



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SECCIÓN DE ELECTRÓNICA DEL ESTADO SÓLIDO

**Aproximación de respuesta no lineal y dinámica de un memristor en
tecnología digital**

TESIS

Que presenta:

M. en C. GERARDO MARCOS TORNEZ XAVIER

Para obtener el grado de

DOCTOR EN CIENCIAS

EN LA ESPECIALIDAD DE INGENIERÍA ELÉCTRICA

Directores de la Tesis:

DR. FELIPE GÓMEZ CASTAÑEDA

DR. JOSÉ ANTONIO MORENO CADENAS

Ciudad de México

Octubre, 2019

Agradecimientos

A mi familia: Gerardo Tornez Garibo, Osmar Xavier Pastrana, Geraldine Esmerada Tornez Xavier, por el gran apoyo y cariño que me han brindado a lo largo de mi vida, ya que sin su amor y consejos no sería la persona que soy hoy en día, a ellos con todo mi amor.

A mis asesores de tesis:

Dr. Felipe Gómez Castañeda y Dr. José Antonio Moreno Cadenas, por su valiosa guía y enseñanza, brindada para la realización de esta tesis.

A mis revisores y sinodales:

Agradezco el trabajo de: Dr. Alejandro Ávila García (CINVESTAV-SEES), Dr. Ramón Peña Sierra (CINVESTAV-SEES), Dr. Víctor Hugo Ponce Ponce (Centro de Investigación en Computación-IPN, CIC-IPN), Dr. Herón Molina Lozano (Centro de Investigación en Computación-IPN, CIC-IPN), por el tiempo dedicado en la revisión de esta tesis; por sus valiosos comentarios y sugerencias.

Un especial agradecimiento al Dr. Mario Alfredo Reyes Barranca, M. en C. Luis Martín Flores Nava, Dr. Oliverio Arellano Cárdenas y M. en C. Emilio Rafael Espinosa García por su asesoría técnica brindada durante mi estancia académica de doctorado en el CINVESTAV. Principalmente, por la confianza y amistad mostradas hacia mi persona.

Al personal administrativo de la Sección: Yesenia Cervantes y Beatriz Urrutia por su apoyo y dedicación.

Un especial agradecimiento al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca económica otorgada para la realización de este trabajo.

Agradezco al CINVESTAV, por la educación recibida.

Reflexión de un científico

Yo he vivido siempre bien contento como investigador científico porque nunca me interesó hacer mucho dinero... Siendo muy joven me encontré con la investigación científica, me encantó y he trabajado en ella toda mi larga vida. Mi trabajo es estimulante, divertido, muy variable, siempre hago lo que me gusta, no tengo jefe ni horario de trabajo, nunca he estado aburrido en mi laboratorio, hago el mejor uso que puedo de mi cerebro, no dejo que me tomen el pelo ni los comerciantes ni los políticos, casi todos mis amigos son científicos y hablo con ellos en su mismo lenguaje, y he sido profesor de muchos científicos, no solo mexicanos, sino de otros países. Y por eso es por lo que siempre estoy bien contento.

Dr. Ruy Pérez Tamayo

CONTENIDO

Contenido	i
Resumen	v
Abstract	vi
Hipótesis	vii
Objetivos	vii
Organización	viii
Capítulo 1 Introducción	1
1.1 Introducción.....	1
Referencias.....	3
Capítulo 2 Aspectos teóricos	6
2.1 Introducción.....	6
2.2 El memristor ideal.....	7
2.2.1 Propiedades del memristor.....	10
2.3 Sistemas memristivos.....	11
2.4 Memristor de Hewlett – Packard.....	14
2.5 Modelo lineal.....	17
2.6 Modelo lineal con función de ventana.....	19
2.7 Modelo no lineal.....	23
2.8 Modelo de Pickett.....	25
2.9 Modelo de umbral adaptable TEAM.....	28

2.10 Modelo de umbral de voltaje adaptable VTEAM.....	30
2.11 Resumen.....	32
Referencias.....	33
Capítulo 3 Red neuronal recurrente memristiva.....	37
3.1 Introducción.....	37
3.2 Problema de Asignación.....	39
3.2.1 Red neuronal de Wang para el problema de asignación.....	40
3.3 Problema de Transporte.....	42
3.3.1 Red neuronal de Wang para el problema de transporte.....	44
3.4 Configuración eléctrica de una neurona.....	45
3.5 El memristor.....	47
3.6 Circuitos CMOS analógicos.....	49
3.6.1 Amplificador de corriente.....	49
3.6.2 Función sigmoideal invertida.....	50
3.6.3 Resistor de valor constante.....	51
3.6.4 Fuente de corriente constante.....	52
3.6.5 Fuente de corriente dinámica.....	56
3.7 Solución analógica del problema de asignación.....	59
3.8 Solución analógica del problema de transporte.....	63
3.9 Conclusiones.....	67
Referencias.....	68

Capítulo 4 Optimizador memristivo para la asignación de tareas.....71

4.1 Introducción.....71

4.2 Asignación de tareas.....72

4.3 Compuerta lógica.....72

4.4 Neurona memristiva.....74

4.5 Proceso de programación de los memristores.....75

4.6 Simulaciones eléctricas.....78

4.7 Conclusiones.....83

Referencias.....84

Capítulo 5 Implementación digital.....85

5.1 Introducción.....85

5.2 Modelo lineal con función de ventana.....87

5.2.1 Ecuación de estado.....88

5.2.2 Resistencia.....89

5.2.3 Funciones de ventana.....90

5.2.4 Potencia.....91

5.2.5 Integrador.....92

5.2.6 Seleccionador.....92

5.2.7 Diagrama completo.....93

5.2.8 Generación del módulo para Co-simulación.....94

5.2.9 Simulaciones eléctricas.....97

5.3	Modelo TEAM.....	100
5.3.1	Control.....	102
5.3.2	Limitador.....	103
5.3.3	Ecuación de estado.....	103
5.3.4	Función de ventana.....	105
5.3.5	Resistencia.....	106
5.3.6	Diagrama general.....	107
5.3.7	Simulaciones eléctricas.....	107
5.4	Modelo VTEAM.....	111
5.4.1	Simulaciones eléctricas.....	112
5.5	Conclusiones.....	115
	Referencias.....	115
	Conclusiones generales.....	117
	Trabajo futuro.....	118
	Apéndice A. Ajuste de parámetros.....	119
	Apéndice B. Desarrollo problema de asignación.....	127
	Apéndice C. Desarrollo problema de transporte.....	134
	Apéndice D. Código TEAM Verilog-A.....	141
	Apéndice E. Código algoritmo húngaro.....	143
	Anexos.....	146

Resumen

Por casi dos siglos toda la teoría de sistemas eléctricos conocida se basó en tres elementos pasivos fundamentales de circuitos: el resistor, el capacitor y el inductor. Pero en 1971, el profesor Leon Chua desafió a la bien establecida percepción de la teoría clásica, proponiendo la existencia de un cuarto elemento fundamental, al cual denominó Memristor (resistor con memoria). El Memristor opera como un resistor variable que, al suprimirse su polarización, mantiene su último valor óhmico, es decir, posee características de memoria no volátil.

Debido al gran interés que ha generado el memristor desde su fabricación por los laboratorios Hewlett-Packard en el año 2008, el mundo académico y el industrial se han dado a la tarea de encontrar nuevos materiales y procedimientos para su fabricación, así como a la realización de modelos matemáticos capaces de representar los fenómenos físicos involucrados en el dispositivo. Esto, con la finalidad de emplear dichos modelos en el análisis, diseño y simulación de nuevos circuitos emergentes para aplicaciones analógicas y digitales, las cuales se piensa que en un futuro puedan revolucionar la industria electrónica.

Basado en lo anterior, este trabajo de tesis se enfoca principalmente en el análisis y estudio de los modelos más notables encontrados en la literatura, que describen el comportamiento dinámico no lineal del memristor, así como el uso de dichos modelos en aplicaciones analógicas y digitales en el campo de las redes neuronales para la solución a problemas de optimización, pertenecientes al área de la programación lineal. Se presenta también la reproducción del comportamiento dinámico y no lineal del memristor, usando una plataforma de desarrollo de sistemas digitales FPGA.

Abstract

For almost two centuries all known electronics were based on three fundamental passive elements of circuits: resistor, capacitor and inductor. But in 1971 Professor Leon Chua challenged the well-established perception of classical electronics by proposing the existence of a fourth fundamental element, which he called memristor (resistor with memory). The memristor operates as a variable resistor that, by suppressing its polarization, maintains its last ohmic value, that is, it has non-volatile memory.

Due to the great interest that the memristor has generated since its manufacture by the Hewlett-Packard laboratories in 2008, the academic and industrial world have been given the task of finding new materials and procedure for its manufacture, as well as the realization of mathematical models capable of representing the physical phenomena involved in the device. This in order to use these models in the analysis, design and simulation of new emerging circuits for analog and digital applications, which in the future can revolutionize the electronics industry.

Based on the above, this thesis work focuses mainly on the analysis and study of the most notable models found in the literature, which describe the non-linear dynamic behavior of the memristor, as well as the use of such models in analog and digital applications in the field of neural networks for the solution to optimization problems belonging to the area of linear programming. The reproduction of the dynamic and non-linear behavior of the memristor using a FPGA digital systems development platform is also presented.

Hipótesis

Es posible usar un recurso computacional programable para la emulación de un memristor en sustitución de un memristor físico real.

Objetivos

Los objetivos de este trabajo están basados en la aplicación de un modelo avanzado del memristor bajo dos enfoques:

1. Desarrollo de un prototipo digital del memristor en tecnología FPGA.

Creación de una componente digital de un sistema memristivo fundamental, como vehículo experimental alternativo a los existentes por medios tecnológicos tradicionales, empleando una metodología de co-simulación de sistemas eléctricos no lineales, a través de la interacción entre el memristor digital y el sistema de cómputo personal o PC.

2. Uso del modelo en redes neuronales.

Demostración del uso de redes neuronales memristivas en la solución de problemas de optimización, se usan los modelos recurrentes de sistemas neuronales del tipo Hopfield. Específicamente los paradigmas de optimización combinatoria de asignación y transporte.

Organización

La estructura de la tesis es la siguiente:

Capítulo 1: Introducción

Se presenta una breve introducción del memristor y su importancia en la industria electrónica; además, se presentan los motivos y objetivos para el desarrollo de este trabajo de investigación. Finalmente, se presenta una descripción de la organización de la tesis.

Capítulo 2: Aspectos teóricos

Se exploran los aspectos teóricos del memristor y los sistemas memristivos, así como el análisis de algunos modelos más notables presentes en la literatura. Este capítulo sirve como base para entender al memristor, el cual será utilizado en los capítulos subsecuentes.

Capítulo 3: Red neuronal recurrente memristiva

Se presenta el desarrollo de una red neuronal recurrente memristiva, la cual combina circuitos híbridos memristor/CMOS. Esta red es capaz de resolver dos problemas de optimización denominados: asignación y transporte.

Capítulo 4: Optimizador memristivo

Se muestra el desarrollo de un optimizador memristivo, el cual se basa en un arreglo de compuertas NOR. Este arreglo se asemeja a una red neuronal recurrente, la cual es capaz de resolver el problema de asignación de tareas.

Capítulo 5: Implementación digital

Se realiza la implementación digital en un dispositivo digital programable FPGA de los modelos no lineal con función de ventana, el modelo de umbral adaptable TEAM y el modelo de umbral de voltaje adaptable VTEAM. La descripción de

dichos modelos se llevó a cabo con la herramienta de diseño System Generator, utilizando una tarjeta de desarrollo Nexys 4 DDR.

Apéndice A

Se muestra un algoritmo en MATLAB para el ajuste de parámetros del modelo de umbral de voltaje adaptable VTEAM, el cual es un modelo general capaz de emular el comportamiento de memristores medidos experimentalmente y de otros modelos propuestos.

Apéndice B

Se presenta el desarrollo matemático de las expresiones que definen el problema de asignación, así como el desarrollo de la ecuación de Wang para un sistema matricial de 3×3 .

Apéndice C

Se presenta el desarrollo matemático de las expresiones que definen el problema de transporte, así como el desarrollo de la ecuación de Wang para un sistema matricial de 3×3 .

Apéndice D

Se muestra la descripción del modelo TEAM en Verilog-A, el cual se utilizó ampliamente en simulaciones SPICE para el análisis y diseño de las aplicaciones analógicas y digitales presentadas en los capítulos 3 y 4.

Apéndice E

Se muestra el código en MATLAB para la solución del problema de asignación de tareas a partir del algoritmo húngaro.

Capítulo 1

1.1 Introducción

Por casi dos siglos toda la teoría de sistemas eléctricos conocida se basó en tres elementos pasivos fundamentales: el resistor, el capacitor y el inductor. Pero en 1971 el profesor Leon Chua desafió a la bien establecida percepción de la teoría clásica proponiendo la existencia de un cuarto elemento fundamental, al cual denominó Memristor (resistor con memoria) [1]. El Memristor opera como un resistor variable que, al suprimir su polarización mantiene su último valor óhmico, es decir, posee memoria no volátil.

En 1976, Chua y Kang introdujeron la definición de los sistemas memristivos, los cuales son capaces de modelar una amplia clase de sistemas dinámicos no lineales, y cuyo comportamiento se asemeja al del memristor. A diferencia del memristor propuesto por Chua, el cambio en los sistemas memristivos depende de una o varias variables de estado, las cuales no están restringidas solamente a la carga o al flujo magnético [2].

Desafortunadamente, Chua no fue capaz de demostrar mediante un desarrollo tecnológico al memristor y su trabajo fue olvidado. No fue sino hasta el año 2008 que investigadores de los laboratorios de Hewlett – Packard fabricaron el primer memristor basado en una nano estructura metal-aislante-metal de TiO_2 [3]. A partir de ese momento, el interés por el estudio de este nuevo dispositivo fue en aumento provocando que muchos investigadores se encuentren, hoy en día, desarrollando modelos que sean capaces de capturar su comportamiento dinámico no lineal, permitiendo de esta manera el uso del memristor en el análisis, diseño y simulación de circuitos electrónicos.

Debido a las dos propiedades que hacen único al memristor: resistencia variable y memoria no volátil, más aparte su gran velocidad de conmutación, alta duración y retención de datos, bajo consumo de energía, alta densidad de integración y

quizás la más importante, su compatibilidad con los procesos de fabricación de tecnología CMOS (esta propiedad es importante, debido a que el escalamiento de los transistores está llegando a sus límites físicos fundamentales por lo que el uso de dispositivos nanométricos, como el Memristor, junto con transistores CMOS podría extender otra década la validez de la Ley de Moore). Todas estas propiedades hacen hoy en día del memristor, un elemento para la creación de nuevos circuitos capaces de revolucionar la industria electrónica.

Aunque a la fecha no existen soluciones electrónicas comerciales basadas en el memristor, en la **Figura 1.1** se muestran las potenciales aplicaciones de éste, siendo las tres áreas más importantes: memorias no volátiles [4-8], circuitos lógicos [9-13] y redes neuronales [14-19].

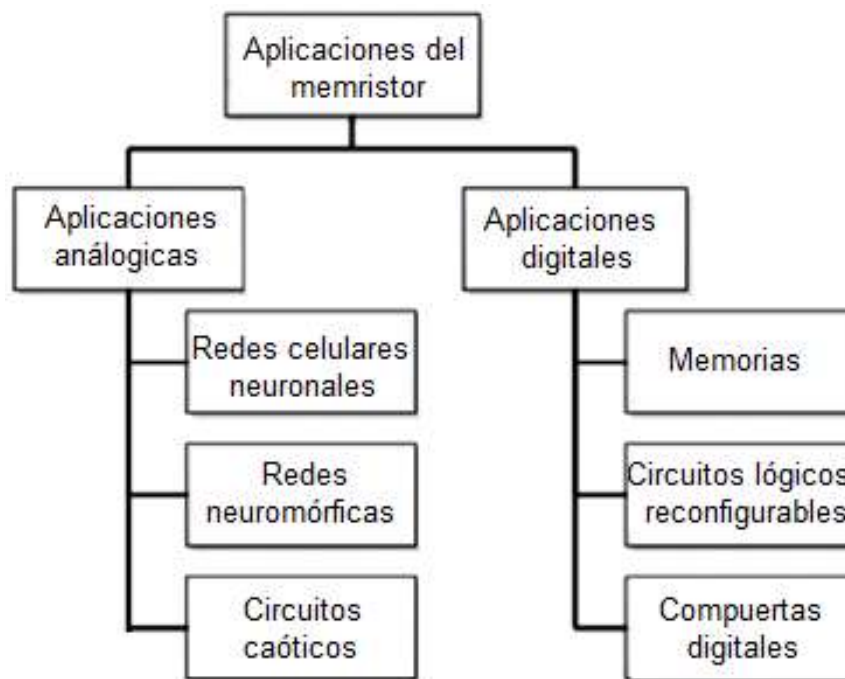


Figura 1.1 Potenciales aplicaciones del memristor

La capacidad para almacenar valores resistivos indefinidamente y su tamaño nanométrico han hecho del memristor un elemento para la creación de memorias no volátiles de gran densidad; pero dado su tamaño y su reducido consumo energético, el uso del memristor en redes neuronales es considerado como muy relevante, ya que en un futuro este dispositivo podría permitir diseñar circuitos

electrónicos analógicos capaces de emular algunas funciones biológicas de un cerebro humano [20].

Debido a todas las características presentadas, la principal motivación de este trabajo de investigación radica en el estudio y análisis del comportamiento dinámico y no lineal de este novedoso dispositivo llamado Memristor, además de su empleo en aplicaciones analógicas y digitales que involucran el uso de redes neuronales, cuyo primer objetivo es la solución de problemas de optimización.

Referencias

- [1] L.O. Chua, Memristor the missing circuit element, *IEEE Trans. Circuit Theory.* 18 (5) (1971) 507-519.
- [2] L.O. Chua, S.M. Kang, Memristive devices and systems, *Proc. IEEE.* 64 (2) (1976) 209-223.
- [3] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, *Nature.* 453 (7191) (2008) 80-83.
- [4] C. Kugeler, M. Meier, R. Rosezin, S. Gilles, R. Waser, High density 3D memory architecture based on resistive switching effect, *Solid State Electron.* 53 (12) (2009) 1287-1292.
- [5] S.H. Jo, K.H. Kim, W. Lu, High density crossbar arrays based on a Si memristive system, *Nano Lett.* 9 (2) (2009) 870-874.
- [6] S.H. Jo, K.H. Kim, T. Chang, S. Gaba, W. Lu, Si memristive devices applied to memory and neuromorphic circuits, 2010 IEEE International Symposium on Circuits and Systems (ISCAS).
- [7] J. Borghetti, Z. Li, J. Straznicky, X. Li, D.A.A. Ohlberg, W. Wu, D.R. Stewart, R.S. Williams, A hybrid nanomemristor/transistor logic circuit capable of self-programming, *Proceedings of the National Academy of Sciences.* 106 (6) (2009) 1699-1703.

- [8] S.H. Jo, W. Lu, CMOS Compatible nanoscale nonvolatile resistance switching memory, *Nano Lett.* 8 (2) (2008) 392-397.
- [9] Q. Xia, W. Robinett, M.W. Cumbie, N. Banerjee, T.J. Cardinali, J.J. Yang, W. Wu, X. Li, W.M. Tong, D.B. Strukov, G.S. Snider, G.M. Ribeiro, R.S. Williams, Memristor-CMOS hybrid integrated circuit for reconfigurable logic, *Nano Lett.* 9 (10) (2009) 3640-3645.
- [10] J. Borhetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R.S. Williams, Memristive switches enable stateful logic operations via material implication, *Nature* 464 (2010) 873-876.
- [11] K.K. Likharev, D.B. Strukov, Prospects for the development of digital CMOL circuits, 2007 IEEE International Symposium on Nanoscale Architectures.
- [12] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, E.G. Friedman, MRL memristor Ratioed logic, CNNA 2012 13th international workshop, Turin Italy.
- [13] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E.G. Friedman, A. Kolodny, U.C. Weiser, Magic memristor aided logic, *IEEE Trans. Circuit Syst.* 61 (11) (2014) 895-899.
- [14] B.L. Barranco, T.S. Gotarredona, Exploiting memristance in adaptive asynchronous spiking neuromorphic nanotechnology systems, 2009 9th IEEE Conference on Nanotechnology.
- [15] G. S. Snider, Spike time dependent learning in memristive nanodevices, 2008 IEEE International Symposium on Nanoscale Architectures.
- [16] Y.V. Pershin, M. Di Ventra, Experimental demonstration of associative memory with memristive neural networks, *Neural Networks* 23 (7) (2010) 881-886.
- [17] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, *Nano Lett.* 10 (4) (2010) 1297-1301.

[18] M. Di Ventra, Y.V. Pershin, Biologically inspired electronics with memory circuit elements, ArXiv e-prints, (2011).

[19] C.Z. Ramos, L.A. Mesa, J.A. Carrasco, T. Masquelier, T.S. Gotarredona, B.L. Barranco, On spike timing dependent plasticity memristive devices and building a self-learning visual cortex, *Frontiers in neuroscience*. 5 (26) (2011) 1-22.

[20] R.S. Williams, How we found the missing memristor, *Nature*. 453 (7191) (2008) 80-83.

Capítulo 2

Aspectos teóricos

2.1 Introducción

En el capítulo 1 se presentan de forma general los conceptos del memristor y los sistemas memristivos, y de su posible impacto en el desarrollo de la industria electrónica. En este capítulo se abordan de una manera más formal los aspectos teóricos que engloban al memristor. Se inicia con la explicación de los argumentos axiomáticos que llevaron a su concepción, seguido por la definición detallada del concepto del memristor y los sistemas memristivos. Además, se presenta una descripción del memristor de TiO_2 fabricado por los laboratorios de Hewlett – Packard, dando pie al análisis de algunos de los modelos con características lineales y no lineales, más notables presentes en la literatura. Este capítulo presenta las bases para la comprensión de resultados mostrados en los capítulos posteriores.

2.2 El memristor ideal

En 1971, el profesor Leon Chua planteó la existencia de un cuarto elemento fundamental de circuitos basándose en argumentos axiomáticos y la teoría electromagnética [1]. Chua observó que combinando por pares las cuatro variables fundamentales de la teoría de circuitos (voltaje v , corriente i , carga q y flujo magnético φ), surgían seis relaciones matemáticas, de las cuales solo cinco se conocían ampliamente, faltando por definir la relación (φ, q) . El conjunto de las posibles parejas se muestra en (2.1).

$$\{(v, \varphi), (i, q), (v, i), (v, q), (i, \varphi), (\varphi, q)\} \quad (2.1)$$

Dos de estas relaciones corresponden a la definición de la carga y el flujo magnético a través de las integrales con respecto al tiempo de la corriente y el voltaje (2.2) y (2.3).

$$q(t) = \int_{-\infty}^t i(\tau) d\tau \quad (2.2)$$

$$\varphi(t) = \int_{-\infty}^t v(\tau) d\tau \quad (2.3)$$

Tres relaciones son dadas por la definición axiomática, vía relación constitutiva, de los tres elementos fundamentales de circuitos: el resistor, el inductor y el capacitor.

$$f_R(v, i) = 0 \quad \text{para el resistor } R \quad (2.4)$$

$$f_L(\varphi, i) = 0 \quad \text{para el inductor } L \quad (2.5)$$

$$f_C(v, q) = 0 \quad \text{para el capacitor } C \quad (2.6)$$

La visión de Chua fue postular que el enlace faltante entre φ y q debía ser completado por otra relación constitutiva (2.7) [2], teorizando de esta manera la existencia de un cuarto elemento faltante, fundamental, de la teoría de circuitos, al cual denominó Memristor (resistor con memoria). La definición axiomática de los 4 elementos de circuitos básicos se observa en la **Figura 2.1**.

$$f_M(\varphi, q) = 0 \quad (2.7)$$

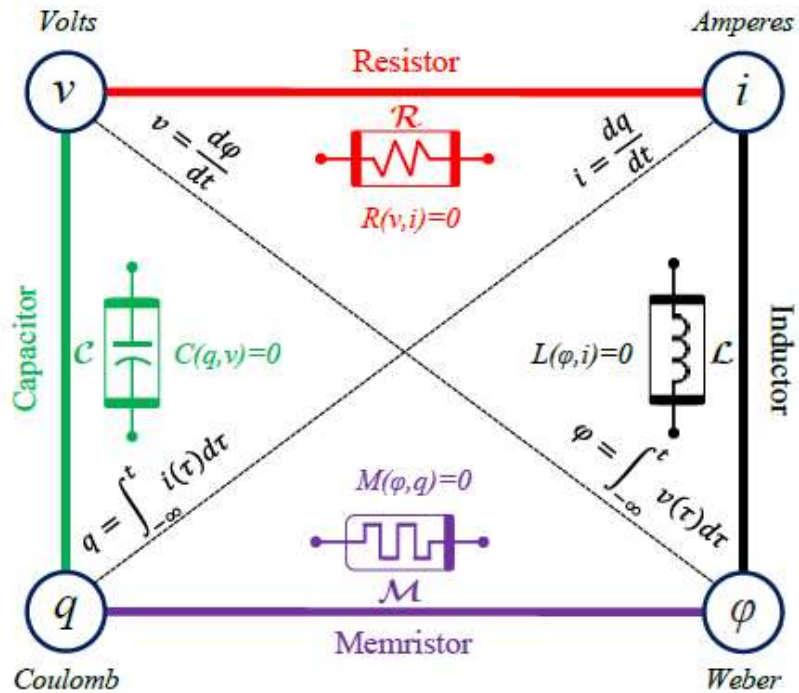


Figura 2.1 Modelo de la relación entre las cuatro variables fundamentales de la teoría de circuitos

Con base en lo anterior, se puede decir que el memristor es un elemento de dos terminales, caracterizado por una relación constitutiva, $f_M(\varphi, q) = 0$, que relaciona de forma no lineal la carga q con el flujo magnético φ , los cuales están definidos a partir de (2.2) y (2.3), respectivamente. De forma más simple, el memristor es un elemento cuya respuesta entrada - salida es determinada por una curva flujo magnético vs. carga ($\varphi - q$). Se dice que el memristor es ideal y pasivo si dicha curva tiene la característica de ser única, no lineal, continuamente diferenciable y monótonamente creciente (**ver la Figura 2.2a**) [1].

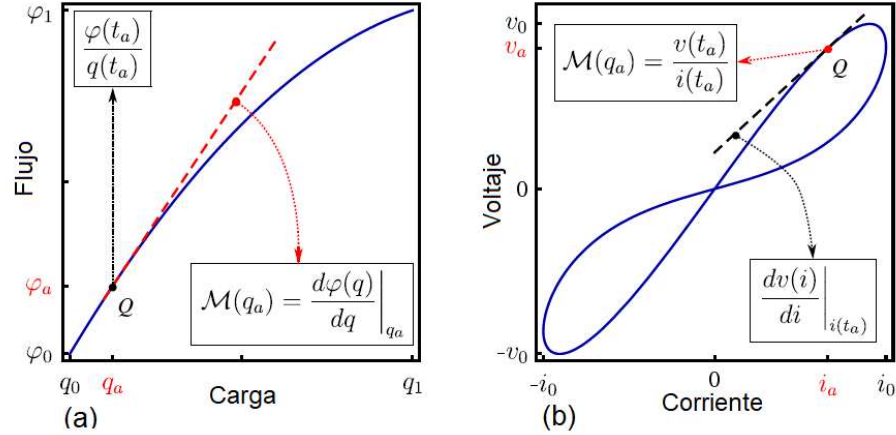


Figura 2.2a) Curva flujo vs. carga. **b)** Relación voltaje – corriente del memristor

El memristor puede ser controlado por la carga (o el flujo magnético) si su relación constitutiva (2.7) se representa a partir de una función explícita (2.8) y (2.9).

$$\varphi = \hat{\varphi}(q) \quad \text{controlado por carga} \quad (2.8)$$

$$q = \hat{q}(\varphi) \quad \text{controlado por flujo} \quad (2.9)$$

Diferenciando (2.8) y (2.9) con respecto al tiempo, se obtienen:

$$\frac{d\varphi}{dt} = \frac{d\hat{\varphi}(q)}{dq} \frac{dq}{dt} \quad (2.10)$$

$$\frac{dq}{dt} = \frac{d\hat{q}(\varphi)}{d\varphi} \frac{d\varphi}{dt} \quad (2.11)$$

Sabiendo que $i = dq/dt$ y $v = d\varphi/dt$, y reemplazando dichas definiciones en (2.10) y (2.11), se obtiene una expresión de la ley de Ohm dependiente de la carga, (para el caso del memristor controlado por la carga) (2.12), cuya representación en el plano voltaje vs. corriente ($v - i$) se aprecia en la **Figura 2.2b**.

$$v = M(q)i \quad (2.12)$$

donde $M(q) = d\hat{\varphi}(q)/dq$ se denomina **memristancia incremental**, tiene como unidad el Ohm (Ω) y representa la pendiente de la curva ($\varphi - q$) en el punto de operación (OP) $Q(q_a, \varphi_a)$ (**ver la Figura 2.2a**).

Para el caso del memristor controlado por el flujo magnético, la expresión de la ley de Ohm se define como:

$$i = W(\varphi)v \quad (2.13)$$

donde $W(\varphi) = d\hat{q}(\varphi)/d\varphi$ se denomina **memductancia incremental**; tiene como unidad el Siemens (S) y representa la pendiente de la curva $q - \varphi$.

Nótese que si la relación entre φ y q es lineal (2.8), la memristancia $M(q)$ será constante y representará un resistor convencional, lineal e invariante en el tiempo, carente de interés, por lo que la parte interesante del memristor radica en su no linealidad; es decir, en su característica de variar su resistencia.

2.2.1 Propiedades del memristor

Una de las propiedades más importantes del memristor es su memoria no volátil, la cual se puede observar al reemplazar q en (2.12) con (2.2) y φ en (2.13) con (2.3), esto es:

$$v = M\left(\int_{-\infty}^t i(\tau)d\tau\right)i \quad (2.14)$$

$$i = W\left(\int_{-\infty}^t v(\tau)d\tau\right)v \quad (2.15)$$

Lo cual revela la propiedad de la memoria no volátil del memristor. Como se observa, la memristancia (o memductancia) incremental depende de la historia completa pasada de la señal de entrada (corriente o voltaje) del memristor desde $t = -\infty$ hasta $t = t_0$. Entonces, se puede concluir que la memristancia (memductancia) se mantiene cambiando, mientras se aplica una señal al memristor. Una vez que esta se remueve, el memristor es capaz de mantener su último estado resistivo por tiempo indefinido hasta que la señal de entrada se aplique nuevamente, sin importar en que instante de tiempo ocurra esto.

Una vez que el voltaje o corriente se especifica, el memristor se comporta como un **resistor no lineal**. En el caso donde la curva $\varphi = \hat{\varphi}(q)$ sea igual a una línea recta, $M(q) = R$ o $W(\varphi) = G$, el memristor se reduce a un **resistor lineal** [1].

Chua estableció tres teoremas que caracterizan al memristor y redes de memristores.

Teorema 1: Criterio de pasividad: establece que un memristor caracterizado por una curva diferenciable $q - \varphi$, es pasivo si y solo si su memristancia incremental es $M(q) \geq 0$ (o memductancia $W(\varphi) \geq 0$) en cualquier instante de tiempo. La potencia instantánea disipada por un memristor se define como

$$p(t) = v(t)i(t) = M(q(t))[i(t)]^2 \quad (2.16)$$

de esta manera, si la memristancia incremental es $M(q) \geq 0$ entonces $p(t) \geq 0$ y el memristor es pasivo (disipativo). De esta forma, solo los memristores caracterizados por una curva $q - \varphi$ monótona creciente, pueden existir en la forma de un dispositivo sin la necesidad de una fuente de energía interna para operar.

Teorema 2: De cerradura: establece que todo sistema de un puerto que contiene solo memristores, es equivalente a un memristor.

Teorema 3: De existencia y unicidad: establece que cualquier red que contenga solo memristores pasivos tiene únicamente una solución.

Aunque en ese tiempo Chua fue capaz de simular la característica $\varphi - q$ y la relación corriente-voltaje del memristor mediante elementos activos, no descartó la posibilidad de que el memristor con una curva $\varphi - q$ monótonamente creciente y sin una fuente de energía interna ya hubiera sido fabricado como una curiosidad de laboratorio, pero no fue identificado.

2.3 Sistemas memristivos

En 1976, Chua y su estudiante Sung Mo Kang generalizaron el concepto del memristor a una interesante clase de sistemas dinámicos no lineales llamados

sistemas memristivos [3], donde el memristor es únicamente un caso especial de esta clase.

Para describir a un sistema memristivo se necesitan dos ecuaciones: **la relación del transporte estático** (2.17), encargada de establecer la relación voltaje – corriente, por medio de una resistencia generalizada, la cual es función de la corriente i y de la variable de estado x .

$$v = R(x, i)i \quad (2.17)$$

y otra denominada **ecuación de estado** (2.18), que describe la dinámica de la variable (o variables) de estado x , como una razón de cambio, donde f expresa la dependencia fundamental de dx/dt en x e i . La variable de estado representa los parámetros físicos internos del dispositivo y no depende de ninguna variable externa, tales como el voltaje o la corriente.

$$dx/dt = f(x, i) \quad (2.18)$$

Las expresiones (2.17) y (2.18) son características de un sistema memristivo controlado por corriente, mientras que para el caso de un sistema controlado por voltaje se tiene:

$$i = G(x, v)v \quad (2.19)$$

$$dx/dt = f(x, v) \quad (2.20)$$

R y G actúan como la memristancia y memductancia del sistema y f determina la evolución de la variable de estado.

Una característica principal de los sistemas memristivos cuando son activados por una señal periódica, es su propiedad de cruce por cero, lo cual significa que la salida del sistema es cero cuando la entrada es cero. Esta propiedad de cruce por cero, se manifiesta en el plano $i - v$ en forma de una **figura de Lissajous**, que es doblemente valuada y siempre pasa por el origen. Chua llamo a esta curva **lazo de histéresis pellizcado** (ver la Figura 2.2b).

La histéresis para el caso de un sistema memristivo controlado por corriente, se forma debido a que los tiempos en que se presentan los valores máximo y mínimo de la señal de entrada (corriente) no corresponden a los tiempos de la señal de salida (voltaje). El pellizco en el origen de la señal de histéresis ocurre debido a que no existe un desfase entre la señal de entrada con la de salida, es decir $i(t) = 0$ y $v(t) = 0$, ocurren al mismo tiempo [4].

Otra propiedad de los sistemas memristivos, es que, al ser operados bajo una señal periódica, la figura de Lissajous varía con la frecuencia de dicha señal. A muy bajas frecuencias, el sistema memristivo se comporta como un **resistor no lineal**, mientras que, a frecuencias extremadamente altas, la figura de Lissajous tiende a una línea recta que pasa por el origen y el sistema memristivo se comporta como un **resistor lineal** (ver la Figura 2.3). La interpretación física que propusieron Chua y Kang para este interesante fenómeno, es que el sistema posee cierta inercia que le impide responder rápidamente ante la rápida variación de la señal de excitación, llevándolo a cierto estado de equilibrio [2]; visto de otra manera, un sistema memristivo excitado con una señal periódica de alta frecuencia, no cuenta con el tiempo suficiente para cambiar su estado [5].

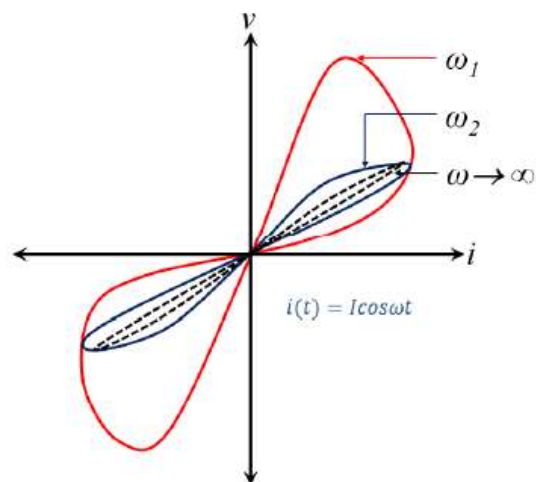


Figura 2.3 Comportamiento de histéresis para un memristor controlado por corriente a diferentes frecuencias.

Chua y Kang [3] consideraron que un sistema memristivo podía modelar a una gran variedad de dispositivos y sistemas dinámicos no lineales, cuyo comportamiento se asemeje al del memristor, mencionando como ejemplo el termistor, el modelo neuronal de Hodgkin-Huxley y los tubos de descarga, pero al igual que el concepto del memristor, la definición de un sistema memristivo fue un ejercicio matemático independiente de cualquier mecanismo físico conocido en ese tiempo.

En este trabajo usaremos el término *memristor* para describir al memristor ideal y los sistemas memristivos. basándonos en la publicación de Chua [4], en la cual establece que todos los dispositivos de memoria no volátil de dos terminales, basados en la conmutación resistiva son memristores, independientemente del material del dispositivo o los mecanismos físicos de operación.

2.4 Memristor de Hewlett - Packard

Desafortunadamente, Chua no fue capaz de demostrar mediante un desarrollo tecnológico el memristor y su trabajo fue olvidado dado que carecía en apariencia, de aplicación práctica. No es hasta el año 2008, que investigadores de los laboratorios de Hewlett Packard (HP), en Palo Alto California, junto con Stanley Williams, sorprendieron a la comunidad científica con una publicación en la revista *Nature*, donde se daban detalles sobre la creación física del memristor [6]. La idea original de Williams y sus colaboradores fue la creación de un arreglo de barras cruzadas (crossbar), donde cada cruce estaría representado por un nano-interruptor, encargado de conectar y desconectar la barra superior con la inferior por medio de un voltaje. Este nano-interruptor debía tener una resistencia que variase entre un valor mínimo R_{ON} y un valor máximo R_{OFF} , y cuya razón resistiva R_{OFF}/R_{ON} fuera de 1000:1, es decir que el estado R_{OFF} fuera 1000 veces más resistivo al paso de la corriente que el estado R_{ON} [7].

La solución a la variación de la resistencia del nano-interruptor se encontró a través de la microscopia de escaneo de tuneleo Scanning Tunneling Microscopy (STM), la cual genera imágenes de resolución atómica escaneando una aguja

muy puntiaguda a través de una superficie, para de esta manera medir la corriente eléctrica que fluye entre la punta de la aguja y la superficie. Al aproximar la punta a la superficie a una distancia de 0.1nm, la corriente de tuneo se incrementa un orden de magnitud. Teniendo esto en cuenta, se necesitaba un mecanismo similar, capaz de cambiar el espacio efectivo entre las dos barras del crossbar una distancia de 0.3 nm para de esta manera obtener la razón resistiva deseada.

Williams y su equipo diseñaron una estructura de aspecto muy similar a un emparedado (**ver la Figura 2.4a**). Esta Figura, se conforma por dos contactos metálicos de platino (Pt) y entre estos existe una película delgada de dióxido de titanio, la cual se divide en dos capas: la capa inferior es dióxido de titanio puro TiO_2 mientras que la superior es dióxido de titanio deficiente de oxígeno TiO_{2-x} , donde x es aproximadamente 0.05, formando de esta manera una estructura de tipo metal/óxido/metal. El TiO_2 actúa como aislante mientras que el TiO_{2-x} como un semiconductor, ya que sus vacancias de oxígeno son donadores de electrones y por lo tanto poseen carga positiva.

El funcionamiento del dispositivo es el siguiente: cuando un voltaje positivo es aplicado en el electrodo superior, las vacancias de oxígeno presentes en la capa de TiO_{2-x} son repelidas obligándolas a bajar a la capa de TiO_2 . De esta forma, la capa de TiO_2 se transforma paulatinamente en TiO_{2-x} lo que provoca un aumento en la conductividad del dispositivo (**ver la Figura 2.4b**). Si, por el contrario, se invierte la polaridad del voltaje aplicado en el electrodo superior, las vacancias de oxígeno son atraídas hacia el electrodo superior, retirándose de la capa de TiO_2 provocando un aumento de la resistencia del dispositivo (**ver la Figura 2.4c**). Si se suprime el voltaje, el valor resistivo de éste permanece intacto e inalterable hasta que se vuelva a reanudar la polarización, esto sin importar cuando ocurra. Debido a esta característica se puede decir que este es un dispositivo con memoria ya que recuerda su último valor de resistencia.

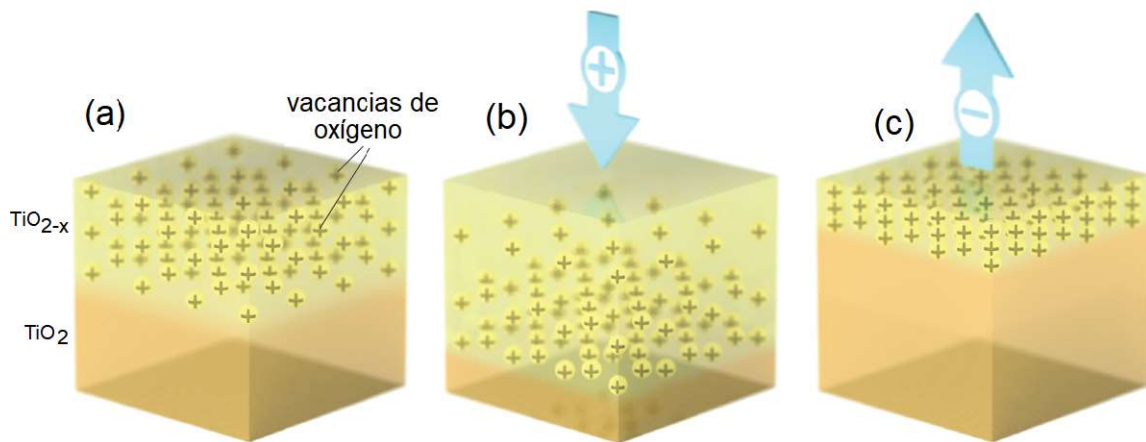


Figura 2.4a) Estructura del memristor de HP. **b)** Funcionamiento del memristor ante una polarización positiva. **c)** Funcionamiento del memristor ante una polarización negativa.

A través de los resultados experimentales obtenidos, los autores observaron que la característica corriente-voltaje (I-V) del nano-interruptor (**ver la Figura 2.5b**) presentaba una similitud con la gráfica descrita por Chua y Kang en [2] (**ver la Figura 2.5a**), además de notar que en ambos casos, el cambio total de la resistencia dependía de la duración del voltaje aplicado, es decir, mientras más tiempo se aplica un voltaje positivo la resistencia disminuye hasta alcanzar su mínimo valor R_{ON} y mientras más tiempo se aplica un voltaje negativo la resistencia aumenta hasta alcanzar su máximo valor R_{OFF} . Por todo esto, Williams y su equipo llegaron a la conclusión de que la memristancia estaba involucrada en su dispositivo y así de esta manera habían fabricado exitosamente al elemento faltante, el Memristor.

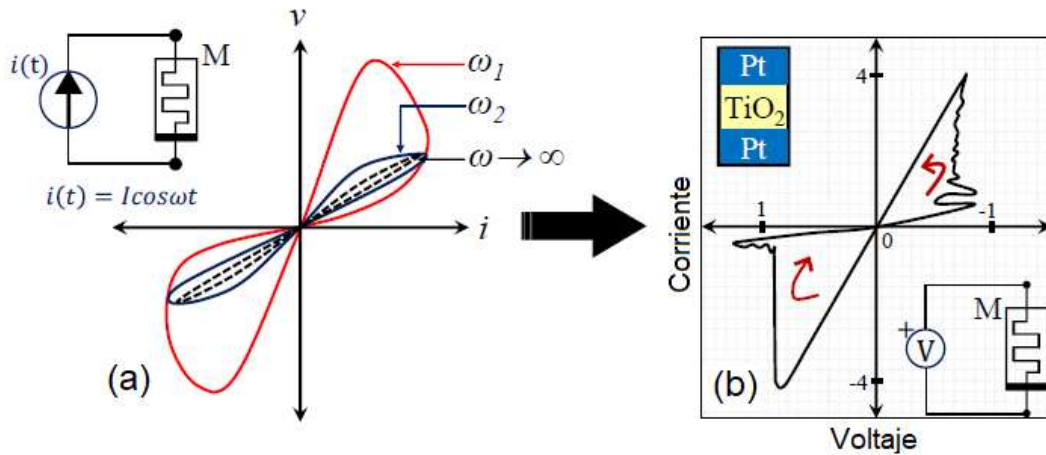


Figura 2.5a) Curva corriente – voltaje descrita por Chua. **b)** Gráfica de la característica corriente – voltaje obtenida del memristor de HP controlado por voltaje.

2.5 Modelo lineal

En [6], los autores presentan un modelo matemático capaz de representar la dinámica del memristor de HP, además de explicar como la memristancia se vuelve más significativa en dispositivos de escala nanométrica, dando de esta manera respuesta a comportamientos anómalos de corriente – voltaje reportados en el pasado [8].

En la **Figura 2.6**, se muestra la imagen geométrica del memristor de HP, la cual como se explicó en la sección anterior, se conforma por una película delgada de dióxido de titanio TiO_2 de espesor D , que se encuentra entre dos contactos metálicos de platino Pt. Esta película se divide en dos regiones:

- TiO_{2-x} región dopada¹ (de longitud $w(t)$) con vacancias de oxígeno, la cual es modulada por la cantidad de carga que pasa por el dispositivo.
- TiO_2 región sin dopar¹ (de longitud $D - w(t)$).

¹ Nota sobre nomenclatura. En esta tesis, la descripción del material que compone a un memristor como “**región dopada**” es para aquella que tiene vacancias de oxígeno y “**región no dopada**” para la que no tiene vacancias de oxígeno. Esta nomenclatura es la usada en la publicación que da origen a la presentación del memristor como el cuarto elemento eléctrico pasivo [6].

Estas regiones pueden ser vistas como dos resistores variables conectados en serie, donde $R_{ON} w(t)/D$ representa la resistencia de la región dopada y $R_{OFF}(D - w(t))$ la resistencia de la región sin dopar. De esta manera, la resistencia efectiva total R_{mem} del dispositivo se puede expresar con la ecuación (2.21).

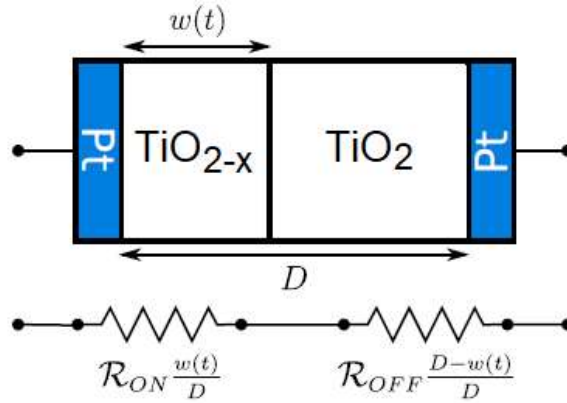


Figura 2.6 Imagen geométrica del memristor de HP con su representación eléctrica.

$$R_{mem} = R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \quad (2.21)$$

$$x = \frac{w(t)}{D} \in [0,1] \quad (2.22)$$

Donde x es el ancho normalizado de la región dopada $w(t)$, R_{ON} y R_{OFF} son los valores de la resistencia cuando $w(t) = D$ y $w(t) = 0$, respectivamente.

Para el caso más simple de conducción electrónica óhmica y arrastre iónico lineal en un campo eléctrico uniforme, el memristor puede ser modelado a partir del siguiente par de ecuaciones: La relación voltaje - corriente (2.23) y la ecuación de estado (2.24), donde $\mu_v = 10^{-14} \text{m}^2 \text{s}^{-1} \text{V}^{-1}$ representa la movilidad iónica promedio.

$$v(t) = R_{mem} i(t) \quad (2.23)$$

$$\frac{dx}{dt} = \mu_v \frac{R_{ON}}{D^2} i(t) \quad (2.24)$$

La expresión (2.24) describe la ecuación de estado, o de arrastre; esta modela la velocidad de arrastre de las cargas [9]. Integrando (2.24) con respecto a t se obtiene (2.25)

$$x = \mu_V \frac{R_{ON}}{D^2} q(t) \quad (2.25)$$

Lo cual significa que el ancho normalizado de la región dopada x es proporcional a la cantidad de carga que pasa a través del dispositivo.

Insertando la ecuación (2.25) en (2.21) y suponiendo que $R_{ON} \ll R_{OFF}$, se puede obtener una expresión simplificada de la memristancia del sistema, resultando en una expresión que define la memristancia en términos de las propiedades del material y la geométrica del dispositivo.

$$M(q) = R_{OFF} \left(1 - \frac{\mu_V R_{ON}}{D^2} q(t) \right) \quad (2.26)$$

Entre más pequeños son los dispositivos que se construyen, la memristancia se vuelve más notoria y en algunos casos dominante; esto, debido a que la memristancia es más importante en dispositivos de escala nanométrica, y su comportamiento es difícil de observar en dispositivos de escala mayor [6,7].

Debido a que en (2.24) el cambio de la variable de estado dx/dt depende linealmente de la carga q , el modelo de HP se puede denominar **modelo lineal de arrastre de dopantes**.

2.6 Modelo lineal con función de ventana

El modelo lineal de arrastre de dopantes carece, de algunas características importantes: 1) Incluso cuando se aplica un voltaje pequeño a dispositivos de escala nanométrica se produce un campo eléctrico elevado, lo que genera no linealidades en el transporte iónico. 2) La región dopada $w(t)$ no puede alcanzar la posición cero ($w(t) = 0$) ya que esto significaría que no existen vacancias de oxígeno presentes en el dispositivo, lo cual es físicamente imposible. 3) La velocidad de arrastre, dw/dt , debe de ser mayor en el centro del dispositivo e

irse reduciendo hasta alcanzar el valor de cero a medida que $w(t)$ se aproxima a los extremos del dispositivo, esto es cuando $w(t) = 0$ o $w(t) = D$, produciendo de esta manera una no linealidad en el transporte iónico; a este fenómeno se le conoce como **efecto de frontera** [10]. 4) Este modelo considera un arrastre de dopantes de tipo lineal a lo largo de todo el memristor, y por lo que se acaba de explicar, físicamente no puede ser posible.

El problema de la no linealidad del modelo lineal se resuelve agregando en (2.24) una función $f(x)$ no lineal normalizada, $x = w(t)/D \in [0,1]$, denominada comúnmente como **función de ventana**.

$$\frac{dx}{dt} = \frac{\mu_v R_{on} i(t)}{D^2} f(x) \quad (2.27)$$

En [11] se establece que una función de ventana $f(x)$ efectiva, debe cumplir las siguientes condiciones:

- Considerar las condiciones de frontera en los extremos del dispositivo, o sea, que el valor de w esté entre 0 y D .
- Imponer arrastre no lineal sobre todo el dispositivo.
- Proveer un enlace entre el modelo lineal de arrastre de dopantes y el no lineal.
- Utilizar un parámetro de control para el ajuste del modelo.
- Ser escalable, es decir, $0 \leq f_{\max}(x) \leq 1$.

Una característica de la función de ventana $f(x)$ es que su valor máximo debe ocurrir en la mitad del dispositivo cuando ($x = 0.5$) y debe ser cero en los extremos de este cuando ($x = 0$ o $x = 1$).

En [6], Strukov et al. proponen la siguiente función de ventana (2.28), la cual impone una no linealidad a lo largo del dispositivo, pero tiene la desventaja de carecer de un parámetro de control (**ver la Figura 2.7**).

$$f(x) = x - x^2 \quad (2.28)$$

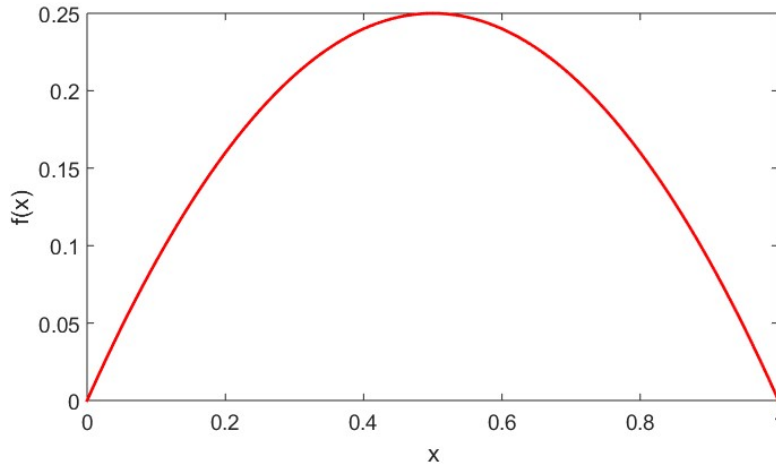


Figura 2.7 Función de ventana de Strukov

Joglekar y Wolf [12] proponen una función de ventana (2.29) que asegura que la velocidad de arrastre en los extremos del dispositivo sea cero.

$$f(x, p) = 1 - (2x - 1)^{2p} \quad (2.29)$$

donde p es un parámetro de control de tipo entero positivo. Para valores mayores de p , la función de ventana adquiere la forma de tipo rectangular, donde se tiene que $f(x, p) = 1$ para cualquier valor de la variable de estado x , y de esta manera el fenómeno de arrastre iónico no lineal decrece (**ver la Figura 2.8**).

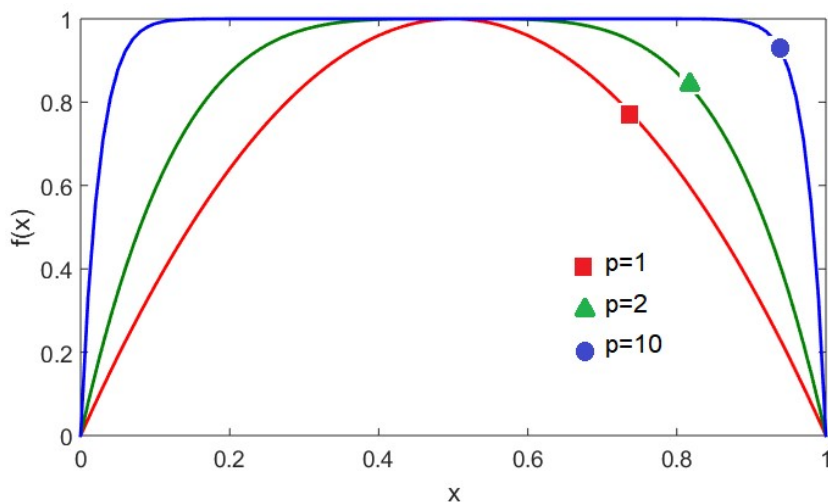


Figura 2.8 Función de ventana de Joglekar para distintos valores de p .

A pesar de que la función de Joglekar agrega una no linealidad al modelo y limita los valores de x , esta función presenta el problema del **estado terminal**, esto es, dado que dx/dt es forzada al valor cero, el estado interno del dispositivo no puede cambiar después de que x alcanza alguno de los extremos.

La ventana de Biolek et al. [13], subsana el problema del estado terminal considerando la corriente que pasa a través del dispositivo; esto quiere decir, que una función de ventana no debe depender solamente de la variable de estado x sino también de la corriente i . p es un entero positivo, $stp(i) = 1$ para $i \geq 0$ y $stp(i) = 0$ para $i < 0$.

$$f(x, p, i) = 1 - (x - stp(-i))^{2p} \quad (2.30)$$

Cuando x comienza en 0, la función es igual a 1, y cuando x se incrementa aproximándose a D , la función se aproxima a 0. Una vez que la corriente cambia de dirección, la función cambia inmediatamente a 1. Entonces, cuando x disminuye de regreso a 0, la función también disminuye a 0. Al cambiar la dirección de la corriente, el ciclo comienza otra vez (**ver la Figura 2.9**).

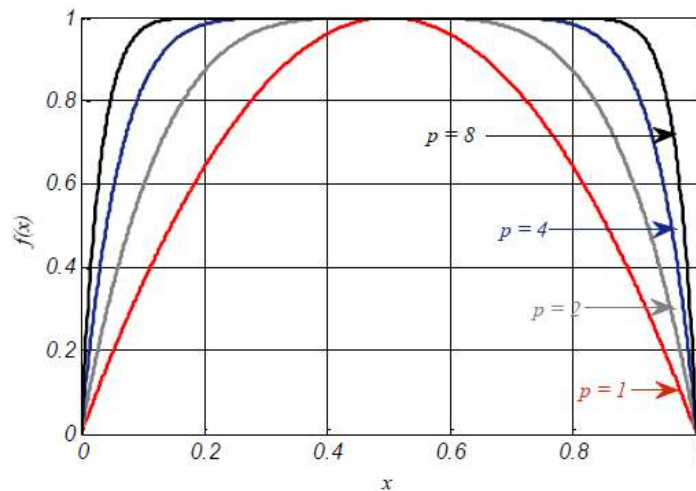


Figura 2.9 Función de ventana de Biolek

A pesar de que la función de ventana de Biolek resuelve el problema del estado terminal, está carece de un factor de escalabilidad que sirva de ajuste, es decir, el máximo valor de $f(x)$ no puede ser cambiado a un valor menor o mayor de 1.

Prodromakis et al. [11] proponen la siguiente función de ventana (2.31).

$$f(x, p) = 1 - [(x - 0.5)^2 + 0.75]^p \quad (2.31)$$

Donde p es un parámetro de control que actúa de tres maneras distintas: 1) permite un escalamiento de la función de ventana, lo cual implica que $f_{\max}(x)$ puede tomar cualquier valor entre 0 y 1. 2) a diferencia del parámetro de control de tipo entero presentado en Joglekar [12] y Biolek [13], p puede tomar cualquier valor positivo de tipo real, permitiendo un mayor grado de flexibilidad. 3) con $p = 1, 4, 10$ y 80 se pueden imponer distintos tipos de arrastre no lineal sobre todo el dispositivo, mientras que con un p mayor se tiene un enlace con el modelo lineal de dopantes (**ver la Figura 2.10**).

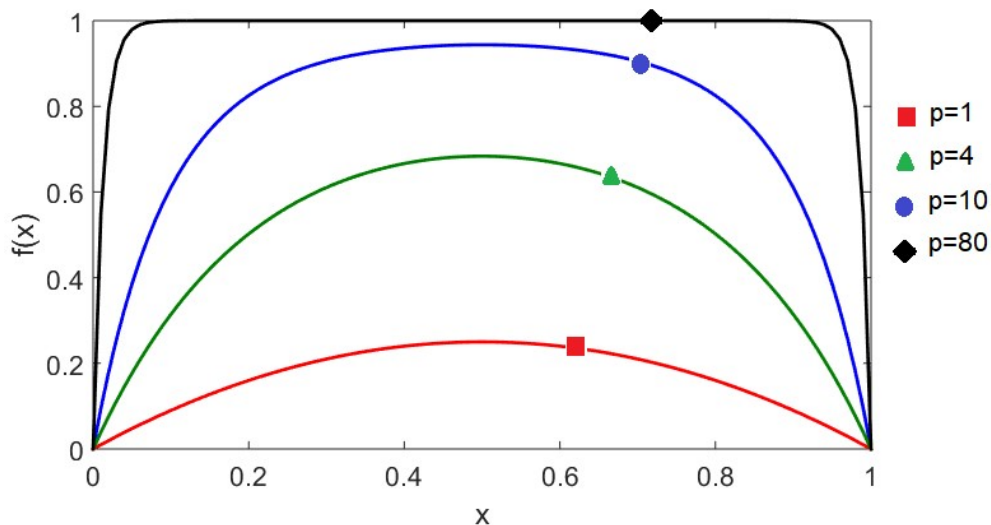


Figura 2.10 Función de ventana de Prodromakis para distintos valores de p .

Aunque la ventana de Prodromakis resuelve la falta de escalabilidad de Biolek, ésta no considera la corriente del dispositivo.

2.7 Modelo no lineal

A pesar de que los modelos explicados anteriormente son intuitivos, populares, fácilmente implementables en SPICE y presentan una cierta no linealidad, (para el caso del modelo lineal con función de ventana), estos modelos no son capaces de capturar el comportamiento complejo no lineal de dispositivos reales [14-16].

Un problema común de estos modelos es la ausencia de un umbral, esto significa que la resistencia del memristor cambia ante cualquier valor del voltaje (o corriente) aplicado. Resultados experimentales han demostrado la presencia de un umbral, lo cual provoca que la curva de histéresis no se aprecie a menos que el voltaje a través del memristor sea mayor a este umbral [17]. Otra característica que se ha observado es la diferencia en la velocidad de conmutación entre los estados ON y OFF, haciendo que la curva de histéresis tenga un comportamiento asimétrico [17, 18]. Por estas razones, el modelo lineal y el lineal con función de ventana, no se ajustan con suficiente precisión al comportamiento de un memristor medido experimentalmente, además de que la derivada de la variable de estado es directamente proporcional a la corriente (2.24).

En [19], se propone un modelo basado en los resultados experimentales reportados en [14], donde la relación entre la corriente y el voltaje se representa con (2.32).

$$i(t) = w(t)^n \beta \sinh(\alpha v(t)) + \chi [\exp(\gamma v(t)) - 1] \quad (2.32)$$

α , β , γ y χ son parámetros de ajuste de tipo real; el exponente n determina como la variable de estado afecta la curva $i - v$ y w es la variable de estado con un valor dentro del intervalo $[0,1]$.

Este modelo presenta un comportamiento de conmutación de tipo asimétrico. Cuando el dispositivo está en el estado ON (R_{ON}) y la variable de estado w se acerca a 1, la corriente es dominada por la primera expresión $\beta \sinh(\alpha v(t))$, la cual describe un fenómeno de tuneleo. Por otra parte, cuando el dispositivo está en el estado OFF (R_{OFF}) y la variable de estado w se acerca a 0, la corriente es dominada por la segunda expresión $\chi [\exp(\gamma v(t)) - 1]$, la cual se asemeja a la ecuación del diodo rectificador. Este modelo presenta una dependencia no lineal del voltaje con la derivada de la variable de estado, como se observa en (2.33).

$$\frac{dw}{dt} = af(w)v(t)^m \quad (2.33)$$

donde a , m son constantes y $f(w)$ representa la función de ventana de Joglekar (2.29). Como se aprecia en (2.33), a diferencia del modelo lineal y lineal con función de ventana, la derivada de la variable de estado tiene una dependencia de forma no lineal con el voltaje de entrada $v(t)$ del memristor.

2.8 Modelo de Pickett

En [18] se presenta un modelo no lineal basado en resultados experimentales realizados en dispositivos de Pt – TiO₂ – Pt. Además, este modelo da una pista sobre los mecanismos físicos que gobiernan el comportamiento dinámico de los memristores compuestos de TiO₂.

En este modelo los autores realizaron un proceso de electro formado a dispositivos de TiO₂. Este es un método que produce vacancias de oxígeno y mejora la conductividad eléctrica. Como resultado del electro formado la mayor parte del óxido se transformó en un canal conductor con vacancias de oxígeno TiO_{2-x}, con un valor resistivo de cientos de ohms y una barrera túnel, de ancho w , entre el canal conductor y el contacto de platino Pt superior (**ver la Figura 2.11**). El ancho de la barrera túnel se puede modular a partir del voltaje aplicado y se denomina variable de estado.

A diferencia de los modelos descritos hasta el momento, donde la región dopada y no dopada del memristor se representan como dos resistores variables conectados en serie, en este modelo, se tiene un resistor R_s conectado en serie con una barrera túnel. Cuando se aplica una polarización positiva en el contacto superior de platino Pt, se produce un incremento en la variable de estado w , ancho de la barrera, ocasionando un incremento exponencial en la resistencia del dispositivo. En el caso contrario, cuando la polarización es negativa, la variable de estado decremента y la resistencia disminuye exponencialmente.

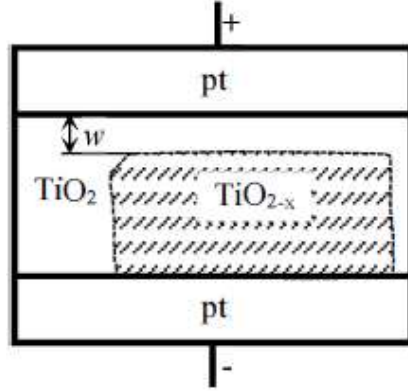


Figura 2.11 Estructura del memristor del modelo de Pickett.

Los resultados presentados en [18], revelaron que el comportamiento dinámico de la conmutación a los estados ON y OFF es altamente no lineal y asimétrico, esto debido a una dependencia de tipo exponencial de la velocidad de arrastre de los dopantes ionizados en la magnitud y polaridad de la corriente o voltaje de excitación. Con el objetivo de determinar una expresión analítica para describir este comportamiento, los autores propusieron la siguiente ecuación de estado.

Para el caso de conmutación hacia el estado OFF, cuando ($i > 0$)

$$\frac{dw}{dt} = c_{\text{off}} \sinh\left(\frac{i}{i_{\text{off}}}\right) \exp\left[-\exp\left(\frac{w - a_{\text{off}}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right] \quad (2.34)$$

Mientras que para el caso de conmutación hacia el estado ON, cuando ($i < 0$)

$$\frac{dw}{dt} = c_{\text{on}} \sinh\left(\frac{i}{i_{\text{on}}}\right) \exp\left[-\exp\left(-\frac{w - a_{\text{on}}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right] \quad (2.35)$$

donde c_{off} , c_{on} , i_{off} , i_{on} , a_{off} , a_{on} , w_c y b son parámetros de ajuste. El parámetro c_{off} es mayor que c_{on} por un orden de magnitud y ambos parámetros influyen en la magnitud del cambio de w . Los parámetros i_{off} e i_{on} son los umbrales de corriente. Bajo estos umbrales, no hay cambio en la variable de estado w . Los valores de a_{off} y a_{on} evitan que w crezca más allá de los límites superior e inferior del dispositivo por lo que no se necesita una función de ventana.

Los resultados experimentales en [18] mostraron que el tiempo de conmutación al estado ON es más rápido que el tiempo de conmutación al estado OFF. Los

autores del modelo atribuyen este comportamiento a la interacción de la corriente de difusión y arrastre en un campo eléctrico. Cuando se aplica una polarización positiva en el contacto superior (**ver la Figura 2.11**), las vacancias de oxígeno cargadas positivamente son repelidas hacia el canal conductor, incrementando la barrera túnel. Esto incrementa la concentración de las vacancias en el canal lo que da como resultado una corriente de difusión de vacancias actuando en dirección opuesta a la corriente de arrastre y un campo eléctrico interno opuesto al campo eléctrico aplicado, lo cual desacelera el arrastre de vacancias, esto para el caso de la conmutación hacia el estado OFF. Sin embargo, cuando se aplica una polarización negativa, el campo eléctrico aplicado externamente, el campo interno y la difusión actúan en la misma dirección acelerando la conmutación hacia el estado ON.

La relación corriente - voltaje se representa con una ecuación implícita basada en el modelo de tuneleo de Simmons [20, 21].

$$i = \frac{j_0 A}{\Delta w^2} \left\{ \phi_I e^{-B\sqrt{\phi_I}} - (\phi_I + e|v_g|) e^{-B\sqrt{\phi_I + e|v_g|}} \right\} \quad (2.36)$$

Donde

$$j_0 = \frac{e}{2\pi h}, \quad w_1 = \frac{1.2\lambda w}{\phi_0}, \quad \Delta w = w_2 - w_1 \quad (2.37)$$

$$\phi_I = \phi_0 - e|v_g| \left(\frac{w_1 + w_2}{w} \right) - \left(\frac{1.15\lambda w}{\Delta w} \right) \ln \left(\frac{w_2(w - w_1)}{w_1(w - w_2)} \right) \quad (2.38)$$

$$B = \frac{4\pi\Delta w\sqrt{2m}}{h} \quad (2.39)$$

$$w_2 = w_1 + w \left(1 - \frac{9.2\lambda}{(3\phi_0 + 4\lambda - 2e|v_g|)} \right) \quad (2.40)$$

$$\lambda = \frac{e^2 \ln(2)}{8\pi\kappa\epsilon_0 w} \quad (2.41)$$

Donde A es el área del canal del memristor, e es la carga del electrón, v_g es el voltaje a través de la barrera túnel, m es la masa del electrón, h es la constante de Planck, κ es la constante dieléctrica y ϕ_0 es la altura de la barrera en electrón volts.

A pesar de que el modelo de Pickett es el más preciso para un memristor de TiO_2 por estar basado en mediciones experimentales, este modelo tiene algunos inconvenientes: 1) es complicado, 2) la expresión de la relación corriente – voltaje (2.36) no es explícita, 3) no presenta la característica de ser un modelo genérico, ya que solo describe a un solo tipo de memristor TiO_2 , 4) la descripción en SPICE del modelo que se propone en [22] es computacionalmente ineficiente, ya que presenta problemas de convergencia [23].

2.9 Modelo de umbral adaptable TEAM

Como se mencionó en la sección anterior, el modelo de Pickett es considerado el modelo más preciso ya que, con base en mediciones experimentales, es capaz de describir el comportamiento dinámico de la variable de estado de un memristor de TiO_2 , pero como se explicó anteriormente, este modelo presenta varias desventajas, las cuales lo hacen inadecuado para el diseño de circuitos basados en dispositivos memristivos. Por tal motivo, surge la necesidad de un modelo simple, preciso y computacionalmente eficiente.

Kvatinsky et al. proponen el modelo denominado TEAM (*ThrEshold Adaptive Memristor model*) [24], para memristores controlados por corriente, el cual presenta el mismo comportamiento al modelo de Pickett, pero con expresiones matemáticas más simples. Para la simplificación del análisis y la eficiencia computacional, este modelo se basa en un par de suposiciones; 1) la variable de estado no presenta cambio alguno si el valor de la corriente i es menor a un cierto umbral, 2) se utiliza una dependencia polinomial en lugar de una dependencia exponencial.

El modelo TEAM establece que la dependencia de la derivada de la variable de estado de la corriente y de ella misma, se puede modelar como la multiplicación

de dos funciones independientes, es decir, una función dependiente de la variable de estado y otra dependiente de la corriente. Bajo esta suposición la ecuación de estado se representa de la siguiente manera

$$\frac{dx}{dt} = \begin{cases} k_{\text{off}} \left(\frac{i(t)}{i_{\text{off}}} - 1 \right)^{\alpha_{\text{off}}} f_{\text{off}}(x), & 0 < i_{\text{off}} < i \\ 0, & i_{\text{on}} < i < i_{\text{off}} \\ k_{\text{on}} \left(\frac{i(t)}{i_{\text{on}}} - 1 \right)^{\alpha_{\text{on}}} f_{\text{on}}(x), & i < i_{\text{on}} < 0 \end{cases} \quad (2.42)$$

$$(2.43)$$

$$(2.44)$$

donde k_{off} , k_{on} , α_{off} y α_{on} son parámetros de ajuste, i_{off} y i_{on} son las corrientes de umbral y x es la variable de estado. El parámetro k_{off} es un número positivo mientras que k_{on} es negativo. Las funciones $f_{\text{off}}(x)$ y $f_{\text{on}}(x)$ son funciones de ventana (2.45) y (2.46) propuestas por Kvatinsky (**ver la Figura 2.12**), las cuales representan la dependencia en la variable de estado y restringen el valor de x a los límites del dispositivo x_{on} y x_{off} . El modelo TEAM tiene la flexibilidad de poder utilizar diferentes funciones de ventanas [11, 12] con el fin de emular diferentes comportamientos.

$$f_{\text{off}}(x) = \exp \left[-\exp \left(\frac{x - a_{\text{off}}}{w_c} \right) \right] \quad (2.45)$$

$$f_{\text{on}}(x) = \exp \left[-\exp \left(-\frac{x - a_{\text{on}}}{w_c} \right) \right] \quad (2.46)$$

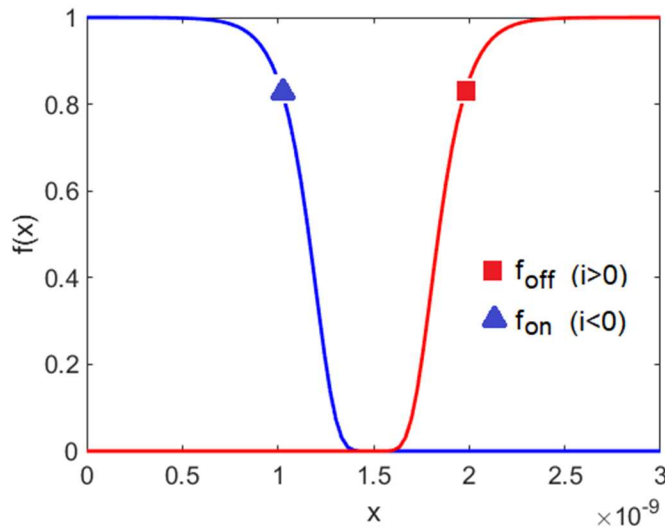


Figura 2.12 Función de ventana de Kvatinsky.

A diferencia de los modelos analizados anteriormente, el modelo TEAM tiene la característica de que la relación corriente - voltaje es indefinida, lo que significa que se puede escoger libremente. Para el caso donde la resistencia cambia linealmente con la variable de estado x , la relación corriente - voltaje se puede especificar con la siguiente expresión.

$$v(t) = \left[R_{ON} + \frac{R_{OFF} - R_{ON}}{x_{off} - x_{on}} (x - x_{on}) \right] i(t) \quad (2.47)$$

Como se observó en [18], el cambio de la resistencia en elementos memristivos es dependiente de un efecto de tuneleo, altamente no lineal, donde cualquier cambio en el ancho de la barrera túnel (variable de estado x) provoca un cambio en la resistencia de forma exponencial, por lo que la relación corriente - voltaje se define con las expresiones (2.48) y (2.49), donde R_{OFF} y R_{ON} son los correspondientes valores resistivos del memristor cuando la variable de estado es x_{off} y x_{on} , respectivamente.

$$v(t) = R_{ON} e^{(\lambda/x_{off} - x_{on})(x - x_{on})} i(t) \quad (2.48)$$

$$\lambda = \ln \left(\frac{R_{OFF}}{R_{ON}} \right) \quad (2.49)$$

Este modelo tiene la característica de ser general y flexible, ya que además de reproducir el comportamiento del modelo de Pickett para un memristor de TiO_2 , con un error medio del 0.2% y una mejora en el tiempo de simulación del 47.5%, el modelo TEAM puede ser usado para ajustar el comportamiento de otros memristores como por ejemplo STT – MRAM [25] y Spintrónico [26]. Cabe reiterar que, debido a estas características, el modelo TEAM se utilizó para el desarrollo de aplicaciones analógicas y digitales enfocadas en redes neuronales, las cuales se tratarán en los capítulos 3 y 4, respectivamente.

2.10 Modelo de umbral de voltaje adaptable VTEAM

Resultados experimentales en memristores han mostrado la existencia de un umbral de voltaje [6], [27, 28] en vez de un umbral de corriente, por lo que en [29] Kvatinsky et al. proponen otro modelo denominado VTEAM (**Voltage ThrEshold**

Adaptive Memristor model), para memristores controlados por voltaje, como una extensión del modelo TEAM [24].

El modelo VTEAM establece, de forma similar al modelo TEAM, que la dependencia de la derivada de la variable de estado en el voltaje y en ella misma, se puede modelar como la multiplicación de dos funciones independientes, es decir, una función dependiente de la variable de estado y otra dependiente del voltaje. Bajo esta suposición la ecuación de estado se representa como:

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{v(t)}{v_{off}} - 1 \right)^{\alpha_{off}} f_{off}(x), & 0 < v_{off} < v \\ 0, & v_{on} < v < v_{off} \\ k_{on} \left(\frac{v(t)}{v_{on}} - 1 \right)^{\alpha_{on}} f_{on}(x), & v < v_{on} < 0 \end{cases} \quad (2.50)$$

$$v_{on} < v < v_{off} \quad (2.51)$$

$$v < v_{on} < 0 \quad (2.52)$$

donde k_{off} , k_{on} , α_{off} y α_{on} son parámetros de ajuste, v_{off} y v_{on} son los voltajes de umbral y x es la variable de estado. El parámetro k_{off} es un número positivo mientras que k_{on} es negativo. Las funciones $f_{off}(x)$ y $f_{on}(x)$ son funciones de ventana (2.45) y (2.46), las cuales representan la dependencia en la variable de estado y restringen el valor de x a los límites del dispositivo x_{on} y x_{off} . Al igual que el modelo TEAM se tiene la flexibilidad de poder utilizar cualquier función de ventana [11,12].

La relación corriente - voltaje es indefinida, lo que significa que se puede escoger libremente. Para el caso de una dependencia lineal de la resistencia y la variable de estado x , la relación corriente - voltaje se puede especificar como:

$$v(t) = \left[R_{ON} + \frac{R_{OFF} - R_{ON}}{x_{off} - x_{on}} (x - x_{on}) \right] i(t) \quad (2.53)$$

Para el caso de una dependencia de tipo exponencial de la resistencia y la variable de estado, la relación corriente - voltaje se define con las expresiones (2.54) y (2.55) donde R_{off} y R_{on} son los correspondientes valores resistivos del memristor cuando la variable de estado es x_{off} y x_{on} , respectivamente.

$$i(t) = \frac{e^{-\frac{\lambda}{x_{\text{off}} - x_{\text{on}}}(x - x_{\text{off}})}}{R_{\text{ON}}} v(t) \quad (2.54)$$

$$\lambda = \ln\left(\frac{R_{\text{OFF}}}{R_{\text{ON}}}\right) \quad (2.55)$$

El modelo VTEAM, al igual que el modelo TEAM, es un modelo simple, general y flexible, ya que ajusta con suficiente precisión el comportamiento dinámico de memristores físicos (Pt – Hf – Ti [30], ferroeléctrico [27] y nano cables metálicos [31]) y de otros modelos memristivos (Yakopcic [32], BCM [33] y TEAM [24]). En el **Apéndice A** se comprueba lo anterior con el uso del algoritmo de optimización denominado recocido simulado (***Simulated Annealing***) para el ajuste de los parámetros del modelo VTEAM a la respuesta de los modelos TEAM y Yakopcic. Ambos modelos (TEAM y VTEAM) presentan su descripción en Verilog-A [34], permitiendo de esta manera su uso en el diseño, análisis y simulación en SPICE de circuitos eléctricos.

2.11 Resumen

En este capítulo se exploró de una manera más formal al cuarto elemento fundamental de circuitos denominado, Memristor partiendo desde su concepción teórica realizada en 1971 por el profesor Leon Chua, hasta su descubrimiento en 2008 por los laboratorios de HP como un dispositivo nanométrico de dióxido de titanio TiO_2 , dando pie al estudio de los modelos matemáticos más notorios encontrados en la literatura, los cuales buscan representar el comportamiento dinámico y los fenómenos físicos involucrados en el memristor.

De los modelos estudiados en este capítulo, los modelos lineal y lineal con función de ventana tienen la característica de ser intuitivos y populares, pero no son capaces de representar con precisión el comportamiento dinámico mostrado en un memristor de TiO_2 medido experimentalmente, por lo que el modelo de Pickett es considerado como el más preciso, pero tiene la desventaja de ser complicado y computacionalmente ineficiente. Ya que el modelo del memristor debe distinguirse principalmente por sus características de eficiencia numérica

dentro de plataformas avanzadas de simulación de circuitos integrados CMOS, el modelo TEAM, el cual presenta el mismo comportamiento al modelo de Pickett, pero con expresiones más simples, es el seleccionado en este trabajo de investigación ya que presenta las siguientes características:

Computacionales: permite el desarrollo de sistemas memristivos para señales digitales, analógicas y mixtas, en donde la emulación de este componente puede soportar una amplia gama de tecnologías de fabricación de memorias resistivas basadas en el dióxido de titanio.

Semi-empírico: permite su actualización, tomando en cuenta nuevos materiales nanométricos dentro de los estándares en la industria electrónica.

Modo corriente o voltaje: el modelo TEAM con parámetros de umbral en modo de corriente puede ser extendido a un modelo con parámetros de umbral en modo de voltaje, o VTEAM.

Referencias

- [1] L.O. Chua, Memristor the missing circuit element, IEEE Trans. Circuit Theory. 18 (5) (1971) 507-519.
- [2] A. Adamatzky, L.O. Chua, Memristor networks, Springer, Switzerland, 2014.
- [3] L.O. Chua, S.M. Kang, Memristive devices and systems, Proc. IEEE. 64 (2) (1976) 209-223.
- [4] L.O. Chua, Resistance switching memories are memristors, Appl. Phys. A: Mater. Sci. Process. 102 (4) (2011) 765-783.
- [5] I. Vourkas, G. Ch Sirakoulis, Memristor-based nanoelectronic computing circuit and architectures, Springer, Switzerland, 2016.
- [6] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature. 453 (7191) (2008) 80-83.
- [7] R.S. Williams, How we found the missing memristor, Nature. 453 (7191) (2008) 80-83.

- [8] G. Dearnaley, A.M. Stoneham, D.V. Morgan, Electrical phenomena in amorphous oxide films, *Rep Prog. Phys.* 33 (1970) 1129-1192.
- [9] H. Elgabra, A.H. Farhat, S. Ahmed Al Hosani, D. Homouz, B. Mohammad, Mathematical modeling of a memristor device, 2012 International Conference on Innovations Technology (IIT). 64 (2) (1976) 209-223.
- [10] J. Zha, H. Huang, Y. Liu, A novel window function for memristor model with application in programming analog circuit, *IEEE. Trans. Circuits Syst. II* 63 (5) (2016) 423-427.
- [11] T. Prodromakis, B.P. Peh, C. Papavassiliou, C. Toumazou, A versatile memristor model with nonlinear dopant kinetics, *IEEE. Trans. Electron. Devices* 58 (9) (2011) 3099-3105.
- [12] Y.N. Joglekar, S.J. Wolf, The elusive memristor: properties of basic electrical circuits, *Eur. J. Phys* 30 (2009) 661-675.
- [13] Z. Biolek, D. Biolek, V. Biolkova, SPICE Model of memristor with nonlinear dopant drift, *Radioengineering* 18 (2) (2009) 210-214.
- [14] J.J. Yang, M.D. Pickett, X. Li, D.A.A. Ohlberg, D.R. Stewart, R.S. Williams, Memristive switching mechanism for metal/oxide/metal nanodevices, *Nature Nanotechnol* 3 (2008) 429-433.
- [15] D.B. Strukov, R.S. Williams, Exponential ionic drift: fast switching and low volatility of thin - film memristors, *Appl. Phys. A. Mater. Sci. Process Nature* 94 (3) (2009) 515-519.
- [16] Z. Kolka, V. Biolka, D. Biolek, J. Vavra, Behavioral model for simplified identification of memristor parameters, 2015 International Joint Conference on Neural Networks (IJCNN).
- [17] Y.V. Pershin, M. Di Ventra, Memory effects in complex materials and nanoscale systems, *Adv. Phys* 60 (2) (2011) 145-227.

- [18] M.D. Pickett, D.B. Strukov, J.L. Borghetti, J.J. Yang, G.S. Snider, D.R. Stewart, R.S. Williams, Switching dynamics in titanium dioxide memristive devices, *J. Appl. Phys.* 106 (2009).
- [19] E. Lehtonen, M. Laiho, CNN using memristors for neighborhood connections, 2010 International Workshop on Cellular Nanoscale Networks and their Applications (CNNA).
- [20] J. Simmons, Electric tunnel effect between dissimilar electrodes separated by a thin insulating film, *J. Appl. Phys.* 34 (1963) 2581-2590.
- [21] J. Simmons, Generalized formula for the electric tunnel effect between dissimilar electrodes separated by a thin insulating film, *J. Appl. Phys.* 34 (6) (1963) 1793-1803.
- [22] H. Abdalla, M.D. Pickett, SPICE modeling of memristors, *IEEE Int. Symp. Circuits Syst* (2011) 1832-1835.
- [23] Z. Kolka, V. Biolka, D. Biolek, Improved model of TiO_2 memristor, *Radioengineering* 24 (2) (2015) 378-383.
- [24] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, TEAM: ThrEshold adaptive memristor model, *IEEE Trans. Circ. Syst.* 60 (1) (2013) 211-221.
- [25] X. Wang, Y. Chen, H. Xi, D. Dimitrov, Spintronic memristor through spin-torque-induced magnetization motion, *IEEE Electron Device Lett.* 30 (3) (2009) 294-297.
- [26] T. Kawahara, 2 Mb SPRAM (spin-transfer torque RAM) with Bit-by-Bit Bi-directional current write and parallelizing-direction current read, *IEEE J. Solid-State Circuits.* 43 (1) (2008) 109-120.
- [27] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, H. Yamada, C. Deranlot, N.D. Mathur, M. Bibes, A. Barthélémy, J. Grollier, A ferroelectric memristor, *Nature. Mater.* 11 (10) (2010) 860-864.

- [28] D. Liu, H. Cheng, X. Zhu, G. Wang, N. Wang, Analog memristor based on thickening/thinning of Ag nanofilaments in amorphous manganite thin films, *ACS Appl. Mater. Interfaces* 5 (21) (2013) 11258-11264.
- [29] S. Kvatinsky, M. Ramadan, E.G. Friedman, A. Kolodny, VTEAM: a general model for voltage-controlled memristors, *IEEE Trans. Circ. Syst. II* 62 (8) (2015) 786-790.
- [30] E. Yalon, A. Gavrilov, S. Cohen, D. Mistele, B. Meyler, J. Salzman, D. Ritter, Resistive switching in probed by a metal-insulator-semiconductor bipolar transistor, *Electron Device Lett.* 33 (1) (2012) 11-13.
- [31] S.L. Jonhson, A. Sundararajan, D.P. Hunley, D.R. Strachan, Memristive switching of single-component metallic nanowires, *Nanotechnology* 21 (12) (2010).
- [32] C. Yakopcic, T.M. Taha, G. Subramanyam, R.E. Pino, S. Rogers, A memristor device model, *IEEE Electron Device Lett.* 32 (10) (2011) 1436-1438.
- [33] F. Corinto, A. Ascoli, A boundary condition-based approach to the modeling of memristor nanostructures, *IEEE Trans. Circuit Syst. I* 59 (11) (2012) 2713-2726.
- [34] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. Weiser, E.G. Friedman, Models of memristors for SPICE simulations, 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel.

Capítulo 3

Red neuronal recurrente memristiva

3.1 Introducción

Hoy en día el memristor representa un interesante objeto de estudio desde su fabricación, a partir de distintos materiales, hasta la introducción de modelos con base en sistemas complejos [1, 2]. Además, está previsto que en la actividad del diseño electrónico se presenten grandes posibilidades de crear sistemas digitales y analógicos emergentes [3], donde el memristor podría ser introducido como un elemento innovador. En el contexto académico, existen diseños electrónicos orientados en concebir sistemas analógicos basados en la configuración de red de Hopfield, donde los tradicionales elementos de ponderación resistiva pueden ser remplazados por el memristor. Este remplazo puede ser implementado usando tecnología electrónica estándar. En esta dirección, los trabajos [4, 5] tratan con dos prototipos demostrativos: un convertidor ADC de 4-bits y una memoria asociativa. Tomando en particular el modelo de Hopfield como un optimizador, este se puede analizar con base en la manera en que se establece su función de energía o costo. La eficiencia para resolver problemas de programación lineal y no lineal [6] es un tema alternativo en el estudio del paradigma de Hopfield. Otra red neuronal recurrente que trabaja como un optimizador y es muy atractiva para la integración en sistemas VLSI es la red de Wang [7], la cual trabaja con resistencias de un solo valor y con base en un patrón de conectividad lateral entre neuronas.

En este capítulo se presenta la solución electrónica a dos problemas de optimización denominados: asignación y transporte, a partir de una red neuronal recurrente memristiva. Esta red trabaja como un optimizador y presenta un

comportamiento similar a la red de Wang. La arquitectura de la red propuesta se basa en el diseño de circuitos híbridos CMOS-memristor utilizando tecnología estándar CMOS pozo n de 0.5 μm . El memristor se utilizó para obtener algunas funciones analógicas tales como: 1) mantener el estado resistivo Off para una fuente de corriente de valor constante, 2) combinación de los estados resistivos ON y OFF en un grupo de memristores para el desarrollo de una fuente de corriente dinámica, cuyo valor decae con el tiempo y 3) selección dinámica de uno de dos memristores para la emulación de un resistor bidireccional de valor constante. Todas ellas representan parámetros dinámicos y estáticos de la red neuronal recurrente de Wang. Además, se presenta el diseño de circuitos CMOS como son: amplificador de corriente de ganancia unitaria e inversores, que permiten representar una función sigmoideal invertida. La conexión entre neuronas se realizó con arreglos de dos memristores, los cuales observan el sentido de la corriente. El modelo memristivo utilizado fue el TEAM [8], el cual representa el comportamiento dinámico de un memristor de conmutación rápida bipolar de TiO_2 . Para verificar la respuesta de la red neuronal memristiva se realizaron simulaciones de las respuestas eléctrica y analítica en Spice y Simulink, donde al comparar ambas respuestas se obtuvieron resultados similares.

3.2 Problema de Asignación

El problema de asignación es un problema clásico de optimización combinatoria, el cual tiene una amplia variedad de aplicaciones como son: asignación de trabajo, diseño de aplicaciones, programación de producción, y diseño de placas de circuito impreso [9-11]. Este problema se centra en la necesidad de asignar cierto número de recursos o entidades a un cierto número de agentes o celdas, con el objetivo de realizar ciertas tareas a un mínimo costo.

Matemáticamente, el problema de asignación se representa de la siguiente manera:

$$\text{minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} v_{i,j} \quad (3.1)$$

$$\text{sujeto a } \sum_{j=1}^n v_{i,j} = 1, \quad i = 1, \dots, n \quad (3.2)$$

$$\sum_{i=1}^n v_{i,j} = 1, \quad j = 1, \dots, n \quad (3.3)$$

$$v_{i,j} \in \{0,1\}, \quad i, j = 1, \dots, n \quad (3.4)$$

donde $v_{i,j}$ es la variable de decisión y $c_{i,j}$ es el costo de asignar la entidad i a la celda j . La entidad i es asignada a la celda j , si y solo si la variable de decisión es $v_{i,j} = 1$. El número de variables de decisión presentes en un problema está dado por n^2 , mientras que el número de restricciones lo está por $2n$. La función objetivo a minimizar es el costo total por la asignación (3.1). Las restricciones dadas por las expresiones (3.2) – (3.4) garantizan que una entidad dada puede ser únicamente asignada a una celda e inversamente que una celda pueda aceptar únicamente a una entidad.

El problema de asignación descrito por las expresiones (3.1) – (3.4) se puede representar en forma matricial como:

$$\text{minimizar } z = \mathbf{c}^T \mathbf{v} \quad (3.5)$$

$$\text{sujeto a } \mathbf{Av} = \mathbf{b} \quad (3.6)$$

donde $\mathbf{c}^T = [c_{11}, c_{12}, \dots, c_{1n}, c_{21}, c_{22}, \dots, c_{2n}, c_{n1}, c_{n2}, \dots, c_{nn}]$ es el vector de costos, $\mathbf{v} = [v_{11}, v_{12}, \dots, v_{1n}, v_{21}, v_{22}, \dots, v_{2n}, v_{n1}, v_{n2}, \dots, v_{nn}]^T$ es el vector decisión, \mathbf{b} es un vector de tamaño $2n$ conformado de unos y la matriz \mathbf{A} se define como:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{B}_1 & \mathbf{B}_2 & \dots & \mathbf{B}_n \end{bmatrix} \quad (3.7)$$

\mathbf{I} es la matriz identidad, \mathbf{B}_i es una matriz de ceros excepto la i th fila, la cual contiene unos. Estas matrices tienen un tamaño de $n \times n$.

3.2.1 Red neuronal de Wang para el problema de asignación

El problema de asignación ha sido investigado ampliamente por mucho tiempo. Además de los métodos clásicos que resuelven el problema, tales como el método simplex y el método húngaro, muchos otros métodos han sido desarrollados [9, 10]. Los métodos existentes tienen la desventaja de no ser tan efectivos y eficientes en aplicaciones de gran escala tales como la asignación del personal en maniobras militares y en aplicaciones en tiempo real, tales como la visión artificial en robótica, esto se debe principalmente a que dichos métodos tienen una naturaleza de procesamiento de tipo secuencial. Por tal motivo, métodos con solución paralela, como las redes neuronales, son más deseables. Una red neuronal consiste en una serie de neuronas, conectadas entre sí por medio de pesos sinápticos; estas neuronas operan simultáneamente para realizar procesamiento en paralelo. Varios modelos de redes neuronales han sido propuestos para resolver el problema de asignación [11-14]. Particularmente, el profesor Wang, basándose en [15], propone una red neuronal recurrente [11], la cual consiste en n^2 neuronas ordenadas espacialmente en forma de una matriz cuadrada (**ver la Figura 3.1**). El estado estable de las neuronas representa la solución del problema de optimización. Wang estableció la siguiente ecuación diferencial (3.8) para describir la dinámica de la red neuronal.

Para $i, j = 1, 2, \dots, n$

$$\frac{du_{ij}(t)}{dt} = -\eta \sum_{k=1}^n v_{ik}(t) - \eta \sum_{l=1}^n v_{lj}(t) + 2\eta - \lambda c_{ij} e^{-\frac{t}{\tau}} \quad (3.8)$$

$$v_{ij} = g[u_{ij}(t)] \quad (3.9)$$

Donde k y l son contadores de filas y columnas, respectivamente. t es la variable tiempo. η y τ son parámetros escalares positivos, cuyo valor es escogido de forma empírica. λ es un parámetro, cuyo valor es $1/\max(c_{i,j})$. c_{ij} representa el conjunto de costos del problema y su valor decae con el tiempo debido a $\exp(-t/\tau)$. La ecuación (3.8) muestra que todos los pesos de auto conexión son 2η , todos los pesos de conexión laterales son η y todos los umbrales de polarización son 2η . La ecuación (3.9) representa la función de activación, cuyo único requisito es la no negatividad. Una propuesta para este trabajo se presenta en (3.10)

$$v_{ij} = \frac{1}{(1 + e^{-\beta u_{ij}})} \quad (3.10)$$

En la **Figura 3.1** se muestra la arquitectura de la red neuronal recurrente propuesta por Wang. Como se puede observar, la salida $v_{i,j}$ de cada neurona $N_{i,j}$ se conecta de forma lateral con las entradas de todas las neuronas ubicadas en su misma fila y columna, además de conectarse con ella misma.

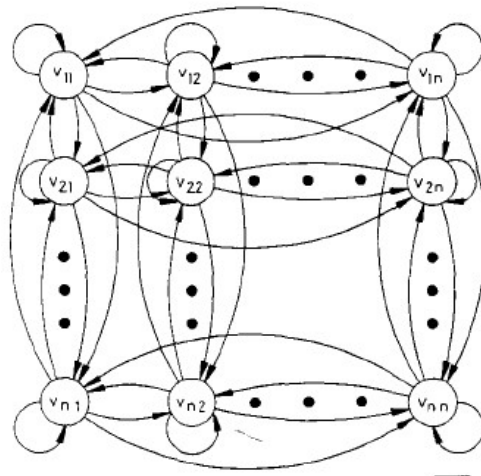


Figura 3.1 Arquitectura de la red neuronal recurrente.

Las expresiones (3.8) y (3.9) se pueden representar en forma matricial como:

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\mathbf{v} - \boldsymbol{\theta}) - \lambda\mathbf{c}e^{-\frac{t}{\tau}} \quad (3.11)$$

$$\mathbf{x} = g[\mathbf{u}(t)] \quad (3.12)$$

Los vectores \mathbf{v} y \mathbf{c} se definen en (3.5) y (3.6), el vector umbral $\boldsymbol{\theta}$ de tamaño $n^2 \times 1$ se define como $\boldsymbol{\theta} = \mathbf{A}^T \mathbf{b} = [2 \ 2 \ 2 \ \dots \ 2]^T$, $\mathbf{u}(t)$ es el vector estado de tamaño $n^2 \times 1$ definido por $\mathbf{u}(t) = [u_{11}, u_{12}, \dots, u_{1n}, u_{21}, u_{22}, \dots, u_{2n}, u_{n1}, u_{n2}, \dots, u_{nn}]^T$. \mathbf{W} es la matriz de pesos de tamaño $n^2 \times n^2$ (3.13), donde \mathbf{I} es la matriz identidad $n \times n$ y \mathbf{P} es una matriz $\mathbf{P} = \mathbf{I} + 1$.

$$\mathbf{W} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} \mathbf{P} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{I} & \mathbf{P} & \mathbf{I} & \dots & \mathbf{I} \\ \vdots & \mathbf{I} & \mathbf{P} & \mathbf{I} & \mathbf{I} \\ \vdots & \vdots & \mathbf{I} & \ddots & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{P} \end{bmatrix} \quad (3.13)$$

Wang realizó la implementación analógica de la red neuronal utilizando amplificadores operacionales, resistores y capacitores. Tiene la desventaja de no ser sencilla e inconveniente para la solución de problemas de gran escala debido a la gran cantidad de interconexiones requeridas y la dificultad de programar los parámetros. En el **Apéndice B** se muestra el desarrollo matemático de las expresiones anteriores para la solución de una red neuronal de 3 x 3, utilizando la herramienta de desarrollo de MATLAB Simulink.

3.3 Problema de Transporte

El problema de transporte, también llamado de distribución, es un problema de programación lineal, cuya descripción puede ser asociada con la siguiente definición [9, 10, 16]. Existe la necesidad de transportar unidades o recursos de un sitio llamado fuente a otro sitio llamado destino, pero con la condición de minimizar el costo de transportación y al mismo tiempo satisfacer las restricciones de oferta y demanda.

Matemáticamente el problema de transporte se representa de la siguiente manera:

$$\text{minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} v_{i,j} \quad (3.14)$$

$$\text{sujeto a } \sum_{j=1}^n v_{i,j} = a_i, \quad i = 1, \dots, n \quad (3.15)$$

$$\sum_{i=1}^n v_{i,j} = b_j, \quad j = 1, \dots, n \quad (3.16)$$

$$v_{i,j} \in \{0,1\}, \quad i, j = 1, \dots, n \quad (3.17)$$

Aplicando un proceso de normalización con $q = \max\{a_i, b_i : i = 1, \dots, n\}$ $v_{ij} = q\bar{v}$, $a_i = q\bar{a}$ y $b_i = q\bar{b}$ las expresiones (3.14) – (3.17) se pueden reescribir como:

$$\text{minimizar } \bar{z} = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} \bar{v}_{i,j} \quad (3.18)$$

$$\text{sujeto a } \sum_{j=1}^n \bar{v}_{i,j} = \bar{a}_i, \quad i = 1, \dots, n \quad (3.19)$$

$$\sum_{i=1}^n \bar{v}_{i,j} = \bar{b}_j, \quad j = 1, \dots, n \quad (3.20)$$

$$\bar{v}_{i,j} \in \{0,1\}, \quad i, j = 1, \dots, n \quad (3.21)$$

El problema de transporte descrito por las expresiones (3.18) – (3.21) se representa en forma matricial como:

$$\text{minimizar } \bar{z} = \mathbf{c}^T \bar{\mathbf{v}} \quad (3.22)$$

$$\text{sujeto a } \mathbf{A} \bar{\mathbf{v}} = \mathbf{d} \quad (3.23)$$

donde $\mathbf{c}^T = [c_{11}, c_{12}, \dots, c_{1n}, c_{21}, c_{22}, \dots, c_{2n}, c_{n1}, c_{n2}, \dots, c_{nn}]$ es el vector de costos, $\bar{\mathbf{v}} = [\bar{v}_{11}, \bar{v}_{12}, \dots, \bar{v}_{1n}, \bar{v}_{21}, \bar{v}_{22}, \dots, \bar{v}_{2n}, \bar{v}_{n1}, \bar{v}_{n2}, \dots, \bar{v}_{nn}]^T$ es el vector decisión, el vector $\mathbf{d} = [\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n, \bar{b}_1, \bar{b}_2, \dots, \bar{b}_n, 0_1, 0_2, \dots, 0_n]^T$ de dimensión $3n$ y \mathbf{A} es una matriz de dimensión $3n \times n^2$ definida como:

$$A = \begin{bmatrix} B_1 & B_2 & \cdots & B_n \\ I & I & \cdots & I \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (3.24)$$

I es la matriz identidad, B_i es una matriz de ceros excepto la i th fila, la cual contiene unos y 0 es la matriz nula. Estas matrices tienen un tamaño de $n \times n$.

3.3.1 Red neuronal de Wang para el problema de transporte

Realizando algunos cambios en la formulación matricial de la ecuación propuesta por Wang (3.8), se puede utilizar la misma red neuronal recurrente para resolver varios problemas de optimización [7,11,15,17]. Por tal motivo, la arquitectura de esta red (**ver la Figura 3.1**) puede ser empleada para resolver el problema de transporte. En [16] se propone la siguiente ecuación diferencial (3.25) para describir la dinámica de las neuronas y se emplea la misma función de activación $g[.]$ descrita en (3.10).

$$\frac{du_{ij}(t)}{dt} = -\eta \sum_{k=1}^n \bar{v}_{ik}(t) - \eta \sum_{l=1}^n \bar{v}_{lj}(t) + \eta \theta_{ij} - \lambda c_{ij} e^{-\frac{t}{\tau}} \quad (3.25)$$

$$\bar{v}_{ij} = g[u_{ij}(t)] \quad (3.26)$$

Donde k y l son contadores de filas y columnas respectivamente. t es la variable tiempo. η y τ son parámetros escalares positivos, cuyo valor es escogido de forma empírica. λ es un parámetro, cuyo valor es $1/\max(c_{i,j})$. c_{ij} representa el conjunto de costos del problema y su valor decae con el tiempo debido a $\exp(-t/\tau)$. La ecuación (3.25) muestra que todos los pesos de auto conexión son 2η , todos los pesos de conexión laterales son η y todos los umbrales de polarización son $\eta\theta_{ij}$.

Ambas expresiones, (3.25) y (3.26), se pueden representar en forma matricial como:

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\bar{\mathbf{v}} - \boldsymbol{\theta}) - \lambda \mathbf{c} e^{-\frac{t}{\tau}} \quad (3.27)$$

$$\mathbf{x} = g[\mathbf{u}(t)] \quad (3.28)$$

Vectores $\bar{\mathbf{v}}$ y \mathbf{c} se definen en (3.22) y (3.23), el vector umbral $\boldsymbol{\theta}$ de tamaño $n \times n$ se define como $\boldsymbol{\theta} = \mathbf{A}^T \mathbf{d}$, $\mathbf{u}(t) = [u_{11}, u_{12}, \dots, u_{1n}, u_{21}, u_{22}, \dots, u_{2n}, u_{n1}, u_{n2}, \dots, u_{nn}]^T$ es el vector estado de tamaño $n^2 \times 1$. \mathbf{W} es la matriz de pesos de tamaño $n^2 \times n^2$ (3.29), donde \mathbf{I} es la matriz identidad $n \times n$ y \mathbf{P} es una matriz $\mathbf{P} = \mathbf{I} + 1$.

$$\mathbf{W} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} \mathbf{P} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{I} & \mathbf{P} & \mathbf{I} & \dots & \mathbf{I} \\ \vdots & \mathbf{I} & \mathbf{P} & \mathbf{I} & \mathbf{I} \\ \vdots & \vdots & \mathbf{I} & \ddots & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{P} \end{bmatrix} \quad (3.29)$$

En el **Apéndice C** se muestra el desarrollo matemático de las expresiones anteriores para la solución de una red neuronal de 3×3 , utilizando la herramienta de desarrollo de MATLAB Simulink.

3.4 Configuración eléctrica de una neurona

A continuación, se muestran los componentes analógicos que conforman la red neuronal recurrente memristiva, la cual resuelve los problemas de optimización: asignación y transporte. En la **Figura 3.2** se muestran los componentes eléctricos de procesamiento que conforman la neurona eléctrica N_{ij} . Los factores de ponderación W_0 o pesos sinápticos se representan a partir de conductancias $1/R_0$, donde el conjunto de resistores proporciona una corriente, la cual se suma con la fuente de corriente de ajuste I_{set} , la fuente de corriente de umbral $I\theta_{i,j}$ y la fuente de corriente de costos $Ic_{i,j}$, para de esta manera formar la corriente resultante $Iz_{i,j}$. Esta variable analógica es la entrada de un amplificador de corriente con ganancia unitaria, el cual copia el voltaje V_{ref} a la terminal de entrada y proporciona al capacitor C_{INT} una réplica de la corriente $Iz_{i,j}$. Este capacitor integra $Iz_{i,j}$ con el fin de obtener el voltaje $Vu_{i,j}$, el cual es leído por una función sigmoideal invertida para de esta manera obtener el voltaje de salida $Vv_{i,j}$ de la neurona N_{ij} .

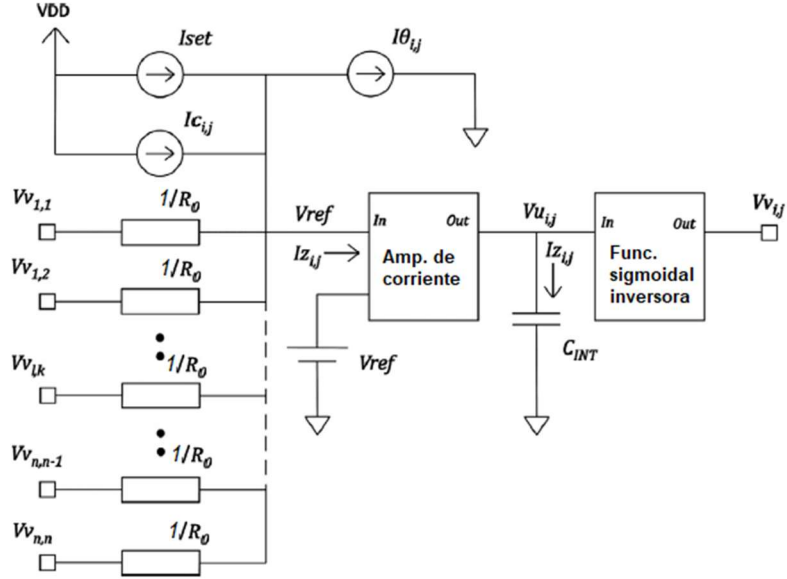


Figura 3.2 Configuración eléctrica de la neurona $N_{i,j}$.

A continuación, se presenta la descripción analítica del sistema eléctrico donde se obtiene una ecuación diferencial equivalente a la propuesta por Wang (3.8). Aplicando la ley de corrientes de Kirchhoff en la entrada del amplificador de corriente se tiene la siguiente expresión para $I_{z_{i,j}}$.

$$I_{z_{i,j}} = \frac{1}{R_0} \sum_{l,k=1}^n (V_{V_{l,k}} - V_{ref}) + I_{c_{i,j}} - I_{\theta_{i,j}} + I_{set} \quad (3.30)$$

$$I_{z_{i,j}} = \frac{1}{R_0} \sum_{l,k=1}^n V_{V_{l,k}} - \frac{2nV_{ref}}{R_0} + I_{c_{i,j}} - I_{\theta_{i,j}} + I_{set} \quad (3.31)$$

El término $-(2nV_{ref})/R_0$ actúa como un voltaje de polarización y su efecto debe ser reducido a cero al ajustar I_{set} con (3.32) y por lo tanto simplificando (3.31) en (3.33).

$$I_{set} = \frac{2nV_{ref}}{R_0} \quad (3.32)$$

$$I_{z_{i,j}} = \frac{1}{R_0} \sum_{l,k=1}^n V_{V_{l,k}} + I_{c_{i,j}} - I_{\theta_{i,j}} \quad (3.33)$$

Se debe notar que la sumatoria contiene una auto conexión, cuyo valor es $2/R_0$. Como se mencionó anteriormente el capacitor C_{INT} integra $I_{z_{i,j}}$ para producir el voltaje $V_{u_{i,j}}$

$$\frac{dV_{u_{i,j}}}{dt} = \frac{1}{C_{INT}} \left\{ \frac{1}{R_0} \sum_{k=1}^n V_{v_{i,k}} + \frac{1}{R_0} \sum_{l=1}^n V_{v_{l,j}} + I_{c_{i,j}} - I_{\theta_{i,j}} \right\} \quad (3.34)$$

Donde (3.34) es equivalente a la ecuación diferencial propuesta por Wang (3.8) pero con signos contrarios, esto es tomado en cuenta con la función sigmoideal invertida. $I_{\theta_{i,j}}$ es una fuente de corriente de valor constante equivalente a los parámetros 2η y $\eta\theta_{ij}$ de los problemas de asignación (3.8) y transporte (3.25), respectivamente. $I_{c_{i,j}}$ es una fuente de corriente, cuyo valor decae con el tiempo y emula el comportamiento del parámetro $\lambda c_{ij} e^{-\frac{t}{\tau}}$, cabe mencionar que este parámetro es el mismo para ambos problemas.

3.5 El memristor

El modelo de memristor que se utilizó para el desarrollo de la red neuronal memristiva fue el TEAM [8], con función de ventana de Kvatinsky y la relación $i - v$ de tipo no lineal, esto debido a las características mencionadas en el capítulo 2. La implementación del modelo TEAM se realizó en Verilog-A con los códigos propuestos en [18] (**ver Apéndice D**) y simulado en la Suite de diseño IC-Nanometer de Mentor. En este trabajo, el uso del memristor como resistencia fija (estado estático) y como resistencia variable (estado dinámico) es la base del diseño de la red neuronal memristiva. El estado estático se representa como una resistencia fija de valor R_{ON} o R_{OFF} , mientras que el estado dinámico se entiende como una resistencia dinámica R_M controlada por la expresión de la evolución de la variable de estado x en el tiempo dx/dt . En las **Figuras 3.3a** y **c** el estado resistivo inicial R_M se encuentra dentro del intervalo $R_{ON} < R_M < R_{OFF}$; al fluir la corriente I_m hacia el lado derecho, la resistencia inicial alcanza el estado estático R_{ON} (**ver la Figura 3.3b**), en el caso contrario cuando I_m fluye en la dirección opuesta R_M tiende a R_{OFF} (**ver la Figura 3.3d**).

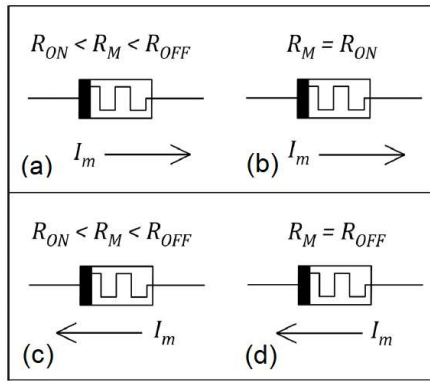


Figura 3.3 Estados dinámico y estático del memristor

En la **Tabla 3.1** se presentan los valores de los parámetros del modelo TEAM utilizados en todas las simulaciones eléctricas. Estos valores están enfocados en memristores cuya tecnología está orientada en la operación de circuitos digitales de conmutación rápida [19].

Tabla 3.1 Parámetros del modelo TEAM

Parámetro	Valor	Comentario
dt	1×10^{-9} s	Diferencial de tiempo
R_{ON}	1×10^3 Ω	Memristancia en $x = 0$ m
R_{OFF}	500×10^3 Ω	Memristancia en $x = 3 \times 10^{-9}$ m
i_{off}	50×10^{-12} A	Corriente de umbral
i_{on}	-50×10^{-12} A	Corriente de umbral
w_c	107×10^{-12} m	Parámetro de ajuste
a_{on}	0 m	Parámetro de ajuste
a_{off}	3×10^{-9} m	Parámetro de ajuste
k_{on}	-1×10^{-8} m/s	Parámetro de ajuste
k_{off}	1×10^{-8} m/s	Parámetro de ajuste
α_{on}	1	Parámetro de ajuste
α_{off}	1	Parámetro de ajuste
x_{on}	0 m	Límite inferior de x
x_{off}	3×10^{-9} m	Límite superior de x
x_{init}	1.5×10^{-9} m	Condición inicial de la variable de estado

3.6 Circuitos CMOS analógicos

A continuación, se presenta la descripción de los circuitos que conforman la neurona eléctrica $N_{i,j}$ (ver la Figura 3.2). Como ya se especificó, estos circuitos se simularon usando la tecnología estándar CMOS pozo n de $0.5 \mu\text{m}$ y los parámetros eléctricos del modelo BSIM3v3 MOSFET de la tecnología CMOS se obtuvieron de ON-semiconductor para después ser ejecutados en la Suite de diseño IC-Nanometer de Mentor. La relación W/L que se empleó en todos los transistores involucrados en los diseños eléctricos es $\mu\text{m}/\mu\text{m}$.

3.6.1 Amplificador de corriente

El amplificador de corriente es útil para aplicaciones de procesamiento de señales en diseños de modo de corriente [20]. En la Figura 3.4 se muestra su diseño, con los tamaños de los transistores; funciona siguiendo el principio translineal del bucle formado por los transistores M1, M2, M3 y M4 transfiriendo V_{ref} a la entrada. Este amplificador está diseñado para proporcionar una ganancia unitaria.

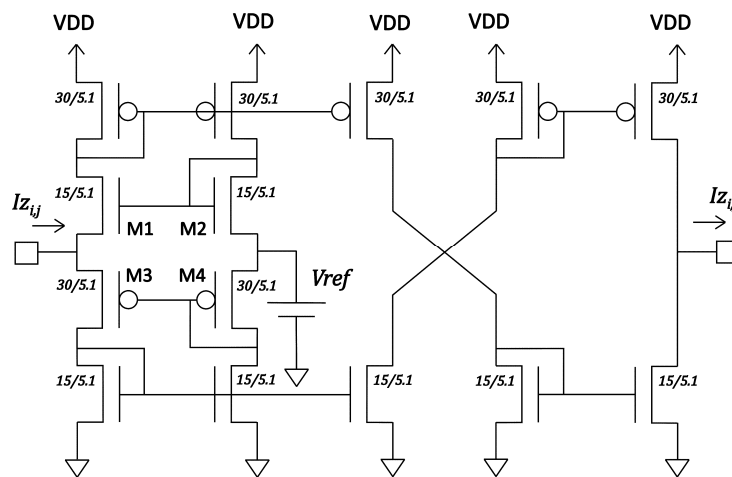


Figura 3.4 Circuito eléctrico de un amplificador de corriente de ganancia unitaria

3.6.2 Función sigmoial invertida

La **Figura 3.5a** muestra un amplificador analógico llamado push-pull, cuya ganancia y función de transferencia son ajustados por el tamaño de los transistores; este aproxima la función sigmoial invertida presentada en (3.10). La **Figura 3.5b** presenta la función de transferencia del amplificador push-pull.

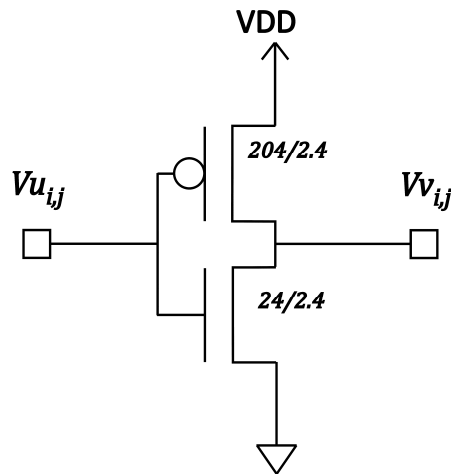


Figura 3.5a. Circuito amplificador push-pull.

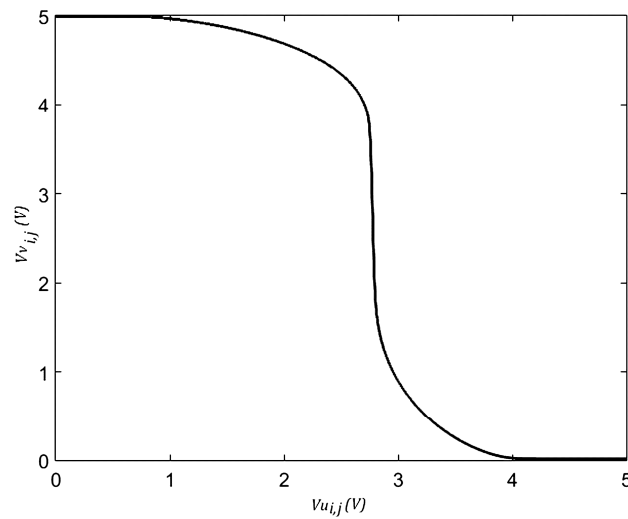


Figura 3.5b. Simulación en DC de la función sigmoial invertida.

3.6.3 Resistor de valor constante

En la **Figura 3.2**, la interconectividad entre neuronas se realiza por medio de un conjunto de memristores, cuyo número total por neurona es determinado por la expresión $2(n - 1) + 2$, donde n es el tamaño del sistema a resolver. Todos los memristores tienen el valor $R_0 = R_{OFF}$. Para mantener este estado resistivo durante todo el proceso de cálculo, el sentido de la corriente nunca debe de cambiar en ningún memristor, $V_{V_{l,k}} > V_{ref}$ en todo momento, sin embargo, se observó que, debido a la dinámica del sistema, el sentido de la corriente cambiaba en algunos memristores ocasionando un cambio resistivo en estos. Para solucionar este problema se propuso un circuito memristivo capaz de emular un resistor bidireccional de valor constante R_{OFF} para la conexión entre los voltajes $V_{V_{l,k}}$ y V_{ref} (**ver la Figura 3.6**).

Existen dos memristores, MEMR1 y MEMR2 con valor inicial $R_0 = R_{OFF}$. Estos memristores son seleccionados de forma alterna por medio de dos compuertas de transmisión TG1 y TG2, respectivamente. Ambas compuertas son controladas por los niveles lógicos V_a y V_b de un comparador conformado por una cadena de cuatro inversores, cuyo umbral por diseño es V_{ref} .

Ambos memristores trabajan siempre en su estado resistivo R_{OFF} . Cuando se presenta la condición $V_{V_{l,k}} > V_{ref}$, la compuerta TG1 se activa permitiendo el flujo de corriente del potencial mayor $V_{V_{l,k}}$ al potencial menor V_{ref} , evitando de esta manera que el estado inicial de MEMR1 cambie. Cuando $V_{V_{l,k}} < V_{ref}$ la compuerta TG2 se activa permitiendo el flujo de corriente del potencial mayor V_{ref} al potencial menor $V_{V_{l,k}}$ evitando de esta manera que el estado inicial de MEMR2 cambie. De esta manera, siempre se tiene el valor de R_{OFF} en todos los memristores.

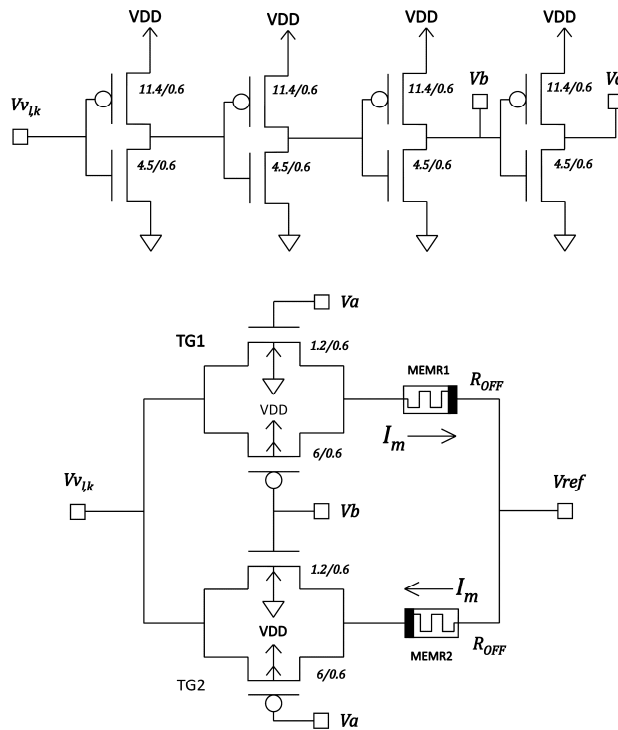


Figura 3.6 Circuito de un resistor bidireccional de valor constante

3.6.4 Fuente de corriente constante

Las fuentes de corriente $I_{\theta_{i,j}}$ y I_{set} se diseñaron siguiendo la misma estrategia, la cual nos permite reducir el número de elementos activos en el diseño, mejorando de esta manera el área del chip. Esta estrategia consiste en aprovechar el tamaño nanométrico y la característica resistiva del memristor con transistores CMOS. Ambas fuentes de corriente siguen una configuración circuital, además de tener la característica de proporcionar un valor de corriente constante invariante en el tiempo. Las Figuras 3.7a y 3.8a muestran la implementación de los circuitos para $I_{set}/2$ y $I_{\theta_{i,j}}$, respectivamente. La fuente de corriente I_{set} se implementó usando dos circuitos conectados en paralelo, donde cada uno proporcionan un valor de corriente $I_{set}/2$. Ambas configuraciones utilizan un conjunto de memristores, en estado R_{OFF} , conectados en paralelo.

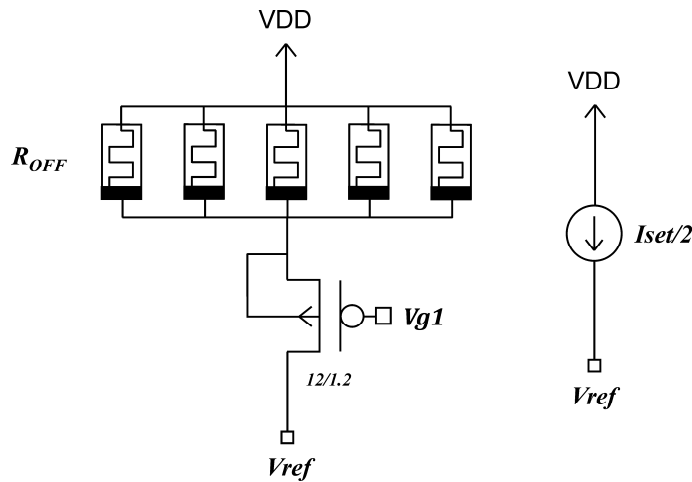


Figura 3.7a Fuente de corriente $I_{set}/2$, compuesta por 5 memristores en R_{OFF} en paralelo, conectados en serie con un transistor MOS canal p. El voltaje analógico V_{g1} ajusta el valor de $I_{set}/2$.

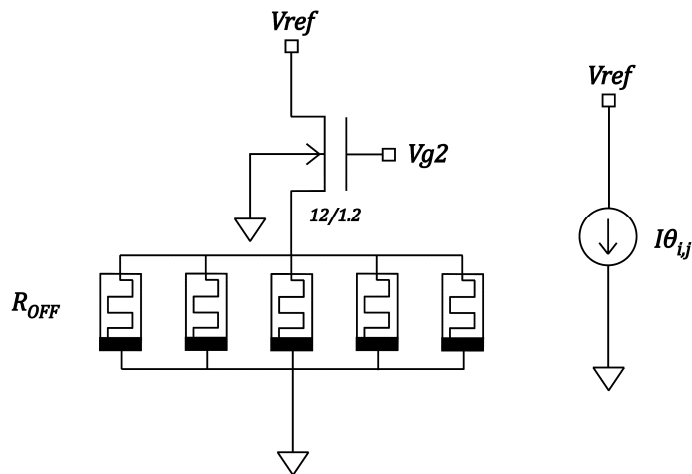


Figura 3.8a Fuente de corriente $I_{\theta_{i,j}}$, compuesta por 5 memristores en R_{OFF} en paralelo, conectados en serie con un transistor MOS canal n. El voltaje analógico V_{g2} ajusta el valor de $I_{\theta_{i,j}}$.

Los voltajes de compuerta V_{g1} y V_{g2} son utilizados para determinar las corrientes en DC $I_{set}/2$ y $I_{\theta_{i,j}}$ deseadas, como se aprecia en las **Figuras 3.7b** y **3.8b**.

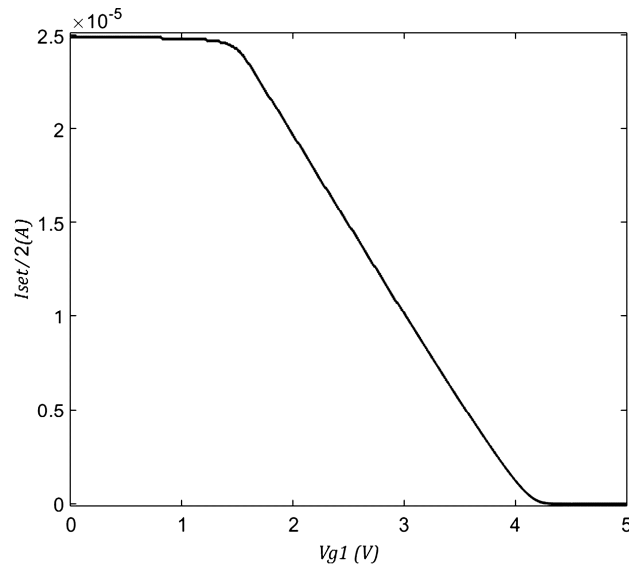


Figura 3.7b Simulación en DC de $I_{set}/2$ versus V_{g1} . La característica lineal permite ajustar el voltaje V_{g1} en el intervalo $1.7\text{ V} < V_{g1} < 3.85\text{ V}$ para valores de corriente dentro del intervalo $1.0\text{ }\mu\text{A} < I_{\theta_{i,j}} < 23.0\text{ }\mu\text{A}$.

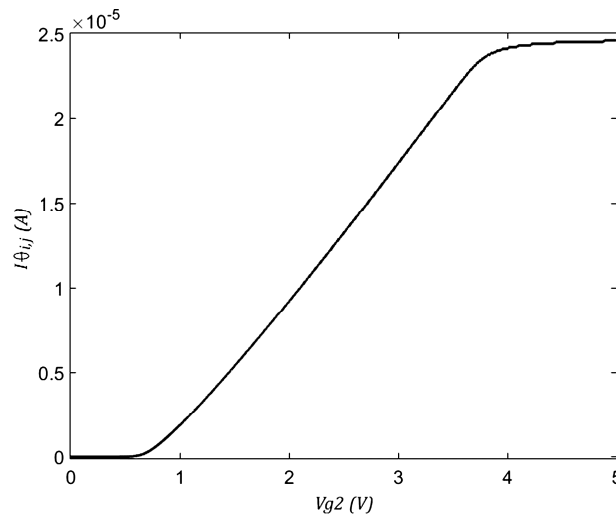


Figura 3.8b Simulación en DC de $I_{\theta_{i,j}}$ versus V_{g2} . La característica lineal permite ajustar el voltaje V_{g2} en el intervalo $1.0\text{ V} < V_{g2} < 3.70\text{ V}$ para valores de corriente dentro del intervalo $1.0\text{ }\mu\text{A} < I_{\theta_{i,j}} < 23.0\text{ }\mu\text{A}$.

Observando las **Figuras 3.7a** y **3.8a**, se aprecia que ambas fuentes de corriente tienen la característica de tener una alta resistencia de salida (**ver las Figuras 3.7c** y **3.8c**).

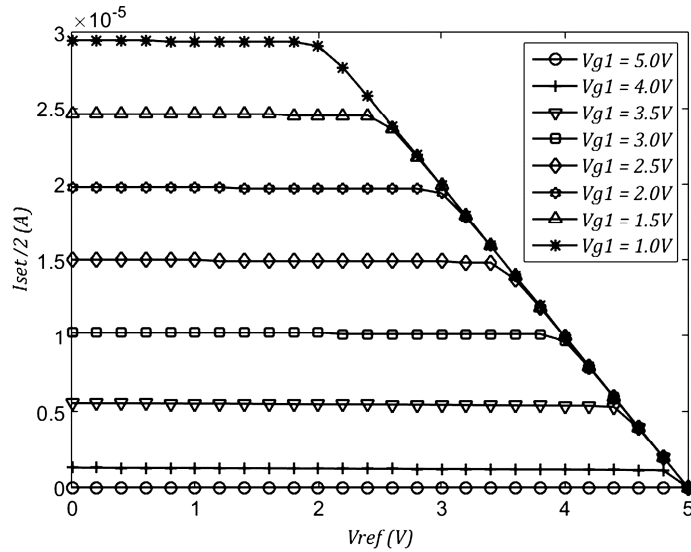


Figura 3.7c Simulación en DC de $I_{set}/2$ versus V_{ref} , con V_{g1} como parámetro. Se escogieron para la red neuronal los siguientes valores: $V_{g1} = 2.5$ V para una corriente $I_{set}/2 = 15 \mu A$ y $V_{ref} = 2.5$ V.

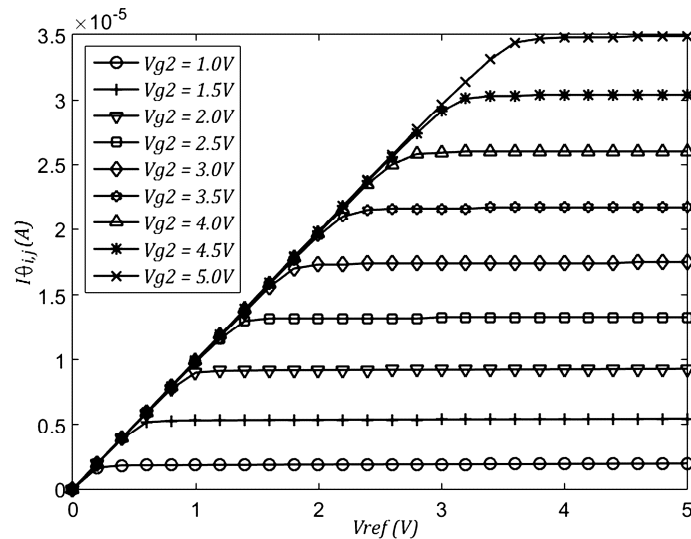


Figura 3.8c Simulación en DC de $I\theta_{i,j}$ versus V_{ref} , con V_{g2} como parámetro. V_{g2} debe ser escogido en el intervalo 0.8 V $< V_{g2} < 3.5$ V para valores de corriente en el intervalo $0 \mu A < I\theta_{i,j} < 20 \mu A$ con $V_{ref} = 2.5$ V.

3.6.5 Fuente de corriente dinámica

Con base en los estados estáticos y dinámicos del memristor, se propone una configuración de tipo serie paralelo, la cual permite emular una corriente transitoria, cuyo valor inicial o valor pico I_{MS} representa el valor inicial del parámetro $\lambda c_{ij} e^{-\frac{t}{\tau}}$ de la ecuación de Wang (3.8). En la **Figura 3.9**, se presenta un diagrama de dicha configuración, donde la diferencia de potencial $(V+) - (V-)$ cambia de cero a un valor V_M en el tiempo T_{START} . V_M es un voltaje que debe ser ajustado para tener el valor deseado de la corriente inicial I_{MS} . Antes de que ocurra el transitorio, el estado resistivo de los memristores conectados en serie es R_{ON} , mientras que los conectados en paralelo tiene el valor de R_{OFF} . La corriente I_m que fluye a través de los memristores conectados en serie provoca que estos conmuten hacia R_{OFF} , mientras que el estado resistivo inicial de los memristores en paralelo se mantiene en R_{OFF} . Esto provoca un efecto transitorio donde el valor inicial de la corriente I_{MS} decae en el tiempo hacia I_{MF} .

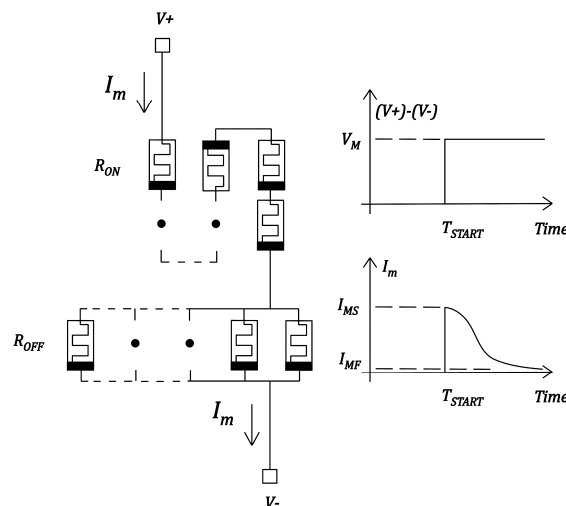


Figura 3.9 Corriente transitoria en un arreglo memristivo, con S_M memristores en serie conectados de forma serial con P_M memristores en paralelo.

Utilizando la ley de Ohm y considerando S_M memristores en serie y P_M memristores en paralelo, el valor de la corriente inicial I_{MS} y la corriente final I_{MF} se puede determinar con (3.35) y (3.36), respectivamente.

$$I_{MS} = \frac{V_M}{\left(R_{ON}S_M + \frac{R_{OFF}}{P_M}\right)} \quad (3.35)$$

$$I_{MF} = \frac{V_M}{\left(R_{OFF}S_M + \frac{R_{OFF}}{P_M}\right)} \quad (3.36)$$

En la **Figura 3.10a** se muestra el circuito para la fuente de corriente $I_{C_{i,j}}$, el cual se basa en una configuración serie - paralelo de memristores, en serie con un transistor. El voltaje de compuerta, V_{g3} , se utiliza para ajustar la amplitud de la corriente inicial o corriente pico, I_{MS} ; este valor se determina a partir de (3.35), donde $V_M = V_M(V_{g3})$ y $S_M = 5$, $P_M = 5$. Si la fuente de corriente $I_{C_{i,j}}$ se desea ajustar a los siguientes valores iniciales: $2 \mu A$, $4 \mu A$, $6 \mu A$, $8 \mu A$ y $10 \mu A$, se utiliza la curva característica $i-v$ en DC, para de esta manera obtener los correspondientes valores de V_{g3} : $3.74 V$, $3.48 V$, $3.24 V$, $3.00 V$ y $2.76 V$ (**ver la Figura 3.10b**). El resultado transitorio de la corriente $I_{C_{i,j}}$ se muestra en la **Figura 3.10c**.

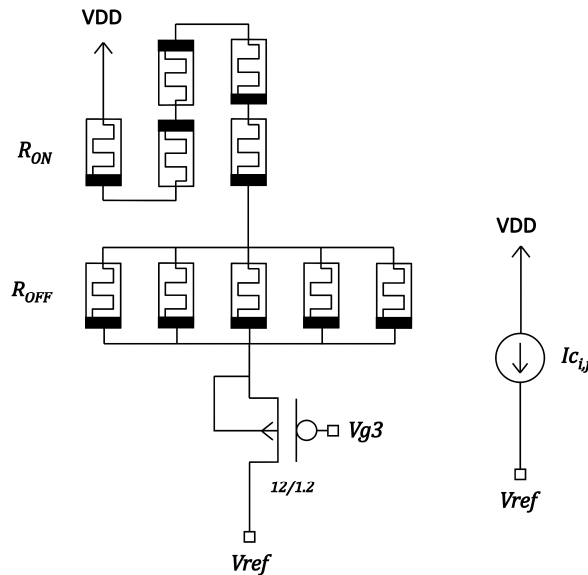


Figura 3.10a Fuente de corriente $I_{C_{i,j}}$, compuesta por 5 memristores en serie en R_{ON} , conectados en serie con 5 memristores en paralelo en R_{OFF} y un transistor MOS canal p. El voltaje analógico V_{g3} ajusta el valor inicial o pico de $I_{C_{i,j}}$.

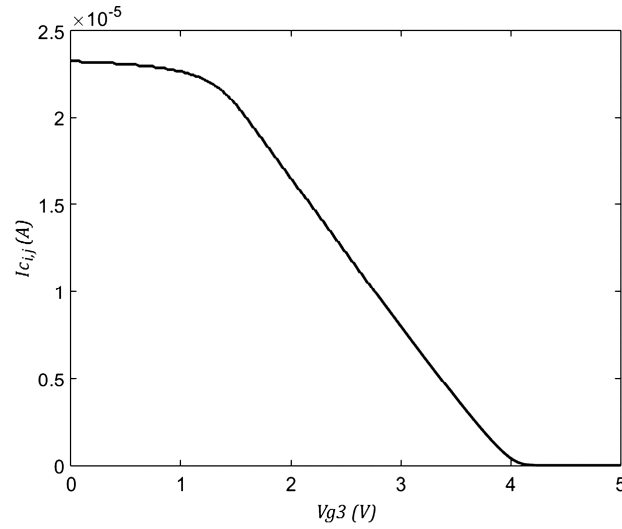


Figura 3.10b Simulación en DC de $IC_{i,j}$ versus V_{g3} . La característica lineal permite ajustar el voltaje V_{g3} en el intervalo $1.5 \text{ V} < V_{g3} < 3.9 \text{ V}$, para valores de corriente dentro del intervalo $1.0 \text{ }\mu\text{A} < IC_{i,j} < 20 \text{ }\mu\text{A}$.

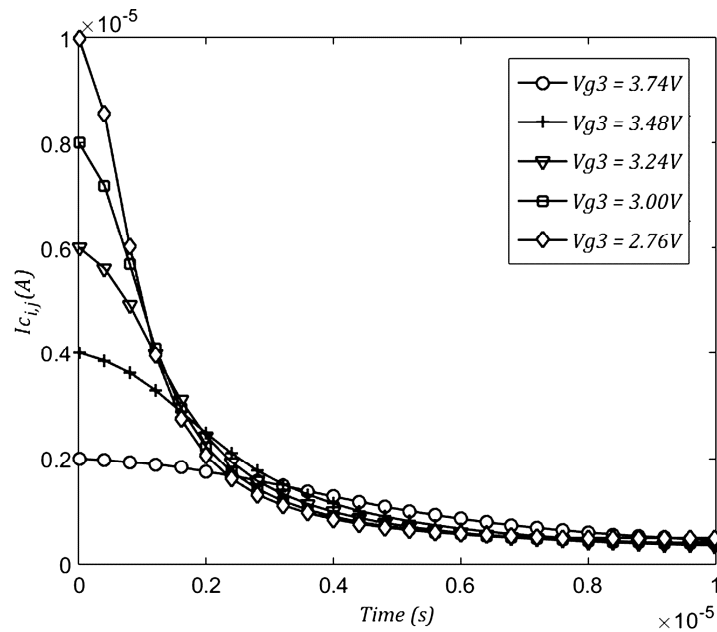


Figura 3.10c Simulación de $IC_{i,j}$ versus tiempo. La corriente pico $IC_{i,j}$ se ajustó a los valores $2 \text{ }\mu\text{A}$, $4 \text{ }\mu\text{A}$, $6 \text{ }\mu\text{A}$, $8 \text{ }\mu\text{A}$ y $10 \text{ }\mu\text{A}$ con $V_{g3} = 3.74 \text{ V}$, 3.48 V , 3.24 V , 3.00 V y 2.76 V , respectivamente.

3.7 Solución analógica del problema de asignación

En esta sección se presenta la solución al problema de asignación usando una red neuronal recurrente memristiva. Considerando la **Figura 3.2** para un sistema de 3 x 3 neuronas, la ecuación (3.32) se evaluó para $V_{ref} = V_{DD}/2 = 2.5V$, $R_0 = R_{OFF} = 500 K\Omega$, obteniendo un valor de $I_{set} = 30 \mu A$. Esta fuente de corriente se implementó con la configuración del circuito de la **Figura 3.7a**. El valor del capacitor integrador es 2.3 pF; por diseño se propuso un valor de 1 pF sumado con la capacitancia de 1.3 pF de la compuerta de la función sigmoïdal no inversora. El valor de la fuente de corriente de umbral $I\theta_{i,j}$, que emula el parámetro 2η de la ecuación (3.8), se calcula como $I\theta_{i,j} = 2\eta(10 \mu A) = 20 \mu A$, con $\eta = 1$ para este caso. $I\theta_{i,j}$ es constante y el mismo para todas las neuronas.

En la **Tabla 3.2a** se muestran los valores de los costos $C_{i,j}$ con su respectiva representación en corriente $I_{C_{i,j}} = (C_{i,j}/\max(C_{i,j}))(10 \mu A)$ de tres problemas de asignación denominados Netwk1, Netwk2 y Netwk3. La **Tabla 3.2b** muestra la solución analógica obtenida, en Volts, de la red neuronal recurrente memristiva en SPICE y la solución numérica con la red de Wang obtenida en Simulink (**ver Apéndice B**). Comparando la solución analógica con la solución numérica, se observa que ambos resultados son iguales considerando que si $V_{V_{i,j}}$ es casi 5V o 0V, este se lee como un '1' o '0' lógico, respectivamente. Las **Figuras 3.11a, b y c** muestran la respuesta eléctrica $V_{V_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para las redes Netwk1, Netwk2 y Netwk3 propuestas en la tabla misma.

Tabla 3.2a Valores de costos $C_{i,j}$ de tres problemas de signación denominados Netwk4, Netwk5 y Netwk6

Neurona (i, j)	Netwk1	Netwk2	Netwk3	Netwk1	Netwk2	Netwk3
	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$	$IC_{i,j}$	$IC_{i,j}$	$IC_{i,j}$
(1, 1)	7	1	8	7.77	1.11	8.88
(1, 2)	6	7	6	6.66	7.77	6.66
(1, 3)	8	3	1	8.88	3.33	1.11
(2, 1)	9	6	4	9.99	6.66	4.44
(2, 2)	4	5	3	4.44	5.55	3.33
(2, 3)	3	4	5	3.33	4.44	5.55
(3, 1)	5	4	5	5.55	4.44	5.55
(3, 2)	8	6	2	8.88	6.66	2.22
(3, 3)	6	9	9	6.66	9.99	9.99

Tabla 3.2b Solución analógica obtenida en SPICE y solución numérica obtenida en Simulink

Neurona (i, j)	Solución analógica $V_{v_{i,j}}$			Solución numérica		
	Netwk1	Netwk2	Netwk3	Netwk1	Netwk2	Netwk3
(1, 1)	0.58382	4.22076	0.03629	0	1	0
(1, 2)	3.97953	0.01529	0.01471	1	0	0
(1, 3)	0.31686	0.60430	4.74946	0	0	1
(2, 1)	0.02195	0.01533	4.04663	0	0	1
(2, 2)	0.71387	0.79085	0.81807	0	0	0
(2, 3)	4.17154	4.07552	0.01708	1	1	0
(3, 1)	4.30346	0.59964	0.81689	1	0	0
(3, 2)	0.19650	4.08336	4.02668	0	1	1
(3, 3)	0.32928	0.22205	0.01484	0	0	0

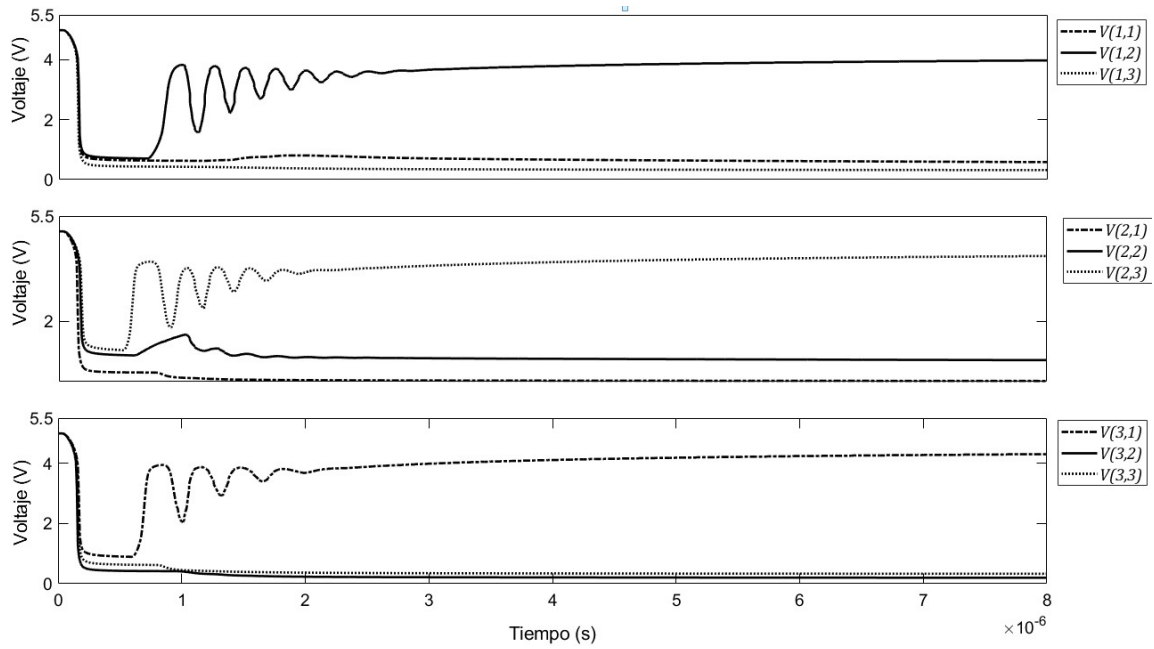


Figura 3.11a Respuesta eléctrica $V_{v_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk1.

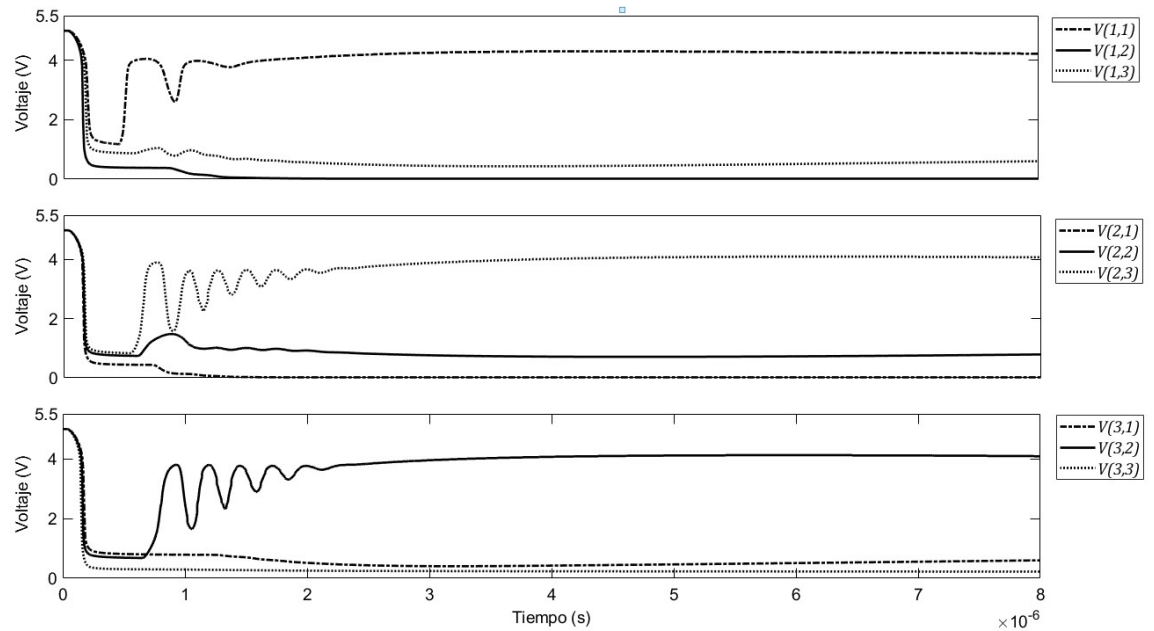


Figura 3.11b Respuesta eléctrica $V_{v_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk2.

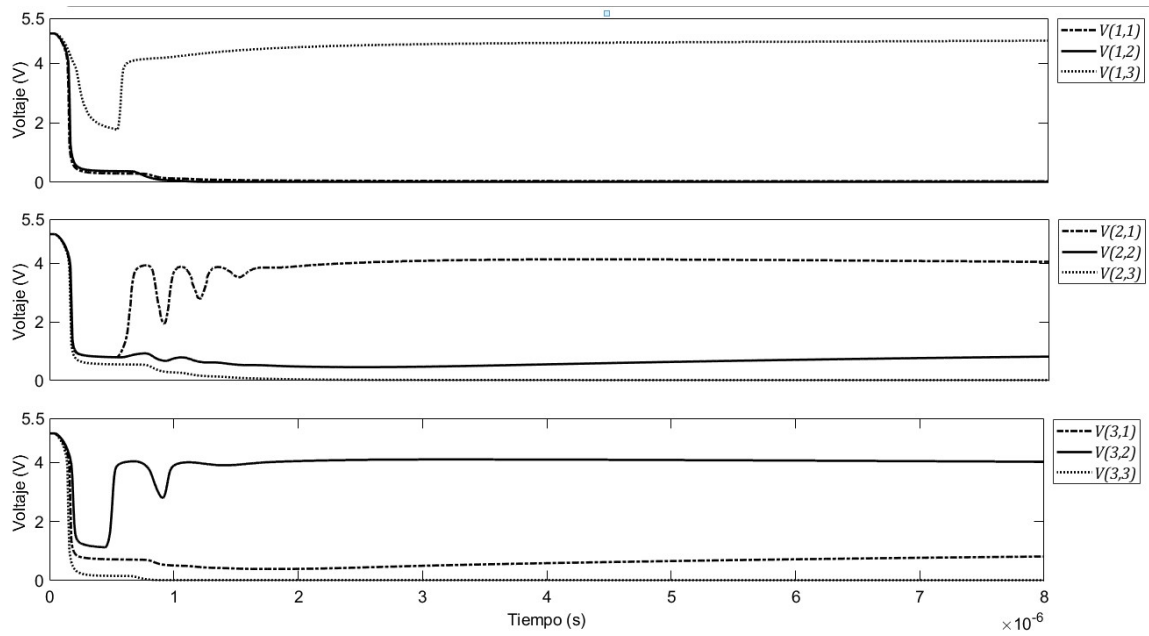


Figura 3.11c Respuesta eléctrica $V_{V_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk3.

3.8 Solución analógica del problema de transporte

En esta sección, se presenta la solución al problema de asignación usando una red neuronal recurrente memristiva. Considerando la **Figura 3.1** para un sistema de 3 x 3 neuronas, la ecuación (3.32) se evaluó para $V_{ref} = V_{DD}/2 = 2.5V$, $R_0 = R_{OFF} = 500 K\Omega$, obteniendo un valor de $I_{set} = 30 \mu A$. Esta fuente de corriente se implementó con la configuración del circuito de la **Figura 3.7a**. El valor del capacitor integrador es 2.3 pF; por diseño, se propuso un valor de 1 pF sumado con la capacitancia de 1.3 pF de la compuerta de la función sigmoïdal no inversora. En la **Tabla 3.3a** se muestran los valores de los costos $C_{i,j}$, la oferta a_i y demanda b_j de tres problemas de transporte denominados Netwk4, Netwk5 y Netwk6. La **Tabla 3.3b** presenta la representación en corriente del costo, la oferta y la demanda, los cuales se calculan como: $I\bar{c}_{i,j} = (C_{i,j}/\max(C_{i,j}))(10 \mu A)$, $I\bar{a}_i = (a_i/\max(a_i, b_j))(10 \mu A)$ y $I\bar{b}_j = (b_j/\max(a_i, b_j))(10 \mu A)$. El valor de la fuente de corriente de umbral $I\theta_{i,j}$, que emula el parámetro $\eta\theta_{ij}$ de la ecuación (3.25), se obtiene de la suma de $I\bar{a}_i$ con $I\bar{b}_j$. En la **Tabla 3.3c** se muestra la solución analógica $Vv_{i,j}$ de las simulaciones en SPICE. En la **Tabla 3.3d** $Vv_{i,j}$ se calcula $Vv_{i,j} = (Vv_{i,j}/5V)(\max(a_i, b_j))$. La oferta y demanda se obtienen como: $a_i = \sum Vv_{i,j}$ con $j = 1,2,3$ y $b_i = \sum Vv_{i,j}$ con $i = 1,2,3$. Adicionalmente, el costo total de Netwk4, Netwk5 y Netwk6 se calcula con $\sum(C_{i,j})(Vv_{i,j})$ con $i, j = 1,2,3$. En la **Tabla 3.3e** se presentan los valores de la solución numérica obtenida con Simulink, donde $V_{i,j} = (V_{i,j} \text{ en Simulink})(5V)(\max(a_i, b_j))$. Comparando los resultados se observa que el costo total obtenido por la red neuronal memristiva **Tabla 3.3d** es menor al costo obtenido por la solución numérica **Tabla 3.3e**. Las **Figuras 3.12a, b y c** muestran la respuesta eléctrica $Vv_{i,j}$, en SPICE, de cada neurona, para las redes Netwk4, Netwk5 y Netwk6.

Tabla 3.3a Valores de costos $C_{i,j}$, oferta a_i y demanda b_j de tres problemas de transporte denominados Netwk4, Netwk5 y Netwk6.

Neurona (i, j)	Netwk4			Netwk5			Netwk6		
	$C_{i,j}$	a_i	b_j	$C_{i,j}$	a_i	b_j	$C_{i,j}$	a_i	b_j
(1, 1)	7	40	25	7	70	50	8	40	30
(1, 2)	6	40	35	4	70	80	6	40	80
(1, 3)	8	40	60	15	70	45	1	40	10
(2, 1)	9	50	25	1	90	50	4	60	30
(2, 2)	4	50	35	19	90	80	3	60	80
(2, 3)	3	50	60	2	90	45	5	60	10
(3, 1)	5	30	25	3	15	50	5	20	30
(3, 2)	8	30	35	12	15	80	2	20	80
(3, 3)	6	30	60	10	15	45	9	20	10

Tabla 3.3b Representación en corriente del costo, la oferta y la demanda

Neurona (i, j)	Netwk4				Netwk5				Netwk6			
	$I\bar{C}_{i,j}$	$I\bar{a}_i$	$I\bar{b}_j$	$I\theta_{i,j}$	$I\bar{C}_{i,j}$	$I\bar{a}_i$	$I\bar{b}_j$	$I\theta_{i,j}$	$I\bar{C}_{i,j}$	$I\bar{a}_i$	$I\bar{b}_j$	$I\theta_{i,j}$
(1, 1)	7.77	6.66	4.16	10.82	3.684	7.77	5.55	13.32	8.88	5.00	3.75	8.750
(1, 2)	6.66	6.66	5.83	12.49	2.105	7.77	8.88	16.65	6.66	5.00	10.0	15.00
(1, 3)	8.88	6.66	10.0	16.66	7.894	7.77	5.00	12.77	1.11	5.00	1.25	6.250
(2, 1)	9.99	8.33	4.16	12.49	0.526	10.0	5.55	15.55	4.44	7.50	3.75	11.25
(2, 2)	4.44	8.33	5.83	14.16	10.00	10.0	8.88	18.88	3.33	7.50	10.0	17.50
(2, 3)	3.33	8.33	10.0	18.33	1.052	10.0	5.00	15.00	5.55	7.50	1.25	8.750
(3, 1)	5.55	5.00	4.16	9.16	1.578	1.66	5.55	7.210	5.55	2.50	3.75	6.250
(3, 2)	8.88	5.00	5.83	10.83	6.315	1.66	8.88	10.54	2.22	2.50	10.0	12.50
(3, 3)	6.66	5.00	10.0	15.00	5.263	1.66	5.00	6.660	9.99	2.50	1.25	3.750

Tabla 3.3c Solución analógica $Vv_{i,j}$ obtenida en SPICE

Neurona (i, j)	Netwk4	Netwk5	Netwk6
	$Vv_{i,j}$	$Vv_{i,j}$	$Vv_{i,j}$
(1, 1)	0.40308	0.01491	0.45477
(1, 2)	2.65081	3.85986	1.29839
(1, 3)	0.17217	0.01457	0.64134
(2, 1)	0.01454	2.49306	1.18499
(2, 2)	0.01454	0.01466	2.51945
(2, 3)	4.08842	2.52635	0.01481
(3, 1)	1.58085	0.30191	0.12369
(3, 2)	0.15685	0.47256	1.10332
(3, 3)	0.66717	0.01459	0.01473

Tabla 3.3d Solución analógica $Vv_{i,j}$ desnormalizada

Neurona (i, j)	Netwk4			Netwk5			Netwk6			
	$Vv_{i,j}$	a_i	b_j	$Vv_{i,j}$	a_i	b_j	$Vv_{i,j}$	a_i	b_j	
(1, 1)	4.8370	38.712	23.981	0.2682	70.002	50.746	7.2763	38.312	28.215	
(1, 2)	31.809	38.712	33.866	69.471	70.002	78.252	20.774	38.312	78.738	
(1, 3)	2.0660	38.712	59.133	0.2626	70.002	46.748	10.261	38.312	10.734	
(2, 1)	0.1745	49.410	23.981	45.056	91.543	50.746	18.959	59.508	28.215	
(2, 2)	0.1745	49.410	33.866	0.2635	91.543	78.252	40.311	59.508	78.738	
(2, 3)	49.061	49.410	59.133	46.223	91.543	46.748	0.2370	59.508	10.734	
(3, 1)	18.970	28.858	23.981	5.4221	14.202	50.746	1.9790	19.867	28.215	
(3, 2)	1.8822	28.858	33.866	8.5172	14.202	78.252	17.653	19.867	78.738	
(3, 3)	8.0060	28.858	59.133	0.2628	14.202	46.748	0.2357	19.867	10.734	
Costo total = 548.641			Costo total = 547.314			Costo total = 438.397				

Tabla 3.3e Solución numérica obtenida con Simulink

Neurona (i, j)	Netwk4			Netwk5			Netwk6			
	$v_{i,j}$	a_i	b_j	$v_{i,j}$	a_i	b_j	$v_{i,j}$	a_i	b_j	
(1, 1)	3.7917	39.946	24.938	0.0000	69.930	49.945	0.1541	39.767	29.773	
(1, 2)	34.899	39.946	34.971	69.930	69.930	79.919	29.763	39.767	79.803	
(1, 3)	1.2559	39.346	59.986	0.0000	69.930	44.995	9.8503	39.767	10.158	
(2, 1)	0.0000	49.976	24.938	44.993	89.991	49.945	29.465	59.947	29.773	
(2, 2)	0.0718	49.976	34.971	0.0000	89.991	79.919	30.327	59.947	79.803	
(2, 3)	49.904	49.976	59.986	44.995	89.991	44.995	0.1541	59.947	10.158	
(3, 1)	21.146	29.973	24.938	4.9497	14.939	49.945	0.1541	20.020	29.773	
(3, 2)	0.0000	29.973	34.971	9.9899	14.939	79.919	19.712	20.020	79.803	
(3, 3)	8.8268	29.973	59.986	0.0000	14.939	44.995	0.1541	20.020	10.158	
Costo total = 554.671			Costo total = 549.428			Costo total = 440.862				

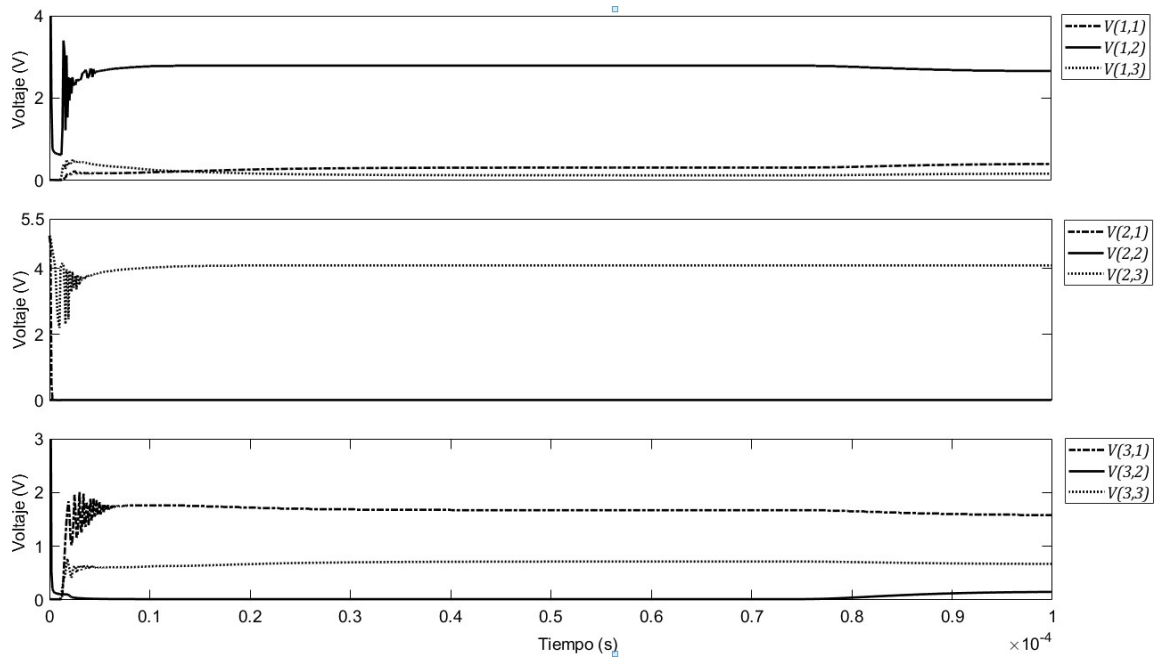


Figura 3.12a Respuesta eléctrica $V_{v_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk4.

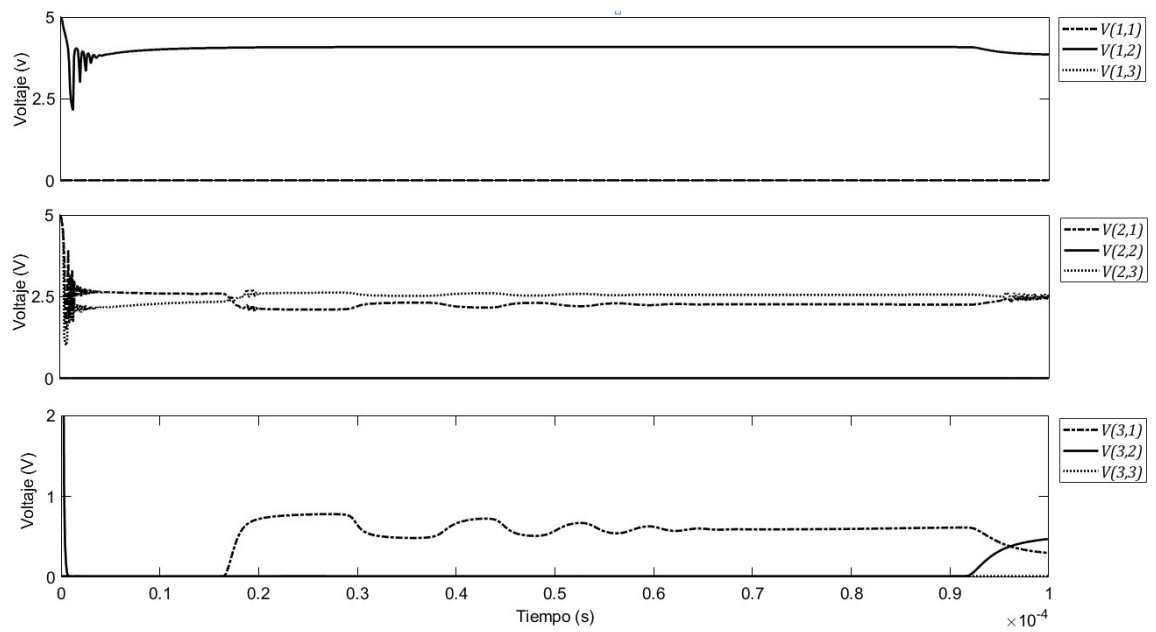


Figura 3.12b Respuesta eléctrica $V_{v_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk5.

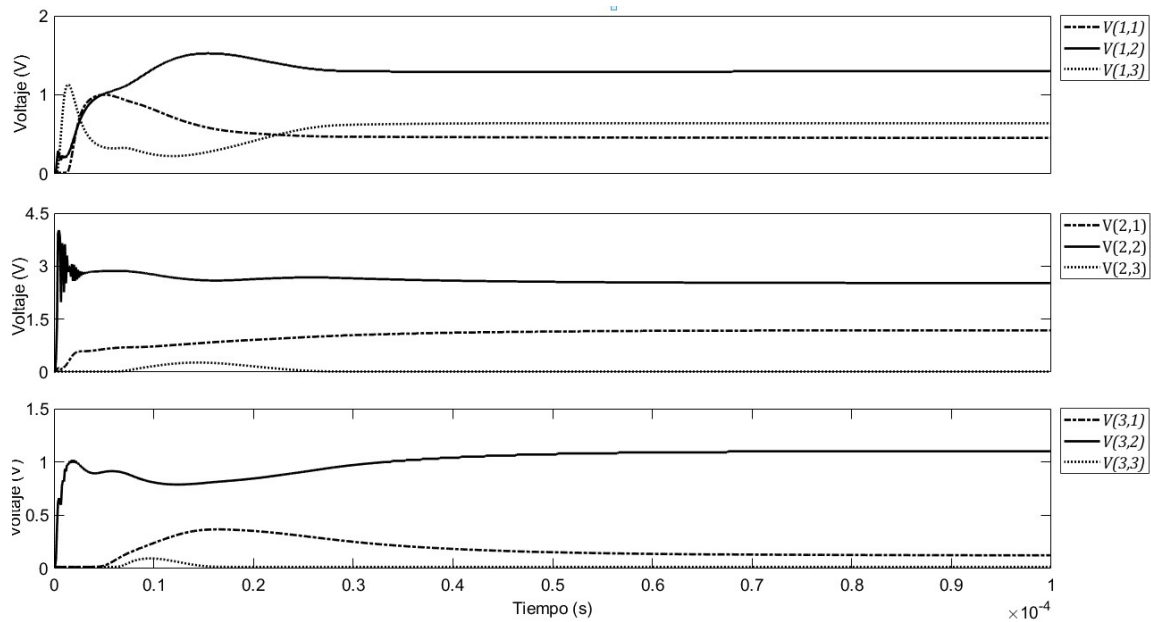


Figura 3.12c Respuesta eléctrica $V_{V_{i,j}}$, en SPICE, de cada neurona $N_{i,j}$ para la red Netwk6.

3.9 Conclusiones

La red neuronal que se estudió en este capítulo fue originalmente modelada por Wang [7] usando componentes resistivos de un solo valor y corrientes producidas por resistores lineales y capacitores para su implementación. En este trabajo se propuso un circuito con configuración memristor/CMOS que se comporta de acuerdo con los parámetros analógicos de la red neuronal, resolviendo en modo de tiempo continuo, problemas de optimización.

Es de destacar que el sistema eléctrico propuesto es innovador, ya que nuevas configuraciones de circuitos fueron introducidas, contribuyendo de esta manera en el área del diseño VLSI al proponer el uso del memristor en sus estados estático y dinámico con transistores CMOS.

Este sistema presenta tres características: 1) No requiere ningún sistema electrónico de programación, sino un flujo de corriente con la dirección indicada en la **Figura 3.3**, donde se elige la polaridad adecuada en el memristor usado. 2) El uso de los memristores con valor R_{OFF} simplifica la implementación de las

fuentes de corriente nombradas I_{set} y $I\theta_{i,j}$ reduciendo el número de transistores utilizados y evitando el uso de voltajes de polarización externos. 3) La configuración serie (R_{ON}) – paralelo (R_{OFF}) presentada en la **Figura 3.10** es una idea original para remplazar un gran elemento capacitivo y resistivo para representar el término transitorio $\lambda_{c_{ij}}e^{-\frac{t}{\tau}}$, el cual es esencial para ajustar los parámetros del problema.

Finalmente, como una probable aplicación a resolver con esta red neuronal memristiva sería la planeación en robótica, por lo que un sistema memristivo a gran escala es posible con el rediseño de los circuitos CMOS presentados en este trabajo.

Referencias

- [1] Y.V. Pershin, M. Di Ventra, Memcomputing and swarm intelligence, <https://arxiv.org/abs/1408.6741>.
- [2] A. Gelencsér, T. Podromakis, C. Toumazou, T. Roska, Biomimetic model of the outer plexiform layer by incorporating memristive devices, *Phys. Rev. E.* 85 (2012) 041918-1-041918-10.
- [3] S.P. Mohanty, Memristor: from basics to deployment, *IEEE Potentials.* 32 (2013) 34-39.
- [4] L. Gao, F. Merrih-Bayat, F. Alibart, X. Guo, B.D. Hoskins, K.T. Cheng, D.B. Strukov, Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing, 2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)
- [5] S.G. Hu, Y. Liu, Z. Liu, T.P. Chen, J.J. Wang, Q. Yu, L.J. Deng, Y. Yin, S. Hosaka, Associative memory realized by a reconfigurable memristive Hopfield neural network, *Nat Commun.* 6 (2015) 1-8.

- [6] U.P. Wen, K.M. Lan, H.S. Shih, A review of Hopfield neural networks for solving mathematical programming problems, *Eur. J. Oper. Res.* 198 (2009) 675-687.
- [7] J. Wang, Analysis and design of an analog sorting network, *IEEE Trans. Neural Netw.* 6 (4) (1995) 962-971.
- [8] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, TEAM: Threshold adaptive memristor model, *IEEE Trans. Circuit Syst.* 60 (1) (2013) 211-221.
- [9] D.G. Luenberger, *Linear and nonlinear programming*, 2nd Edition. Addison-Wesley, MA (1984).
- [10] H.A. Taha, *Operations research: an introduction*. Pearson Prentice Hall, USA (2007).
- [11] J. Wang, Analogue neural network for solving the assignment problem, *Electron. Lett.* 28 (11) (1992) 1047-1050.
- [12] S.P. Eberhardt, T. Duad, D.A. Kerns, T.X. Brown, A.P. Thakoor, Competitive neural architecture for hardware solution to the assignment problem, *Neural Netw.* 4 (4) (1991) 431-442.
- [13] W.J. Wolfe, J.M. MacMillan, G. Brady, R. Mathews, J.A. Rothman, D. Mathis, M.D. Orosz, C. Anderson, G. Alaghband, Inhibitory grids and the assignment problem, *IEEE Trans. Neural Netw.* 4 (2) (1993) 319-331.
- [14] K. Urahama, Analog circuit for solving assignment problem, *IEEE Trans. Circuits Syst. I* 40 (5) (1994) 426-429.
- [15] J. Wang, An analog neural network for linear programming: analysis, design, and simulation, *Proc. IEEE Int. Symp. On Circuits and Systems*, San Diego, CA, 1992.
- [16] P.H. Siqueira, Application of Wang's recurrent neural network to solve the transportation problem, *Int. J. Comput. Sci. Netw. Secur.* 12 (7) (2012) 50-54.

- [17] J. Wang, Analysis and design of a recurrent neural network for linear programming, IEEE Trans. Circuit Syst. 40 (9) (1993) 613-618.
- [18] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. Weiser, E.G. Friedman, Models of memristors for SPICE simulations, 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel.
- [19] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, MRL memristor ratioed logic, 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA).
- [20] G. Palmisano, G. Palumbo, S. Pennisi, CMOS Current amplifiers, Springer, USA, 1999, pp.45-48.

Capítulo 4

Optimizador memristivo para la asignación de tareas

4.1 Introducción

Actualmente, el diseño de circuitos integrados CMOS que realicen funciones analógicas de forma eficiente es un tema de investigación. Esto ocurre principalmente en los nuevos circuitos de procesamiento paralelo para los sistemas neuromórficos, donde comúnmente los componentes de vectores o matrices son valores analógicos. Desde el punto de vista de la ingeniería, hay un optimizador integrado en los sistemas neuromórficos. Algunos ejemplos de estos optimizadores son el algoritmo de retro – propagación del error (Error back-propagation) y el cómputo del ganador toma todo (Winner-take-all). Un casi óptimo desempeño puede ser reproducido a nivel de circuito con dichos optimizadores para patrones de series de tiempo. Sin embargo, con la aparición de las redes neuronales pulsadas, las cuales son la tercera generación de modelos de redes neuronales artificiales, se requieren nuevos métodos de diseño de hardware. Una opción es la posibilidad de incluir resistores con memoria o memristores de escala nanométrica en circuitos CMOS, para la reproducción a nivel de hardware de algoritmos de aprendizaje para las redes neuronales pulsadas. Estos algoritmos de aprendizaje se convertirán en nuevos optimizadores adecuados para la emulación del tiempo real de las funciones del cerebro. En este contexto, se presenta el desarrollo a nivel de hardware de un optimizador CMOS/memristor capaz de calcular en forma paralela la solución del

problema de Asignación de Tareas. Las simulaciones eléctricas se realizaron usando la Suite de desarrollo IC-Nanometer de Mentor Graphics.

4.2 Asignación de Tareas

La Asignación de Tareas se refiere a un paradigma de la programación lineal, donde se encuentra la mejor configuración de un conjunto de **procesadores** con un conjunto de **figuras de mérito**, maximizando el desempeño del sistema. La asignación de tareas resuelve la asociación óptima entre procesadores con figuras de mérito (FoMs). Un procesador tiene la capacidad de realizar un conjunto de diferentes tareas, denominado $T = \{T_1, T_2, \dots, T_N\}$ donde N es el número de tareas a realizar. Igualmente, estos procesadores pueden realizar las tareas en T con la correspondiente FoM = $[F_1, F_2, \dots, F_N]$, donde cada F_i es un número real en el intervalo $[0, 1]$. El sistema en consideración tiene N procesadores, donde cada procesador tiene su propio FoM $_i$ donde: $i = 1, 2, \dots, N$. La asignación de tareas es definida por la matriz de permutación **P** que maximiza el desempeño del sistema. Esta matriz indica que tareas realizará cada procesador. La matriz **F** especifica las figuras de mérito de los procesadores, cuya entradas son: $F_{i,j}$ donde el primer índice $i = 1, 2, \dots, N$ es la tarea y el segundo índice $j = 1, 2, \dots, N$ el procesador. El problema de asignación puede ser resuelto por los métodos clásicos: simplex y húngaro [1].

4.3 Compuerta lógica

Una de las aplicaciones más llamativas de los dispositivos memristivos es su uso en circuitos lógicos como compuertas lógicas [2-4], abriendo una oportunidad para la creación de nuevas arquitecturas computacionales diferentes a la de Von Neumann, donde los cálculos y el almacenamiento de datos se realizan por separado. En este trabajo se empleó la familia lógica denominada MRL (Memristor Ratioed Logic) [5] donde la resistencia programable de los dispositivos memristivos es utilizada para el cálculo de las funciones Booleanas AND/OR. Una compuerta OR consiste en dos memristores conectados en serie con la polaridad opuesta, donde la terminal de salida V_{out} es el nodo común entre

ambos memristores y las terminales V_{in1} y V_{in2} son las entradas de señal de la compuerta lógica (**ver la Figura 4.1a**). Cuando ambas entradas son idénticas (ambas son '1' lógico o '0' lógico) la caída de voltaje entre las entradas es cero debido a que no existe un flujo de corriente dentro del circuito por lo que el voltaje de salida es igual al voltaje de entrada. Sin embargo, con entradas diferentes fluye una corriente de la terminal de mayor voltaje (donde el '1' es aplicado) a la terminal de menor voltaje (donde el '0' es aplicado) cambiando el estado inicial de los memristores (**ver la Figura 4.1b**). Si se asume inicialmente que $R_1 = R_{OFF}$ y $R_2 = R_{ON}$ al final del proceso computacional $R_1 = R_{ON}$ y $R_2 = R_{OFF}$ (**ver la Figura 4.1c**). Además, si $R_{OFF} \gg R_{ON}$, el voltaje de salida V_{out} se determina a partir de un divisor de voltaje entre ambos memristores.

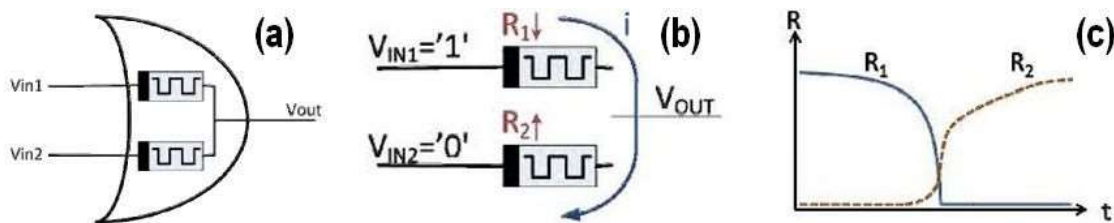


Figura 4.1a) Esquemático de la compuerta lógica OR. **b)** Comportamiento de la compuerta OR. **c)** Comportamiento de la resistencia de los memristores.

El estado inicial de los memristores no tiene efecto sobre el resultado final, pero si genera un retardo en el tiempo de conmutación. Cuando el estado inicial es alto (aproximadamente R_{OFF}) la corriente que fluye a través de la compuerta es relativamente pequeña produciendo que el tiempo de estabilización sea mayor. También es posible que los memristores no conmuten completamente y alcancen su máximo y mínimo estado [R_{OFF} y R_{ON}], si el voltaje no es aplicado por suficiente tiempo o el nivel del voltaje es demasiado bajo haciendo difícil distinguir entre los diferentes niveles de salida. El número de entradas de la compuerta OR se puede extender a n entradas (**ver la Figura 4.2a**), además se puede conectar a un inversor CMOS para tener una compuerta de tipo NOR (**ver la Figura 4.2b**).

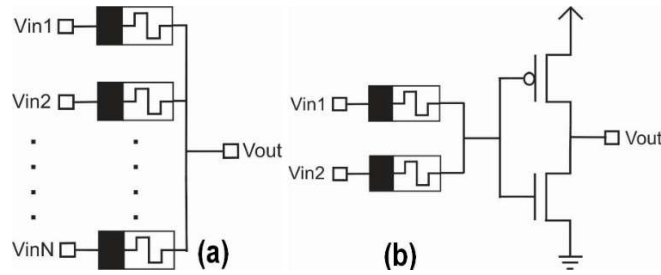


Figura 4.2a) Esquemático de una compuerta OR con N entradas. **b)** Esquemático de una compuerta lógica NOR de dos entradas.

4.4 Neurona memristiva

La parte fundamental del optimizador memristivo consiste en una neurona eléctrica, la cual se basa en la estructura y principio de funcionamiento de la compuerta lógica NOR de la familia MRL [5] (descrita anteriormente). La neurona $N(i, j)$ con $i = 1, 2, \dots, N$ y $j = 1, 2, \dots, N$ tiene un total de $2(N - 1)$ entradas, conectadas con un capacitor (**ver la Figura 4.3a**). Esta neurona se puede replicar N^2 veces para formar un arreglo neuronal $N \times N$ de compuertas NOR. Este arreglo se denomina optimizador memristivo y corresponde a las dimensiones de una matriz analógica denominada **F**. La interconectividad de las neuronas se basa únicamente en conexiones laterales donde la salida $V_0(i, j)$ de la neurona $N(i, j)$ se conecta con todas las neuronas vecinas de su misma fila y columna (**ver la Figura 4.3b**).

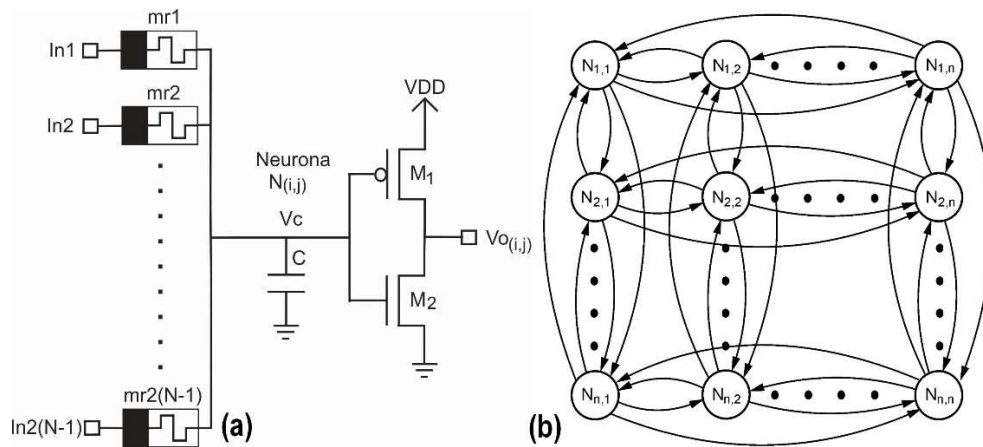


Figura 4.3a) Neurona eléctrica memristiva. **b)** Interconectividad del arreglo de neuronas

El número total de memristores empleados en un arreglo $N \times N$ se expresa con la ecuación (4.1). Para el arreglo de dimensiones 4×4 propuesto, $N = 4$, se tienen 16 neuronas donde cada una tiene 6 entradas (**ver la Figura 4.4**) y el número total de memristores empleados es 96. Este arreglo eléctrico se comporta como un sistema neuronal competitivo de tipo ganador toma todo (**Winner-Take-All**).

$$N_{mr} = 2(N)^2(N - 1) \quad (4.1)$$

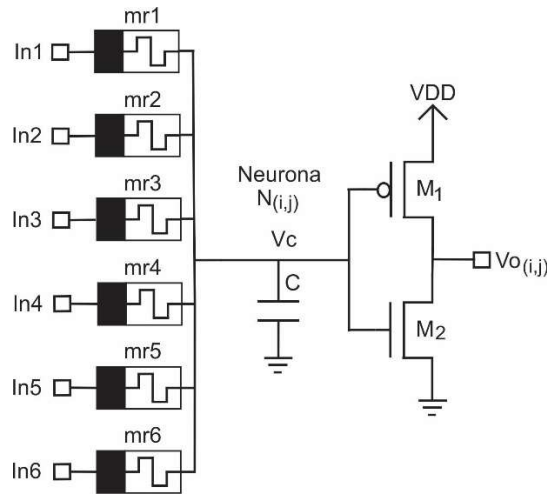


Figura 4.4 Neurona eléctrica empleada para el arreglo de dimensiones 4×4 .

4.5 Proceso de programación de los memristores

Con el objetivo de transferir los valores de la matriz analógica F a la variable de estado de los memristores, es necesario realizar un proceso de acondicionamiento de datos. Este consiste en cambiar el rango de operación de la matriz F , donde cada $F_{i,j}$ es un número real en el intervalo $[0 - 1]$, esto con el objetivo de evitar que el estado inicial del memristor sean los extremos del dispositivo $[R_{OFF}$ y $R_{ON}]$, ya que se puede producir un retardo mayor en el tiempo de estabilización del sistema. Este proceso de acondicionamiento se realiza de la siguiente manera: teniendo la matriz F y tomando la entrada $F_{i,j}$ se aplica la expresión (4.2) para obtener el valor $\theta_{i,j}$ con $i = 1, \dots, 4$ y $j = 1, \dots, 4$, donde β es un factor de compactación encontrado de forma empírica dentro del intervalo $0.20 \leq$

$\beta \leq 0.80$. El conjunto de valores $\{\theta_{i,j}\}$ contiene las entradas de la matriz θ la cual corresponde a la matriz F .

$$\theta_{i,j} = (0.5 + \beta(F_{i,j} - 0.5)) \quad (4.2)$$

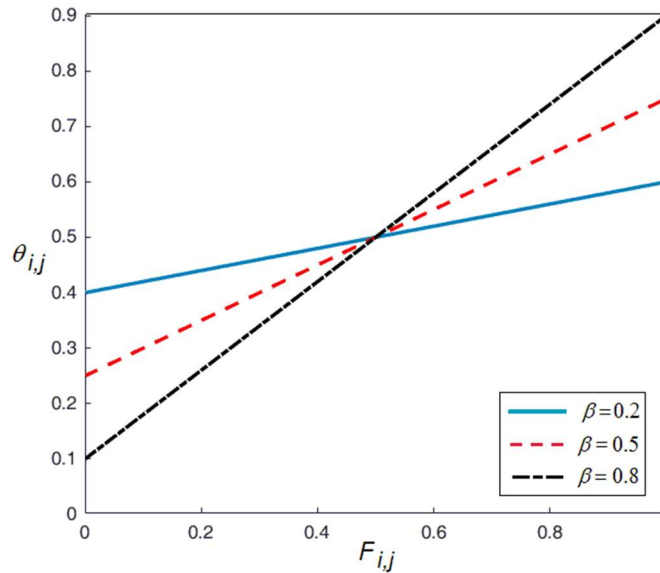


Figura 4.5 Muestra los valores de $\theta_{i,j}$ obtenidos con (4.2) para distintos valores de $F_{i,j}$ con valores de β .

En este trabajo se resolvieron cuatro problemas de asignación cuyas matrices de entrada son: F_1 , F_2 , F_3 y F_4 . Utilizando $\beta = 0.5$ se obtienen las matrices: θ_1 , θ_2 , θ_3 y θ_4 donde los valores de $\theta_{i,j}$ están en el intervalo $[0.25 - 0.75]$.

$$F_1 = \begin{bmatrix} 0.25 & 0.56 & 0.82 & 0.29 \\ 0.50 & 0.62 & 0.90 & 0.48 \\ 0.70 & 0.48 & 0.20 & 0.50 \\ 0.63 & 0.24 & 0.06 & 0.37 \end{bmatrix}$$

$$\theta_1 = \begin{bmatrix} 0.375 & 0.530 & 0.660 & 0.395 \\ 0.500 & 0.560 & 0.700 & 0.490 \\ 0.600 & 0.490 & 0.350 & 0.500 \\ 0.565 & 0.370 & 0.280 & 0.435 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 0.50 & 0.41 & 0.70 & 0.39 \\ 0.40 & 0.08 & 0.55 & 0.35 \\ 0.35 & 0.50 & 0.40 & 0.80 \\ 0.20 & 0.27 & 0.76 & 0.61 \end{bmatrix}$$

$$\theta_2 = \begin{bmatrix} 0.500 & 0.455 & 0.600 & 0.445 \\ 0.450 & 0.290 & 0.525 & 0.425 \\ 0.425 & 0.500 & 0.450 & 0.650 \\ 0.350 & 0.385 & 0.630 & 0.555 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 0.55 & 0.37 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.26 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.07 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix}$$

$$\theta_3 = \begin{bmatrix} 0.525 & 0.435 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.380 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.285 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

$$\mathbf{F}_4 = \begin{bmatrix} 0.55 & 0.80 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.85 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.75 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad \boldsymbol{\theta}_4 = \begin{bmatrix} 0.525 & 0.650 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.675 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.625 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

Con la ecuación (4.3) y los valores generados de $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, $\boldsymbol{\theta}_3$ y $\boldsymbol{\theta}_4$ se obtienen las matrices \mathbf{x}_{init_1} , \mathbf{x}_{init_2} , \mathbf{x}_{init_3} y \mathbf{x}_{init_4} las cuales representan el valor inicial de la variable de estado de cada memristor. De esta manera la variable de estado de todos los memristores (mr1-mr6) de la neurona N(i,j) (ver la Figura 4.4) presentan el mismo valor inicial $x_{init}(i,j)$.

$$\mathbf{x}_{init} = D * \boldsymbol{\theta} \quad (4.3)$$

Donde D = 3nm es el tamaño del memristor.

$$\mathbf{x}_{init_1} = \begin{bmatrix} 1.125\text{nm} & 1.590\text{nm} & 1.980\text{nm} & 1.185\text{nm} \\ 1.500\text{nm} & 1.680\text{nm} & 2.100\text{nm} & 1.470\text{nm} \\ 1.800\text{nm} & 1.470\text{nm} & 1.050\text{nm} & 1.500\text{nm} \\ 1.695\text{nm} & 1.110\text{nm} & 0.840\text{nm} & 1.305\text{nm} \end{bmatrix}$$

$$\mathbf{x}_{init_2} = \begin{bmatrix} 1.500\text{nm} & 1.365\text{nm} & 1.800\text{nm} & 1.335\text{nm} \\ 1.350\text{nm} & 0.870\text{nm} & 1.575\text{nm} & 1.275\text{nm} \\ 1.275\text{nm} & 1.500\text{nm} & 1.350\text{nm} & 1.950\text{nm} \\ 1.050\text{nm} & 1.155\text{nm} & 1.890\text{nm} & 1.665\text{nm} \end{bmatrix}$$

$$\mathbf{x}_{init_3} = \begin{bmatrix} 1.575\text{nm} & 1.305\text{nm} & 0.975\text{nm} & 1.950\text{nm} \\ 2.025\text{nm} & 1.425\text{nm} & 1.140\text{nm} & 1.455\text{nm} \\ 1.755\text{nm} & 1.875\text{nm} & 1.545\text{nm} & 0.855\text{nm} \\ 1.965\text{nm} & 1.530\text{nm} & 1.935\text{nm} & 1.905\text{nm} \end{bmatrix}$$

$$\mathbf{x}_{init_4} = \begin{bmatrix} 1.575\text{nm} & 1.950\text{nm} & 0.975\text{nm} & 1.950\text{nm} \\ 2.025\text{nm} & 1.425\text{nm} & 2.025\text{nm} & 1.455\text{nm} \\ 1.755\text{nm} & 1.875\text{nm} & 1.545\text{nm} & 1.875\text{nm} \\ 1.965\text{nm} & 1.530\text{nm} & 1.935\text{nm} & 1.905\text{nm} \end{bmatrix}$$

A partir de la expresión de la memristancia del modelo TEAM [6] (4.4) y la variable de estado inicial x_{init} se determina la resistencia inicial de cada memristor.

$$R_{init}(i,j) = R_{ON} \exp\left(\frac{\lambda}{x_{off} - x_{on}} (x_{init}(i,j) - x_{on})\right), \quad \lambda = \ln(R_{off}/R_{on}) \quad (4.4)$$

4.6 Simulaciones eléctricas

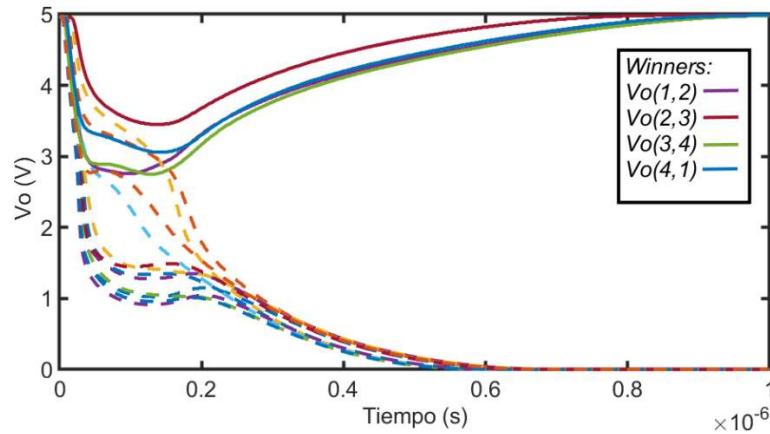
Como se mencionó anteriormente para mostrar el desempeño del optimizador memristivo propuesto, se resolvieron cuatro problemas de asignación F_1 , F_2 , F_3 y F_4 donde el principal objetivo es encontrar una manera de asignar cuatro tareas a cuatro procesadores con la finalidad de maximizar el desempeño de un sistema. Los memristores que conforman el arreglo 4×4 fueron simulados utilizando el modelo de umbral adaptable TEAM presentado en [6]. Este modelo se describió con el lenguaje de descripción de hardware Verilog-A usando las líneas de código reportadas en [7] (**ver Apéndice D**). Los parámetros de ajuste utilizados en el modelo TEAM están orientados al funcionamiento de sistemas digitales de conmutación rápida [5] estos se presentan en la **Tabla 4.1**.

Tabla 4.1 Parámetros eléctricos del modelo TEAM usados en simulación.

Parámetro	Valor	Comentario
dt	1×10^{-9} s	Diferencial de tiempo
R_{ON}	$100 \times 10^3 \Omega$	Memristancia en $x = 0$ m
R_{OFF}	$2 \times 10^6 \Omega$	Memristancia en $x = 3 \times 10^{-9}$ m
i_{off}	10×10^{-12} A	Corriente de umbral
i_{on}	-10×10^{-12} A	Corriente de umbral
w_c	107×10^{-12} m	Parámetro de ajuste
a_{on}	0 m	Parámetro de ajuste
a_{off}	3×10^{-9} m	Parámetro de ajuste
k_{on}	-5×10^{-8} m/s	Parámetro de ajuste
k_{off}	5×10^{-8} m/s	Parámetro de ajuste
α_{on}	1	Parámetro de ajuste
α_{off}	1	Parámetro de ajuste
x_{on}	0 m	Límite inferior de x
x_{off}	3×10^{-9} m	Límite superior de x

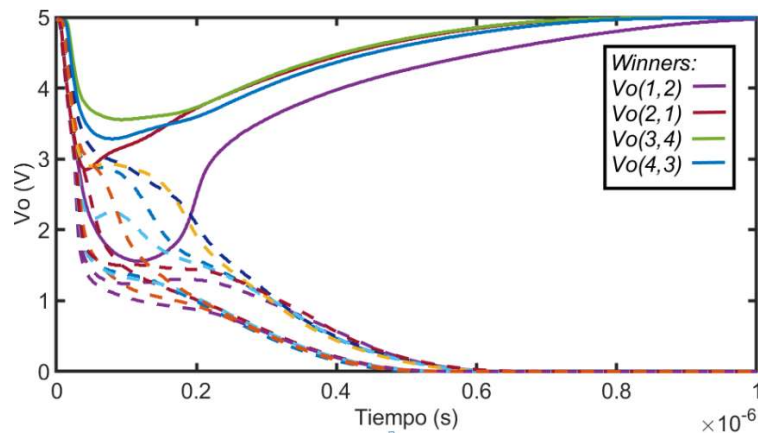
Para un tiempo óptimo de ejecución, las simulaciones eléctricas del arreglo neuronal 4×4 , se realizaron usando la Suite de desarrollo IC-Nanometer de

Mentor Graphics. El inversor CMOS de la neurona $N(i, j)$ (**ver la Figura 4.4**) fue simulado usando tecnología estándar de pozo n de 0.5 micras. La razón W/L de los transistores M1 y M2 es $360/0.6$ y $120/0.6$, donde W y L se encuentran en micras. El capacitor C tiene un valor de 1 pF y $VDD = 5V$ como fuente de alimentación. Durante el proceso de simulación existe una competencia entre las dieciséis neuronas que conforman el arreglo. Al concluir este proceso de simulación se tiene como resultado cuatro neuronas ganadoras (**Winners**) y doce perdedoras (**Losers**). Con únicamente una neurona ganadora por fila y columna. En las **Figuras 4.6, 4.7, 4.8 y 4.9** se muestra gráficamente las soluciones a los cuatro problemas a resolver F_1, F_2, F_3 y F_4 donde se observan todas las salidas V_o de las dieciséis neuronas. Al comienzo de la simulación en $t = 0s$ el voltaje de todos los capacitores V_c se inicializa a un valor de $0V$, es decir los capacitores se encuentran descargados, por lo que la salida $V_o(i, j)$ de todas las neuronas $N(i, j)$ tienen el valor inicial de $5V$. Durante el proceso de competencia, las salidas $V_o(i, j)$ que alcanzan un valor aproximado a $5V$ son neuronas ganadoras y las salidas con un valor aproximado a $0V$ son neuronas perdedoras. El tiempo de simulación para cada caso es de $1\mu s$. Los trazos con los valores de $5V$ y $0V$ obtenidos de la simulación se pueden interpretar como un '1' y '0' lógico respectivamente. Con estos valores se obtienen las matrices de permutación P_1, P_2, P_3 y P_4 la cuales nos indican que elementos de las matrices F_1, F_2, F_3 y F_4 se deben escoger para que sumados den como resultado el máximo valor del sistema. Para verificar el funcionamiento del optimizador memristivo se realizó una comparación entre nuestros resultados y los obtenidos por el algoritmo húngaro descrito en el **Apéndice E**.



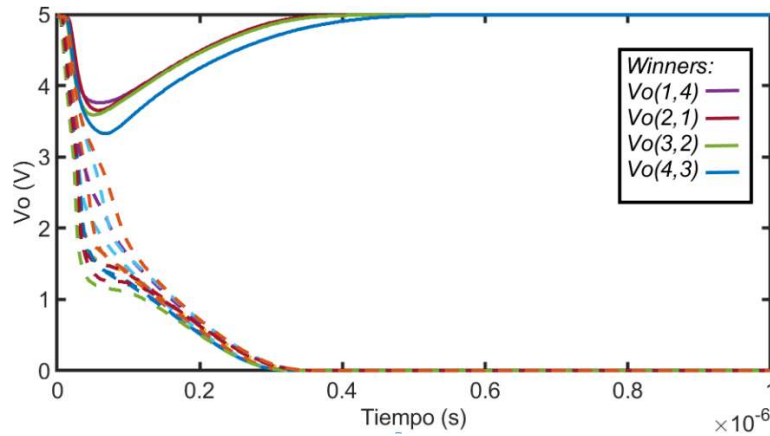
$$\mathbf{F}_1 = \begin{bmatrix} 0.25 & \mathbf{0.56} & 0.82 & 0.29 \\ 0.50 & 0.62 & \mathbf{0.90} & 0.48 \\ 0.70 & 0.48 & 0.20 & \mathbf{0.50} \\ \mathbf{0.63} & 0.24 & 0.06 & 0.37 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 \end{bmatrix}$$

Figura 4.6 Respuesta dinámica de las 16 neuronas para la solución del problema \mathbf{F}_1 . Los valores en negrita de la matriz \mathbf{F}_1 representan la solución obtenida con el algoritmo húngaro mientras que los valores en negrita de la matriz \mathbf{P}_1 indican la solución con el optimizador memristivo.



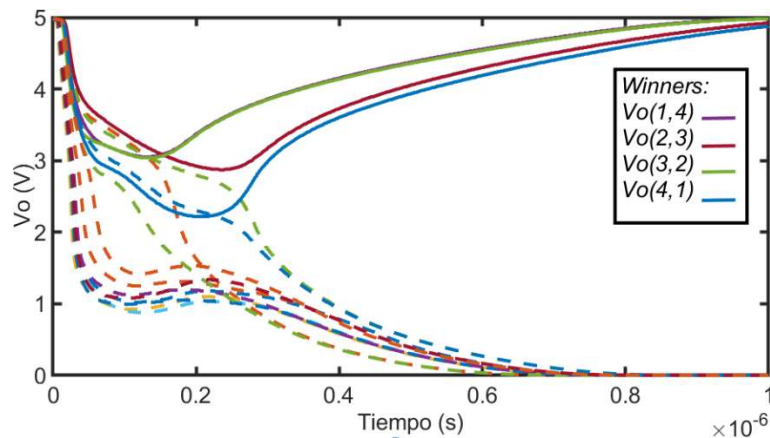
$$\mathbf{F}_2 = \begin{bmatrix} 0.50 & \mathbf{0.41} & 0.70 & 0.39 \\ \mathbf{0.40} & 0.08 & 0.55 & 0.35 \\ 0.35 & 0.50 & 0.40 & \mathbf{0.80} \\ 0.20 & 0.27 & \mathbf{0.76} & 0.61 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 \end{bmatrix}$$

Figura 4.7 Respuesta dinámica de las 16 neuronas para la solución del problema \mathbf{F}_2 . Los valores en negrita de la matriz \mathbf{F}_2 representan la solución obtenida con el algoritmo húngaro mientras que los valores en negrita de la matriz \mathbf{P}_2 indican la solución con el optimizador memristivo.



$$\mathbf{F}_3 = \begin{bmatrix} 0.55 & 0.37 & 0.15 & \mathbf{0.80} \\ \mathbf{0.85} & 0.45 & 0.26 & 0.47 \\ 0.67 & \mathbf{0.75} & 0.53 & 0.07 \\ 0.81 & 0.52 & \mathbf{0.79} & 0.77 \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \end{bmatrix}$$

Figura 4.8 Respuesta dinámica de las 16 neuronas para la solución del problema \mathbf{F}_3 . Los valores en negrita de la matriz \mathbf{F}_3 representan la solución obtenida con el algoritmo húngaro mientras que los valores en negrita de la matriz \mathbf{P}_3 indican la solución con el optimizador memristivo.



$$\mathbf{F}_4 = \begin{bmatrix} 0.55 & 0.80 & 0.15 & \mathbf{0.80} \\ 0.85 & 0.45 & \mathbf{0.85} & 0.47 \\ 0.67 & \mathbf{0.75} & 0.53 & 0.75 \\ \mathbf{0.81} & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad \mathbf{P}_4 = \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 \end{bmatrix}$$

Figura 4.9 Respuesta dinámica de las 16 neuronas para la solución del problema \mathbf{F}_4 . Los valores en negrita de la matriz \mathbf{F}_4 representan la solución obtenida con el algoritmo húngaro mientras que los valores en negrita de la matriz \mathbf{P}_4 indican la solución con el optimizador memristivo.

A continuación, se analizan los voltajes finales alcanzados para una neurona ganadora y una perdedora para el primer problema: matriz F_1 . Con base en la matriz de permutación P_1 se observa que la neurona $N(2,4)$ es de tipo perdedora ($V_o(2,4) = 0$) (en negrita), mientras que la neurona $N(1,2)$ es de tipo ganadora ($V_o(1,2) = 1$) (en negrita).

$$F_1 = \begin{bmatrix} 0.25 & 0.56 & 0.82 & 0.29 \\ 0.50 & 0.62 & 0.90 & 0.48 \\ 0.70 & 0.48 & 0.20 & 0.50 \\ 0.63 & 0.24 & 0.06 & 0.37 \end{bmatrix} \quad P_1 = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 \end{bmatrix}$$

En la **Figura 4.10a** se muestran los voltajes alcanzados por la neurona perdedora $N(2,4)$, donde las terminales de entrada con señal de 0V indican que la neurona que se conecta con dicha terminal es perdedora, mientras que las terminales con una señal de 5V indican que la neurona conectada a dicha terminal es ganadora. En la **Figura 4.10b** se observa la evolución de la memristancia de todos los memristores que conforman la neurona $N(2,4)$; aquellos, memristores conectados a salidas de neuronas perdedoras cambian de su estado inicial a su máximo valor R_{OFF} mientras que los memristores conectados con las salidas de neuronas ganadoras conmutan a su mínimo valor R_{ON} .

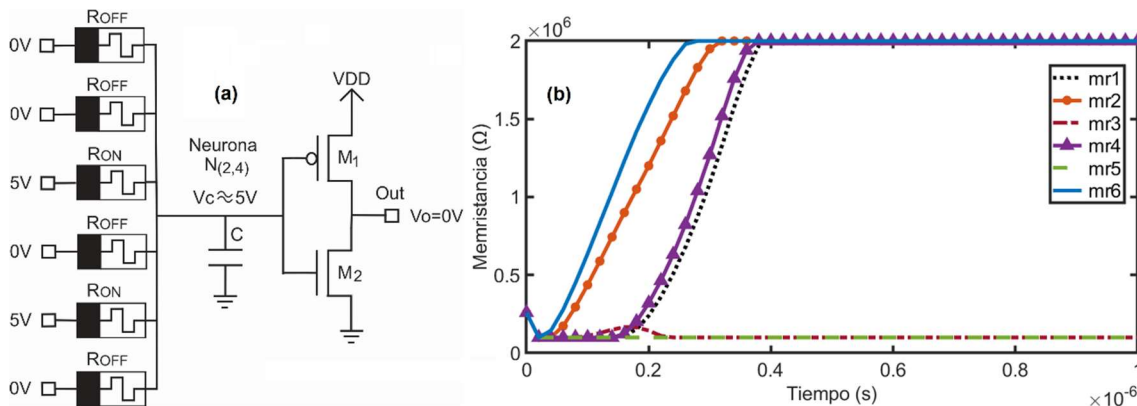


Figura 4.10a) Voltajes alcanzados por una neurona perdedora. **b)** Memristancia de los memristores de una neurona perdedora.

Para el caso de la neurona ganadora $N(1,2)$ (ver la **Figura 4.10c**) todas las neuronas conectadas a las terminales de entrada son del tipo perdedoras, por lo

que la memristancia de todos los memristores conmutan de su estado inicial a su máximo valor R_{OFF} . Esto se puede ver gráficamente en la **Figura 4.10d**.

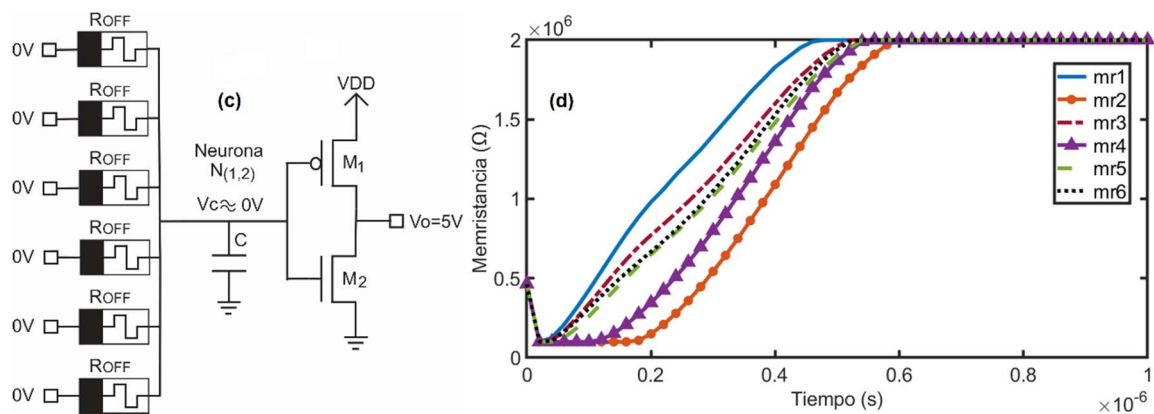


Figura 4.10c) Voltajes alcanzados por una neurona ganadora. **d)** Memristancia de los memristores de una neurona ganadora.

4.7 Conclusiones

Se presentó el desarrollo de un optimizador CMOS/memristor capaz de calcular en forma paralela la solución al problema de asignación de tareas. Como un vehículo experimental se usaron cuatro matrices numéricas F_1 , F_2 , F_3 y F_4 demostrando, vía simulaciones eléctricas en SPICE, que el circuito analógico memristivo propuesto es estable y capaz de encontrar las respectivas matrices solución P_1 , P_2 , P_3 y P_4 , para de esta manera obtener el costo máximo del sistema. Al ser un sistema de competencia que trabaja en forma paralela, el tiempo de procesamiento es de aproximadamente $1\mu s$. Estos resultados se compararon con el algoritmo húngaro (ver **Apéndice E**) presentando una respuesta similar; de esta manera se valida un medio para la solución al problema de asignación de tareas a nivel de hardware, además, de probar el uso adecuado de memristores con transistores CMOS. Este circuito memristivo puede ser visto como un arreglo lógico de compuertas NOR, el cual se comporta como un sistema competitivo de tipo ganador toma todo (Winner Take All). Las simulaciones eléctricas se realizaron usando la Suite de desarrollo IC-Nanometer de Mentor Graphics. Para los transistores CMOS se utilizó tecnología estándar de pozo n de 0.5 micras.

Referencias

- [1] H.A. Taha, Operations research: an introduction. Pearson Prentice Hall, USA (2007).
- [2] J. Borhetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R.S. Williams, Memristive switches enable stateful logic operations via material implication, Nature 464 (2010) 873-876.
- [3] S. Shin, K. Kim, S.M. Kang, Reconfigurable stateful NOR gate for large scale logic array integrations, IEEE Trans. Circuit Syst. 58 (7) (2011) 442-446.
- [4] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E.G. Friedman, A. Kolodny, U.C. Weiser, Magic memristor aided logic, IEEE Trans. Circuit Syst. 61 (11) (2014) 895-899.
- [5] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, E.G. Friedman, MRL memristor Ratioed logic, CNNA 2012 13th international workshop, Turin Italy.
- [6] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, TEAM: threshold adaptive memristor model, IEEE Trans. Circuit Syst. 60 (1) (2013) 211-221.
- [7] S. Kvatinsky, K. Talisveyberg, D. Fliter, E.G. Friedman, A. Kolodny, U.C. Weiser, Models of memristors for SPICE simulations, 2012 IEEE 27th Convention of electrical and electronics engineers in Israel.

Capítulo 5

Implementación digital

5.1 Introducción

En [1, 2] se presentó una metodología a nivel de software para el modelado y simulación del cuarto elemento fundamental de circuitos, el memristor, basado en el uso de las herramientas de desarrollo Simulink y Simscape. La herramienta Simulink proporciona un potente entorno de modelado de alto nivel para sistemas de procesamiento de señales digitales (DSP) por lo que, se utiliza ampliamente para el desarrollo y verificación de algoritmos. Este entorno es de la forma entrada/salida, por lo que el memristor no puede ser visto como un elemento de dos terminales, sino como una caja negra con una entrada (voltaje) y una salida (corriente). Para solucionar este esquema se utilizó la herramienta de modelado Simscape, perteneciente a Simulink, la cual permite modelar y simular sistemas físicos de las áreas técnicas (ingeniería eléctrica, hidráulica, mecánica etc.). Esta herramienta es capaz de distinguir valores físicos y unidades, además de permitir trabajar conjuntamente con componentes de Simulink haciendo que el modelado de sistemas sea más eficiente y poderoso. De esta manera, se presenta al memristor como un elemento de dos terminales de tipo flotado, capaz de interactuar con otros componentes eléctricos, lo que permite analizar, diseñar y simular aplicaciones de circuitos eléctricos.

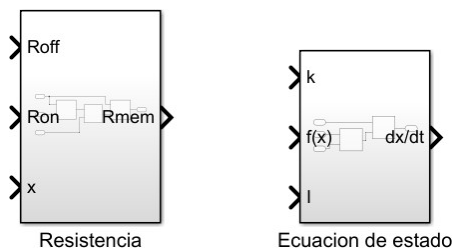
En este capítulo se presenta una propuesta (**Hardware/Software**) [3] para el modelado y simulación del memristor que combina las herramientas Simulink, Simscape y Xilinx System Generator. Además, se utiliza el método denominado co-simulación [4, 5], el cual consiste en verificar una parte del diseño a nivel de

hardware y otra a nivel de software, reduciendo de esta manera los tiempos de diseño y validación de un sistema. (Xilinx System Generator es una herramienta de software que permite diseñar e implementar, de forma sencilla y rápida, sistemas DSP a nivel de hardware FPGA (**Field Programmable Gate Array**)). A diferencia de la metodología propuesta en [1, 2] la descripción y cálculo de las expresiones características del memristor se realizaron a nivel de hardware (FPGA), mientras que la excitación y visualización de los resultados a nivel software (MATLAB). En este Capítulo se realiza la implementación digital de los modelos: **lineal con función de ventana** [6,11,12], **TEAM: ThrEshold Adaptive Memristor** [7] y **VTEAM: Voltage ThrEshold Adaptive Memristor** [8], estudiados en el capítulo 2. El dispositivo lógico programable utilizado fue un FPGA de la familia Artix-7 con una tarjeta Nexys 4; también se usa una PC con Intel Core™ i7 2.66 GHZ CPU, 8 GB RAM, corriendo en un sistema operativo Windows 7 Professional. Los cálculos se realizaron con aritmética de punto flotante de 32 bits para de esta manera obtener una mayor precisión al comparar la respuesta numérica de los modelos analíticos con la respuesta en hardware de los modelos implementados digitalmente.

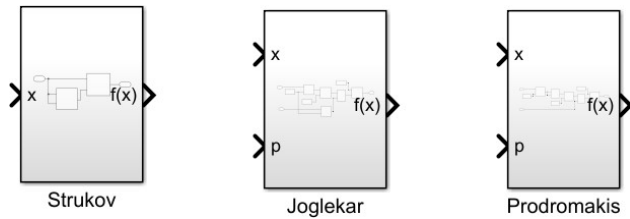
5.2 Modelo lineal con función de ventana

Como una primera implementación digital se desarrolló un sistema capaz de emular la respuesta numérica del modelo lineal con función de ventana en un dispositivo lógico programable FPGA. El sistema propuesto se conforma de un total de ocho bloques, los cuales, a su vez, contienen bloques de propiedad intelectual (IP) pertenecientes a la herramienta de desarrollo Xilinx System Generator capaces de proporcionar un rango de funcionalidad desde operaciones aritméticas hasta funciones complejas para el procesamiento digital de señales, además de estar diseñados para correr a altas velocidades y ser eficientes en área [9, 10]. Estos ocho bloques se dividen en tres subconjuntos: 1) Expresiones del modelo: son bloques que describen las expresiones matemáticas características del modelo (Ecuación de estado y Resistencia). 2) Funciones de ventana: son bloques que resuelven las expresiones de las ventanas propuestas por Strukov, Joglekar y Prodromakis. 3) Elementos matemáticos y de control: son bloques encargados de controlar el flujo de la información a través del sistema y de realizar ciertas operaciones matemáticas (Integración y Potencia). (ver la Figura 5.1). Todos los bloques se realizaron en el entorno gráfico de Simulink para su posterior implementación a nivel Hardware. A continuación, se presenta una descripción sobre el funcionamiento y configuración de cada bloque.

Expresiones del modelo



Funciones de ventana



Elementos matemáticos y de control

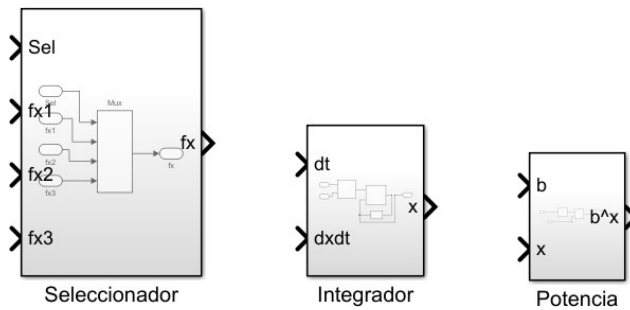


Figura 5.1 Bloques del sistema digital para la emulación del modelo no lineal de arrastre de dopantes.

5.2.1 Ecuación de estado

Una expresión característica del memristor es la ecuación de estado, la cual como se explicó en el capítulo 2, es la encargada de representar la velocidad de difusión de arrastre de dopantes (vacancias de oxígeno) a lo largo del dispositivo (5.1), donde μ_v es la movilidad iónica de arrastre promedio, D la distancia entre los electrodos del memristor, y x es el ancho normalizado de la región dopada. La representación y evaluación digital de dicha expresión se realizó con el bloque denominado Ecuación de estado, el cual está conformado a partir de dos bloques IP de tipo multiplicador (**ver la Figura 5.2**).

$$\frac{dx}{dt} = ki(t)f(x), \quad k = \frac{\mu_v R_{on}}{D^2} \quad (5.1)$$

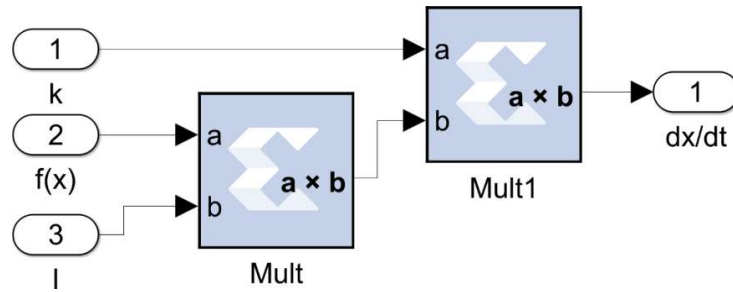


Figura 5.2 Arquitectura del bloque Ecuación de estado para la evaluación de la velocidad de difusión de arrastre de dopantes.

5.2.2 Resistencia

El cálculo de la resistencia total del memristor, R_{mem} , la cual representa la suma de las resistencias de la región dopada y no dopada del memristor (5.2), se llevó a cabo con el bloque denominado Resistencia. Con el objetivo de reducir el uso de los recursos lógicos disponibles del FPGA, se desarrolló algebraicamente (5.2) para tener $R_{mem}(x) = R_{off} - (R_{off} - R_{on})x$, la cual se conformó digitalmente por los bloques IP mostrados en la **Figura 5.3**. Este bloque ayuda a representar la relación entre el voltaje aplicado y la corriente del memristor a partir de la ley de Ohm $v(t) = R_{mem}(x)i(t)$.

$$R_{mem}(x) = R_{on}x + R_{off}(1 - x) \quad (5.2)$$

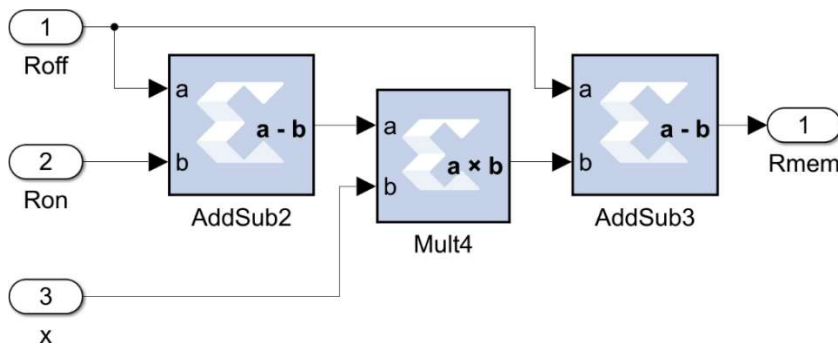


Figura 5.3 Arquitectura del bloque Resistencia para el cálculo de la resistencia total del memristor.

5.2.3 Funciones de ventana

Como se mencionó en [6], en dispositivos de escala nanométrica, pequeños voltajes pueden producir campos eléctricos elevados, los cuales producen no linealidades en el transporte iónico manifestándose en los bordes de la película delgada, donde la velocidad de la frontera entre la región dopada y no dopada gradualmente decrece a cero. Este fenómeno denominado arrastre no lineal de dopantes puede ser modelado por una función de ventana $f(x)$. En este sistema se realizó la descripción digital de las funciones de ventana propuestas por: Strukov [6], Joglekar [11] y Prodromakis [12], las cuales se representan con las expresiones (5.3), (5.4) y (5.5), respectivamente. En las **Figuras 5.4, 5.5 y 5.6** se muestran los elementos digitales empleados para cada ventana.

Strukov

$$f(x) = x - x^2 \quad (5.3)$$

Joglekar donde: p es un valor entero positivo.

$$f(x) = 1 - (2x - 1)^{2p} \quad (5.4)$$

Prodromakis donde: p es un número entero positivo y j es un número real positivo.

$$f(x) = j(1 - [(x - 0.5)^2 + 0.75]^p) \quad (5.5)$$

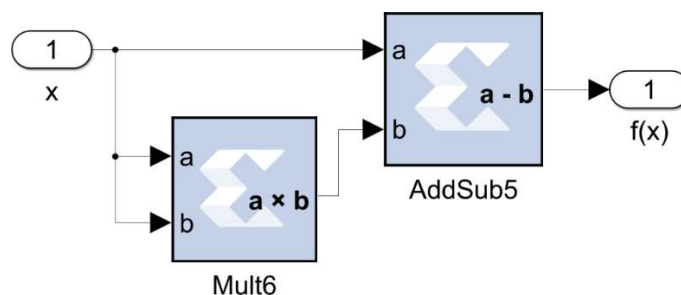


Figura 5.4 Arquitectura del bloque para la función de ventana de Strukov.

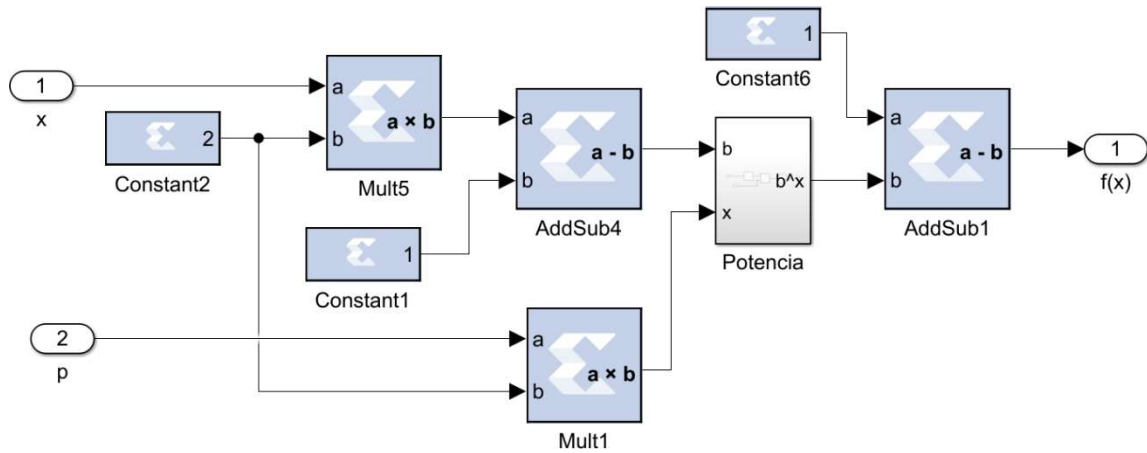


Figura 5.5 Arquitectura del bloque para la función de ventana de Joglekar.

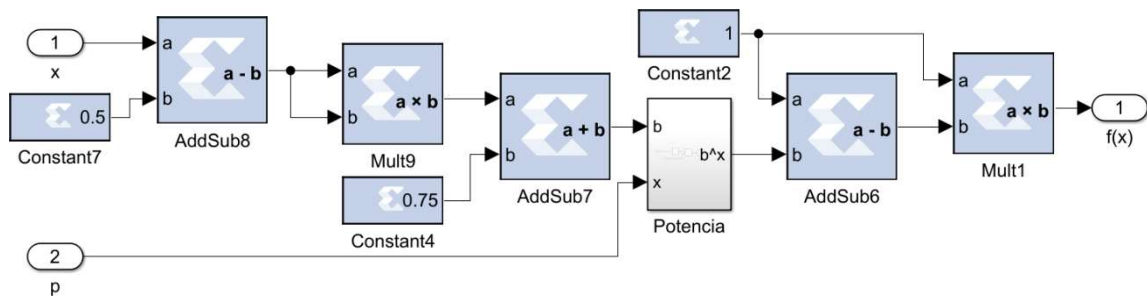


Figura 5.6 Arquitectura del bloque para la función de ventana de Prodromakis con $j = 1$.

5.2.4 Potencia

A pesar de que el toolbox Xilinx Blockset de System Generator cuenta con una variedad de componentes capaces de realizar funciones aritméticas, este no presenta un componente dedicado para la función aritmética de potencia, por lo que se realizó un bloque para dicha tarea. Este bloque se conforma con los componentes mostrados en la **Figura 5.7** y es de suma importancia ya que permite controlar la linealidad del modelo con las funciones de ventana (5.4) y (5.5).

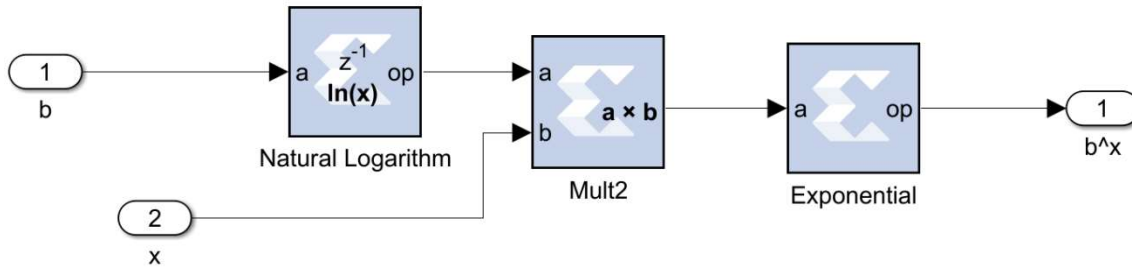


Figura 5.7 Arquitectura del bloque para el cálculo de la potencia.

5.2.5 Integrador

La ecuación de estado (5.1) es una ecuación diferencial, la cual al ser resuelta, proporciona la variable de estado x encargada de indicar la posición de la frontera que divide la región dopada y no dopada del dispositivo. El bloque denominado Integrador resuelve digitalmente esta expresión a partir de sumas sucesivas y se conforma con los componentes presentados en la **Figura 5.8**. El elemento nombrado (Register), almacena el valor inicial de la variable de estado x_{init} para el tiempo $t = 0$.

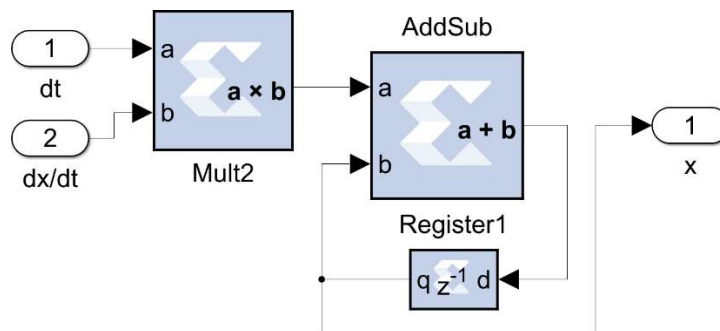


Figura 5.8 Arquitectura del bloque para el cálculo de la integral por medio de sumas sucesivas.

5.2.6 Seleccionador

Como se mencionó anteriormente, el sistema propuesto es capaz de evaluar el modelo lineal para tres funciones de ventana. Por lo que se desarrolló un bloque que controla la función de ventana a utilizar. Este bloque únicamente es

conformado por un multiplexor 3 a 1 y es controlado por la llave de selección Sel (ver la Figura 5.9).

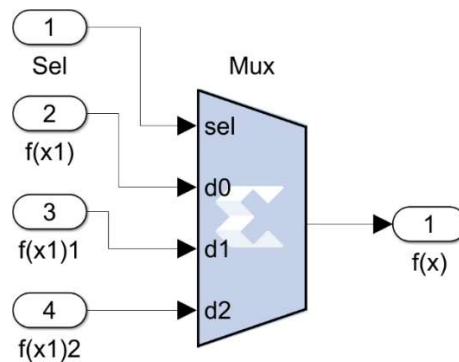


Figura 5.9 Arquitectura del bloque Seleccionador para las diferentes funciones de ventana.

La **Tabla 5.1** presenta las distintas combinaciones de la llave de selección Sel para la elección de las diferentes funciones de ventana.

Tabla 5.1 Combinaciones de la llave de selección Sel

Selección	Función de ventana
Sel = 00	Strukov
Sel = 01	Joglekar
Sel = 10	Prodromakis

5.2.7 Diagrama completo

Una vez descritos los ocho bloques con componentes de la herramienta Xilinx System Generator, se realizó la interconexión entre ellos para formar un sistema digital capaz de emular la respuesta numérica del modelo no lineal de arrastre de dopantes (ver la Figura 5.10). Este sistema presenta siete terminales de entrada y dos terminales de salida. Los bloques (In) se utilizan para convertir el tipo de dato proveniente de Simulink al formato de punto flotante utilizado por el sistema de hardware modelado con System Generator, mientras que los bloques (Out) son utilizados para realizar el proceso contrario.

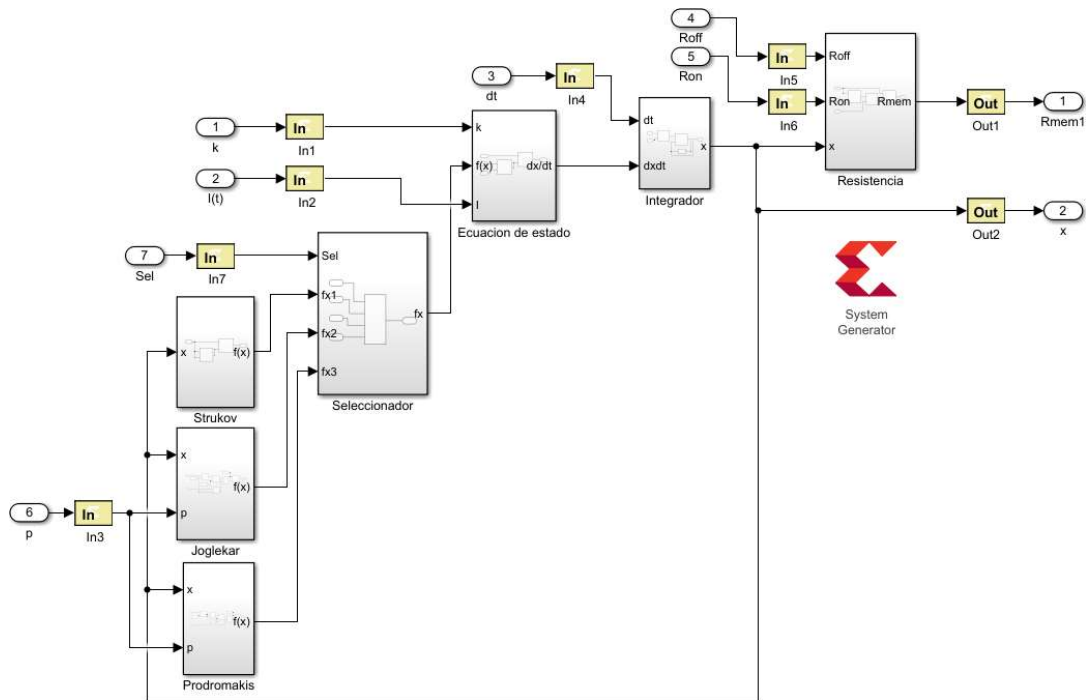


Figura 5.10 Diagrama completo del sistema para el cálculo del modelo no lineal de arrastre de dopantes.

5.2.8 Generación del módulo para Co-simulation

Se construyó una versión en hardware del sistema mostrado en la **Figura 5.10** utilizando el método Hardware/Software Co-simulation de System Generator. Para esto se realizó el siguiente procedimiento descrito en [10]:

- 1) Doble clic en el icono denominado System Generator de la **Figura 5.10** para desplegar el siguiente cuadro de diálogo (**ver la Figura 5.11**).

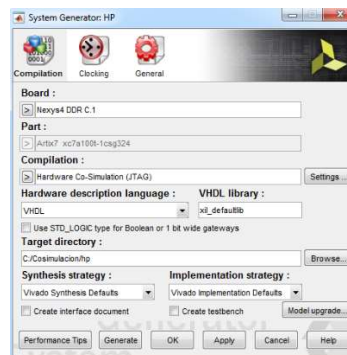


Figura 5.11 Cuadro de diálogo para la configuración del dispositivo FPGA.

- 2) En la pestaña de la sección **Board** seleccionar el nombre de la tarjeta a utilizar, en este caso, se utilizó la tarjeta **Nexys4 DDR C.1**.
- 3) Indicar en **Compilation** y **Hardware description language** el tipo de compilación a utilizar (Hardware Co-Simulation (JTAG)) y el lenguaje de descripción de hardware (VHDL).
- 4) Dar clic en el botón **Generate** para crear el archivo **bit-stream**, el cual es usado para configurar el FPGA, además, este archivo es descargado en el dispositivo lógico durante el proceso de Co-simulación.
- 5) Finalmente, cuando el proceso de compilación termina, se crea la librería denominada `hp_hwcosim_lib`, la cual almacena el archivo bit-stream y el bloque `HP_hwcosim`, que sirve como enlace con dicho archivo (**ver la Figura 5.12**).

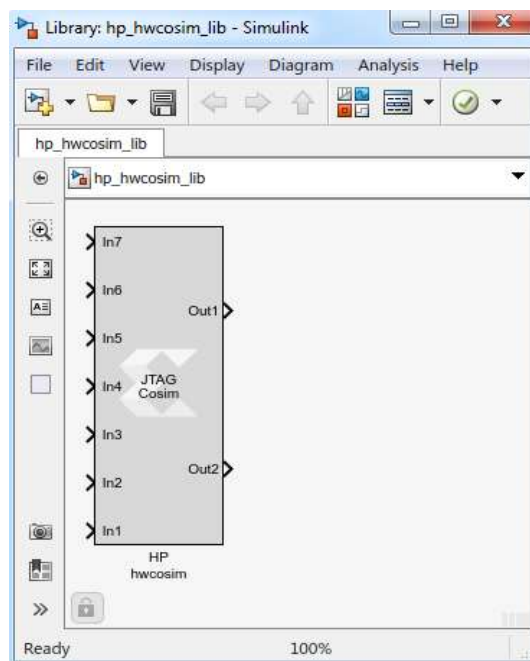


Figura 5.12 Librería y bloque generados para la simulación del sistema.

Para poder realizar la Co-simulación del sistema se utilizaron: 1) bloques de Simulink, utilizados para definir los parámetros de entrada del modelo, 2) bloque `HP_hwcosim`, el cual permite que el sistema sea ejecutado en el FPGA y 3) componentes eléctricos de la librería Simscape (**ver la Figura 5.13**).

Se utilizó un resistor variable como representación eléctrica del memristor, el cual cumple con la característica de ser un elemento pasivo y flotado capaz de interactuar con otros elementos eléctricos. La salida Out1 del bloque HP_hwcosim es un valor numérico representativo de la resistencia total del dispositivo, pero este valor no puede ser utilizado directamente en el potenciómetro debido a que no es una señal física, es decir no tiene unidades, por lo que se empleó el bloque Simulink-PS Converter, para realizar esta conversión y de esta manera transferir un valor óhmico, con unidades, al potenciómetro. Al circular una señal AC sinusoidal de voltaje a través de las terminales del potenciómetro (P - N), se genera una señal de corriente, con unidades, cuyo valor no puede ser transferido directamente al bloque HP_hwcosim por lo que es necesario hacer una conversión con el bloque PS Simulink-Converter; de esta manera, se obtiene un valor numérico sin unidades representativo de la corriente que sirve como entrada del bloque HP_hwcosim.

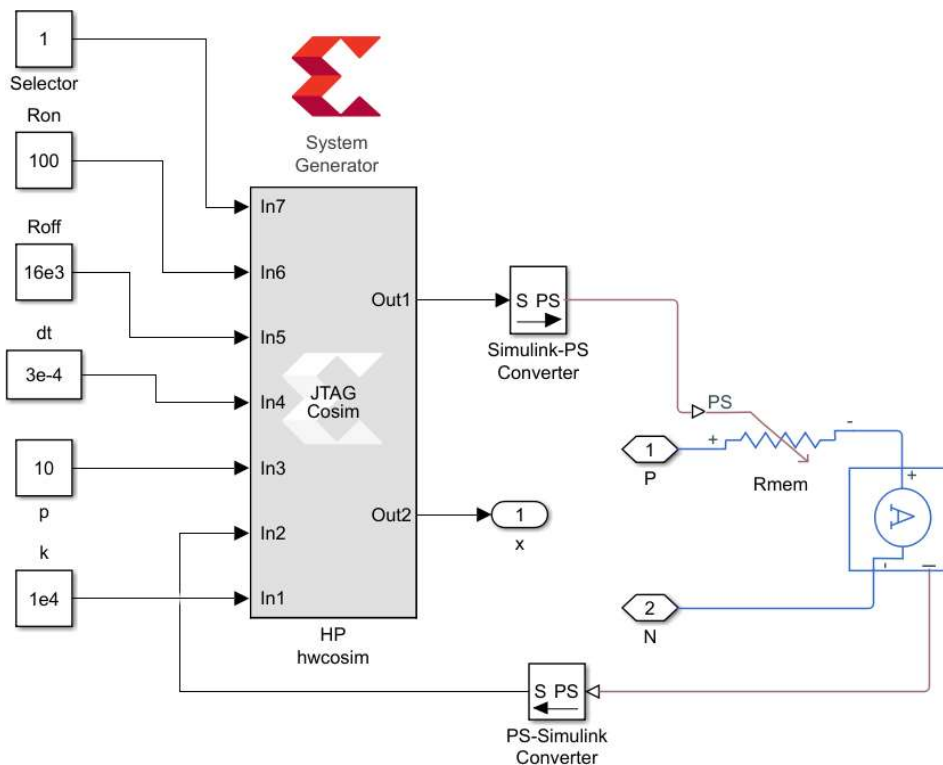


Figura 5.13 Diagrama con las tres herramientas utilizadas Simulink, Simscape y Xilinx System Generator.

5.2.9 Simulaciones eléctricas

Para fines de visualización, en la simulación, se creó un subsistema con todos los elementos de la **Figura 5.13** denominado **Memristor HP**. Este subsistema presenta únicamente tres terminales, entradas del memristor (P y N) y salida de la variable de estado (x) (**ver la Figura 5.14**). Se aplicó una señal sinusoidal de 1.2V pico, con una frecuencia $F_g = 0.3\text{mHz}$, la cual es equivalente a $F_0 = 1\text{Hz}$. Esta equivalencia se obtuvo con las expresiones (5.6) y (5.7) y se realizó debido a que los bloques (IP) de System Generator presentan un tiempo mínimo de simulación de 1s, esto con el fin de evitar problemas de sincronización al realizar la comunicación entre el hardware (FPGA) y el software (MATLAB). Por tal motivo el tiempo de simulación que se manejó es de segundos.

$$F_g = \frac{M}{F_0} C^2 \quad (5.6)$$

$$M = \frac{T_{F_0}}{dt} \quad (5.7)$$

donde F_0 es el valor de la frecuencia a convertir. F_g frecuencia equivalente de F_0 . T_{F_0} periodo de la señal F_0 . C número de ciclos presentes en un periodo de la señal F_0 . dt diferencial de tiempo.

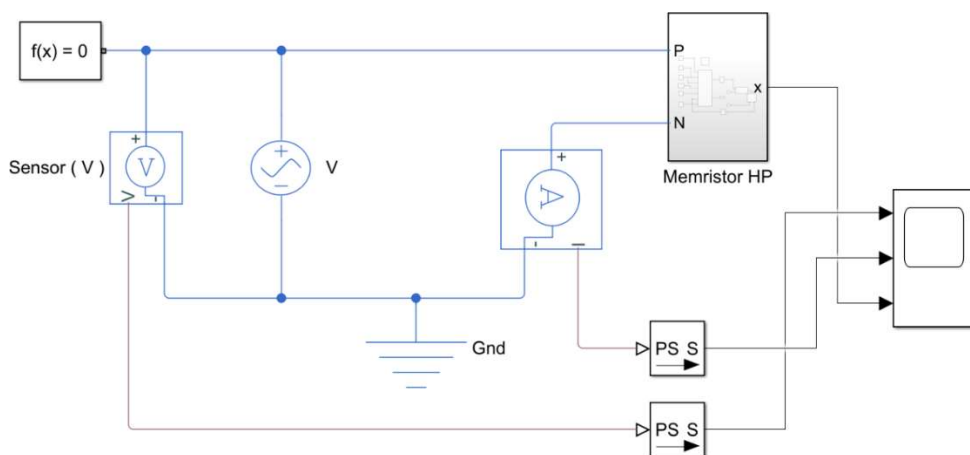


Figura 5.14 Diagrama eléctrico empleado en la Co-simulación y evaluación del sistema propuesto

Los resultados obtenidos se muestran en las **Figuras 5.15, 5.16 y 5.17** para la corriente del memristor, la dinámica de la resistencia y la curva V-I de histéresis, respectivamente. Cabe señalar que las expresiones del modelo lineal con función de ventana se resolvieron a nivel de Hardware (FPGA), mientras que la excitación del sistema y visualización de los resultados se realizó a nivel de Software (MATLAB) con las herramientas de desarrollo Simulink y Simscape. La comunicación entre el FPGA y el entorno de trabajo de Simulink se realizó a través del puerto JTAG y los resultados obtenidos con el sistema propuesto se compararon con los reportados en SPICE en [13] presentando una gran similitud.

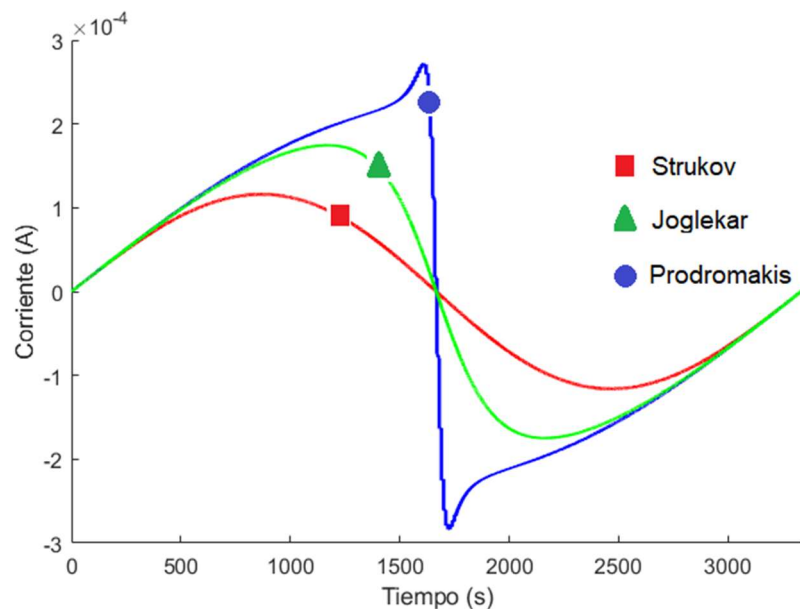


Figura 5.15 Corriente resultante del memristor para las tres funciones de ventana. Strukov $I_{mem} \approx 100 \mu A$, Joglekar $I_{mem} \approx 300 \mu A$ y Prodromakis $I_{mem} \approx 180 \mu A$ para un voltaje máximo aplicado de 1.2V.

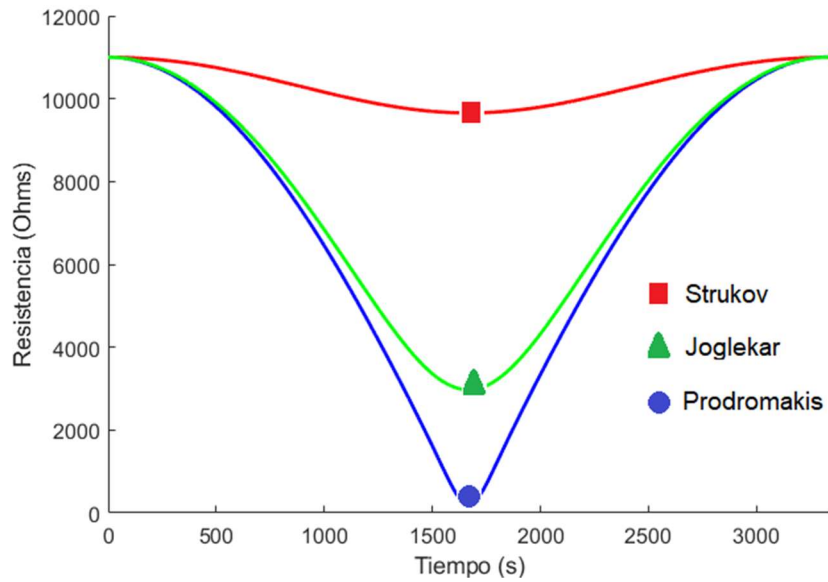


Figura 5.16 Variación de la resistencia del memristor para las tres funciones de ventana. Strukov $R_{mem}=11K\Omega-12K\Omega$, Joglekar $R_{mem}=0\Omega-11K\Omega$ y Prodromakis $R_{mem}=3K\Omega-11K\Omega$.

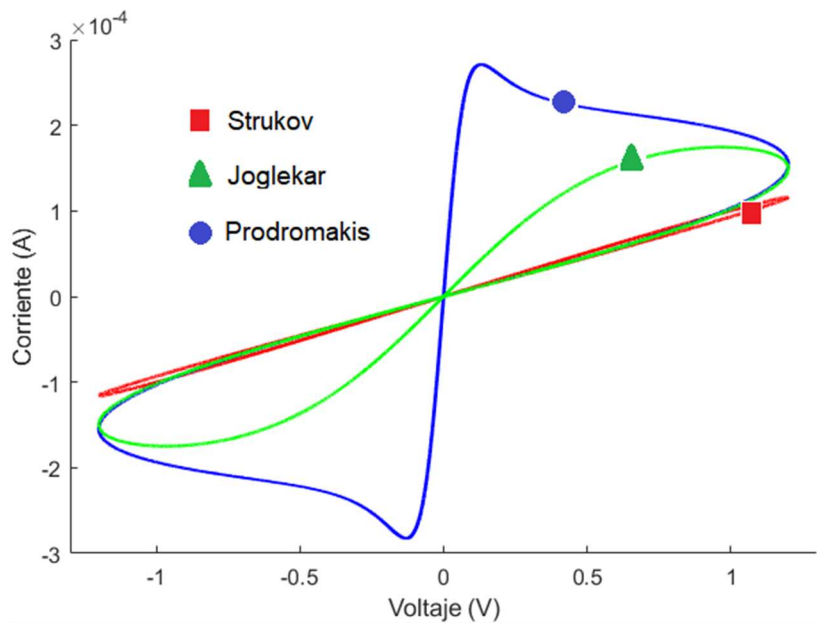


Figura 5.17 Curva de histéresis voltaje-corriente del memristor para las tres funciones de ventana.

En la **Tabla 5.2** se muestran los recursos lógicos utilizados para el desarrollo del sistema y en la **Tabla 5.3** se presentan los parámetros usados para el modelo.

Tabla 5.2 Recursos lógicos utilizados del FPGA

	DSPs (240)	LUTs (63400)	Registers (126800)
Ecuación de estado	6	202	0
Resistencia	3	887	0
Strukov	3	482	0
Joglekar	16	3346	32
Prodromakis	13	3640	32
Integrador	3	489	122
Total	44	9046	186

Tabla 5.3 Parámetros empleados del modelo lineal con función de ventana

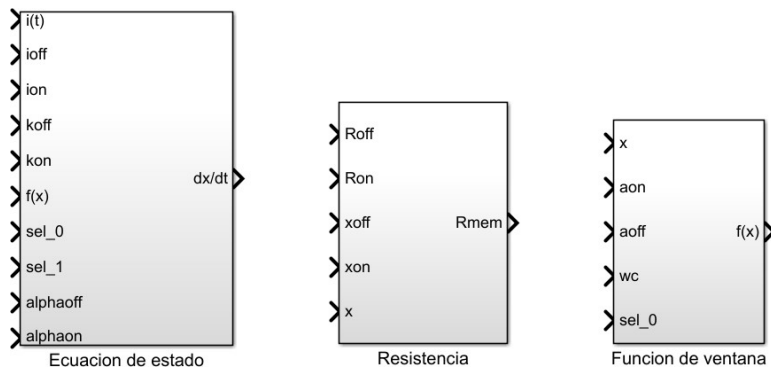
Parámetro	Valor	Comentario
dt	3×10^{-4} s	Diferencial de tiempo
R_{ON}	100 Ω	Memristancia en $x = 0$ m
R_{OFF}	16×10^3 Ω	Memristancia en $x = 10 \times 10^{-9}$ m
μ_v	1×10^{-14} m ² s ⁻¹ V ⁻¹	Movilidad de dopantes
x_{init}	1.5×10^{-9} m	Condición inicial de la variable de estado

5.3 Modelo TEAM

Como una segunda implementación digital, se realizó el desarrollo de un sistema capaz de emular la respuesta numérica del modelo de umbral TEAM en un dispositivo lógico programable FPGA. Para esta segunda implementación digital se siguió el mismo enfoque utilizado en la primera parte de este capítulo: co-simulación, la cual permitió simular una parte del sistema a nivel Hardware (FPGA) y otra a nivel de Software (MATLAB) utilizando el entorno de trabajo de Simulink. El sistema propuesto se conformó de un total de siete bloques, cada uno descrito con bloques (IP) pertenecientes a la herramienta de desarrollo Xilinx

System Generator. Estos siete bloques se dividen en dos subconjuntos: 1) Expresiones del modelo: son bloques que describen las expresiones matemáticas características del modelo TEAM (Ecuación de estado, Resistencia y Función de ventana). 2) Elementos matemáticos y de control: son bloques encargados de controlar el flujo de la información a través del sistema y de realizar operaciones matemáticas (Limitador, Control, Integrador y Potencia) (**ver la Figura 5.18**). A continuación, se presenta una descripción más detallada sobre el funcionamiento y configuración de cada bloque. Cabe señalar que la descripción de los bloques Potencia e Integrador no se presenta en esta sección debido a que su arquitectura y funcionamiento siguen el mismo principio de los bloques presentados en las secciones 5.2.4 y 5.2.5.

Expresiones del modelo



Elementos matemáticos y de control

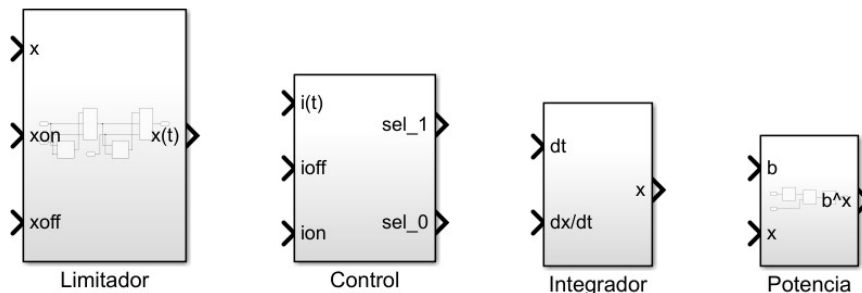


Figura 5.18 Bloques que integran el sistema digital para la emulación del modelo TEAM (ThrEshold Adaptive Memristor Model).

5.3.1 Control

Debido a que el modelo TEAM es de conmutación de tipo umbral de corriente, es decir la resistencia del memristor no cambia con corrientes menores a una corriente de umbral, se desarrolló un bloque capaz de comparar la corriente $i(t)$ que fluye a través de las terminales del memristor con las corrientes de umbral i_{on} e i_{off} . Los elementos digitales que conforman este bloque se presentan en la **Figura 5.19**, donde las terminales de salida sel_0 y sel_1 son las llaves de selección para los bloques: Función de ventana y Ecuación de estado. El funcionamiento del bloque se muestra en la **Figura 5.20** para las siguientes condiciones de operación del modelo: $i > i_{off}$, $i < i_{on}$ e $i_{on} < i < i_{off}$.

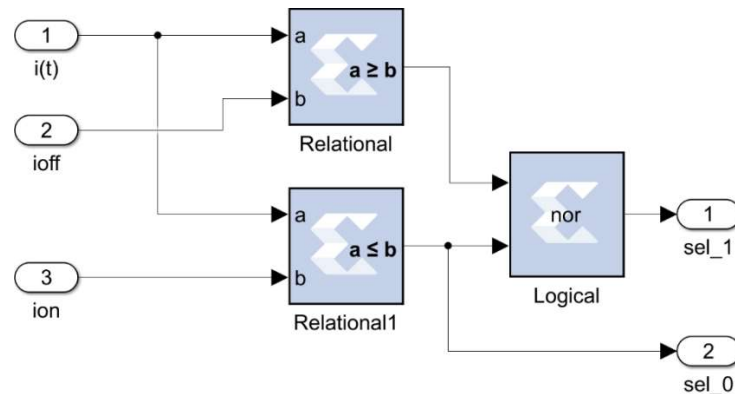


Figura 5.19 Arquitectura del bloque Control para las corrientes de umbral i_{off} e i_{on} .

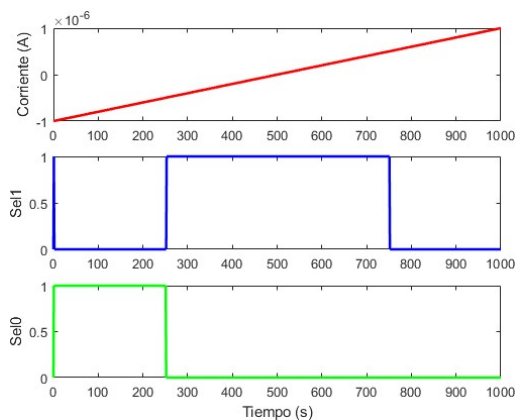


Figura 5.20 Respuesta del bloque de control para una corriente de $i = \pm 1\mu A$ y corrientes de umbral $i_{off} = 0.5\mu A$ $i_{on} = -0.5\mu A$.

5.3.2 Limitador

La función de este bloque es evitar que el valor de la variable de estado x sobrepase las dimensiones físicas del memristor, es decir, la variable de estado debe estar entre los valores x_{on} y x_{off} : dimensiones mínima y máxima del dispositivo. Para evitar esta situación es necesario cumplir con las siguientes restricciones:

Cuando

1. $x < x_{on} \rightarrow x = x_{on}$
2. $x > x_{off} \rightarrow x = x_{off}$
3. $x_{on} < x < x_{off} \rightarrow x = x$

Estas restricciones se implementaron utilizando multiplexores y comparadores conectados en una arquitectura de tipo cascada (ver la **Figura 5.21**).

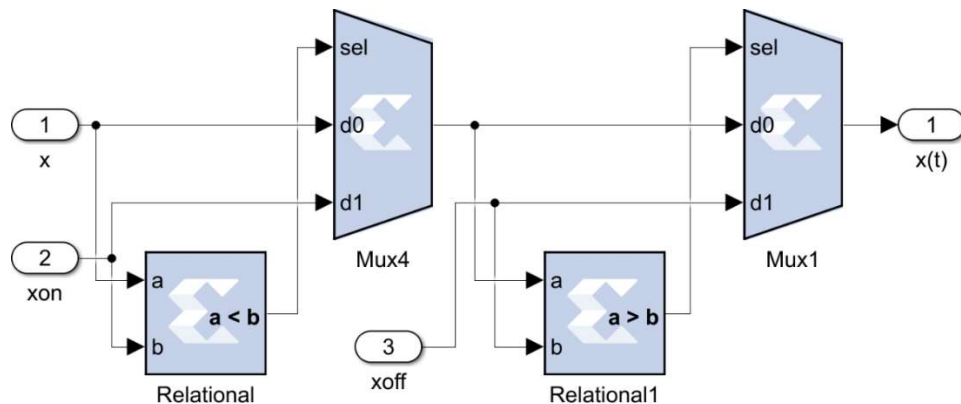


Figura 5.21 Arquitectura del bloque Limitador.

5.3.3 Ecuación de estado

Este bloque describe la ecuación de estado característica del modelo TEAM, la cual representa la velocidad de difusión de arrastre de dopantes a lo largo del dispositivo (5.8), donde k_{off} , k_{on} , α_{off} y α_{on} : parámetros de ajuste, i_{off} e i_{on} : corrientes de umbral y x : ancho efectivo de la barrera túnel. La representación y evaluación digital se realizó con el bloque denominado Ecuación de estado, el cual se conformó a partir de los elementos mostrados en la **Figura 5.22**. Se

propuso una arquitectura de tipo multiplexado con el fin de evaluar cada expresión dentro del mismo bloque y optimizar los recursos lógicos disponibles del FPGA (BRAMs (135), DSPs (240), LUTs (63400), Registers (126800)). Los multiplexores Mux, Mux1, Mux2 y Mux3 controlan los parámetros a utilizar para cada expresión en función de las llaves de selección sel_0 y sel_1. El control de estas llaves de selección se realizó por medio del bloque descrito en la sección 5.3.1.

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{i(t)}{i_{off}} - 1 \right)^{\alpha_{off}} f_{off}(x), & 0 < i_{off} < i \\ 0, & i_{on} < i < i_{off} \\ k_{on} \left(\frac{i(t)}{i_{on}} - 1 \right)^{\alpha_{on}} f_{on}(x), & i < i_{on} < 0 \end{cases} \quad (5.8)$$

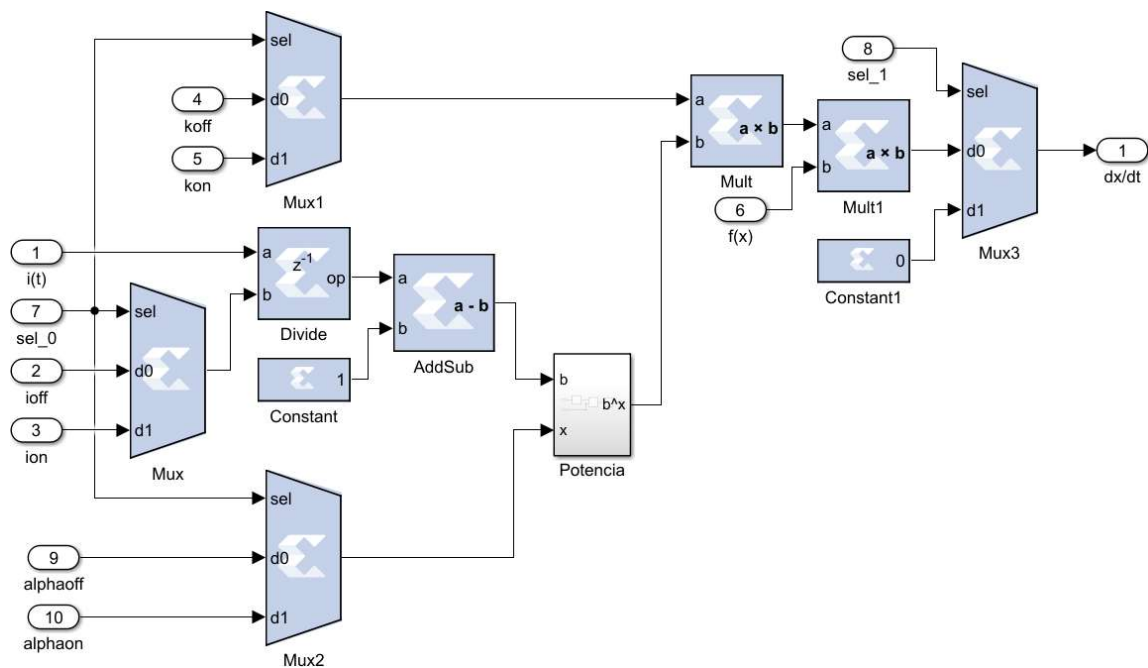


Figura 5.22 Arquitectura de tipo multiplexado del bloque Ecuación de estado para la evaluación de la velocidad de arrastre de dopantes.

La **Tabla 5.4** muestra los parámetros de salida de cada multiplexor, así como la expresión a evaluar para las diferentes combinaciones de las llaves de selección sel_0 y sel_1.

Tabla 5.4 Combinaciones de las llaves de selección sel_0 y sel_1

Selector	Mux	Mux1	Mux2	Expresión.	Selector	Mux3	Expresión
sel_0 = 0	i _{off}	k _{off}	α _{off}	5.8a	sel_1 = 0	dx/dt	5.8a o 5.8c
sel_0 = 1	i _{on}	k _{on}	α _{on}	5.8c	sel_1 = 1	0	5.8b

5.3.4 Función de ventana

La función de ventana que se implementó fue la propuesta por Kvatinsky [7], la cual se compone de dos expresiones (5.9a y 5.9b), donde a_{off} , a_{on} y w_c son parámetros de ajuste y x es el ancho efectivo de la barrera túnel. Además, cada expresión se evalúa en un rango de operación diferente: $i > 0$ e $i < 0$. Al tener dos expresiones, se implementó una arquitectura de tipo multiplexado con el fin de evaluar ambas expresiones en el mismo bloque optimizando los recursos lógicos del FPGA (ver la Figura 5.23).

$$f_{off}(x) = \exp \left[-\exp \left(\frac{x - a_{off}}{w_c} \right) \right] \quad i > 0 \quad (5.9a)$$

$$f_{on}(x) = \exp \left[-\exp \left(-\frac{x - a_{on}}{w_c} \right) \right] \quad i < 0 \quad (5.9b)$$

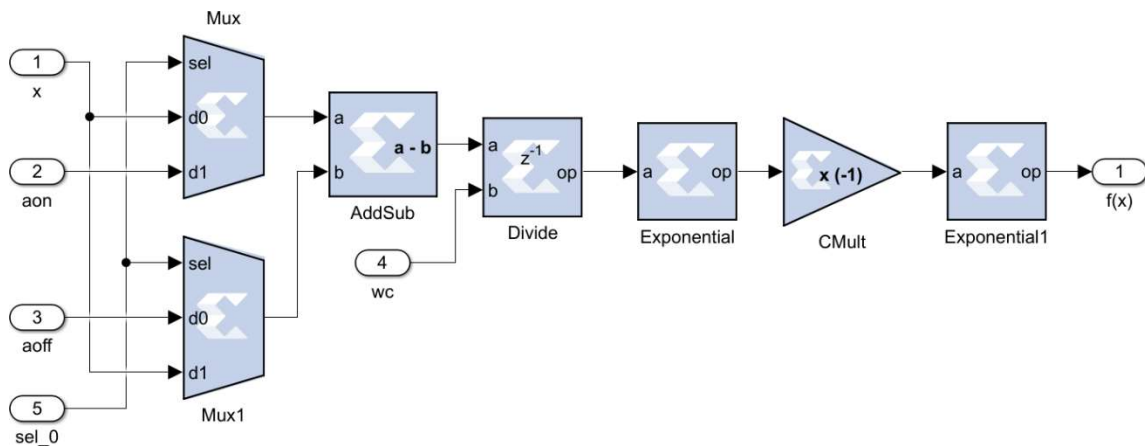


Figura 5.23 Arquitectura de tipo multiplexado del bloque Función de ventana de Kvatinsky.

La **Tabla 5.5** muestra los parámetros de salida de cada multiplexor y la expresión a evaluar para las diferentes combinaciones de la llave de selección sel_0. El control de esta llave de selección se realizó por medio del bloque descrito en la sección 5.3.1.

Tabla 5.5 Combinaciones de la llave de selección sel_0 para el control de los multiplexores

Selector	Mux	Mux1	Ecu.
sel_0 = 0	x(t)	a _{off}	5.9a
sel_0 = 1	a _{on}	x(t)	5.9b

5.3.5 Resistencia

Como se explica en [7] cualquier cambio en el ancho de la barrera tunel x provoca un cambio en la resistencia del dispositivo de forma exponencial (5.10), donde x_{off} , x_{on} son las dimensiones máxima y mínima del memristor; R_{OFF} , R_{ON} son los valores resistivos máximo y mínimo. El cálculo de la resistencia se realizó con los componentes presentados en la **Figura 5.24**. Esta expresión, altamente no lineal de tipo exponencial, ayuda a representar la relación entre el voltaje aplicado y la corriente del memristor a partir de la ley de Ohm $v(t) = R_{mem}(x)i(t)$.

$$R_{mem} = R_{ON}e^{(\lambda/x_{off}-x_{on})(x-x_{on})} \quad \lambda = \ln(R_{off}/R_{on}) \quad (5.10)$$

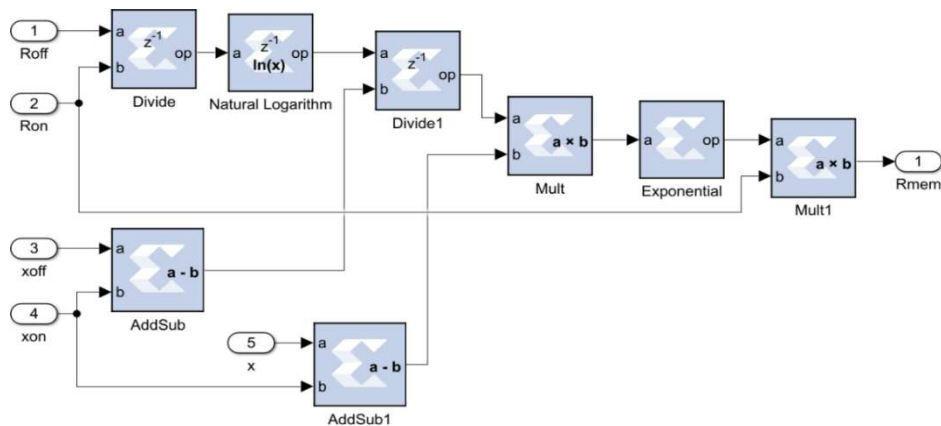


Figura 5.24 Arquitectura del bloque Resistencia para el cálculo de la resistencia total del memristor.

5.3.6 Diagrama completo

Una vez descritos los siete bloques con componentes de la herramienta Xilinx System Generator, se realizó la interconexión entre ellos para formar un sistema digital capaz de emular la respuesta numérica del modelo TEAM (ver la **Figura 5.25**). El procedimiento para la generación del módulo utilizado para la co-simulation fue el descrito en la sección 5.2.8. Por tal motivo no se abordará en esta sección.

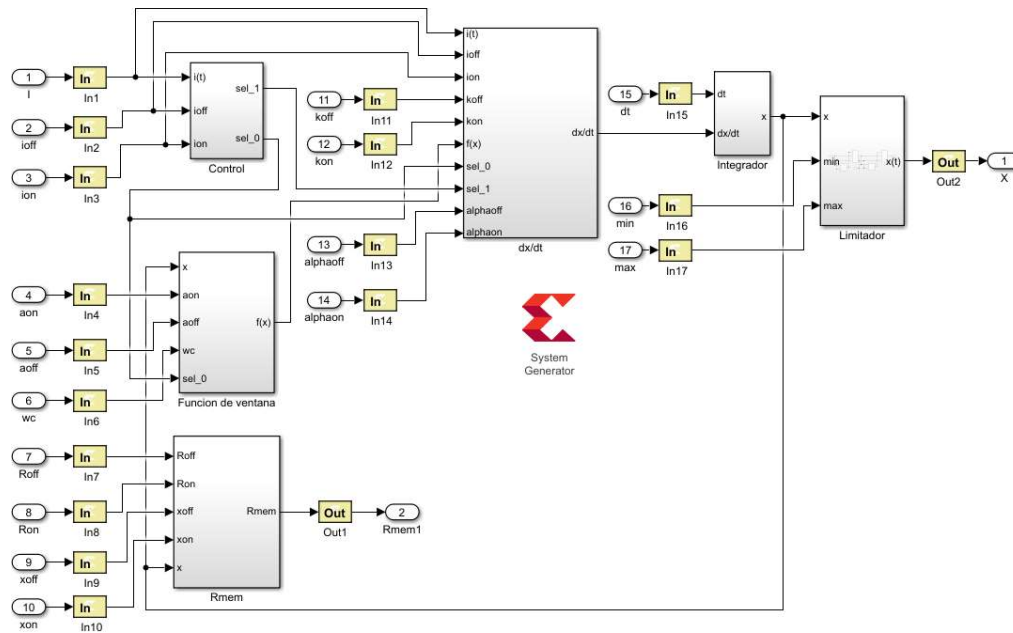


Figura 5.25 Diagrama completo del sistema para el cálculo del modelo TEAM.

5.3.7 Simulaciones eléctricas

Para validar el funcionamiento del sistema propuesto, se aplicó entre las terminales del bloque **Memristor TEAM** una señal sinusoidal de 1V pico con una frecuencia de $F_g = 1\text{mHz}$, la cual es equivalente a $F_0 = 1\text{MHz}$, esta equivalencia se obtuvo con las expresiones (5.6) y (5.7) (ver la **Figura 5.26**). Los resultados obtenidos de la co-simulación se muestran en las **Figuras 5.27, 5.28 y 5.29** para la corriente del memristor, la dinámica de la resistencia y la curva V-I de histéresis, respectivamente. Cabe señalar que, como en la primera implementación digital, las expresiones del modelo TEAM se resolvieron a nivel

de Hardware (FPGA), mientras que la excitación del sistema y la visualización de los resultados se realizaron a nivel de Software (MATLAB) con las herramientas de desarrollo Simulink y Simscape.

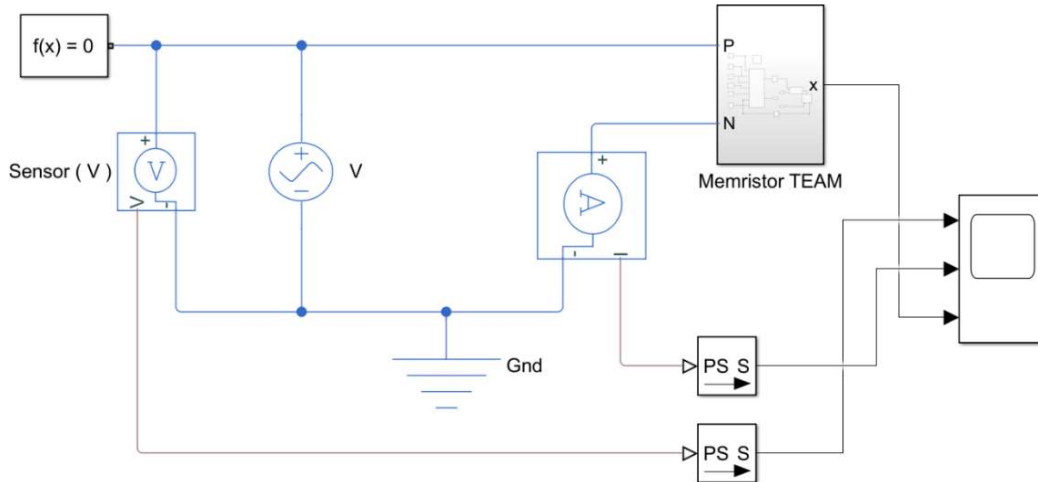


Figura 5.26 Diagrama eléctrico empleado en la Co-simulación y evaluación del sistema propuesto.

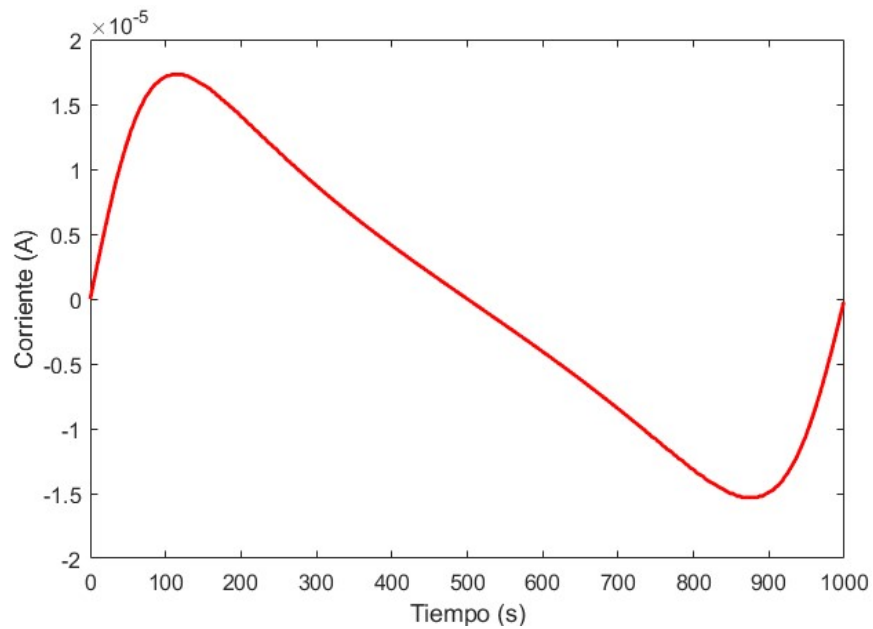


Figura 5.27 Corriente resultante del memristor $I_{mem} \approx 20 \mu A$ para un voltaje máximo aplicado de 1V.

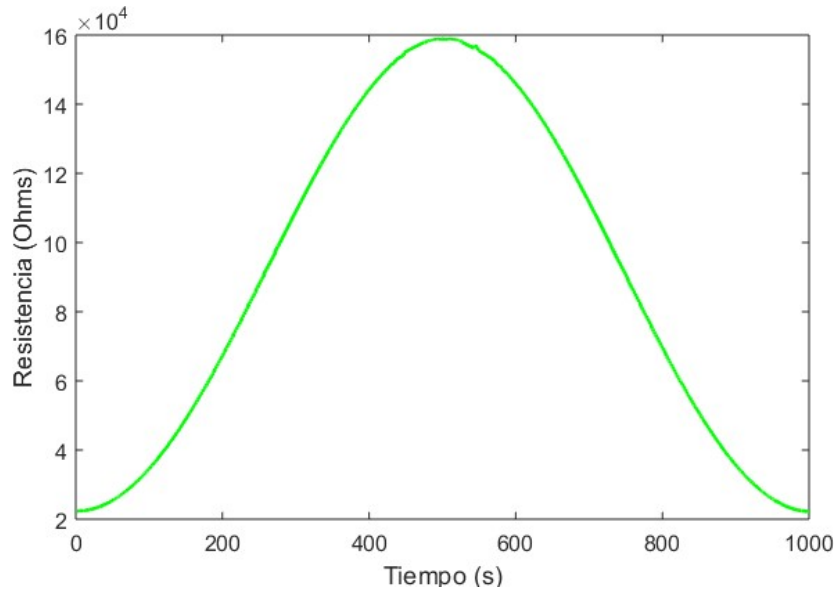


Figura 5.28 Variación de la resistencia del memristor $R_{mem} = 20K\Omega-160K\Omega$.

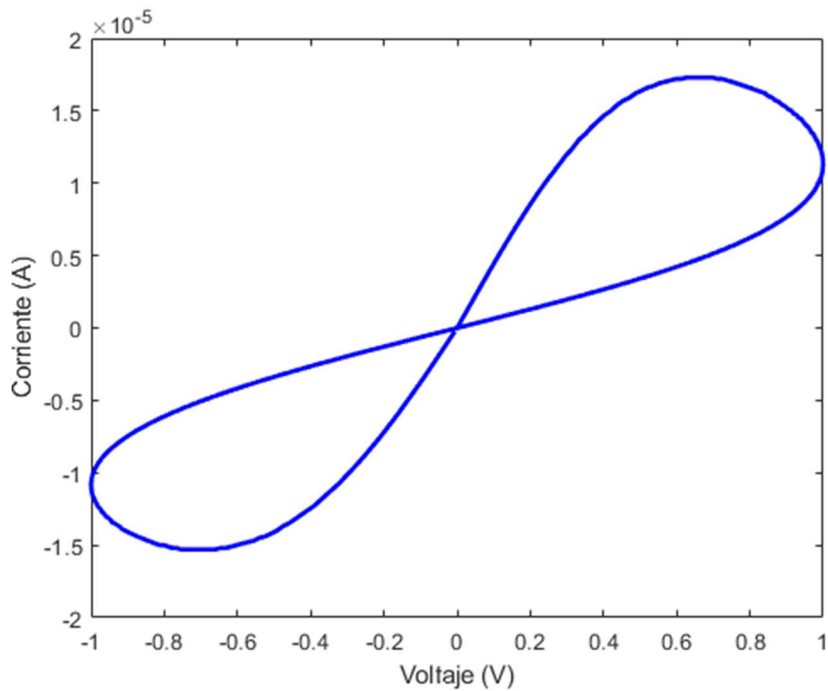


Figura 5.29 Curva de histéresis voltaje-corriente del memristor.

En la **Tabla 5.6** se muestran los recursos lógicos utilizados para el desarrollo del sistema y en la **Tabla 5.7** se presentan los parámetros usados para el modelo TEAM.

Tabla 5.6 Recursos lógicos utilizados del FPGA

	DSPs (240)	LUTs (63400)	Registers (126800)
Ecuación de estado	16	3802	64
Resistencia	13	4846	96
Función de ventana	14	2869	32
Control	0	120	0
Limitador	0	184	0
Integrador	3	552	32
Total	46	12373	224

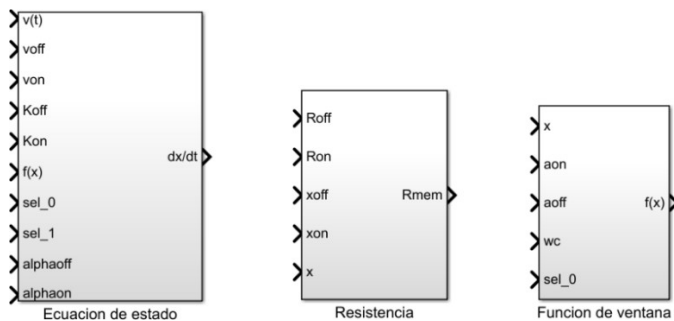
Tabla 5.7 Parámetros empleados para el modelo TEAM

Parámetro	Valor	Comentario
dt	1×10^{-9} s	Diferencial de tiempo
R_{ON}	1×10^3 Ω	Memristancia en $x = 0$ m
R_{OFF}	500×10^3 Ω	Memristancia en $x = 3 \times 10^{-9}$ m
i_{off}	50×10^{-12} A	Corriente de umbral
i_{on}	-50×10^{-12} A	Corriente de umbral
w_c	107×10^{-12} m	Parámetro de ajuste
a_{on}	0 m	Parámetro de ajuste
a_{off}	3×10^{-9} m	Parámetro de ajuste
k_{on}	-1×10^{-8} m/s	Parámetro de ajuste
k_{off}	1×10^{-8} m/s	Parámetro de ajuste
α_{on}	1	Parámetro de ajuste
α_{off}	1	Parámetro de ajuste
x_{on}	0 m	Límite inferior de x
x_{off}	3×10^{-9} m	Límite superior de x
x_{init}	1.5×10^{-9} m	Condición inicial de la variable de estado

5.4 Modelo VTEAM

Finalmente, como una tercera implementación digital se presenta el desarrollo de un sistema capaz de emular la respuesta numérica del modelo VTEAM en un dispositivo FPGA. Las expresiones matemáticas de este modelo presentan una similitud con las empleadas en el modelo TEAM, donde la única diferencia es el tipo de umbral (voltaje en lugar de corriente), por lo cual el modelo VTEAM se considera una extensión de dicho modelo. El sistema propuesto se conformó de siete bloques (**ver la Figura 5.30**), los cuales presentan la misma configuración y funcionamiento que los descritos en el modelo TEAM (**ver la Figura 5.18**) con la única diferencia del uso de voltaje en lugar de corriente. Se siguió el mismo enfoque usado en las secciones 5.2 y 5.3 para la validación del sistema, por tal motivo no se explicará en esta sección y solo se mostrarán los resultados.

Expresiones del modelo



Elementos matemáticos y de control

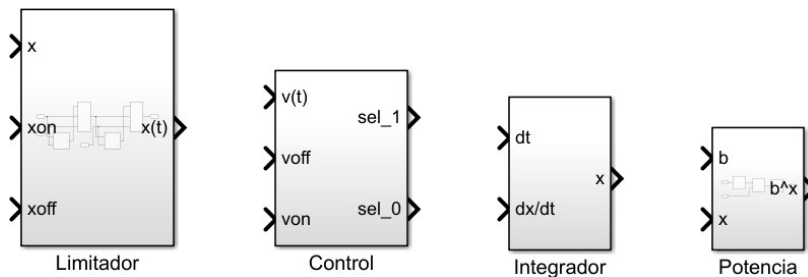


Figura 5.30 Bloques que integran el sistema digital para la emulación del modelo VTEAM (Voltage ThrEshold Adaptive Model).

5.4.1 Simulaciones eléctricas

Para validar el funcionamiento del sistema propuesto, se aplicó una señal sinusoidal de 1V pico con una frecuencia de $F_g = 5\text{mHz}$, la cual es equivalente a $F_0 = 5\text{MHz}$, esta equivalencia se obtuvo con las expresiones (5.6) y (5.7). Los resultados obtenidos de la Co-simulación se muestran en las **Figuras 5.31, 5.32** y **5.33** para la corriente del memristor, la dinámica de la resistencia y la curva V-I de histéresis, respectivamente. En la **Tabla 5.8** se muestran los recursos lógicos utilizados para el desarrollo del sistema y en la **Tabla 5.9** se presentan los parámetros usados para el modelo VTEAM.

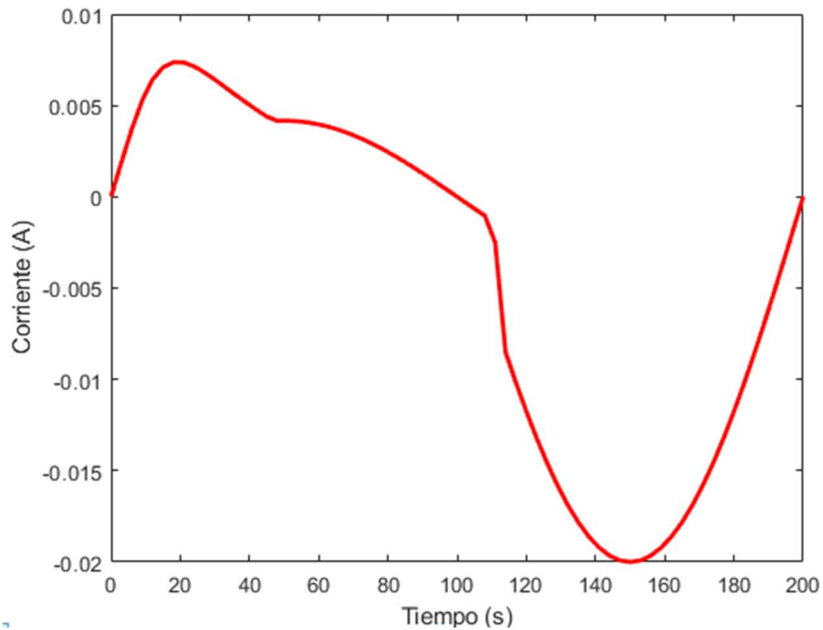


Figura 5.31 Corriente resultante del memristor $I_{\text{mem}} \approx 5 \text{ mA}$ para un voltaje máximo aplicado de 1V.

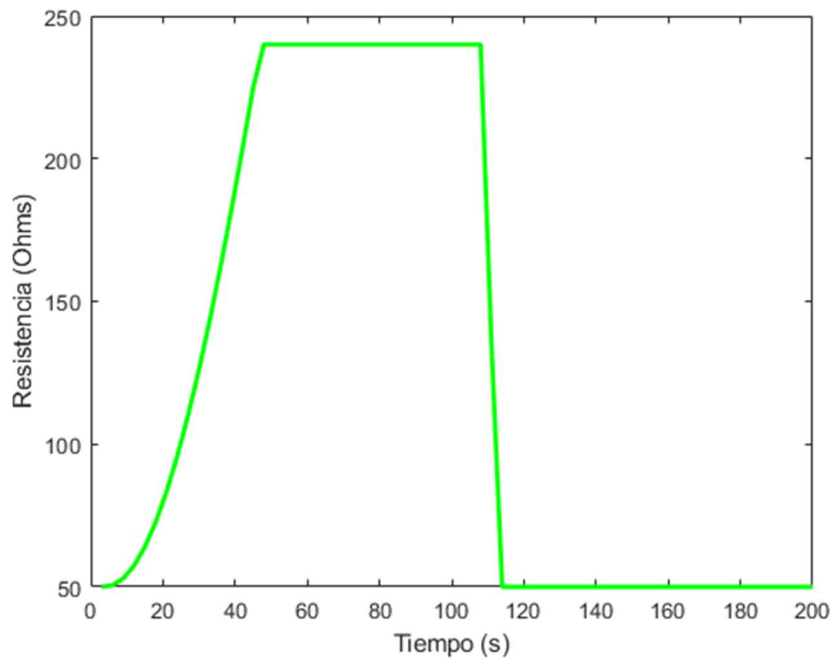


Figura 5.32 Variación de la resistencia del memristor $R_{mem} = 50\Omega\text{-}240\Omega$.

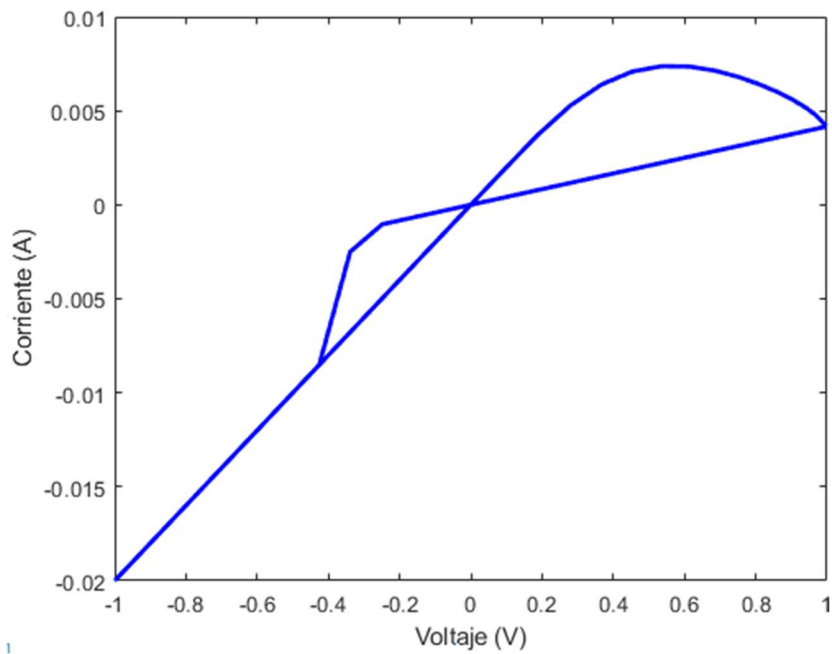


Figura 5.33 Curva de histéresis voltaje-corriente del memristor.

Tabla 5.8 Recursos lógicos utilizados del FPGA

	DSPs (240)	LUTs (63400)	Registers (126800)
Ecuación de estado	16	3802	64
Resistencia	3	2412	32
Función de ventana	14	2869	34
Control	0	120	0
Limitador	0	184	0
Integrador	3	552	32
Total	36	9939	162

Tabla 5.9 Parámetros usados para el modelo VTEAM

Parámetro	Valor	Comentario
dt	1×10^{-9} s	Diferencial de tiempo
R_{ON}	50 Ω	Memristancia en $x = 0$ m
R_{OFF}	240 Ω	Memristancia en $x = 3 \times 10^{-9}$ m
v_{off}	0.02V	Voltaje de umbral
v_{on}	-0.2V	Voltaje de umbral
w_c	107×10^{-12} m	Parámetro de ajuste
a_{on}	0 m	Parámetro de ajuste
a_{off}	3×10^{-9} m	Parámetro de ajuste
k_{on}	-14.5506 m/s	Parámetro de ajuste
k_{off}	0.0023 m/s	Parámetro de ajuste
α_{on}	1.03851	Parámetro de ajuste
α_{off}	1.0174	Parámetro de ajuste
x_{on}	0 m	Límite inferior de x
x_{off}	3×10^{-9} m	Límite superior de x
x_{init}	0 m	Condición inicial de la variable de estado

5.5 Conclusiones

En este capítulo se presentó una nueva metodología para el modelado y simulación del cuarto elemento fundamental de circuitos, el memristor, la cual se basó en el uso de las herramientas de desarrollo: Simulink, Simscape y Xilinx System Generator.

Los modelos matemáticos del memristor: lineal con función de ventana, TEAM (ThrEshold Adaptive Memristor Model) y VTEAM (Voltage ThrEshold Adaptive Memristor Model) se implementaron digitalmente en un dispositivo lógico programable (FPGA) utilizando bloques (IP) de la herramienta Xilinx System Generator. Se utilizó aritmética en punto flotante (32 bits) permitiendo obtener resultados similares al modelo numérico.

Debido a que Xilinx System Generator es un lenguaje gráfico de alto nivel el tiempo utilizado en la descripción de los modelos fue menor comparado con el tiempo empleado con algún lenguaje de programación VHDL o Verilog. Además, la comunicación entre el dispositivo FPGA y el entorno de Simulink es transparente para el usuario.

Para verificar el funcionamiento de los sistemas desarrollados se utilizó el método de la Co-simulación, donde las expresiones matemáticas de cada modelo se resolvieron a nivel de hardware (FPGA), mientras que la excitación y visualización de los resultados se realizó a nivel de software (MATLAB). El resistor variable de la librería de Simscape, permitió tener un elemento de tipo flotado para simulaciones eléctricas.

Referencias

- [1] K. Zaplatilek, Memristor modeling in Matlab&Simulink, Proceedings of European Computing Conference.
- [2] M.A. Gutiérrez Mondragón, Modelo no-lineal de memristores basado en red neuronal artificial, Tesis de Maestría en Ciencias, México: CINVESTAV, 2016.

- [3] G.M. Tornez Xavier, M.A. Gutiérrez Mondragón, L.M. Flores Nava, F. Gómez Castañeda, J.A. Moreno Cadenas, Memristor Hybrid Model for Nonlinear Analog Circuit Design 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control ([CCE](#)), Mexico City.
- [4] J.A. Rowson, Hardware/Software Co-Simulation, 1994 31th ACM/IEEE Design Automation Conference.
- [5] E. Todorovich, N. Acosta, Co-simulation de Hw/Sw usando FPGAs.
- [6] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, *Nature* 453 (2008) 80-83.
- [7] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, TEAM: threshold adaptive memristor model, *IEEE Trans. Circuit Syst.* 60 (1) (2013) 211-221.
- [8] S. Kvatinsky, M. Ramadan, E.G. Friedman, A. Kolodny, VTEAM: a general model for Voltage-controlled memristors, *IEEE Trans. Circuit Syst.* 62 (8) (2013) 786-790.
- [9] Xilinx Blockset Reference Guide.
- [10] S. Areibi, Tutorial-Using Xilinx System Generator 14.6 for Co-Simulation on Digilent NEXYS3 (Spartan-6) Board, (2017).
- [11] Y.N. Joglekar, S.J. Wolf, The elusive memristor: properties of basic electrical circuits, *Eur. J. Phys.* 30 (4) (2009) 661-675.
- [12] T. Prodromakis, B.P. Peh, C. Papavassiliou, C. Toumazou, A versatile memristor model with non-linear dopant kinetics, *IEEE Trans. Electron. Devices.* 58 (9) (2011) 3099-3105.
- [13] A.F. Adzmi, A. Nasrudin, W.F.H. Abdullah, S.H. Herman, Memristor Spice model for designing analog circuit, 2012 IEEE Student Conference on Research and Development.

Conclusiones generales

Se puede concluir este trabajo de investigación reconociendo que los dos objetivos planteados originalmente fueron alcanzados, a saber:

1. Se ha creado una componente digital de desarrollo de sistemas memristivos fundamentales, para ser usada como vehículo experimental alternativo a los existentes por medios tecnológicos tradicionales.

Esta componente digital ha mostrado experimentalmente un esquema de desarrollo de sistemas memristivos usando tecnología de sistemas digitales en la forma de componentes FPGA, que incluye plataformas numéricas de análisis de comportamiento (MATLAB/Simulink) y co-simulación para la definición de núcleos (cores): en este caso, un memristor numérico o “virtual”.

2. Se usaron redes neuronales memristivas en la solución de problemas de optimización.

Estas redes neuronales han mostrado el potencial innovador y emergente que el memristor puede tener como nuevo componente eléctrico en el diseño de circuitos CMOS analógicos y digitales.

Trabajo futuro

En este trabajo de tesis se muestra que, a través de los resultados satisfactorios obtenidos, es posible continuar con esta línea de investigación. Se proponen los siguientes segmentos de investigación dentro del área de sistemas memristivos.

- Uso del dispositivo digital programable para desarrollar sistemas eléctricos con comportamiento caótico.
- Caracterización de parámetros típicos de memristores particulares (corriente y voltaje de umbral, Resistencias ON y OFF, entre otros).
- Desarrollo de programador de memristores para integración en circuitos CMOS y, su evaluación por co-simulación.
- Desarrollo de algoritmos dependientes de tiempo para ajustar la resistencia de memristores que emulen la función sináptica de sistemas neuromórficos y, su evaluación por co-simulación.

Apéndice A

Se presenta una metodología que nos permite acelerar el tiempo de ajuste de parámetros del modelo VTEAM, el cual es un modelo general capaz de emular el comportamiento dinámico de memristores medidos experimentalmente y de otros modelos matemáticos del memristor. Para este caso se realizó el ajuste de parámetros del modelo VTEAM a la respuesta de los modelos TEAM y Yakopcic, los cuales se simularon experimentalmente en SPICE. En este apéndice se muestran los códigos utilizados en MATLAB, cabe enfatizar que estos fueron creados por el grupo de trabajo del profesor Kvatinsky y únicamente se acondicionaron para automatizar el proceso de ajuste. En la **Figura A.1** se muestra la metodología propuesta.

El bloque denominado optimizador se conforma por tres archivos .m: 1) Ajuste automatizado, 2) Error MSE y 3) Modelo VTEAM, los cuales trabajan en conjunto para ajustar los parámetros del modelo. Este bloque utiliza una base de datos compuesta por valores de voltaje, corriente y tiempo. Esta base de datos se puede obtener de un memristor medido experimentalmente o a partir de un modelo matemático que emule el comportamiento de un memristor físico.

El archivo **Ajuste automatizado** se encarga principalmente de ajustar los parámetros deseados del modelo, utilizando el algoritmo de optimización de recocido simulado (Simulated Annealing), el cual es un algoritmo de búsqueda metaheurística, donde el objetivo es encontrar el valor óptimo de una función objetivo en un espacio de búsqueda grande. Este algoritmo está inspirado en el proceso de recocido del acero y la cerámica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. Cabe mencionar que el algoritmo de recocido simulado no se describe en este archivo, sino que es invocado con el comando *simulannealbnd*, el cual es un archivo existente en las librerías de MATLAB. También en este archivo.m se definen los parámetros del modelo que se quieren ajustar.

El archivo **Error MSE** se encarga de calcular el error cuadrático medio por medio de la expresión (A1), la cual evalúa los valores de voltaje y corriente de referencia, ($v_{r,n}$, $i_{r,n}$ obtenidos del memristor medido experimentalmente o de un modelo), con los valores de voltaje y corriente, ($v_{m,n}$, $i_{m,n}$, calculados por el modelo VTEAM), N es el número total de muestras.

$$e_{v,i} = \sqrt{\frac{1}{N} \left(\frac{\sum_{n=1}^N (v_{m,n} - v_{r,n})^2}{\bar{v}_r^2} + \frac{\sum_{n=1}^N (i_{m,n} - i_{r,n})^2}{\bar{i}_r^2} \right)} \quad (A1)$$

El archivo **Modelo VTEAM** contiene la descripción del modelo y se utiliza para obtener el valor de la corriente que pasa a través del memristor al aplicar un voltaje de polarización. Estos valores de corriente y voltaje sirven para evaluar el error cuadrático medio y en base a qué tan grande o pequeño sea el error, modificar el valor de los parámetros a ajustar. Las expresiones (A1) – (A4) representan la ecuación de estado del modelo, mientras que la expresión (A5) es la relación voltaje – corriente del modelo. Para este caso los parámetros a ajustar son: k_{off} , k_{on} , α_{off} y α_{on} , mientras que los voltajes umbrales v_{off} , v_{on} y las resistencias R_{OFF} y R_{ON} son valores establecidos, los cuales pueden ser fácilmente obtenidos de la gráfica de histéresis del memristor.

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{v(t)}{v_{off}} - 1 \right)^{\alpha_{off}} f_{off}(x), & (A2) \\ 0, & (A3) \\ k_{on} \left(\frac{v(t)}{v_{on}} - 1 \right)^{\alpha_{on}} f_{on}(x), & (A4) \end{cases}$$

$$v(t) = \left[R_{ON} + \frac{R_{OFF} - R_{ON}}{x_{off} - x_{on}} (x - x_{on}) \right] i(t) \quad (A5)$$

El proceso de automatización es el siguiente: Inicialmente se declara el número máximo de corridas (*corrida_max*) que es el número de veces que se va a ejecutar el optimizador. Este optimizador utiliza los tres archivos descritos anteriormente para obtener un valor MSE, el cual es almacenado en memoria. Una vez que el número de corrida es igual a *corrida_max* del conjunto de valores MSE se selecciona el menor, para de esta manera obtener los valores finales

k_{off} , k_{on} , α_{off} y α_{on} y así graficar la curva de histéresis del modelo VTEAM con la del modelo TEAM.

El proceso de automatización se realizó principalmente debido a que al utilizar los códigos del grupo de trabajo del profesor Kvatinsky, el ciclo de optimización se ejecuta solo una vez, lo que ocasiona que el tiempo de ajuste sea extremadamente tardado. Por lo que al establecer un número de corrida máximo y utilizar la opción de simulación en paralelo de MATLAB con la instrucción **par for**, con la cual se utilizan de forma paralela todos los núcleos de la PC, por lo que el tiempo de ajuste de parámetros se acelera considerablemente.

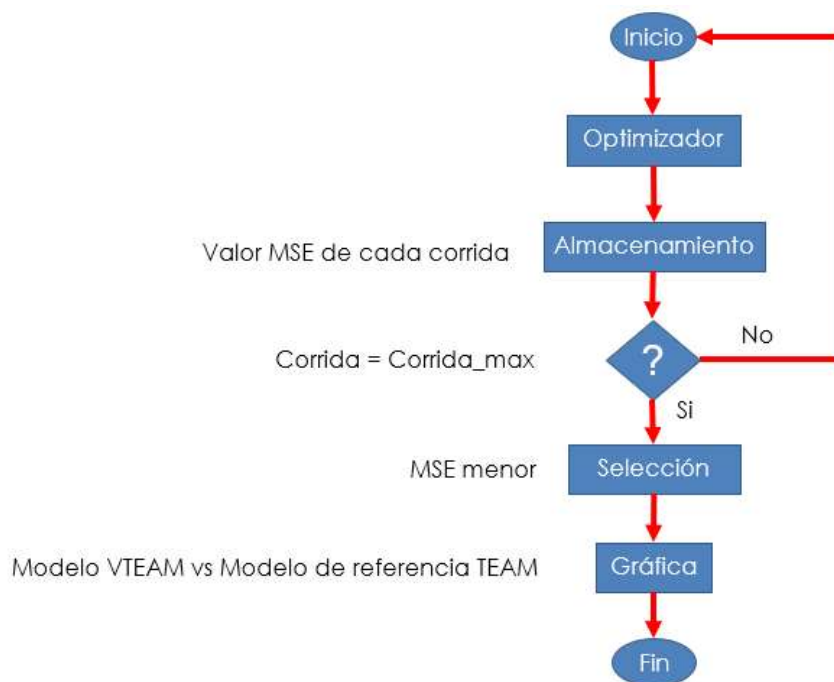


Figura A.1 Diagrama a bloques de la metodología propuesta.

En la **Tabla A1** se presentan en color rojo los valores finales de los parámetros ajustados para ambos modelos. Para el caso del modelo TEAM se obtuvo un error del 0.12% y para el modelo de Yakopcic un error del 0.25%, ambos errores son considerablemente pequeños y aceptables. Con estos resultados podemos verificar lo reportado en [29] capítulo 2, donde se establece que el modelo VTEAM es un modelo general capaz de aproximar la respuesta de memristores medidos experimentalmente, así como de otros modelos. Aunque nosotros observamos

que el procedimiento de ajuste puede ser tardado y complicado quedando por probar algún otro método de optimización. En las **Figuras A.2** y **A.3** se presentan las gráficas comparativas de los dos modelos TEAM y Yakopcic con el modelo VTEAM.

Tabla A1 Valores finales de los parámetros a ajustar

Parámetro	TEAM	Yakopcic
R_{ON}	50 Ω	387 Ω
R_{OFF}	240 Ω	1069.5 Ω
v_{off}	0.02V	0.16V
v_{on}	-0.2V	-0.05V
k_{on}	-14.5506 m/s	-2.9706e - 4 m/s
k_{off}	0.0023 m/s	5.8737e - 7 m/s
α_{on}	1.03851	4.9488
α_{off}	1.0174	3.2458
Error	0.12%	0.25%

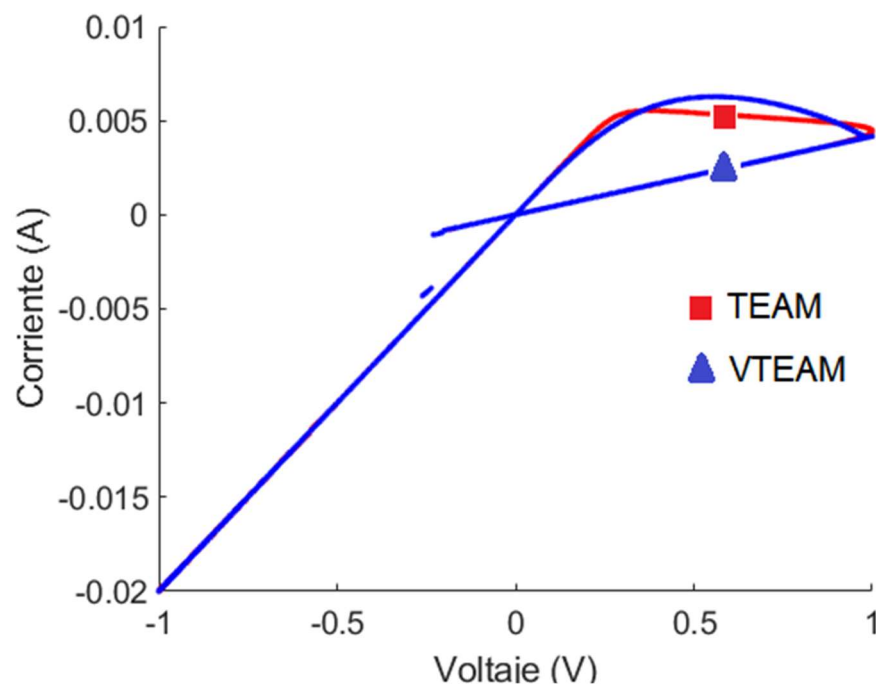


Figura A.2 Curva de histéresis del modelo VTEAM vs. modelo TEAM

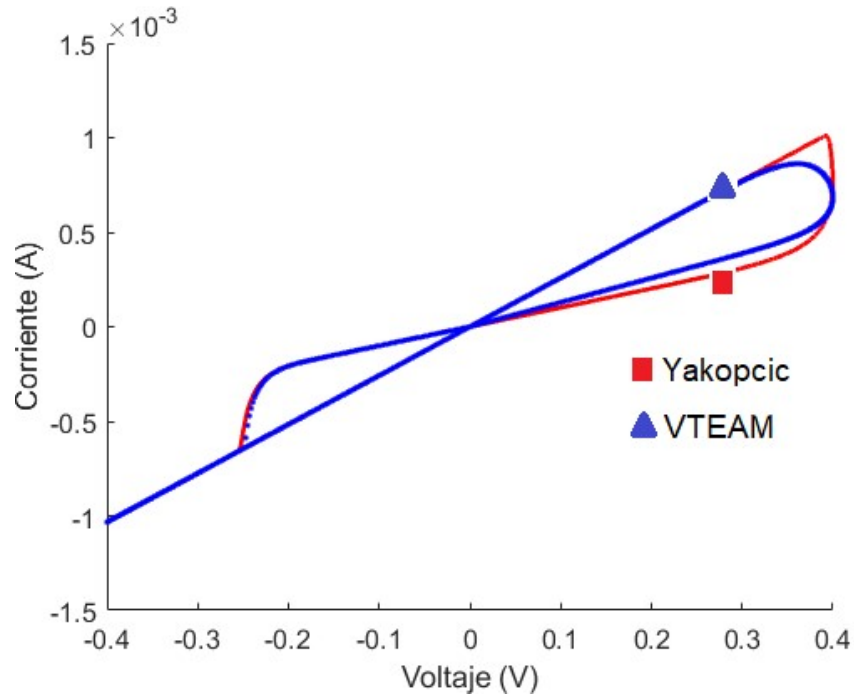


Figura A.3 Curva de histéresis del modelo VTEAM vs. modelo de Yakopcic

A continuación, se presentan los códigos en MATLAB para el ajuste de parámetros.

Ajuste automatizado.m

```
% Automatización del programa de Ajuste de parámetros del Modelo VTEAM

clear
clc
n = 50; % Numero de veces a ejecutar
load datos.dat;
v_ref = datos(:,2); % Insert the voltage vector of the reference
data.
i_ref = datos(:,3); % Insert the current vectore of the reference
data.
t_ref = datos(:,1); % Insert the time steps vector of the reference
data.
par_opt = zeros(n,4);

% Define the parameters of the model
D = 3e-9; % Device length
x_init = 0; % init value for state variable [0,D]
iv = 0; % I-V relationship, linear = 0, exponential=1.

% Define your input
v_input = v_ref;
t_vec = t_ref;
```

```

parfor i=1:n
input_parameters = [2,-2,2,2]; % valores iniciales de los parametros a
ajustar

lb = [0,-15,1,1]; % limites maximos
ub = [15,0,15,15]; % limites minimod

%-----
% Parametros de configuracion
options = optimoptions('simulannealbnd');
options = optimoptions(@simulannealbnd,'ReannealInterval',500);

fprintf('Corrida: %d ',i)
[optres1,fval,exitflag,output]=simulannealbnd(@mse,input_parameters,lb,
ub,options);

error(i)=fval;
par_opt(i,:)=optres1;

% Run the desired model in this case the team_op_model is used.
[v_modell,i_modell,~]=vteam_model(v_input,t_vec,x_init,iv,optres1);
i_modell = i_modell';

end
%-----
% Buscando la corrida con menor MSE
MejorInd = find(error==min(error));
MejorErr = error(MejorInd);
MejorPar = par_opt(MejorInd,:);
% Corriendo el Modelo TEAM con el mejor conjunto de parámetros
[v_modell,i_modell,~]=vteam_model(v_input,t_vec,x_init,iv,MejorPar);
i_modell = i_modell';
% Graficando en rojo el modelo ajustado y en azul los datos de
referencia
figure(1);
hold on;
plot(v_ref,i_ref,'.r');
plot(v_modell,i_modell,'.b');
title('Simulated Annealing');

```

Error mse.m

```

%% Here the fitting criterion is implemented.

function out = mse(input_parameters)
load datos.dat;
% voltage, current, and time vector of the reference data.

v_ref = datos(:,2); % Insert the voltage vector of the reference
data.
i_ref = datos(:,3); % Insert the current vectore of the reference
data.

```



```

t_ref = datos(:,1); % Insert the time steps vector of the reference
data.

% Define the parameters of the model.
D      = 3e-9;      % Device length.
x_init = 0;        % init value for state variable [0,D].
iv     = 0;        % I-V relationship, linear = 0, exponential=1.

% Define your input
v_input = v_ref;
t_vec   = t_ref;

N=length(t_vec);

% Run the desired model in this case the team_op_model is used.
[v_model,i_model,~]=vteam_model(v_input,t_vec,x_init,iv,input_parameter
s);
i_model = i_model';

% The output is the RMSE in this case. The algorithms will try to
minimize
% this output using a pointer the function mse using @mse in the run
script
out     = sqrt(( (sum((v_model-v_ref).^2)/sum(v_ref.^2)) +...
                (sum((i_model-i_ref).^2)/sum(i_ref.^2)) )/N);

end

```

Modelo VTEAM.m

```

function [v,i,x] =
vteam_model(v_input,t_vec,x_init,iv,input_parameters)

%Fill parameters here
%-----
Roff      = 240; %240
Ron       = 50; %50
voff     = 0.02; %Must be positive; %0.02
von      = -0.2; %Must be negative; %-0.2
koff     = input_parameters(1,1);
kon      = input_parameters(1,2); %-10
alphaoff  = input_parameters(1,3); %1
alphaon   = input_parameters(1,4); %3

D        = 3e-9;
xoff     = D;
xon      = 0;
aoff     = 3e-9;
aon      = 0;
wc       = 107e-12;
%-----

v=v_input;

```

```

lambda=reallog (Roff/Ron);

for j=1:length(t_vec)
    if j==1
        x(j)=x_init;
        if iv==0
            i(j)=v(j)/(Ron+((Roff-Ron)/(xoff-xon))*(x(j)-xon));
        elseif iv==1
            i(j)=v(j)/(Ron*exp(lambda*(x(j)-xon)/(xoff-xon)));
        end
    else
        dt=t_vec(j)-t_vec(j-1);
        if v(j)>=voff
            dxdt(j)=(koff*((v(j)/voff-1)^alphaoff));
            x(j)=x(j-1)+dxdt(j)*dt*exp(-exp((x(j-1)-aoff)/wc));
        elseif v(j)<=von
            dxdt(j)=(kon*((v(j)/von)-1)^alphaon);
            x(j)=x(j-1)+dxdt(j)*dt*exp(-exp(aon-(x(j-1))/wc));
        else
            dxdt(j)=0;
            x(j)=x(j-1);
        end

        if (x(j)<0)
            x(j)=0;
            dxdt(j)=0;
        elseif (x(j) > D)
            x(j)=D;
            dxdt(j)=0;
        end

        if iv==0
            i(j)=v(j)/(Ron+((Roff-Ron)/(xoff-xon))*(x(j)-xon));
        elseif iv==1
            i(j)=v(j)/(Ron*exp(lambda*(x(j)-xon)/(xoff-xon)));
        end
    end
end
end
end

```

Apéndice B

Como se explicó en el capítulo 3, el problema de asignación se puede definir matemáticamente por medio de las expresiones (3-1) – (3.4), las cuales definen la función objetivo y las restricciones del problema. A continuación, se presenta el desarrollo matemático para un sistema de $n \times n$ neuronas donde $n = 3$.

$$\text{minimizar } z = \sum_{i=1}^3 \sum_{j=1}^3 c_{i,j} v_{i,j} \quad (3.1)$$

$$\text{sujeto a } \sum_{j=1}^3 v_{i,j} = 1, \quad i = 1, \dots, 3 \quad (3.2)$$

$$\sum_{i=1}^3 v_{i,j} = 1, \quad j = 1, \dots, 3 \quad (3.3)$$

$$v_{i,j} \in \{0,1\}, \quad i, j = 1, \dots, 3 \quad (3.4)$$

Como se explicó el conjunto de expresiones (3-1) – (3.4), tiene su representación matricial utilizando (3.5) y (3.6). Al resolver (3.5) da como resultado el desarrollo de la función objetivo z , la cual es la sumatoria del producto del costo con la variable de decisión.

$$\text{minimizar } z = \mathbf{c}^T \mathbf{v} \quad (3.5)$$

$$\text{minimizar } z = [c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \quad c_{31} \quad c_{32} \quad c_{33}] \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix}$$

$$z = c_{11}v_{11} + c_{12}v_{12} + c_{13}v_{13} + c_{21}v_{21} + c_{22}v_{22} + c_{23}v_{23} + c_{31}v_{31} + c_{32}v_{32} + c_{33}v_{33}$$

Resolviendo (3.6) utilizando la definición de (3.7) se obtienen las seis restricciones del problema.

$$\text{sujeto a } \mathbf{Av} = \mathbf{b} \quad (3.6)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{Av} = \mathbf{b}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$v_{11} + v_{21} + v_{31} = 1$$

$$v_{12} + v_{22} + v_{32} = 1$$

$$v_{13} + v_{23} + v_{33} = 1$$

$$v_{11} + v_{12} + v_{13} = 1$$

$$v_{21} + v_{22} + v_{23} = 1$$

$$v_{31} + v_{32} + v_{33} = 1$$

Estas seis restricciones indican que la sumatoria de las variables de decisión siempre debe ser uno, esto quiere decir que por fila y columna solamente la salida de una neurona debe permanecer en estado alto '1', mientras que las salidas restantes de las demás neuronas de esa misma fila y columna en estado bajo '0'.

Al obtener las expresiones de la función objetivo y las restricciones del problema el siguiente paso es realizar el desarrollo matemático de la ecuación diferencial que define a la red neuronal recurrente de Wang (3.8). Como ya se explicó en el capítulo 3 la expresión (3.8) tiene su representación matricial con (3.11).

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\mathbf{v} - \boldsymbol{\theta}) - \lambda c e^{-\frac{t}{\tau}} \quad (3.11)$$

Utilizando la definición de la matriz de pesos (3.13) y la definición del vector umbral $\boldsymbol{\theta} = \mathbf{A}^T \mathbf{b}$ se tiene que

$$\mathbf{W} = \mathbf{A}^T \mathbf{A}$$

$$\mathbf{W} = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

Reemplazando en (3.11) la matriz \mathbf{W} , el vector $\boldsymbol{\theta}$ y las definiciones de los vectores \mathbf{v} y \mathbf{c} expresadas en (3.5) y (3.6), respectivamente, se obtiene:

$$\frac{du(t)}{dt} = -\eta \begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix} \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} - \lambda \begin{bmatrix} c_{11} e^{-\frac{t}{\tau}} \\ c_{12} e^{-\frac{t}{\tau}} \\ c_{13} e^{-\frac{t}{\tau}} \\ c_{21} e^{-\frac{t}{\tau}} \\ c_{22} e^{-\frac{t}{\tau}} \\ c_{23} e^{-\frac{t}{\tau}} \\ c_{31} e^{-\frac{t}{\tau}} \\ c_{32} e^{-\frac{t}{\tau}} \\ c_{33} e^{-\frac{t}{\tau}} \end{bmatrix}$$

$$\frac{du_{11}}{dt} = -2\eta v_{11} - \eta(v_{12} + v_{13}) - \eta(v_{21} + v_{31}) + 2\eta - \lambda c_{11} e^{-\frac{t}{\tau}}$$

$$\frac{du_{12}}{dt} = -2\eta v_{12} - \eta(v_{11} + v_{13}) - \eta(v_{22} + v_{32}) + 2\eta - \lambda c_{12} e^{-\frac{t}{\tau}}$$

$$\frac{du_{13}}{dt} = -2\eta v_{13} - \eta(v_{11} + v_{12}) - \eta(v_{23} + v_{33}) + 2\eta - \lambda c_{13} e^{-\frac{t}{\tau}}$$

$$\frac{du_{21}}{dt} = -2\eta v_{21} - \eta(v_{22} + v_{23}) - \eta(v_{11} + v_{31}) + 2\eta - \lambda c_{21} e^{-\frac{t}{\tau}}$$

$$\frac{du_{22}}{dt} = -2\eta v_{22} - \eta(v_{21} + v_{23}) - \eta(v_{12} + v_{32}) + 2\eta - \lambda c_{22} e^{-\frac{t}{\tau}}$$

$$\frac{du_{23}}{dt} = -2\eta v_{23} - \eta(v_{21} + v_{22}) - \eta(v_{13} + v_{33}) + 2\eta - \lambda c_{23} e^{-\frac{t}{\tau}}$$

$$\frac{du_{31}}{dt} = -2\eta v_{31} - \eta(v_{32} + v_{33}) - \eta(v_{11} + v_{21}) + 2\eta - \lambda c_{31} e^{-\frac{t}{\tau}}$$

$$\frac{du_{32}}{dt} = -2\eta v_{32} - \eta(v_{31} + v_{33}) - \eta(v_{12} + v_{22}) + 2\eta - \lambda c_{32} e^{-\frac{t}{\tau}}$$

$$\frac{du_{33}}{dt} = -2\eta v_{33} - \eta(v_{31} + v_{32}) - \eta(v_{13} + v_{23}) + 2\eta - \lambda c_{33} e^{-\frac{t}{\tau}}$$

Estas ecuaciones diferenciales representan la dinámica de cada una de las neuronas $N_{i,j}$ que conforman la red neuronal de Wang. Para el caso de la neurona $N_{1,1}$, cuya dinámica se representa por la ecuación diferencial du_{11}/dt el termino

$\eta(v_{12} + v_{13})$ representa la conexión lateral con las neuronas de su misma fila, $\eta(v_{21} + v_{31})$ la conexión lateral con las neuronas de su misma columna, $2\eta v_{11}$ la conexión con ella misma, 2η es el umbral de polarización y $\lambda c_{11} e^{-\frac{t}{\tau}}$ es el factor de costo, donde el parámetro η , peso sináptico, tiene un valor de 1 para todos los casos. La representación con bloques de Simulink de la expresión de la neurona $N_{1,1}$ se muestra en la **Figura B.1**.

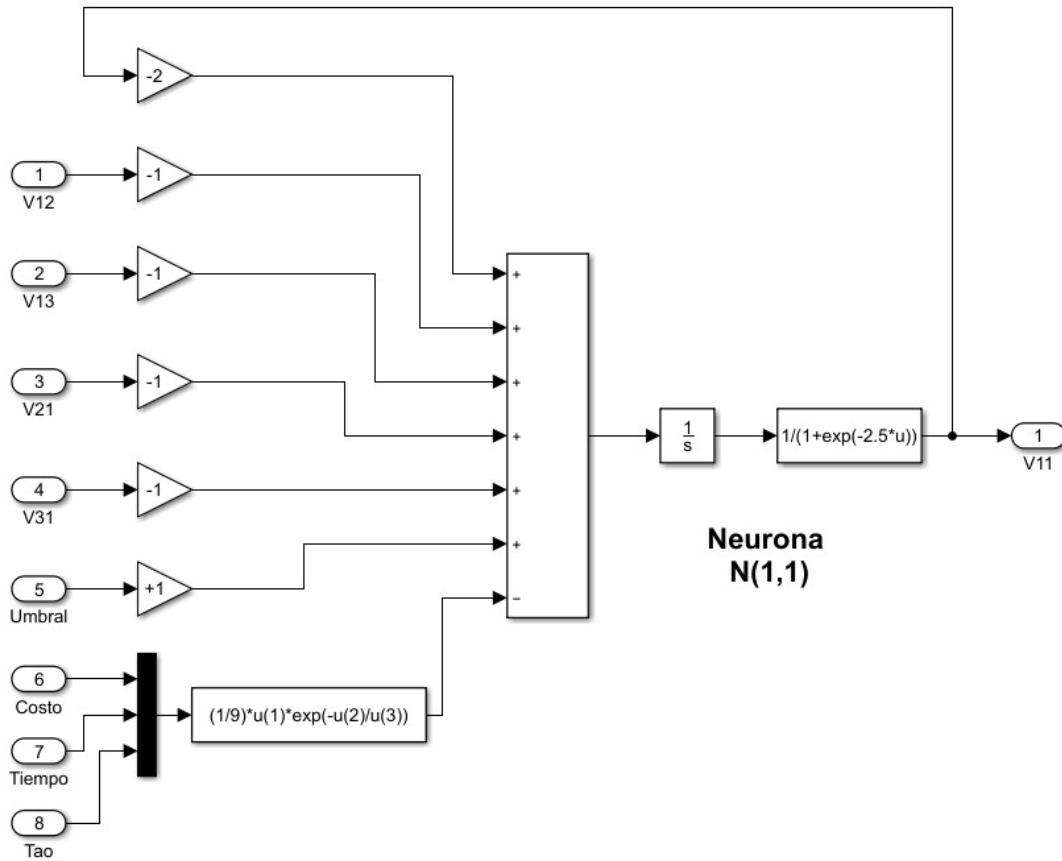


Figura B.1 Diagrama a bloques de la neurona $N_{1,1}$

A partir de los términos de las conexiones laterales y la auto conexión de todas las neuronas se realiza la interconexión de todas las neuronas obteniendo de esta manera la configuración de la red neuronal (**ver la Figura 3.1**). En la **Figura B.2** se muestra el diagrama completo con todas las neuronas para el sistema 3x3 propuesto.

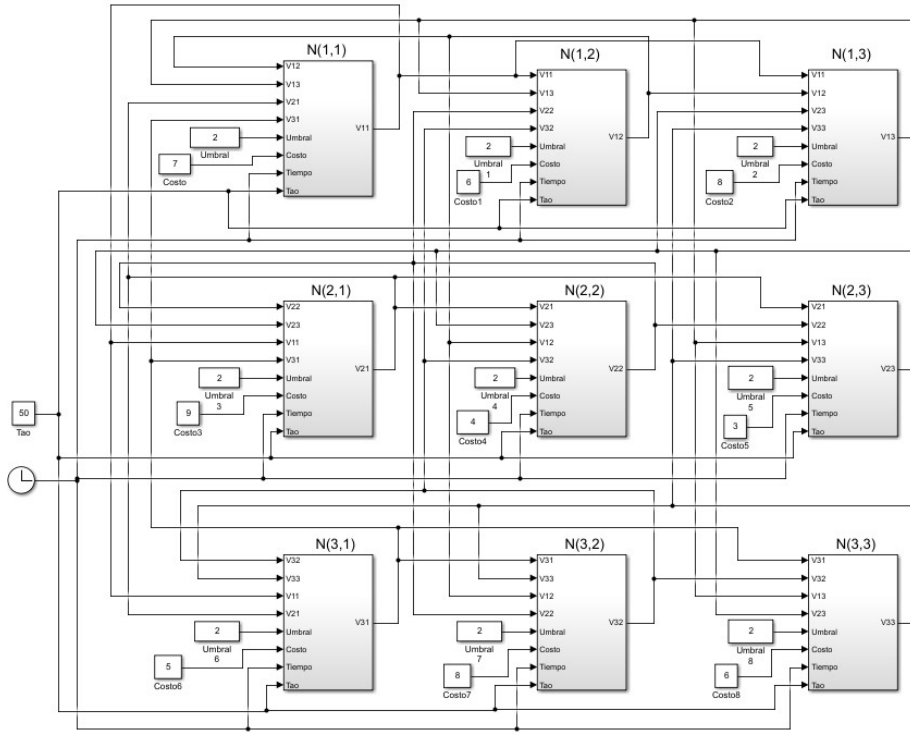


Figura B.2 Diagrama completo de la red neuronal

Las **Figuras B.3, 4 y 5** muestran la respuesta numérica $v_{i,j}$ de cada neurona $N_{i,j}$ para las redes Netwk1, Netwk2 y Netwk3, utilizando Simulink.

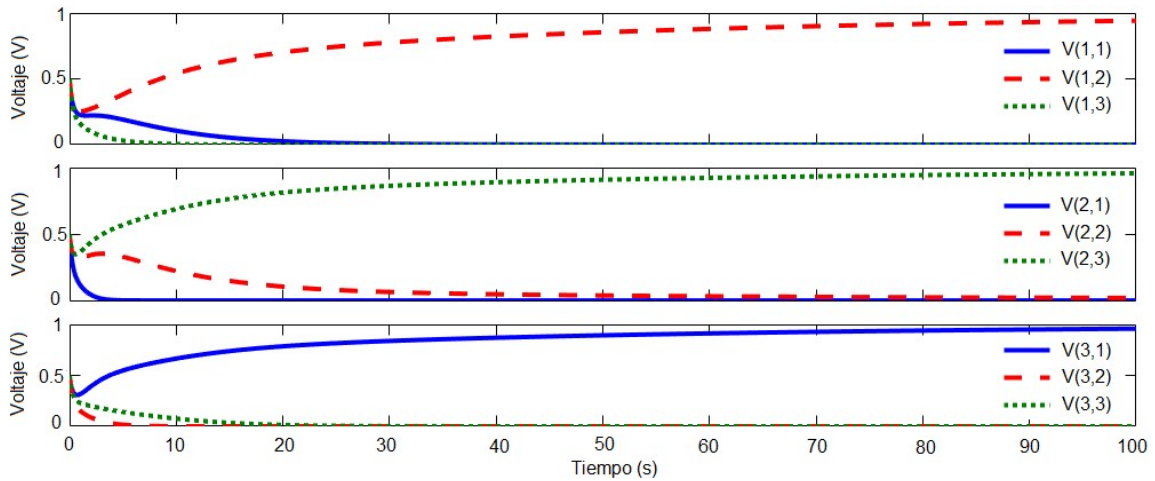


Figura B.3 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk1.

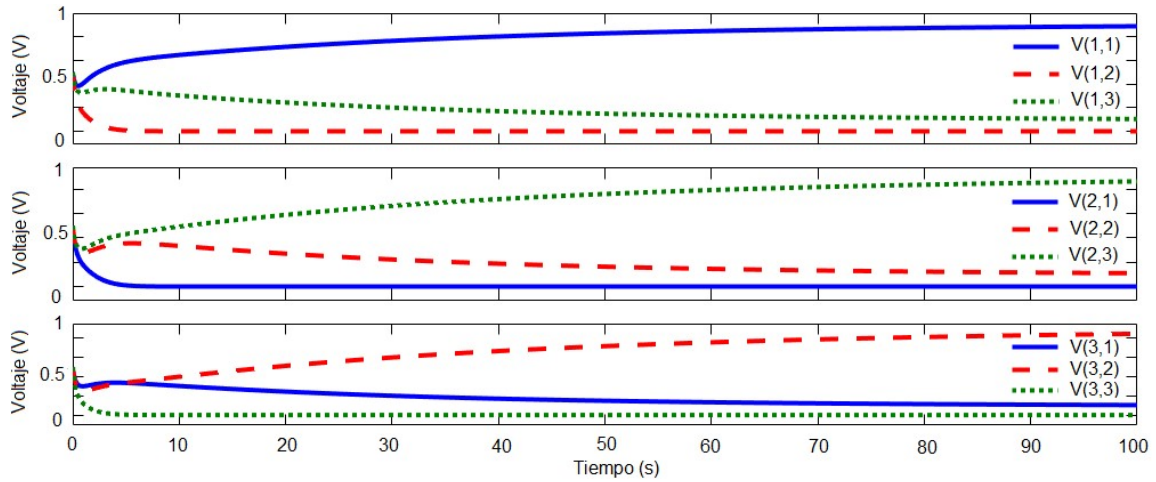


Figura B.4 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk2.

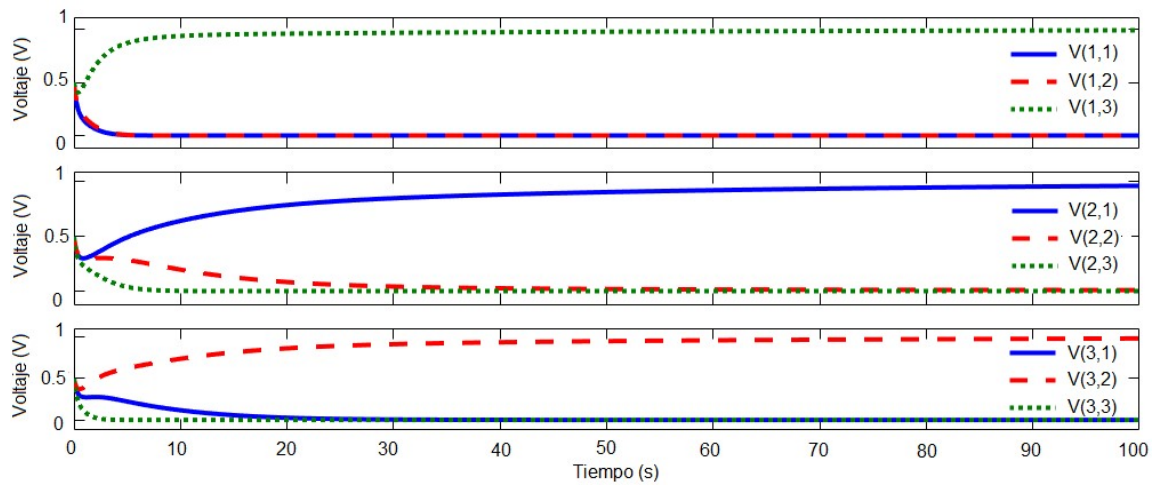


Figura B.5 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk3.

Apéndice C

Como se explicó en el capítulo 3, el problema de transporte se puede definir matemáticamente por medio de las expresiones (3-14) – (3.17), las cuales definen la función objetivo y las restricciones del problema. Estas expresiones al ser normalizadas se convierten en (3.18) – (3.21). A continuación, se presenta el desarrollo matemático para un sistema de $n \times n$ neuronas donde $n = 3$.

$$\text{minimizar } \bar{z} = \sum_{i=1}^3 \sum_{j=1}^3 c_{i,j} \bar{v}_{i,j} \quad (3.18)$$

$$\text{sujeto a } \sum_{j=1}^3 \bar{v}_{i,j} = \bar{a}_i, \quad i = 1, \dots, 3 \quad (3.19)$$

$$\sum_{i=1}^3 \bar{v}_{i,j} = \bar{b}_j, \quad j = 1, \dots, 3 \quad (3.20)$$

$$\bar{v}_{i,j} \in \{0,1\}, \quad i, j = 1, \dots, 3 \quad (3.21)$$

Como se explicó, el conjunto de expresiones (3-18) – (3.21), tiene su representación matricial utilizando (3.22) y (3.23). Al resolver (3.22) da como resultado el desarrollo de la función objetivo z .

$$\text{minimizar } \bar{z} = \mathbf{c}^T \bar{\mathbf{v}} \quad (3.22)$$

$$\text{minimizar } \bar{z} = [c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \quad c_{31} \quad c_{32} \quad c_{33}] \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix}$$

$$\bar{z} = c_{11}\bar{v}_{11} + c_{12}\bar{v}_{12} + c_{13}\bar{v}_{13} + c_{21}\bar{v}_{21} + c_{22}\bar{v}_{22} + c_{23}\bar{v}_{23} + c_{31}\bar{v}_{31} + c_{32}\bar{v}_{32} + c_{33}\bar{v}_{33}$$

Resolviendo (3.23) y utilizando la definición de (3.24) se obtienen las seis restricciones del problema.

$$\text{sujeto a } \mathbf{A}\bar{\mathbf{v}} = \mathbf{d} \quad (3.23)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \\ \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \bar{\mathbf{a}} \\ \bar{\mathbf{b}} \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}\bar{\mathbf{v}} = \mathbf{d}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\bar{v}_{11} + \bar{v}_{12} + \bar{v}_{13} = \bar{a}_1$$

$$\bar{v}_{21} + \bar{v}_{22} + \bar{v}_{23} = \bar{a}_2$$

$$\bar{v}_{31} + \bar{v}_{32} + \bar{v}_{33} = \bar{a}_3$$

$$\bar{v}_{11} + \bar{v}_{21} + \bar{v}_{31} = \bar{b}_1$$

$$\bar{v}_{12} + \bar{v}_{22} + \bar{v}_{32} = \bar{b}_2$$

$$\bar{v}_{13} + \bar{v}_{23} + \bar{v}_{33} = \bar{b}_3$$

Estas seis restricciones indican que la sumatoria de las variables de decisión de las salidas de las neuronas, ubicadas en la misma fila, debe ser igual a la oferta, mientras que la sumatoria de las variables de decisión de las neuronas ubicadas en la misma columna debe ser igual a la demanda.

Al obtener las expresiones de la función objetivo y las restricciones del problema el siguiente paso es realizar el desarrollo matemático de la ecuación diferencial que definen a la red neuronal recurrente de Wang (3.25). Como ya se explicó en el capítulo 3 la expresión (3.25) tiene su representación matricial con (3.27).

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\bar{\mathbf{v}} - \boldsymbol{\theta}) - \lambda c e^{-\frac{t}{\tau}} \quad (3.27)$$

Utilizando la definición de la matriz de pesos (3.29) y la definición del vector umbral $\boldsymbol{\theta} = \mathbf{A}^T \mathbf{d}$ se tiene que

$$\mathbf{W} = \mathbf{A}^T \mathbf{A}$$

$$\mathbf{W} = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{a}_1 + \bar{b}_1 & \bar{a}_1 + \bar{b}_2 & \bar{a}_1 + \bar{b}_3 \\ \bar{a}_2 + \bar{b}_1 & \bar{a}_2 + \bar{b}_2 & \bar{a}_2 + \bar{b}_3 \\ \bar{a}_3 + \bar{b}_1 & \bar{a}_3 + \bar{b}_2 & \bar{a}_3 + \bar{b}_3 \end{bmatrix}$$

Reemplazando en (3.27) la matriz \mathbf{W} , el vector $\boldsymbol{\theta}$ y las definiciones de los vectores $\bar{\mathbf{v}}$ y \mathbf{c} expresadas en (3.22) y (3.23), respectivamente se obtiene:

$$\frac{du(t)}{dt} = -\eta \begin{pmatrix} \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix} - \begin{bmatrix} \bar{a}_1 + \bar{b}_1 \\ \bar{a}_1 + \bar{b}_2 \\ \bar{a}_1 + \bar{b}_3 \\ \bar{a}_2 + \bar{b}_1 \\ \bar{a}_2 + \bar{b}_2 \\ \bar{a}_2 + \bar{b}_3 \\ \bar{a}_3 + \bar{b}_1 \\ \bar{a}_3 + \bar{b}_2 \\ \bar{a}_3 + \bar{b}_3 \end{bmatrix} \end{pmatrix} - \lambda \begin{bmatrix} c_{11}e^{-\frac{t}{\tau}} \\ c_{12}e^{-\frac{t}{\tau}} \\ c_{13}e^{-\frac{t}{\tau}} \\ c_{21}e^{-\frac{t}{\tau}} \\ c_{22}e^{-\frac{t}{\tau}} \\ c_{23}e^{-\frac{t}{\tau}} \\ c_{31}e^{-\frac{t}{\tau}} \\ c_{32}e^{-\frac{t}{\tau}} \\ c_{33}e^{-\frac{t}{\tau}} \end{bmatrix}$$

$$\frac{du_{11}}{dt} = -2\eta\bar{v}_{11} - \eta(\bar{v}_{12} + \bar{v}_{13}) - \eta(\bar{v}_{21} + \bar{v}_{31}) + \eta(\bar{a}_1 + \bar{b}_1) - \lambda c_{11}e^{-\frac{t}{\tau}}$$

$$\frac{du_{12}}{dt} = -2\eta\bar{v}_{12} - \eta(\bar{v}_{11} + \bar{v}_{13}) - \eta(\bar{v}_{22} + \bar{v}_{32}) + \eta(\bar{a}_1 + \bar{b}_2) - \lambda c_{12}e^{-\frac{t}{\tau}}$$

$$\frac{du_{13}}{dt} = -2\eta\bar{v}_{13} - \eta(\bar{v}_{11} + \bar{v}_{12}) - \eta(\bar{v}_{23} + \bar{v}_{33}) + \eta(\bar{a}_1 + \bar{b}_3) - \lambda c_{13}e^{-\frac{t}{\tau}}$$

$$\frac{du_{21}}{dt} = -2\eta\bar{v}_{21} - \eta(\bar{v}_{22} + \bar{v}_{23}) - \eta(\bar{v}_{11} + \bar{v}_{31}) + \eta(\bar{a}_2 + \bar{b}_1) - \lambda c_{21}e^{-\frac{t}{\tau}}$$

$$\frac{du_{22}}{dt} = -2\eta\bar{v}_{22} - \eta(\bar{v}_{21} + \bar{v}_{23}) - \eta(\bar{v}_{12} + \bar{v}_{32}) + \eta(\bar{a}_2 + \bar{b}_2) - \lambda c_{22}e^{-\frac{t}{\tau}}$$

$$\frac{du_{23}}{dt} = -2\eta\bar{v}_{23} - \eta(\bar{v}_{21} + \bar{v}_{22}) - \eta(\bar{v}_{13} + \bar{v}_{33}) + \eta(\bar{a}_2 + \bar{b}_3) - \lambda c_{23}e^{-\frac{t}{\tau}}$$

$$\frac{du_{31}}{dt} = -2\eta\bar{v}_{31} - \eta(\bar{v}_{32} + \bar{v}_{33}) - \eta(\bar{v}_{11} + \bar{v}_{21}) + \eta(\bar{a}_3 + \bar{b}_1) - \lambda c_{31}e^{-\frac{t}{\tau}}$$

$$\frac{du_{32}}{dt} = -2\eta\bar{v}_{32} - \eta(\bar{v}_{31} + \bar{v}_{33}) - \eta(\bar{v}_{12} + \bar{v}_{22}) + \eta(\bar{a}_3 + \bar{b}_2) - \lambda c_{32}e^{-\frac{t}{\tau}}$$

$$\frac{du_{33}}{dt} = -2\eta\bar{v}_{33} - \eta(\bar{v}_{31} + \bar{v}_{32}) - \eta(\bar{v}_{13} + \bar{v}_{23}) + \eta(\bar{a}_3 + \bar{b}_3) - \lambda c_{33}e^{-\frac{t}{\tau}}$$

Estas ecuaciones diferenciales representan la dinámica de cada una de las neuronas $N_{i,j}$ que conforman la red neuronal de Wang. Para el caso de la neurona $N_{1,1}$, cuya dinámica se representa por la ecuación diferencial du_{11}/dt el termino

$\eta(\bar{v}_{12} + \bar{v}_{13})$ representa la conexión lateral con las neuronas de su misma fila, $\eta(\bar{v}_{21} + \bar{v}_{31})$ la conexión lateral con las neuronas de su misma columna, $2\eta\bar{v}_{11}$ la conexión con ella misma, $\eta(\bar{a}_1 + \bar{b}_1)$ es el umbral de polarización y $\lambda_{c11}e^{-\frac{t}{\tau}}$ es el factor de costo, donde el parámetro η , el cual es el peso sináptico, tiene un valor de 1 para todos los casos. La representación con bloques de Simulink de la expresión de la neurona $N_{1,1}$ se muestra en la **Figura C.1**.

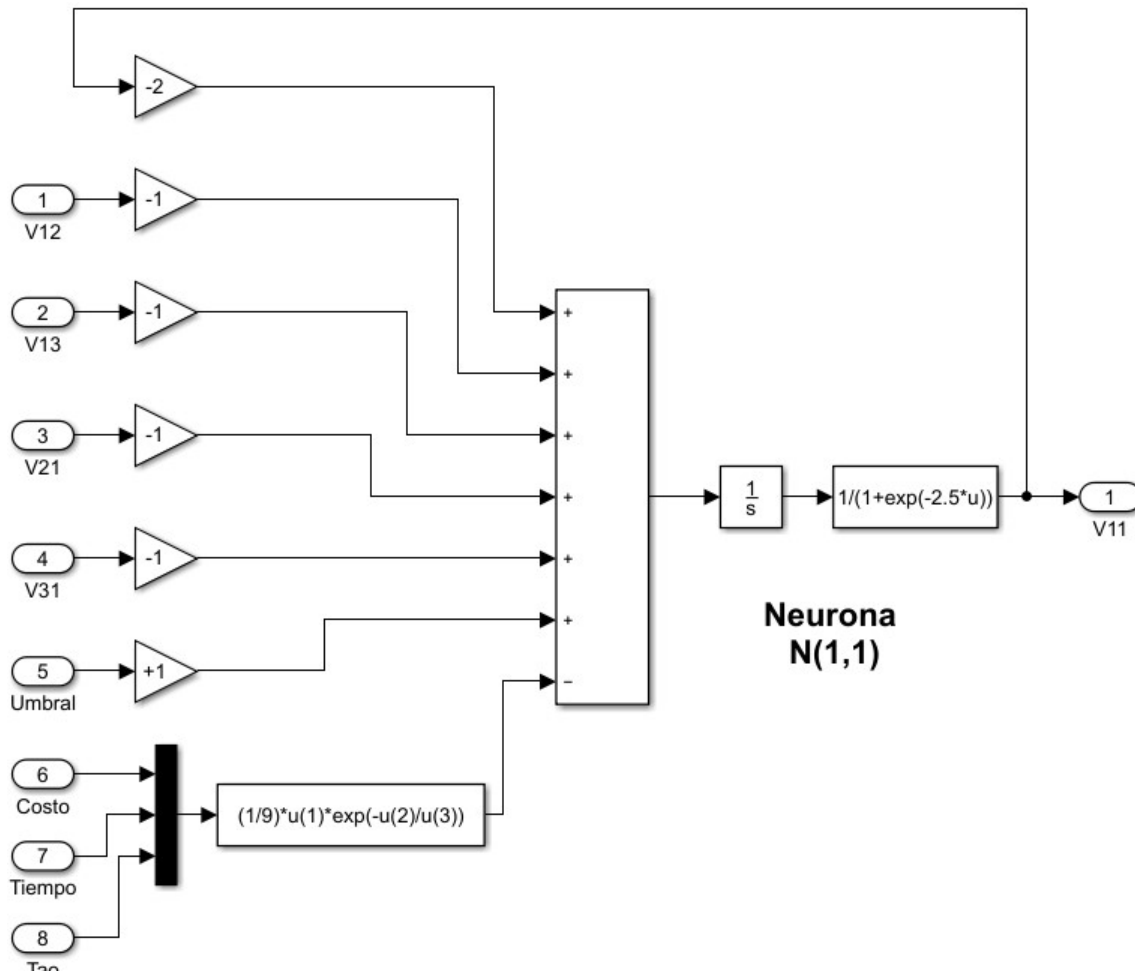


Figura C.1 Diagrama a bloques de la neurona $N_{1,1}$

A partir de los términos de las conexiones laterales y la auto conexión de todas las neuronas se realiza la interconexión de todas las neuronas obteniendo de esta manera la configuración de la red neuronal (**ver la Figura 3.1**). En la **Figura C.2** se muestra el diagrama completo con todas las neuronas para el sistema 3x3 propuesto.

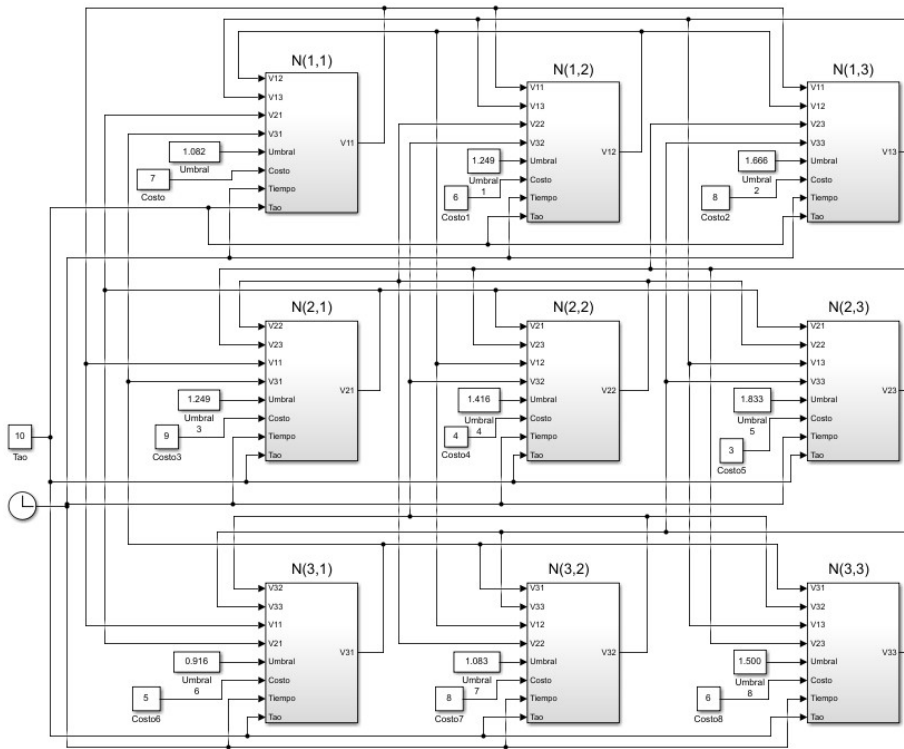


Figura C.2 Diagrama completo de la red neuronal

Las **Figuras C.3, 4 y 5** muestran la respuesta numérica $v_{i,j}$ de cada neurona $N_{i,j}$ para las redes Netwk4, Netwk5 y Netwk6, utilizando Simulink.

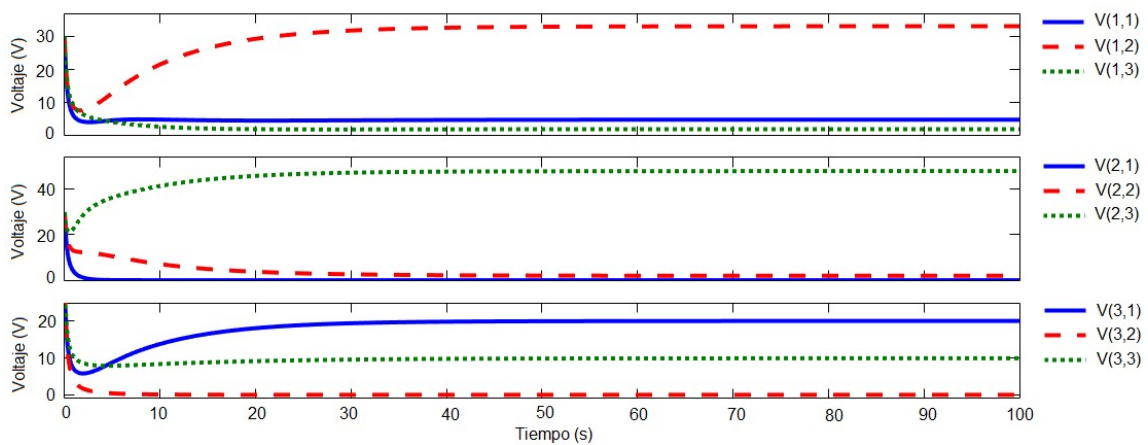


Figura C.3 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk4.

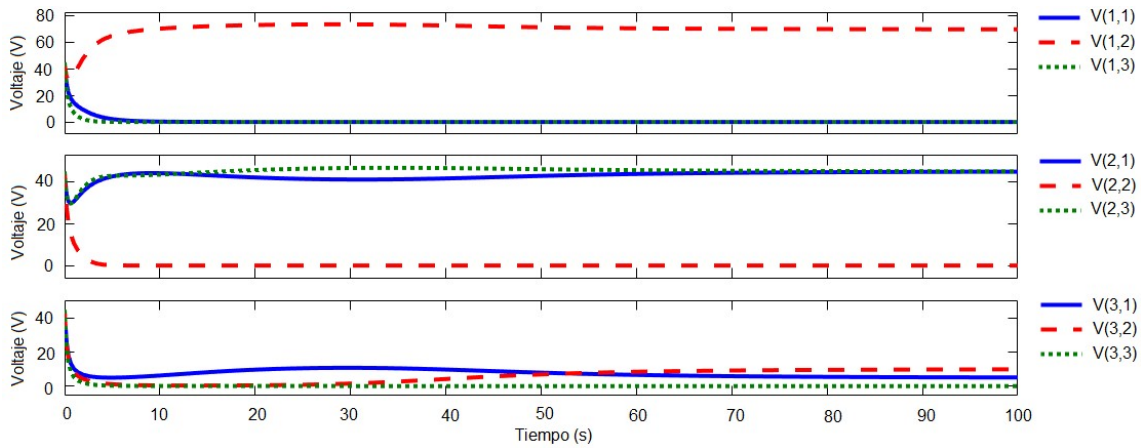


Figura C.4 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk5.

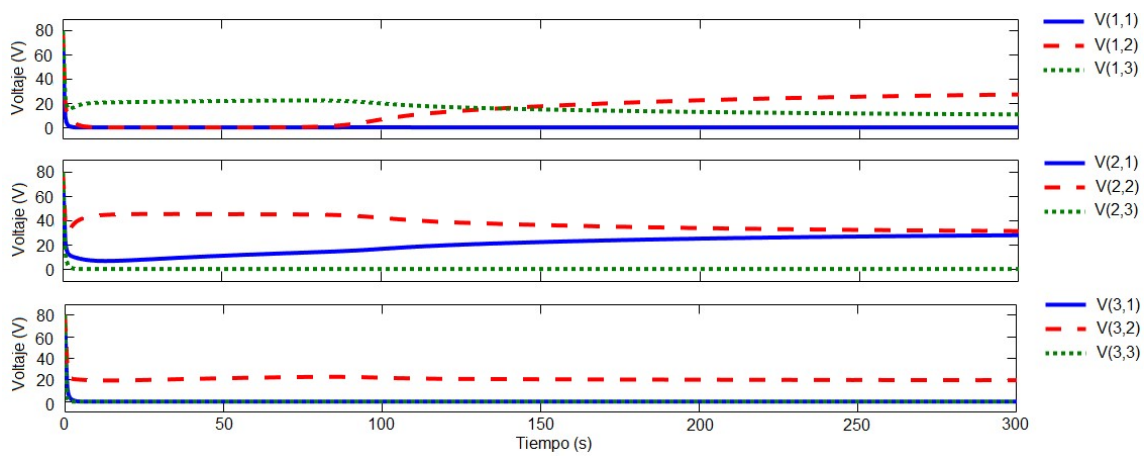


Figura C.5 Respuesta numérica $v_{i,j}$, en Simulink, de cada neurona $N_{i,j}$ para la red Netwk6.

Apéndice D

```
// Modelo TEAM para el memristor
`include "disciplines.vams"
`include "constants.h"
module ModeloTEAM(p,n,i);

parameter real dt=1e-7;
parameter real Ron=100e3;
parameter real Roff=1e6;
parameter real i_off=100e-15;
parameter real i_on=-100e-15;
parameter real x_c=107e-12;
parameter real a_on=0;
parameter real a_off=3e-9;
parameter real K_on=-5e-8;
parameter real K_off=5e-8;
parameter real Alpha_on=1;
parameter real Alpha_off=1;
parameter real IV_relation=1;
parameter real x_on=0;
parameter real x_off=3e-09;
parameter real D=3e-09;

// local variables
real x;
real dxdt;
real lambda;
real x_last;
real w_last;
real init_state;
real first_iteration;
real w_position;

input p; //positive pin
input i; //positive pin
output n; //negative pin
electrical p,n,i,gnd;
ground gnd;

analog begin

init_state = V(i,gnd);
```

```

if (first_iteration==0) begin
    w_last=init_state*D; // if this is the first iteration start with w_init
    x_last=init_state*D; // if this is the first iteration start with x_init
end

///// Evaluacion de dxdt //////////////////////////////////////
if (l(p,n) >= i_off) begin
    dxdt = K_off*pow((l(p,n)/i_off-1),Alpha_off);
end
if (l(p,n) <= i_on) begin
    dxdt = K_on*pow((l(p,n)/i_on-1),Alpha_on);
end
if ((i_on<l(p,n)) && (l(p,n)<i_off)) begin
    dxdt=0;
end

////////// Funcion de ventana de Kvatinsky //////////
if (l(p,n) >= 0) begin
    x = x_last+dt*dxdt*exp(-exp((x_last-a_off)/x_c));
end
if (l(p,n) < 0) begin
    x = x_last+dt*dxdt*exp(-exp((a_on-x_last)/x_c));
end

////////// Evaluacion de los limites //////////
if (x>=D) begin
    dxdt=0;
    x=D;
end
if (x<=0) begin
    dxdt=0;
    x=0;
end

lambda = ln(Roff/Ron);
x_last=x;
w_position = x/D;

if (IV_relation==1) begin
    V(p,n) <+ Ron*l(p,n)*exp(lambda*(x-x_on)/(x_off-x_on));
end
else if (IV_relation==0) begin
    V(p,n) <+ (Roff*x/D+Ron*(1-x/D))*l(p,n);
end
end
first_iteration=1;
end // end analog
endmodule

```

Apéndice E

Munkres.m

```
function [assignment,cost] = munkres(costMat,tip0)
assignment = zeros(1,size(costMat,1));
cost = 0;
validMat = costMat == costMat & costMat < Inf;
bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
costMat(~validMat) = bigM;
validCol = any(validMat,1);
validRow = any(validMat,2);
nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
    return
end
maxv=10*max(costMat(validMat));
dMat = zeros(n) + maxv;
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);
minR = min(dMat,[],2);
minC = min(bsxfun(@minus,dMat,minR));
zP = dMat == bsxfun(@plus,minC,minR);

starZ = zeros(n,1);
while any(zP(:))
    [r,c]=find(zP,1);
    starZ(r)=c;
    zP(r,:)=false;
    zP(:,c)=false;
end
while 1
    if all(starZ>0)
        break
    end
    coverColumn = false(1,n);
    coverColumn(starZ(starZ>0))=true;
    coverRow = false(n,1);
    primeZ = zeros(n,1);
    [rIdx, cIdx] =
find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR(~coverRow),minC(~c
overColumn)));
    while 1
        cR = find(~coverRow);
        cC = find(~coverColumn);
        rIdx = cR(rIdx);
        cIdx = cC(cIdx);
        Step = 6;
        while ~isempty(cIdx)
            uZr = rIdx(1);
            uZc = cIdx(1);
            primeZ(uZr) = uZc;
        end
    end
end
```

```

        stz = starZ(uZr);
        if ~stz
            Step = 5;
            break;
        end
        coverRow(uZr) = true;
        coverColumn(stz) = false;
        z = rIdx==uZr;
        rIdx(z) = [];
        cIdx(z) = [];
        cR = find(~coverRow);
        z = dMat(~coverRow,stz) == minR(~coverRow) + minC(stz);
        rIdx = [rIdx(:);cR(z)];
        cIdx = [cIdx(:);stz(ones(sum(z),1))];
    end
    if Step == 6

[minval,rIdx,cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~coverRow),minC(~coverColumn));
        minC(~coverColumn) = minC(~coverColumn) + minval;
        minR(coverRow) = minR(coverRow) - minval;
    else
        break
    end
end
rowZ1 = find(starZ==uZc);
starZ(uZr)=uZc;
while rowZ1>0
    starZ(rowZ1)=0;
    uZc = primeZ(rowZ1);
    uZr = rowZ1;
    rowZ1 = find(starZ==uZc);
    starZ(uZr)=uZc;
end
end
rowIdx = find(validRow);
colIdx = find(validCol);
starZ = starZ(1:nRows);
vIdx = starZ <= nCols;
assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
pass = assignment(assignment>0);
pass(~diag(validMat(assignment>0,pass))) = 0;
assignment(assignment>0) = pass;
cost = trace(costMat(assignment>0,assignment(assignment>0))) * tipo;

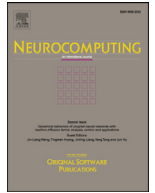
function [minval,rIdx,cIdx]=outerplus(M,x,y)
ny=size(M,2);
minval=inf;
for c=1:ny
    M(:,c)=M(:,c)-(x+y(c));
    minval = min(minval,min(M(:,c)));
end
[rIdx,cIdx]=find(M==minval);

```

Hungaro.m

```
clear all;
% tipo = -1 (Función max)
% tipo = 1 (Función min)
tipo = 1;
a=[0.25 0.56 0.82 0.29; 0.50 0.62 0.90 0.48; 0.70 0.48 0.20 0.50; 0.63
0.24 0.06 0.37]*tipo;
%a=[0.5 0.41 0.70 0.39; 0.40 0.08 0.55 0.35; 0.35 0.5 0.40 0.80; 0.20
0.27 0.76 0.61]*tipo;
%a=[0.55 0.37 0.15 0.80; 0.85 0.45 0.26 0.47; 0.67 0.75 0.53 0.07; 0.81
0.52 0.79 0.77]*tipo;
%a=[0.55 0.80 0.15 0.80; 0.85 0.45 0.80 0.47; 0.67 0.75 0.53 0.75; 0.81
0.52 0.79 0.77]*tipo;
[assignment,cost] = munkres(a,tipo);
```

ANEXOS



Memristive recurrent neural network



Gerardo Marcos Tornez Xavier, Felipe Gómez Castañeda*, Luis Martín Flores Nava,
José Antonio Moreno Cadenas

Center for Research and Advanced Studies of the IPN, Cinvestav-IPN, Electrical Engineering Department, Av. Instituto Politécnico Nacional 2508, CP 07360 Mexico City, Mexico

ARTICLE INFO

Article history:

Received 8 November 2016

Revised 5 August 2017

Accepted 7 August 2017

Available online 18 August 2017

Communicated by Duan Shukai

Keywords:

Memristor

Team model

Neural network

Hopfield

Continuous-time signal

Analog VLSI design

ABSTRACT

It is reported a continuous-time neural network in CMOS that uses memristors. These nanodevices are used to achieve some analog functions such as constant current sourcing, decaying term emulation, and resistive connection; all of them representing parameters of the neural network. The expected dynamics of this silicon circuit with these functional memristors is demonstrated via SPICE simulations based on 0.5 μm , n-well CMOS technology. The neural circuit is operative by finding the optimal solution of small-size combinatorial optimization problems, namely: “Assignment” and “Transportation”. It was chosen fast switching titanium dioxide memristors, which are modeled with nonlinear window functions and tunneling effect with the TEAM paradigm. This analog network belongs to an early recurrent model, which is electrically redesigned to take into account memristive arrays but keeping its original convergence properties. The behavioral and electrical analysis is done via Simulink-SPICE simulation. The outcome VLSI functional blocks combine both current and voltage to represent the variables in the recurrent model.

© 2017 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

The nonlinear and dynamic $i-v$ characteristics of the nanoscale memory component or memristor were reported by HP Labs [1]. This electrical component was described earlier in the theoretical work by Chua (see first reference in [1]). The memristor represents a modern focus of attention with various topic avenues: from proposing its technology with different materials [2,3] to introducing it in parallel models of computing approaches with bases in complex systems [4,5]. Furthermore, it is envisaged that the electronic design activity has potential possibilities of creating emerging analog and digital systems [6], where the memristor might be introduced as an innovative element. In the academic context, there are electronic designs oriented to demonstrate analog systems based on the Hopfield network configuration, where the traditional weighting resistive elements can be replaced by memristors. This replacement can be implemented in standard electronic technology. In this direction, works in [7,8] deal with two demonstrative prototypes: a 4-bit ADC converter and an associative memory. Taking in particular the Hopfield model as an optimizer, it can

be analyzed according to the manner its energy or cost function is stated. The efficiency to solve linear and nonlinear programming problems [9] is an alternate issue in the study of the Hopfield paradigm. Another recurrent neural network that works as an optimizer and becomes attractive for VLSI integration was presented by Wang [10]. This neural model operates with single-valued resistors with a lateral connectivity pattern among neurons.

In this paper, we solve two combinatorial optimization problems with the electrical recurrent network, working as an optimizer, that was used by Wang, where memristors are utilized to represent analog parameters, both static (permanent in time) and dynamic (decaying in time).

We show standard continuous-time CMOS circuits in analog neurons: unit-gain current amplifiers for computing integrals of aggregated currents and inverters for sigmoid functions. The connection between neurons is made with two-memristor arrays that observe the sense of the current. Simulations of the electrical and analytical response of the memristive network done both in Spice and Simulink, provide numerical reference comparisons; they were nearly similar.

The main topic in the electrical design in this work was concerned with the change of value of the resistance in working memristors, i.e. when a current flows through. The objective of using memristors was achieved in three manners: 1) Keeping the Off-state resistance, i.e. a constant-value reference current was implemented by setting a particular analog voltage at the memris-

* Corresponding author.

E-mail addresses: gtornez@cinvestav.mx (G.M. Tornez Xavier), fgomez@cinvestav.mx, felipegomezcastaneda@gmail.com (F. Gómez Castañeda), lmflores@cinvestav.mx (L.M. Flores Nava), jmoreno@cinvestav.mx (J.A. Moreno Cadenas).

tors terminals, 2) Combining both Off and On-state resistances in a group of memristors, i.e. a dynamic decaying current source with a predetermined peak value was obtained and, 3) Selecting dynamically one of two memristors, i.e. a constant-value bidirectional resistor was emulated.

2. The memristor and its model for circuit design

At present, the memristor is modeled with high-performance software, reproducing its nonlinear and frequency dependence when continuous-time signals are applied to its two terminals. The suitable electrical relation in circuit design is $i-v$; for the memristor, it comes from observing the Constitutive Relation [11]; it is reproduced in (1), where ϕ is the magnetic flux and q is the electrical charge.

$$f_M(\phi, q) = 0 \tag{1}$$

Using (1) the memristance or $R(q)$ is defined in (2). Therefore, the $i-v$ function of the memristor is defined in (3).

$$R(q) = \frac{df_M(q)}{dq} \tag{2}$$

$$v = R(q)i \tag{3}$$

The main feature that is drawn from (3) is shown when $i=0$ at time t_0 , then the memristor stores charge and memorizes the last memristance value, e.g. $R = R(q \text{ at } t_0)$. A generalization of (2) and (3) supports circuit-theoretic properties of the memristor, which are expressed by (4) and (5). Where, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ represents a state-variable vector.

$$v = R(\mathbf{x}, i) \tag{4}$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, i) \tag{5}$$

Both (4) and (5) have minimal representation by reducing \mathbf{x} to a scalar-variable x ; the common case is when x measures the relative position of a frontier between two adjacent and variable resistive regions in the fabricated nanomaterial. In principle, a low resistivity region that contains moving oxygen vacancies can displace into the other region of high resistivity due to the flow of current in the memristor. Whether w is the width of the region with oxygen vacancies and D is the width of the memristor, the state-variable is given by $x = w/D$.

Most memristor models approach the $i-v$ function observing a nonlinear ion drift behavior [12] and tunneling [13] as dominant conduction mechanisms. They might provide numerical means of the charge-kinetics identification in nanometric materials. However, computation effective features are preferred in memristive circuit design; the TEAM model itself is suitable [14]. In TEAM, x is introduced by modeling dx/dt . This expression is given in (6) replacing w/D by its complement i.e. $x = 1 - w/D$ or $x = d/D$. Where, $d = D - w$, is the effective width of the tunneling gap.

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{i(t)}{i_{off}} - 1 \right)^{\alpha_{off}} f_{off}(x), & 0 < i_{off} < i \\ 0, & i_{on} < i < i_{off} \\ k_{on} \left(\frac{i(t)}{i_{on}} - 1 \right)^{\alpha_{on}} f_{on}(x), & i < i_{on} < 0 \end{cases} \tag{6a}$$

In (6), k_{off} , k_{on} , α_{off} , α_{on} , i_{on} and i_{off} are parameters that fit the model with experimental data. Finally, the $i-v$ relation is nonlinear in x according to (7).

$$v(t) = R_{ON} e^{(\lambda/x_{off} - x_{on})(x - x_{on})} i(t) \tag{7}$$

R_{ON} and R_{OFF} are the resistances at $x = x_{on}$ and $x = x_{off}$, respectively, and $\lambda = \ln(R_{OFF}/R_{ON})$. The TEAM model was introduced in

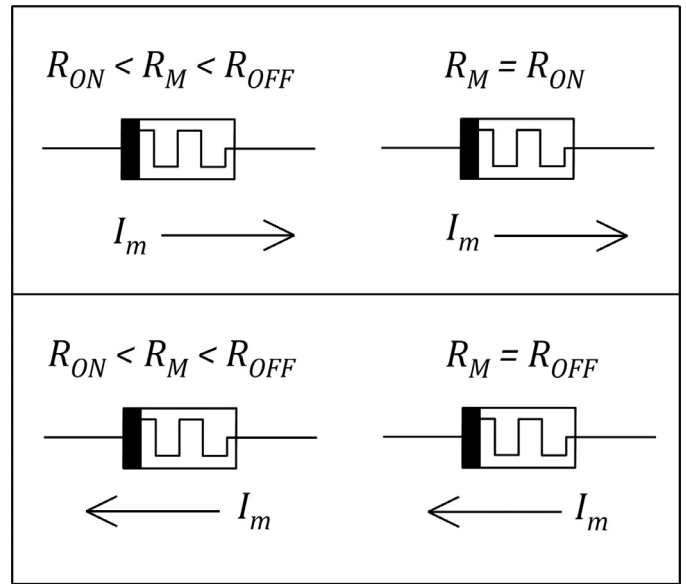


Fig. 1. Memristive states before (left, $R_{ON} < R_M < R_{OFF}$) and after (right, $R_M = R_{ON}$ or $R_M = R_{OFF}$) due to flow of current. This simple procedure sets the values R_{ON} or R_{OFF} in memristors used in the recursive neural network in this work.

the circuit design community and oriented to fast digital systems. These characteristics are useful in the recurrent neural network as can be appreciated further.

3. Memristance for design

The static and dynamic states of the memristor with fast switching features, support the neural network design in this work. The static states are either R_{ON} or R_{OFF} whereas the dynamic states are characterized by dx/dt as a changing resistance. R_{ON} and R_{OFF} are available when the current does not change the direction; therefore, the signal voltage polarity at the memristor terminals should be kept during the analog process. In Fig. 1, the initial resistive states are within the interval $R_{ON} < R_M < R_{OFF}$ (on the left) and the ending states after flowing the current I_m through the memristor (on the right); where R_M is the resistance value of the memristor. For the changing resistance case, when dx/dt is not zero, we take advantage proposing a parallel-series configuration that is suitable to emulate a transient current, whose initial or peak value I_{MS} can represent a parameter of the neural network namely, $\lambda c_i \exp(-t/\tau)$ which is introduced later. Fig. 2 presents a conceptual diagram of this configuration, where the analog differential potential $(V+) - (V-)$ starts to change from zero to V_M at time T_{START} . V_M is an analog voltage that should be found to reach the initial or peak current I_{MS} . Before this transient occurs, the series memristors are in the On-state resistance and the parallel ones in the Off-state resistance. The current through the series memristors causes a change in their individual resistance towards the value R_{OFF} , decreasing I_m down to I_{MF} . Although there are reported analytical expressions that would approximate the transient current I_m , see for example [15], we use the TEAM model to observe that I_m decays from I_{MS} toward I_{MF} . This transient characteristic replaces that one originally described by a discharging capacitor through a resistor in the electrical neural model [10]. Using Ohm's Law and considering that there are S_M series memristors and P_M parallel memristors in Fig. 2, the initial (or Peak) and final currents I_{MS} and I_{MF} are given by (8) and (9), respectively.

$$I_{MS} = \frac{V_M}{(R_{ON} S_M + \frac{R_{OFF}}{P_M})} \tag{8}$$

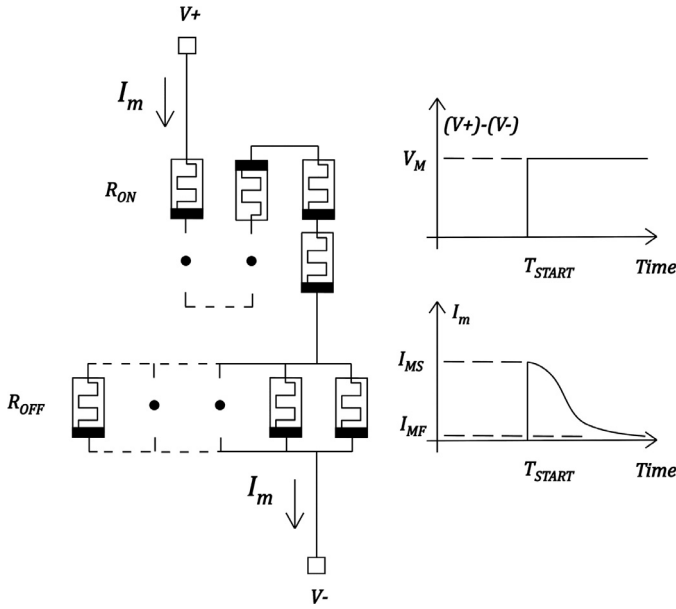


Fig. 2. Transient current in a memristive array. Using S_M series memristors connected with S_P parallel memristors, whose initial memristance is set at R_{ON} and R_{OFF} , respectively, can emulate a decaying current I_m when a voltage step is applied at time T_{START} . The peak current I_{MS} is established according to (8). The final current I_{MF} approaches the value in (9).

$$I_{MF} = \frac{V_M}{(R_{OFF}S_M + \frac{R_{OFF}}{I_M})} \quad (9)$$

Changing the number of series or parallel memristors and fixing the voltage difference $(V+)-(V-)$, we can find V_M that satisfies the condition $I_m = I_{MS}$. Finally, as a design constraint, we should observe that $(V+)-(V-)$ is limited to an interval.

4. Recurrent neural network

The supporting theory refers to an analog neural network model with n neurons, where the reference neuron N_i recurrently connects the other ones through weighting factors $W_{i,j}$ and whose dynamics ends up solving linear programming problems. The network configuration of this model is built with neurons arranged as a vector, where $W_{i,j}$ with $i, j = 1, 2, \dots, n$ can be represented by a symmetric matrix \mathbf{W} as shown in Fig. 3. There exists a self-connection, whose weighting factor is $W_{i,i}$. Using the intermediate variable z_i , the state variable u_i relates with v_j according to (10) and (11), with $i = 1, 2, \dots$. Therefore, neuron N_i integrates $du_i = z_i dt$, where u_i is the argument of a sigmoid function as denoted in (12) to provide v_i . β is a gain factor.

$$z_i = \frac{du_i}{dt} \quad (10)$$

$$z_i = v_1 W_{i,1} + v_2 W_{i,2} + \dots + v_j W_{i,j} + \dots + v_n W_{i,n} \quad (11)$$

$$v_i = \frac{1}{(1 + e^{-\beta u_i})} \quad (12)$$

In particular, for a linear programming problem, whose statement is

$$\text{Minimize: } f(\mathbf{v}) = \mathbf{c}^T \mathbf{v}$$

$$\text{Subject to: } \mathbf{A}\mathbf{v} = \mathbf{b}$$

The matrix \mathbf{W} in the recurrent neural network is defined by (13) [16], where η is a positive scalar parameter.

$$\mathbf{W} = -\eta \mathbf{A}^T \mathbf{A} \quad (13)$$

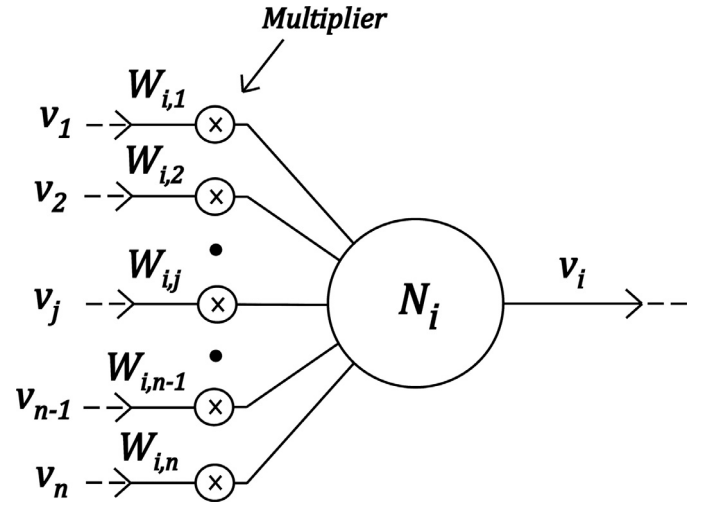


Fig. 3. Neuron N_i for vector array. The input analog signals: v_1, v_2, \dots, v_n are multiplied by weighting factors: $W_{i,1}, W_{i,2}, \dots, W_{i,n}$, where they are real numbers. Neuron N_i performs an addition, a temporal integration and a nonlinear transformation.

The entries of \mathbf{A} are the coefficients of the linear equations that establish the problem. Additionally, \mathbf{c} and \mathbf{b} are column vectors of parameters; \mathbf{v} is a column vector of the decision variable. The constraint equation might include slack variables where $\{\mathbf{b}, \mathbf{c} \text{ and } \mathbf{v}\} \in \mathbb{R}^n$.

In this case, u_i and v_i are the entries of the column vectors \mathbf{u} and \mathbf{v} respectively, and they are related according to the state Eq. (14). At the steady state, i.e. $du/dt = \mathbf{0}$, \mathbf{v} presents the solution.

$$\frac{d\mathbf{u}}{dt} = \mathbf{W}\mathbf{v} + \eta \mathbf{A}^T \mathbf{b} - \mathbf{c}\xi(0)e^{-\frac{t}{\tau}} \quad (14)$$

In (14), the column vector \mathbf{c} decays in time due to $\xi(0)\exp(-t/\tau)$; where, $\xi(0)$ and τ are the initial (or peak) value and the time constant, respectively.

Wang [16] demonstrated the stability and convergence to near optimal solutions of the recurrent neural network described by the set of equations in (14), which are suitable for circuit implementation in VLSI design. Wang also established a methodology of electrical networks with similar convergence properties in the case of combinatorial optimization problems [10], leading to a matrix of n by n neurons i.e. organized in n rows by n columns. The new network configuration uses neuron, as shown in Fig. 4, in a matrix array, where the factor Q is equal to -1 if $k = i$ or $l = j$, otherwise Q is zero. This factor establishes connections with neurons in the same column and row of the reference neuron $N_{i,j}$. There are only lateral inhibitory connections, all of them having the same value $-W_0$, also there exists a self-connection whose weighting factor is $-2W_0$.

The individual state equations that formulate the ‘‘Assignment’’ problem [17] are given by (15) and (16), where k and l are columns and rows counters, respectively.

$$\frac{du_{i,j}}{dt} = -\eta \sum_{k=1}^n v_{i,k} - \eta \sum_{l=1}^n v_{l,j} + 2\eta - \lambda c_{i,j} e^{-\frac{t}{\tau}} \quad (15)$$

$$v_{i,j} = \frac{1}{(1 + e^{-\beta u_{i,j}})} \quad (16)$$

In (15), the term $-\eta$ that multiplies the summations is the inhibitory weighting factor in the lateral connections; 2η represents a threshold; $c_{i,j}$ belongs to the set of costs in a problem, and decays in time due to $\exp(-t/\tau)$ and λ which is a constant. According to our experimental results, it is convenient to replace (16) by

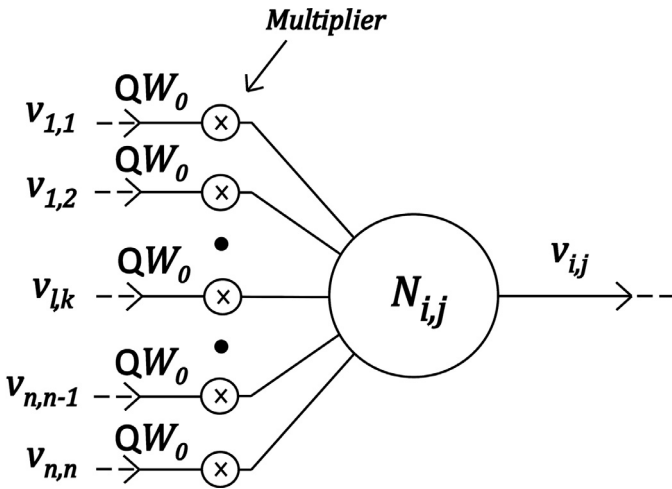


Fig. 4. Neuron N_{ij} for matrix array. The input analog signals: $v_{1,1}, v_{1,2}, \dots, v_{n,n}$ are multiplied by weighting factors: QW_0, QW_0, \dots, QW_0 , where, $Q \in \{0, -1\}$ obeys to a lateral connection rule and W_0 is a real positive number. Neuron N_{ij} performs an addition, a temporal integration and a nonlinear transformation.

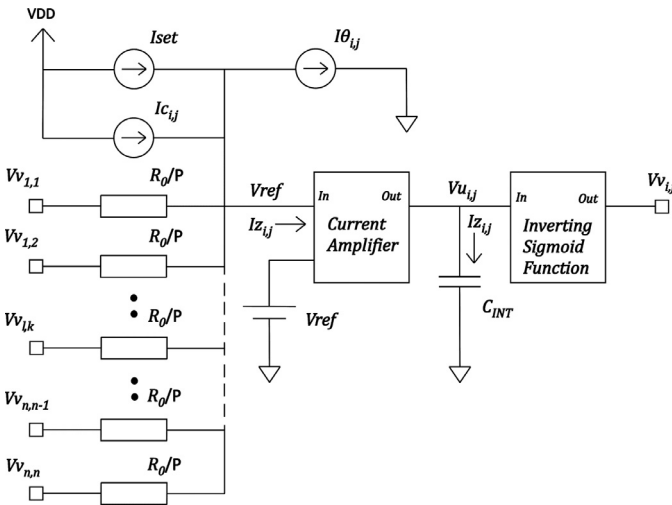


Fig. 5. Electrical configuration of neuron N_{ij} for matrix array. This system has interconnection resistors R_0/P , where $P \in \{0, 1\}$ obeys to a lateral connection rule. The current amplifier adds the arriving currents at the input node with constant voltage $Vref$ to create Iz_{ij} . It is copied and integrated by capacitor C_{INT} to produce Vu_{ij} . Finally, Vu_{ij} is transformed by the inverting sigmoid function to provide the output voltage Vv_{ij} .

the inverting sigmoid, which is defined in Eq. (17).

$$v_{i,j} = \frac{1}{(1 + e^{-\beta u_{i,j}})} \quad (17)$$

We remark that $c_{i,j}$ and 2η are the entries of new matrices: \mathbf{C} and $\boldsymbol{\theta}$, respectively. $c_{i,j}$ and 2η electrically use memristors to encode “Assignment” and “Transportation” problems, treated below as demonstrative applications.

5. Electrical configuration of the neuron

In this section, we show the analog components that build the recurrent neural network, which solves the “Assignment” and “Transportation” problems. In Fig. 5 are shown the processing components in terms of voltage and current variables for the reference neuron N_{ij} shown in Fig. 4. The weighting factors W_0 are taken as conductances proportional to $1/R_0$, where the set of resistors whose individual value R_0 provides a current that is summed along

Table 1
Electrical and model parameters.

$I\theta_{i,j}$	2η	Parameter, constant
$Ic_{i,j}$	$\lambda c_{i,j} e^{-\frac{V}{T}}$	Parameter, decaying

with $Iset$, $I\theta_{i,j}$ and $Ic_{i,j}$ to form the current $Iz_{i,j}$. This analog variable is entered to the current amplifier with unitary gain, which outputs a replica of $Iz_{i,j}$ to the capacitor C_{INT} . This capacitor works as an integrator of $Iz_{i,j}$ in order to obtain the voltage $Vu_{i,j}$. The inverting sigmoid function reads $Vu_{i,j}$ to produce $Vv_{i,j}$. The current amplifier also copies the voltage $Vref$ to its input.

Below is the analytical description of this electrical system that leads to a differential equation that is equivalent to (15), considering that the existing connections are those that fulfill the condition: P is equal to 1 if $k=j$ or $l=i$ (otherwise, $P=0$ and, there is no connection).

Applying Kirchhoff’s Current Law at the input of the current amplifier, $Iz_{i,j}$ is expressed as

$$Iz_{i,j} = \frac{1}{R_0} \sum_{l,k=1}^n (Vv_{l,k} - Vref) + Ic_{i,j} - I\theta_{i,j} + Iset \quad (18)$$

$$Iz_{i,j} = \frac{1}{R_0} \sum_{l,k=1}^n Vv_{l,k} - \frac{2nVref}{R_0} + Ic_{i,j} - I\theta_{i,j} + Iset \quad (19)$$

The term $-(2nVref)/R_0$ acts as a bias voltage. Its effect can be reduced to zero by setting $Iset$ to

$$Iset = \frac{2nVref}{R_0} \quad (20)$$

Thus, (19) is simplified to

$$Iz_{i,j} = \frac{1}{R_0} \sum_{l,k=1}^n Vv_{l,k} + Ic_{i,j} - I\theta_{i,j} \quad (21)$$

We should note that the summation in (21) contains a self-connection, whose value is $2/R_0$.

The next expression after (21) is (22). Where, C_{INT} is the capacitor that integrates $Iz_{i,j}$ to produce $Vu_{i,j}$.

$$\frac{dVu_{i,j}}{dt} = \frac{1}{C_{INT}} \left\{ \frac{1}{R_0} \sum_{k=1}^n Vv_{i,k} + \frac{1}{R_0} \sum_{l=1}^n Vv_{l,j} + Ic_{i,j} - I\theta_{i,j} \right\} \quad (22)$$

(22) is equivalent to (15) but with converse signs on the terms of the right hand side to be taken into consideration with the inverting sigmoid function. Table 1 relates electrical and model parameters.

The analog input currents $I\theta_{i,j}$ and $Ic_{i,j}$ are the entries of the matrices $\boldsymbol{\theta}$ and \mathbf{C} , respectively that represent the combinatorial optimization problems in this work.

6. Analog CMOS circuits

We have chosen a low cost standard $0.5 \mu\text{m}$, n-well, CMOS technology to design the analog neural network circuits, that were simulated as 3-by-3 matrices of neurons. The power supply VDD is 5 V. The BSIM3v3 MOSFET model electrical parameters of this CMOS technology are provided by ON-Semiconductor to run on the IC-Nanometer Suite by Mentor.

The CMOS circuits that form the Current Amplifier and the Inverting Sigmoid Function are shown in Figs. 6 and 7a, respectively. The W/L ratio of the transistors is shown as $\mu\text{m}/\mu\text{m}$ (this nomenclature applies for all the electrical diagrams). This Current Amplifier is useful for signal processing applications in current-mode designs [18]. It works following the translinear principle of the loop

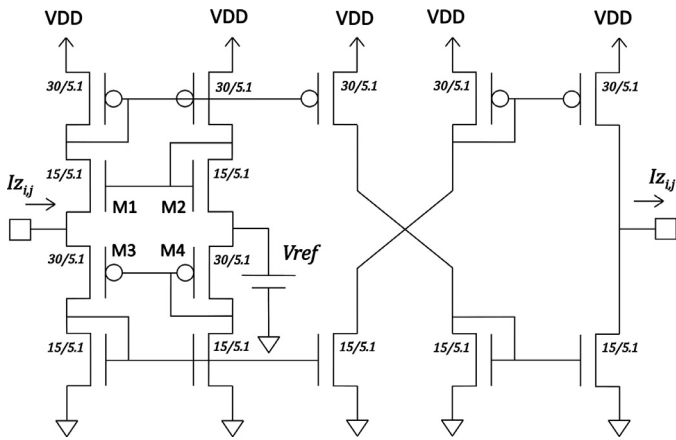


Fig. 6. Current amplifier. This is a current mode analog amplifier designed for unitary gain, where the input current $I_{z_{ij}}$ on the left, is provided as a copy on the right.

formed by transistors M1, M2, M3 and M4 transferring V_{ref} to the input. It has unitary gain. $V_{ref} = VDD/2$ was chosen. Fig. 7a is a push-pull analog amplifier, whose gain and DC transfer function are set by the transistors size, approximating the inverting non-linear sigmoid function in (17) and shifting to the first quadrant. Fig. 7b presents $Vv_{i,j}$ versus $Vu_{i,j}$ of the push pull amplifier, where the gain at $Vv_{i,j} = VDD/2$ is moderate.

7. Memristive components

The TEAM model was implemented with Verilog-A instructions [19]. They were run on the IC-Nanometer Suite by Mentor. Table 2 presents the values of the parameters used in the simulations of the recurrent neural networks. These values correspond to the memristors used whose technology is oriented to fast digital electronics [20]. Using the values in Table 2 and the setup in Fig. 8a, the obtained simulated response is shown in Fig. 8b, where the frequency of the sinusoidal signal covers the range: $1\text{ MHz} < f < 3\text{ MHz}$. Fig. 8c shows applied voltage V_m , memristive current I_m , effective width d and memristance R_M (from

Table 2
TEAM parameters used in simulations.

Parameters	Values
dt	$1 \times 10^{-9}\text{ s}$
R_{ON}	$1 \times 10^3\text{ ohm}$
R_{OFF}	$500 \times 10^3\text{ ohm}$
i_{off}	$50 \times 10^{-12}\text{ A}$
i_{on}	$-50 \times 10^{-12}\text{ A}$
W_C	$107 \times 10^{-12}\text{ m}$
a_{on}	0 m
a_{off}	$3 \times 10^{-9}\text{ m}$
k_{on}	$-1 \times 10^{-8}\text{ m/s}$
k_{off}	$1 \times 10^{-8}\text{ m/s}$
α_{on}	1
α_{off}	1
x_{on}	0 m
x_{off}	$3 \times 10^{-9}\text{ m}$

top to down) for 1 MHz, 2 MHz and 3 MHz. In Table 2, doing $dt = 1 \times 10^{-9}\text{ s}$ allows saving computation time. We simulated lots of networks setting their individual integrating capacitors C_{INT} at initial conditions of combinatorial cases of either 0V or 5V and choosing the best solution.

In Fig. 5, the set of resistors, whose number by neuron is $2(n - 1) + 2$, is composed by memristors with a resistive value: $R_0 = R_{OFF}$ namely, in the Off-state resistance. This condition demands that the sense of the flowing current never changes, whatever its magnitude is; however, we note that the sense of the signal current flow in any memristor might change. Fig. 9 depicts the actual memristive circuit that emulates a constant R_{OFF} for the reference connection between $Vv_{l,k}$ and V_{ref} . There, two memristors are selected by a “comparator”, which is formed by a chain of 4 digital inverters, whose threshold is V_{ref} by design. Memristors MEMR1 and MEMR2 work always in the Off-state resistance. They are selected alternatively when $Vv_{l,k} > V_{ref}$ and $Vv_{l,k} < V_{ref}$, by the transmission gates TG1 and TG2, respectively. TG1 and TG2 are operated by opposite logic levels due to Va and Vb.

Both I_{set} and $I_{\theta_{i,j}}$ provide constant currents and have similar circuit configuration, which also uses memristors in the Off-state resistance. Figs. 10 and 11 show the implementation of the circuits for $I_{set}/2$ and $I_{\theta_{i,j}}$ respectively, where the difference $VDD - V_{ref}$

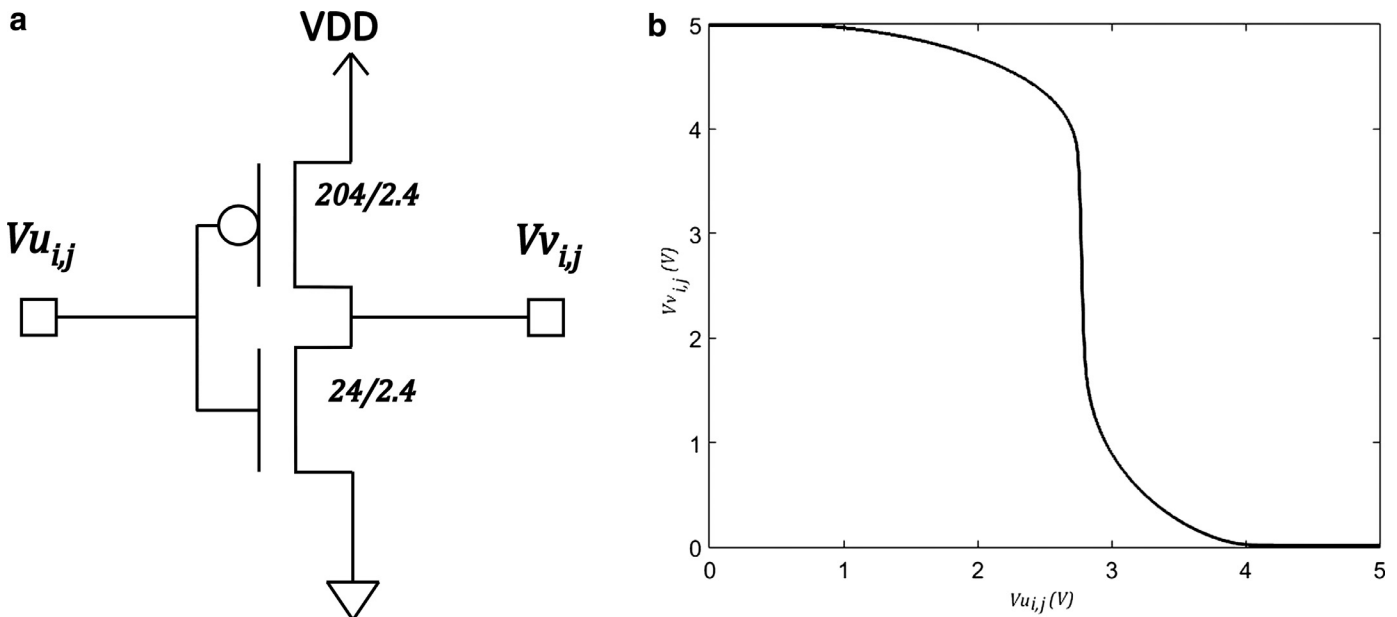


Fig. 7. a. Inverting sigmoid function. This is a push-pull CMOS amplifier that approximates (17). b. Simulation in DC of the inverting sigmoid function.

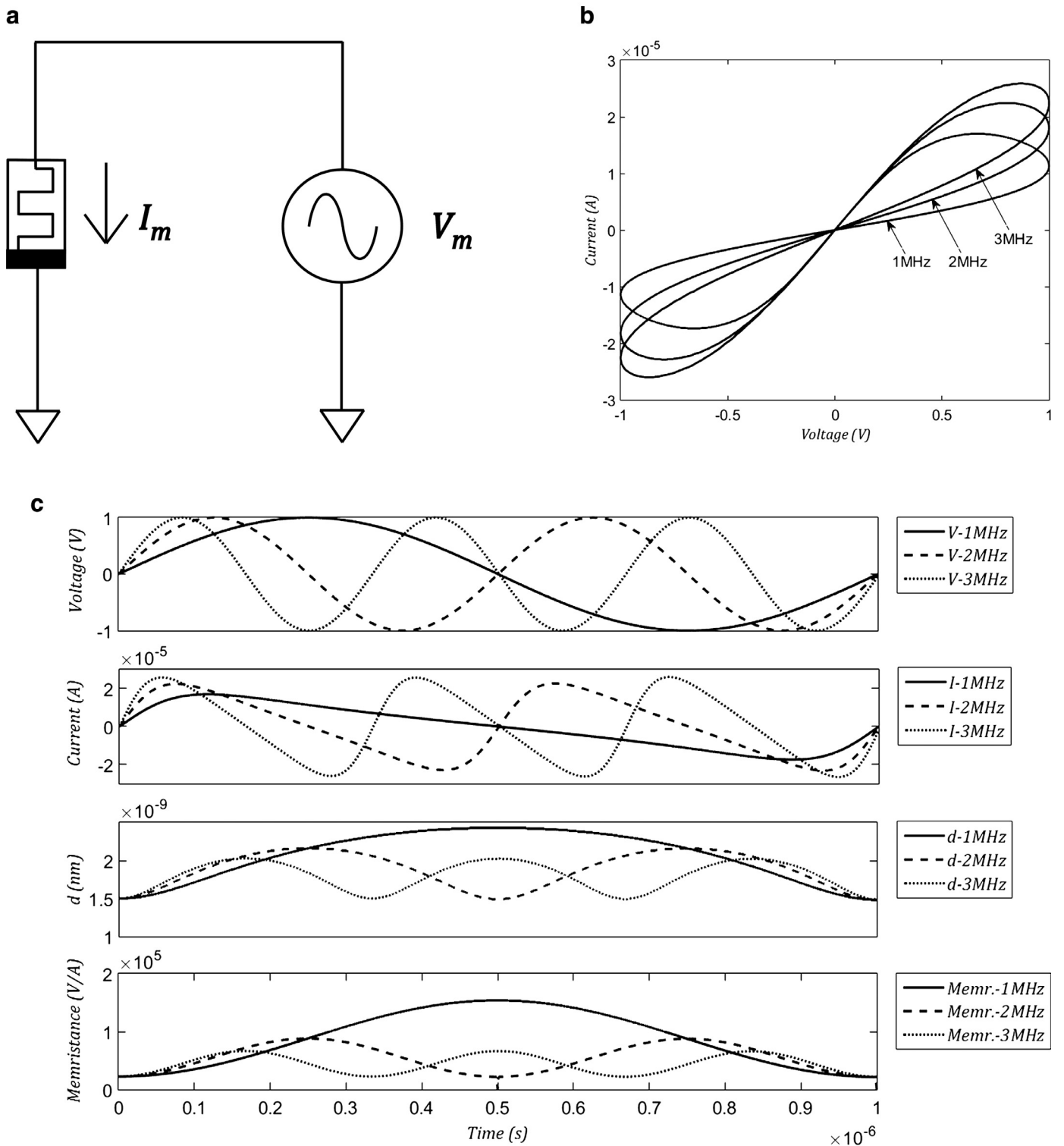


Fig. 8. a. Setup for simulating the $i-v$ characteristic of the memristor. b. Simulation of the $i-v$ characteristic of the memristor with frequency as parameter. c. Simulation as function of time. Top to down graphs: Applied sinusoidal voltage V_m , Current I_m , Effective width d , Memristance R_m .

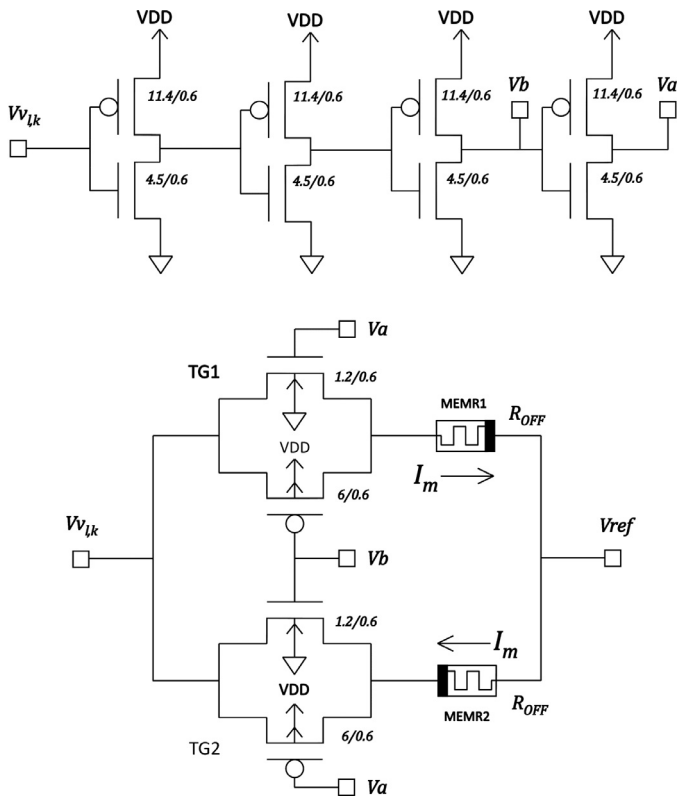


Fig. 9. Circuit for emulating the resistive connection $R_0 = R_{OFF}$ between neurons. Upper diagram: The chain of 4 digital inverters acts as a comparator to the analog voltage $V_{V_{lk}}$. The logic threshold of each inverter is set to V_{ref} by design. Lower diagram: Switched paths between $V_{V_{lk}}$ and V_{ref} due to transmission gates TG1 and TG2 that avoid change of sense of current in MEMR1 and MEMR2 and keep them at R_{OFF} . The digital voltages V_a and V_b for controlling TG1 and TG2 come from the comparator (chain of 4 inverters).

takes part of the design. This memristive strategy reduces active electronics to a minimal and gets a high output resistance. Based on the electrical simulation, V_{g1} and V_{g2} are determined for specific DC currents of $I_{set}/2$ and $I\theta_{i,j}$, which are shown in Fig. 12

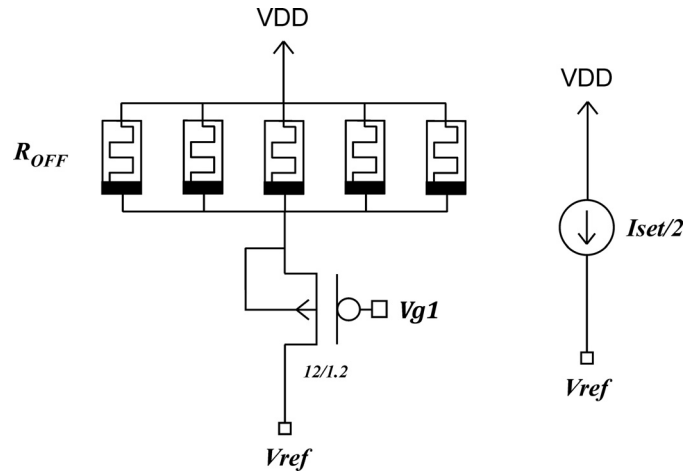


Fig. 10. Current source $I_{set}/2$. It is implemented using 5 parallel memristors in the R_{OFF} state in series with a p-channel MOS transistor. The analog voltage V_{g1} sets the current.

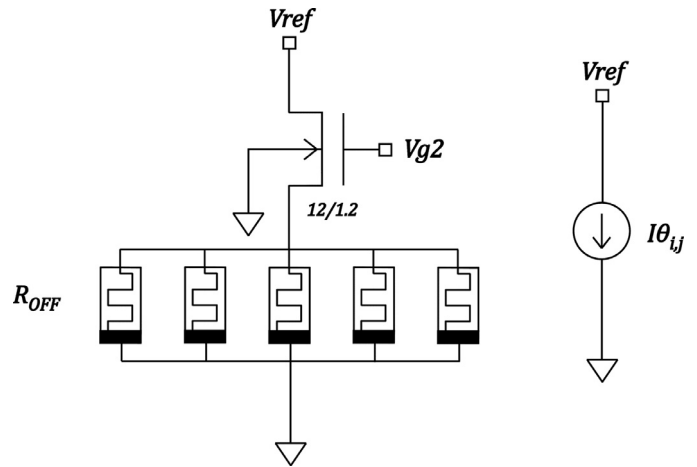


Fig. 11. Current source $I\theta_{i,j}$. It is implemented using 5 parallel memristors in the R_{OFF} state in series with an n-channel MOS transistor. The analog voltage V_{g2} sets the current.

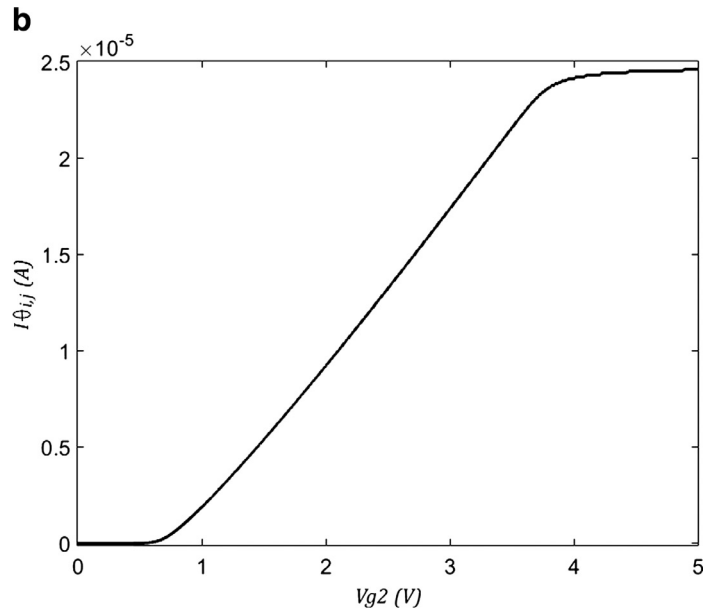
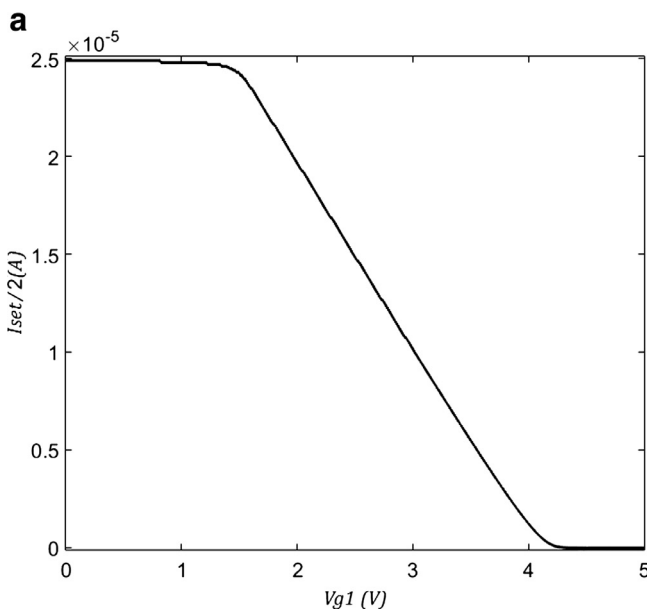


Fig. 12. a. Simulation in DC of $I_{set}/2$ versus V_{g1} . The useful linear characteristic is in $1.7V < V_{g1} < 3.85V$ for $1.0 \mu A < I_{set}/2 < 23.0 \mu A$, where the circuit in Fig. 10 works properly. b. Simulation in DC of $I\theta_{i,j}$ versus V_{g2} . The useful linear characteristic is in $1.0V < V_{g2} < 3.70V$ for $1.0 \mu A < I\theta_{i,j} < 23.0 \mu A$, where the circuit in Fig. 11 works properly.

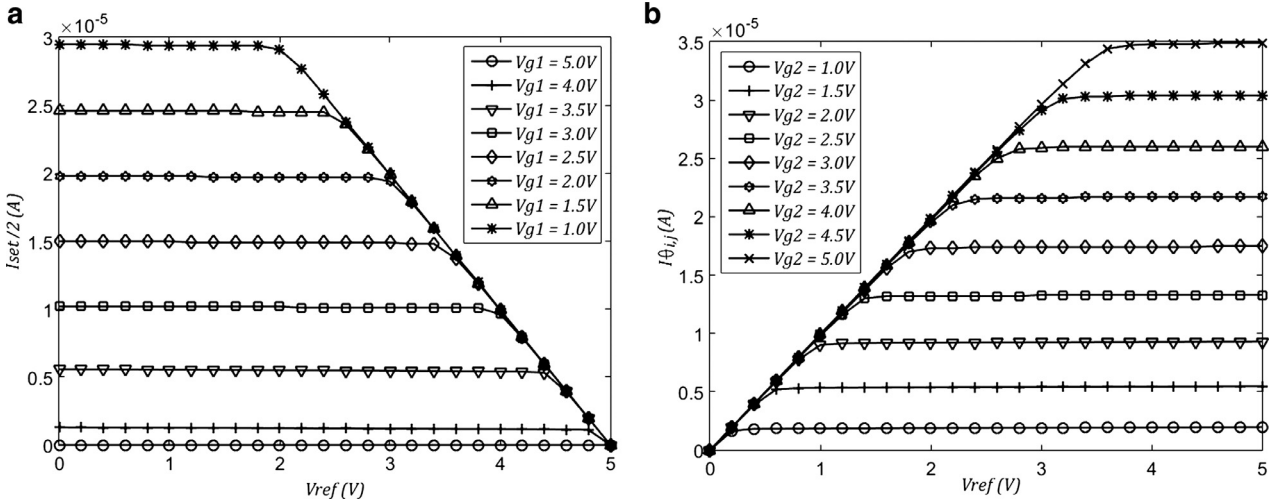


Fig. 13. a. Simulation in DC of $I_{set}/2$ versus V_{ref} , with V_{g1} as parameter. It is chosen $V_{g1} = 2.5V$ for $I_{set}/2 = 15 \mu A$ and $V_{ref} = 2.5V$. These values are used in the neural network. b. Simulation in DC of $I_{\theta_{i,j}}$ versus V_{ref} , with V_{g2} as parameter. V_{g2} should be chosen in the interval $0.8V < V_{g2} < 3.5V$ to be inside the interval $0 \mu A < I_{\theta_{i,j}} < 20 \mu A$. $V_{ref} = 2.5V$.

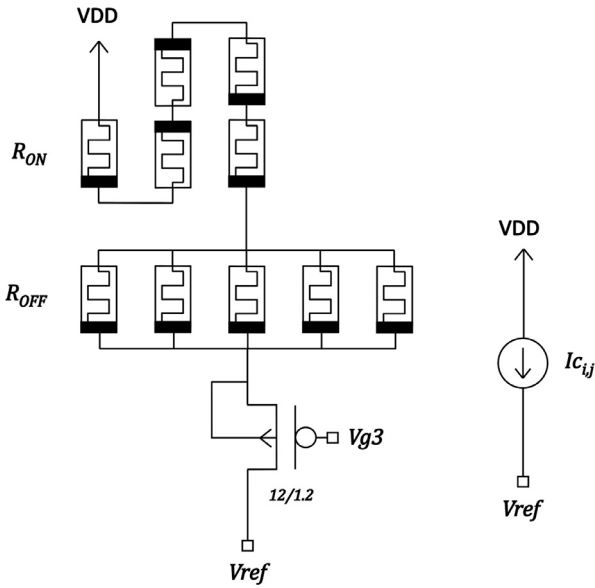


Fig. 14. Current source $I_{c_{i,j}}$. It is implemented using 5 parallel memristors in the R_{OFF} state in series with 5 series memristors in the R_{ON} state and a p-channel MOS transistor. The analog voltage V_{g3} sets the peak current.

and b, respectively. Fig. 13a and b provides evidence that both current sources have high output resistance at the working voltage $V_{ref} = 2.5V$. Fig. 14 refers to the circuit configuration of $I_{c_{i,j}}$, which emulates a decaying current and replaces $\lambda c_{i,j} \exp(-t/\tau)$ in the model presented by Wang. Although there is a series transistor, and (9) predicts a lower final current I_{MF} , its decaying characteristic is still suitable. The analog voltage V_{g3} sets the initial amplitude of the current or the peak current; this value is determined using (8), where $V_M = V_M(V_{g3})$ and $S_M = 5$, $P_M = 5$. The I-V DC characteristic is drawn in Fig. 15a. Whether the values $2 \mu A$, $4 \mu A$, $6 \mu A$ and $10 \mu A$ are chosen as the peak currents, the corresponding values of V_{g3} are $3.74V$, $3.48V$, $3.24V$, $3.00V$ and $2.76V$. The transient outcome, with these peak currents is shown in Fig. 15b. In the circuits that approach the currents $I_{set}/2$, $I_{\theta_{i,j}}$ and $I_{c_{i,j}}$ the source terminal of the series transistor sets the voltage across the corresponding memristive array following to V_{g1} , V_{g2} and V_{g3} , respectively. At this point, Fig. 10 deserves noting that it is designed for $I_{set}/2$ due to functionality reason namely, the series p-MOS transistor works properly. The actual current source I_{set} is implemented by two parallel circuits providing a current equals to $I_{set}/2$ each.

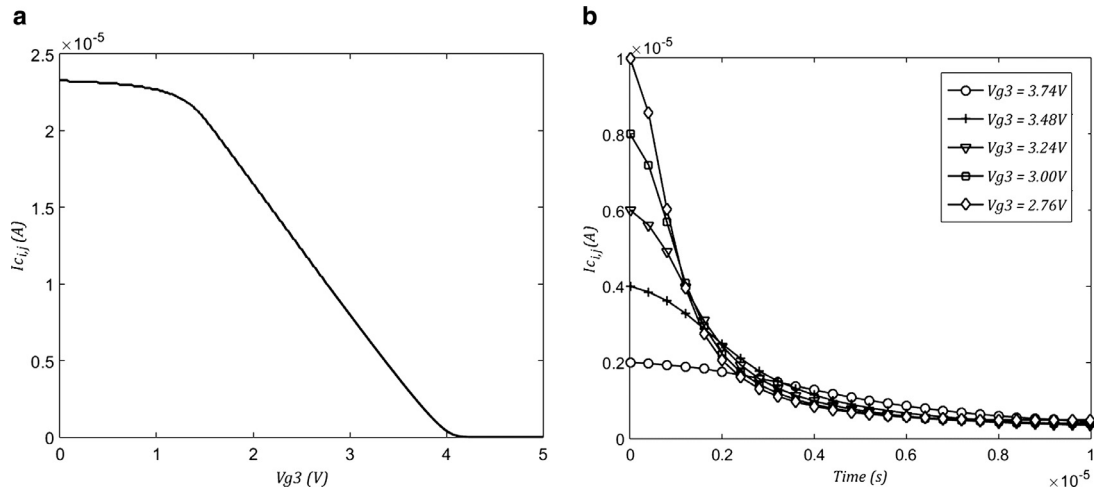


Fig. 15. a. Simulation in DC of $I_{c_{i,j}}$ versus V_{g3} . The peak current $I_{c_{i,j}}$ can be set in the interval $1.0 \mu A < I_{c_{i,j}} < 20 \mu A$ corresponding to the interval $1.5V < V_{g3} < 3.9V$, where the circuit in Fig. 14 works properly. b. Simulation of $I_{c_{i,j}}$. The peak current is set with V_{g3} . From Fig. 15a, the values: $I_{c_{i,j}} = 10 \mu A$, $8 \mu A$, $6 \mu A$, $4 \mu A$, and $2 \mu A$ are set using $V_{g3} = 2.76V$, $3.0V$, $3.24V$, $3.48V$ and $3.74V$, respectively.

Table 3-A
Numerical costs and electrical currents.

Neuron (<i>ij</i>)	Netwk1	Netwk2	Netwk3	Netwk1	Netwk2	Netwk3
	C_{ij}	C_{ij}	C_{ij}	I_{Cij}	I_{Cij}	I_{Cij}
(1, 1)	7	1	8	7.77	1.11	8.88
(1, 2)	6	7	6	6.66	7.77	6.66
(1, 3)	8	3	1	8.88	3.33	1.11
(2, 1)	9	6	4	9.99	6.66	4.44
(2, 2)	4	5	3	4.44	5.55	3.33
(2, 3)	3	4	5	3.33	4.44	5.55
(3, 1)	5	4	5	5.55	4.44	5.55
(3, 2)	8	6	2	8.88	6.66	2.22
(3, 3)	6	9	9	6.66	9.99	9.99

Table 3-B
Analog and numerical solutions.

Neuron (<i>ij</i>)	Analog solution V_{vij}			Numerical solution		
	Netwk1	Netwk2	Netwk3	Netwk1	Netwk2	Netwk3
(1, 1)	0.58382	4.22076	0.03629	0	1	0
(1, 2)	3.97953	0.01529	0.01471	1	0	0
(1, 3)	0.31686	0.60430	4.74946	0	0	1
(2, 1)	0.02195	0.01533	4.04663	0	0	1
(2, 2)	0.71387	0.79085	0.81807	0	0	0
(2, 3)	4.17154	4.07552	0.01708	1	1	0
(3, 1)	4.30346	0.59964	0.81689	1	0	0
(3, 2)	0.19650	4.08336	4.02668	0	1	1
(3, 3)	0.32928	0.22205	0.01484	0	0	0

8. Analog solution of the “Assignment” problem

In this section, we present the solution of “Assignment” problems using the memristive recurrent neural network. Considering Fig. 5 for a 3-by-3 system of neurons, Eq. (20) evaluated for $V_{ref}=VDD/2=2.5V$, $R_0 = R_{OFF} = 500 \times 10^3 \text{ Ohm}$, which is implemented with the circuit configuration of Fig. 10, leads to a value of $I_{set} = 30 \mu A$. The value of the integrating capacitor C_{INT} is 2.3 pF, which is proposed as 1.0 pF by design in parallel with 1.3 pF due to the gate capacitances of the inverting sigmoid function. The value of the threshold current source $I_{\theta_{i,j}}$ is constant and the same for all neurons, which is established as $I_{\theta_{i,j}} = 20 \mu A$. In Table 3-A are shown the numerical costs and their respective electrical currents $I_{C_{i,j}}$ in μA of 3 “Assignment” problems identified as Netwk1, Netwk2 and Netwk3. Table 3-B shows their analog solution in volts along with the expected numerical solution. Fig. 16a, b and c presents the electrical traces $V_{v_{i,j}}$ of Netwk1, Netwk2 and Netwk3 based on the SPICE simulations. Comparing the analog solution with the numerical solution, we observe that both are equal, considering that if $V_{v_{i,j}}$ is nearly 5V or 0V, it is read as “1” or “0”, respectively. Appendix A presents an algebraic analysis of the “Assignment” problem for completeness.

9. Analog solution of the “Transportation” problem

In this section, we present the solution of “Transportation” problems using the memristive recurrent neural network. Considering Fig. 5 for a 3-by-3 system of neurons, Eq. (20) evaluated for $V_{ref}=VDD/2=2.5V$, $R_0 = R_{OFF} = 500 \times 10^3 \text{ Ohm}$, which is implemented with the circuit configuration of Fig. 10, leads to a value of $I_{set} = 30 \mu A$. The value of the integrating capacitor C_{INT} is 2.3 pF, which is proposed as 1.0 pF by design in parallel with 1.3 pF due to the gate capacitances of the inverting sigmoid function. In Table 4-A are shown the numerical costs, and the offer and demand values of 3 “transportation” problems, identified as Netwk4, Netwk5 and Netwk6. Table 4-B presents the current in μA of the corresponding values of Table 4-A. $I_{C_{i,j}}$, I_{a_i} , and I_{b_j} are calculated

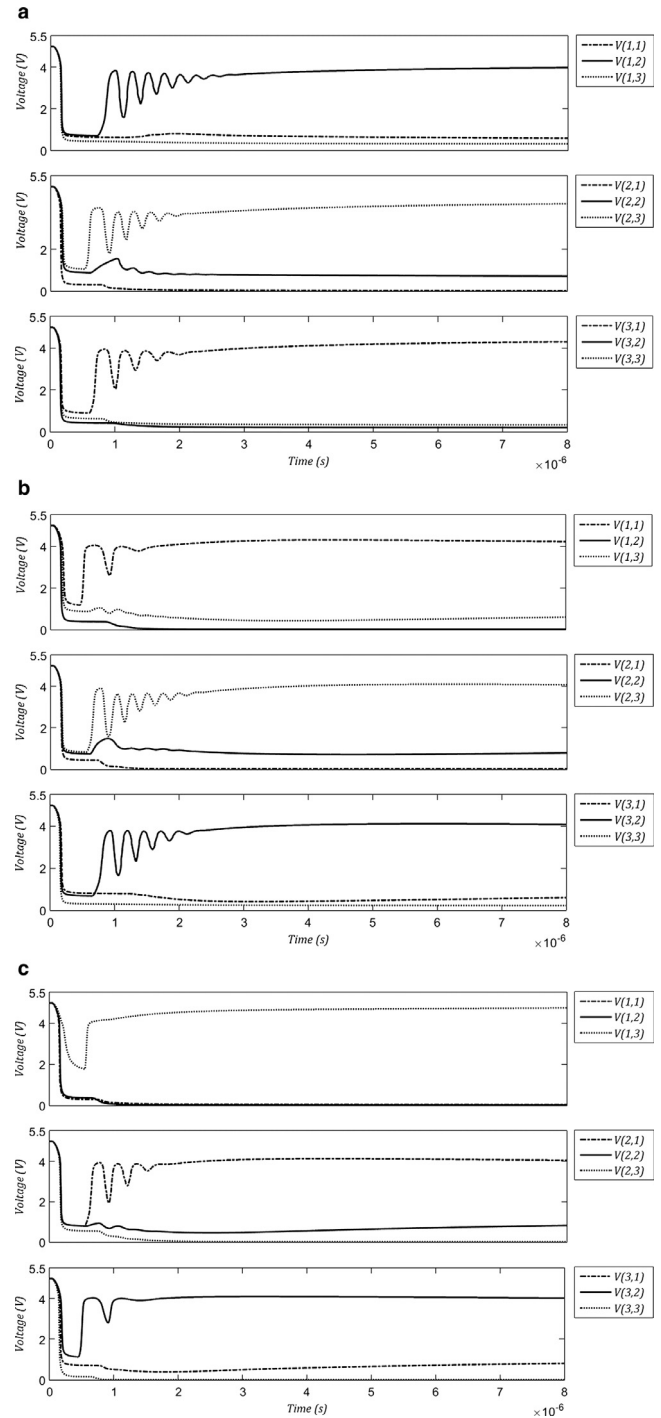


Fig. 16. a. Simulation of Netwk1. b. Simulation of Netwk2. c. Simulation of Netwk3.

Table 4-A
Numerical costs, and offer and demand values.

Neuron (<i>ij</i>)	Netwk4			Netwk5			Netwk6		
	C_{ij}	a_i	b_j	C_{ij}	a_i	b_j	C_{ij}	a_i	b_j
(1, 1)	7	40	25	7	70	50	8	40	30
(1, 2)	6	40	35	4	70	80	6	40	80
(1, 3)	8	40	60	15	70	45	1	40	10
(2, 1)	9	50	25	1	90	50	4	60	30
(2, 2)	4	50	35	19	90	80	3	60	80
(2, 3)	3	50	60	2	90	45	5	60	10
(3, 1)	5	30	25	3	15	50	5	20	30
(3, 2)	8	30	35	12	15	80	2	20	80
(3, 3)	6	30	60	10	15	45	9	20	10

Table 4-B
Electrical currents.

Neuron (i,j)	Netwk4				Netwk5				Netwk6			
	$\bar{I}_{c_{i,j}}$	\bar{I}_{a_i}	\bar{I}_{b_j}	I_{θ}	$\bar{I}_{c_{i,j}}$	\bar{I}_{a_i}	\bar{I}_{b_j}	I_{θ}	$\bar{I}_{c_{i,j}}$	\bar{I}_{a_i}	\bar{I}_{b_j}	I_{θ}
(1, 1)	7.77	6.66	4.16	10.82	3.684	7.77	5.55	13.32	8.88	5.00	3.75	8.750
(1, 2)	6.66	6.66	5.83	12.49	2.105	7.77	8.88	16.65	6.66	5.00	10.0	15.00
(1, 3)	8.88	6.66	10.0	16.66	7.894	7.77	5.00	12.77	1.11	5.00	1.25	6.250
(2, 1)	9.99	8.33	4.16	12.49	0.526	10.0	5.55	15.55	4.44	7.50	3.75	11.25
(2, 2)	4.44	8.33	5.83	14.16	10.00	10.0	8.88	18.88	3.33	7.50	10.0	17.50
(2, 3)	3.33	8.33	10.0	18.33	1.052	10.0	5.00	15.00	5.55	7.50	1.25	8.750
(3, 1)	5.55	5.00	4.16	9.16	1.578	1.66	5.55	7.210	5.55	2.50	3.75	6.250
(3, 2)	8.88	5.00	5.83	10.83	6.315	1.66	8.88	10.54	2.22	2.50	10.0	12.50
(3, 3)	6.66	5.00	10.0	15.00	5.263	1.66	5.00	6.660	9.99	2.50	1.25	3.750

Table 4-C
Analog solution.

Neuron (i,j)	Netwk4 v_{ij} (V)	Netwk5 v_{ij} (V)	Netwk6 v_{ij} (V)
(1, 1)	0.40308	0.01490	0.45477
(1, 2)	2.65081	3.85953	1.29839
(1, 3)	0.17217	0.01459	0.64134
(2, 1)	0.01454	2.50312	1.18499
(2, 2)	0.01454	0.01464	2.51945
(2, 3)	4.08842	2.56797	0.01481
(3, 1)	1.58085	0.30123	0.12369
(3, 2)	0.15685	0.47318	1.10332
(3, 3)	0.66717	0.01460	0.01473

as: $\bar{I}_{c_{i,j}} = (c_{i,j}/\max(c_{i,j}))(10 \mu\text{A})$, $\bar{I}_{a_i} = (a_i/\max(a_i, b_j))(10 \mu\text{A})$ and, $\bar{I}_{b_j} = (b_j/\max(a_i, b_j))(10 \mu\text{A})$.

I_{θ} comes from adding \bar{I}_{a_i} and \bar{I}_{b_j} . Table 4-C is the analog solution $v_{i,j}$ in volts from simulations. In Table 4-D $v_{i,j}$ is calculated by $v_{i,j} = (v_{i,j} \text{ in Volts}/(5 \text{ volts}))(\max(a_i, b_j))$. The offer a_i and demand b_j values in Table 4-D are computed as $a_i = \sum v_{i,j}$ with $j=1,2,3$ and $b_j = \sum v_{i,j}$ with $i=1,2,3$. Additionally, the total cost of Netwk4, Netwk5 and Netwk6 are calculated as $\sum (c_{i,j})(v_{i,j})$ with $i, j=1,2,3$. In Table 4-E are shown the values from numerical simulations with Simulink, where $v_{i,j} = (v_{i,j} \text{ in Simulink})(5 \text{ volts})(\max(a_i, b_j))$. Fig. 17a, b and c presents the electrical traces $Vv_{i,j}$ of Netwk4, Netwk5, and Netwk6 respectively, based on SPICE simulations. The total costs by the memristive neural networks in Table 4-D are lower than those by the numerical solution in Table 4-E. Appendix B presents an algebraic analysis of the ‘‘Transportation’’ problem for completeness.

10. Comment on performance

The starting random state value of the memristors deserves a comment on the performance of the whole system. Setting every memristor at R_{ON} or R_{OFF} should be done before the ana-

log computing cycle begins. Naming T_{SET} as the setting period, T_{SET} should be long enough to change in parallel some memristors from R_{ON} to R_{OFF} and the other ones conversely. From simulations, the smallest value of the computing cycle time is $T_{PROCESS} = 8.0 \mu\text{s}$. Also, from simulations $T_{SET} > 650 \text{ ns}$. Defining $P = 100\% \times (T_{PROCESS} - T_{SET})/(T_{PROCESS})$ as a Performance Figure of Merit and using the above numbers, we get $P = 91\%$. Therefore, this analog system supports an acceptable performance.

11. Conclusions

This paper was motivated by the availability of advanced memristor models for circuit design, in particular the TEAM model. The studied neural network was originally modeled by its author using one-value resistive components and decaying currents, leading to linear resistors and discharging capacitors for its implementation; in contrast, we have proposed memristive/CMOS circuit configurations that behave according to the analog parameters of the neural network, solving in continuous-time mode the optimization problems. For this goal, we have chosen electrical parameter values in the TEAM model that belong to fast switching memristors; their dynamic characteristics are complementary of those used in present analog designs, e.g. slow memristive Hopfield networks. We observe that our electrical system represents an innovative system, where new configuration circuits were introduced.

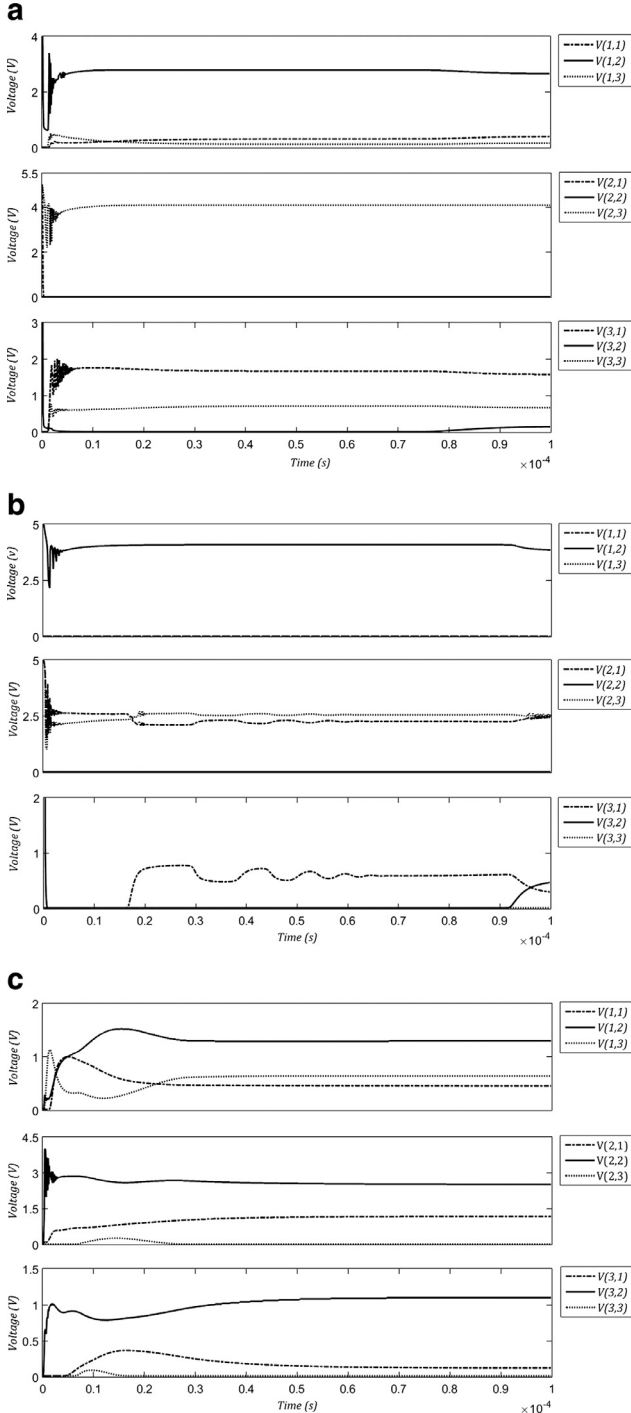
This work contributes in the VLSI design area by proposing the use of memristors working in one of two memristances R_{OFF} or R_{ON} . This system sustains 3 features, namely: **1)** It does not require any programming electronics but flowing current in the direction indicated in Fig. 1, where the proper polarity in the used memristor is chosen. **2)** Using R_{OFF} simplifies the implementation of the current sources: I_{set} and $I_{\theta_{ij}}$ namely, reduces transistor count and avoids external analog bias voltages. Observing the model by Dr. Wang given by (22), where the value of the resistive interconnections R_0 is constant, it can be chosen as R_{OFF} . **3)** The parallel (R_{OFF})-series (R_{ON}) configuration presented in Fig. 14 is an original idea

Table 4-D
Analog solution.

Neuron (i,j)	Netwk4			Netwk5			Netwk6		
	v_{ij}	a_i	b_j	v_{ij}	a_i	b_j	v_{ij}	a_j	b_j
(1, 1)	4.8370	38.712	23.981	0.2682	70.002	50.746	7.2763	38.312	28.215
(1, 2)	31.809	38.712	33.866	69.471	70.002	78.752	20.774	38.312	78.738
(1, 3)	2.0660	38.712	59.133	0.2626	70.002	46.748	10.261	38.312	10.734
(2, 1)	0.1745	49.410	23.981	45.056	91.543	50.746	18.959	59.508	28.215
(2, 2)	0.1745	49.410	33.866	0.2635	91.543	78.252	40.311	59.508	78.738
(2, 3)	49.061	49.410	59.133	45.223	91.543	46.748	0.2370	59.508	10.734
(3, 1)	18.970	28.858	23.981	5.4221	14.202	50.546	1.9790	19.867	28.215
(3, 2)	1.8822	28.858	33.866	8.5172	14.202	78.252	17.653	19.867	78.738
(3, 3)	8.0060	28.858	59.133	0.2628	14.202	46.748	0.2357	19.867	10.734
	Total cost = 548.641			Total cost = 547.314			Total cost = 438.397		

Table 4-E
Numerical solution.

Neuron (i, j)	Netwk4			Netwk5			Netwk6		
	v_{ij}	a_i	b_j	v_{ij}	a_i	b_j	v_{ij}	a_i	b_j
(1, 1)	3.7917	39.946	24.938	0.0000	69.930	49.945	0.1541	39.767	29.773
(1, 2)	34.899	39.946	34.971	69.930	69.930	79.919	29.763	39.767	79.803
(1, 3)	1.2559	39.346	59.986	0.0000	69.930	44.995	9.8503	39.767	10.158
(2, 1)	0.0000	49.976	24.938	44.993	89.991	49.945	29.465	59.947	29.773
(2, 2)	0.0718	49.976	34.971	0.0000	89.991	79.919	30.327	59.947	79.803
(2, 3)	49.904	49.976	59.986	44.995	89.991	44.995	0.1541	59.947	10.158
(3, 1)	21.146	29.973	24.938	4.9497	14.939	49.945	0.1541	20.020	29.773
(3, 2)	0.0000	29.973	34.971	9.9899	14.939	79.919	19.712	20.020	79.803
(3, 3)	8.8268	29.973	59.986	0.0000	14.939	44.995	0.1541	20.020	10.158
	Total cost = 554.671			Total cost = 549.428			Total cost = 440.862		

**Fig. 17.** a. Simulation of Netwk4. b. Simulation of Netwk5. c. Simulation of Netwk6.

of replacing a large capacitive and resistive element to represent the decaying term: $\lambda c_{ij} \exp(-t/\tau)$ in (15), which is essential to set the parameters of the problem. Alternative ways of interconnecting memristors different from the crossbar architecture, which leads to cellular arrays, support the physical implementation of the analog system in this paper.

Planning in robotics would be a problem for solving by this recurrent neural network. A larger memristive system based on the model by Wang, working as optimizer, is possible by redesigning the CMOS circuits already exposed.

Appendix A

The “Assignment” problem is a linear programming problem, whose description can be associated with the statement:

“There is the need to assign available ‘resources’ to ‘agents’ to perform a set of tasks at minimal cost”.

This problem is expressed as follows.

$$\text{Minimize } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} v_{ij} \quad (\text{A.1})$$

$$\text{Subject to } \sum_{j=1}^n v_{ij} = 1, \quad i = 1, \dots, n \quad (\text{A.2})$$

$$\sum_{i=1}^m v_{ij} = 1, \quad j = 1, \dots, n \quad (\text{A.3})$$

$$v_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n;$$

where, c_{ij} represents one element of the cost and v_{ij} is the decision variable.

In particular, we obtain the solution to the “Assignment” problem for a 3-by-3 system, using the recurrent neural network by Wang [10]. The mathematical formulation of the supporting analog neural network is

$$\frac{du_{ij}(t)}{dt} = -\eta \sum_{k=1}^n v_{ik}(t) - \eta \sum_{l=1}^n v_{lj}(t) + 2\eta - \lambda c_{ij} e^{-\frac{t}{\tau}} \quad (\text{A.4})$$

$$v_{ij} = g(u_{ij}(t)) \quad (\text{A.5})$$

where, η and λ are constants. $g(\cdot)$ is a sigmoid function, providing the solution is in the range [0, 1].

Eqs. (A.1)–(A.3) for the case of a 3-by-3 system are now represented by:

$$\text{Minimize } z = \sum_{i=1}^3 \sum_{j=1}^3 c_{ij} v_{ij} \quad (\text{A.6})$$

$$\text{Subject to } \sum_{j=1}^3 v_{ij} = 1, \quad i = 1, \dots, 3 \quad (\text{A.7})$$

$$\sum_{i=1}^3 v_{ij} = 1, \quad j = 1, \dots, 3$$

$$v_{ij} \in [0, 1], \quad i = 1, \dots, 3; \quad j = 1, \dots, 3 \tag{A.8}$$

Eqs. (A.6)–(A.9) get a matrix form as follows

$$\text{Minimize } z = \mathbf{c}^T \mathbf{v} \tag{A.9}$$

$$\text{Subject to } \mathbf{A}\mathbf{v} = \mathbf{b} \tag{A.10}$$

Eq. (A.10) can be developed for the 3-by-3 system, to have

$$\text{Minimize } z = [c_{11} \ c_{12} \ c_{13} \ c_{21} \ c_{22} \ c_{23} \ c_{31} \ c_{32} \ c_{33}] \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} \tag{A.11}$$

Eq. (A.11) uses matrix \mathbf{A} and vector \mathbf{b} as follows, also for the 3-by-3 system.

$$\mathbf{A} = \begin{bmatrix} I & I & I \\ B_1 & B_2 & B_3 \end{bmatrix} \tag{A.12}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{A.13}$$

$$\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{A.14}$$

Finally, the set of restrictions are expressed as

$$\mathbf{A}\mathbf{v} = \mathbf{b} \tag{A.15}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{A.16}$$

$$v_{11} + v_{21} + v_{31} = 1 \tag{A.17}$$

$$v_{12} + v_{22} + v_{32} = 1 \tag{A.18}$$

$$v_{13} + v_{23} + v_{33} = 1 \tag{A.19}$$

$$v_{11} + v_{12} + v_{13} = 1 \tag{A.20}$$

$$v_{21} + v_{22} + v_{23} = 1 \tag{A.21}$$

$$v_{31} + v_{32} + v_{33} = 1 \tag{A.22}$$

The set of differential Eq. (A.4) has its matrix form given as

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\mathbf{v} - \boldsymbol{\theta}) - \lambda \mathbf{c}e^{-\frac{t}{\tau}} \tag{A.23}$$

The matrix of weights \mathbf{W} is defined as

$$\mathbf{W} = \mathbf{A}^T \mathbf{A} \tag{A.24}$$

$$\mathbf{W} = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \tag{A.25}$$

The threshold matrix $\boldsymbol{\theta}$ is given as

$$\boldsymbol{\theta} = \mathbf{A}^T \mathbf{b} \tag{A.26}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{A.27}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \tag{A.28}$$

Replacing matrices \mathbf{W} y $\boldsymbol{\theta}$ in the below matrix form, where λ is defined as: $1/c_{max}$, we get for the 3-by-3 system, the below algebraic analysis:

$$\frac{d\mathbf{u}(t)}{dt} = -\eta(\mathbf{W}\mathbf{v} - \boldsymbol{\theta}) - \lambda \mathbf{c}e^{-\frac{t}{\tau}} \tag{A.29}$$

$$\frac{d\mathbf{u}(t)}{dt} = -\eta \left(\begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \right) - \lambda \begin{bmatrix} c_{11}e^{-\frac{t}{\tau}} \\ c_{12}e^{-\frac{t}{\tau}} \\ c_{13}e^{-\frac{t}{\tau}} \\ c_{21}e^{-\frac{t}{\tau}} \\ c_{22}e^{-\frac{t}{\tau}} \\ c_{23}e^{-\frac{t}{\tau}} \\ c_{31}e^{-\frac{t}{\tau}} \\ c_{32}e^{-\frac{t}{\tau}} \\ c_{33}e^{-\frac{t}{\tau}} \end{bmatrix} \tag{A.30}$$

$$\frac{du_{11}}{dt} = -2\eta v_{11} - \eta(v_{12} + v_{13}) - \eta(v_{21} + v_{31}) + 2\eta - \lambda c_{11}e^{-\frac{t}{\tau}} \tag{A.31}$$

$$\frac{du_{12}}{dt} = -2\eta v_{12} - \eta(v_{11} + v_{13}) - \eta(v_{22} + v_{32}) + 2\eta - \lambda C_{12} e^{-\frac{t}{\tau}} \tag{A.32}$$

$$\frac{du_{13}}{dt} = -2\eta v_{13} - \eta(v_{11} + v_{12}) - \eta(v_{23} + v_{33}) + 2\eta - \lambda C_{13} e^{-\frac{t}{\tau}} \tag{A.33}$$

$$\frac{du_{21}}{dt} = -2\eta v_{21} - \eta(v_{22} + v_{23}) - \eta(v_{11} + v_{31}) + 2\eta - \lambda C_{21} e^{-\frac{t}{\tau}} \tag{A.34}$$

$$\frac{du_{22}}{dt} = -2\eta v_{22} - \eta(v_{21} + v_{23}) - \eta(v_{12} + v_{32}) + 2\eta - \lambda C_{22} e^{-\frac{t}{\tau}} \tag{A.35}$$

$$\frac{du_{23}}{dt} = -2\eta v_{23} - \eta(v_{21} + v_{22}) - \eta(v_{13} + v_{33}) + 2\eta - \lambda C_{23} e^{-\frac{t}{\tau}} \tag{A.36}$$

$$\frac{du_{31}}{dt} = -2\eta v_{31} - \eta(v_{32} + v_{33}) - \eta(v_{11} + v_{21}) + 2\eta - \lambda C_{31} e^{-\frac{t}{\tau}} \tag{A.37}$$

$$\frac{du_{32}}{dt} = -2\eta v_{32} - \eta(v_{31} + v_{33}) - \eta(v_{12} + v_{22}) + 2\eta - \lambda C_{32} e^{-\frac{t}{\tau}} \tag{A.38}$$

$$\frac{du_{33}}{dt} = -2\eta v_{33} - \eta(v_{31} + v_{32}) - \eta(v_{13} + v_{23}) + 2\eta - \lambda C_{33} e^{-\frac{t}{\tau}} \tag{A.39}$$

Appendix B

The ‘‘Transportation’’ problem, also known as ‘‘Distribution’’ problem is a linear programming problem, whose description can be associated with the statement:

‘‘There is the need to transport units from a site call ‘The Source’ to other site called ‘The Destination’ minimizing the cost of sending and at the same time satisfying offer and demand restrictions’’.

This problem is expressed as follows.

$$\text{Minimize } z = \sum_{i=1}^g \sum_{j=1}^h c_{ij} v_{ij} \tag{B.1}$$

$$\text{Subject to } \sum_{j=1}^h v_{ij} = a_i, \quad i = 1, \dots, g \tag{B.2}$$

$$\sum_{i=1}^g v_{ij} = b_j, \quad j = 1, \dots, h$$

$$v_{ij} \geq 0, \quad i = 1, \dots, g; \quad j = 1, \dots, h \tag{B.3}$$

where, c_{ij} represents one element of the cost and v_{ij} is the decision variable. There would be slack components in order to have a square matrix of size $n \times n$.

In particular, we obtain the solution to the ‘‘Transportation’’ problem for a 3-by-3 system, using the recurrent neural network by Wang [10] and following the method proposed in [21]. The mathematical formulation of the supporting analog neural network is

$$\frac{du_{ij}(t)}{dt} = -\eta \sum_{k=1}^n \bar{v}_{ik}(t) - \eta \sum_{l=1}^n \bar{v}_{lj}(t) + \eta \theta_{ij} - \lambda c_{ij} e^{-\frac{t}{\tau}} \tag{B.4}$$

$$\bar{v}_{ij} = g(u_{ij}(t)) \tag{B.5}$$

where, η and λ are constants. $g(\cdot)$ is a sigmoid function, providing the solution is in the range $[0, 1]$; therefore, the values of a , b y v in Eqs. (B.1)–(B.3) are also in this range.

Defining: $q = \max\{a_i, b_i : i = 1, 2, 3\}$ then, a new set of variables is

$$v = q\bar{v}, \quad a = q\bar{a} \quad y \quad b = q\bar{b}.$$

Using the new values of a y b , Equations (B.1)–(B.3) for the case of a 3-by-3 system are now represented by

$$\text{Minimize } z = \sum_{i=1}^3 \sum_{j=1}^3 c_{ij} \bar{v}_{ij} \tag{B.6}$$

$$\text{Subject to } \sum_{j=1}^3 \bar{v}_{ij} = \bar{a}_i, \quad i = 1, \dots, 3 \tag{B.7}$$

$$\sum_{i=1}^3 \bar{v}_{ij} = \bar{b}_j, \quad j = 1, \dots, 3$$

$$\bar{v}_{ij} \in [0, 1], \quad i = 1, \dots, 3; \quad j = 1, \dots, 3 \tag{B.8}$$

Eqs. (B.7)–(B.10) get a matrix form as follows

$$\text{Minimize } \bar{z} = \mathbf{c}^T \bar{\mathbf{v}} \tag{B.9}$$

$$\text{Subject to } \mathbf{A} \bar{\mathbf{v}} = \mathbf{d} \tag{B.10}$$

Eq. (B.11) can be developed for the 3-by-3 system, to have

$$\text{Minimize } \bar{z} = [c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \quad c_{31} \quad c_{32} \quad c_{33}] \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix} \tag{B.11}$$

Eq. (B.11) uses matrix \mathbf{A} and vector \mathbf{d} as follows, also for the 3-by-3 system

$$\mathbf{A} = \begin{bmatrix} B_1 & B_2 & B_3 \\ I & I & I \\ 0 & 0 & 0 \end{bmatrix} \tag{B.12}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.13}$$

$$\mathbf{d} = \begin{bmatrix} \bar{a} \\ \bar{b} \\ 0 \end{bmatrix} \tag{B.14}$$

$$\mathbf{d} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.15}$$

Finally, the offer and demand restrictions are expressed as

$$A\bar{v} = d \tag{B.16}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_1 \\ \bar{b}_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.17}$$

$$\bar{v}_{11} + \bar{v}_{12} + \bar{v}_{13} = \bar{a}_1 \tag{B.18}$$

$$\bar{v}_{21} + \bar{v}_{22} + \bar{v}_{23} = \bar{a}_2 \tag{B.19}$$

$$\bar{v}_{31} + \bar{v}_{32} + \bar{v}_{33} = \bar{a}_3 \tag{B.20}$$

$$\bar{v}_{11} + \bar{v}_{21} + \bar{v}_{31} = \bar{b}_1 \tag{B.21}$$

$$\bar{v}_{12} + \bar{v}_{22} + \bar{v}_{32} = \bar{b}_2 \tag{B.22}$$

$$\bar{v}_{13} + \bar{v}_{23} + \bar{v}_{33} = \bar{b}_3 \tag{B.23}$$

The set of differential equation (B.4) has its matrix form given as

$$\frac{du(t)}{dt} = -\eta(W\bar{v} - \theta) - \lambda ce^{-\frac{t}{\tau}} \tag{B.24}$$

The matrix of weights W is defined as

$$W = A^T A \tag{B.25}$$

$$W = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \tag{B.26}$$

The offer and demand limits are related by matrix θ as

$$\theta = A^T d \tag{B.27}$$

$$\theta = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.28}$$

$$\theta = \begin{bmatrix} \bar{a}_1 + \bar{b}_1 & \bar{a}_1 + \bar{b}_2 & \bar{a}_1 + \bar{b}_3 \\ \bar{a}_2 + \bar{b}_1 & \bar{a}_2 + \bar{b}_2 & \bar{a}_2 + \bar{b}_3 \\ \bar{a}_3 + \bar{b}_1 & \bar{a}_3 + \bar{b}_2 & \bar{a}_3 + \bar{b}_3 \end{bmatrix} \tag{B.29}$$

Replacing matrices W y θ in the below matrix form, where λ is defined as: $1/c_{max}$, we get for the 3-by-3 system, the below algebraic analysis.

$$\frac{du(t)}{dt} = -\eta(W\bar{v} - \theta) - \lambda ce^{-\frac{t}{\tau}} \tag{B.30}$$

$$\frac{du(t)}{dt} = -\eta \begin{pmatrix} \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{12} \\ \bar{v}_{13} \\ \bar{v}_{21} \\ \bar{v}_{22} \\ \bar{v}_{23} \\ \bar{v}_{31} \\ \bar{v}_{32} \\ \bar{v}_{33} \end{bmatrix} - \begin{bmatrix} \bar{a}_1 + \bar{b}_1 \\ \bar{a}_1 + \bar{b}_2 \\ \bar{a}_1 + \bar{b}_3 \\ \bar{a}_1 + \bar{b}_1 \\ \bar{a}_1 + \bar{b}_2 \\ \bar{a}_1 + \bar{b}_3 \\ \bar{a}_1 + \bar{b}_1 \\ \bar{a}_1 + \bar{b}_2 \\ \bar{a}_1 + \bar{b}_3 \end{bmatrix} - \lambda \begin{bmatrix} c_{11}e^{-\frac{t}{\tau}} \\ c_{12}e^{-\frac{t}{\tau}} \\ c_{13}e^{-\frac{t}{\tau}} \\ c_{21}e^{-\frac{t}{\tau}} \\ c_{22}e^{-\frac{t}{\tau}} \\ c_{23}e^{-\frac{t}{\tau}} \\ c_{31}e^{-\frac{t}{\tau}} \\ c_{32}e^{-\frac{t}{\tau}} \\ c_{33}e^{-\frac{t}{\tau}} \end{bmatrix} \end{pmatrix} \tag{B.31}$$

$$\frac{du_{11}}{dt} = -2\eta\bar{v}_{11} - \eta(\bar{v}_{12} + \bar{v}_{13}) - \eta(\bar{v}_{21} + \bar{v}_{31}) + \eta(\bar{a}_1 + \bar{b}_1) - \lambda c_{11}e^{-\frac{t}{\tau}} \tag{B.32}$$

$$\frac{du_{12}}{dt} = -2\eta\bar{v}_{12} - \eta(\bar{v}_{11} + \bar{v}_{13}) - \eta(\bar{v}_{22} + \bar{v}_{32}) + \eta(\bar{a}_1 + \bar{b}_2) - \lambda c_{12}e^{-\frac{t}{\tau}} \tag{B.33}$$

$$\frac{du_{13}}{dt} = -2\eta\bar{v}_{13} - \eta(\bar{v}_{11} + \bar{v}_{12}) - \eta(\bar{v}_{23} + \bar{v}_{33}) + \eta(\bar{a}_1 + \bar{b}_3) - \lambda c_{13}e^{-\frac{t}{\tau}} \tag{B.34}$$

$$\frac{du_{21}}{dt} = -2\eta\bar{v}_{21} - \eta(\bar{v}_{22} + \bar{v}_{23}) - \eta(\bar{v}_{11} + \bar{v}_{31}) + \eta(\bar{a}_2 + \bar{b}_1) - \lambda c_{21}e^{-\frac{t}{\tau}} \tag{B.35}$$

$$\frac{du_{22}}{dt} = -2\eta\bar{v}_{22} - \eta(\bar{v}_{21} + \bar{v}_{23}) - \eta(\bar{v}_{12} + \bar{v}_{32}) + \eta(\bar{a}_2 + \bar{b}_2) - \lambda c_{22}e^{-\frac{t}{\tau}} \tag{B.36}$$

$$\frac{du_{23}}{dt} = -2\eta\bar{v}_{23} - \eta(\bar{v}_{21} + \bar{v}_{22}) - \eta(\bar{v}_{13} + \bar{v}_{33}) + \eta(\bar{a}_2 + \bar{b}_3) - \lambda c_{23}e^{-\frac{t}{\tau}} \tag{B.37}$$

$$\frac{du_{31}}{dt} = -2\eta\bar{v}_{31} - \eta(\bar{v}_{32} + \bar{v}_{33}) - \eta(\bar{v}_{11} + \bar{v}_{21}) + \eta(\bar{a}_3 + \bar{b}_1) - \lambda c_{31}e^{-\frac{t}{\tau}} \tag{B.38}$$

$$\frac{du_{32}}{dt} = -2\eta\bar{v}_{32} - \eta(\bar{v}_{31} + \bar{v}_{33}) - \eta(\bar{v}_{12} + \bar{v}_{22}) + \eta(\bar{a}_3 + \bar{b}_2) - \lambda c_{32}e^{-\frac{t}{\tau}} \tag{B.39}$$

$$\frac{du_{33}}{dt} = -2\eta\bar{v}_{33} - \eta(\bar{v}_{31} + \bar{v}_{32}) - \eta(\bar{v}_{13} + \bar{v}_{23}) + \eta(\bar{a}_3 + \bar{b}_3) - \lambda c_{33}e^{-\frac{t}{\tau}} \tag{B.40}$$

References

[1] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, *Nature* 453 (2008) 80–83.
 [2] S.K. Tripathi, R. Kaur, M. Rani, Oxide nanomaterials and their applications as a memristor, *Solid State Phenom.* 222 (2015) 67–97.

- [3] N.M. Muhammad, N. Duraisamy, K. Rahman, H.W. Dang, J. Jo, K.H. Choi, Fabrication of printed memory device having zinc-oxide active nano-layer and investigation of resistive switching, *Curr. Appl. Phys* 13 (2013) 90–96.
- [4] Y.V. Pershin, M. Di Ventra, Memcomputing and swarm intelligence, <https://arxiv.org/abs/1408.6741>.
- [5] A. Gelencsér, T. Podromakis, C. Toumazou, T. Roska, Biomimetic model of the outer plexiform layer by incorporating memristive devices, *Phys. Rev. E* 85 (2012) 041918–1–041918–10.
- [6] S.P. Mohanty, Memristor: from basics to deployment, *IEEE Potentials* 32 (2013) 34–39.
- [7] L. Gao, F. Merrikh-Bayat, F. Alibart, X. Guo, B.D. Hoskins, K.T. Cheng, D.B. Strukov, Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6623031>.
- [8] S.G. Hu, Y. Liu, Z. Liu, T.P. Chen, J.J. Wang, Q. Yu, L.J. Deng, Y. Yin, S. Hosaka, Associative memory realized by a reconfigurable memristive Hopfield neural network, *Nat Commun.* 6 (2015) 1–8.
- [9] U.P. Wen, K.M. Lan, H.S. Shih, A review of Hopfield neural networks for solving mathematical programming problems, *Eur. J. Oper. Res* 198 (2009) 675–687.
- [10] J. Wang, Analysis and design of an analog sorting network, *IEEE Trans. Neural Netw.* 6 (4) (1995) 962–971.
- [11] A. Adamatzky, L. Chua (Eds.), *Memristor Networks*, Springer, Switzerland, 2014, pp. 1–13.
- [12] T. Prodromakis, B.P. Peh, C. Papavassiliou, C. Toumazou, A versatile memristor model with nonlinear dopant kinetics, *IEEE Trans. Electron. Devices* 58 (9) (2011) 3099–3105.
- [13] M.D. Pickett, D.B. Strukov, J.L. Borghetti, J.J. Yang, G.S. Snider, D.R. Stewart, R.S. Williams, Switching dynamics in titanium dioxide memristive devices, *J. Appl. Phys* 106 (2009) 0744508–1–0744508–6.
- [14] S. Kvatinsky, E.G. Friedman, A. Kolodny, U.C. Weiser, TEAM: threshold adaptive memristor model, *IEEE Trans. Circuit Syst.* 60 (1) (2013) 211–221.
- [15] M.S. Feali, A. Ahmadi, Transient response characteristic of memristor circuits and biological-like current spikes, *Neural Comput. Appl.* (2016), doi:10.1007/s00521-016-2248-1.
- [16] J. Wang, Analysis and design of a recurrent neural network for linear programming, *IEEE Trans. Circuit Syst.* 40 (9) (1993) 613–618.
- [17] J. Wang, Analogue neural network for solving the assignment problem, *Electron. Lett.* 28 (11) (1992) 1047–1050.
- [18] G. Palmisano, G. Palumbo, S. Pennisi, *CMOS Current Amplifiers*, Springer, USA, 1999, pp. 45–48.
- [19] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. Weiser, E.G. Friedman, Models of memristors for SPICE simulations, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6377081>.
- [20] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U.C. Weiser, MRL memristor ratioed logic, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6331426>.
- [21] P.H. Siqueira, Application of Wang's recurrent neural network to solve the transportation problem, *Int. J. Comput. Sci. Netw. Secur.* 12 (7) (2012) 50–54.



Gerardo Marcos Tornez Xavier. He is a doctoral student in the Electrical Engineering Department of Cinvestav-IPN, Mexico City, Mexico. He studies dynamical properties of fundamental memristive circuits. His academic achievements include designing with FPGA a multilayer neural network for predicting electrical response of solar modules.



Felipe Gómez Castañeda. He is researcher at the Center for Research and Advanced Studies of the IPN, Cinvestav-IPN, Mexico City, Mexico. He teaches neuro-fuzzy systems for digital realization with FPGA and analog VLSI design. His areas of interests include metaheuristics and adaptive systems, memristive hardware, analog realization of Sigma-Delta modulators.



Luis Martín Flores Nava. He is a lecturer at the Center for Research and Advanced Studies of the IPN, Cinvestav-IPN, Mexico City, Mexico. He teaches digital systems for realization with FPGA. His areas of interests include robotic vision systems and memristive hardware.



José Antonio Moreno Cadenas. He is professor at the Center for Research and Advanced Studies of the IPN, Cinvestav-IPN, Mexico City, Mexico. He teaches neuro-fuzzy systems for digital realization and use in robotic systems. His areas of interests include vision systems, data mining for geographical data and analysis of complex systems with fuzzy and neural networks.

Memristive Optimizer for the Assignment Task

G.M. Tornez-Xavier¹, L.M. Flores-Nava¹, F. Gómez-Castañeda¹, J.A. Moreno-Cadenas¹

¹Department of Electrical Engineering, CINVESTAV-IPN, Mexico City, Mexico

phone (52) 55 5747 3800 Ext. 6261

e-mail: {gtornez, lmflores, fgomez, jmoreno}@cinvestav.mx

Abstract — This work shows an analog CMOS matrix compatible with memristors, where they work as dynamic resistors. They are also programmed to initial state values according to one assignment optimization task, taken here as an application vehicle. The whole matrix accomplishes a parallel competitive computation, where a winner-take-all mechanism is inherent by architecture. The expected complexity due to area in silicon of this memristive circuit and its connectivity are moderate, as is evident from observing its electrical diagram. Also, its performance is acceptable as proven via SPICE simulations, using 0.5-micron CMOS process. Lower processing time might be possible with latest CMOS technologies.

Keywords — Optimizer, memristor, TEAM, Verilog-A, CMOS, competitive system.

I. INTRODUCTION

At present, the design of hardware that should realize efficient and specific analog functions for CMOS integrated circuits is a research issue. This happens in new parallel processing circuits in the category of neuromorphic systems (NMSs), where analog values in the components of vectors or matrices are common. Likewise, the expected results in the processes in NMSs should be optimal. Some functions in early artificial neural systems (ANSs) namely, recognition, function approximation and classification are supported by self-organization and competitive mechanisms. From an engineering point of view, there is an optimizer, which is embedded in both NMSs and ANSs. Type-examples of these optimizers are the back-propagation error algorithm for layered neural networks and the winner-take-all computation. An intelligent and near-optimal performance can already be reproduced with the above optimizers in hardware for time-series patterns. However, taking into account NMSs with temporal spiking neural networks (SNNs), which are the third generation of the artificial neural network models, requires new hardware design methods. This trend is motivated in part for the possibility of including nano-technological memory resistors or memristors in CMOS, which promise the reproduction in hardware of learning algorithms in SNNs. These temporal-learning algorithms would become new optimizers suitable for real-time processing of intelligent brain functions.

In this context, we present an optimizer in hardware that computes in parallel the solution of the assignment task, receiving analog-entries matrices and giving binary-entries matrices. The assignment task refers to a linear programming paradigm that finds the best configuration of a set of “processors” to a set of “figures of merit”, maximizing the performance of the system. This task can be solved by the Hungarian algorithm [1].

This paper is organized as follows. Section II, reviews the assignment task. Section III introduces the memristor and the way its internal state is modeled by the ThrEshold Adaptive Memristor (TEAM) model. Section IV presents the electrical CMOS-memristor architecture that solves the assignment problem. The electrical simulations via SPICE of the competitive CMOS-memristor analog circuit are shown in Section V. This paper ends with concluding remarks, in Section VI.

II. THE ASSIGNMENT TASK

The following statement explains in brief the objective of this task.

The assignment task solves the optimal association of “processors” with selected “figures of merit” (FoMs). One processor has the capability to realize a set of different tasks, we may say $T = \{T_1, T_2, \dots, T_N\}$, where: N is the number of tasks to realize. Likewise, this processor can realize the tasks in T with the corresponding $FoM = \{F_1, F_2, \dots, F_N\}$, where each F_i is a real number in the interval $[0, 1]$. The system in consideration has N processors, where each processor has its own FoM_i where: $i=1, 2, \dots, N$. The assignment task is defined by the singular permutation matrix \mathbf{P} that maximizes the performance of the system. This matrix indicates which processor does which task. The matrix \mathbf{F} specifies the Figures of Merit of the processors, whose entries are: $F_{i,j}$, where the first index $i=1, 2, \dots, N$ counts the Task and the second index $j=1, 2, \dots, N$ counts the Processor.

Four Figure of Merit matrices, \mathbf{F}_1 , \mathbf{F}_2 , \mathbf{F}_3 , and \mathbf{F}_4 are presented below along with the respective optimal matrices: \mathbf{P}_1 , \mathbf{P}_2 , \mathbf{P}_3 , and \mathbf{P}_4 . These optimization cases are used to show the performance of the memristive optimizer in this paper.

$$\mathbf{F}_1 = \begin{bmatrix} 0.25 & 0.56 & 0.82 & 0.29 \\ 0.50 & 0.62 & 0.90 & 0.48 \\ 0.70 & 0.48 & 0.20 & 0.50 \\ 0.63 & 0.24 & 0.06 & 0.37 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} 0.50 & 0.41 & 0.70 & 0.39 \\ 0.40 & 0.08 & 0.55 & 0.35 \\ 0.35 & 0.50 & 0.40 & 0.80 \\ 0.20 & 0.27 & 0.76 & 0.61 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 0.55 & 0.37 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.26 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.07 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad P_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 0.55 & 0.80 & 0.15 & 0.80 \\ 0.85 & 0.45 & 0.85 & 0.47 \\ 0.67 & 0.75 & 0.53 & 0.75 \\ 0.81 & 0.52 & 0.79 & 0.77 \end{bmatrix} \quad P_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

III. THE MEMRISTOR

The memristor is a non-linear resistor with memory that was theoretically predicted by L.O. Chua [2] and recognized as a nanostructure by HP Labs in the first decade of 2000's [3]. At present, there are different research topics involving this electrical component in engineering, physical, computing and biological disciplines. Modeling the memristor for designing electrical systems should take into account the way its state variable x changes. Following the presentation given in [4], we count the models that observe: 1) linear ion drift, 2) a window function, 3) nonlinear ion drift, and 4) a Simmons tunnel barrier. In [4], it is remarked that a good computing model of the memristor fulfills both accuracy and numerical efficiency for a broad set of electrical parameters associated to its technology. Therefore, the authors in [4] present the TEAM model for the titanium oxide memristor approaching the Simmons tunnel barrier, whose original expressions are complex. Fig. 1a) shows the state variable in TEAM, which is complementary to the common definition namely, $x = D - w$; D is the width of the used nanometric material and w is the width of the region the oxygen vacancies fill by displacement. Fig. 1b) also presents its electrical symbol. In TEAM, the dynamic change of x or dx/dt is given in 3 intervals of current defined as:

$$0 < i_{off} < i; \quad i_{on} < i < i_{off}; \quad i < i_{on} < 0.$$

Whose value is modeled with Eqs. (1), (2) and (3).

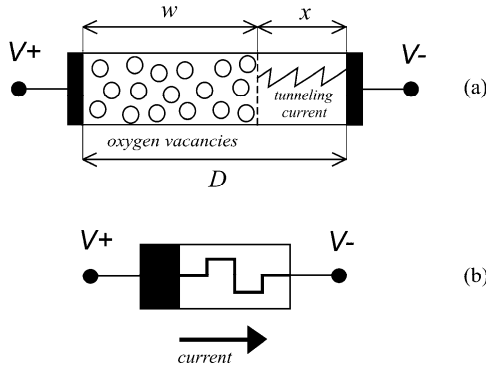


Fig. 1 a) Graphic definition of state variable, b) Symbol

$$\frac{dx}{dt} = K_{off} \left(\frac{i(t)}{i_{off}} - 1 \right)^{\alpha_{off}} f_{off}(x) \quad 0 < i_{off} < i \quad (1)$$

$$\frac{dx}{dt} = K_{on} \left(\frac{i(t)}{i_{on}} - 1 \right)^{\alpha_{on}} f_{on}(x) \quad i < i_{on} < 0 \quad (2)$$

$$\frac{dx}{dt} = 0 \quad i_{on} < i < i_{off} \quad (3)$$

Where: $f_{on}(x)$ and $f_{off}(x)$, are window functions that are evaluated with Eqs. (4) and (5).

$$f_{off}(x) = \exp \left[-\exp \left(\frac{x - a_{off}}{w_c} \right) \right] \quad (4)$$

$$f_{on}(x) = \exp \left[-\exp \left(-\frac{x - a_{on}}{w_c} \right) \right] \quad (5)$$

Finally, the current-voltage relation in the TEAM model can be approached with Eq. (6).

$$v(t) = R_{on} \exp \left(\frac{\lambda}{x_{off} - x_{on}} (x - x_{on}) \right) i(t) \quad (6)$$

Where: λ , is a fitting parameter given by Eq. (7).

$$\lambda = \ln \left(\frac{R_{off}}{R_{on}} \right) \quad (7)$$

We simulated the memristor using the Verilog-A code for the TEAM model reported in [5] with technological and electrical parameters oriented to fast digital systems as given in Table 1. Fig. 2 illustrates the current-voltage characteristic for $f = 2, 3$ and 4 MHz, taken from (6). Fig. 3 shows the applied voltage, the current, the state variable and the memristance as a function of time, in the interval $0-1.0 \mu s$. Fig. 3 (top) considers 3 sinusoidal signals for 2, 3, and 4 MHz with 1 V of amplitude. This type of titanium oxide memristor was used for the memristive Neuron, described in the next Section.

Table 1 TEAM electrical parameters used in simulations.

Parameter	Value	Comment
dt	1×10^{-9} s	Differential of time
R_{on}	$100 \times 10^3 \Omega$	Memristance at $x = 0$ m
R_{off}	$2 \times 10^6 \Omega$	Memristance at $x = 3 \times 10^{-9}$ m
i_{off}	10×10^{-12} A	Threshold current
i_{on}	-10×10^{-12} A	Threshold current
w_c	107×10^{-12} m	Fitting parameter
a_{on}	0 m	Fitting parameter
a_{off}	3×10^{-9} m	Fitting parameter
k_{on}	-5×10^{-8} m/s	Fitting parameter
k_{off}	5×10^{-8} m/s	Fitting parameter
α_{on}	1	Fitting parameter
α_{off}	1	Fitting parameter
x_{on}	0 m	Lower limit of x
x_{off}	3×10^{-9} m	Upper limit of x

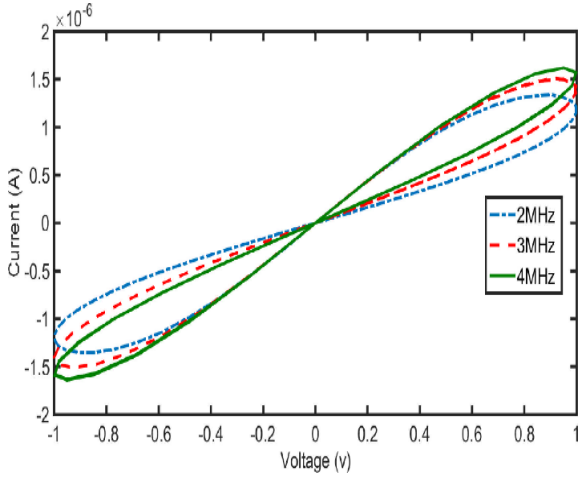


Fig. 2 I-V characteristics in frequency

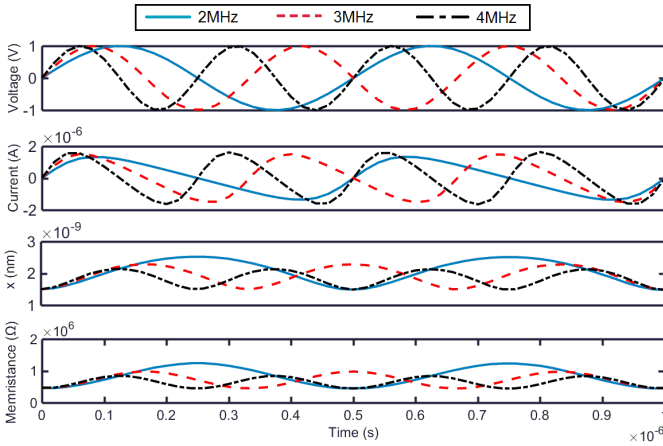


Fig. 3 (from top to bottom) Applied voltage, current, state variable and memristance

IV. MEMRISTIVE NEURON

The memristive CMOS circuit, which should be replicated to build a matrix of dimension $N \times N$ corresponding to the matrices F_1, F_2, F_3 , and F_4 , is presented in Fig. 4. This circuit is called Neuron $N(i, j)$, where: $i=1, 2, \dots, N$ and $j=1, 2, \dots, N$. The electrical matrix behaves as a winner-take-all or competitive system. In order to transfer the values of the entries of the matrix F , the state variable of the memristors of $N(i, j)$ are initially programmed equally to an analog value $\theta_{i,j}$ and proportional to a linear function G of the entry $F_{i,j}$ namely, $\theta_{i,j} = D G(F_{i,j})$, where: D is the width of the memristor. The Neurons settle into a stable matrix P , whose entries are given by the set $\{Vo_{i,j}\}$, which are the output voltages of the CMOS inverters (M1 and M2). Fig. 5 presents the interconnectivity of the Neurons for a $N \times N$ array with lateral connections, without self-connection.

In general, the number of memristors for a $N \times N$ array is $N_{mr} = 2(N)^2(N-1)$; with $N = 4$, $N_{mr} = 96$. In fact, the Neuron $N(i, j)$ follows the circuit configuration of the NOR digital gate

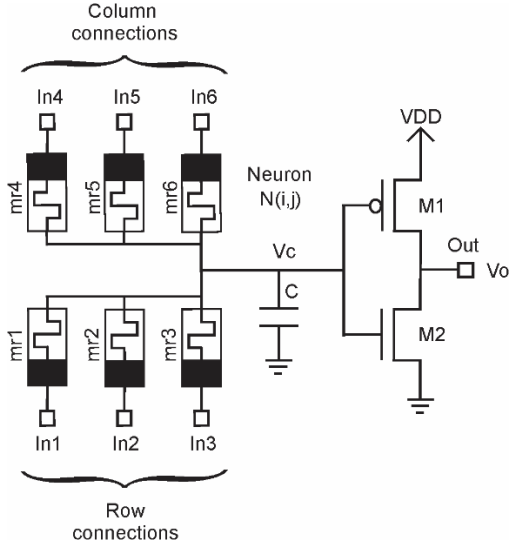


Fig. 4 Memristive Neuron

analyzed in [6], where fast logic gates are designed with memristors. Therefore, we observe that the electrical matrix with memristive Neurons is equivalent to a NOR-gate array, which will be stable and settle into a permutation matrix for all possible initial values of the state variable in the memristors. This is true for architecture principle.

V. ELECTRICAL SIMULATIONS

Using the TEAM model, which was described above, and with the programming lines given in [5] for optimal running time, electrical simulations of the 4×4 Neuron array were realized using ADMS platform of Mentor Graphics. In Fig. 4, the CMOS inverter was simulated using 0.5-micron, n-well, standard technology. Both C and the input capacitance of M1 and M2, which integrate current from the memristors and establish V_C , have the values 1 pF and 0.868 pF, respectively. The ratio W/L of M1 and M2 are 360/0.6 and 120/0.6, where W and L are in μm and $V_{DD} = 5V$.

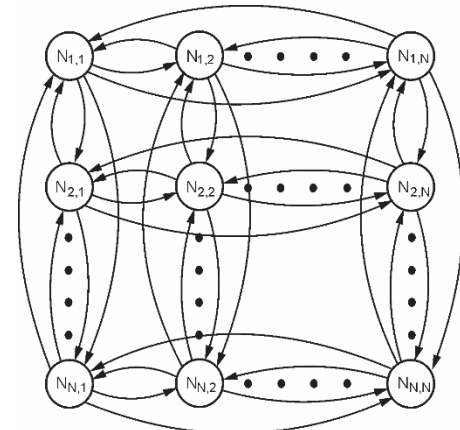


Fig. 5 Interconnectivity of the Neurons array

The six memristors (mr1 to mr6) in Fig. 4 are “programmed” equally as follows. Having the request matrix F and taking the entry $F_{i,j}$ then, the initial value $\theta_{i,j}$ of the state variable of the memristors in $N(i, j)$ is given by Eq. (8) with $i=1..,4$ and $j=1..,4$. Moreover, the set $\{\theta_{i,j}\}$ contains the entries of the initial state matrix θ .

$$\theta_{i,j} = D \left(0.5 + \beta(F_{i,j} - 0.5) \right) \quad (8)$$

The graph of Eq. 8 is shown in Fig. 6, where the values of β were found empirically in the interval: $0.20 \leq \beta \leq 0.80$ and $D = 3 \times 10^{-9}$ m.

Making $\beta=0.5$, the values for $\theta_{i,j}$ are in the range [0.25 to 0.75] for the matrices $\theta_1, \theta_2, \theta_3$, and θ_4 , which correspond to matrices: F_1, F_2, F_3 , and F_4 .

$$\theta_1 = (3 \times 10^{-9}) \begin{bmatrix} 0.375 & 0.530 & 0.660 & 0.395 \\ 0.500 & 0.560 & 0.700 & 0.490 \\ 0.600 & 0.490 & 0.350 & 0.500 \\ 0.565 & 0.370 & 0.280 & 0.435 \end{bmatrix}$$

$$\theta_2 = (3 \times 10^{-9}) \begin{bmatrix} 0.500 & 0.455 & 0.600 & 0.445 \\ 0.450 & 0.290 & 0.525 & 0.425 \\ 0.425 & 0.500 & 0.450 & 0.650 \\ 0.350 & 0.385 & 0.630 & 0.555 \end{bmatrix}$$

$$\theta_3 = (3 \times 10^{-9}) \begin{bmatrix} 0.525 & 0.435 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.380 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.285 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

$$\theta_4 = (3 \times 10^{-9}) \begin{bmatrix} 0.525 & 0.650 & 0.325 & 0.650 \\ 0.675 & 0.475 & 0.675 & 0.485 \\ 0.585 & 0.625 & 0.515 & 0.625 \\ 0.655 & 0.510 & 0.645 & 0.635 \end{bmatrix}$$

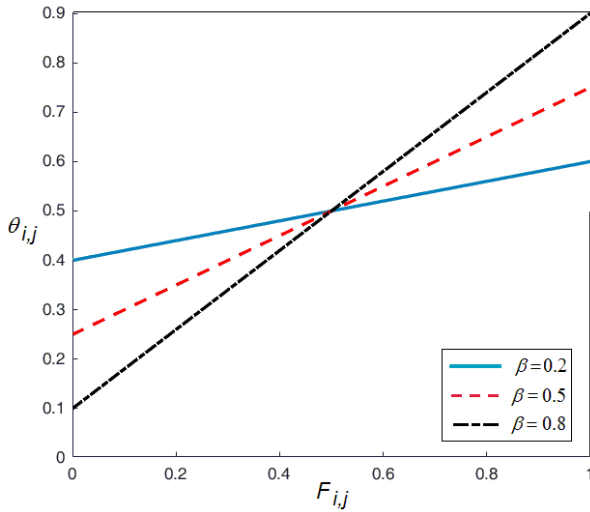


Fig. 6 Initial value $\theta_{i,j}$ of the state variable

Setting $V_C = 0V$ as the initial condition of the simulations, we obtain the expected matrices P_1, P_2, P_3 , and P_4 at the end of the time course namely, at 1×10^{-6} s. Fig. 7 shows the simulations for computing P_1, P_2, P_3 , and P_4 using the initial values in matrices $\theta_1, \theta_2, \theta_3$, and θ_4 . The traces with the expected 5V-value in the permanent regime are recognized as “Winners”. Additionally, Figs. 8a) and 8c) show the final voltages of the “Winner” and the “Loser”, respectively. Also, Figs. 8b) and 8d) show the memristance of the “Winner” and the “Loser”, respectively. In the “Winners” all the memristors get R_{OFF} and, in the “Losers” only two memristors remain at R_{ON} . There is a transient regime inside the interval: $0 < t < 0.2 \times 10^{-6}$ s, where the initial memristance charges C and itself decreases up to R_{ON} due to the difference potential. In that moment the voltage at C reaches a value proportional to $\theta_{i,j}$ and with this value the competition starts.

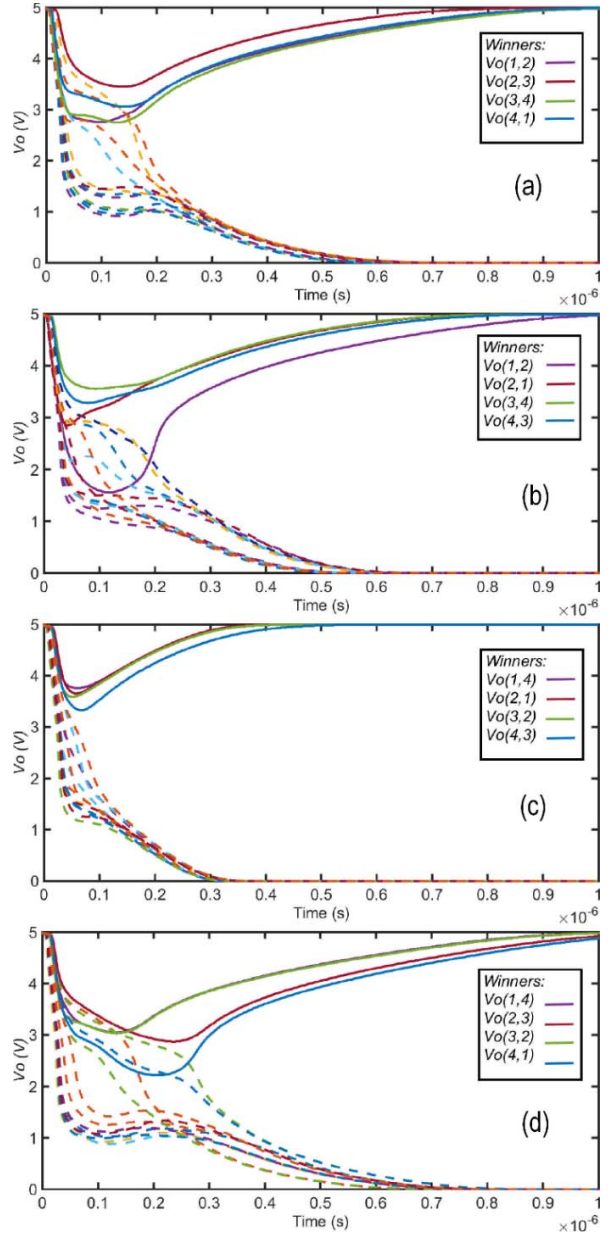


Fig. 7 SPICE simulation for computing a) P_1 , b) P_2 , c) P_3 , and d) P_4

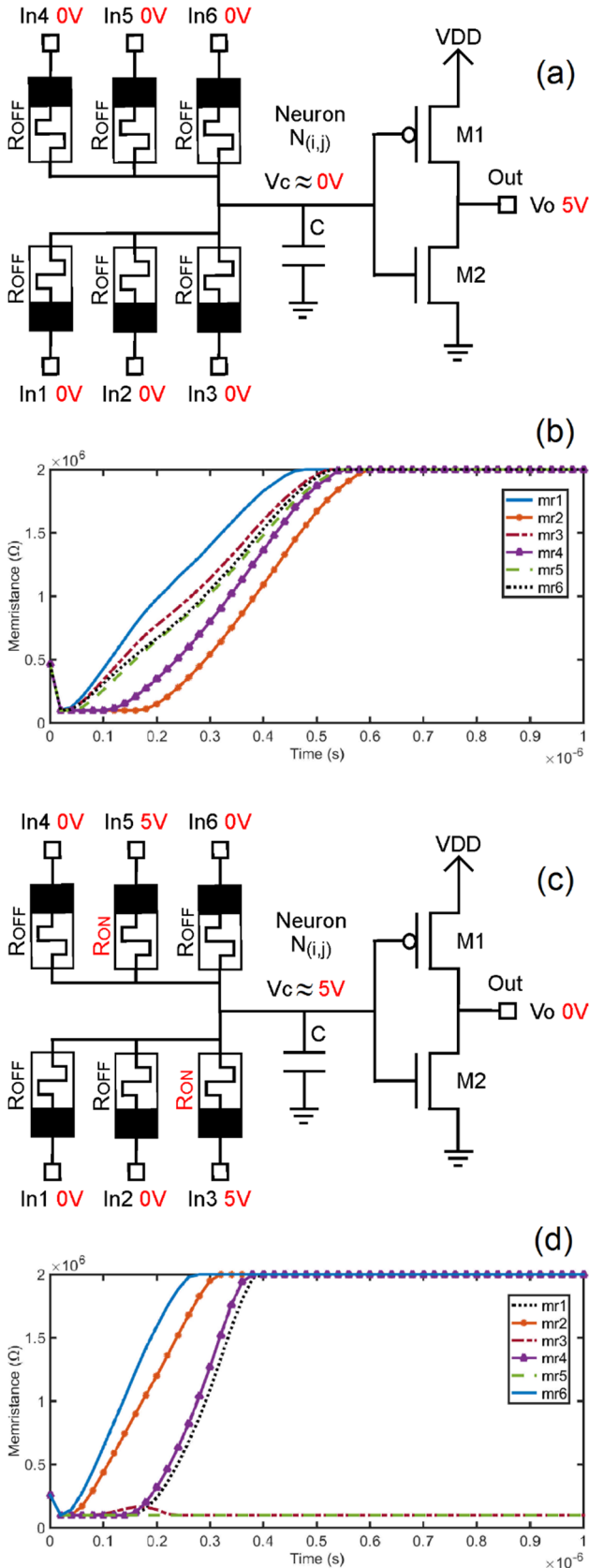


Fig. 8 Schematic diagrams a) and c) for “Winner” and “Loser”, respectively and, electrical simulations b) and d) for “Winner” and “Loser”, respectively.

VI. CONCLUDING REMARKS

This memristive analog circuit, whose alternative configuration can be seen as an array of six-input NOR gates behaves as a competitive system. The memristors change their memristance from established initial states in favor of the extreme values of either R_{ON} or R_{OFF} and solving in parallel the four assignment tasks. In other words, it is solved a combinatorial optimization problem. In fact, the nonlinear differential equations of this circuit encode in their permanent regime the solution. The analytical task associated to these equations is left out of this paper. Considering the end of the transient regime, we observe: 1) for the “Winner” Neuron, all of its memristors get R_{OFF} and, 2) for the “Loser” Neuron, two of its memristors get R_{ON} and the rest R_{OFF} . We have used 4 numerical matrices F_1 , F_2 , F_3 , and F_4 as an experimental vehicle to demonstrate via electrical simulations in SPICE that the proposed memristive analog circuit is stable and finds the respective solution matrices P_1 , P_2 , P_3 , and P_4 . Following conservative design rules for the geometric sizes in M1 and M2 in the CMOS inverters, these transistors can provide the transient currents through the memristors capable to charge dynamically the input capacitive node of their neighbor connected CMOS inverters.

This memristive analog optimizer has demonstrated via electrical simulations its correct operation with processing time lower than $1.0 \mu s$. Finding other circuit configurations memristor/CMOS for reducing the number of memristive elements is a pendent research issue.

REFERENCES

- [1] H. A. Taha, Operations research: an introduction. Pearson Prentice Hall, USA 2007.
- [2] L. O. Chua “Memristor: the missing circuit element,” IEEE Trans Circuit Theory, vol. 18(5), pp. 507–519, September 1971.
- [3] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams “The missing memristor found,” Nature, vol. 453(7191), pp. 80-83, May 2008.
- [4] S. Kvatinsky, E. G. Friedman, A. Kolodny, U. C. Weiser “TEAM: ThrEshold adaptive memristor model,” IEEE Trans. Circuits Syst, vol. 60(1), pp. 211-221, January 2013.
- [5] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. C. Weiser, E. G. Friedman “Models of memristors for SPICE simulations”, IEEEI 2012 27th Convention of Eliat, Israel. doi: 10.1109/EEEL.2012.6377081.
- [6] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, E. G. Friedman “MRL-memristor ratioed logic”, CNNA 2012 13th international workshop. Turin, Italy. doi: 10.1109/CNNA.2012.6331426 2013.

Memristor Hybrid Model for Nonlinear Analog Circuit Design

G.M. Tornez-Xavier¹, M.A. Gutiérrez-Mondragón¹, L.M. Flores-Nava¹,
F. Gómez-Castañeda¹, J.A. Moreno-Cadenas¹

¹Department of Electrical Engineering, CINVESTAV-IPN, Mexico City, Mexico
Phone (52) 55 5747 3800 Ext. 6261
E-mail: {gtornez, mario.gutierrez, lmflores, fgomez, jmoreno} @cinvestav.mx

Abstract — In this work we present the realization of a hybrid model (hardware-software) for memristors based on the nonlinear dopant model for three window functions: Strukov, Joglekar and Prodromakis. Proposing a methodology that allows us to design nonlinear analog circuits with memristors in an easier and faster way using the tools of Simulink, Simscape and System Generator. In addition, we tested the functionality of our hybrid model through the design of an oscillator without reactance, also verifying the effectiveness and scope of our proposed methodology.

Keywords — memristor, nonlinear ion drift model, windows functions, oscillators, FPGA.

I. INTRODUCTION

In 1971, Leon O. Chua [1] considered the existence of a new passive element of circuits of two terminals called memristor (memory and resistor). He observed that by combining in pairs the four fundamental circuit variables: voltage (v), current (i), magnetic flux (ϕ), and charge (q) six mathematical relationships were obtained, where the relationship between flux and charge was still unknown Fig. 1.

In 2008, Stanley Williams and his team of HP (Hewlett-Packard Laboratories) fabricated the memristor for the first time with the aim of creating nano switches whose resistance ranged from a minimum value R_{ON} to a maximum value R_{OFF} [2]. Since then the interest of the memristor in the field of non-volatile memories [3], programmable analog electronics [4] and neural networks [5] has drawn attention. For this reason there is a need to create models that reproduce their physical behavior for use in the creation of circuits and systems. There are also works in which the main interest is the creation of a memristor emulator in FPGA (Field Programmable Gate Array) [6].

This article focuses on the creation of a hybrid model of the HP model for three window functions: Strukov, Joglekar and Prodromakis implemented in a FPGA or programmable logic device using: Simulink, Simscape and System Generator working environment. Finally, the design of an analog oscillator without reactance is presented to demonstrate the effectiveness of our hybrid model.

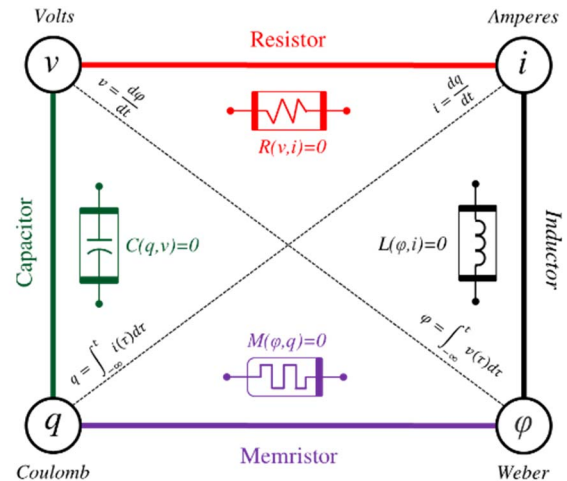


Fig. 1 Model of the relationship between the four fundamental variables in circuit theory.

A) Linear ion drift model

Fig. 2 shows the structure of the physical model of the memristor device consisting of two regions: a region doped with oxygen vacancies (TiO_{2-x}), which acts as a conductor and another region without doping (TiO_2) that acts as a dielectric. These regions are between two platinum contacts: the doped region and the non-doped region that can be seen as two variable resistors connected in series.

When a positive voltage is applied the oxygen vacancies of the TiO_{2-x} layer are repelled, moving towards the non-doped TiO_2 layer causing an increase in the conductivity throughout the device until reaching the R_{ON} state when $w = D$. When a negative voltage is applied the oxygen vacancies are attracted and pushed out of the TiO_{2-x} layer increasing the resistance of the whole device until it reaches the R_{OFF} state when $w = 0$. If the voltage of the device is suppressed the oxygen vacancies do not move causing the resistive value to remain intact and unalterable, so that the memristor is able to remember its last value of voltage or current applied. The thickness of the device is represented by D and w is the width of the doped region which is modulated by the amount of charge passing through the device.

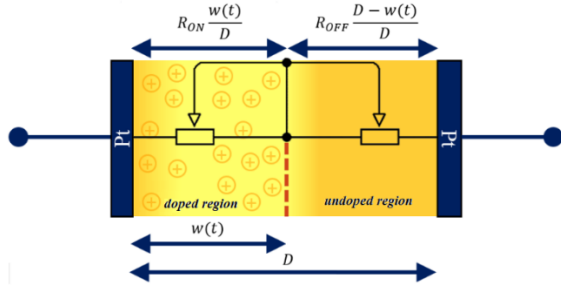


Fig. 2 Pt/TiO₂/Pt structure of HP[®] Memristor.

The total resistance of the memristor R_{MEM} is represented by (1) where x is the normalized width of the doped region.

$$R_{MEM}(x) = R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D}\right) \quad (1)$$

$$x = \frac{w}{D} \in [0,1] \quad (2)$$

The speed of the boundary between the doped and the non-doped region is represented by dx/dt where μ_v is the average ion mobility in $m^2V^{-1}s^{-1}$.

$$\frac{dx(t)}{dt} = \mu_v \frac{R_{ON}}{D^2} i(t) \quad (3)$$

And the current-voltage relationship of the memristor is established by Ohm's law.

$$v(t) = (R_{ON}x + R_{OFF}(1-x))i(t) \quad (4)$$

B) Nonlinear ion drift model

In this model, dx/dt varies in a non-linear way, because when a small voltage is applied to the memristor it produces a large electric field causing a non-linear ion drift. In addition, Strukov found that when the boundary between the doped and the non-doped region reaches either end of the device $w = 0$ or $w = D$ the boundary velocity is zero, producing non-linearities in the ionic transport [2]. To solve this problem, a window function $f(x)$ is added to the linear model.

$$\frac{dx}{dt} = ki(t)f(x) \quad (5)$$

$$k = \frac{\mu_v R_{ON}}{D^2} \quad (6)$$

C) Window Function

The window function adds a non-linear behavior at the ends of the device. There are several window functions that have been proposed. Strukov [2] proposed the first window function with the linear model.

$$f(x) = x - x^2 \quad (7)$$

Joglekar and Wolf [7] presented the following function where we have a positive control parameter p that controls the linearity of the model.

$$f(x) = 1 - (2x - 1)^{2p} \quad (8)$$

Prodromakis [8] has the following window function which has a scalability factor j that allows $f(x)$ to be less than or greater than 1 and p can be any positive real number.

$$f(x) = j(1 - [(x - 0.5)^2 + 0.75]^p) \quad (9)$$

II. METHODOLOGY

When it comes to the modeling of the memristor in the environment of Simulink, this cannot be treated as a physical element of two terminals since Simulink handles it as a black box with an input for the voltage and an output for the current so it cannot be used in the simulation of electrical systems. To solve this problem Simulink has a tool called Simscape that allows the simulation and modeling of physical systems in different engineering areas: electrical, hydraulic, mechanical etc.

Figure 3 shows the methodology addressed in [9] for the implementation of the nonlinear dopant drift model (5) using Simulink blocks and a potentiometer from the Simscape electrical element library, that physically represents the memristor. In this paper, we present a hybrid option where we work part of the design at hardware level and another at software level.

One way to address this problem is to use the System Generator development tool, which offers a function blocks library that can be implemented in an FPGA device. Because some memristor parameters are too large and others small is necessary to use floating point arithmetic (IEEE 754-2008 standard). Fig. 4 shows the complete diagram with all the blocks used for the realization of the nonlinear ion drift model with the described window functions.

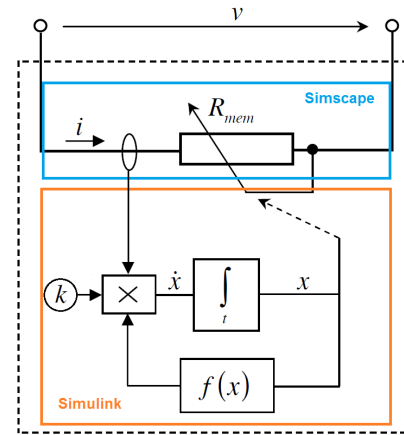


Fig. 3 Scheme of the memristor.

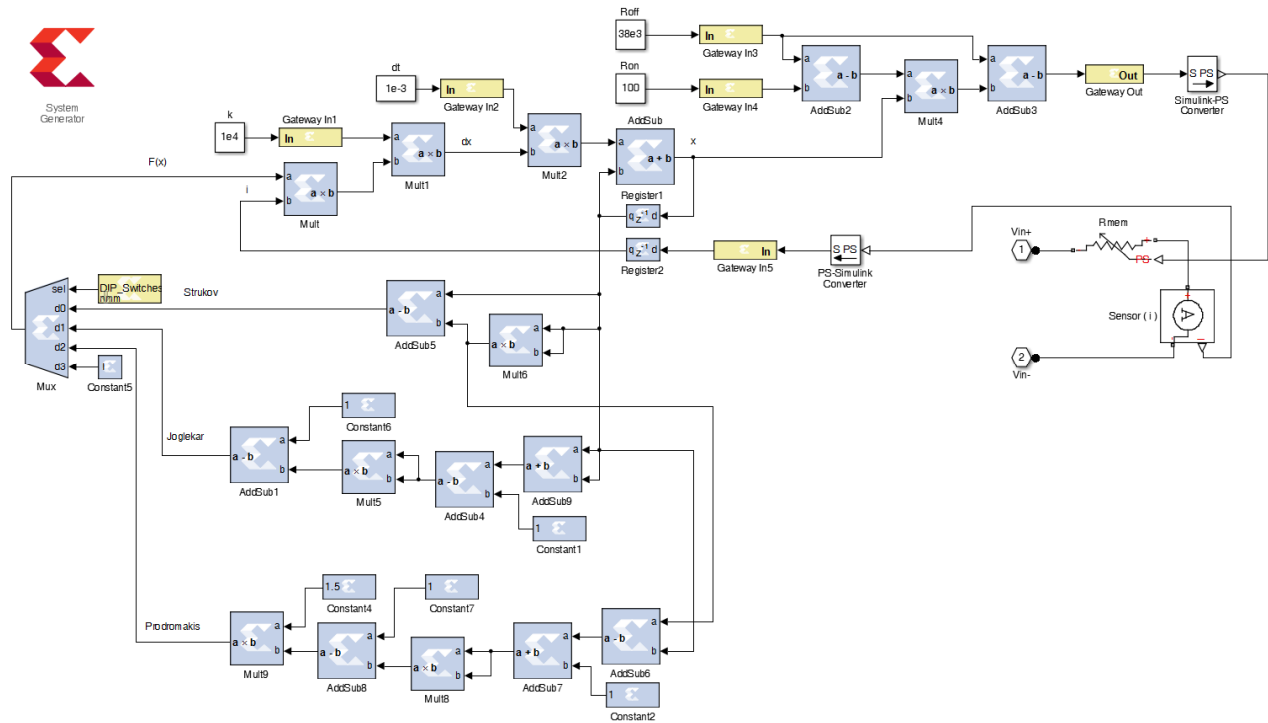


Fig. 4 Complete diagram of the nonlinear dopant model.

The method that allows us to implement the model built in Simulink at Hardware level is using a block called Hardware / Software Co-simulation (JTAG Co-sim). Fig. 5 shows the JTAG Co-sim block where the calculation of the window functions, the x state variable and the R_{MEM} memristance value is performed at hardware level, which is transferred to the two-terminal potentiometer of Simscape for having the voltage-current memristor ratio. A Spartan-6E FPGA was used and the logical resources used were: 4308 LUTs, 637 Registers, 40 DSPs and 2 RAMB16B.

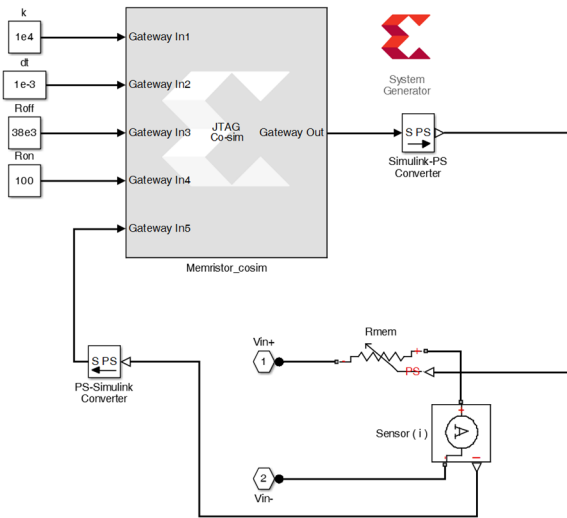


Fig. 5 Diagram of the hybrid model.

III. HYBRID MODEL TEST

To evaluate the functionality of our hybrid model of the memristor, the Fig. 6 shows the circuit assembled with a sinusoidal signal of 1.0V of peak and frequency of 0.3mHz.

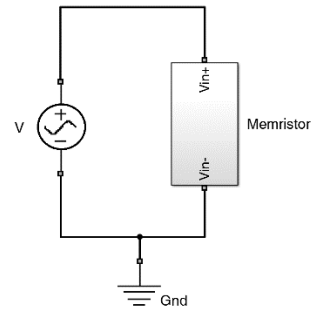


Fig. 6 Measuring circuit with the following parameters: $R_{ON}=100\Omega$, $R_{OFF}=38K\Omega$, $k=1.0 \times 10^4$, $x_{init}=0.3145$

In Fig. 7 the results of the simulation of the memristor are shown, where for the Strukov function we have a current $I_{mem} = 41\mu A$ and a variation from $22K\Omega$ to $26K\Omega$ for R_{mem} . In Joglekar there is a control parameter $p = 1$ where $I_{mem} = 116\mu A$ and the value of R_{mem} varies from $1.8K\Omega$ to $26K\Omega$. In Prodromakis we have the control parameters $p = 2$ and $j = 1.5$ where $I_{mem} = 67\mu A$ and R_{mem} varies from $6K\Omega$ to $26K\Omega$ observing that with Joglekar and Prodromakis a greater variation in the resistive value of the memristor can be obtained.

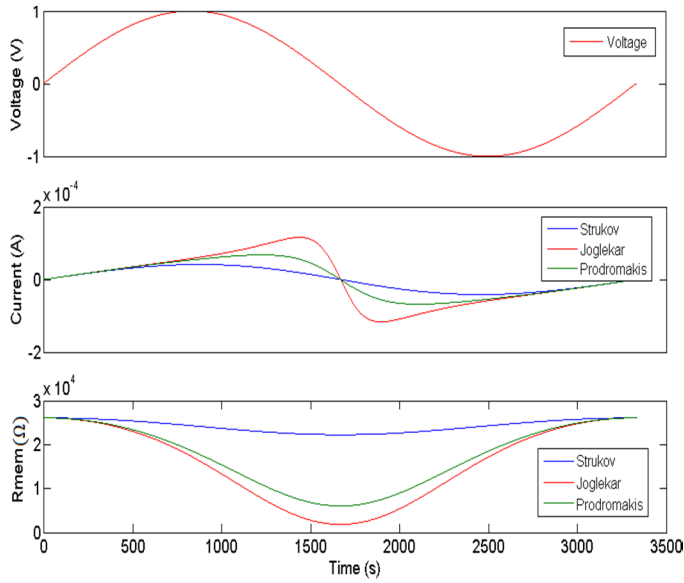


Fig. 7 Transient analysis.

Fig. 8, reproduces the voltage-current relationship of the memristor, the hysteresis function can be compared for all proposed window functions, observing that the waveforms are symmetrical with respect to the origin.

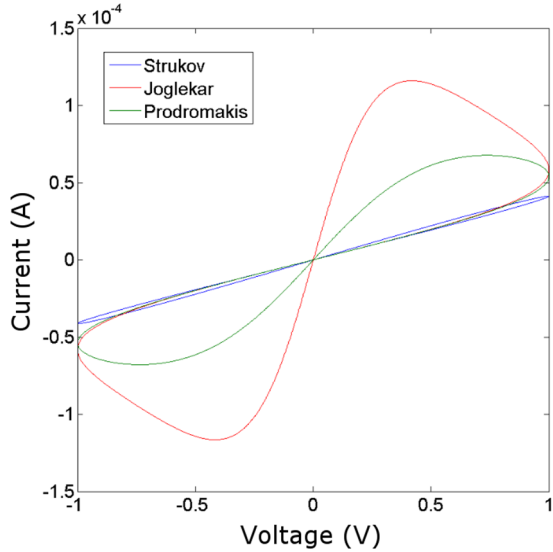


Fig. 8 Voltage-Current relationship of the memristor.

IV. MEMRISTOR-BASED APPLICATION

In this article we will perform the analysis of an oscillator without a reactance based on a memristor [10]. According to Fig. 9, the circuit simulates the loading and unloading response of an RC circuit through a voltage divider formed from a resistor and a serial memristor. The control module can be developed with two comparators in parallel, where the output of each serves as input of an AND gate responsible for generating the oscillation.

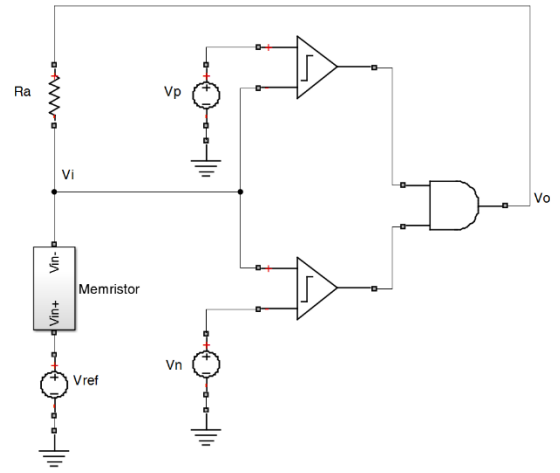


Fig. 9 Oscillator circuit without reactance.

The oscillation function is performed based on the voltage at the V_i node, however, to facilitate the analysis the voltage in the memristor is considered, which can be described as:

$$V_M(t) = (V_o(t) - V_{ref}) \frac{R_m}{R_m + R_a} \quad (10)$$

The variation of the memristance R_M is defined based on the polarization voltages of the comparators: V_p and V_n , presenting a maximum resistance value in R_{Mp} and minimum in R_{Mn} . These values can be determined from (10) as:

$$R_{Mp} = R_a \frac{V_p - V_{ref}}{V_{oh} - V_p}; \quad R_{Mn} = R_a \frac{V_n - V_{ref}}{V_{ol} - V_n} \quad (11)$$

The polarization voltages in the comparators V_p and V_n prevent R_M from acquiring a constant value, due the border of the regions of the memristor reached some of the ends. In this way, R_{Mp} and R_{Mn} should be defined such that the border does not reach the saturation.

$$R_{on} < R_{Mn} < R_{Mp} < R_{off} \quad (12)$$

According to the value of R_a , the necessary condition for correct circuit oscillation can be obtained by substituting (11) into (12)

$$R_{on} \left(\frac{V_{ol} - V_n}{V_n - V_{ref}} \right) < R_a < R_{off} \left(\frac{V_{oh} - V_p}{V_p - V_{ref}} \right) \quad (13)$$

Regarding the resistance variation in the memristor, the HP model is considered, where the movement of the boundary between the doped and the non-doped region is described, using the state equation.

$$\frac{dx(t)}{dt} = k i(t) f(x) \rightarrow k = \frac{\mu_v R_{on}}{D^2} \quad (14)$$

For the analysis of the oscillator the function of window of Joglekar was used with $p = 1$, since, to a greater value in p , the function becomes linear, presenting an undesired effect in

the analysis of the circuit. To facilitate the development of expressions, the window function can be described as:

$$f(x) = 4x(1-x) \quad (15)$$

The resistance of the memristor is defined according to the sum of the resistances in series with respect to the doped and non-doped region $R_{MEM}(x) = R_{ON}x + R_{OFF}(1-x)$ which can be defined as:

$$R_{MEM}(x) = R_{OFF} - x(R_{OFF} - R_{ON}) \quad (16)$$

The current flowing through the memristor can be determined analytically from the diagram of Fig. 9.

$$i_m(t) = \frac{V_o(t) - V_{ref}}{R_a + R_{MEM}} \quad (17)$$

Using (17) and the window function of Joglekar, it is possible to obtain a version of the equation of state (18) which will allow to obtain an expression for the calculation of the frequency.

$$\frac{dx}{dt} = \frac{k(V_o(t) - V_{ref})(4x(1-x))}{R_a + R_{OFF} - x(R_{OFF} - R_{ON})} \quad (18)$$

With (18), it is possible to obtain the time the output lasts in high state T_H and in low state T_L of the oscillator. Thus, integrating from x_{Mp} to x_{Mn} and as a function of time T_H , considering $V_o(t) = V_{OH}$ we have:

$$\frac{1}{4k(V_{OH} - V_{ref})} \int_{x_{Mn}}^{x_{Mp}} \left(\frac{R_{OFF} + R_a}{x} + \frac{R_{on} + R_a}{1-x} \right) dx = \int_0^{T_H} dt \quad (19)$$

From this expression, x_{Mp} and x_{Mn} correspond to the value of the boundary when the resistance value is R_{Mp} and R_{Mn} , respectively. These variables are determined through the equation of the resistance of the memristor (16), clearing the state variable, resulting in:

$$x_{Mp} = \frac{R_{Mp} - R_{off}}{R_{off} - R_{on}} \quad x_{Mn} = \frac{R_{Mn} - R_{off}}{R_{off} - R_{on}} \quad (20)$$

By solving the integral (19) and making the corresponding substitutions, it is possible to obtain an equation of the positive half-cycle of the signal T_H .

$$T_H = \frac{(R_a + R_{off}) \ln \left(\frac{R_{Mn} - R_{off}}{R_{Mp} - R_{off}} \right) + (R_{on} + R_a) \ln \left(\frac{R_{on} - R_{Mp}}{R_{on} - R_{Mn}} \right)}{4k(V_{OH} - V_{ref})} \quad (21)$$

For the negative half-cycle of the signal, the state equation (18) is integrated from x_{Mn} to x_{Mp} as a function of T_L and with $V_o(t) = V_{OL}$.

$$T_L = \frac{(R_a + R_{off}) \ln \left(\frac{R_{Mp} - R_{off}}{R_{Mn} - R_{off}} \right) + (R_{on} + R_a) \ln \left(\frac{R_{on} - R_{Mn}}{R_{on} - R_{Mp}} \right)}{4k(V_{OL} - V_{ref})} \quad (22)$$

Taking into account the ratio of the time in which the signal is active T_H and its period T , the expression for the oscillator duty cycle can be obtained as:

$$D = \frac{T_H}{T} \quad (23)$$

From (23) the period T is determined from the sum of the time in high and low of the signal ($T = T_H + T_L$). And with the reciprocal of the period, the working frequency of the oscillator can be determined as:

$$f = \frac{1}{T_H + T_L} \quad (24)$$

Based on the above expression, it can be noted that the oscillation frequency of the circuit can be controlled by different parameters. For example, by means of the reference voltage, the polarization values in the comparators, etc. But the easiest way to control the oscillation frequency is by means of the resistance R_a , placed in series with the memristor, since it is an easy and accessible parameter to adjust.

V. OSCILLATOR RESULTS

Using the developed memristor model, the following datas were used: $R_{ON} = 100\Omega$, $R_{OFF} = 38K\Omega$, $k = 1.0 \times 10^4$. And the circuit parameters $V_{OL} = 0V$, $V_{OH} = 2V$, $V_n = 0.5V$, $V_p = 1.75V$ and $V_{ref} = 1.0V$. Fig. 10 shows the co-simulation results considering $R_a = 3k\Omega$ and substituting in (11), the resistance variation expected in the memristor is $R_{Mn} = 3k\Omega$ and $R_{Mp} = 9K\Omega$.

In addition, the oscillation interval for the memristance and the voltage variation in the memristor can be noted, with the switching condition being fulfilled when the input voltage V_i is equal to V_p or V_n . Finally, the output signal with a 50% circuit duty cycle is also shown.

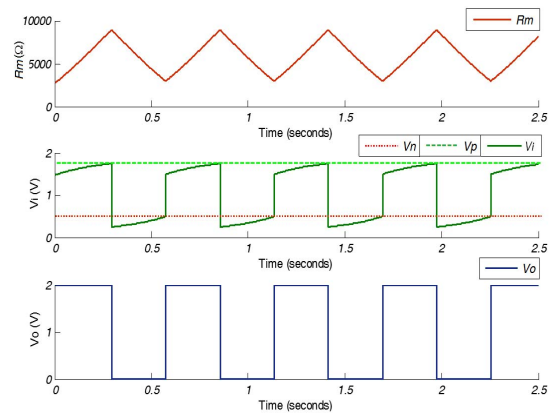


Fig. 10 Results of co-simulation with $R_a=3k\Omega$

As mentioned, the frequency of the circuit can be controlled through the resistor R_a , therefore, a sweep from $1\text{k}\Omega$ to $12\text{k}\Omega$ was performed to verify the presented analysis. In Fig. 11, a graph of the behavior of the frequency as a function of the variation of the resistance R_a is shown.

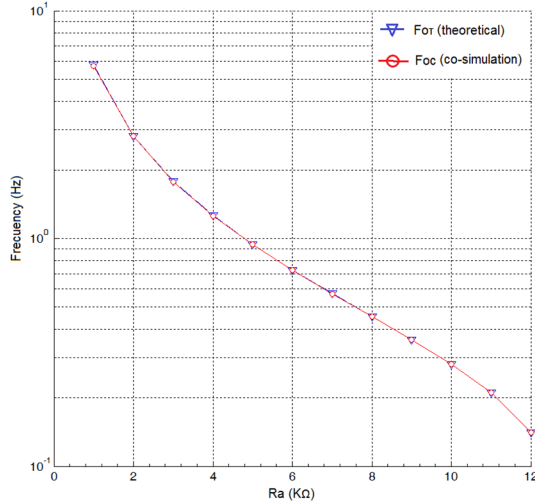


Fig. 11 Graph of the behavior of the frequency

Table 1 shows a comparison of the data obtained from the co-simulation and those obtained theoretically. In addition, the error rate calculated as $(\text{abs}(F_{OT} - F_{OC}) / F_{OT}) * 100$ is shown, showing a value below 1%. So, it can be mentioned that the deviation of the data obtained in the co-simulation with the expected ones through the mathematical expressions are acceptable.

Table 1. Comparison of theoretical frequency versus co-simulation.

Ra (kΩ)	F _{OT} (Hz) (theoretical)	F _{OC} (Hz) (co-simulation)	Error (%)
1	5.7899	5.7471	0.7392
2	2.8205	2.8089	0.4112
3	1.7879	1.7793	0.4810
4	1.2626	1.2594	0.2534
5	0.9436	0.9416	0.2119
6	0.7286	0.7278	0.1097
7	0.5730	0.5720	0.1745
8	0.4543	0.4537	0.1320
9	0.3595	0.3591	0.1112
10	0.2805	0.2803	0.0713
11	0.2108	0.2106	0.0948
12	0.1412	0.1412	0.0000

VI. CONCLUSIONS

In this work, the implementation of the nonlinear dopant model was performed in an FPGA, demonstrating its functionality reproducing the voltage-current relationship of the memristor. In addition, this work shows a different methodology in the modeling and simulation of the memristor from which four advantages can be observed: 1) Design time is shorter, since the high level design is used which avoids the use of the hardware programming language, 2) Easy communication between the FPGA and Simulink, because the asynchronous serial interface that is used is transparent to the user 3) Simulink blocks can be used to define excitation signals and observe the results obtained, 4) With this methodology the use of floating-point arithmetic in an FPGA is simple allowing to obtain values similar to the numerical model. Finally, it is observed the importance of having a hybrid model of the memristor in order to analyze and design nonlinear analog circuits in general.

REFERENCES

- [1] L.O. Chua, "Memristor-The missing circuit element," IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507-519, Sep. 1971.
- [2] D.B. Strukov, G.S. Snider, D.R. Stewart and R.S. Williams, "The missing memristor found," Nature, vol. 453, pp. 80-83, May. 2008.
- [3] S.H. Jo, K.H. Kim, T. Chang, S. Gaba and W. Lu, "Si memristive devices applied to memory and neuromorphic circuits". In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 13-16, Jun. 2010.
- [4] R. Berdan, T. Prodromakis and C. Toumazou, "High precision analogue memristor state tuning," Electron. Lett, vol. 48, no. 18, August. 2012.
- [5] A. Wu, Z. Zeng and J. Chen, "Analysis and design of winner-take-all behavior based on a novel memristive neural network," Neural Comput & Applic, vol. 24, pp. 1595-1600, 2012.
- [6] I. Yourkas, V. Ntinis, G. Sirakoulis and A. Rubio, "A digital memristor emulator for FPGA-Based artificial neural networks". *IEEE International Verification and Security Workshop (IVSW)*. Spain. doi: 10.1109/IVSW:2016.7566607
- [7] Y.N. Joglekar and S.J. Wolf, "The elusive memristor: Properties of basic electrical circuits," Eur. J. Phys, vol. 30, no. 4, pp. 661-675, Jul.2009.
- [8] T. Prodromakis, C. Papavassiliou, and C. Toumazou, "A versatile memristor model with nonlinear dopant kinetics," IEEE Trans. Electron Devices, vol. 58, no. 9, pp. 3099-3105, Sep.2011.
- [9] K. Zaplatilek, "Memristor modeling in Matlab & Simulink," In *Proceedings of the European Conference*. France, April 2011.
- [10] A.G. Radwan, M.E. Fouda, "Studies in Systems decision and control on the mathematical modeling of memristor, memcapacitor and memductor" Springer 2015.