



Proyecto final - Modelo actividad neuronal

Guía de actividades

Objetivo de la Actividad	Comprender e implementar procedimientos en el computador que permitan obtener una estimación de la solución de modelos descritos por ecuaciones diferenciales ordinarias.
Resultado de la actividad	Desarrollar una interfaz gráfica en Python que permita a un usuario estimar la solución del modelo computacional de actividad neuronal probando diferentes valores de parámetros y métodos de solución de sistemas numéricos.
Fecha de la actividad	22 de abril de 2022

Descripción del problema:

Potencial de acción:

Las neuronas han desarrollado un mecanismo de señalización eléctrica para comunicarse con otras neuronas. Esta señalización consiste en la generación de potenciales de acción.

Los potenciales de acción son cambios súbitos en el potencial eléctrico que se genera a partir del flujo de iones a lo largo de la membrana plasmática de la neurona. Esto se debe especialmente a dos aspectos:

- Existe una diferencia en la concentración de cargas dentro y fuera de la membrana (diferencia de potencial = voltaje) denominado potencial de membrana (V_m), definido como: $V_m = V_{in} - V_{out}$.
- La membrana celular es permeable de manera selectiva a algunos iones (canales iónicos) [1].

El gradiente de concentración de iones extra e intracelular en reposo conlleva a que las neuronas mantengan una diferencia de potencial negativo, denominado potencial de reposo (resting potential). Su valor oscila entre -90mV y -40mV, dependiendo del tipo de neurona.

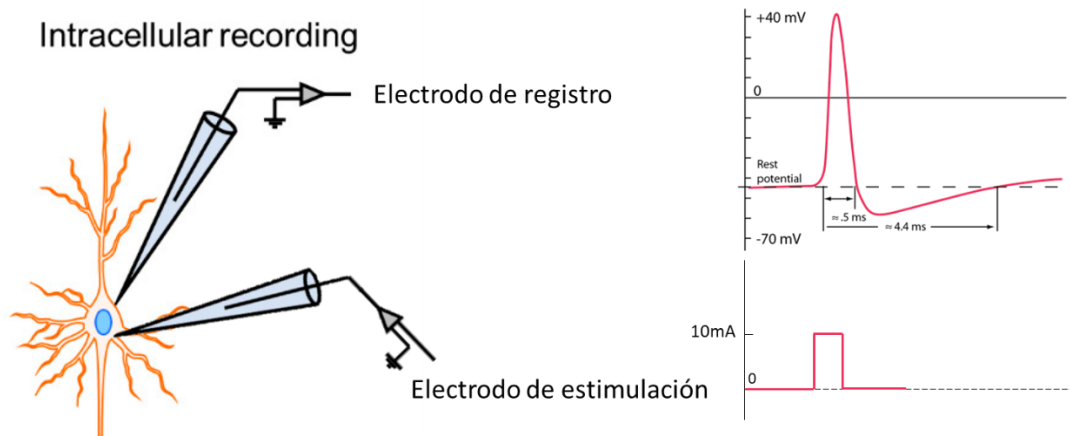


Figura 1. Generación de un potencial de acción. Modificado de [1].



Los principales iones que interactúan tanto dentro como fuera de la célula son: Na^+ , K^+ y Cl^- . La concentración de iones de K^+ dentro de la célula es alrededor de 10 veces mayor que afuera, mientras que hay una mayor concentración extracelular de Na^+ y Cl^- . De esta forma, si a una neurona, que naturalmente mantiene un voltaje negativo, entra una corriente positiva (por ejemplo, iones de Na^+), esto hace que el potencial de membrana se vuelva menos negativo, es decir, ocurre una despolarización. Por el contrario, si una corriente positiva (K^+) sale de la célula, o si iones negativos (Cl^-) entran a ella, la membrana de potencial se hiperpolariza [1].

Un potencial de acción, también conocido como spike (o espiga), es una respuesta muy rápida (menor a 5ms), donde el potencial de membrana de una neurona pasa de ser negativo a positivo, como se observa en la Figura 1. En una primera fase, la neurona mantiene un potencial de reposo (V_{rest}) negativo. Luego, como consecuencia de algún estímulo despolarizante, los canales de sodio se abren permitiendo que los iones de sodio entren a la célula. Cuando el nivel de despolarización alcanza cierto umbral (threshold), ocurre un potencial de acción. Finalmente, la permeabilidad del potasio se restaura y el potencial de membrana vuelve a su estado de reposo.

En la práctica las neuronas pueden ser aisladas del tejido cerebral y ser colocadas en medios que permitan mantenerlas vivas por mucho tiempo. Estas neuronas pueden ser "pinchadas" con un electrodo muy fino y medir su potencial eléctrico por medio de un multímetro o un osciloscopio con características especiales. Si medimos la diferencia de potencial entre el medio extracelular e intracelular, encontraremos que existe un potencial de membrana negativo. Si estimulamos eléctricamente la neurona (por ejemplo, inyectando iones con carga positiva), podemos simular el escenario en el que la neurona recibe un mensaje de otra neurona.

Modelo matemático de Izhikevich:

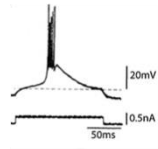
Existen modelos matemáticos que representan complejos aspectos moleculares de una neurona de manera precisa (Hodgkin y Huxley); sin embargo, requieren un alto poder computacional cuando grandes grupos de neuronas interconectadas son representadas. Además, desde su primer desarrollo se ha encontrado que las neuronas son capaces de producir potenciales de acción de formas muy particulares; por ejemplo, algunas neuronas de la corteza disparan rápido ante la estimulación eléctrica, posiblemente para representar aspectos visuales de manera precisa, pero las neuronas cortico-talámicas disparan ráfagas cortas de potenciales de acción. Por este motivo, son necesarios modelos más completos de redes de neuronas que permitan representar todos estos tipos de células (Figura 2).



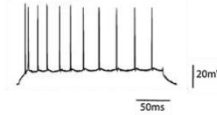
Programación Científica

Facultad de Ingeniería

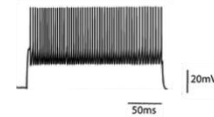
Neurona tálamo-cortical



Neurona hipocampal



Neurona Corteza visual



Neurona talámica

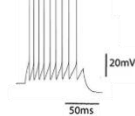


Figura 2. Diferentes tipos de neuronas poseen una forma particular de producir potenciales de acción.

Izhikevich (2003) encontró una manera de representar distintos grupos de neuronas utilizando solamente 2 ecuaciones diferenciales:

$$\frac{dv}{dt} = 0.04v^2 + 4v + 140 - u + I \quad (1)$$

$$\frac{du}{dt} = a(bv - u) \quad (2)$$

Con la siguiente condición:

$$\text{si } v \geq 30\text{mV}, \text{ entonces } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (3)$$

Donde v representa el potencial de membrana, y u representa una variable de recuperación de la membrana, es decir la tendencia de la neurona a volver al potencial de reposo. Los parámetros a , b , c y d hacen referencia al tiempo de recuperación de u , la sensibilidad de u , el valor de reinicio del potencial de membrana v después de la generación de un spike, y el reinicio de u después de la generación un spike respectivamente.



Programación Científica

Facultad de Ingeniería

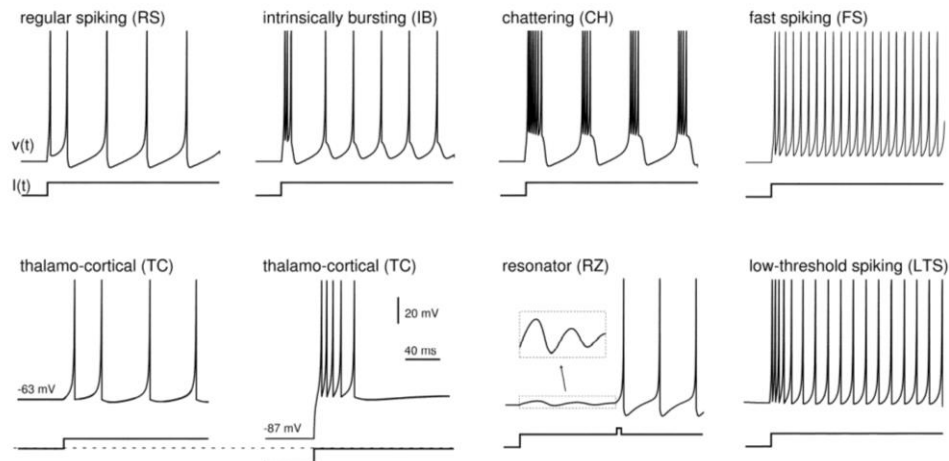


Figura 3. Tipos de spikes (Tomado de [2]).

Variando los parámetros a , b , c y d se podrán representar neuronas capaces de generar spikes complejos, como se observa en la Figura 3.

Instrucciones para el desarrollo del Proyecto.

Requerimientos:

- Se debe implementar una interfaz gráfica que le permita al usuario interactuar con el programa y poder visualizar diferentes tipos de potenciales de acción dependiendo de los parámetros seleccionados (Regular Spiking, fast spiking, chattering, Intrinsically bursting, thalamo-cortical, etc). En la figura 4 se muestra un ejemplo de cómo puede ser desarrollada esta interfaz, así como los aspectos básicos que debe incluir.
- La solución del modelo de Izhikevich (eq. 1, 2 y 3) debe hacerse implementando los métodos vistos en clase: *Euler hacia adelante*, *Euler hacia atrás*, *Euler modificado*, *Runge-Kutta de segundo orden*, *Runge-Kutta de cuarto orden* y *solve_ivp*.
- El usuario debe poder escoger cuál método utilizar y seleccionar la cantidad de variables en la gráfica del modelo ($V(t)$, $u(t)$), éstas deben ser completamente identificables mediante un *label*.
- El botón de salir debe ser funcional.
- El usuario debe poder exportar e importar los datos de la gráfica en formato binario tipo 'double' (puede ser un archivo para cada variable y vector de tiempo).
- Se deben poder modificar los parámetros y el tiempo de simulación (en milisegundos).
- La interfaz debe integrar una ventana de Matplotlib para poder interactuar con la gráfica (Figura 4).

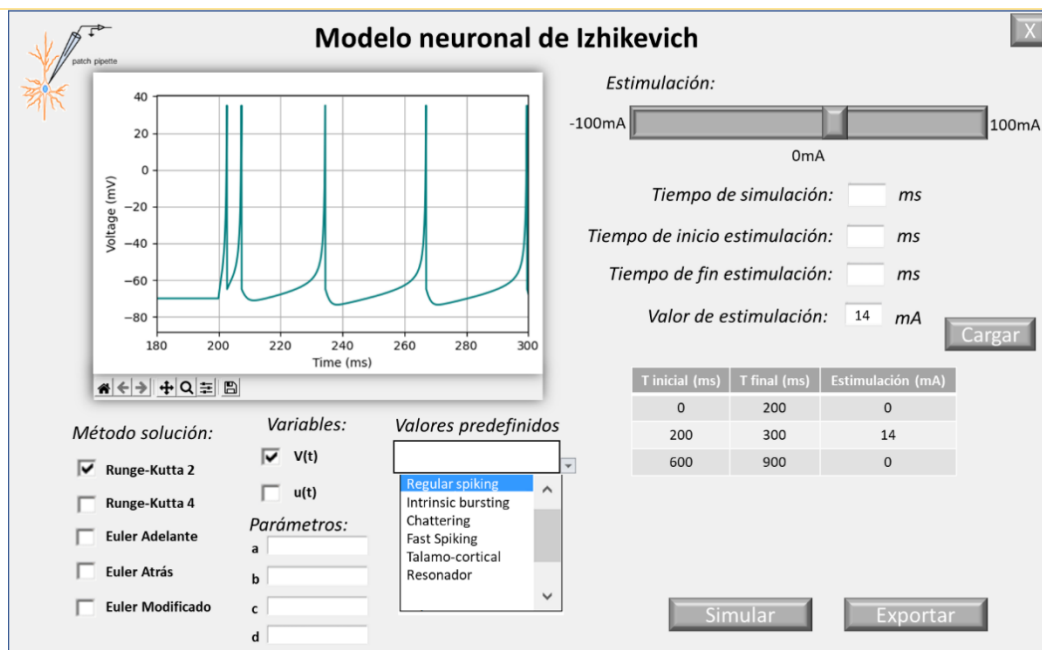


Figura 4. Modelo para la interfaz gráfica. El usuario debe poder escoger las variables dependientes para incluir en las gráficas, el método de solución y debe ser capaz de modificar los parámetros.

Entregables del proyecto:

Este proyecto contempla la entrega de dos productos: el primero consiste en la interfaz y código desarrollado del sistema y el segundo, en un video que presente el trabajo realizado. El video debe tener una duración de mínimo 6 minutos y máximo 7 minutos, que incluya los siguientes aspectos:

- Breve descripción del problema.
- Se debe explicar qué evalúa el modelo matemático.
- Comentar cómo funciona el código.
- Incluir una demostración de la interfaz y programa realizado en donde se presenten diferentes escenarios de simulación cambiando los valores de las variables del modelo.
- Comentar qué aspectos se podrían mejorar del modelo, como por ejemplo si se pudiera incluir alguna variable adicional o si es realista en relación con la problemática discutida.
- Comentar qué posibles problemas se encontraron al momento de realizar la interfaz, así como las soluciones propuestas.

Instrucciones adicionales:

- El proyecto debe ser presentado por grupos de 3 personas (pueden ser de secciones diferentes).
- El código entregado debe estar debidamente comentado.



Programación Científica

Facultad de Ingeniería

- La presentación y código deben ser subidas a Bloque Neón el día anterior de la presentación.

Cronograma de entregas:

Entrega 1:

2 de mayo: Entrega de presentación (3-4 diapositivas) con el contexto general de la problemática del proyecto, así como, con la descripción de las ecuaciones y de los diferentes parámetros incluidos en el modelo. (15%)

Entrega 2:

9 de mayo: Entrega del “esqueleto” de la interfaz gráfica realizada en Python con la librería Tkinter (20%).

Entrega 3:

23 de mayo: Entrega de la interfaz gráfica funcional (80%).

Entrega 4:

Semana 30 de mayo: Presentación final, entrega del video y versión final del código (85%).

Evaluación:

El proyecto final se va a evaluar bajo los siguientes criterios:

- Presentación (50% repartidos entre entrega 1 y entrega 4 con los porcentajes mencionados arriba)
- Código (50% repartidos entre entrega 2, 3 con los porcentajes mencionados arriba)

Referencias:

[1] <http://hyperphysics.phy-astr.gsu.edu/hbase/Biology/actpot.html>

[2] Izhikevich, Eugene M. "Simple model of spiking neurons." IEEE Transactions on neural networks 14.6 (2003): 1569-1572.