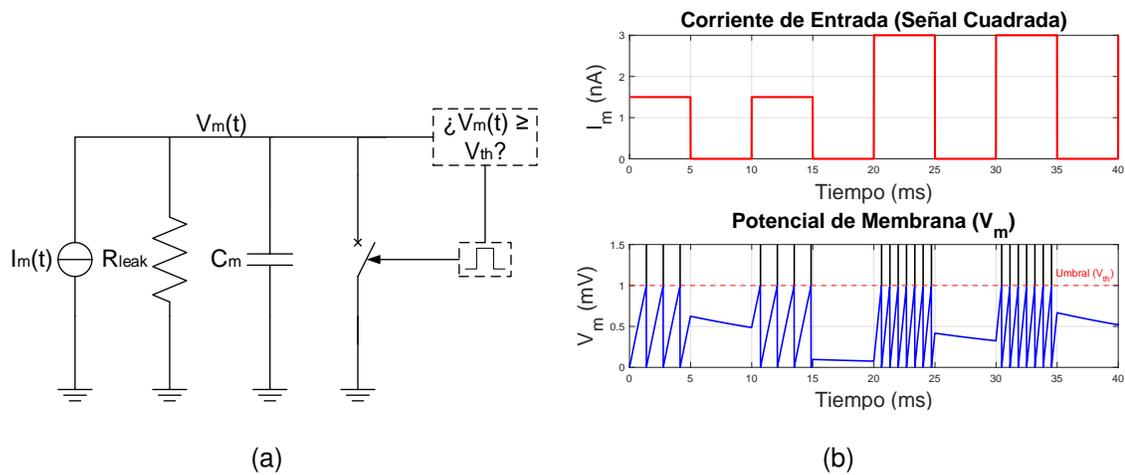


Figura 2.1: Modelo de la neurona LIF: (a) Circuito esquemático, (b) Gráfica de corriente de entrada vs *spikes* de salida.

El proceso de disparo ocurre cuando $V_m(t)$ alcanza o supera V_{th} , momento en el cual la neurona emite un pico y se reinicia. En la implementación circuital de la figura 2.1(a), este reinicio se logra mediante un interruptor controlado por voltaje que descarga rápidamente el capacitor. Este mecanismo reproduce dos aspectos clave de las neuronas biológicas: la generación de potenciales de acción y el periodo refractario (breve intervalo de tiempo después de un pico durante el cual la neurona no puede volver a disparar).

La frecuencia de disparo de la neurona LIF depende directamente de la magnitud de $I_m(t)$, como se ilustra en la figura 2.1(b). Si la corriente de entrada es débil, la fuga dominará, evitando que el potencial alcance el umbral y, por lo tanto, impidiendo la generación de picos. Por el contrario, corrientes más intensas provocarán una carga más rápida del capacitor, aumentando la frecuencia de disparo. Esta relación entre la intensidad de corriente en la entrada y la tasa de picos permite que las SNNs codifiquen información en patrones temporales de actividad, una propiedad fundamental para aplicaciones en inteligencia artificial y procesamiento neuromórfico.

La principal ventaja del modelo LIF radica en su simplicidad computacional y circuital, lo que facilita su implementación en sistemas embebidos y dispositivos neuromórficos. A diferencia de modelos más complejos, como Hodgkin-Huxley o Izhikevich [40], la LIF requiere menos parámetros y recursos de procesamiento, lo que la hace ideal para simulaciones a gran escala y aplicaciones en tiempo real.

En el diseño de sistemas neuromórficos, las neuronas LIF suelen implementarse utilizando componentes analógicos como amplificadores operacionales de transconductancia (OTAs) para integrar corrientes y transistores como interruptores controlados por voltaje [41]. Recientemente, los memristores han emergido como una alternativa prometedora para emular tanto la resistencia de fuga como la plasticidad sináptica, permitiendo diseños más compactos y energéticamente eficientes [42].

2.4 CONCLUSIONES

En este capítulo se describieron las redes neuronales de picos (SNNs) como un paradigma fundamental en inteligencia artificial y neurociencia computacional, destacando su capacidad para emular el procesamiento temporal de las neuronas biológicas. A diferencia de las redes neuronales tradicionales, las SNNs procesan información mediante pulsos (*spikes*) y dinámicas basadas en el tiempo, lo que las acerca a los principios de codificación de información observados en sistemas biológicos. Se analiza en profundidad el mecanismo *integrate-and-fire*, eje central de las SNNs, que combina la integración de señales sinápticas con umbrales de disparo y periodos refractarios. Este modelo, junto a variantes como *Leaky Integrate-and-Fire* (LIF) y adaptaciones no lineales, permiten reproducir comportamientos neuronales complejos, desde implementaciones bioinspiradas hasta aproximaciones híbridas para hardware neuromórfico. Finalmente, se caracteriza el comportamiento temporal de las SNNs, demostrando su capacidad para codificar información en latencias entre picos y frecuencias de disparo.

CAPÍTULO 3

EL MEMRISTOR

3.1 FUNDAMENTOS DEL MEMRISTOR

El memristor (o *memory resistor*) es el cuarto elemento pasivo fundamental en la teoría de circuitos, propuesto teóricamente por Leon O. Chua en 1971 [43]. A diferencia de los elementos tradicionales (resistencia R , capacitancia C e inductancia L) que relacionan pares de variables eléctricas (voltaje v , corriente i , carga q y flujo magnético ϕ) de manera lineal, el memristor establece una relación no lineal entre las variables de carga q y flujo ϕ , completando el conjunto de conexiones fundamentales entre estas variables. La carga eléctrica y la corriente están vinculadas por:

$$q(t) = \int_{-\infty}^t i(\tau) d\tau + q_0 \quad (3.1)$$

donde q_0 es la carga inicial. De forma análoga, el flujo magnético y el voltaje se relacionan mediante:

$$\varphi(t) = \int_{-\infty}^t v(\tau) d\tau + \varphi_0 \quad (3.2)$$

siendo φ_0 el flujo inicial. En la figura 3.1 se ilustran las relaciones entre estas variables para los elementos pasivos clásicos: Resistencia ($R(v, i)$), inductancia ($L(\varphi, i)$) y capacitancia ($C(q, v)$). El memristor relaciona las dos variables restan-

tes de carga q y flujo ϕ . Esta relación se denomina *memristancia* (M) y puede expresarse en función del flujo magnético $M(\varphi(t))$ o de la carga eléctrica $M(q(t))$ [1].

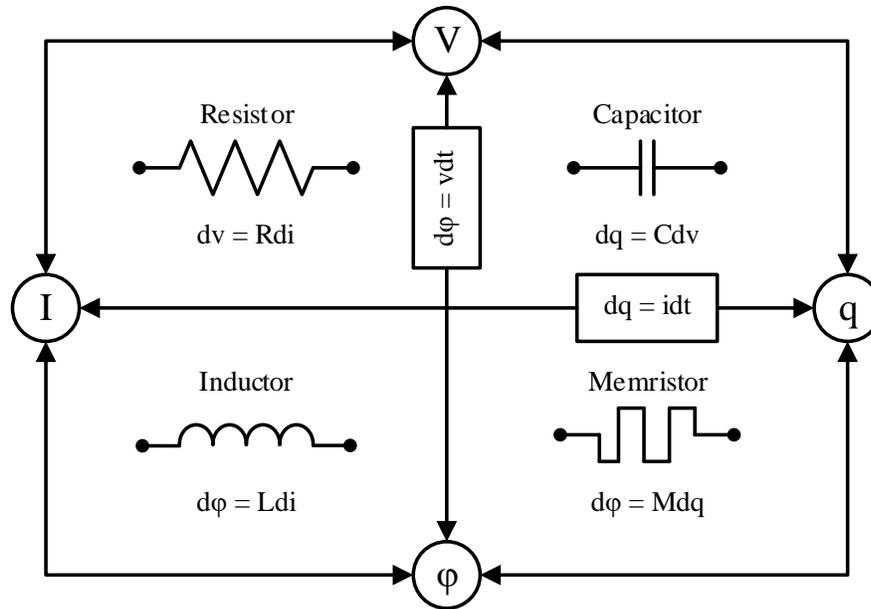


Figura 3.1: Elementos de circuito fundamentales de dos terminales: resistencia, capacitor, inductor y memristor [1].

Un memristor controlado por carga q se define por [43]:

$$v(t) = M(q(t)) i(t) \quad (3.3)$$

donde

$$M(q(t)) = \frac{d\varphi(q(t))}{dq(t)} \quad (3.4)$$

De manera similar, un memristor controlado por flujo ϕ se define por:

$$i(t) = W(\varphi(t)) v(t) \quad (3.5)$$

siendo $W(\varphi(t))$ la *memductancia*, dada por:

$$W(\varphi(t)) = \frac{dq(\varphi(t))}{d\varphi(t)} \quad (3.6)$$

En el plano corriente-voltaje ($I - V$), su curva de histéresis siempre cruza el origen, formando un *lazo pinchado*. Este patrón, imposible de replicar con R , L o

C , refleja su dependencia del historial de señales aplicadas. A altas frecuencias, el lazo se contrae hasta aproximarse a una línea recta, comportándose como una resistencia convencional [1].

Por otro lado, cuando se elimina la excitación externa al dispositivo, el memristor retiene su estado de resistencia (*memristancia*), actuando como una memoria *no volátil*. Esta propiedad lo hace ideal para almacenamiento de información y computación inspirada en sinapsis biológicas.

La capacidad del memristor para modificar su resistencia en función del flujo de carga (*plasticidad*) emula el fortalecimiento/debilitamiento de conexiones sinápticas en el cerebro [38]. Esto ha impulsado su uso en redes neuronales neuromórficas, donde dispositivos memristivos imitan el aprendizaje biológico.

3.2 MEMRISTORES EN NEURONAS TIPO LIF

El modelo *Leaky Integrate-and-Fire* (LIF) es representado mediante el circuito equivalente mostrado en la figura 2.1(a), cuya dinámica es gobernada por la ecuación diferencial (2.2). Como se ilustra en la figura 2.1(b), cuando el potencial de membrana $V_m(t)$ alcanza el umbral V_{th} , la neurona genera un *spike* y se reinicia al potencial de reposo u_{rest} . Si bien esta dinámica puede implementarse electrónicamente usando comparadores y circuitos de reset, estos componentes adicionales incrementan significativamente la complejidad del sistema [39].

Una solución innovadora, propuesta en [42], sustituye la resistencia de fuga R_{leak} y el interruptor de la figura 2.1(a) por un memristor volátil. Este dispositivo exhibe una *memconductancia* variable (dependiente de la acumulación de carga o flujo), permitiendo un intercambio abrupto en su resistencia interna y “olvidando” su estado. El sistema opera en tres fases características:

- **Estado de reposo:** El memristor mantiene alta resistencia, simulando la ele-

vada resistencia de membrana en condiciones basales.

- **Fase de integración:** La corriente de entrada $I_{in}(t) \approx \frac{V_{in}(t)}{R_s}$ carga progresivamente el capacitor C_m , haciendo que $V_m(t)$ aumente exponencialmente.
- **Disparo y reset:** Cuando $V_m(t)$ supera V_{th} , el memristor conmuta abruptamente a baja resistencia, descargando rápidamente C_m (generando un *spike*) y luego regresa espontáneamente a alta resistencia, eliminando la necesidad de circuitos de reset externos.

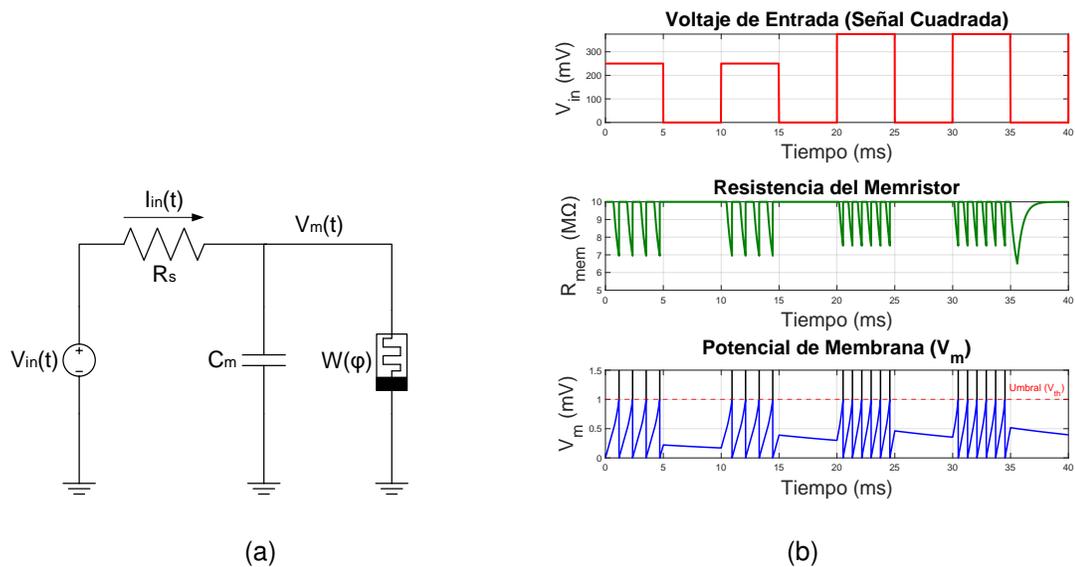


Figura 3.2: Modelo de la neurona LIF basada en memristor volátil: **(a)** Circuito esquemático, **(b)** Gráfica de voltaje de entrada vs *spikes* de salida.

Esto simplifica la arquitectura al eliminar componentes activos adicionales, permite un comportamiento más similar al biológico emulando procesos naturales de despolarización/repolarización y propone una mayor eficiencia energética al aprovechar las propiedades intrínsecas del memristor volátil. Esta implementación se esquematiza en la figura 3.2(a) y consta de una red para convertir voltaje en corriente ($V_{in}(t) \rightarrow I_{in}(t)$) debido a R_s , un capacitor C_m como elemento integrador de membrana y un memristor en paralelo que regula dinámicamente la fuga de carga [44]. Esto implica que la ecuación (2.2) se reescriba como:

$$C_m \frac{dV_m(t)}{dt} = I_{memristor} + \frac{V_{in}(t)}{R_s} \quad (3.7)$$

donde

$$I_{memristor} = -W(\varphi(t))(V_m(t) - u_{rest}) \quad (3.8)$$

La figura 3.2(b) presenta la evolución temporal del sistema descrito por la ecuación diferencial (3.7), donde se observa la relación no lineal entre la excitación de entrada y la respuesta de disparo neuronal, demostrando que el cambio en la *memconductancia* volátil durante el pulso de corriente afecta directamente a la cantidad de *spikes* generados.

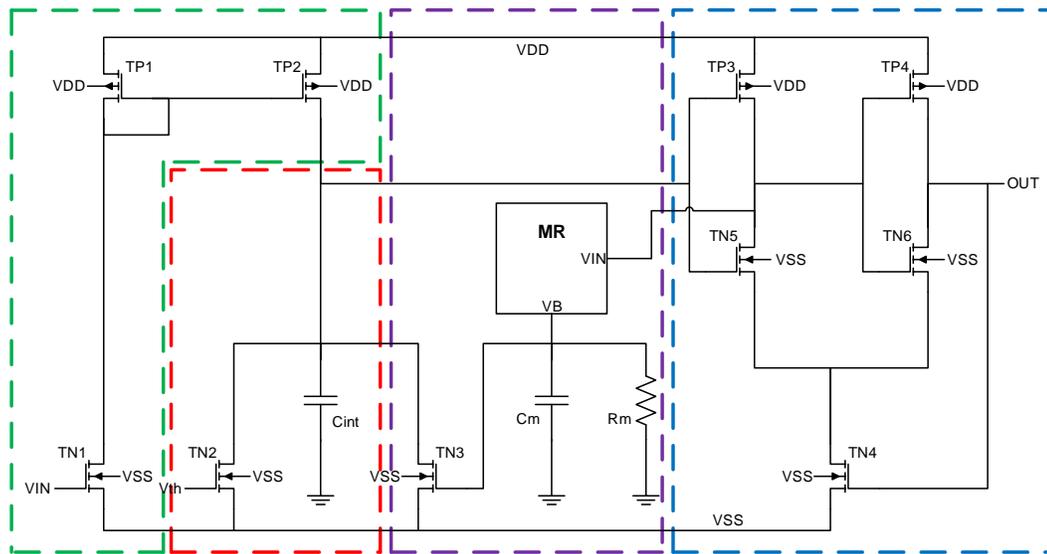


Figura 3.3: Neurona LIF basada en memristor dividido en cuatro bloques [2]: Espejo de corriente (verde). Integrador (rojo). Schmitt trigger (azul). Módulo de reset memristivo (morado).

Por otro lado, una implementación más compleja del circuito neuronal, representada en la figura 3.3 y basada en el trabajo de [2], emplea una arquitectura modular que divide al sistema en cuatro bloques funcionales interconectados:

- Espejo de corriente:** Esta etapa replica y aísla la corriente de entrada, proporcionando una señal estable al integrador. Su diseño simétrico garantiza una inyección de corriente precisa independientemente de las variaciones en la fuente de alimentación.

- **Etapa integradora:** Se implementa un capacitor C_{int} que acumula la corriente inyectada según la relación:

$$V_m(t) = \frac{1}{C_{int}} \int I_{in}(\tau) d\tau \quad (3.9)$$

donde $V_m(t)$ representa al potencial de membrana que crece exponencialmente hasta alcanzar el umbral de disparo.

- **Disparador Schmitt trigger:** Este circuito comparador con histéresis proporciona alta inmunidad al ruido mediante sus umbrales simétricos, generando *spikes* digitales bien definidos cuando $V_m(t)$ supera V_{th} y evita oscilaciones indebidas durante la fase de reset.
- **Módulo de reset memristivo (MRE):** Reemplaza los circuitos de reset convencionales mediante una transición automática del memristor a estado conductor durante el *spike* y un auto-reseteo pasivo a alta resistividad gracias a la volatidad intrínseca.

Esta arquitectura ofrece ventajas significativas como una mejora en la precisión del disparo gracias al *Schmitt trigger*, un mejor control dinámico para sintonizar los tiempos de integración (C_{int}) y reset (C_m) y la dinámica no volátil del memristor permite una disminución progresiva de la frecuencia de los *spikes* ante estímulos repetidos, así como una adaptación a patrones de entrada.

3.3 MEMRISTOR BASADO EN OTA

El diseño del memristor (MR) utilizado en la figura 3.3 como módulo de reset fue propuesto en [2], empleando un amplificador operacional de transconductancia (OTA) como bloque básico. El OTA actúa esencialmente como una fuente de corriente controlada por voltaje, donde la transconductancia (G_m) convierte la diferencia entre los voltajes de entrada (V^+ y V^-) en una corriente de salida (I_G).

Además, el valor de G_m puede ajustarse mediante un voltaje de polarización V_B . La relación voltaje-corriente del OTA y el valor de G_m vienen dados por:

$$I_G = G_m(V^+ - V^-), \quad G_m = \frac{k}{\sqrt{2}}(V_B - V_S - 2V_{th}) \quad (3.10)$$

donde V_B es el voltaje de polarización, V_S es el voltaje de fuente en el transistor que se utilice para polarizar el OTA, V_{th} es el voltaje de umbral de los MOSFET y $k = \mu_{n,p}C_{ox} \left(\frac{W}{L}\right)$ es el parámetro característico de la tecnología.

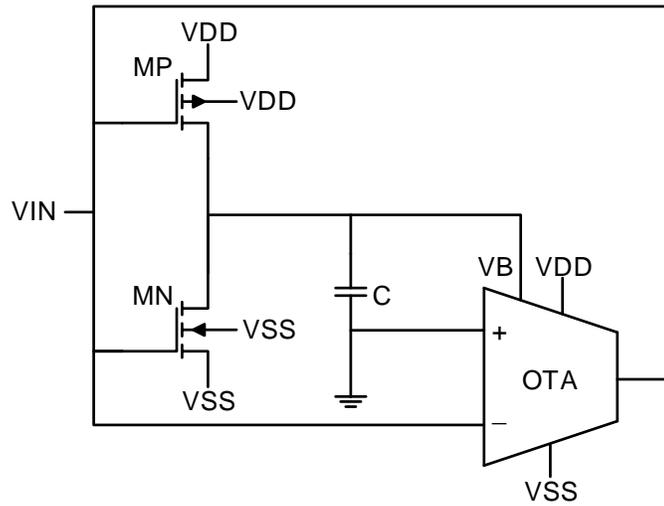


Figura 3.4: Emulador de memristor aterrizado [2].

El diagrama del circuito empleado para caracterizar el comportamiento memristivo se ilustra en la figura 3.4 y está compuesto por un inversor CMOS, un capacitor y un OTA. Como se describe en [2], la dinámica del sistema está gobernada por la interacción entre la transconductancia del OTA y el integrador formado por el inversor CMOS y el capacitor. La ecuación característica de este integrador está dada por:

$$V_B(t) = \frac{1}{C} \int I_{out}(\tau) d\tau \quad \rightarrow \quad V_B(t) = \frac{g_m}{C} \int V_{in}(\tau) d\tau, \quad (3.11)$$

donde g_m representa la transconductancia equivalente del inversor CMOS. Por lo tanto, la ecuación (3.10) se reescribe para expresar la transconductancia efectiva (G_m) del OTA en función del voltaje de entrada $V_{in}(t)$:

$$G_m = \frac{k}{\sqrt{2}} \left(\frac{g_m}{C} \int V_{in}(\tau) d\tau - V_S - 2V_{th} \right) \quad (3.12)$$

Para garantizar una aproximación de una transconductancia (g_m) lineal en el rango de operación de la señal de entrada ($-0.5V \leq V_{in} \leq 0.5V$), se dimensionan los transistores del inversor CMOS con $L = 1.08\mu m$, $W_N = 2.16\mu m$ y $W_P = 10.8\mu m$. Estas dimensiones, son seleccionadas considerando que el voltaje de polarización V_B debe asegurar que el voltaje de fuente (V_S) de un transistor interno permanezca como una tierra virtual, como se ilustra en la figura 3.5(b). Esto resulta en un voltaje de polarización en gran señal de $0.71V$, compatible con los niveles de voltaje típicos en tecnología CMOS 180 nm.

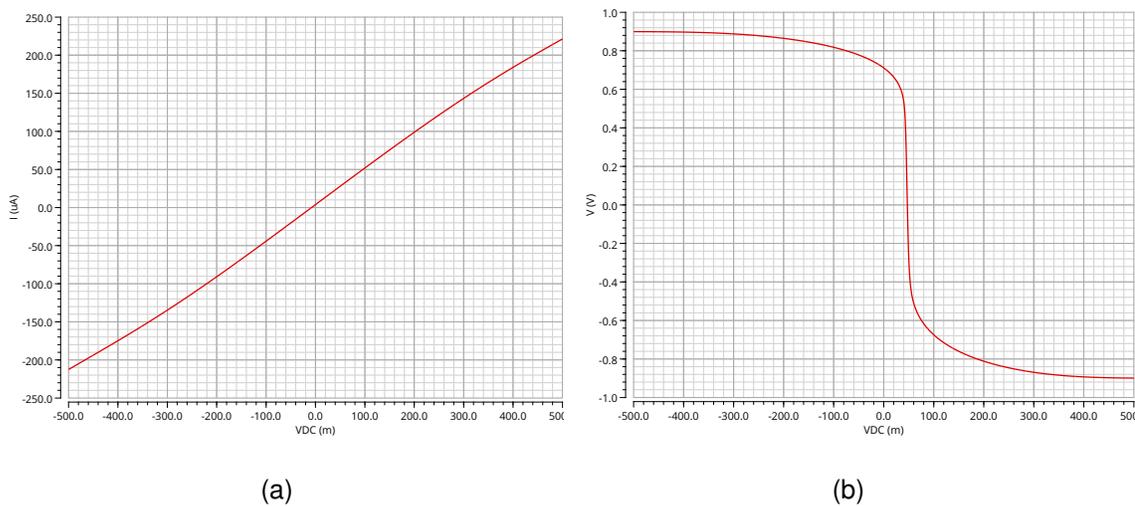


Figura 3.5: Análisis DC del inversor CMOS: (a) Transconductancia (g_m). (b) Voltaje de salida (V_B).

Dado que V_S actúa como tierra virtual, y considerando una señal de entrada $V_{in}(t) = A_m \sin(\omega t)$, donde A_m representa la amplitud y ω indica la frecuencia angular de la forma de onda sinusoidal. Como consecuencia, la *memductancia* se expresa como:

$$W(\varphi_{IN}(t)) = \frac{k}{\sqrt{2}} \left(\frac{A_m g_m \cos(\omega t - \pi)}{\omega C} - 2V_{th} \right) \quad (3.13)$$

En la figura 3.6 se ilustra una simulación en MATLAB de la ecuación (3.13), considerando una señal de entrada con amplitud de $A_m = 0.5V$ y una frecuencia de $f = \frac{\omega}{2\pi} = 100kHz$, una transconductancia del OTA de $g_m = 1ms$, un capacitor

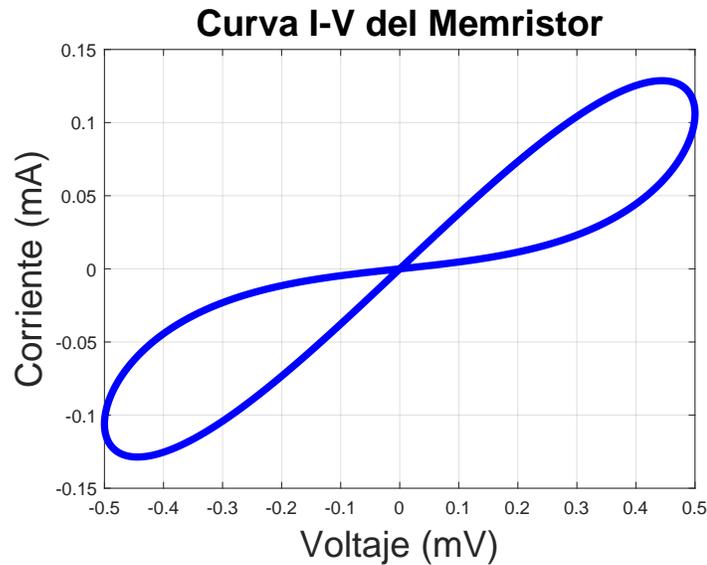


Figura 3.6: Curva característica del Memristor[2].

de $C = 1nF$ y las constantes de la tecnología $V_{th} \approx 0.5V$ y $k \approx 300\mu\frac{A}{V}$. Se puede observar el lazo de histéresis y la simetría en el cuadrante I y III del plano $I-V$. Para obtener una G_m constante en una región lineal que obtenga este comportamiento, se propone optimizar diferentes topologías de OTAs a través de una metodología que combina el método gm/I_D y algoritmos evolutivos (NSGA-II).

3.4 CONCLUSIONES

En este capítulo, se analizó en detalle la aplicación de memristores en neuronas LIF, comenzando con una revisión del modelo matemático clásico y su adaptación memristiva, seguido de una descripción de topologías de neuronas LIF memristivas y del comportamiento del memristor obtenido de la literatura. Finalmente, se enfatiza en la necesidad de una G_m lineal para generar un comportamiento más fiel a las ecuaciones que describen al memristor.

CAPÍTULO 4

OPTIMIZACIÓN gm/I_D Y NSGA-II

4.1 DISEÑO CMOS DEL OTA

Los amplificadores operacionales de transconductancia (OTA, por sus siglas en inglés) son componentes esenciales en el diseño de circuitos analógicos, utilizados en aplicaciones como filtros de precisión [45], circuitos de muestreo y retención [46], convertidores de datos analógico-digitales [47] y digital-analógicos [48]. A diferencia de los amplificadores operacionales convencionales, que proporcionan un voltaje de salida a partir de un voltaje de entrada, un OTA convierte el voltaje de entrada en una corriente de salida mediante una ganancia de transconductancia (G_m), representada en el plano $I - V$. Los OTAs ofrecen ventajas significativas frente a los amplificadores de voltaje tradicionales, como un mayor ancho de banda, un rango dinámico extendido, una mayor resistencia de salida y una respuesta de fase más estable [49].

Entre los parámetros que definen el rendimiento de un OTA se encuentran la ganancia de transconductancia lineal, la ganancia en lazo abierto (A_v) y un gran ancho de banda (BW). En la mayoría de los casos, todos los transistores en una topología CMOS OTA deben operar en la región de saturación [50]. Estas y otras características eléctricas pueden caracterizarse mediante simulaciones en los do-

minios de corriente continua (DC), corriente alterna (AC) y tiempo. El simulador de circuitos más utilizado es el *Simulation Program with Integrated Circuit Emphasis* (SPICE), disponible como software de código abierto bajo el nombre de ngspice. Sin embargo, el diseño CMOS de un amplificador es muy complejo cuando se realiza mediante prueba y error, lo que conlleva un gran número de simulaciones SPICE. Afortunadamente, el diseño de un amplificador CMOS puede automatizarse y optimizarse combinando el método g_m/I_D [31] y el algoritmo NSGA-II [51].

En este trabajo, se pretende lograr una ganancia de transconductancia estable y bien definida en una región lineal del OTA para diseñar el emulador de memristor de la figura 3.4. Para ello, se estudian tres topologías CMOS de OTA: el *Current Mirror OTA* (figura 4.1) [3], el *Folded Cascode OTA* (figura 4.2) [4] y el *Low-Voltage Current Mirror OTA* (figura 4.3) [5]. En cada topología de OTA, se incluye un voltaje de polarización V_B , asociado al diseño del memristor mostrado en la figura 3.4, el cual requiere de un OTA.

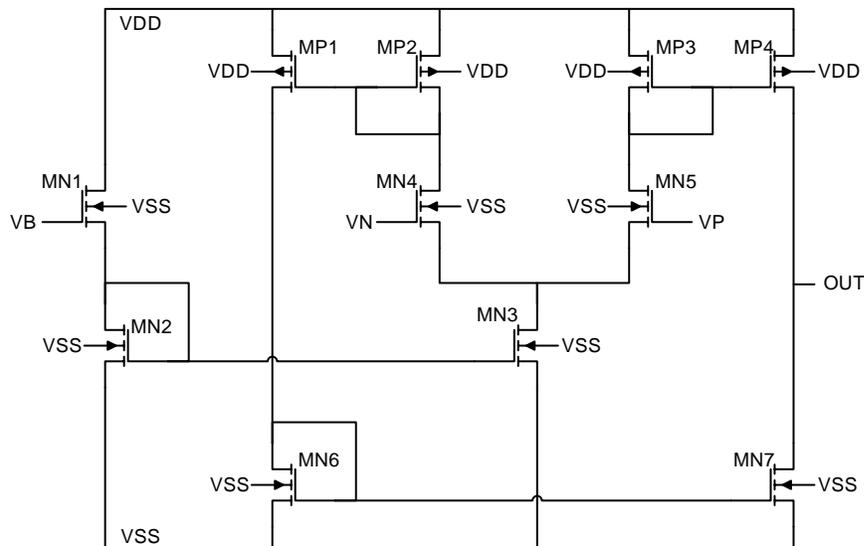


Figura 4.1: Topología *Current Mirror OTA* [3].

La metodología automatizada de dimensionamiento propuesta implica la descripción de los OTAs en un netlist compatible con SPICE, donde el análisis DC permite evaluar la ganancia de transconductancia, graficada en el plano $I - V$ al

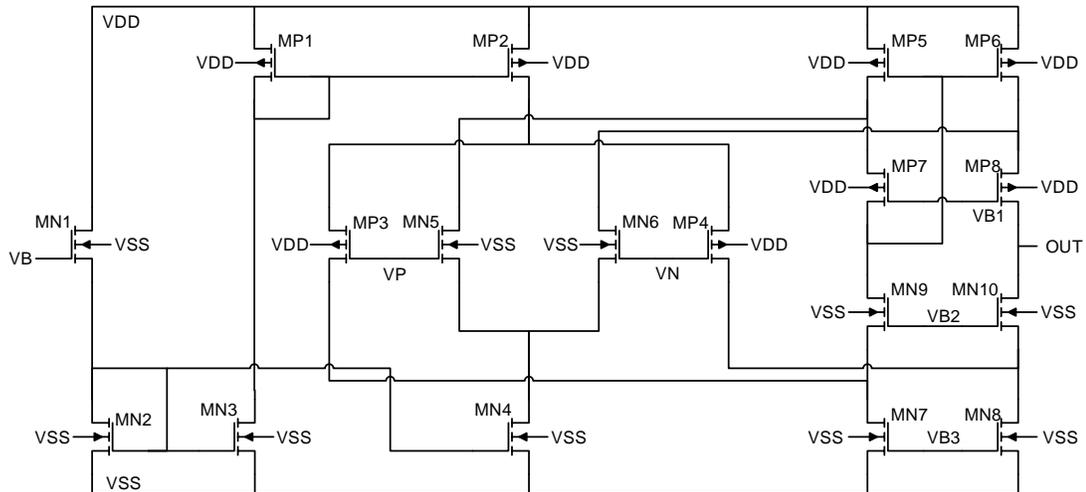


Figura 4.2: Topología *Folded Cascode OTA* [4].

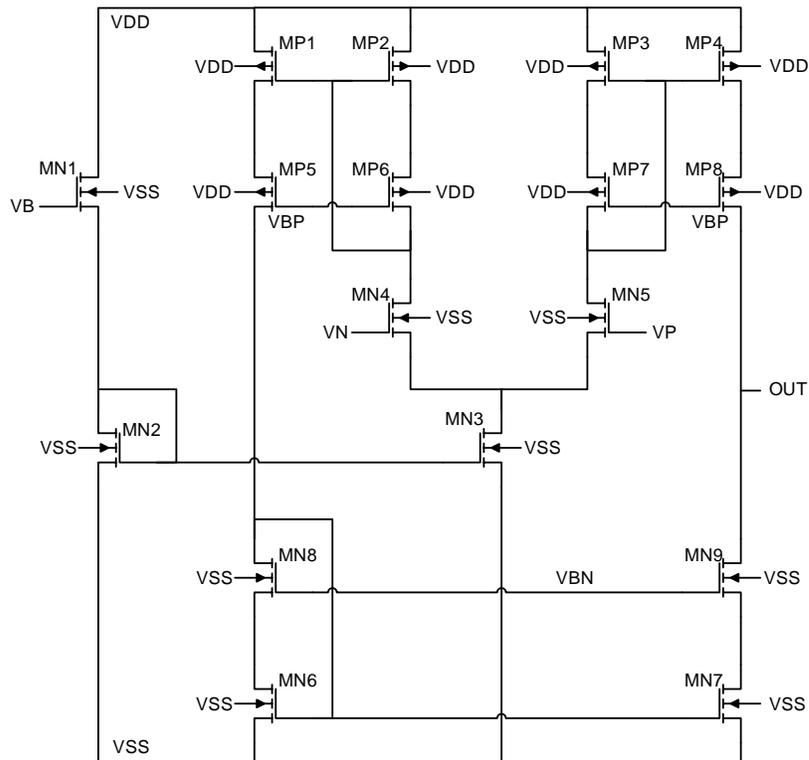


Figura 4.3: Topología *Low Voltage Current Mirror OTA* [5].

barrer el voltaje de entrada (en el rango de $-0.2V$ a $0.2V$) y observar la corriente de salida correspondiente. La figura 4.4 muestra una comparación entre la curva $I - V$ ideal y la obtenida de un OTA CMOS real, resaltando las desviaciones causadas

por no linealidades. Por lo tanto, para maximizar la linealidad de la transconductancia en el rango de voltaje denominado región lineal $[-0.2V .. 0.2V]$, se cuantifica mediante una ecuación que calcula el error cuadrático medio (RMSE, (4.1)). Este cálculo compara los valores de corriente ideales y reales en n puntos, garantizando un comportamiento predecible y óptimo para aplicaciones críticas, como la dinámica en el diseño del emulador de memristor.

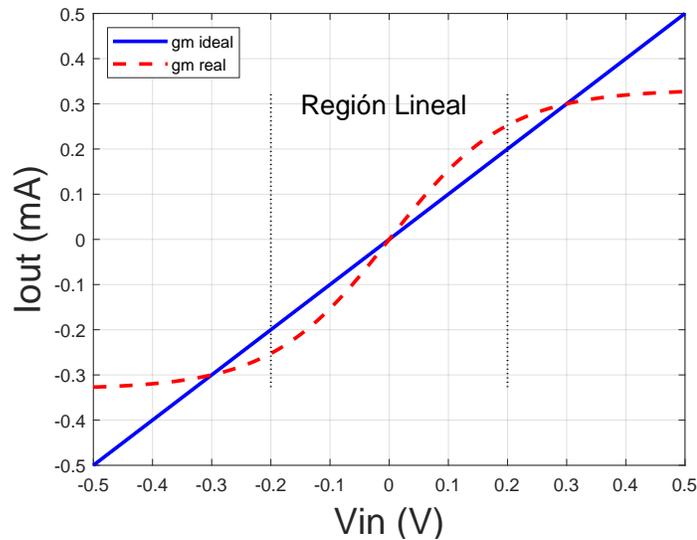


Figura 4.4: Comparación entre la transconductancia ideal y real en el plano $I - V$ para una topología OTA CMOS.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (I_{ideal} - I_{real})^2} \quad (4.1)$$

4.2 MÉTODO DE DISEÑO gm/I_D Y CODIFICACIÓN ENTERA

Los transistores MOSFET son fundamentales en el diseño de circuitos integrados CMOS, utilizándose tanto en aplicaciones digitales como analógicas. Sin embargo, el diseño manual de MOSFETs para cumplir con especificaciones analógicas es complejo debido al gran número de variables involucradas y los amplios rangos

de valores factibles. Afortunadamente, hace una década, los autores en [31] introdujeron el método gm/I_D , que hoy en día se considera una solución sistemática para optimizar amplificadores CMOS donde los transistores operan en regiones de inversión fuerte, moderada y débil. Este enfoque de dimensionamiento se basa en la relación de transconductancia (g_m) con la corriente de drenaje (I_D) de un transistor dado y es independiente de sus tamaños (W/L), dependiendo únicamente de la polarización del mismo. Este método permite unificar el análisis de pequeña y gran señal, además de facilitar la optimización de características del amplificador como ganancia, ancho de banda, ruido y consumo de potencia [32].

Hace una década, los autores en [51] aplicaron el método gm/I_D dentro de un bucle de optimización ejecutando NSGA-II, donde los tamaños se codificaron como variables enteras que multiplican el parámetro lambda (λ) definido por la tecnología CMOS, que para TSMC 180 nm es $\lambda = 90nm$. La nueva contribución documentada en esta Tesis está dedicada a alcanzar un valor de transconductancia deseado para una topología OTA específica. Este problema de optimización implica evaluar una ganancia de transconductancia lineal medida como un valor RMSE, como se indica en (4.1). Garantizar una transconductancia lineal puede mejorar el diseño de un emulador de memristor, como se mostrará en las siguientes secciones.

La metodología gm/I_D comienza con la caracterización de los parámetros de los modelos de tecnología CMOS para los transistores de canal N y canal P. Básicamente, se generan tablas de búsqueda (LUTs) para relacionar la eficiencia de transconductancia (gm/I_D), ganancia intrínseca (g_m/g_o), densidad de corriente (I_D/W) y frecuencia de transición (f_T) para diferentes longitudes de canal (L), así como voltajes V_{GS} , V_{DS} y V_{SB} . Según las especificaciones de diseño, se elige L (canales cortos para alto ancho de banda, canales largos para alta ganancia) y se obtiene el valor óptimo de gm/I_D . Finalmente, las LUTs se comparan para determinar la relación W/L que cumple con las restricciones de diseño [52].

Un código MATLAB propuesto en [32] automatiza la generación de estas LUTs y extrae los valores de W para la L elegida y las condiciones de polarización corres-

pendientes para cada transistor, basándose en un netlist SPICE predefinido. Para garantizar que los tamaños W/L sean múltiplos enteros de λ , se puede realizar una codificación entera en la generación del netlist SPICE [51]. Por lo tanto, se debe definir un parámetro global λ (LAMBDA) en el netlist SPICE, de modo que los tamaños W/L se expresen como múltiplos de λ , como se muestra en el código 1. Se puede apreciar que las variables de diseño W y L de los transistores se codifican en múltiplos del parámetro λ , de manera que el ancho de cada transistor (W_X) se expresa como $W_X = V_X \cdot \lambda$, donde V_X es un entero adimensional y X denota cada variable de diseño en las diferentes topologías OTA.

Código 1 Ejemplo de uso de λ como escalamiento de parámetros en una netlist de SPICE.

```
.PARAM LAMBDA = 0.09u
```

```
*
```

```
M_P D G S B PMOS L=12*LAMBDA W=V1*LAMBDA
```

```
M_N D G S B NMOS L=12*LAMBDA W=V2*LAMBDA
```

Como se discutió anteriormente, el dimensionamiento de transistores en el diseño analógico CMOS presenta desafíos significativos debido a las múltiples compensaciones entre las características eléctricas del OTA en los dominios de análisis DC, AC y temporal. Para abordar el problema de dimensionamiento, se recomienda realizar un prediseño de un OTA aplicando el método g_m/I_D . Se considera lograr un valor de ganancia de transconductancia de 1 mS y tomando en cuenta las especificaciones objetivo listadas en la tabla 4.1.

Tabla 4.1: Especificaciones objetivo para el dimensionamiento de los OTAs CMOS que se muestran en las figuras 4.1, 4.2 y 4.3.

Parámetro	Valor
Tecnología CMOS TSMC	180 nm
Voltaje de Alimentación (V)	± 0.9
Capacitancia de Carga (pF)	0.1
Ganancia (dB)	≥ 40
Ancho de Banda (kHz)	≥ 100
Transconductancia	$\approx 1 \text{ mS}$

Las tres topologías de OTA estudiadas, mostradas en las figuras 4.1, 4.2 y 4.3,

Tabla 4.2: Variables de codificación de transistores para las topologías: (a) Current Mirror OTA. (b) Folded Cascode OTA. (c) Low Voltage Current Mirror OTA.

Transistor	$W/L(\mu m)$
M_{N1}	17.28/1.08
M_{N2}, M_{N3}	$W_1/1.08$
M_{N4}, M_{N5}	$W_2/1.08$
M_{N6}, M_{N7}	$W_3/1.08$
$M_{P1}, M_{P2}, M_{P3}, M_{P4}$	$W_4/1.08$

(a)

Transistor	$W/L(\mu m)$	Transistor	$W/L(\mu m)$
M_{N1}	17.28/1.08	M_{N1}	17.28/1.08
M_{N2}, M_{N3}, M_{N4}	$W_1/1.08$	M_{N2}, M_{N3}	$W_1/1.08$
M_{N5}, M_{N6}	$W_2/1.08$	M_{N4}, M_{N5}	$W_2/1.08$
$M_{N7}, M_{N8}, M_{N9}, M_{N10}$	$W_3/1.08$	$M_{N6}, M_{N7}, M_{N8}, M_{N9}$	$W_3/1.08$
M_{P1}, M_{P2}	$W_4/1.08$	$M_{P1}, M_{P2}, M_{P3}, M_{P4}$	$W_4/1.08$
M_{P3}, M_{P4}	$W_5/1.08$	$M_{P5}, M_{P6}, M_{P7}, M_{P8}$	$W_5/1.08$
M_{P5}, M_{P6}	$W_6/1.08$		
M_{P7}, M_{P8}	$W_7/1.08$		

(b)

(c)

tienen voltajes de polarización configurados de la siguiente manera: $V_B = 0.71V$, $V_{B3} = V_{BN} = -0.2V$, $V_{B2} = 0.0V$ y $V_{B1} = V_{BP} = 0.2V$. La codificación entera puede observarse en la tabla 4.2, que muestra todas las variables de diseño W asociadas a los tamaños de los transistores para cada topología de OTA, considerando una $L = 12\lambda = 1.08 \mu m$. Los resultados del dimensionamiento aplicando el método g_m/I_D se resumen en la tabla 4.3, la cual incluye los anchos de los transistores (W_X), derivados de las tablas proporcionadas en la tabla 4.2. La tabla 4.3 también incluye los parámetros eléctricos clave, como la ganancia en lazo abierto (A_v), el ancho de banda (BW) y la ganancia de transconductancia (G_m).

Los resultados de simulaciones en Cadence-Virtuoso, utilizando los datos en la tabla 4.3, proporcionan las características de transferencia (I_{OUT} vs V_{IN}) de cada topología de OTA. La curva $I - V$ mostrada en la figura 4.5(a), correspondiente al

Tabla 4.3: Primera iteración con el método gm/I_D para dimensionar los OTAs.

Topología	$W_1(\mu m)$	$W_2(\mu m)$	$W_3(\mu m)$	$W_4(\mu m)$	$W_5(\mu m)$	$W_6(\mu m)$	$W_7(\mu m)$	A_v (dB)	BW (kHz)	G_m (mS)
Figura 4.1	118.8	14.4	25.2	1004.4	-	-	-	34.82	3233	0.943
Figura 4.2	118.8	14.4	25.2	1004.4	169.2	1004.4	1033.2	84.12	16.59	0.590
Figura 4.3	118.8	14.4	25.2	1004.4	1033.2	-	-	25.78	524.3	0.708

OTA de la figura 4.1, muestra una transconductancia con buena linealidad pero con una marcada asimetría en su rango de voltaje de entrada operativo. En contraste, la figura 4.5(b) muestra la curva para el OTA de la figura 4.2, que presenta un comportamiento lineal solo en la región positiva de V_{IN} , perdiendo linealidad en la región negativa. Finalmente, la curva $I-V$ mostrada en la figura 4.5(c), asociada al OTA de la figura 4.3, exhibe una similitud parecida al de la figura 4.5(a), aunque comparte el problema de asimetría observado en las otras topologías de OTA. Estos resultados de simulación $I-V$ demuestran que, aunque todas las topologías logran cierta linealidad en la transconductancia, ninguna alcanza un comportamiento completamente simétrico en todo el rango de voltaje de entrada operativo. Es por esta razón que en esta Tesis se ejecuta NSGA-II para lograr una transconductancia deseada que mejore el diseño de un memristor CMOS, como se muestra en las siguientes secciones.

4.3 ALGORITMO EVOLUTIVO NSGA-II

Los resultados de dimensionamiento presentados en la tabla 4.3 y la figura 4.5 demuestran que la caracterización automatizada mediante el método gm/I_D por sí sola no garantiza una transconductancia constante en toda la región lineal, como la definida en la figura 4.4. Para abordar esta limitación, se propone una metodología de dimensionamiento basada en el algoritmo NSGA-II, que presenta la capacidad de manejar variables enteras discretas ($V1, V2, \dots$) durante la exploración del espacio de diseño y la capacidad de optimizar múltiples objetivos simultáneamente bajo restricciones definidas [33]. El proceso de dimensionamiento implica la escritura de netlists SPICE manipulando valores enteros para generar los individuos asociados

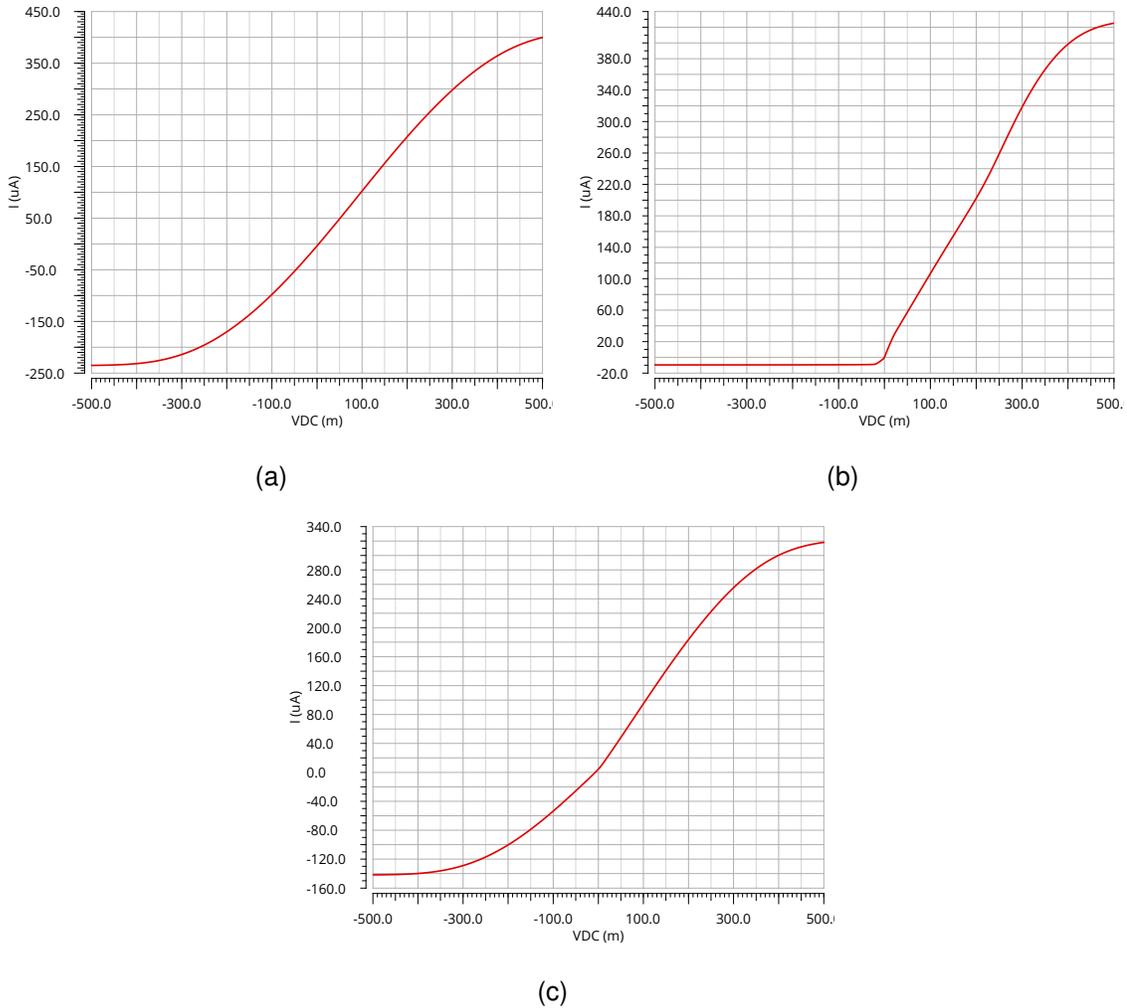


Figura 4.5: Simulación de la característica $I - V$ de las tres topologías OTAs utilizando los tamaños de la primera iteración aplicando el método g_m/I_D : (a) Current Mirror OTA. (b) Folded Cascode OTA. (c) Low Voltage Current Mirror OTA.

a los parámetros físicos del circuito.

4.3.1 ALGORITMO NSGA-II

El algoritmo NSGA-II (Non-dominated Sorting Genetic Algorithm II) resulta particularmente útil para resolver problemas complejos de optimización multiobjetivo, como los presentes en el diseño de circuitos integrados analógicos [34]. Este algo-

ritmo evolutivo multiobjetivo ha demostrado especial eficacia en el dimensionamiento de transistores para OTAs CMOS, donde se requieren cuidadosos equilibrios en múltiples parámetros de rendimiento que están en conflicto.

En NSGA-II, el proceso de optimización comienza con la generación de una población inicial de soluciones potenciales. En esta Tesis, la población consta de 20 individuos cuyos parámetros derivan de valores base obtenidos de la metodología g_m/I_D (mostrados en la tabla 4.3) para restringir el espacio de búsqueda. Cada individuo representa un conjunto de dimensiones de transistores para una topología OTA específica, codificadas mediante variables enteras V_X en el rango [40, 8000], correspondientes a múltiplos del parámetro tecnológico λ que determinan las dimensiones físicas de los transistores (W/L).

NSGA-II opera mediante un proceso iterativo de 50 generaciones, donde cada ciclo comprende etapas fundamentales de evaluación, selección y recombinación, según se describe en el pseudocódigo del algoritmo 1 para resolver los objetivos definidos en (4.2). Durante la fase de evaluación, cada candidato de topología OTA se simula eléctricamente usando SPICE para obtener valores precisos de los parámetros objetivo: ganancia en lazo abierto (A_v), ancho de banda (BW) y linealidad de transconductancia. Estas simulaciones proporcionan una cuantificación precisa del rendimiento del circuito, superando aproximaciones analíticas iniciales.

$$\begin{aligned}
 &\text{maximizar} && f_x(\text{Transconductancia lineal, Ganancia DC, BW}) \\
 &\text{sujeto a:} && A_v \geq 40dB, BW \geq 100kHz, RMSE \leq 10^{-9}, \text{ Margen de Fase} \geq 55^\circ
 \end{aligned} \tag{4.2}$$

La selección para reproducción emplea un mecanismo de torneo binario que prioriza soluciones en los frentes de Pareto (PF) más avanzados. El concepto de no-dominancia juega un papel crucial en esta etapa, permitiendo clasificar soluciones según su capacidad para optimizar múltiples objetivos simultáneamente sin que ninguna solución sea claramente superior en todos los aspectos. Complementariamente, la métrica de distancia de hacinamiento (*crowding distance*) preserva la diversidad poblacional, evitando convergencia prematura a óptimos locales mientras mantiene un amplio espectro de soluciones alternativas.

Algoritmo 1 Algoritmo NSGA-II

-
- 1: Definir: *Número de generaciones* G_{max} , *Tamaño de la población* P_N , *Tamaño de bits para los individuos*, *Espacios de búsqueda de los parámetros MOSFET (canal W y L)*
 - 2: Crear una netlist SPICE del OTA que se está optimizando
 - 3: Inicializar la población \mathbf{P} en función de la primera iteración de las variables de diseño W_X utilizando el método g_m/I_D
 - 4: Actualizar los valores de las variables de diseño en la lista de conexiones SPICE *.lib
 - 5: Asignar rango (nivel) según la dominancia de Pareto - ordenar
 - 6: Generar población de hijos
 - 7: Selección de torneo binario
 - 8: Recombinación y mutación
 - 9: **for** $G = 1$ to G_{max} **do**
 - 10: **for** cada padre e hijo en la población **do**
 - 11: Asignar rango (nivel) según Pareto - ordenar
 - 12: Generar conjuntos de vectores no dominados a lo largo del Frente de Pareto
 - 13: Bucle (interior) añadiendo soluciones a la siguiente generación comenzando desde el primer frente hasta que se encuentren N individuos que determinan la distancia de hacinamiento entre los puntos de cada frente
 - 14: **end for**
 - 15: Seleccionar puntos (elitistas) en el frente inferior (con rango inferior) y fuera de una distancia de aglomeración
 - 16: Crear la próxima generación
 - 17: Selección de torneo binario
 - 18: Recombinación y mutación
 - 19: **end for**
 - 20:
 - 21: **return** Población de última generación =0
-

Los operadores genéticos aplicados presentan características específicas adaptadas a los retos del diseño analógico. El *Simulated Binary Crossover* (SBX) opera con probabilidad $p_c = 0.9$ y parámetro de distribución $\eta = 15$, favoreciendo la generación de descendencia que preserva características parentales intermedias mientras explora regiones adyacentes del espacio de diseño. Por su parte, la mutación polinómica actúa con probabilidad $p_m = 0.5$ y $\eta = 20$, introduciendo modificaciones controladas que ayudan a superar posibles estancamientos en la optimización. Estos valores se seleccionaron basándose en estudios empíricos que demuestran su efectividad para problemas similares de dimensionamiento de circuitos analógicos.

Un aspecto crítico de la implementación implica el manejo de restricciones de diseño, particularmente el offset DC, margen de fase, condiciones de saturación de transistores y requisitos mínimos de ganancia/ancho de banda para garantizar el cumplimiento de las especificaciones objetivo de la tabla 4.1. NSGA-II incorpora estos requisitos mediante un mecanismo de penalización que afecta la aptitud (*fitness*) de soluciones no factibles, dirigiendo gradualmente la población hacia regiones del espacio de diseño que satisfacen las especificaciones técnicas.

La convergencia del algoritmo produce un frente de Pareto (PF) que contiene soluciones óptimas no dominadas, cada una representando un punto específico en el espacio de compensación de objetivos. Para el diseño de OTAs, esto se traduce en un conjunto de configuraciones que van desde versiones de alta-ganancia/baja-velocidad hasta diseños de ganancia-moderada/gran-ancho-de-banda, incluyendo opciones intermedias que equilibran ambos parámetros mientras mantienen una respuesta de transconductancia lineal.

4.3.2 DIMENSIONAMIENTO DE OTAS USANDO NSGA-II

El proceso de optimización multiobjetivo para el dimensionamiento de transistores en OTAs utilizando el algoritmo NSGA-II sigue una metodología sistemática que integra técnicas computacionales como se describe en la figura 4.6. El problema de dimensionamiento descrito por (4.2) considera variables de diseño correspondientes a las dimensiones geométricas de los transistores en cada topología, codificadas como múltiplos enteros del parámetro tecnológico λ dentro de un rango restringido entre 4 y 8000. Estas variables determinan completamente la configuración del amplificador y se optimizan simultáneamente para maximizar tres objetivos críticos: linealidad de transconductancia, ganancia DC y ancho de banda, mientras satisfacen cuatro restricciones descritas por (4.2), y además garantizan que todos los transistores permanezcan en la región de saturación.

La población inicial para el algoritmo NSGA-II se genera a partir de valores

prediseñados obtenidos mediante la metodología g_m/I_D , asegurando un punto de partida técnicamente sólido. A lo largo del proceso evolutivo, cada configuración candidata se somete a una evaluación rigurosa mediante un procedimiento de simulación de tres etapas ejecutado en SPICE. La primera etapa realiza un análisis de punto de operación (OP) para verificar que todos los transistores operen en saturación, condición esencial para el funcionamiento adecuado del amplificador. La segunda etapa caracteriza la respuesta en frecuencia y extrae valores de ganancia y ancho de banda mediante análisis AC. La tercera etapa evalúa la linealidad de transconductancia a través de análisis DC de las características de transferencia corriente-voltaje. Este enfoque de simulación multi-etapa captura precisamente las interdependencias entre varios parámetros de rendimiento.

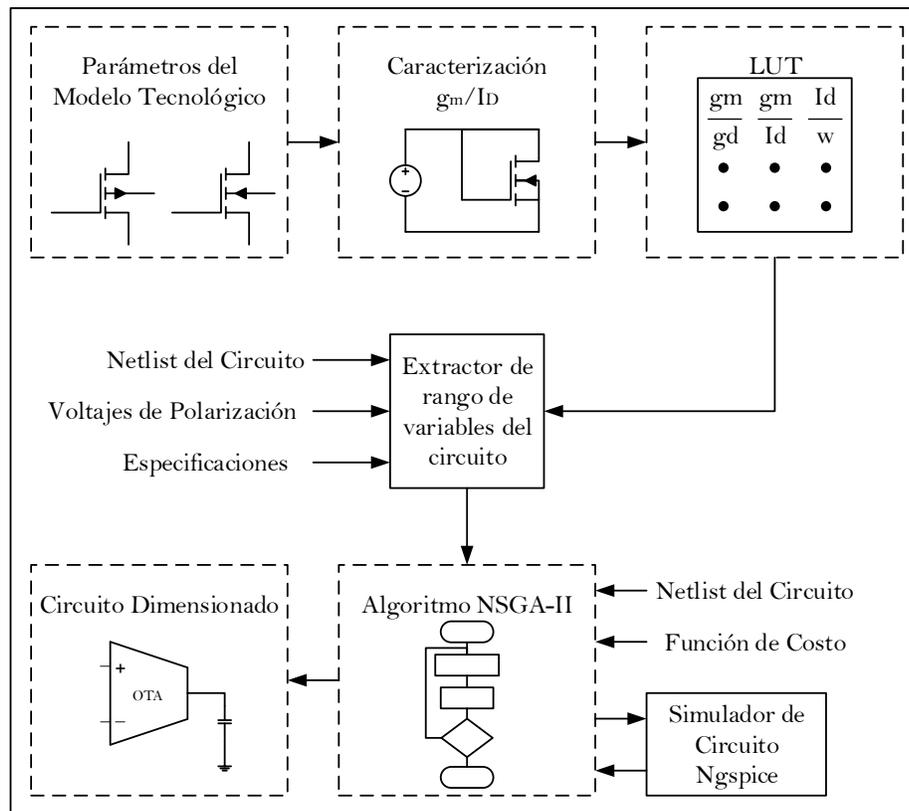


Figura 4.6: Enfoque propuesto para optimizar el dimensionamiento del circuito.

La gestión de restricciones implementa un esquema de validación binaria donde cada condición se evalúa independientemente. Las soluciones que violan cual-

quier restricción reciben penalizaciones de fitness, dirigiendo gradualmente la población hacia regiones factibles del espacio de diseño. Se presta especial atención a especificaciones críticas que incluyen margen de fase, offset de salida, requisitos mínimos de ganancia y ancho de banda, además de mantener todos los transistores en saturación.

La implementación computacional optimiza la eficiencia mediante integración directa entre el núcleo NSGA-II, programado en Python, y SPICE. Este acoplamiento evita la sobrecarga asociada con lenguajes interpretados y permite un procesamiento rápido de resultados. Cada evaluación genera un archivo de parámetros único que se incorpora automáticamente a los scripts de simulación, garantizando consistencia en las condiciones de análisis.

Tabla 4.4: Última generación con NSGA-II para dimensionamiento de OTAs.

Topología	$W_1(\mu m)$	$W_2(\mu m)$	$W_3(\mu m)$	$W_4(\mu m)$	$W_5(\mu m)$	$W_6(\mu m)$	$W_7(\mu m)$	A_v (dB)	BW (kHz)	G_m (mS)
Figura 4.1	115.2	21.6	7.2	637.2	-	-	-	35.86	2512	1.096
Figura 4.2	10.8	115.2	97.2	525.6	284.4	698.4	561.6	39.13	1741	0.835
Figura 4.3	126	18	32.4	680.4	680.4	-	-	45.6	813.4	0.867

Los resultados de dimensionamiento proporcionados por NSGA-II para la generación final se presentan en la figura 4.7, donde se pueden apreciar los 20 individuos para cada topología OTA. Los mejores resultados de dimensionamiento para cada topología OTA se resumen en la tabla 4.4, donde la solución óptima se identifica por la línea gruesa en la figura 4.7, mostrando la linealidad más precisa dentro de la región lineal elegida.

4.4 CONCLUSIONES

En este capítulo se describieron las topologías de OTAs seleccionadas para emular la dinámica del memristor analizado en el capítulo 3. Se propone una metodología híbrida que combina el método gm/I_D —para caracterizar la población inicial de transistores— combinado con el algoritmo NSGA-II, encargado de la optimización multiobjetivo. Además, se definen las restricciones de diseño, los objetivos

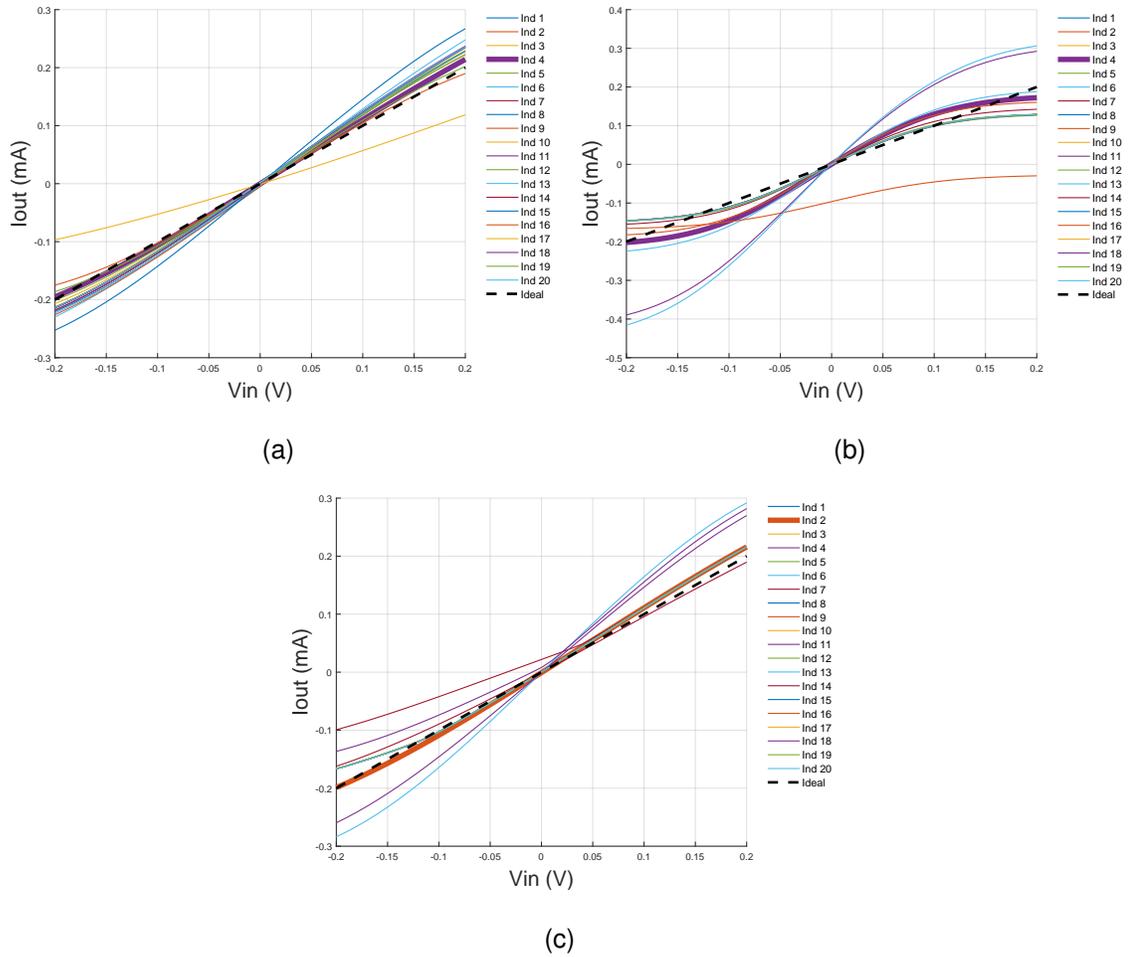


Figura 4.7: Resultados de dimensionamiento con NSGA-II para las topologías: (a) *Current Mirror OTA*, (b) *Folded Cascode OTA*, (c) *Low Voltage Current Mirror OTA*.

de la optimización para optimizar la ganancia en lazo abierto, el ancho de banda y una transconductancia lineal, y la codificación entera de las dimensiones W de los transistores en cada topología, limitando el espacio de búsqueda a múltiplos enteros de λ , según las reglas de diseño CMOS 180nm.

SIMULACIONES Y RESULTADOS

5.1 OTAS DISEÑADOS CON PRIMERA ITERACIÓN gm/I_D

En la sección 4.2 del capítulo 4 se realizó el dimensionamiento de los OTAs a través de la caracterización gm/I_D . Estos valores fueron utilizados para generar la primera población en el algoritmo evolutivo NSGA-II. Sin embargo, para comprobar su comportamiento en la emulación de circuitos complejos, se propone evaluarlos en las aplicaciones del emulador de memristor aterrizado de la figura 3.4 y en la neurona de picos de la figura 3.3.

5.1.1 MEMRISTOR BASADO EN OTAS CON gm/I_D

En la figura 5.1 se ilustran los lazos de histéresis del emulador de memristor aterrizado de la figura 3.4, usando las tres topologías de OTAs, y cada OTA diseñado con el método gm/I_D . Los lazos de las figuras 5.1(a) y 5.1(c) se observan adecuados para la emulación del comportamiento del memristor, son pinchados muy cerca del origen y con curvas casi simétricas. Sin embargo, aunque el lazo de la figura 5.1(b) esté pinchado en el origen, presenta deformaciones en su curva para cumplir la simetría entre el cuadrante I y III.

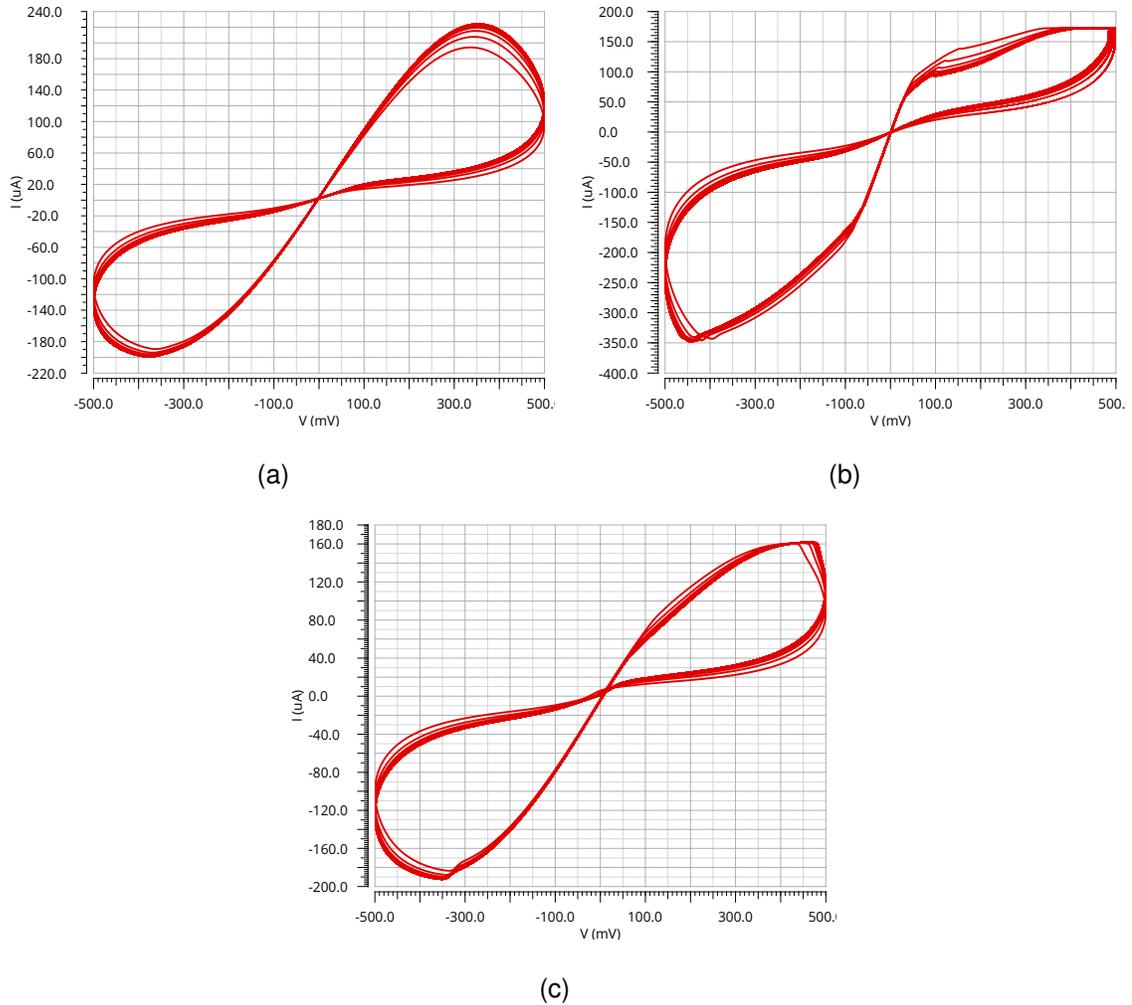


Figura 5.1: Lazos de histéresis del emulador de memristor CMOS pre-diseñado con gm/I_D para las topologías: (a) *Current Mirror OTA*. (b) *Folded Cascode OTA*. (c) *Low Voltage Current Mirror OTA*.

Tabla 5.1: Dimensiones de transistores para la neurona de picos basada en memristor de la figura 3.3.

Transistor	$W/L(\mu m)$
T_{N1}	10.8/1.08
T_{N2}, T_{N3}	2.16/1.08
T_{N4}, T_{N5}, T_{N6}	0.22/0.18
T_{P1}, T_{P2}	21.6/1.08
T_{P3}, T_{P4}	0.55/0.18

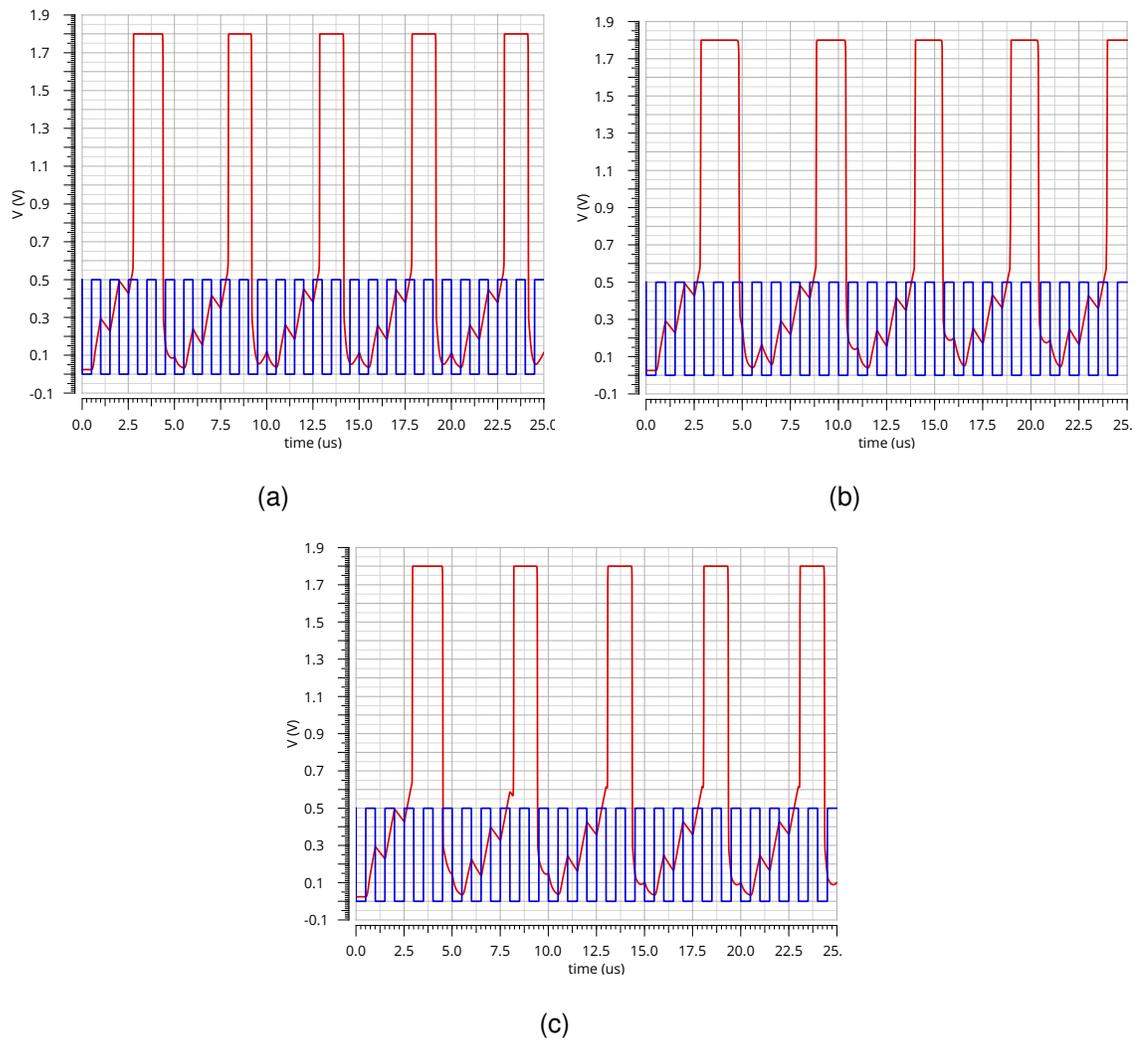
5.1.2 NEURONA DE PICOS BASADA EN MEMRISTOR CON gm/I_D 

Figura 5.2: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) pre-diseñada con gm/I_D (señal de entrada: $T = 1\mu s$, azul) para las topologías ($C_{int} = 10pF$ y $C_m = 1000pF$): **(a)** *Current Mirror OTA*, **(b)** *Folded Cascode OTA*. **(c)** *Low Voltage Current Mirror OTA*.

Por otro lado, haciendo uso de los memristores diseñados con la caracterización gm/I_D y las dimensiones de transistores en la tabla 5.1, se emula el comportamiento de la neurona de picos de la figura 3.3. En la figura 5.2 se muestra la respuesta temporal de la neurona basada en memristor con los diferentes OTAs dimensionados con el método gm/I_D . El tiempo de integración en las figuras 5.2(a)

y 5.2(b) son similares, mientras que la integración de la figura 5.2(c) toma un poco más de tiempo después del voltaje de umbral para disparar el pico. Sin embargo, el tiempo que se mantiene el pico en las figuras 5.2(a) y 5.2(c) es similar, a diferencia de la figura 5.2(b), cuyo tiempo manteniendo el pico es mayor.

5.2 OTAS OPTIMIZADOS CON NSGA-II

Por último, en la sección 4.3 en el capítulo 4 se dimensionan los OTAs con el algoritmo evolutivo NSGA-II. Estos valores son reportados en la tabla 4.4 y a continuación serán evaluados en distintas pruebas que verifiquen su eficacia en la emulación de circuitos complejos, para ello, se proponen las aplicaciones del memristor aterrizado de la figura 3.4 y la neurona de picos de la figura 3.3, así como su robustez a la variación PVT (Proceso-Voltaje-Temperatura).

5.2.1 ESQUINAS PVT DE OTAS CON NSGA-II

La validación final de los OTAs se realiza mediante análisis PVT en Cadence-Virtuoso, cuyos resultados se muestran en la figura 5.3. Las soluciones optimizadas mantienen su robustez bajo variaciones de proceso (SS, SF, TT, FS, FF), voltaje y temperatura, como se detalla en la tabla 5.2. Los análisis AC y DC revelan una respuesta en la frecuencia en las esquinas PVT, y valores de la transconductancia consistentes en la región lineal. El caso típico (TT a 60°C, línea roja) es presentado y cuantificado en la tabla 4.4. Estos resultados confirman la eficacia de la metodología para el diseño automatizado de circuitos analógicos en condiciones reales.

Tabla 5.2: Parámetros para el análisis PVT.

Parámetro	Valores
Modelos de Proceso	SS, SF, TT, FS, FF
Temperatura (°C)	-40, 60, 120
Voltaje (V)	1.8±0.18

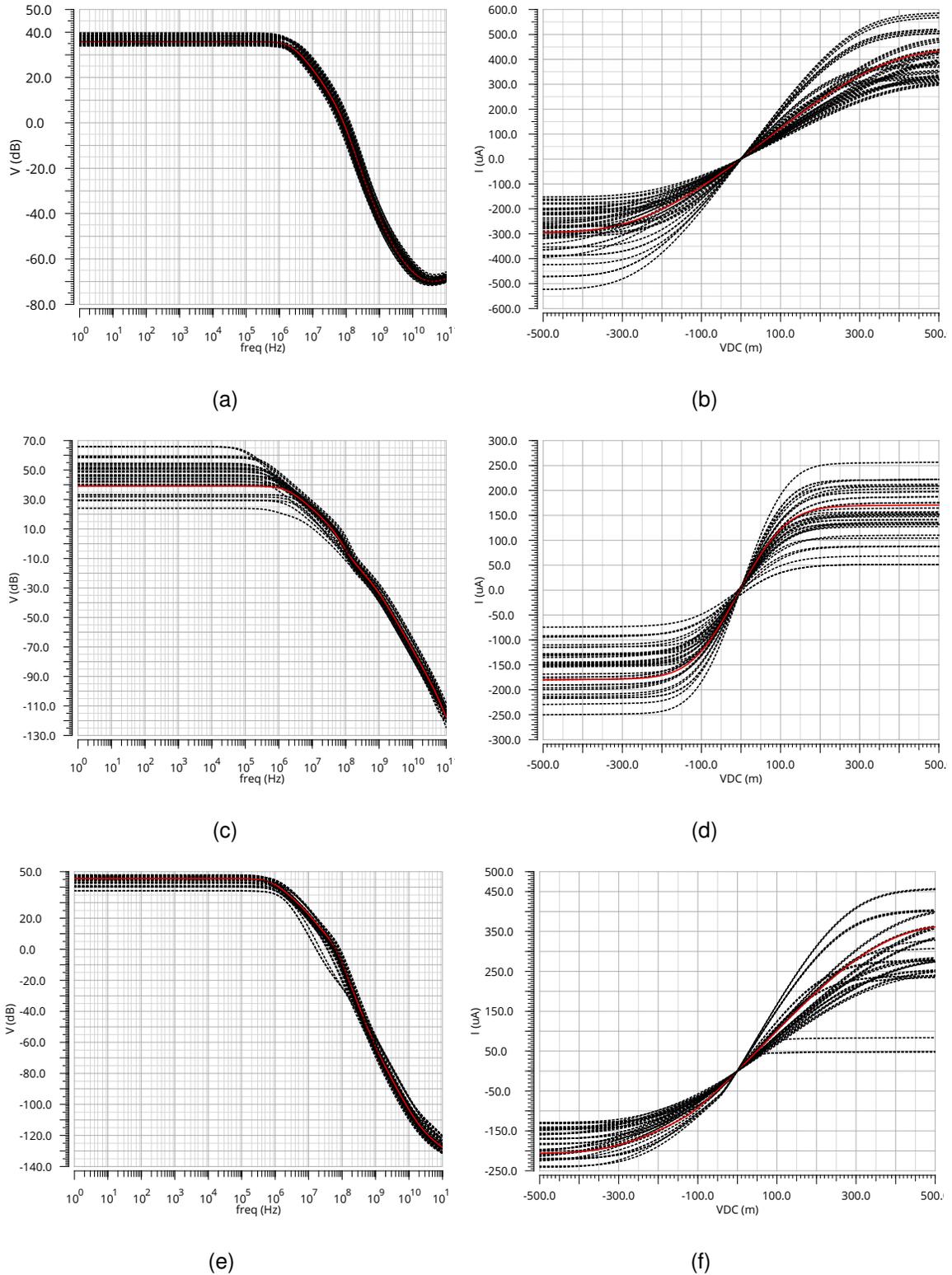


Figura 5.3: Esquinas PVT para las topologías: *Current Mirror OTA*: (a) Análisis AC y (b) Análisis DC. *Folded Cascode OTA*: (c) Análisis AC y (d) Análisis DC. *Low Voltage Current Mirror OTA*: (e) Análisis AC y (f) Análisis DC.

5.2.2 MEMRISTOR BASADO EN OTA CON NSGA-II

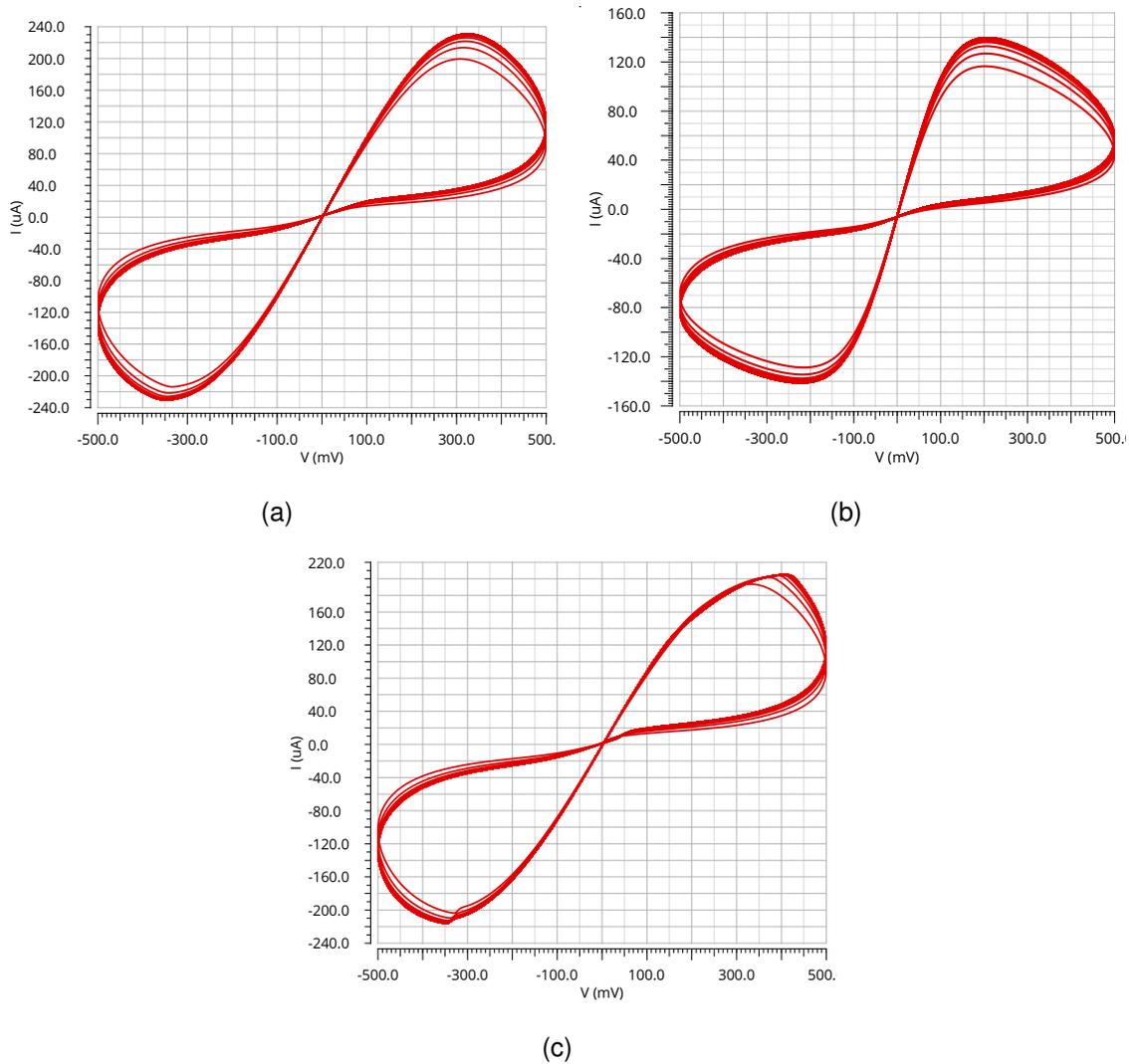


Figura 5.4: Lazos de histéresis del emulador de memristor CMOS optimizados con NSGA-II para las topologías: **(a)** *Current Mirror OTA*. **(b)** *Folded Cascode OTA*. **(c)** *Low Voltage Current Mirror OTA*.

Utilizando las dimensiones especificadas en la tabla 4.4, los valores de capacitancia y dimensiones del inversor CMOS descritos en la sección 3.3 en el capítulo 3, y las condiciones del caso típico (TT a 60°C), se obtienen las curvas de histéresis en el plano $I - V$ mostradas en la figura 5.4, donde se presentan las respuestas de cada topología de OTA para el emulador de memristor presentado en la figura

3.4. Se puede observar que, a diferencia de los lazos en la figura 5.1, estos lazos son pinchados muy cerca del origen y mantienen una simetría importante entre los cuadrantes I y III.

5.2.3 NEURONA DE PICOS BASADA EN MEMRISTOR CON NSGA-II

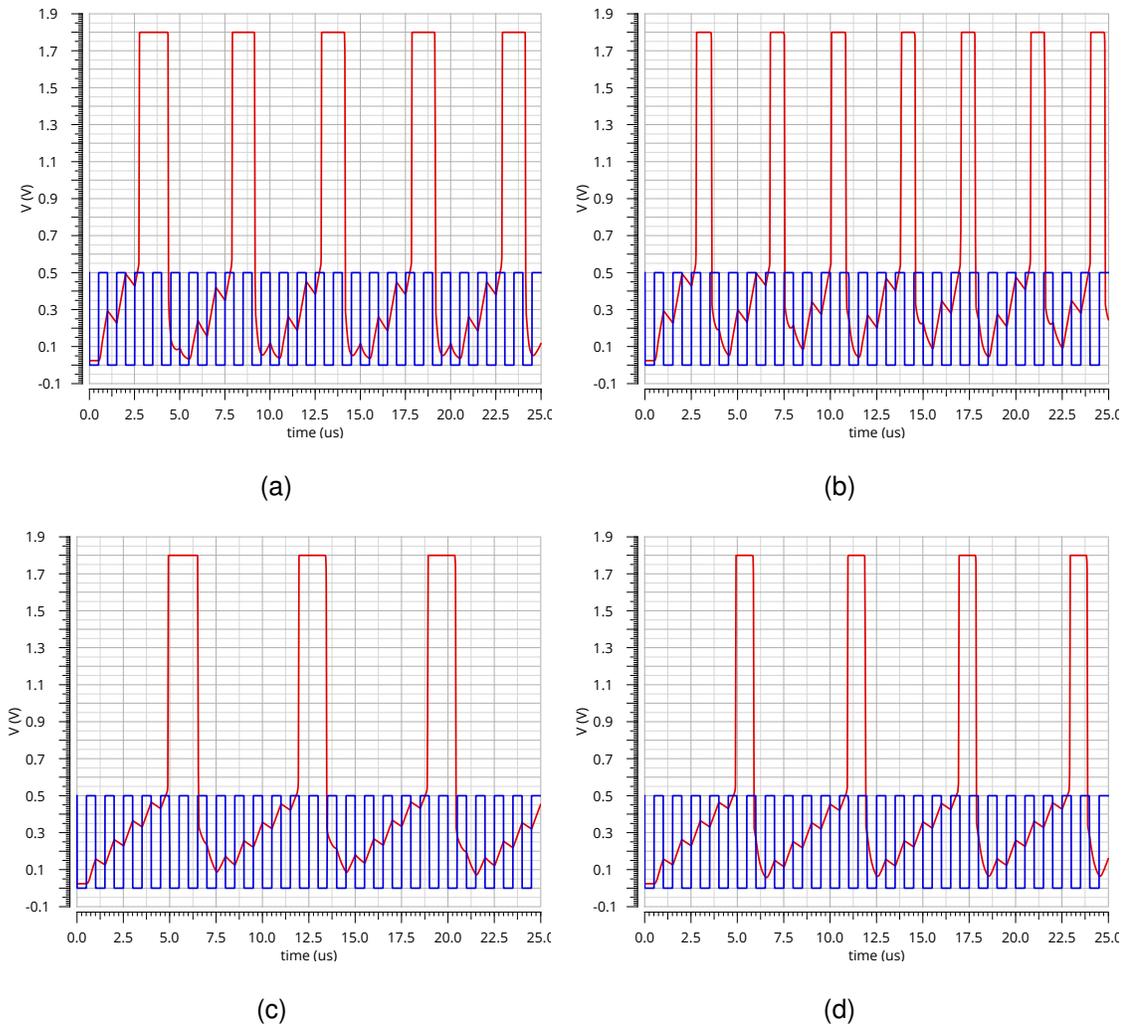


Figura 5.5: Respuesta temporal (rojo) de la neurona de picos (Figura 3.3) con *Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$

Finalmente, para evaluar los OTAs optimizados con el algoritmo evolutivo NSGA-

II, se propone evaluar la respuesta en el tiempo para la neurona LIF en la figura 3.3, recorriendo los valores de alimentación de $\pm 0.9V$ a los valores de $V_{SS} = 0V$ y $V_{DD} = 1.8V$.

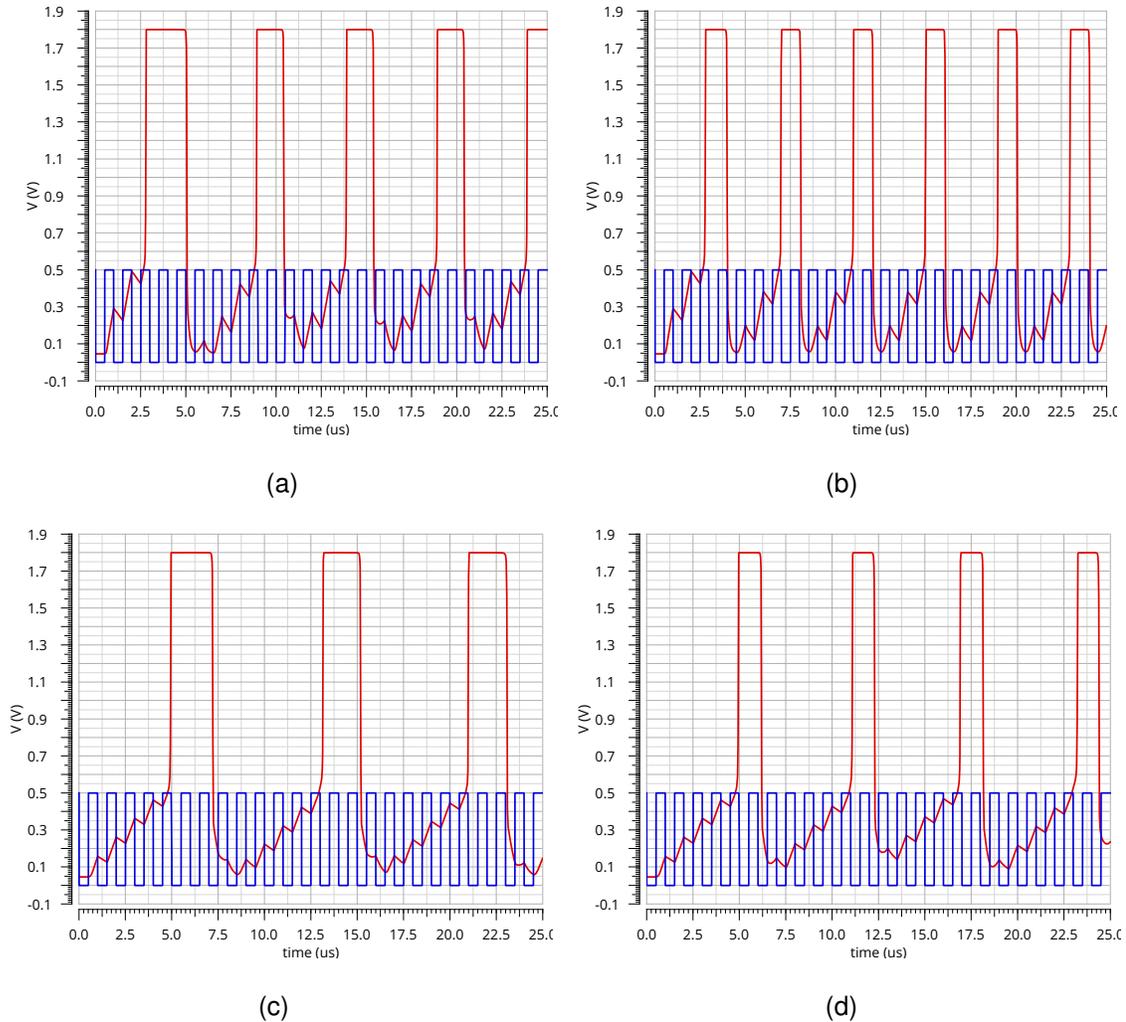


Figura 5.6: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Folded Cascode OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$

Se introduce una señal de entrada cuadrada con amplitud de $0.5V$ y un período $T = 1\mu s$ y se realizan variaciones en las capacitancias de C_{int} y C_m . Las respuestas en el tiempo en las figuras 5.5, 5.6 y 5.7, donde $C_{int} = 10pF$ y $C_m = 1000pF$ tienen un comportamiento similar a lo reportado en la figura 5.2. Un comportamiento

similar se puede apreciar entre las respuestas temporales con los mismos valores de capacitancia. El tiempo de integración es el mismo para todos, el disparo del pico se atrasa para la topología del *Low Voltage Current Mirror OTA* y el tiempo que se mantiene el pico es mayor en la topología *Folded Cascode OTA*.

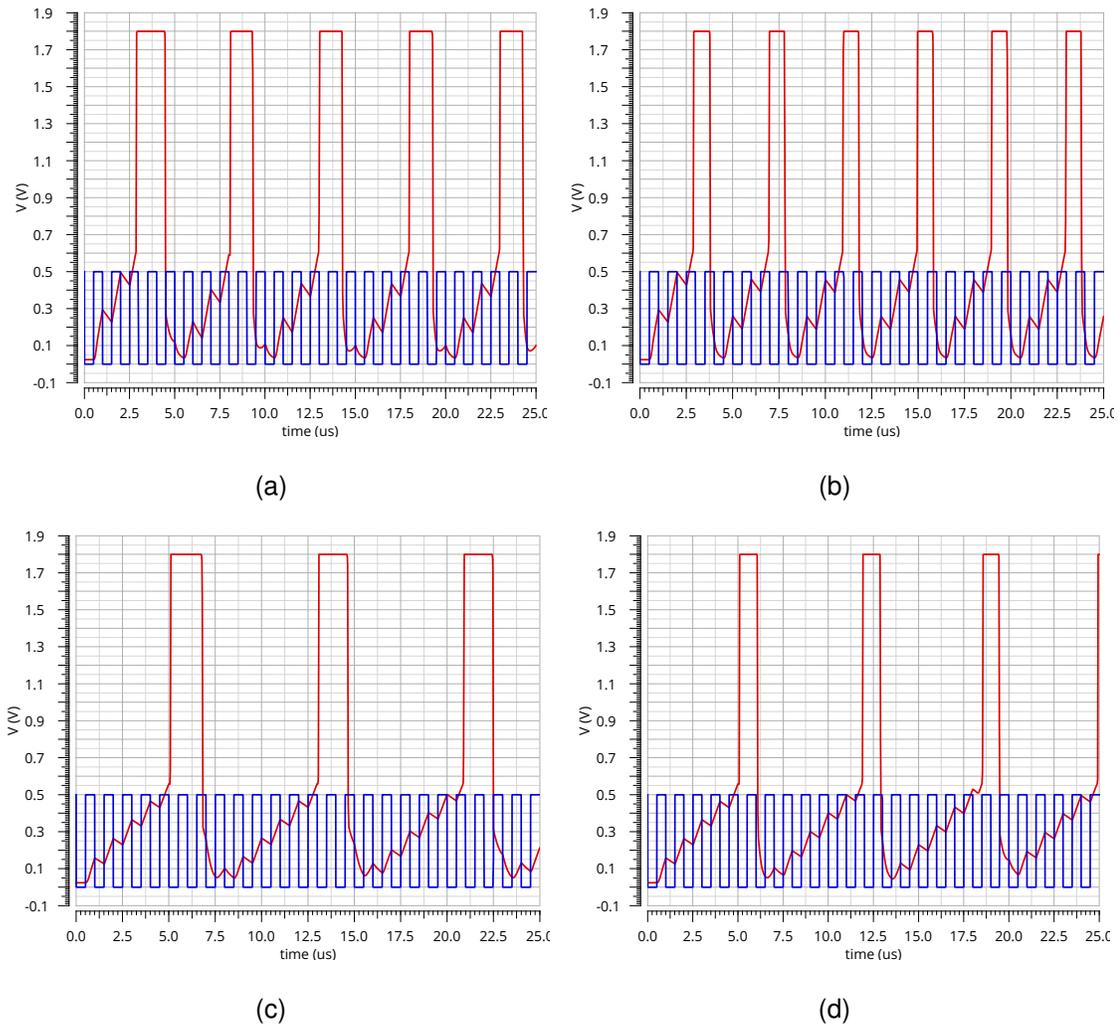


Figura 5.7: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Low Voltage Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 1\mu\text{s}$, azul): **(a)** $C_{int} = 10\text{pF}$ y $C_m = 1000\text{pF}$. **(b)** $C_{int} = 10\text{pF}$ y $C_m = 500\text{pF}$. **(c)** $C_{int} = 20\text{pF}$ y $C_m = 1000\text{pF}$. **(d)** $C_{int} = 20\text{pF}$ y $C_m = 500\text{pF}$

Adicionalmente, se introduce una señal de entrada cuadrada con amplitud de 0.5V y un período $T = 50\mu\text{s}$ y, del mismo modo, se realizan variaciones en las capacitancias de C_{int} y C_m . Durante este período de tiempo, se puede apreciar la

respuesta de los picos generados. En las figuras 5.8, 5.9 y 5.10 se puede apreciar las respuestas en el tiempo para las tres topologías de OTAs optimizadas con NSGA-II.

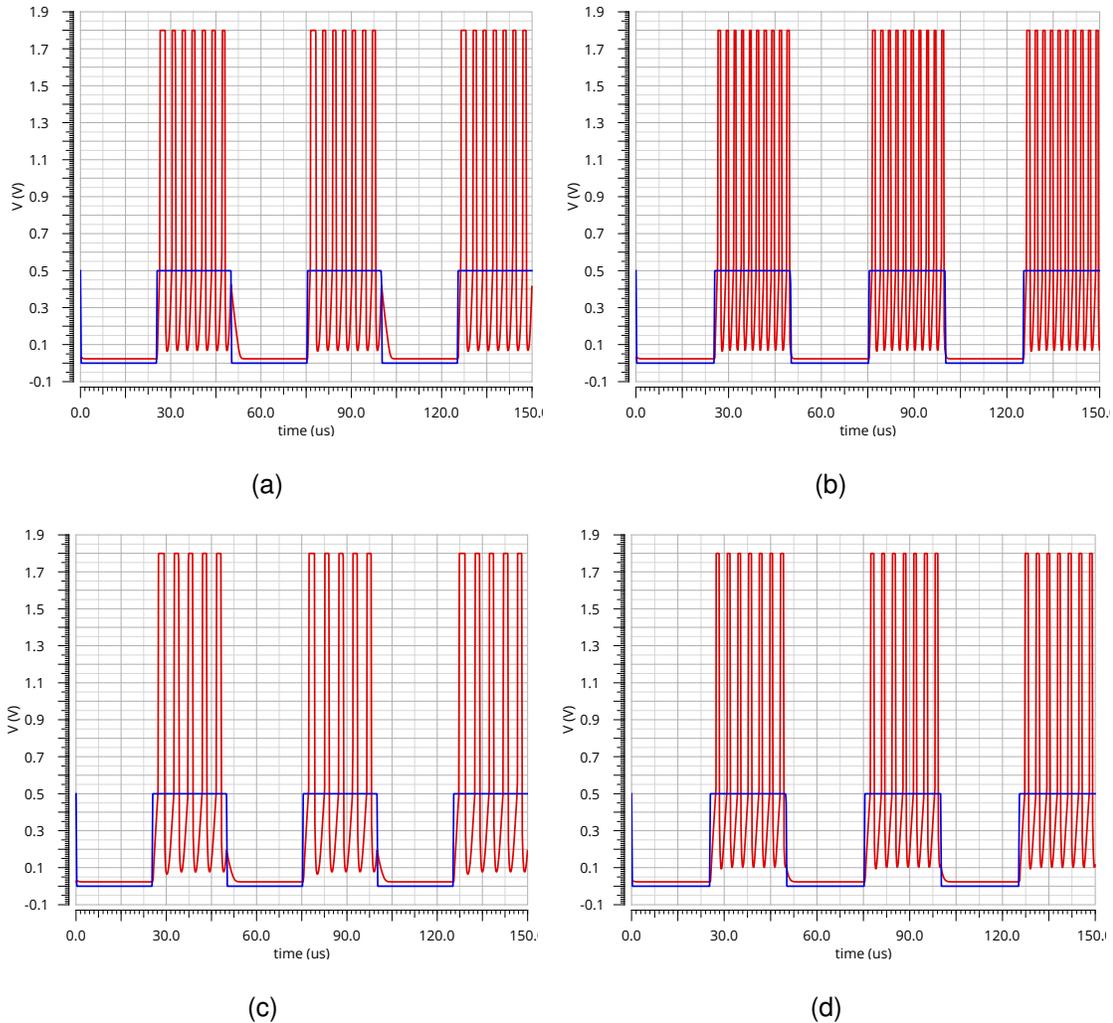


Figura 5.8: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$

De acuerdo con lo anterior, la cantidad de picos que genera la topología *Current Mirror OTA* con $C_{int} = 10pF$ y $C_m = 1000pF$ son siete (figura 5.8(a)), con $C_{int} = 10pF$ y $C_m = 500pF$ son diez (figura 5.8(b)), con $C_{int} = 20pF$ y $C_m = 1000pF$ son cinco (figura 5.8(c)) y con $C_{int} = 20pF$ y $C_m = 500pF$ son siete (figura 5.8(d)).

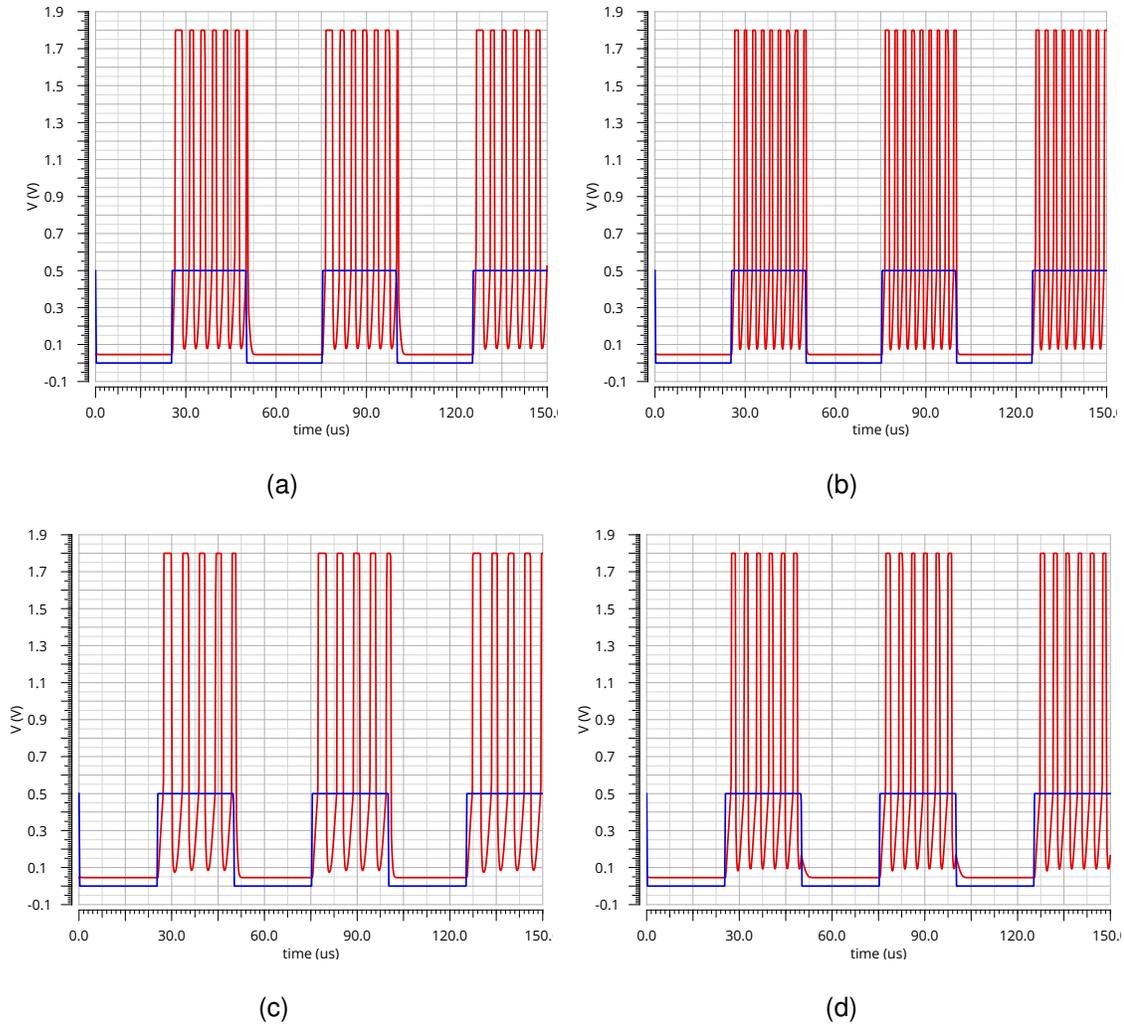


Figura 5.9: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Folded Cascode OTA* para diferentes capacitancias (Señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$

Del mismo modo, la cantidad de picos que genera la topología *Folded Cascode OTA* con $C_{int} = 10pF$ y $C_m = 1000pF$ son siete (figura 5.9(a)), con $C_{int} = 10pF$ y $C_m = 500pF$ son nueve (figura 5.9(b)), con $C_{int} = 20pF$ y $C_m = 1000pF$ son cinco (figura 5.9(c)) y con $C_{int} = 20pF$ y $C_m = 500pF$ son seis (figura 5.9(d)).

Finalmente, la cantidad de picos que genera la topología *Low Voltage Current Mirror OTA* con $C_{int} = 10pF$ y $C_m = 1000pF$ son siete (figura 5.10(a)), con $C_{int} =$

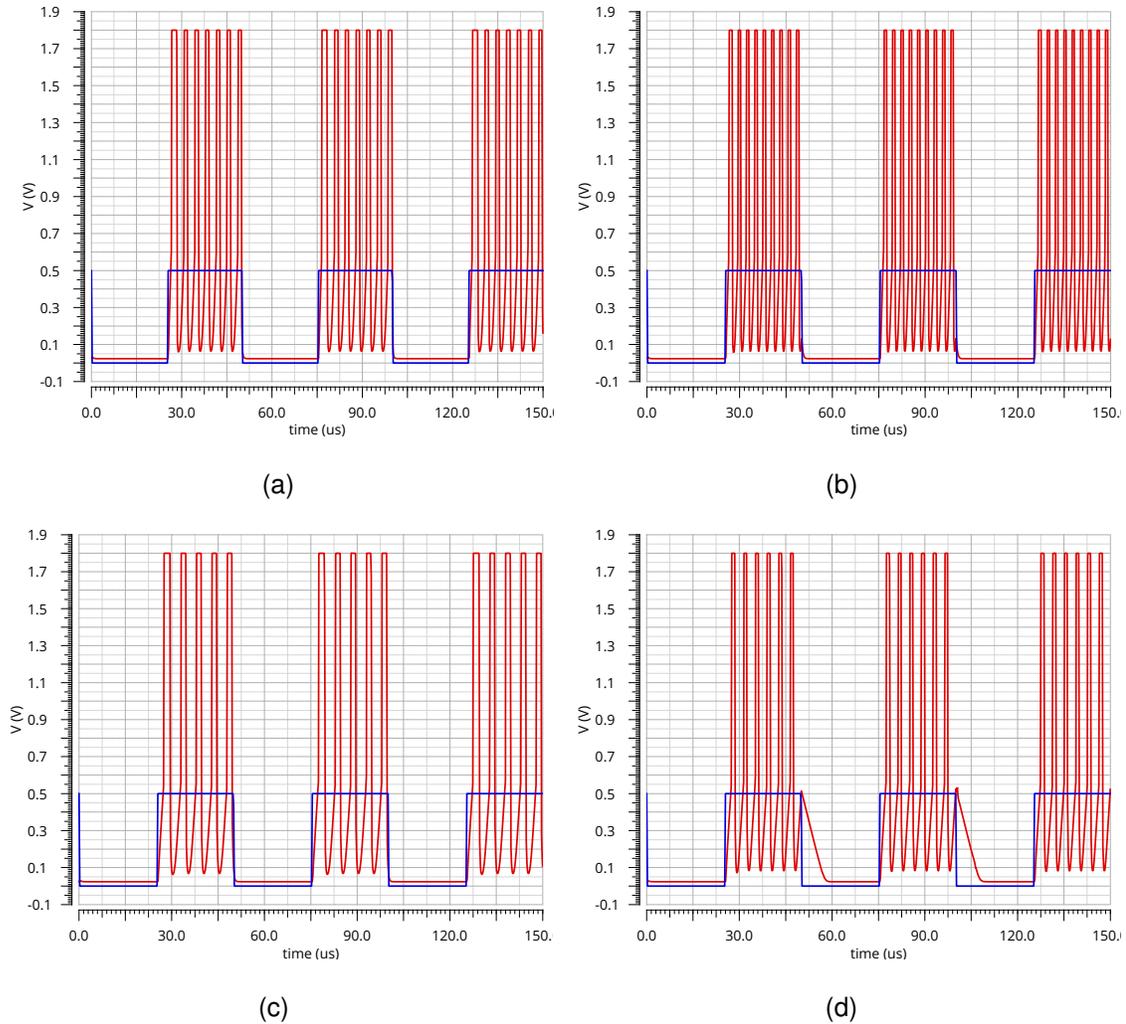


Figura 5.10: Respuesta temporal (rojo) de la neurona de picos (figura 3.3) con *Low Voltage Current Mirror OTA* para diferentes capacitancias (señal de entrada: $T = 50\mu s$, azul): **(a)** $C_{int} = 10pF$ y $C_m = 1000pF$. **(b)** $C_{int} = 10pF$ y $C_m = 500pF$. **(c)** $C_{int} = 20pF$ y $C_m = 1000pF$. **(d)** $C_{int} = 20pF$ y $C_m = 500pF$

$10pF$ y $C_m = 500pF$ son nueve (figura 5.10(b)), con $C_{int} = 20pF$ y $C_m = 1000pF$ son cinco (figura 5.10(c)) y con $C_{int} = 20pF$ y $C_m = 500pF$ son seis (figura 5.10(d)). De esta forma, se observa que las topologías *Folded Cascode OTA* y *Low Voltage Current Mirror OTA* generan el mismo número de picos durante el período $T = 10\mu s$, esto se debe a que en la optimización con NSGA-II alcanzaron una transconductancia (G_m) similar, a diferencia de la topología *Current Mirror OTA*. Esta diferencia

de G_m afecta directamente al tiempo de disparo y que se mantiene el pico debido al módulo de reset memristivo (MRE).

5.3 CONCLUSIONES

En este capítulo se comparó el dimensionamiento con el método gm/I_D puro y utilizando el algoritmo evolutivo NSGA-II. Se evalúan los tamaños de los transistores obtenidos en el capítulo 4 en aplicaciones con circuitos complejos, como el emulador de memristor del capítulo 3 y en consecuencia, la neurona de picos basada en memristor del capítulo 2. Esta comparación favorece directamente a los valores obtenidos con NSGA-II en los lazos de histéresis del memristor, pero obteniendo un comportamiento similar en la neurona de picos, únicamente diferenciado por el valor de transconductancia en la topología OTA, que afecta directamente al tiempo de disparo del pico y que se mantiene el mismo.

CAPÍTULO 6

CONCLUSIONES GENERALES Y TRABAJO FUTURO

En este trabajo de Tesis se describió el diseño y optimización de una neurona de picos utilizando memristores y amplificadores operacionales de transconductancia (OTA). A lo largo de los capítulos se desarrolla el marco teórico y práctico que integra modelos neuronales, dispositivos memristivos y técnicas de optimización evolutiva.

Primeramente, se introducen las SNNs, centrándose en el modelo *leaky integrate-and-fire* (LIF) por su eficiencia y simplicidad para emular comportamientos neuronales básicos. Se explora el uso de memristores en SNNs, proponiendo un emulador basado en OTAs para generar *spikes* y garantizar su dinámica.

Posteriormente, se optimizan tres topologías de OTA mediante el método gm/I_D combinado con el algoritmo NSGA-II, ajustando parámetros como la transconductancia (G_m) para su implementación en sistemas complejos.

Finalmente, se validan los OTAs optimizados mediante simulaciones, comparando resultados con el método tradicional y analizando su robustez frente a variaciones (análisis PVT). Además, se verifica su compatibilidad con el memristor y la neurona LIF previamente diseñados.

El trabajo logra una integración efectiva entre modelos bioinspirados (SNNs), dispositivos emergentes (memristores) y técnicas de optimización avanzada (NSGA-II), demostrando que la combinación de métodos analíticos (gm/I_D) y evolutivos mejora el desempeño de circuitos analógicos para aplicaciones neuromórficas. Los resultados sientan bases para futuros desarrollos en hardware de redes neuronales eficientes y adaptables.

A continuación, se mencionan algunas propuestas de trabajos futuros que podrían derivarse de esta investigación:

- Desarrollar una red neuronal de picos basada en dispositivos con memoria (como el memristor), tomando como base la neurona reportada en el desarrollo de esta Tesis.
- Comparar modelos de neuronas de picos en implementaciones analógica/digital con soluciones basadas en tecnología CMOS para aplicaciones de clasificación.
- Optimizar modelos de dispositivos activos o amplificadores tradicionales en tecnología CMOS mediante algoritmos evolutivos.

BIBLIOGRAFÍA

- [1] Yu-chen Wu and Jun-wen Feng. Development and application of artificial neural network. *WIRELESS PERSONAL COMMUNICATIONS*, 102(2):1645–1656, SEP 2018.
- [2] Xabier Basogain Olabe. *Redes Neuronales Artificiales y sus Aplicaciones*. Escuela de Ingenieros, 1998.
- [3] Yuchun Wu. Application of artificial neural network in communication signal processing. *AGRO FOOD INDUSTRY HI-TECH*, 28(3):1920–1924, MAY-JUN 2017.
- [4] Pranab Dey. Artificial neural network in diagnostic cytology. *CYTOJOURNAL*, 19, APR 2 2022.
- [5] Chi-Ching Tsang, Chenyang Zhao, Yueh Liu, Ken P. K. Lin, James Y. M. Tang, Kar-On Cheng, Franklin W. N. Chow, Weiming Yao, Ka-Fai Chan, Sharon N. L. Poon, Kelly Y. C. Wong, Lianyi Zhou, Oscar T. N. Mak, Jeremy C. Y. Lee, Suhui Zhao, Antonio H. Y. Ngan, Alan K. L. Wu, Kitty S. C. Fung, Tak-Lun Que, Jade L. L. Teng, Dirk Schnieders, Siu-Ming Yiu, Susanna K. P. Lau, and Patrick C. Y. Woo. Automatic identification of clinically important aspergillus species by artificial intelligence-based image recognition: proof-of-concept study. *EMERGING MICROBES & INFECTIONS*, 14(1), DEC 31 2025.
- [6] Keerthi Nalliboyina and Sakthivel Ramachandran. An energy-efficient hybrid cmos spiking neuron circuit design with a memristive based novel t-type arti-

- cial synapse. *AEU - International Journal of Electronics and Communications*, 173:154982, 2024.
- [7] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.
- [8] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain Sciences*, 12(7), 2022.
- [9] Zhikui Duan, Xin Zhang, Shaobo He, Xinmei Yu, Peng Xiong, Jiahui Chen, and Qiang Wang. Implementation of a fully integrated memristive chua’s chaotic circuit with a voltage-controlled oscillator. *Integration*, 99:102258, 2024.
- [10] Usha Kumari and Rekha Yadav. Secure and reliable communication using memristor-based chaotic circuit. *ANALOG INTEGRATED CIRCUITS AND SIGNAL PROCESSING*, 120(2-3):155–171, SEP 2024.
- [11] Zhangzhi Zhou, Mi Lin, Xuanxuan Zhou, and Chong Zhang. Implementation of memristive emotion associative learning circuit. *COGNITIVE NEURODYNAMICS*, 19(1), DEC 2025.
- [12] Zhipeng Xia, Xiao Sun, Zhenlong Wang, Jialin Meng, Boyan Jin, and Tianyu Wang. Low-power memristor for neuromorphic computing: From materials to applications. *NANO-MICRO LETTERS*, 17(1), DEC 2025.
- [13] Miao Hu, Yiran Chen, J. Joshua Yang, Yu Wang, and Hai Helen Li. A compact memristor-based dynamic synapse for spiking neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(8):1353–1366, 2017.
- [14] Pulak Mondal, Subhasish Banerjee, Mourina Ghosh, Ankur Singh, and Santosh Kumar. Design of a vdba-based memristor emulator and its application for bio-sensing through instrument amplifier. *IEEE OPEN JOURNAL OF NANOTECHNOLOGY*, 6:35–43, 2025.

- [15] John Vista and Ashish Ranjan. Flux controlled floating memristor employing vcta: Incremental or decremental operation. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 40(2):364–372, February 2021.
- [16] Sagar Surendra Prasad, Prashant Kumar, and Rajeev Kumar Ranjan. Resistorless memristor emulator using cfta and its experimental verification. *IEEE Access*, 9:64065–64075, 2021.
- [17] Vipin Kumar Sharma, Mohd. Samar Ansari, and T. Parveen. Tunable memristor emulator using off-the-shelf components. *Procedia Computer Science*, 171:1064–1073, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).
- [18] Rajeev Ranjan, Nishtha Rani, Ratnadeep Pal, Sajal K. Paul, and Gaurav Kanyal. Single ccta based high frequency floating and grounded type of incremental/decremental memristor emulator and its application. *Microelectronics Journal*, 60:119–128, 02 2017.
- [19] Sagar Surendra Prasad, Prashant Kumar, Niranjan Raj, Pankaj Kumar Sharma, Bindu Priyadarshini, Rajeev Kumar Ranjan, and Pipat Prommee. A compact floating and grounded memristor model using single active element. *AEU - International Journal of Electronics and Communications*, 157:154426, 2022.
- [20] Sagar, Niranjan Raj, Vijay Kumar Verma, and Rajeev Kumar Ranjan. Electronically tunable flux-controlled resistorless memristor emulator. *IEEE Canadian Journal of Electrical and Computer Engineering*, 45(3):311–317, 2022.
- [21] Gaurav Kanyal, Pratik Kumar, Sajal K. Paul, and Ashok Kumar. Ota based high frequency tunable resistorless grounded and floating memristor emulators. *AEU - International Journal of Electronics and Communications*, 92:124–145, 2018.

- [22] Suchitra Gupta and Shireesh Kumar Rai. New grounded and floating decremental/incremental memristor emulators based on cdda and its application. *Wireless Personal Communications*, 113, 07 2020.
- [23] Mourina Ghosh, Pulak Mondal, Shekhar S. Borah, and Santosh Kumar. Resistorless memristor emulators: Floating and grounded using ota and vdba for high-frequency applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(3):978–986, 2023.
- [24] Sagar, Rajeev Kr. Ranjan, and Sung-Mo Kang. Resistorless floating/grounded memristor emulator model with electronic tunability. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(7):2340–2344, 2023.
- [25] Carlos Sanchez-Lopez, Victor Carbajal-Gómez, Miguel Carrasco-Aguilar, and Illiani Carro. Fractional-order memristor emulator circuits. *Complexity*, 2018, 05 2018.
- [26] Prashant Kumar, Rajeev Kumar Ranjan, and Sung-Mo Kang. A memristor emulation in 180-nm cmos process for spiking signal generation and chaos application. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(4):1757–1770, 2024.
- [27] Paul G. A. Jespers and Boris Murmann. *Systematic Design of Analog CMOS Circuits: Using Pre-Computed Lookup Tables*. Cambridge University Press, 2017.
- [28] Chaiyanut Aueamnuay, Ajmal Vadakkan Kayyil, Narayana Bhagirath Thota, Praveen Kumar Venkatachala, and David J. Allstot. gm/id-based frequency compensation of cmos two-stage operational amplifiers. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.
- [29] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

- [30] Abdelaziz Lberni, Amin Sallem, Malika Alami Marktani, Nouri Masmoudi, Abdelaziz Ahaitouf, and Ali Ahaitouf. Influence of the operating regimes of mos transistors on the sizing and optimization of cmos analog integrated circuits. *AEU-INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATIONS*, 143, JAN 2022.
- [31] João D. Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S. Cardoso. Spiking neural networks: A survey. *IEEE Access*, 10:60738–60764, 2022.
- [32] Luigi Fortuna and Arturo Buscarino. Spiking neuron mathematical models: A compact overview. *Bioengineering*, 10(2), 2023.
- [33] Li-Ye Niu, Ying Wei, Wen-Bo Liu, Jun-Yu Long, and Tian-hao Xue. Research progress of spiking neural network in image classification: a review. *Applied Intelligence*, 53(16):19466–19490, March 2023.
- [34] Wulfram Gerstner, Wulfram, Kistler, and Werner M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. 08 2002.
- [35] Umayia Mushtaq, Md. Waseem Akram, Dinesh Prasad, and Aminul Islam. An energy and area-efficient spike frequency adaptable lif neuron for spiking neural networks. *Computers and Electrical Engineering*, 119:109562, 2024.
- [36] E.M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.
- [37] Arati Kumari Shah, Eou-Sik Cho, Jisun Park, Hyungsoon Shin, and Seong-jae Cho. A compact integrate-and-fire neuron circuit embedding operational transconductance amplifier for fidelity enhancement. *IEEE Access*, 11:53932–53938, 2023.
- [38] Huihui Peng, Lin Gan, and Xin Guo. Memristor-based spiking neural networks: cooperative development of neural network architecture/algorithms and memristors. *Chip*, 3(2):100093, 2024.

- [39] L. Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.
- [40] Dmitri Strukov, Gregory Snider, Duncan Stewart, and Stan Williams. The missing memristor found. *Nature*, 453:80–3, 06 2008.
- [41] Deepthi M.S., Shashidhara H.R., Jayaramu Raghu, and Rudraswamy S.B. Simulation-based effective comparative analysis of neuron circuits for neuro-morphic computation systems. *Neurocomputing*, 614:128758, 2025.
- [42] Andrzej Handkiewicz, Grzegorz Krzywoszyja, Wojciech Zajac, and Mariusz Naumowicz. Filter design based on multi-port and multi-dimensional gc circuits. *CIRCUITS SYSTEMS AND SIGNAL PROCESSING*, 2025 FEB 22 2025.
- [43] Thanat Nonthaputha, Montree Kumngern, and Nawaphol Thepnarin. A simple and accurate cmos sample-and-hold circuit using dual output-ota. *PRZEGLAD ELEKTROTECHNICZNY*, 96(10):163–166, 2020.
- [44] Yuhua Liang, Ruiwen Liu, and Zhangming Zhu. A 16 khz-bw 93.1 db-sndr dt 2-1 mash adc with cascading-cls technique for accelerometer measurement applications. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 74, 2025.
- [45] Seunghwa Shin, Gyeong-Gu Kang, Gyu-Wan Lim, and Hyun-Sik Kim. A display source-driver ic featuring multistage-cascaded 10-bit dac and true-dc-interpolative super-ota buffer. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 59(4):1050–1066, APR 2024.
- [46] Faraz Hashmi, M. Nizamuddin, and Syed Umar Amin. Performance evaluation and optimization of triple cascode operational transconductance amplifiers using gnrfet technology for low power smart devices. *MATERIALS RESEARCH EXPRESS*, 12(2), FEB 1 2025.
- [47] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill Higher Education, 2016.

-
- [48] E. Tlelo-Cuautle and A.C. Sanabria-Borbon. Optimising operational amplifiers by evolutionary algorithms and gm/id method. *International Journal of Electronics*, 103(10):1665–1684, 2016.
- [49] Sachchida Nand Singh, Geetika Srivastava, Syed Shamroz Arshad, and Sachchidanand Shukla. A cmos low pass filter based on improved current mirror for biomedical application. *JOURNAL OF OPTICS-INDIA*, 54(1, SI):31–38, FEB 2025.
- [50] Farid Bautista, Sergio Martínez, Graciano Dieck, and Olivier Rossetto. An ultra-low voltage high gain operational transconductance amplifier for biomedical applications. 2007.
- [51] Sougata Ghosh and Vijaya Bhadauria. An ultra-low-power bulk-driven subthreshold super class-ab rail-to-rail cmos ota with enhanced small and large signal performance suitable for large capacitive loads. *Microelectronics Journal*, 115:105208, 2021.
- [52] Remya Jayachandran, Edet Bijoy Kumaradhas, and Arun S. Design of second order low pass filter using inverter-based cmos operational transconductance amplifier (ota) through gm/id methodology. In *2024 IEEE International Conference of Electron Devices Society Kolkata Chapter (EDKCON)*, pages 519–524, 2024.