Dynamic Foundations of Neuroscience Project 2

The Morris Leccar Model

Kevin Slote

Georgia State University Department of Mathematics and Statistics

February 2023

Moris-Lecar Model

Steady-state values of gating variables:

$$M_\infty = \frac{1}{2}\left(1 + \tanh\left(\frac{V - V_1}{V_2}\right)\right) \tag{1}$$

$$N_\infty = \frac{1}{2}\left(1 + \tanh\left(\frac{V - V_3}{V_4}\right)\right) \tag{2}$$

Differential equations for membrane potential and gating variables:

$$\frac{dV}{dt} = \frac{-g_{Ca} \cdot m_\infty(V) \cdot (V - V_{Ca}) - g_K \cdot w \cdot (V - V_K) - g_L \cdot (V - V_L) + I_{app}}{C} \tag{3}$$

$$\frac{dw}{dt} = r(V) \cdot (w_\infty(V) - w) \tag{4}$$

Lyaponov Exponent

The algorithm uses delay embedding to reconstruct the dynamics of the data and find the closest neighbor for each vector using euclidean distance. The distances between the vectors increase according to a power law, which is related to the highest Lyapunov exponent. To calculate this exponent, the logarithm of the distance trajectory is taken, which gives a set of lines whose slope is an approximation of lambda. The mean log trajectory is extracted and a straight line is fitted to give the desired parameter lambda.

$$X_i = [x_i, x_{(i+lag)}, x_{(i+2*lag)}, \dots, x_{(i+(emb_dim-1)*lag)}]$$

$$d_i(k) = c * e^{\lambda * k}$$

$$log(d_i(k)) = log(c) + \lambda * k$$

Tables and Figures

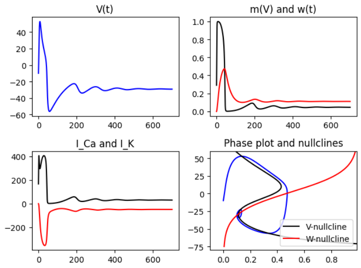


Figure: With Initial conditions $V1 = -2$, $V2 = 18$, $V3 = 2$, $V4 = 30$, $gCa = 4.4$, $gK = 4$, $gL = 2$, $VCa = 120$, $VK = -84$, $VL = -60$, $C = 20$, $Iapp = 0$. This system is pre-Hopf-bifurcation.

Saddle-Node One (Excitability 1)

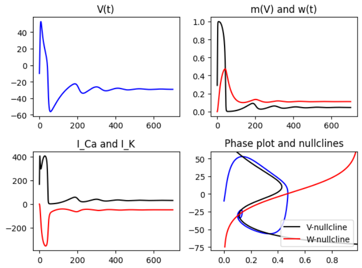


Figure: With Initial conditions $V1 = -2$, $V2 = 18$, $V3 = 2$, $V4 = 30$, $gCa = 5.4$, $gK = 4$, $gL = 2$, $VCa = 120$, $VK = -84$, $VL = -60$, $C = 20$, $Iapp = 0$. This system is pre-Hopf-bifurcation.

Changing gCa moved the w nullcline up and caused the system to shift to a saddle node bifurcation. The lyaponov exponent is $\delta = -0.000876$. Positive imples subcritical.

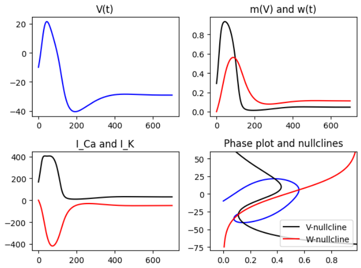
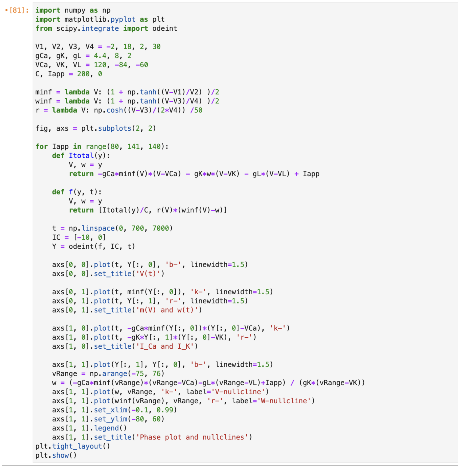


Figure: With Initial conditions $V1 = -2$, $V2 = 18$, $V3 = 2$, $V4 = 30$, $gCa = 5.4$, $gK = 4$, $gL = 2$, $VCa = 120$, $VK = -84$, $VL = -60$, $C = 20$, $Iapp = 0$. This system is pre-Hopf-bifurcation.

Changing C moved the w nullcline up and caused the system to shift to a saddle node bifurcation. The lyaponov exponent is $\delta = -0.000876$. Negative implies supercritical.

Python Code



```
•[81]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

V1, V2, V3, V4 = -2, 18, 2, 30
gCa, gK, gL = 4.4, 8, 2
VCa, VK, VL = 120, -84, -60
C, Iapp = 200, 0

minf = lambda V: (1 + np.tanh((V-V1)/V2) )/2
winf = lambda V: (1 + np.tanh((V-V3)/V4) )/2
r = lambda V: np.cosh((V-V3)/(2*V4)) /50

fig, axs = plt.subplots(2, 2)

for Iapp in range(80, 141, 140):
    def Itotal(y):
        V, w = y
        return -gCa*minf(V)*(V-VCa) - gK*w*(V-VK) - gL*(V-VL) + Iapp

    def f(y, t):
        V, w = y
        return [Itotal(y)/C, r(V)*(winf(V)-w)]

    t = np.linspace(0, 700, 7000)
    IC = [-10, 0]
    Y = odeint(f, IC, t)

    axs[0, 0].plot(t, Y[:, 0], 'b-', linewidth=1.5)
    axs[0, 0].set_title('V(t)')

    axs[0, 1].plot(t, minf(Y[:, 0]), 'k-', linewidth=1.5)
    axs[0, 1].plot(t, Y[:, 1], 'r-', linewidth=1.5)
    axs[0, 1].set_title('m(V) and w(t)')

    axs[1, 0].plot(t, -gCa*minf(Y[:, 0])*(Y[:, 0]-VCa), 'k-')
    axs[1, 0].plot(t, -gK*Y[:, 1]*(Y[:, 0]-VK), 'r-')
    axs[1, 0].set_title('I_Ca and I_K')

    axs[1, 1].plot(Y[:, 1], Y[:, 0], 'b-', linewidth=1.5)
    vRange = np.arange(-75, 76)
    w = (-gCa*minf(vRange)*(vRange-VCa)-gL*(vRange-VL)+Iapp) / (gK*(vRange-VK))
    axs[1, 1].plot(w, vRange, 'k-', label='V-nullcline')
    axs[1, 1].plot(winf(vRange), vRange, 'r-', label='W-nullcline')
    axs[1, 1].set_xlim(-0.1, 0.99)
    axs[1, 1].set_ylim(-80, 60)
    axs[1, 1].legend()
    axs[1, 1].set_title('Phase plot and nullclines')
plt.tight_layout()
plt.show()
```

Figure: code available at https://github.com/kslote1/morris-leccar.git