



**Universitat de les
Illes Balears**

Facultad de Ciencias

Memoria del Trabajo de Fin de Grado

Determinación de parámetros de modelos de neuronas

Yasmina Marín Mas

Grado de Física

Año académico 2015-16

DNI del alumno: 43210014Y

Trabajo tutelado por Raúl Toral Garcés
Departamento de Física & Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC)

Se autoriza a la Universidad a incluir este trabajo en el Repositorio Institucional para su consulta en acceso abierto y difusión en línea, con fines exclusivamente académicos y de investigación	Autor		Tutor	
	Sí	No	Sí	No
	X		X	

Palabras clave del trabajo:
Determinación, parámetros, modelos, neuronas

ÍNDICE GENERAL

Índice general	i
1 Objetivos del trabajo	1
2 Introducción a modelos neuronales	3
2.1 La neurona, el origen y la propagación del potencial de acción	3
2.2 Diferentes modelos neuronales	6
2.2.1 Izhikevich	6
2.2.2 FitzHugh-Nagumo	7
2.2.3 Hindmarsh-Rose	8
3 Optimización de funciones y su dificultad	10
4 El método Temple Simulado	12
4.1 Descripción del algoritmo	12
4.2 Pasos del algoritmo	14
5 Aplicaciones	15
5.1 Comparación de parámetros de mismos modelos mediante barrido de puntos	15
5.2 Comparación de parámetros de mismos modelos mediante <i>Temple Simulado</i>	17
5.3 Comparación de parámetros de diferentes modelos mediante <i>Temple Simulado</i>	19
5.3.1 Régimen de tren de pulsos regulares	20
5.4 Ajuste de los parámetros de un modelo neuronal a una salida experimental dada	25
6 Resultados	27
Bibliografía	29

OBJETIVOS DEL TRABAJO

La complejidad computacional en los modelos neuronales dificulta la determinación de los parámetros de dichos modelos. El objetivo inicial de este trabajo se basaba en reproducir los datos de una salida neuronal experimental mediante el ajuste a las predicciones de un modelo, ya conocido y con un conjunto dado de parámetros a determinar. En un primer estudio, se decidió atacar el problema desde una perspectiva mucho más simple, para luego ir aumentando el nivel de dificultad de forma que nos permita resolver con uno de los métodos más efectivos de los que se dispone, el método *Temple Simulado*, el problema de encontrar los valores de un vector N-dimensional $\vec{a} = (a_1, a_2, \dots, a_N)$ que minimice una función de error real $E(\vec{a})$ escapando de los mínimos locales.

Así pues, se modificó el objetivo del trabajo para incluir primero la reproducción de una serie numérica, generada a partir de las ecuaciones de un modelo neuronal con los parámetros ya conocidos, mediante la comparación de ésta con la salida del mismo o diferente modelo neuronal, con los parámetros desconocidos, para devolver así, mediante la optimización de la función de error, dichos parámetros y generarla con éxito.

Es solo al final del trabajo cuando volvemos brevemente al objetivo inicial y consideramos el ajuste de parámetros para la reproducción de una señal experimental.

El problema se plantea de la manera siguiente:

Dada una serie temporal $\vec{x}(t)$, generada mediante la resolución numérica de las ecuaciones diferenciales de un modelo concreto

$$\frac{d\vec{x}(t)}{dt} = f_1(\vec{x}(t), \vec{b}), \quad (1.1)$$

debemos encontrar el conjunto de parámetros \vec{a} para que $\vec{z}(t, \vec{a})$, solución de la ecuación diferencial

$$\frac{d\vec{z}(t)}{dt} = f(\vec{z}(t), \vec{a}) \quad (1.2)$$

se acerque lo máximo posible a $\vec{x}(t)$. Si $f_1 = f$ la solución óptima sería $\vec{a} = \vec{b}$ dando lugar a

$$\vec{z}(t) = \vec{x}(t). \quad (1.3)$$

En el caso de que $f \neq f_1$ no se conoce de antemano el valor óptimo del conjunto de parámetros \vec{a} . Un criterio para encontrar dichos parámetros es la optimización de la función de error, normalizada respecto al tiempo, con forma

$$E(\vec{a}) = \frac{1}{T} \int_0^T (\vec{x}(t) - \vec{z}(t))^2 dt \geq 0 \quad (1.4)$$

para así encontrar el mínimo global de dicha función respecto a los parámetros del vector \vec{a} y comprobar que estos parámetros generan una aproximación $\vec{z}(t)$ suficientemente buena para la serie numérica $\vec{x}(t)$ deseada.

El núcleo de este trabajo radica en la aplicación de algoritmos de optimización a problemas de dinámica neuronal, los cuales necesitan dos componentes esenciales. La primera, la capacidad de la función de error de distinguir qué modelo neuronal reproduce mejor una serie numérica temporal dada, útil en caso de discrepancia de dicho criterio a simple vista, como veremos más adelante. La segunda componente es un algoritmo de búsqueda que explora el espacio de parámetros para encontrar, en un tiempo razonable, el conjunto de parámetros óptimo.

Las series numéricas a reproducir son las soluciones de las ecuaciones de los modelos neuronales Izhikevich, Fitz-Hugh Nagumo y Hindmarsh-Rose mediante dos tipos de algoritmos de búsqueda. El primero, basado en la comparación punto por punto de dos series numéricas con la forma de la ecuación (1.2), mediante un barrido amplio del espacio de parámetros en una región determinada del mismo. El segundo, un algoritmo de búsqueda más sofisticado, desarrollado mediante analogía con un problema de Física Estadística llamado *Temple Simulado*.

En resumen, en este trabajo se intentará reproducir los parámetros de un modelo neuronal dado por la ecuación (1.2) comparándolo con la salida de una serie numérica con la forma de la ecuación (1.1). En concreto, distinguiremos estos tres casos:

1. $f_1 = f$, haciendo la comparación de mismos modelos neuronales y verificar que $\vec{b} = \vec{a}$,
2. $f_1 \neq f$ y $\vec{b} \neq \vec{a}$, haciendo la comparación de diferentes modelos neuronales.
3. f_1 desconocida ya que la salida neuronal se obtiene a partir de datos experimentales.

Para la comparación de diferentes modelos neuronales, será necesario reescalar los dos ejes de coordenadas añadiendo un parámetro adicional, τ , en el eje temporal; además de otro parámetro, γ , proporcional a la corriente externa aplicada al primer modelo que permita la pulsación en el segundo modelo. Se discutirá la utilidad de los distintos métodos y la comparación de los errores obtenidos junto con sus limitaciones.

INTRODUCCIÓN A MODELOS NEURONALES

2.1 La neurona, el origen y la propagación del potencial de acción

Nuestro cerebro es una complicada red de unos cien mil millones de neuronas (10^{11}). La neurona es la célula principal del sistema nervioso, con una forma bastante diferente del resto de las células con forma de esfera o de disco. El soma es el cuerpo principal de la neurona de donde se extiende un largo cable llamado axón. La información se transmite utilizando señales eléctricas, que se propagan a lo largo del axón. Esta señal se conoce como impulso nervioso, o potencial de acción. Debido a la capacidad de cambiar su potencial y transmitirlo, las neuronas forman parte de los denominados sistemas excitables recibiendo estos impulsos en la sinapsis, que es la conexión entre neuronas. Luego, la información es transmitida desde una dendrita a un axón y así sucesivamente.

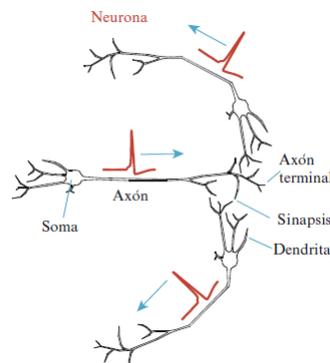


Figura 2.1: Red neuronal formada por tres neuronas que intercambian potenciales de acción la una con la otra. Imagen de ([1])

El papel principal lo lleva a cabo el potencial de membrana o potencial de acción, que es la diferencia de potencial entre los lados de la membrana celular que separa dos soluciones de diferente concentración iónica, $V = V_1 - V_2 \sim \frac{kT}{q} \log \frac{c_1}{c_2}$. Aquí k es la constante de Boltzman, T la temperatura absoluta, c_1 la concentración iónica en la parte intracelular de la membrana, c_2 la concentración iónica en la parte extracelular de la membrana y q la carga del electrón. El término $\frac{kT}{q} \log \frac{c_1}{c_2}$ es proporcional al flujo neto de iones debido a la diferencia de concentraciones entre los dos lados de la membrana. Por ejemplo, la concentración de iones K^+ es alta en la región intracelular y baja en la región extracelular. Al contrario, la concentración de iones Na^+ y Cl^- es alta y baja en las regiones extra e intracelular, respectivamente. Normalmente, la resisten-

cia de la membrana es muy alta de manera que ésta actúa como aislante para el movimiento de estos iones.

Por otra parte, la membrana presenta unos agujeros llamados canales iónicos que permiten el flujo de iones en el interior y exterior de la membrana. Estos canales son selectivos para cada ión, es decir, cada canal sólo permite el paso de un ión particular. La apertura y cierre de dichos canales son controlados por sus respectivas puertas, activadas por la sensibilidad que presentan hacia el potencial de membrana y otros factores.

En reposo, el potencial de acción es negativo, debido a la distribución desigual de concentración de iones, con los canales de K^+ abiertos y los de Na^+ cerrados, con un valor típico de alrededor de -70 mV para todas las neuronas. Según el tipo de estímulo que recibe la neurona, puede ocurrir una hiperpolarización o despolarización de la membrana. El primer caso, corresponde a un aumento de la diferencia de potencial, resultante de que aumente la densidad de iones con carga negativa dentro de la célula. Al contrario, la despolarización indica una disminución de la densidad de iones con carga negativa. Si el estímulo eléctrico supera un valor umbral, la despolarización mediante la apertura de algunos canales Na^+ dispara el potencial de acción, dando lugar a una señal eléctrica en forma de pulso que se propaga por el axón de la neurona. Una vez desarrollado el potencial de acción, éste cambia las propiedades de polarización de las zonas adyacentes, desplazándose a lo largo de la neurona. El retorno al potencial de reposo se debe a la apertura de los canales K^+ y al cierre de los canales Na^+ , repolarizando e hiperpolarizando la membrana. Durante el periodo refractario (definido como el tiempo de relajación del potencial de acción en el que la célula excitable no responde a ningún estímulo no puede producirse un segundo potencial de acción, aunque le lleguen significantes estímulos a la neurona.

La figura (2.2), de elaboración propia con uno de los modelos estudiados, representa el origen y la propagación de un potencial de acción con sus respectivas fases.

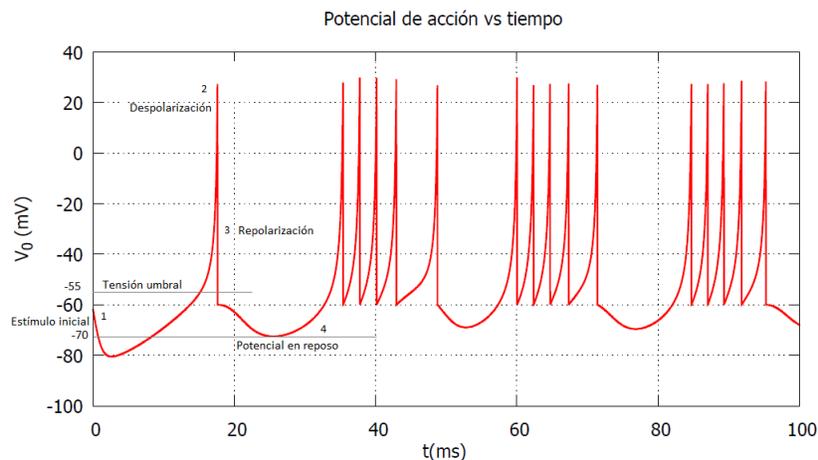


Figura 2.2: Fases del origen y la propagación del potencial de acción para el modelo de Izhikevich. 1: Estímulo inicial o pulso de entrada. 2: Despolarización de la membrana una vez alcanzado la tensión umbral. 3: Repolarización de la membrana. 4: Retorno al potencial de reposo.

El modelo de Hodgkin-Huxley ([1], [2]) (premio Nobel en 1963) describe la membrana celular como

un simple circuito eléctrico, mostrado en la figura (2.3). Los condensadores, con una cierta capacitancia, denotan las propiedades capacitivas de la membrana celular. Los canales de sodio y potasio son modelados por las resistencias con sus respectivas conductancias, g_{Na} y g_K . La tendencia de la fluctuación de iones hacia el interior y exterior de la membrana celular debido a la diferencia de concentración se denota por las baterías con voltajes E_{Na} y E_K en el circuito. Estos voltajes dependen de la diferencia de concentración en el interior y exterior de la membrana para cada ión.

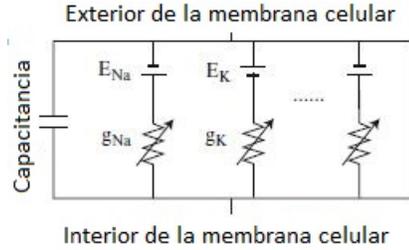


Figura 2.3: Circuito equivalente de la membrana celular y canales iónicos descrita por Hodgkin-Huxley. Imagen modificada de [1]

De esta manera, modelaron el potencial de acción mediante el sistema de las siguientes ecuaciones diferenciales no lineales que describen la evolución temporal en el circuito de la figura (2.3):

$$C \frac{dv}{dt} = -\bar{g}_{Na} m^3 h (v - E_{Na}) - \bar{g}_K n^4 (v - E_K) - \bar{g}_L (v - E_L) + I_{ext}(t), \quad (2.1)$$

$$\frac{dm}{dt} = \alpha_m(v)(1 - m) - \beta_m(v)m, \quad (2.2)$$

$$\frac{dn}{dt} = \alpha_n(v)(1 - n) - \beta_n(v)n, \quad (2.3)$$

$$\frac{dh}{dt} = \alpha_h(v)(1 - h) - \beta_h(v)h. \quad (2.4)$$

$$E_{Na} = 115 \text{ mV}, \quad E_K = -12 \text{ mV}, \quad E_L = 10,6 \text{ mV}$$

$$\bar{g}_{Na} = 120 \text{ ms/cm}^2, \quad \bar{g}_K = 36 \text{ ms/cm}^2, \quad \bar{g}_L = 0,3 \text{ ms/cm}^2$$

$$\alpha_n(v) = \frac{0,1 - 0,01v}{\exp(1 - 0,1v) - 1}, \quad \alpha_m(v) = \frac{2,5 - 0,01v}{\exp(2,5 - 0,1v) - 1}, \quad \alpha_h(v) = 0,07 \exp\left(-\frac{v}{20}\right)$$

$$\beta_n(v) = 0,125 \exp\left(-\frac{v}{80}\right), \quad \beta_m(v) = 4 \exp\left(-\frac{v}{18}\right), \quad \beta_h(v) = \frac{1}{\exp(3 - 0,1v) + 1}$$

donde $v(t)$ es el potencial de membrana, \bar{g}_{Na} , \bar{g}_K y \bar{g}_L representan la máxima conductancia en los respectivos canales. Las variables adimensionales n , m y h toman valores entre 0 y 1, denotando así la probabilidad de obertura de un canal. De esta manera, n y m se usan para controlar las puertas de los canales de Na^+ , mientras que h controla la obertura de las puertas de los canales de K^+ . El término L denota cualquier residuo a través de la membrana celular no perteneciente a los iones de Na^+ ni K^+ . La velocidad del cambio de estado de cierre a obertura, o viceversa, de las puertas se representa mediante las α 's y β 's, respectivamente. I_{ext} representa la corriente externa aplicada a la neurona. Los valores de los parámetros se eligieron para reproducir de la mejor manera posible los valores del potencial de acción medidos en el axón de un calamar gigante.

2.2 Diferentes modelos neuronales

La complejidad del modelo de Hodgkin-Huxley ha hecho buscar diferentes modelos más sencillos basados, ya sea en la simplificación de éste, o bien, describiendo los comportamientos de las corrientes iónicas desde una perspectiva más simple.

En esta sección se incluye una introducción a los modelos neuronales utilizados en nuestro trabajo. Se describirán sus respectivas formulaciones matemáticas (ecuaciones diferenciales ordinarias), sus parámetros y el significado de éstos.

2.2.1 Izhikevich

Mediante el uso de metodologías de bifurcación de sistemas dinámicos, Izhikevich (IZ) ([3]) propuso un modelo neuronal basado en un sistema bidimensional de ecuaciones diferenciales ordinarias, dadas por:

$$\frac{dv(t)}{dt} = \alpha v^2 + \beta v + \gamma - u + I_{ext}(t), \quad (2.5)$$

$$\frac{du(t)}{dt} = a(bv - u) \quad (2.6)$$

con los parámetros fijados mediante los valores,

$$\alpha = 0,04 \text{ mV}^{-1} \text{ ms}^{-1},$$

$$\beta = 5 \text{ ms}^{-1},$$

$$\gamma = 140 \text{ mV ms}^{-1}$$

y condiciones iniciales v_0 y u_0 .

La variable v representa el potencial de membrana de una neurona y u representa una variable de ajuste para la repolarización de la membrana, ambas tienen dimensión de voltaje. $I_{ext}(t)$ representa la corriente de entrada, a, c y d son parámetros dimensionales y b un parámetro adimensional. Cuando el potencial alcanza su punto máximo (30 mV), se emite un pulso, de manera que tanto el potencial de membrana como la variable de recuperación son reiniciados de acuerdo con

$$\text{si } v \geq 30 \text{ mV} \begin{cases} v = c \\ u = u + d \end{cases} \quad (2.7)$$

La parte $\alpha v^2 + \beta v + \gamma$ se obtuvo mediante el ajuste del comienzo del pulso en una neurona cortical, de este modo v está medido en mV y t en ms . El potencial en reposo de este modelo oscila entre los -70 y -60 mV dependiendo del valor de b .

- El parámetro a representa la escala de tiempo de la variable u . Cuanto más bajo sea el valor, más lenta será la recuperación. Un valor típico es $a = 0,02 \text{ ms}^{-1}$
- El parámetro b representa la sensibilidad de la variable u a las fluctuaciones del potencial de membrana v por debajo del umbral. Cuanto mayores sean u y v las posibles oscilaciones surgirán con más fuerza. Un valor típico es $b = 0,2$.
- El parámetro c representa el valor de reajuste pos-pico para la variable del potencial de membrana v . Un valor típico es $c = -65 \text{ mV}$.

- El parámetro d representa el valor de reajuste pos-pico para la variable de recuperación u . Un valor típico es $d = 2mV$.

A diferencia de otros modelos, éste puede reproducir hasta 20 patrones distintos de disparos neuronales [4] dependiendo del valor de los parámetros a , b , c y d . Por ejemplo, puede presentar disparos en forma de picos, ráfagas o ambos con una frecuencia de adaptación en el primer disparo como muestra la figura (2.4).

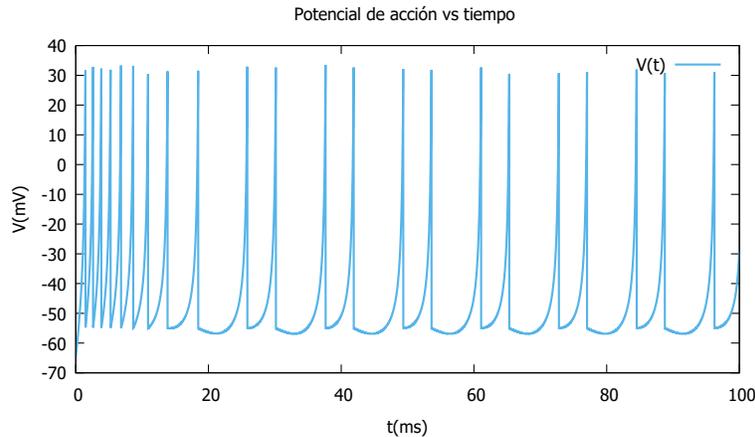


Figura 2.4: Salida neuronal del modelo (IZ) en función del tiempo. La neurona dispara con una ráfaga de picos *phasic bursting* seguido por una secuencia de picos individuales. Se caracteriza por un alto voltaje de restablecimiento ($c = -55mV$) y un gran salto después de alcanzar el pico de u , $d = 4mV$. Corriente de entrada constante a $I_{ext} = 40mA$, $a = 0,02ms^{-1}$ y $b = 0,2$

2.2.2 FitzHugh-Nagumo

El modelo de FitzHugh-Nagumo (FN) [5] es otra aproximación en dos dimensiones del modelo de Hodgkin-Huxley que surgió de la motivación de separar las propiedades matemáticas de la excitación y propagación de las propiedades electroquímicas del flujo de iones potasio y sodio. El modelo elimina adiabáticamente las puertas h y m y únicamente conserva una variable similar a n , indicada por w . Así pues, su sistema de ecuaciones diferenciales presenta la forma:

$$\frac{dv(t)}{dt} = v - \frac{v^3}{3} - w + I_{ext}(t) \quad (2.8)$$

$$\frac{dw(t)}{dt} = \epsilon(v + a - bw) \quad (2.9)$$

La variable v representa el potencial de membrana y w es una variable de recuperación lenta esencial para la repolarización, ambas adimensionales. I_{ext} representa la corriente de excitación. a , b y ϵ son parámetros adimensionales y t el tiempo. La constante ϵ suele ser muy pequeña ($\epsilon \ll 1$), haciendo que la relajación de v sea muy rápida.

La falta de reinicialización pos-pico prohíbe la existencia de salidas en forma de ráfagas, sólo será posible con la presencia de ruido o con otros modelos 2-D. La figura (2.5) muestra la trayectoria de la

componente v en función del tiempo con valores típicos de los parámetros y una corriente externa que provoca la generación de múltiples potenciales de acción.

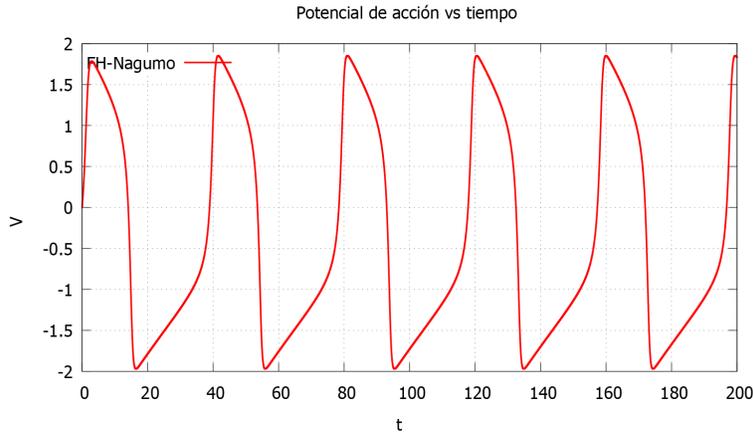


Figura 2.5: Salida neuronal del modelo (FN) en función del tiempo. La neurona dispara con picos individuales. Corriente de excitación a $I_{ext} = 0,5$. Valor de los parámetros: $\epsilon = 0,08$, $a = 0,7$, $b = 0,8$.

2.2.3 Hindmarsh–Rose

El modelo de Hindmarsh-Rose (HR) [6] es otra aproximación en tres dimensiones del modelo de Hodgking-Huzley descrito como un sistema de tres ecuaciones diferenciales no lineales:

$$\begin{aligned}\frac{dx(t)}{dt} &= y - ax^3 + bx^2 - z + I_{ext} \\ \frac{dy(t)}{dt} &= c - dx^2 - y \\ \frac{dz(t)}{dt} &= r(s(x - x_1) - z)\end{aligned}\tag{2.10}$$

donde la variable $x(t)$ es una variable proporcional al potencial de membrana $v(t)$, la variable $y(t)$ describe un conjunto de canales de iones rápidos por donde se realiza el transporte de iones de sodio y de potasio, la variable $z(t)$ representa la dinámica de otros canales más lentos por donde se realiza el transporte de otros iones; todas ellas adimensionales. El modelo tiene siete parámetros: a, b, c, d, r, s, x_1 ; los cuatro primeros modelan el funcionamiento de los canales iónicos rápidos, r es la escala de tiempo que modela el funcionamiento de los canales iónicos lentos, s es la escala de influencia del potencial de membrana en la dinámica lenta, x_1 es un potencial de referencia. I_{ext} es la proporcional corriente externa aplicada a la neurona.

La figura (2.6) muestra la trayectoria de la componente x en función del tiempo con valores típicos de los parámetros que generan un muestra neuronal típica en forma de ráfagas.

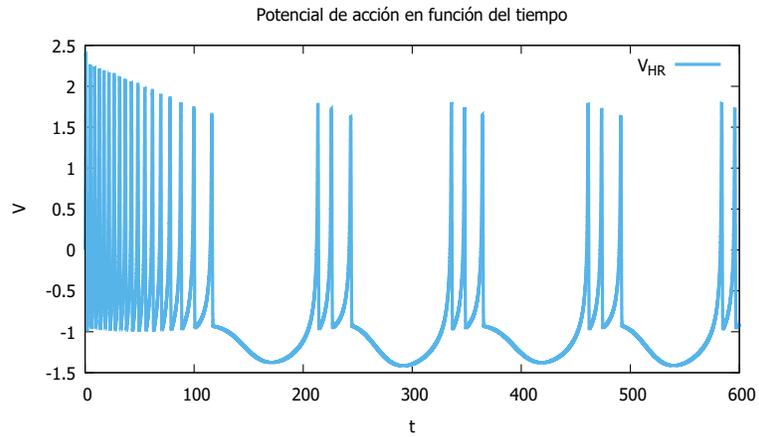


Figura 2.6: Salida neuronal del modelo (HR) en función del tiempo. La neurona dispara en forma de ráfagas. Corriente de excitación a $I_{ext} = 2,6$. Valor de los parámetros: $a = 1$, $b = 3$, $c = 1$, $d = 5$, $r = 0,006$, $s = 4$, $x_1 = -1,6$.

OPTIMIZACIÓN DE FUNCIONES Y SU DIFICULTAD

Imaginemos que tenemos dos salidas neuronales de modelos diferentes determinados con los conjuntos de parámetros \vec{a}_1 y \vec{a}_2 y queremos compararlos con una serie temporal obtenida experimentalmente o mediante resolución numérica de un modelo. Para una mayor dificultad, imaginemos que la salida del primer modelo pulsa al mismo tiempo que la experimental pero con diferente amplitud o forma del pulso; mientras que la salida del segundo modelo tiene la misma amplitud y forma pero no pulsa en todas las ocasiones en las que lo hace la señal experimental. A primera vista, ¿cuál se asemeja mejor a nuestros resultados experimentales? En este caso, podríamos discrepar fácilmente en nuestras respuestas ya que ambos modelos son muy semejantes entre sí. Para cuantificar la calidad del ajuste de un modelo teórico a unos datos dados, es necesario definir una función de error o función de adaptación que nos mida la diferencia entre la salida de un modelo con unos parámetros dados y nuestra salida experimental. Adoptamos el criterio de que el modelo que se acerque más a la reproducción de la salida experimental será el que tenga un valor de la función error más pequeño. Matemáticamente y adaptado a nuestras series temporales, queremos obtener la minimización

$$E(\vec{a}) = \frac{1}{T} \int_0^T (\vec{x}(t) - \vec{z}(t, \vec{a}))^2 dt \geq 0 \quad (3.1)$$

donde \vec{a} es un vector N-dimensional, T corresponde al tiempo de integración, $\vec{x}(t)$ una serie temporal conocida, obtenida a partir de los datos experimentales o numéricos y $\vec{z}(t, \vec{a})$ una serie temporal dependiente del conjunto de parámetros \vec{a} de las salidas de los dos modelos que queremos comparar.

Una vez definido el criterio de comparación de modelos, necesitamos obtener un algoritmo que encuentre el mínimo de la función de error y, como consecuencia, determinar el mejor conjunto de parámetros que reproduce dicha serie temporal. La elección de un algoritmo de búsqueda en un problema de optimización no es tarea fácil y depende de muchos factores, siendo uno de los más importantes la estructura de la función $E(\vec{a})$ en el espacio N-dimensional de parámetros \vec{a} . Una propiedad importante de este espacio es la densidad de mínimos locales, es decir, puntos donde el gradiente local es cero o el punto con el valor más pequeño dentro de un intervalo finito, ver figura (3.1). El objetivo del algoritmo de optimización es escapar de todos esos mínimos locales y encontrar el mínimo global, el punto con el valor más pequeño de la función de error en todo el espacio accesible de soluciones. Otra característica a tener en cuenta es que cuanto mayor sea el número de parámetros a determinar aumentará la dificultad de encontrar el mínimo global correspondiente a un único conjunto de parámetros.

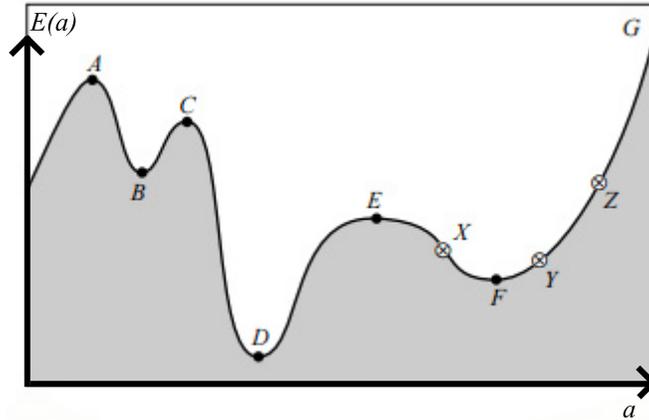


Figura 3.1: Extremos de una función unidimensional dentro de un intervalo. Los puntos A,C y E corresponden a máximos locales. Los puntos B y F corresponden a mínimos locales. El máximo y mínimo global corresponden a los puntos G y D, respectivamente. Los puntos X, Y y Z rodean el mínimo F, ya que Y es menor que X y Z. Imagen modificada de ([7])

Elegiremos dos métodos de búsqueda diferentes para lograr la optimización global de la función error. El primero, muy simple, basado en el recorrido de una parte significativa del espacio de parámetros y evaluando la función de error punto por punto para lograr la minimización de ésta. El segundo, más sofisticado está inspirado en el proceso de calentar un material y luego disminuir lentamente la temperatura, minimizando así la energía del sistema. Fluctuaciones aleatorias en la energía nos permiten escapar de los mínimos locales y lograr el mínimo global. Pero si se enfría muy rápidamente podríamos no escapar de los mínimos locales y al final del proceso obtener un mínimo metaestable de la energía.

Existen otros métodos de minimización como los siguientes:

- Método *Simplex*: definido como un objeto geométrico de $N + 1$ lados en el espacio N -dimensional de parámetros \vec{a} . El algoritmo genera iterativamente nuevos vértices del *simplex* que reemplazan a los actuales, y hacen que éste se comprima progresivamente entorno a un mínimo en el espacio de búsqueda. Este algoritmo se vuelve ineficiente ante una función de error que presenta varios mínimos locales que ocultan el mínimo global.
- Método *descenso por gradiente*: extrae información de la primera derivada para acelerar el proceso de minimización. La dirección de búsqueda se toma como menos el gradiente. Este método es ineficiente para la función de error a minimizar ya que no se dispone de una función bien definida, sino de una salida numérica de la simulación.
- Escalada o *Hill Climbing*: definido como un bucle que constantemente se desplaza en la dirección de un valor ascendente obtenido en base a la función de error. En cada iteración se busca minimizar la función de error por lo que se selecciona el mejor de los estados siguientes y se comprueba que es mejor o igual que el actual; si es así pasa a ser el nuevo estado actual. El algoritmo escalada con reinicio aleatorio permite avanzar en los casos donde el algoritmo se ha quedado atascado: mínimo local, planicie y riscos. En lugar de empezar otra vez al azar y quedar atrapado una y otra vez en mínimos locales sigue siendo más eficiente descender unos cuantos pasos y escapar de los mínimos locales como realiza *Temple Simulado*.

EL MÉTODO TEMPLE SIMULADO

4.1 Descripción del algoritmo

Temple simulado, ("Simulated Annealing") ([8], [9], [10]) es una técnica para resolver problemas de optimización combinatoria, como el problema de minimización de una función de muchas variables independientes. Este método, introducido en 1982 por Kirkpatrick et. al., fue motivado por una analogía a la mecánica estadística del enfriamiento de sólidos, ([11]). En física de la materia condensada, el término templado se refiere al proceso térmico de obtener un estado de baja energía de un sólido en un baño térmico. En la fase líquida, las partículas permanecen desordenadas mientras que en el estado fundamental de la fase sólida permanecen altamente ordenadas en una red y con una mínima energía. El objetivo de la simulación consiste en calentar un sólido hasta pasado su punto de fusión y luego enfriarlo de nuevo en un estado cristalino. La estructura final del sólido dependerá de cómo se realice el enfriamiento. Para obtener el estado fundamental del sólido es necesario que la temperatura inicial sea lo suficientemente alta y que se enfríe lentamente. De lo contrario, el sólido se congelará en un estado meta-estable en lugar del estado fundamental, no alcanzando así el mínimo global de energía.

El método *Temple Simulado* sigue una estrategia muy similar a los de mejora iterativa pero con la diferencia de que ocasionalmente acepta peores soluciones. De esta manera, permite que haya una serie de perturbaciones para poder alcanzar el máximo global. Gracias a que ahora cada movimiento puede transformar una solución en una peor solución, es posible escapar de los mínimos locales que podría haber si se hubiese quedado atascado sólo aceptando soluciones mejores. La simulación de este proceso es muy similar al del sistema físico. *Temple simulado* tiene como objetivo encontrar una configuración (distribución de partículas) que minimice el coste (energía) del sistema. Aunque el algoritmo no garantiza encontrar el mínimo global, se encontrará cerca de las soluciones óptimas con muchas menos evaluaciones que otros algoritmos.

El algoritmo de *Metropolis* genera una secuencia de estados del sólido de la siguiente manera. Dado un estado actual del sólido con una energía, generamos un nuevo estado aplicando un mecanismo de perturbación que transforme el actual estado en uno nuevo mediante una pequeña distorsión, por ejemplo el desplazamiento de una partícula. Luego, evaluamos el cambio de energía ΔE . Si se reduce la energía, $\Delta E < 0$, lo aceptamos como el estado actual. Si la diferencia de energía aumenta, $\Delta E > 0$, el nuevo estado con energía superior puede ser aceptado con una cierta probabilidad. En sistemas físicos, los saltos a energía superiores pueden ocurrir, pero éstos son moderados con la actual temperatura, T . A altas temperaturas, la probabilidad de aceptar estados con mayor energía es alta; mientras que a bajas

temperaturas la probabilidad es baja. El algoritmo de Metropolis modela dicha probabilidad mediante la distribución de Boltzman:

$$P = \exp\left(\frac{-\Delta E}{kT}\right), \quad \Delta E > 0 \quad (4.1)$$

donde k es la constante de Boltzman. En la práctica, la probabilidad de aceptación se compara con un número aleatorio entre 0 y 1 generado a partir de una distribución uniforme, $\hat{U}(0, 1)$. Si el número aleatorio es menor que la probabilidad de aceptación, se acepta el estado; sino se rechaza. De esta manera, muy pocos improbables movimientos serán aceptados.

Mientras en sistemas físicos, la temperatura tiene un significado físico; en métodos de optimización sin naturaleza física, la temperatura es simplemente un parámetro de control que modela un programa de enfriamiento mediante su disminución progresiva y así modera la cantidad de peores soluciones aceptadas.

La tabla (4.1) muestra el resumen de las diferencias entre los dos algoritmos descritos.

Metropolis	Temple Simulado
$P = \exp\left(\frac{-\Delta E}{kT}\right)$	$P = \exp\left(\frac{-\Delta C}{\Delta C_{promedio} T}\right)$
P: probabilidad que una partícula se mueva de un estado de más baja a energía a una más alta.	P: probabilidad que una configuración con un coste superior sea aceptado.
ΔE : diferencia de energía desde un estado con energía más baja a otro con energía más alta.	ΔC : diferencia del coste entre la configuración actual y la nueva.
T: temperatura absoluta relacionada con la densidad de movimiento de las partículas.	T: valor adimensional relacionado con la probabilidad de aceptación de una configuración con un coste superior.
k : constante de Boltzman	$\Delta C_{promedio}$: valor promedio del recorrido por ciclo de temperatura de todos los ΔC aceptados, normaliza ΔC .

Cuadro 4.1: Resumen comparativo del mecanismo entre los algoritmos *Metropolis* y *Temple Simulado*

Existen distintos métodos para reflejar el patrón de enfriamiento y de forma indirecta el progreso de la optimización. En este trabajo disminuimos la temperatura exponencialmente de acuerdo con la siguiente expresión

$$T_{i+1} = F \cdot T_i \quad (4.2)$$

siendo i el término i -ésimo. siendo F un factor definido como

$$F = \left(\frac{T_f}{T_i}\right)^{\frac{1}{k}} \quad (4.3)$$

donde T_f y T_i corresponden a la temperatura final e inicial, respectivamente. El parámetro k representa el número de ciclos de temperatura.

Existen otros múltiples mecanismos como el enfriamiento lineal o el enfriamiento logarítmico, aunque la exponencial es la más usual. En ([9]) podemos encontrar la discusión de estos patrones de disminución de la temperatura.

4.2 Pasos del algoritmo

- 1 Elegimos una configuración inicial de los valores de \vec{a} .
- 2 Seleccionamos el número de ciclos k , número de muestras dentro de un ciclo m , T_f , T_i . Calculamos E .
- 3 Perturbamos la configuración con valores discretos aleatorios con unos límites superior e inferior establecidos cercanos a la configuración actual de la manera siguiente,

$$\vec{a}' = \vec{a} + 0,1 \Delta\vec{a}(2\hat{U} - 1) \quad (4.4)$$

siendo $\Delta\vec{a} = (\vec{a}_{max} - \vec{a}_{min})$ la diferencia entre los valores máximos y mínimos de los parámetros y \hat{U} un número aleatorio entre 0 y 1 distribuido de manera uniforme.

- 4 Si la nueva configuración \vec{a}' es mejor, $E(\vec{a}') < E(\vec{a})$, se acepta como la configuración actual
- 5 Si la nueva configuración es peor, $E(\vec{a}') > E(\vec{a})$, se genera un número aleatorio ente 0 y 1 distribuido de manera uniforme, \hat{U} . Si el número aleatorio es menor que la probabilidad de aceptación, se acepta la configuración; sino se rechaza.
- 6 Continuamos perturbando la configuración actual aleatoriamente con la actual temperatura hasta realizar las m muestras.
- 7 Guardamos la mejor configuración dentro de cada ciclo. Disminuimos la temperatura un factor E .
- 8 Volvemos al paso 3
- 9 Continuamos el proceso hasta alcanzar T_f .

La figura (4.1) muestra los diagramas del algoritmo.

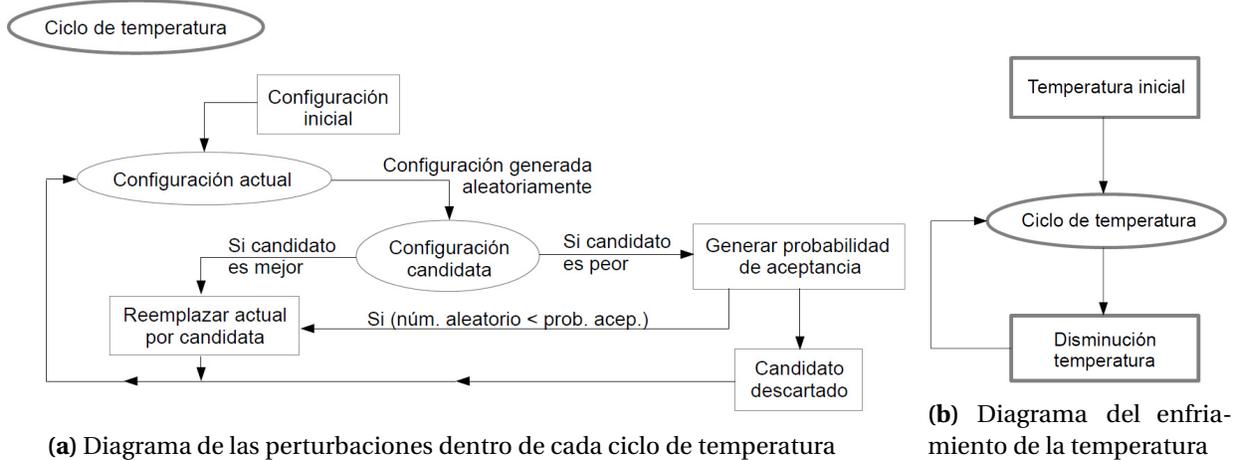


Figura 4.1: Diagramas del funcionamiento del algoritmo Temple Simulado

5.1 Comparación de parámetros de mismos modelos mediante barrido de puntos

Como test del algoritmo, en esta sección se quiere hallar los parámetros de una salida neuronal, es decir el potencial de membrana, del modelo de Izhikevich mediante la comparación con otra salida neuronal del mismo modelo con sus parámetros ya conocidos. Para agilizar la búsqueda nos centramos sólo en los parámetros a y b del modelo, quedando los parámetros c y d fijados. Primero, resolvemos las ecuaciones de dichos modelos mediante un método de resolución de EDO's. Para un mayor nivel de exactitud en la solución numérica, se escogió el método de Runge-Kutta de 4 orden ([12]), el cual nos facilitó las soluciones del sistema formado por las ecuaciones (2.5) y (2.6), dándonos el valor numérico de las variables u y v en un intervalo de tiempo determinado.

El método utilizado en este apartado es el de búsqueda sistemática. Consiste en un barrido del espacio 2-dimensional de los parámetros y así evaluar la función de error en una serie de puntos equidistantes en cada dimensión del espacio de parámetros creando una malla.

Para realizar nuestra aplicación a sistemas neuronales, es necesario adaptar la ecuación de la función de error (1.4) de tal manera que la podamos implementar en la búsqueda numérica de los parámetros y minimizarla. De este modo tendría la forma siguiente:

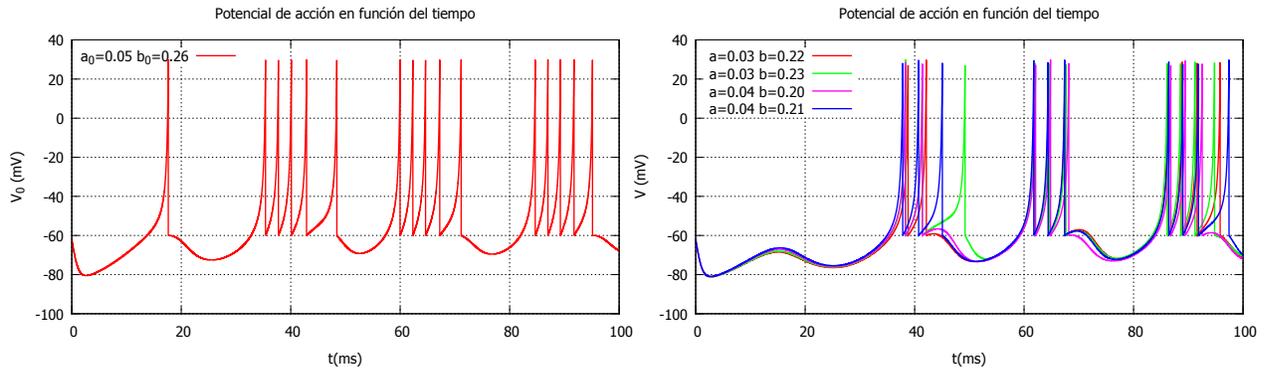
$$E(\vec{a}) = \frac{1}{N} \sum_{i=0}^N (v(i) - v_0(i))^2 \geq 0 \quad (5.1)$$

donde $v_0(i) \equiv v(t_i, a_0, b_0)$ es valor del potencial de membrana evaluado en el tiempo $t = i \cdot h$, siendo h el paso de integración. Este valor depende de los parámetros a_0 y b_0 con valores ya conocidos. El valor $v(i) \equiv v(t_i, a, b)$ representa el potencial de membrana que depende de los parámetros a ajustar a y b en el tiempo $t = i \cdot h$. N es la dimensión del vector, el cociente entre el intervalo de tiempo y el paso de integración, $N = \Delta t / h$. La normalización de la función es necesaria ya que de lo contrario, el error aumentaría al aumentar el intervalo temporal independientemente de acercarse o no a los valores deseados de a y b debido al sumatorio. La reproducción óptima de la salida neuronal v_0 se producirá cuando la función de error sea mínima, es decir su valor valga cero, lo que debería ocurrir para $a = a_0$ y $b = b_0$

Para encontrar los valores óptimos de a y b , barremos el espacio de parámetros entre unos límites prefijados. Para cada pareja (a, b) , calculamos el valor de la función de error buscando así el valor mínimo

del error que nos devolverá los valores de a_0 y b_0 .

Para una comparativa gráfica podemos observar las figuras siguientes, la figura de la izquierda representa la trayectoria que queremos reproducir mientras que la de la derecha muestra diferentes trayectorias obtenidas durante el barrido de puntos con sus parámetros correspondientes.



(a) Trayectoria de la salida experimental con los parámetros fijados.

(b) Trayectorias generadas con los parámetros obtenidos mediante el barrido de puntos.

Figura 5.1: Salidas neuronales del modelo IZ con una corriente externa sinusoidal, $I = I_0 \sin^2(\omega t) = 10 \sin^2(0,126t) \text{ mA}$. Condiciones iniciales: $v_0 = -62 \text{ mV}$, $u_0 = 0,2 \text{ mV}$. Parámetros del modelo conocidos: $c = -60 \text{ mV}$, $d = 0 \text{ mV}$. Parámetros de la trayectoria a determinar: $a_0 = 0,05 \text{ ms}^{-1}$, $b_0 = 0,26$.

La representación del valor de la función error en función de los parámetros a y b nos permite estimar gráficamente el valor de los parámetros que hace mínima dicha función y acotar así su valor.

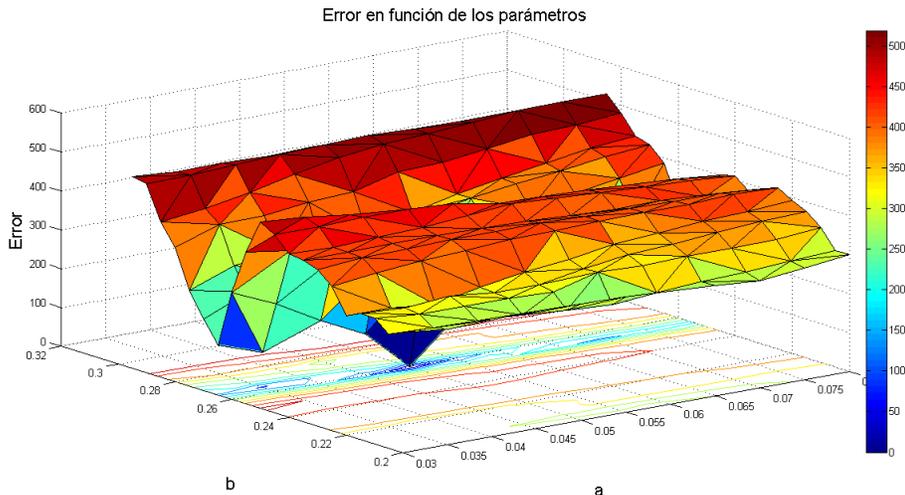


Figura 5.2: Gráfica de contorno y superficie de la función de error en función de los parámetros a determinar a y b . Los parámetros a y b están representados en los ejes X e Y respectivamente.

Según la figura 5.2 podemos observar que los valores que devuelven la trayectoria deseada corresponden a $a = 0,05ms^{-1}$ y $b = 0,26$, los cuales coinciden con los parámetros conocidos de dicha salida. Cabe destacar la existencia de dos mínimos locales y un mínimo global correspondiente a los valores óptimos ya mencionados.

5.2 Comparación de parámetros de mismos modelos mediante *Temple Simulado*

En vista de la simplicidad del método de la sección 5.1 junto a sus resultados, se propone hallar los parámetros de una salida neuronal del modelo de Izhikevich comparándola con otra salida neuronal del mismo modelo con los parámetros ya conocidos, realizando la búsqueda del mínimo global de la función de error (5.1) a través del algoritmo *Temple Simulado*. Primero, hallamos la salida neuronal del modelo experimental con los parámetros fijados, $v_0(i)$, resolviendo las EDO's con un Runge-Kutta de 4º orden. A continuación, calculamos la salida neuronal del modelo teórico, $v(i)$, resolviendo las EDO's con unos valores iniciales del conjunto de parámetros que queremos llegar a conocer, $\vec{a} = (a, b)$. Calculamos el valor de la función de error evaluada en el tiempo. De esta manera, obtenemos la configuración inicial de la función a minimizar, la cual perturbaremos para hallar la configuración óptima que minimice el coste, o en otras palabras, la función de error.

Establecemos la temperatura inicial y final, el número de ciclos de temperatura y el número de muestras en cada ciclo. Calculamos el factor de decrecimiento de la temperatura con cada iteración. Perturbamos aleatoriamente la configuración actual con la definición de los nuevos parámetros según la ecuación 4.4, con

$$\vec{a}' \in [\vec{a} + 0,1\Delta\vec{a}, \vec{a} - 0,1\Delta\vec{a}] \quad (5.2)$$

Establecemos límites inferiores y superiores a los valores de los nuevos parámetros acordes con el modelo neuronal seleccionado. Si los parámetros no son del orden adecuado, por ejemplo siendo éstos negativos en el modelo de Izhikevich, no generarán ninguna salida neuronal. Así, establecemos los siguientes límites

$$\begin{aligned} a' &\in [0, 0,7] \\ b' &\in [0, 1] \end{aligned} \quad (5.3)$$

Calculamos la nueva configuración de la función de error con los nuevos parámetros, calculamos la diferencia del coste entre la configuración nueva y la actual, $\Delta C = E(\vec{a}') - E(\vec{a})$. Si la nueva configuración es mejor que la actual, la aceptamos con probabilidad 1, sino la aceptamos sólo si un número aleatorio, $\hat{U}(0, 1)$, es menor que la probabilidad de aceptación (4.1). En caso de aceptar, la nueva configuración se establece como la configuración actual, aumentamos el número de configuraciones aceptadas, i_{pacc} , y calculamos el nuevo promedio de la diferencia del coste

$$\begin{aligned} \vec{a} &= \vec{a}' \\ E(\vec{a}) &= E(\vec{a}') \\ i_{pacc} &= i_{pacc} + 1 \\ \Delta C_{promedio} &= \frac{\Delta C_{promedio} \cdot (i_{pacc} - 1) + \Delta C}{i_{pacc}} \end{aligned} \quad (5.4)$$

5.2. Comparación de parámetros de mismos modelos mediante *Temple Simulado*

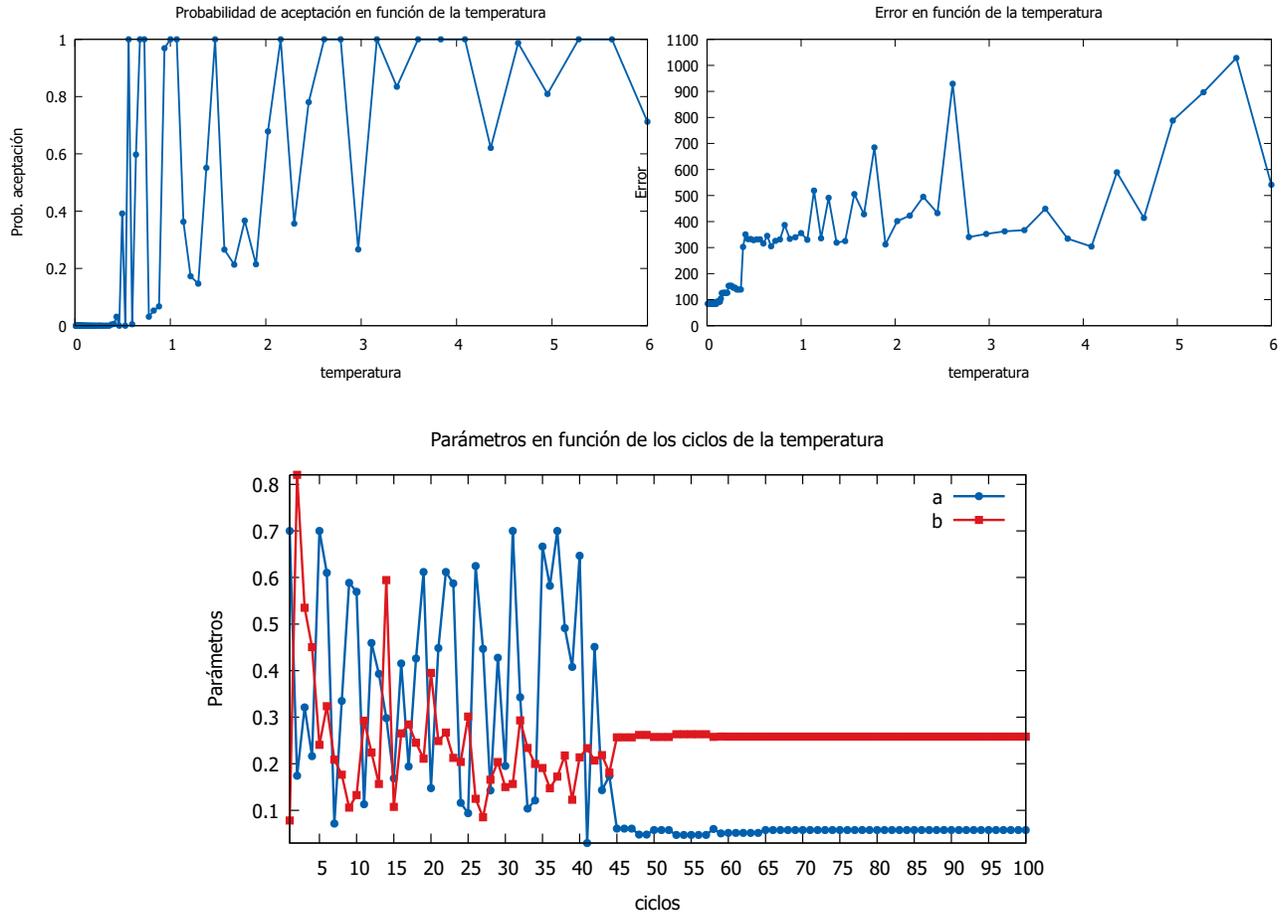


Figura 5.3: Salidas de la minimización de la función de error a partir del algoritmo *Temple Simulado*. Número de ciclos, $k = 100$. Número de muestras, $m = 300$. Parámetros fijados del modelo experimental, $a_0 = 0,05ms^{-1}$ y $b_0 = 0,26$

Repetimos este proceso tantas veces como número de muestras y guardamos la mejor configuración de parámetros al final de cada ciclo de temperatura. Finalmente, disminuimos la temperatura para el siguiente ciclo mediante (4.2) y volvemos a empezar con las perturbaciones. Realizamos este proceso hasta alcanzar la temperatura final.

La figura (5.3) muestra los resultados obtenidos de la minimización. Se puede observar en el panel superior de la izquierda como la probabilidad de aceptar configuraciones peores a la actual disminuye con la temperatura. Así mismo, en el panel superior de la derecha el valor de la función de error va decreciendo con la temperatura, alcanzando un valor óptimo de aproximadamente 84. El panel inferior muestra el valor de los parámetros a la salida de cada ciclo. Puede observarse que en la iteración número 65, el algoritmo ya ha encontrado los valores óptimos, $a = 0,058 ms^{-1}$ y $b = 0,258$, siendo casi idénticos a los fijados en el modelo experimental.

La figura (5.4) muestra el valor de la función de error para cada conjunto de valores a y b . Cabe destacar la existencia de dos mínimos locales y un mínimo global correspondiente a los valores óptimos encontrados.

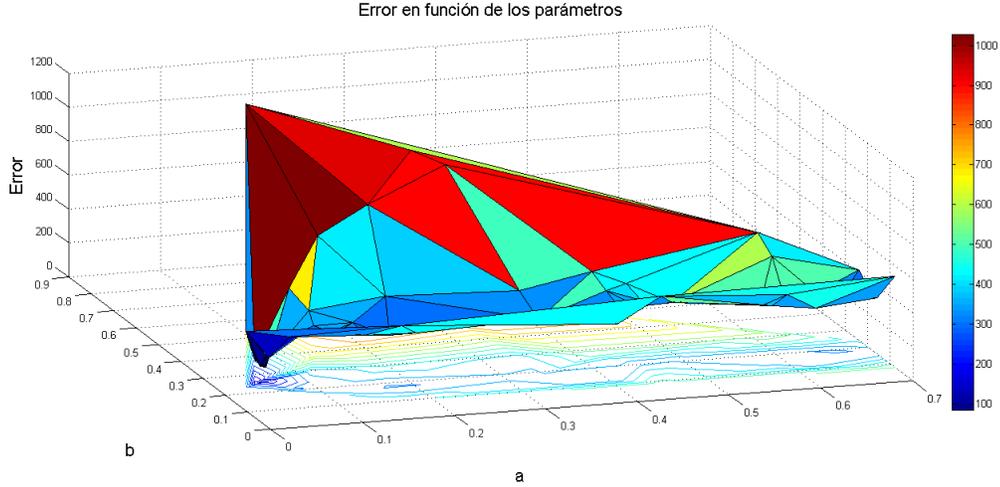


Figura 5.4: Gráfica de contorno y superficie de la función error en función de los parámetros a determinar a y b. Los parámetros a y b están representados en los ejes X e Y respectivamente.

5.3 Comparación de parámetros de diferentes modelos mediante *Temple Simulado*

A la vista que el método *Temple Simulado* es capaz de hallar con alta precisión los parámetros por comparación de dos mismos modelos neuronales, vamos a dar un paso hacia adelante, para ahora intentar reproducir los parámetros del modelo de Izhikevich comparándolo con las salidas neuronales de los modelos de Hindmarsh-Rose y Fitzhugh-Nagumo con este mismo método, explicado en la sección 5.1. Para ello, se lleva a cabo la comparación de la serie numérica correspondiente al modelo de Hindmarsh-Rose, con los parámetros fijados,

$$\frac{d\vec{v}_0(t)}{dt} = f_1(\vec{v}_0(t), \vec{b}) \quad (5.5)$$

con la salida de la serie numérica correspondiente al modelo de Izhikevich, con los parámetros a determinar,

$$\frac{d\vec{v}(t)}{dt} = f(\vec{v}(t), \vec{a}) \quad (5.6)$$

donde

$$f_1 \neq f, \quad \vec{b} \neq \vec{a}$$

Al tratarse de diferentes modelos neuronales, introducimos dos parámetros adicionales a determinar, λ y τ , debido a la reescalación de los ejes v y t . El parámetro λ , nos permite ajustar las amplitudes de las dos salidas neuronales, mientras que τ ajusta la frecuencia de pulsación de los modelos.

Al conocer los valores máximos y mínimos de las amplitudes en ambos modelos, consideramos buscar el parámetro λ de una forma más intuitiva, en lugar de pasarlo como parámetro desconocido. Buscamos reescalar las amplitudes $v \rightarrow av + b$ de manera que se igualen los valores máximo y mínimo de v y v_0 mediante la resolución del sistema

$$\begin{aligned} v_{max} &= a \cdot v_{0_{max}} + b \\ v_{min} &= a \cdot v_{0_{min}} + b \end{aligned} \quad (5.7)$$

con soluciones

$$\begin{aligned} a &= \frac{v_{0max} - v_{0min}}{v_{max} - v_{min}} \\ b &= v_{0max} - \frac{v_{0max} - v_{0min}}{1 - \frac{v_{min}}{v_{max}}} \end{aligned} \quad (5.8)$$

Si la corriente de entrada es constante, la frecuencia de pulsación también lo es. Por lo tanto, podríamos calcular el parámetro τ como el cociente entre las frecuencias de ambos modelos.

$$\tau = \frac{v_0}{v} \quad (5.9)$$

En cambio, si la corriente de entrada depende del tiempo, tanto la frecuencia como el parámetro τ no son constantes. Por ello, nos pareció más conveniente la búsqueda del parámetro τ mediante el algoritmo, también utilizado para la búsqueda de los parámetros a , b , c , d y I_{ext} del modelo de Izhikevich. De esta manera, podemos modificar la forma de la corriente de entrada con total libertad.

Dada la diferencia de escalas entre los dos modelos, modificamos la corriente de entrada para el de Izhikevich como

$$\gamma \cdot I_{ext_{HR/FN}} = I_{ext_{IZ}} \quad (5.10)$$

y consideramos a γ un parámetro más para el ajuste. El reescalado de los ejes, también modifica la forma de la función de error (5.1) como

$$E(\vec{a}) = \frac{1}{N} \sum_{i=0}^N (av(i/\tau) + b - v_0(i))^2 \quad (5.11)$$

Para operar dentro del régimen de pulsación deseado según nuestra salida experimental, establecemos una configuración inicial que permita llegar a esa salida.

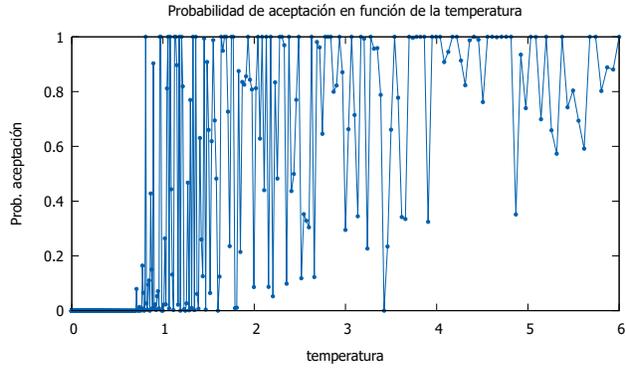
5.3.1 Régimen de tren de pulsos regulares

En un primer estudio, generamos una salida neuronal con los parámetros fijados de los modelos Hindmarsh-Rose y Fitzhugh-Nagumo con forma de pulsos regulares. Una vez establecida la configuración inicial en el modelo de Izhikevich, aplicamos el algoritmo *Temple Simulado*, perturbando la configuración inicial mediante (4.4) limitando adecuadamente cada parámetro.

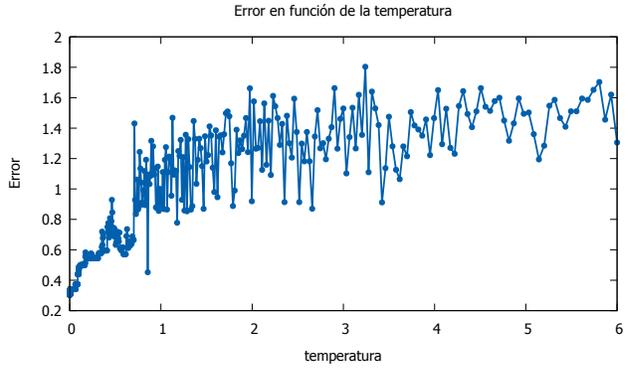
Ajuste al modelo de Hindmarsh-Rose

La figura (5.5) muestra los resultados obtenidos de la minimización de la función de error de un tren de pulsos regulares para el ajuste al modelo de Hindmarsh-Rose. El panel superior de la izquierda muestra la probabilidad de aceptación en función de la temperatura. El panel superior de la derecha muestra el valor de la función de error en función de la temperatura, alcanzando un valor óptimo de 0,308. Los paneles siguientes muestran los valores de los parámetros a la salida de cada ciclo. Puede observarse que en la iteración número 641, el algoritmo ya ha encontrado los valores óptimos, $a = 0,029 \text{ ms}^{-1}$, $b = 0,964$, $c = -55,060 \text{ mV}$, $d = 1,852 \text{ mV}$, $\tau = 0,254$ y $\gamma I_{ext_{HR}} = 31,040 \text{ mA}$.

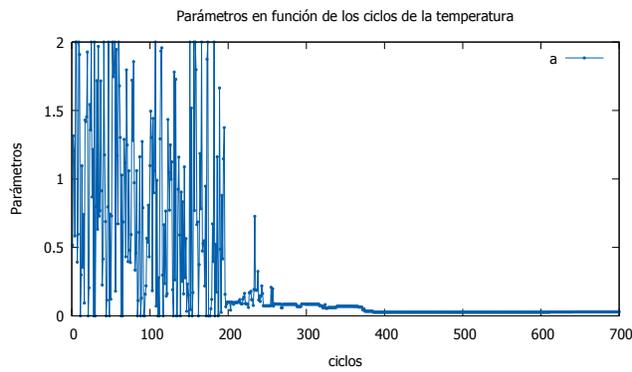
5.3. Comparación de parámetros de diferentes modelos mediante *Temple Simulado*



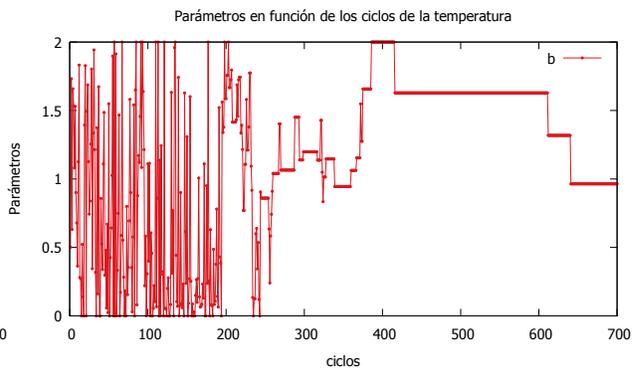
(a) Probabilidad de aceptación en función de la temperatura



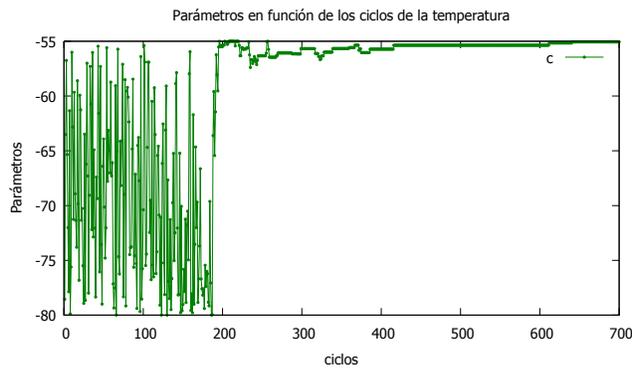
(b) Función de error en función de la temperatura



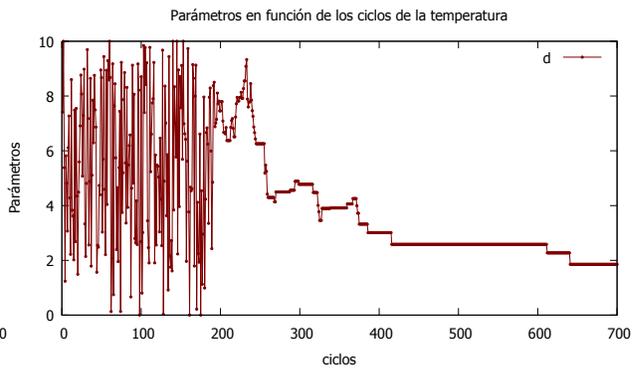
(c) Parámetro a en función del número de ciclos de temperatura



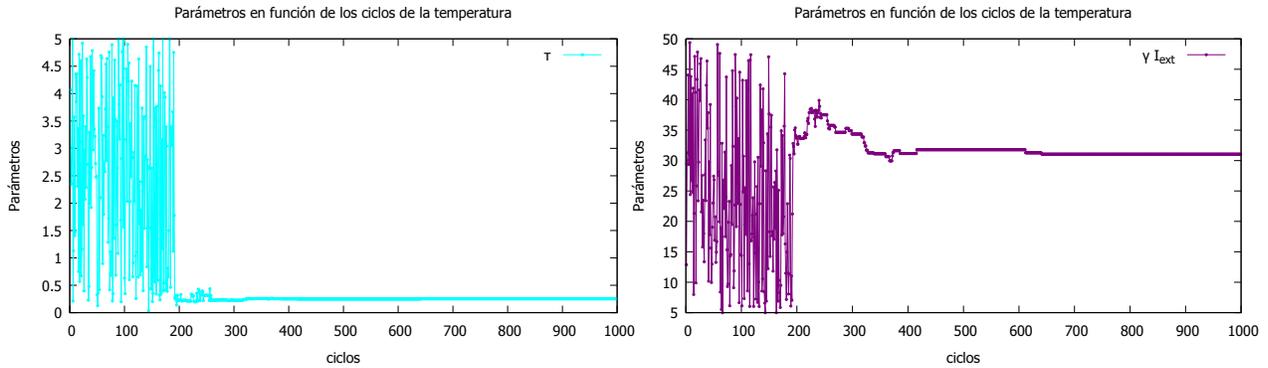
(d) Parámetro b en función del número de ciclos de temperatura



(e) Parámetro c en función del número de ciclos de temperatura



(f) Parámetro d en función del número de ciclos de temperatura



(g) Parámetro τ en función del número de ciclos de temperatura

(h) Parámetro I_{ext} en función del número de ciclos de temperatura

Figura 5.5: Salidas de la minimización de la función de error a partir del algoritmo *Temple Simulado*. Número de ciclos, $k = 1000$. Número de muestras, $m = 10^4$. Parámetros fijados del modelo experimental, $a = 1$, $b = 3$, $c = 1$, $d = 5$, $s = 1$, $r = 0,001$, $x_1 = -1,6$ y $I_{ext} = 1,5$

La figura (5.6) muestra las salidas neuronales de los dos modelos, aplicando los parámetros encontrados en el modelo de Izhikevich. Podemos observar que los parámetros encontrados generan un tren de pulsos dentro del mismo régimen de pulsación que el modelo Hindmarsh-Rose.

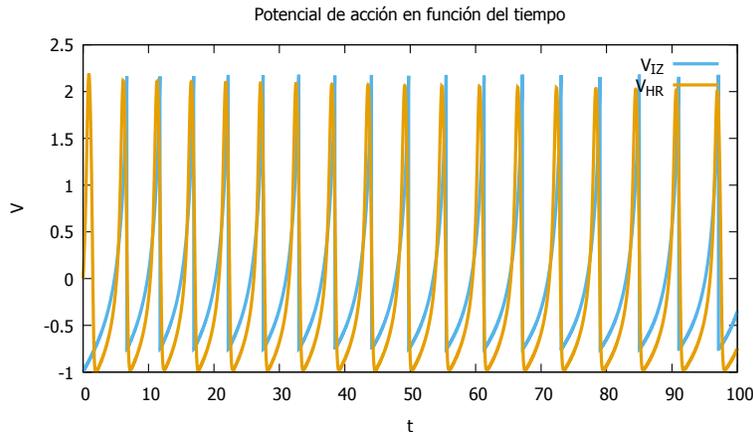
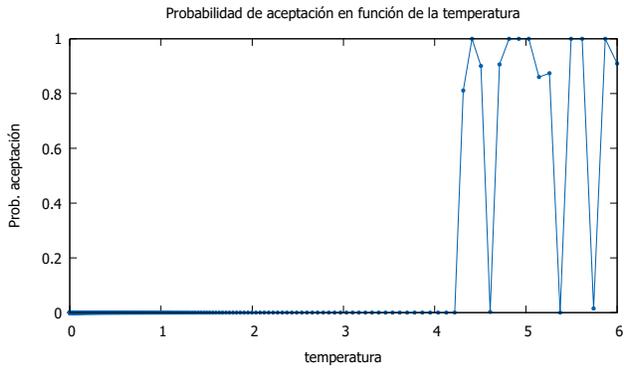


Figura 5.6: Salida neuronal del modelo Hindmarsh-Rose junto con la salida del modelo Izhikevich con los parámetros encontrados numéricamente

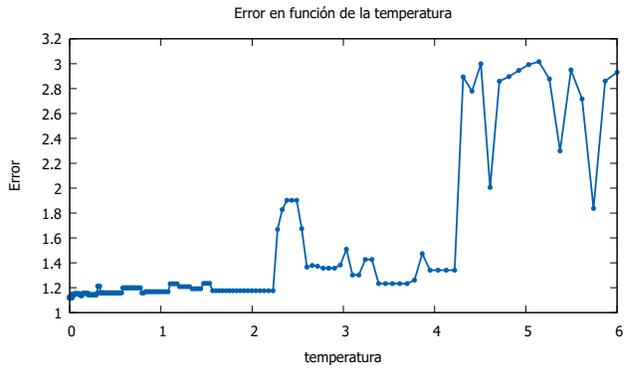
Ajuste al modelo Fitzhugh-Nagumo

La figura (5.7) muestra los resultados obtenidos de la minimización de la función de error de un tren de pulsos regulares para el modelo de Fitzhugh-Nagumo. El panel superior de la izquierda muestra la probabilidad de aceptación en función de la temperatura. El panel superior de la derecha muestra el valor de la función de error en función de la temperatura, alcanzando un valor óptimo de 1,12. Los paneles siguientes muestran los valores de los parámetros a la salida de cada ciclo. Puede observarse que en

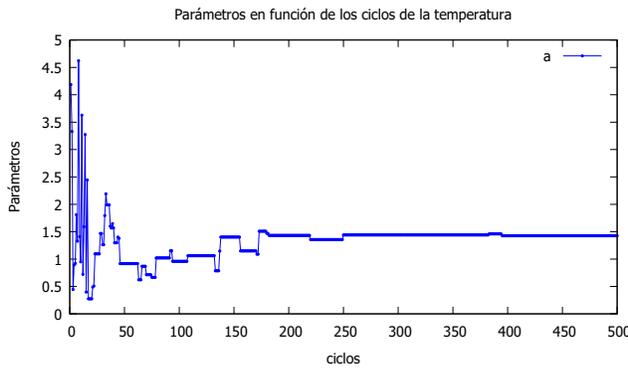
la iteración número 395, el algoritmo ya ha encontrado los valores óptimos, $a = 1,426 \text{ ms}^{-1}$, $b = 4,852$, $c = -76,158 \text{ mV}$, $d = 9,338 \text{ mV}$, $\tau = 0,012$ y $\gamma I_{extFN} = 36,621 \text{ mA}$.



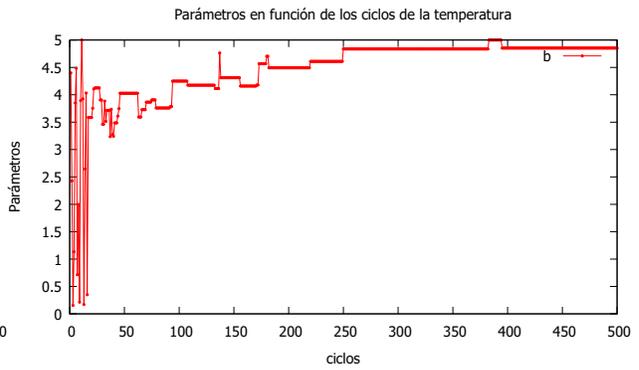
(a) Probabilidad de aceptación en función de la temperatura



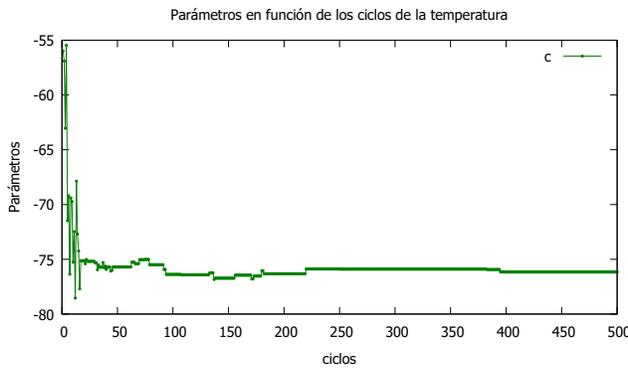
(b) Función de error en función de la temperatura



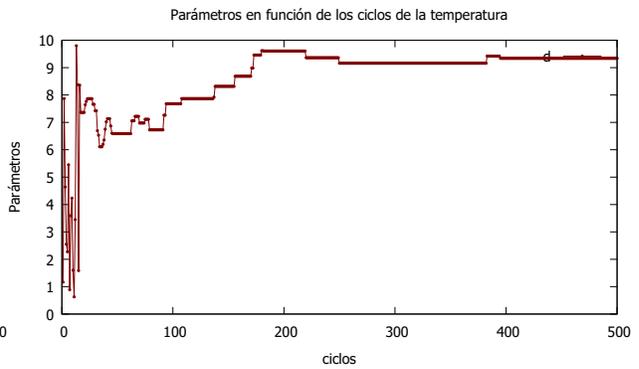
(c) Parámetro a en función del número de ciclos de temperatura



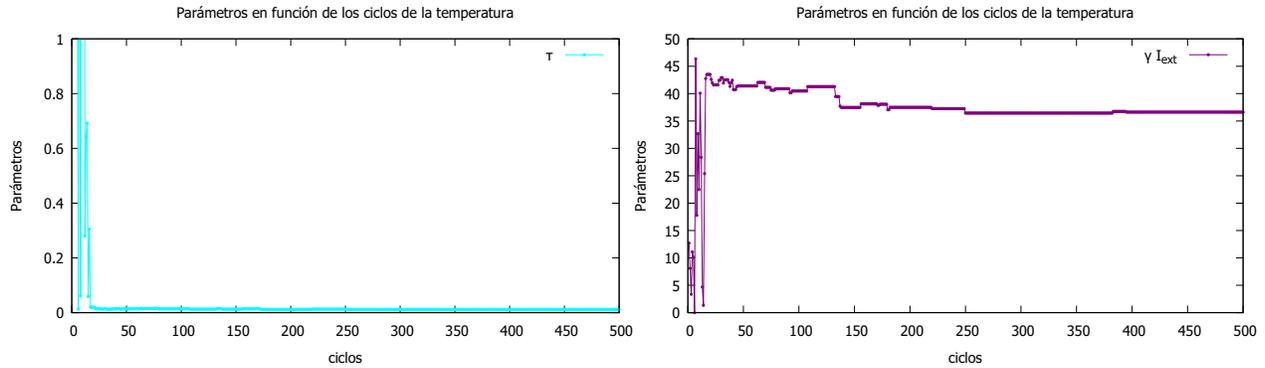
(d) Parámetro b en función del número de ciclos de temperatura



(e) Parámetro c en función del número de ciclos de temperatura



(f) Parámetro d en función del número de ciclos de temperatura



(g) Parámetro τ en función del número de ciclos de temperatura

(h) Parámetro I_{ext} en función del número de ciclos de temperatura

Figura 5.7: Salidas de la minimización de la función de error a partir del algoritmo *Temple Simulado*. Número de ciclos, $k = 500$. Número de muestras, $m = 10^3$. Parámetros fijados del modelo experimental, $a = 0,7$, $b = 0,8$, $c = 0,08$, $I_{ext} = 0,5$

La figura (5.8) muestra las salidas neuronales de los modelos de Fitzhugh-Nagumo e Izhikevich, aplicando los parámetros encontrados en el modelo de Izhikevich. Podemos observar que los parámetros encontrados generan un tren de pulsos dentro del mismo régimen de pulsación que el modelo Fitzhugh-Nagumo, con aproximadamente la misma frecuencia y amplitud.

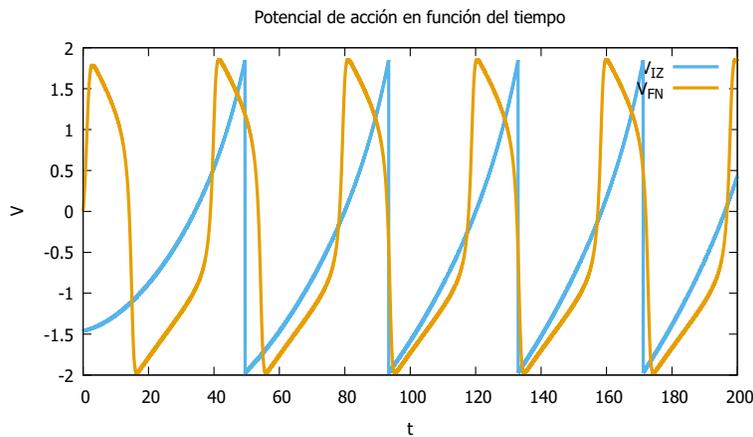
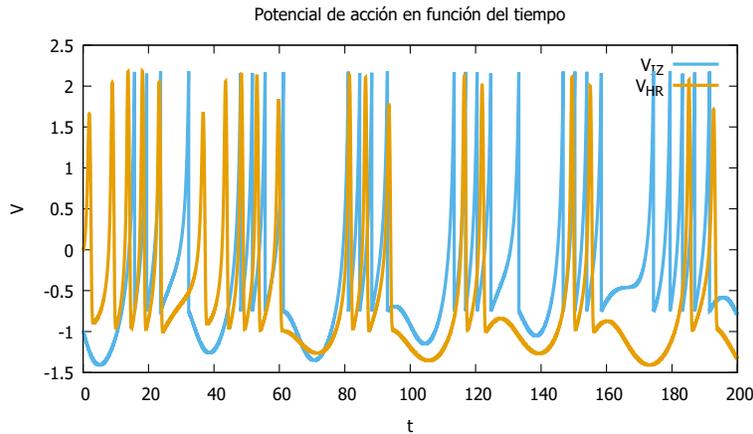


Figura 5.8: Salida neuronal del modelo Fitzhugh-Nagumo junto con la salida del modelo Izhikevich con los parámetros encontrados numéricamente

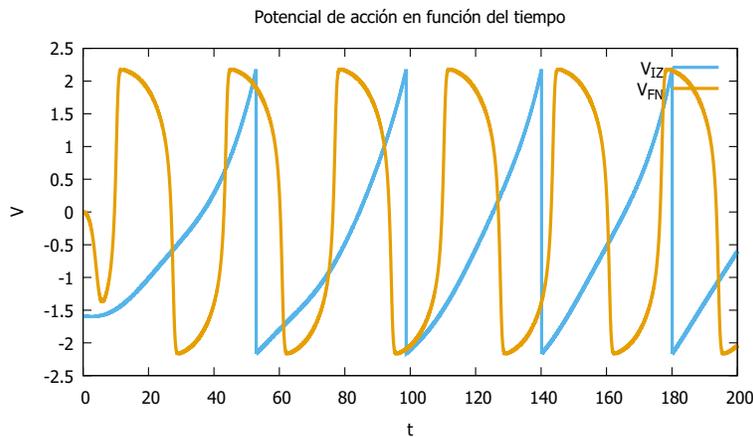
Comparación de diferentes modelos neuronales con los parámetros fijados

Una aplicación directa a los resultados obtenidos en (5.3.1) es la salida neuronal del modelo de Izhikevich con los parámetros encontrados, generada a partir de una corriente de entrada sinusoidal, $I_{ext} = I_0 \sin^2(\omega t)$. Comparamos esta salida neuronal con la generada a partir de los modelos de Hindmarsh-Rose y Fitzhugh-Nagumo, aplicando la misma corriente de entrada sinusoidal.

La figura (5.9) muestra los resultados de las salidas neuronales.



(a) Salida neuronal del modelo Hindmarsh-Rose junto con la salida del modelo Izhikevich con los parámetros encontrados numéricamente.



(b) Salida neuronal del modelo Fitzhugh-Nagumo junto con la salida del modelo Izhikevich con los parámetros encontrados numéricamente.

Figura 5.9: Salidas neuronales de los modelos experimentales junto con la salida neuronal del modelo de Izhikevich aplicando una corriente externa sinusoidal, $I_{ext} = I_0 \sin^2(\omega t) = 2 \sin^2(0,092 t) \text{ mA}$.

5.4 Ajuste de los parámetros de un modelo neuronal a una salida experimental dada

Por último, es posible profundizar más en el estudio de la determinación de parámetros neuronales mediante Temple Simulado, comparando el modelo teórico a estudiar con una salida neuronal aplicando una corriente externa con forma de ruido blanco, ambos obtenidos mediante datos¹ experimentales. En

¹Cultivo de una célula de rata, proporcionado por el Dr. Miguel Maravall; Private Communication, Sussex Neuroscience, University of Sussex

este caso la señal se puede pensar que ha sido generada mediante una función f_1 desconocida. La figura (5.10) muestra los resultados de la salida neuronal experimental junto con la salida neuronal del modelo de Izhikevich con los parámetros encontrados numéricamente mediante el método *Temple Simulado*. Podemos observar que los valores de los parámetros encontrados son capaces de pulsar dentro del mismo régimen de pulsos y generar una salida con un valor de la función de error de aproximadamente 302, la cual es bastante buena. Los parámetros encontrados en la simulación numérica son $a = 0,034ms^{-1}$, $b = 0,106$, $c = -66,892mV$, $d = 4,749mV$, $\tau = 1,309$, $\gamma = 65,645$.

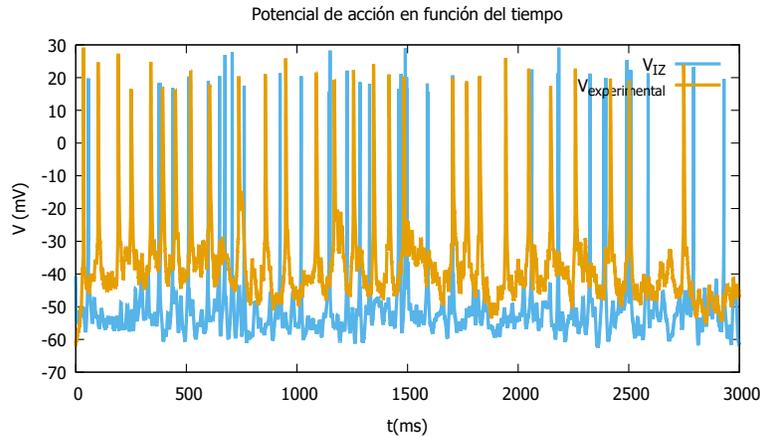


Figura 5.10: Salida neuronal de datos experimentales en función del tiempo junto con la salida neuronal numérica del modelo de Izhikevich. Número de ciclos, $k = 50$. Número de muestras, $m = 10^4$.

Podemos observar que el reescalado de la amplitud no es muy preciso debido a las significativas diferencias entre la amplitud de algunos de los pulsos. Para una posible mejora al ajuste anterior, acotamos el intervalo de perturbación con pequeñas fluctuaciones alrededor de los parámetros obtenidos en (5.10) y añadimos tres parámetros adicionales, en un principio fijados del modelo, a los seis del ajuste que genera la salida de la figura. Los resultados obtenidos de esta nueva simulación no han generado ninguna mejora significativa al ajuste de seis parámetros, pero han servido para averiguar que para el ajuste también se tendría que considerar la inicialización de la variable u del modelo de Izhikevich.

RESULTADOS

Las aplicaciones de la sección anterior (5) muestran la efectividad del algoritmo *Temple Simulado* a la hora de hallar un total de seis parámetros desconocidos e independientes entre sí, resultado de la minimización del error al querer reproducir usando el modelo de Izhikevich pulsos neuronales generados mediante otros modelos o datos experimentales. En una primera aplicación, la obtención de los parámetros mediante el sistema de barrido de puntos, resultó útil para comprender la estructura de la función de error a minimizar e implementarla en un algoritmo de búsqueda sencillo, el cual efectivamente nos devolvió, exactamente, la solución de los parámetros a determinar junto con sus trayectorias.

Una vez determinado el comportamiento de la función de error con los parámetros, decidimos hallar la solución mediante el algoritmo *Temple Simulado* con tres aplicaciones diferentes:

- Primero, hemos verificado que el algoritmo sea capaz de reproducir la misma serie neuronal del modelo Izhikevich. De esta forma, comprobamos que la solución numérica devolvía aproximadamente los parámetros impuestos, generando así un valor suficientemente pequeño de la función de error. Los resultados de las figuras (5.3) muestran cómo la probabilidad de aceptación de estados con energía más altas va disminuyendo con la temperatura, como cabía esperar. Del mismo modo, la disminución de la función de error con la temperatura muestra la correcta elección del rango de temperaturas. Si la temperatura inicial no hubiese sido lo suficientemente alta o la disminución de ésta no lo suficientemente lenta, tal vez nos hubiésemos quedado atascados en el mínimo local, con un valor de la función de error aproximado de 350 para una temperatura de $T \sim 0,5$, en lugar de hallar el valor correcto del mínimo global correspondiente a la misma figura. Por otra parte, los límites inferiores y superiores de la configuración de los nuevos parámetros son necesarios ya que los modelos estudiados no aceptan parámetros con valores negativos. Además, cuanto más pequeño sea el rango establecido por la perturbación aleatoria, más posibilidades tiene el algoritmo de encontrar los parámetros óptimos en un mínimo de tiempo. Estableciendo la perturbación (4.4), nos aseguramos intentar pasar por el máximo de puntos cercanos a la configuración actual y así, evitar dar saltos enormes que dificulten hallar los parámetros en un mínimo de tiempo.
- Luego, atacamos el problema buscando los parámetros por comparación de distintos modelos neuronales. Los resultados de las figuras (5.5 y 5.7) muestran que el modelo de Izhikevich se ajusta mejor al modelo de Hindmarsh-Rose que al modelo de Fitzhugh-Nagumo, ya que el mínimo de la función de error es menor respecto al modelo de Hindmarsh-Rose. Cabe destacar que en la aplicación con los parámetros encontrados de la figura (5.9), observamos que los diferentes modelos neuronales con la misma corriente de entrada sinusoidal son capaces de generar diferentes regímenes de pulsos.

Esto es debido a que el modelo de Izhikevich no sólo depende de los parámetros hallados para el ajuste a cada modelo, sino también del parámetro adicional γ , correspondiente a la amplitud del pulso, el cual permite la pulsación del modelo de Izhikevich con la misma corriente sinusoidal de los dos diferentes modelos numéricos. Por otra parte, era de esperar que el modelo de Izhikevich se adaptara más al modelo de Hindmarsh-Rose que al Fitzhugh-Nagumo ya que el primero tiene más regímenes de pulso en común con el Izhikevich que el segundo. No obstante, teniendo en cuenta las limitaciones de los modelos neuronales, podemos concluir que el modelo de Izhikevich se asemeja bastante bien a los dos modelos usados como prueba, encontrando unos muy buenos parámetros que reproducen prácticamente los dos modelos en el rango de parámetros investigado.

- Por último, hemos intentado ajustar el modelo de Izhikevich a una salida verdaderamente experimental. Hemos usado una serie temporal correspondiente a una neurona de una rata de laboratorio gentilmente cedida por el Dr. Miguel Maravall de la Universidad de Sussex. El resultado de la figura (5.10) muestra que los valores de los parámetros encontrados en la simulación son capaces de generar pulsos dentro del mismo régimen de pulsación e incluso muchos de ellos con la misma frecuencia. No obstante vemos que el reescalado de la amplitud no es muy preciso debido a las significativas diferencias entre la amplitud de algunos de los pulsos. En vista de las discrepancias y con el fin de lograr un ajuste mejor, hemos intentamos acotar el intervalo de perturbación lo máximo posible teniendo en cuenta los valores de los parámetros encontrados en la simulación. Para ello, añadimos tres parámetros adicionales, en un principio fijados del modelo, a los seis del ajuste que genera la salida de la figura. Los resultados obtenidos de esta nueva simulación no generaron ninguna mejora significativa al ajuste de seis parámetros, pero nos fue útil para averiguar que para el ajuste también se tendría que considerar la inicialización de la variable u del modelo de Izhikevich.

BIBLIOGRAFÍA

- [1] S. Doi, J. Inoue, Z. Pan, and K. Tsumoto, *Computational Electrophysiology*. Springer, 2010. 2.1, 2.1, 2.3
- [2] L. N. Long and G. Fang, “A review of biologically plausible neuron models for spiking neural networks,” in *AIAA infoTech@Aerospace Conference*. 2.1
- [3] E. M. Izhikevich *et al.*, “Simple model of spiking neurons,” *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003. 2.2.1
- [4] E. M. Izhikevich, “Which model to use for cortical spiking neurons?,” *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004. 2.2.1
- [5] R. FitzHugh, “Impulses and physiological states in models of nerve membrane,” *Biophys. J.*, 1961. 2.2.2
- [6] J. Hindmarsh and R. Rose, “A model of neuronal bursting using three coupled first order differential equations,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 221, no. 1222, pp. 87–102, 1984. 2.2.3
- [7] W. H. Press, S. Teukolsky, W. Vetterling, and B. Flannery, “Numerical recipes in fortran (cambridge),” 1992. 3.1
- [8] S. Kirkpatrick, G. Gelatt, C. D., JR., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, 1983. 4.1
- [9] R. Salazar and R. Toral, “Simulated annealing using hybrid monte carlo,” *Journal of Statistical physics*, vol. 89, no. 5-6, pp. 1047–1060, 1997. 4.1, 4.1
- [10] R. Toral and P. Colet, *Stochastic numerical methods: an introduction for students and scientists*. John Wiley & Sons, 2014. 4.1
- [11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953. 4.1
- [12] J. F. Epperson, *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013. 5.1