Práctica 2 - Dinámica de sistemas acoplados

Zablotsky, Amir Nicolás
Redes Neuronales 2022
Instituto Balseiro, UNCuyo

EJERCICIO 1

Se implementó un modelo computacional para simular la dinámica de dos neuronas Hodgkin-Huxley interactuando de manera simétrica.

La corriente de interacción sináptica está dada por

$$I_{\text{syn}}(t) = -g_{\text{syn}} s(t) \left(V_{\text{post}} - V_{\text{syn}}\right)$$

donde

$$\frac{ds}{dt} = \frac{s_\infty(V_{pre}) - s}{\tau}$$

con $s_\infty(V) = 0{,}5\left(1 + \tanh(V/5)\right)$. De esta manera nuestro sistema queda descrito por seis ecuaciones diferenciales acopladas, que se enseñan en la Ec. 1, donde $i = \{1, 2\}$ representa a cada neurona, y $j \neq i$ corresponde a la otra neurona.

$$\begin{cases} C\frac{dV_i}{dt} = I_{\text{ext}} + I_{\text{syn}} - g_{\text{Na}} m^3 h (V_i - V_{\text{Na}}) - g_{\text{K}} n^4 (V_i - V_{\text{K}}) - g_{\text{l}} (V_i - V_{\text{l}}) \\ \frac{dx_i}{dt} = \frac{x_\infty(V_i) - x_i}{\tau_x(V_i)}\,, \quad x = m, h, n \\ \frac{ds_i}{dt} = \frac{s_\infty(V_j) - s_i}{\tau} \end{cases} \tag{1}$$

Debajo se describen los parámetros y funciones utilizadas para modelar las neuronas HH, y los parámetros y funciones correspondientes en particular a la interacción sináptica entre las mismas:

$$\text{Neuronas HH} \begin{cases} C = 1\,\mu\text{F/cm}^2 \\ I_{\text{ext}} = 12\,\text{nA} \ \text{(de manera que el sistema presente oscilaciones periódicas)} \\ V_{\text{Na}} = 50\,\text{mV}, \ \ g_{\text{Na}} = 120\,\text{mS/cm}^2 \\ V_{\text{K}} = -77\,\text{mV}, \ \ g_{\text{K}} = 36\,\text{mS/cm}^2 \\ V_{\text{l}} = -54{,}4\,\text{mV}, \ \ g_{\text{l}} = 0{,}3\,\text{mS/cm}^2 \\ x_\infty(V) = \frac{a_x}{a_x + b_x}, \ \ \tau_x = \frac{1}{a_x + b_x}\,\text{ms}\,, \ \ x = m, h, n \\ a_m = 0{,}1(V + 40)/\left(1 - e^{-\frac{V+40}{10}}\right), \ \ b_m = 4e^{-\frac{V+65}{18}} \\ a_h = 0{,}07e^{-\frac{V+65}{20}}, \ \ b_h = 1/\left(1 + e^{-\frac{V+35}{10}}\right) \\ a_n = 0{,}01(V + 55)/\left(1 - e^{-\frac{V+55}{10}}\right), \ \ b_n = 0{,}125e^{-\frac{V+65}{80}} \end{cases}$$

$$\text{Interacción sináptica} \begin{cases} V_{\text{syn}} \in \{-80\,\text{mV}, 0\,\text{mV}\} \ \text{(dependiendo el caso estudiado)} \\ g_{\text{syn}} \in \left[0\,\text{mS/cm}^2, 2\,\text{mS/cm}^2\right] \\ s_\infty(V) = 0{,}5\left(1 + \tanh\left(V/5\right)\right), \ \ \tau = 3\,\text{ms} \end{cases}$$

Para implementar la dinámica, se utilizó el algoritmo de Euler con un paso temporal $dt = 0{,}001\,\text{ms}$, y se dejaron transcurrir las dinámicas durante 2000 ms para asegurarse de que los sistemas alcancen el régimen estacionario. En cada simulación se inicializa cada neurona con un juego de parámetros $(V, m, h, n, s) = \left(V_{\text{rand}} \in [-30\,\text{mV}, -10\,\text{mV}], 0, 0, 0, 0\right)$.

En la Fig. 1 se observan los potenciales de membrana para las dos neuronas en función del tiempo (en el régimen estacionario), para los casos de potencial sináptico nulo y $-80\,\text{mv}$. Podemos ver como cuando $V_{\text{syn}} = 0\,\text{mV}$ ambas neuronas se sincronizan en fase (interacción excitatoria), por lo que los disparos de ambas neuronas ocurren simultáneamente (picos superpuestos). En contra parte, en el caso de $V_{\text{syn}} = -80\,\text{mV}$ vemos que las neuronas disparan de manera alternada, es decir que se sincronizan en antifase (interacción inhibitoria).

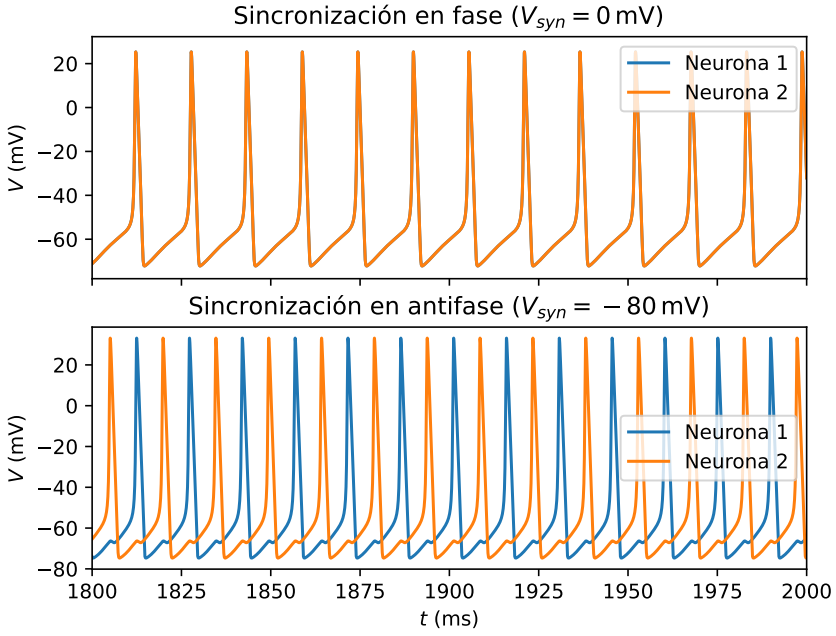

Figura 1: Potencial de membrana en función del tiempo para ambas neuronas, en el régimen estacionario. La figura superior corresponde a un sistema con $V_{\text{syn}} = 0\,\text{mV}$ en el cual las neuronas se sincronizan en fase (interacción excitatoria), y la figura inferior corresponde a un sistema con $V_{\text{syn}} = -80\,\text{mV}$ en el cual se sincronizan en antifase (interacción inhibitoria).

Este efecto puede verse más claramente en la Fig. 2, en la cual se muestra el desfasaje entre los disparos de ambas neuronas para ambos casos de potencial sináptico, en función de g_{syn}. Se registró el desfasaje para distintos valores de $g_{\text{syn}} \in \left[0\,\text{mS/cm}^2, 2\,\text{mS/cm}^2\right]$ separados cada $0{,}1\,\text{mS/cm}^2$. A excepción de los g_{syn} más chicos, correspondientes al acoplamiento nulo/muy chico entre neuronas, vemos que luego del transitorio las neuronas se sincronizan en fase para el caso $V_{\text{syn}} = 0\,\text{mV}$ y antifase para $V_{\text{syn}} = -80\,\text{mV}$, concordando con lo visto en la Fig. 1. Cabe destacar que en el caso de $g_{\text{syn}} = 0$ las neuronas están desacopladas, por lo que nunca se sincronizan.

Por último en la Fig. 3 se ilusta la tasa de disparo en función de g_{syn}, para ambos casos de potencial sináptico. Se puede ver como en ambos casos la tasa de disparo disminuye cuanto más fuerte es el acoplamiento entre las neuronas, y en particular disminuye más rápido cuando la interacción es excitatoria (potencial sináptico nulo).

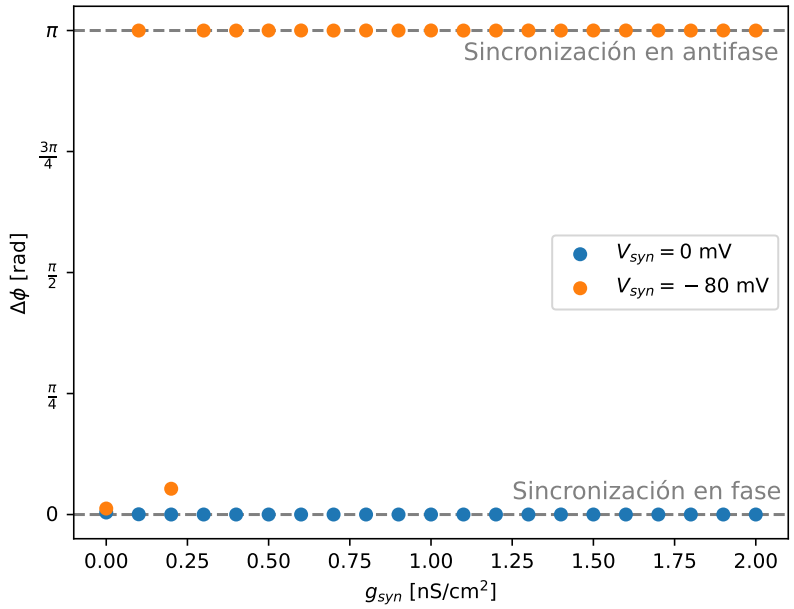

Figura 2: Desfasaje entre los disparos de ambas neuronas, en función de g_{syn}. Se observa como, al estar acopladas, las neuronas se sincronizan en fase cuando $V_{\text{syn}} = 0\,\text{mV}$ y en antifase cuando $V_{\text{syn}} = -80\,\text{mV}$.

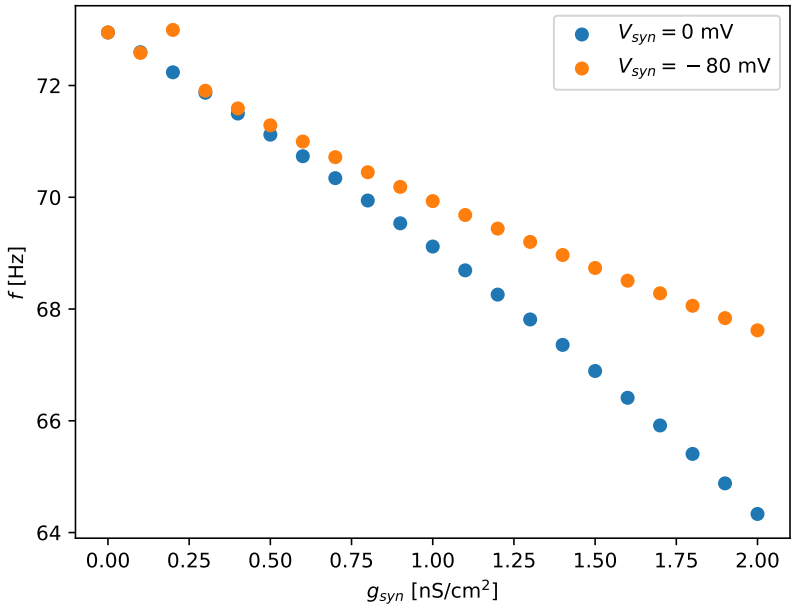

Figura 3: Tasa de disparo en función de g_{syn}, para ambos potenciales sinápticos. Se observa que en ambos casos la tasa de disparo disminuye al aumentar g_{syn}, y en particular ocurre más rápido cuando la interacción es excitatoria.

EJERCICIO 2

Se estudió la dinámica de un sistema con dos poblaciones de neuronas (excitatorias e inhibitorias) en las cuales se modelan las tasas de disparo de las poblaciones con una relación $f - I$ semilineal mediante las siguientes ecuaciones diferenciales acopladas:

$$\begin{cases} \tau \frac{df_e}{dt} = -f_e + S(g_{ee} f_e - g_{ei} f_i + I_e) \\ \tau \frac{df_i}{dt} = -f_i + S(g_{ie} f_e - g_{ii} f_i + I_i)\,, \end{cases}$$

donde $S(f) = fH(f)$, siendo $H(f)$ la función de Heaviside.

Lo primero que se ve es que, para que haya actividad, debe cumplirse que el argumento de ambas funciones de Heaviside sea mayor que 0, es decir

$$g_{ee}f_e - g_{ei}f_i + I_e > 0$$

$$g_{ie}f_e - g_{ii}f_i + I_i > 0$$

ya que en caso contrario las tasas de disparo de ambas poblaciones decaen exponencialmente a 0.

Luego estudio los equilibrios de las ecuaciones (con la condición previamente mencionada). Para ello, igualo las derivadas a 0 y resuelvo para f_e y f_i:

$$\begin{cases} (g_{ee}-1)f_e - g_{ei}f_i + I_e = 0 \\ g_{ie}f_e - (g_{ii}+1)f_i + I_i = 0\,. \end{cases}$$

Resolviendo numéricamente el sistema de ecuaciones, obtenemos el único equilibrio del sistema:

$$(f_e, f_i)^* = \left(\frac{(g_{ii}+1)I_e - g_{ei}I_i}{1 + g_{ei}g_{ie} + g_{ii} - g_{ee}(1+g_{ii})}, \frac{(1-g_{ee})I_i + g_{ie}I_e}{1 + g_{ei}g_{ie} + g_{ii} - g_{ee}(1+g_{ii})} \right).$$

Ahora busco la condición para la cual este equilibrio es estable. Para ver en que caso es un equilibrio estable estudio la matriz Jacobiana del sistema, que se observa debajo:

$$J = \frac{1}{\tau} \begin{pmatrix} g_{ee}-1 & -g_{ei} \\ g_{ie} & -(g_{ii}+1) \end{pmatrix}.$$

Para que el equilibrio sea estable debe cumplirse que $\mathrm{Det}(J) > 0$ y $\mathrm{Tr}(J) < 0$, es decir

$$\begin{cases} \frac{1}{\tau^2}\left[g_{ie}g_{ei} - (g_{ee}-1)(g_{ii}+1)\right] > 0 \\ \frac{1}{\tau}\left(g_{ee} - g_{ii} - 2\right) < 0 \end{cases},$$

por lo que finalmente obtenemos las siguientes condiciones de estabilidad:

$$\begin{cases} g_{ie}g_{ei} - g_{ee}g_{ii} - g_{ee} + g_{ii} + 1 > 0 \\ g_{ee} - g_{ii} - 2 < 0 \end{cases}.$$

APÉNDICE

Código Ejercicio 1

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

###### Definiciones y funciones ######

dt=0.001 # Paso de tiempo
tau = 3 #ms
C=1 #uF/cm^2
v_syn=0 #mV
g_syn=np.linspace(0,2,21) #uS/cm^2
I_ext=12 #nA
g_Na=120 #mS/cm^2
g_K=36 #mS/cm^2
g_Cl=0.3 #mS/cm^2
v_Na=50 #mV
```

```
v_K=-77 #mV
v_Cl=-54.4 #mV

def a_m(v):
    return 0.1*(v+40)/(1-np.exp(-(v+40)/10))
def b_m(v):
    return 4*np.exp(-(v+65)/18)
def a_h(v):
    return 0.07*np.exp(-(v+65)/20)
def b_h(v):
    return 1/(1+np.exp(-(v+35)/10))
def a_n(v):
    return 0.01*(v+55)/(1-np.exp(-(v+55)/10))
def b_n(v):
    return 0.125*np.exp(-(v+65)/80)
def m_inf(v):
    return a_m(v)/(a_m(v)+b_m(v))
def h_inf(v):
    return a_h(v)/(a_h(v)+b_h(v))
def n_inf(v):
    return a_n(v)/(a_n(v)+b_n(v))
def tau_m(v):
    return 1/(a_m(v)+b_m(v))
def tau_h(v):
    return 1/(a_h(v)+b_h(v))
def tau_n(v):
    return 1/(a_n(v)+b_n(v))
def s_inf(v):
    return 0.5*(1+np.tanh(v/5))

class Neuron:
    def __init__(self, v, m, h, n, s,g_):
        self.v=[v]
        self.m=[m]
        self.h=[h]
        self.n=[n]
        self.s=[s]
        self.g_syn=g_

    def get_v(self):
        return self.v

    def update(self, v_pre):
        self.v.append(self.v[-1]+(dt/C)*(I_ext-g_Na*self.m[-1]**3*self.h[-1]*(self.v[-1]-v_Na)\
        -g_K*self.n[-1]**4*(self.v[-1]-v_K)-g_Cl*(self.v[-1]-v_Cl)-self.g_syn*self.s[-1]\
        *(self.v[-1]-v_syn)))
        self.m.append(self.m[-1]+dt*(m_inf(self.v[-1])-self.m[-1])/tau_m(self.v[-1]))
        self.h.append(self.h[-1]+dt*(h_inf(self.v[-1])-self.h[-1])/tau_h(self.v[-1]))
        self.n.append(self.n[-1]+dt*(n_inf(self.v[-1])-self.n[-1])/tau_n(self.v[-1]))
        self.s.append(self.s[-1]+dt*(s_inf(v_pre)-self.s[-1])/tau)

    def get_all(self):
        return self.v, self.m, self.h, self.n, self.s


def get_period(n):
    peaks, _ = sig.find_peaks(n.get_v(),height=0)
    return np.mean(np.diff(peaks[-10:])), peaks[-10:]
```

```
def get_desfase(n1,n2):
    return 2*np.pi*np.abs((np.mean(get_period(n2)[1]-get_period(n1)[1]))/get_period(n1)[0])

###### Potencial de membrana vs t ######

seed1=np.random.uniform(-30,-10)
seed2=np.random.uniform(-30,-10)
n1=Neuron(seed1,0,0,0,0,2)
n2=Neuron(seed2,0,0,0,0,2)

fig, (ax1, ax2) = plt.subplots(2, sharex=True)
v_syn=0
for i in range(2000000):
    n1.update(n2.get_v()[-1]), n2.update(n1.get_v()[-1])

ax1.plot([i*dt for i in range(len(n1.get_v()))],n1.get_v(), label='Neurona 1')
ax1.plot([i*dt for i in range(len(n2.get_v()))],n2.get_v(), label='Neurona 2')
ax1.set_title('Sincronizacin en fase ($V_{syn}=0\,$mV)')
ax1.set_ylabel('$V$ (mV)')
ax1.set_xlim(1800,2000)
ax1.legend()

v_syn=-80
seed1=np.random.uniform(-30,-10)
seed2=np.random.uniform(-30,-10)
n1=Neuron(seed1,0,0,0,0,2)
n2=Neuron(seed2,0,0,0,0,2)
for i in range(2000000):
    n1.update(n2.get_v()[-1]), n2.update(n1.get_v()[-1])

ax2.plot([i*dt for i in range(len(n1.get_v()))],n1.get_v(), label='Neurona 1')
ax2.plot([i*dt for i in range(len(n2.get_v()))],n2.get_v(), label='Neurona 2')
ax2.set_title('Sincronizacin en antifase ($V_{syn}=-80\,$mV)')
ax2.set_xlabel('$t$ (ms)')
ax2.set_ylabel('$V$ (mV)')
ax2.legend()

plt.show()

###### Simulaciones para distintos g_syn y v_syn ######

v_syn=0 #mV
desfasajeV0=[]
tasaV0=[]
for g in g_syn:
    seed1=np.random.uniform(-30,-10)
    seed2=np.random.uniform(-30,-10)
    n1=Neuron(seed1,0,0,0,0,g)
    n2=Neuron(seed2,0,0,0,0,g)
    for i in range(2000000):
        n1.update(n2.get_v()[-1]), n2.update(n1.get_v()[-1])
    desfasajeV0.append(get_desfase(n1,n2))
    tasaV0.append(1/get_period(n1)[0])

v_syn=-80 #mV
desfasajeV_80=[]
tasaV_80=[]
```

```
for g in g_syn:
    seed1=np.random.uniform(-30,-10)
    seed2=np.random.uniform(-30,-10)
    n1=Neuron(seed1,0,0,0,0,g)
    n2=Neuron(seed2,0,0,0,0,g)
    for i in range(2000000):
        n1.update(n2.get_v()[-1]), n2.update(n1.get_v()[-1])
    desfasajeV_80.append(get_desfase(n1,n2))
    tasaV_80.append(1/get_period(n1)[0])

###### Figura desfasajes ######

fig=plt.figure()
ax=fig.add_subplot(111)
ax.scatter(g_syn, desfasajeV0,label="$V_{syn}=0$ mV")
ax.scatter(g_syn, desfasajeV_80,label="$V_{syn}=-80$ mV")
ax.set_xlabel("$g_{syn}$ [nS/cm$^2$]")
ax.set_ylabel("$\Delta \phi$ [rad]")
ax.set_yticks(ticks=[0, np.pi/4,np.pi/2, 3*np.pi/4,np.pi])
ax.set_yticklabels(labels=[r'$0$', r'$\frac{\pi}{4}$',r'$\frac{\pi}{2}$',
    r'$\frac{3\pi}{4}$', r'$\pi$'])
ax.axhline(y=np.pi, color='grey', linestyle='--',zorder=0)
ax.axhline(y=0, color='grey', linestyle='--',zorder=0)
ax.text(1.25, 0.1, "Sincronizacin en fase", fontsize=12, color='grey')
ax.text(1.10, 2.95, "Sincronizacin en antifase", fontsize=12, color='grey')
ax.legend()

plt.show()

###### Figura tasas de disparo ######

plt.scatter(g_syn, 1000*np.array(tasaV0)/dt,label="$V_{syn}=0$ mV")
plt.scatter(g_syn, 1000*np.array(tasaV_80)/dt,label="$V_{syn}=-80$ mV")
plt.xlabel("$g_{syn}$ [nS/cm$^2$]")
plt.ylabel("$f$ [Hz]")
plt.legend()

plt.show()
```