Simple Neuron Action Potential Stimulator

What is this project about

My project is a neuron action potential simulator that simulates how neurons fire when they get stimulated. The main idea is to track the membrane potential and how it changes when sodium and potassium ions move in and out. The neuron has different states like resting, depolarizing (when it fires), repolarizing (going back down), and recovering. I also use the Nernst equation and Goldman equation to calculate the voltage based on ion concentrations and ion permeability.

Features implemented

Here's what I implemented:

- **A SimpleNeuron class** that tracks the neuron's voltage, state, ion concentrations, and channel gates
- **Hodgkin-Huxley style ion channel gating** with physically meaningful m, h, and n gates representing protein subunits
- **Simulation of sodium and potassium ion channels** with voltage-dependent opening and closing dynamics (h-gate implements the refractory period)
- **Conductance-based ionic current calculations** (I_Na, I_K, I_leak) using the Hodgkin-Huxley formulation with proper driving forces
- **Dynamic ion concentration tracking** where ionic currents cause real-time changes in Na^+ and K^+ concentrations inside and outside the cell
- **Sodium-potassium pump (Na^+/K^+-ATPase)** that runs in the background using Michaelis-Menten kinetics to restore ion concentrations (3 Na^+ out, 2 K^+ in per cycle)
- **Goldman-Hodgkin-Katz equation** for calculating membrane potential based on ion concentrations and permeabilities
- **A stimulation function** where you can inject current to increase m-gate opening probability and trigger action potentials
- **Absolute refractory period** implemented through h-gate dynamics - the neuron can't fire again until h-gate recovers (>0.8)
- **Four-state machine** (resting → depolarizing → repolarizing → recovering) with voltage-dependent state transitions
- **Exponential decay dynamics** for stimulus response and channel recovery periods
- **Step-by-step update function** that advances the simulation with configurable time steps (default dt=0.01 ms)
- **Plotting functionality** to visualize voltage over time and observe action potential waveforms
- **Support for multiple action potentials** in sequence with configurable stimulus timing, strength, and duration
- **Debug output** showing real-time ion concentrations, equilibrium potentials, permeabilities, channel states, pump status, and current neuron state

Overall Introduction and Current Limitations

We tried two versions. The first is the simplest version, where most of the parameters are based on experience, which means that we tried several times and found the best parameters based on proportions.

Even though it works well for the result data, it still doesn't have physical meaning behind it. Hence we modified it. First, we standardized all the units (it requires some complex unit conversions in the equations). Then we chose to import some HH equation ideas—we used the detailed gate concept (which represents protein subunits on the membrane) and used the conductance equation to calculate the ion current accurately. Then we use the ion current to calculate the change in concentration (this is based on the equation: $\Delta concentration = flux/depth$, where $flux = I/(z \cdot F)$).

Also, we used the gate concept to change the permeability of ions (but the discrete changing is not accurate, though I kept it since it's more physical). In addition, we changed the pump mechanism—we use the Michaelis-Menten equation to calculate the pump rate based on the concentration of substrates.

Then we met a problem. Since we were using the Goldman equation to calculate the membrane potential, but we were also using the gate and current mechanisms. In classic HH models, people use the membrane capacitance equation to calculate voltage instead of the Goldman equation. We tried to change it to the membrane capacitance equation, but the result was strange without using an ODE solver (we don't want to use it right now, so it's deliberate) and doesn't suit the current state machine. Hence we gave up and changed back to the previous method using the Goldman equation. Since we are using a mixed approach, the result is less accurate than the simplest one, but we now have physical meaning behind most of the parameters. There are still some problems, which we will mention in the next section. To solve the current limitations, we have to use the full HH equations, which means we have to use an ODE solver. We will leave it for after we study Differential Equations and Numerical Methods properly.

Code Interpretation

Libraries used

- **numpy** - for math calculations and the logarithm functions in the equations
- **matplotlib** - for plotting the voltage graph

Functions

`__init__(self)` **function**: We define the variables and parameters that will be used in the subsequent code.

`calculate_eq(self, ion)` **function**: We use the Nernst equation to calculate the equilibrium potentials of Na^+ and K^+.

`calculate_voltage(self)` **function**: We use the Goldman equation to calculate the membrane potential.

`pump_cycle(self)` **function**: We use Michaelis-Menten kinetics to calculate the pump rate and update the concentration changes.

`p_cal(self, ion)` **function**: We calculate permeability based on gate states. Note that since we chose to maintain the n-gate of potassium, the p_K will pull down the p_Na, preventing it from exceeding +30 mV (which is slightly lower than experimental data).

`ion_current(self, ion)` **function**: We calculate the ionic currents (I_Na, I_K, I_leak) using the conductance-based model. The current is determined by maximum conductance, gate states, and the driving force (V_m - E_ion).

`d_concentration_ms(self, ion)` **function**: We calculate the rate of concentration change (mM/ms) caused by ionic currents flowing through channels. The conversion factor `fac` translates current into concentration changes.

`can_stimulate(self)` **function**: We check whether the neuron is ready to be stimulated by verifying that the h-gate is sufficiently recovered (>0.8) and the neuron is in the resting state, implementing the absolute refractory period constraint.

`apply_stimulation(self, strength)` **function**: We stimulate by increasing the m-gate opening probability. The `self.Na_stim_m` increases slightly each time, which affects the ion current and permeability, thereby affecting the voltage. The `self.Na_stim_m` is updated in the `update(self)` function at every time step. After the stimulation period, it undergoes exponential decay.

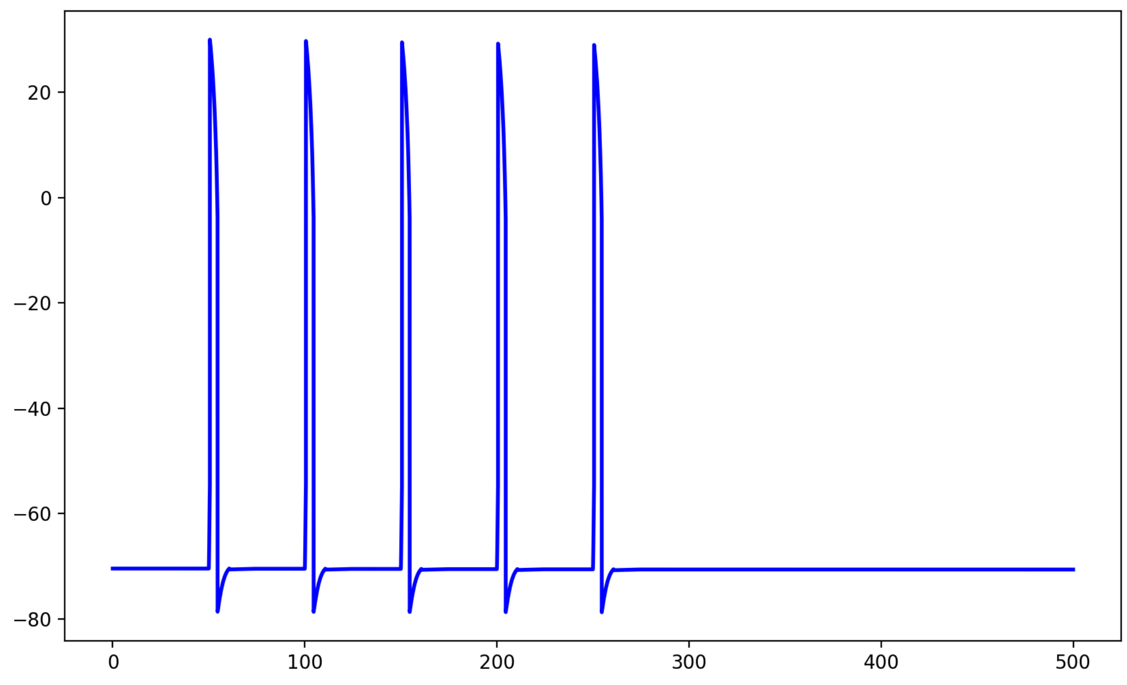
`update(self)` **function**: This is the state machine that runs at every time step after being called, until the time steps are exhausted. We have four states, and voltage is mainly used as the switching condition. The absolute refractory period is implemented here through changes in `self.Na_h_gate` and conditional evaluation of it.

`current_ion_state(self)` **function**: We print out the current state of the neuron including ion concentrations, equilibrium potentials, permeabilities, membrane potential, channel states, pump status, and the current state for debugging purposes.

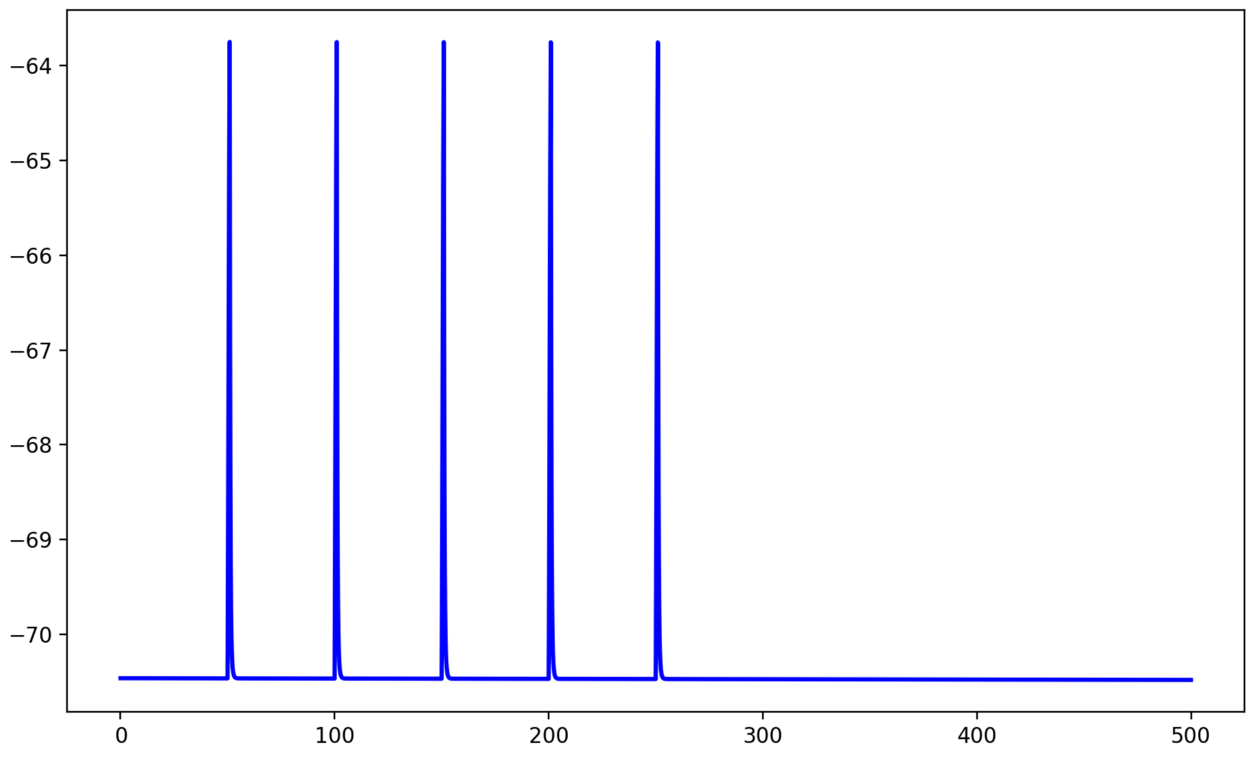
`stimulate_and_plot(sti_str, sti_interval, ap_num, duration, sti_dur=1.0, dt=0.01)`: We simulate current injection by generating square current pulses—we call `apply_stimulation(self, strength)` at every time step during the stimulation time period.

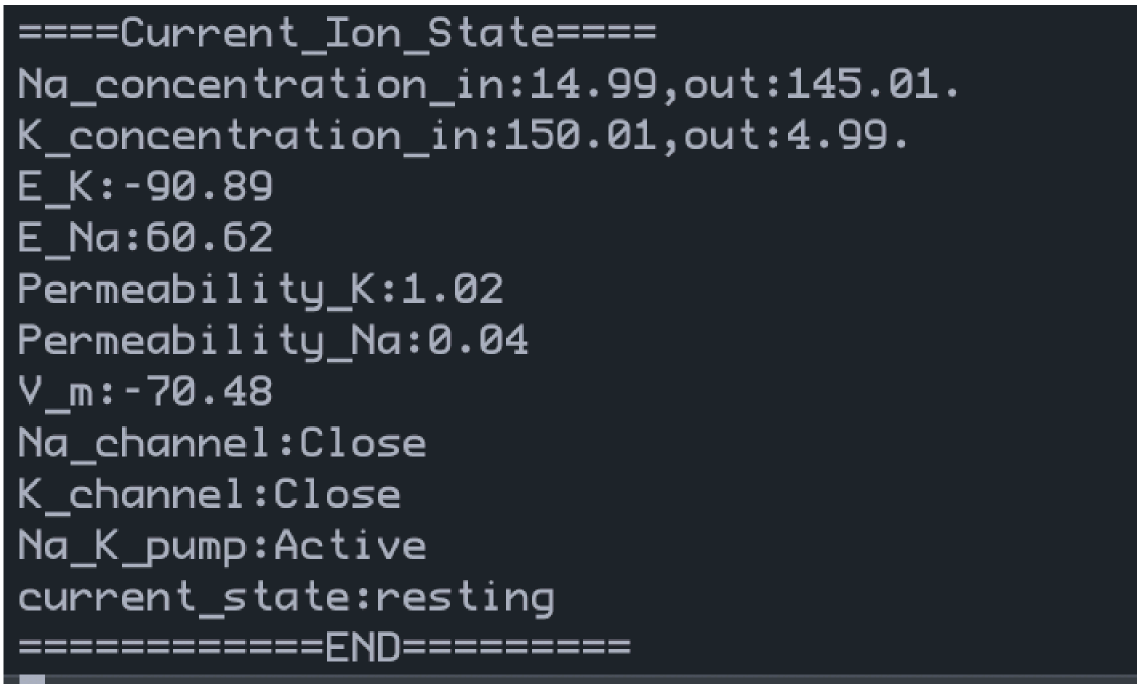
Output exhibition

The current version:
We use 60mA current to stimulate the neuron.



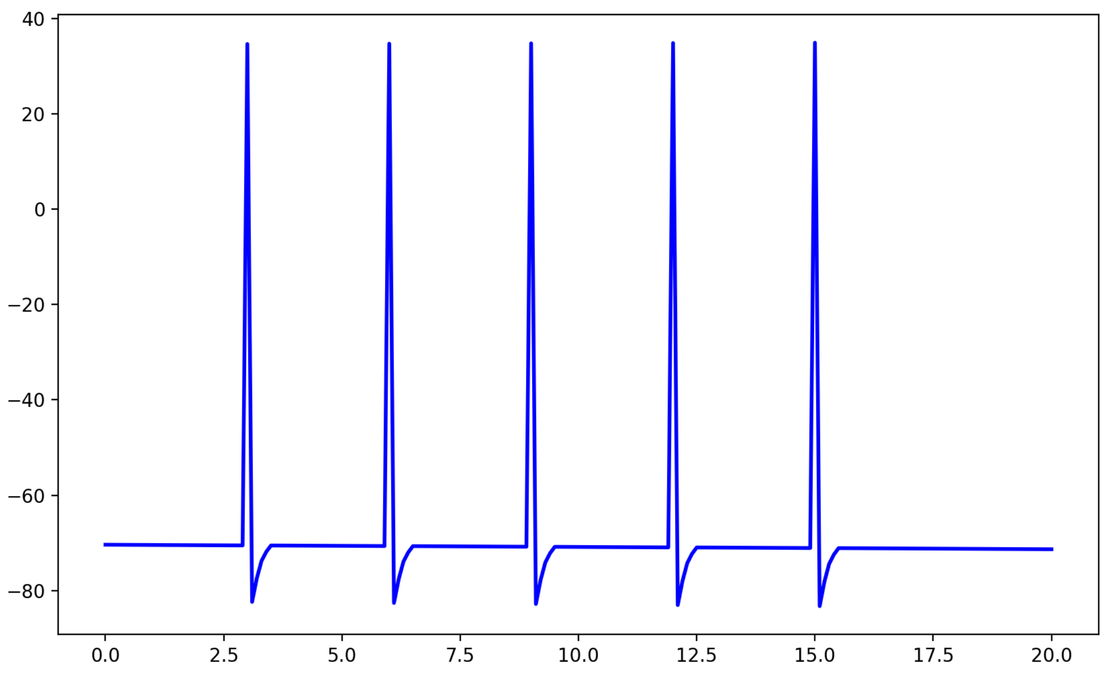
We use 30mA to stimulate the neuron. Note that there is no action potential.





```
====Current_Ion_State====
Na_concentration_in:14.99,out:145.01.
K_concentration_in:150.01,out:4.99.
E_K:-90.89
E_Na:60.62
Permeability_K:1.02
Permeability_Na:0.04
V_m:-70.48
Na_channel:Close
K_channel:Close
Na_K_pump:Active
current_state:resting
============END=========
```

The previous version:



Note that this one seems more accurate, the reason I mentioned in the section: Overall Introduction and Current Limitations.

How does it work?

Here is neuroscience information to help people understand the logic behind the code:
The neuron receives enough current to depolarize the membrane potential beyond the threshold, and then the action potential starts. In our code, we choose to increase the permeability of Na^+. At every step, we calculate the voltage (which is the membrane potential) using the Goldman equation. When it increases enough, we change the state to depolarizing. (The permeability can be regarded as the channel permeability, or the conductance.)
When the voltage goes beyond the threshold, the m-gate of the Na^+ channel opens rapidly (we increase the permeability of Na^+), then closes (which takes about 2ms). However, the h-gate of Na^+ simultaneously closes (see Figure 4.9). When the h-gate closes and the membrane potential almost reaches +40mV (which is near the equilibrium potential of Na^+), the voltage-gated K^+ channel opens. Note that the K^+ channel is actually stimulated at the same time as the Na^+ channel when current is injected, but it opens with a ~2ms delay. In our code, we simply open the K^+ channel and change the permeability of K^+. During this time, the h-gate of the Na^+ channel is recovering.
When the membrane potential almost reaches the equilibrium potential of K^+, the K^+ channel starts closing (in our code we use exponential decay). When the voltage returns to the resting membrane potential and the h-gate of the Na^+ channel is fully open again, we can stimulate it again. (Notice that there is an absolute

refractory period, which we implement in our code through a conditional check.)

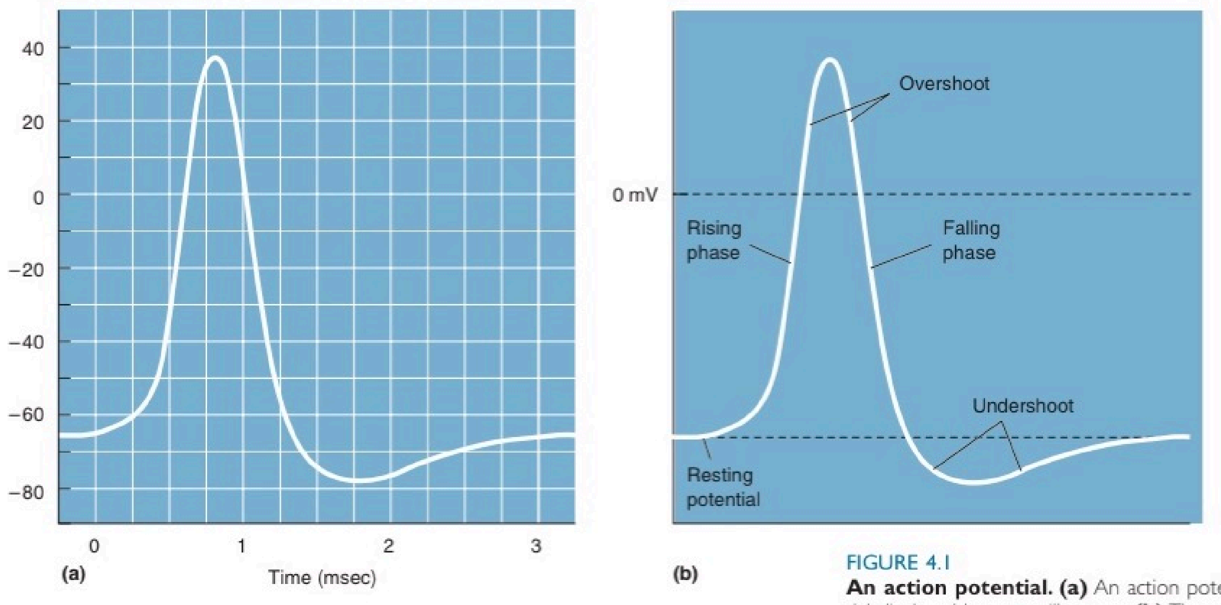
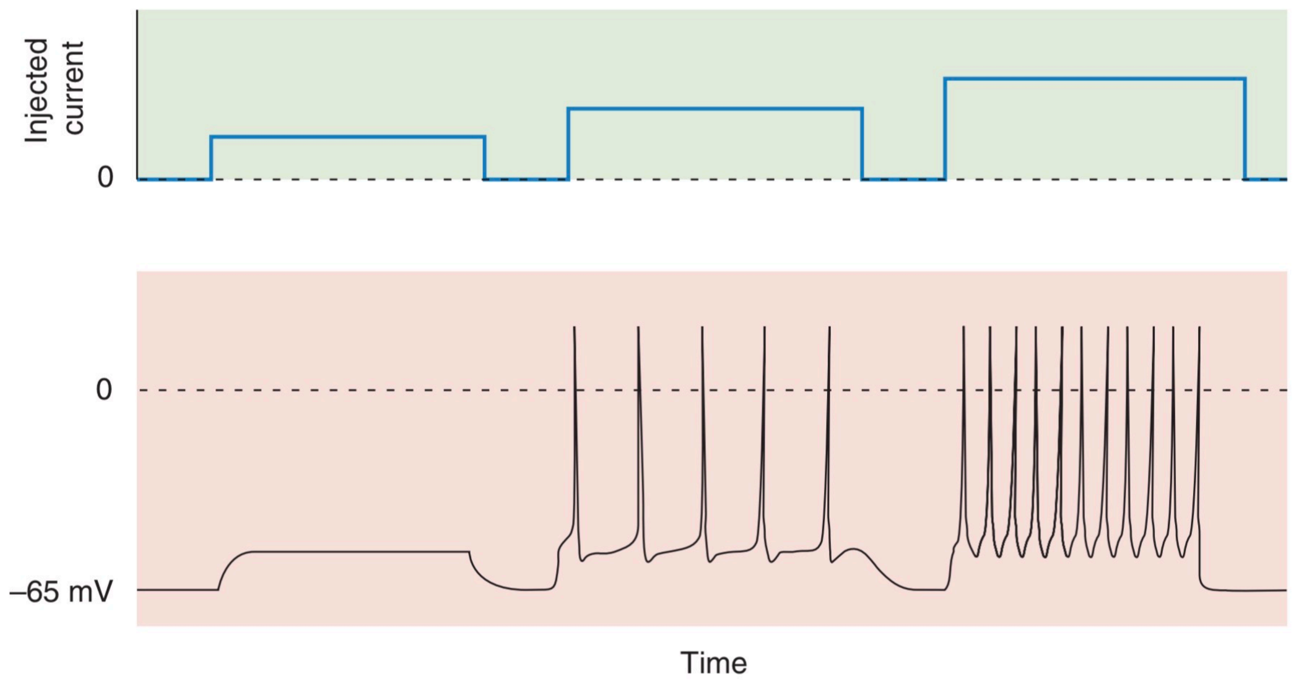


FIGURE 4.1
An action potential. (a) An action potential displayed by an oscilloscope. **(b)** The parts of an action potential.



If injected current does not depolarize the membrane to threshold, no action potentials will be generated.

If injected current depolarizes the membrane beyond threshold, action potentials will be generated.

The action potential firing rate increases as the depolarizing current increases.

This is about the h gate of sodium channel

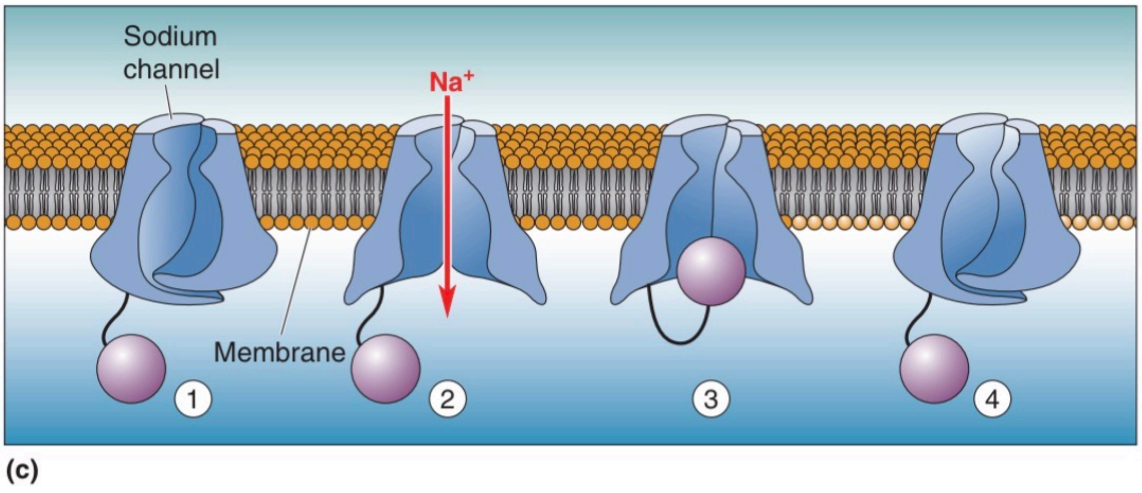


FIGURE 4.9
The opening and closing of sodium channels upon membrane depolarization. **(a)** This trace shows the electrical potential across a patch of membrane. When the membrane potential is changed from −65 to −40 mV, the sodium channels pop open. **(b)** These traces show how three different channels respond to the voltage step. Each line is a record of the electrical current that flows through a single channel. ① At −65 mV, the channels are closed, so there is no current. ② When the membrane is depolarized to −40 mV, the channels briefly open and current flows inward, represented by the downward deflection in the current traces. Although there is some variability from channel to channel, all of them open with little delay and stay open for less than 1 msec. Notice that after they have opened once, they close and stay closed as long as the membrane is maintained at a depolarized V_m. ③ The closure of the sodium channel by steady depolarization is called inactivation. ④ To deinactivate the channels, the membrane must be returned to −65 mV again. **(c)** A model for how changes in the conformation of the sodium channel protein might yield its functional properties. ① The closed channel ② opens upon membrane depolarization. ③ Inactivation occurs when a globular portion of the protein swings up and occludes the pore. ④ Deinactivation occurs when the globular portion swings away and the pore closes by movement of the transmembrane domains.

Mathematical Formulations in the Neuron Action Potential Simulator

---

1. Equilibrium and Membrane Potentials

Nernst Equation

The Nernst equation calculates the equilibrium potential for a specific ion across the membrane:

$$E_{ion} = \frac{RT}{F} \ln \left(\frac{[ion]_{out}}{[ion]_{in}} \right)$$

Applied to Na⁺ and K⁺: E_{Na}, E_K

Constants:

- $R = 8.314$ J/(mol·K) - Gas constant
- $T = 310$ K - Absolute temperature (37°C)
- $F = 96485$ C/mol - Faraday constant
- $\frac{RT}{F} \times 1000 \approx 26.7$ mV at physiological temperature

Goldman-Hodgkin-Katz (GHK) Equation

The GHK equation calculates the membrane potential considering multiple ion species:

$$V_m = \frac{RT}{F} \ln\left(\frac{P_K[K^+]_{out} + P_{Na}[Na^+]_{out}}{P_K[K^+]_{in} + P_{Na}[Na^+]_{in}}\right)$$

Where:

- P_K, P_{Na} - Ion permeabilities
- $[ion]_{in}$, $[ion]_{out}$ - Intracellular and extracellular ion concentrations

---

2. Na⁺/K⁺-ATPase Pump (Michaelis-Menten Kinetics)

The sodium-potassium pump follows Michaelis-Menten kinetics:

$$v_{Na} = \frac{[Na^+]_{in}}{[Na^+]_{in} + K_{Na}}, \quad v_K = \frac{[K^+]_{out}}{[K^+]_{out} + K_K}$$

$$I_{pump} = I_{pump,max} \cdot v_{Na} \cdot v_K$$

Concentration changes (3 Na⁺ out, 2 K⁺ in per cycle):

$$\frac{d[Na^+]_{in}}{dt} = -3 \cdot I_{pump} \cdot fac, \quad \frac{d[K^+]_{in}}{dt} = +2 \cdot I_{pump} \cdot fac$$

Conversion factor: $fac = \frac{10^{-3}}{F \cdot V_{cell}}$, where $V_{cell} = \frac{r_{cell}}{3}$

Parameters:

- $K_{Na} = 10.0$ mM, $K_K = 1.5$ mM - Half-saturation constants
- $I_{pump,max} = 0.5$ μA/cm²
- $r_{cell} = 10^{-3}$ cm

---

3. Hodgkin-Huxley Ion Channel Gating

Sodium channel permeability:

$$P_{Na} = P_{Na,min} + (P_{Na,max} - P_{Na,min}) \cdot m^3 h$$

Potassium channel permeability:

$$P_K = P_{K,max} \cdot n^4 + P_{K,leak}$$

Parameters:

- m - Na⁺ activation gate (3 gates), h - Na⁺ inactivation gate (1 gate)
- n - K⁺ activation gate (4 gates)
- $P_{Na,min} = 0.04$, $P_{Na,max} = 5.0$
- $P_{K,max} = 2.0$, $P_{K,leak} = 1.0$

---

4. Ionic Currents (Conductance-Based Model)

$$I_{Na} = g_{Na,max} \cdot m^3 h \cdot (V_m - E_{Na})$$

$$I_K = g_{K,max} \cdot n^4 \cdot (V_m - E_K)$$

$$I_{leak} = g_{leak} \cdot (V_m - E_{leak})$$

Parameters:

- $g_{Na,max} = 120$ mS/cm², $g_{K,max} = 36$ mS/cm², $g_{leak} = 0.3$ mS/cm²
- $E_{leak} = -70$ mV

Concentration changes from ionic currents:

$$\frac{d[Na^+]_{in}}{dt} = -I_{Na} \cdot fac, \quad \frac{d[K^+]_{in}}{dt} = -I_K \cdot fac$$

---

5. Time-Dependent Gate Dynamics

Exponential decay (stimulus and recovery):

$$m_{stim}(t + \Delta t) = m_{stim}(t) \cdot e^{-\Delta t/\tau_{stim}}$$

$$P_K(t + \Delta t) = 1.0 + (P_K(t) - 1.0) \cdot e^{-k \cdot \Delta t}$$

Linear gate dynamics:

$$h(t + \Delta t) = h(t) \pm \alpha \cdot \Delta t \quad \text{(bounded in [0,1])}$$

Stimulus-induced activation:

$$m_{stim}(t + \Delta t) = \min(1.0, m_{stim}(t) + k \cdot I_{stim} \cdot \Delta t)$$

$$m_{total} = \text{clip}(m_{base} + m_{stim}, 0, 1)$$

Time constants:

- $\tau_{stim} = 0.5$ ms, $\tau_h = 7.0$ ms, $\tau_m = 1.0$ ms
- $k = 0.009$ (stimulus sensitivity), decay rate $= 0.5$

---

6. Membrane Capacitance Equation (Reference)

$$C_m \frac{dV_m}{dt} = I_{stim} - I_{Na} - I_K - I_{leak}$$

Where $C_m = 1.0$ μF/cm²

---

7. Parameters Summary

Initial concentrations: $[Na^+]_{in} = 15$ mM, $[Na^+]_{out} = 145$ mM, $[K^+]_{in} = 150$ mM, $[K^+]_{out} = 5$ mM

Initial gates: $h = 1.0$, $m = 0.05$, $n = 0.32$

Simulation: $\Delta t = 0.01$ ms, $\varepsilon = 10^{-6}$

---

8. State Transitions

Resting → Depolarizing: $V_m \geq -55$ mV
Depolarizing → Repolarizing: $V_m \geq 30$ mV or $h < 0.2$
Repolarizing → Recovering: $V_m \leq -65$ mV
Recovering → Resting: $V_m \leq -65$ mV and $P_K \leq 1.05$

Note: This is a simplified model based on Hodgkin-Huxley (1952) and Goldman-Hodgkin-Katz (GHK) Equation with dynamic ion concentrations and Na^+/K^+-ATPase pump kinetics.