Neuronal Dynamics: A Generalized Leaky Integrate and Fire Model (GLIF)

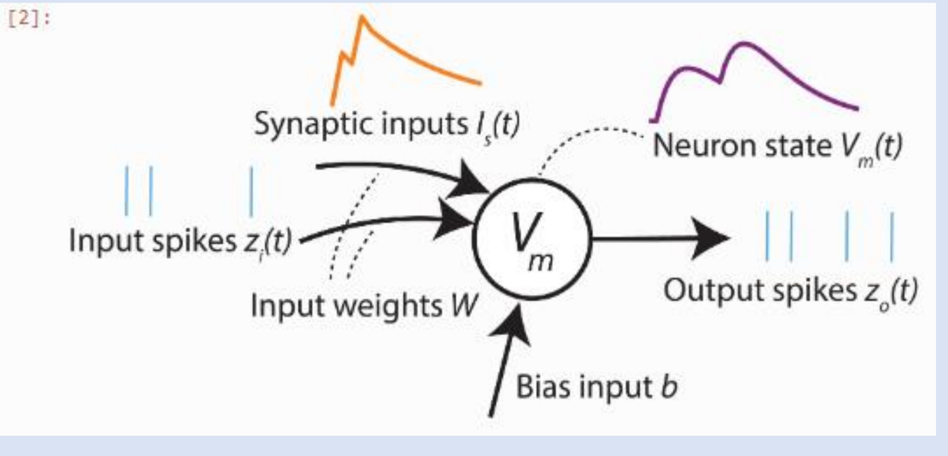
South Carolina

Zach Abdulrahman
Department of Mathematics

Source Code

Background

The Leaky Integrate-and-Fire (LIF) model simplifies neurons to circuits that accumulate signals until a firing threshold is met, incorporating charge leakage over time for efficiency in simulating large neural networks. In contrast, the Hodgkin-Huxley (HH) model offers detailed simulations of neuronal activity, focusing on sodium and potassium ion dynamics during action potentials, essential for understanding neural mechanisms at a molecular level.



Motivation

Choosing a GLIF allows for the efficient simulation of brain activity, preferable over the computationally intensive HH model for large-scale networks and offering greater complexity than a singular LIF neuron for dynamic studies. Exploring the GLIF model can enhance AI algorithms, offer insights into neurological disorder treatments, and innovate brain-computer interface designs.

Dynamics: between spikes

$$I_j'(t) = -k_j I_j(t);\; j = 1, \ldots, N$$

$$V'(t) = \frac{1}{C}\left(I_e(t) + \sum_j I_j(t) - G\left(V(t) - E_L\right)\right)$$

$$\Theta_s'(t) = -b_s \Theta_s(t)$$

Reset: if $V(t) > \Theta_\infty + \Theta_s(t)$

$$I_j(t_+) \leftarrow f_j \times I_j(t_-) + \delta I_j$$

$$V(t_+) \leftarrow E_L + f_v \times \left(V(t_-) - E_L\right) - \delta V$$

$$\Theta_s(t_+) \leftarrow \Theta_s(t_-) + \delta\Theta_s$$

Challenges

The GLIF model, although effective for cerebellar neuron responses, fails to capture the variable firing patterns of CA1 hippocampal neurons, often producing unrealistically regular spikes due to its handling of adaptation currents.

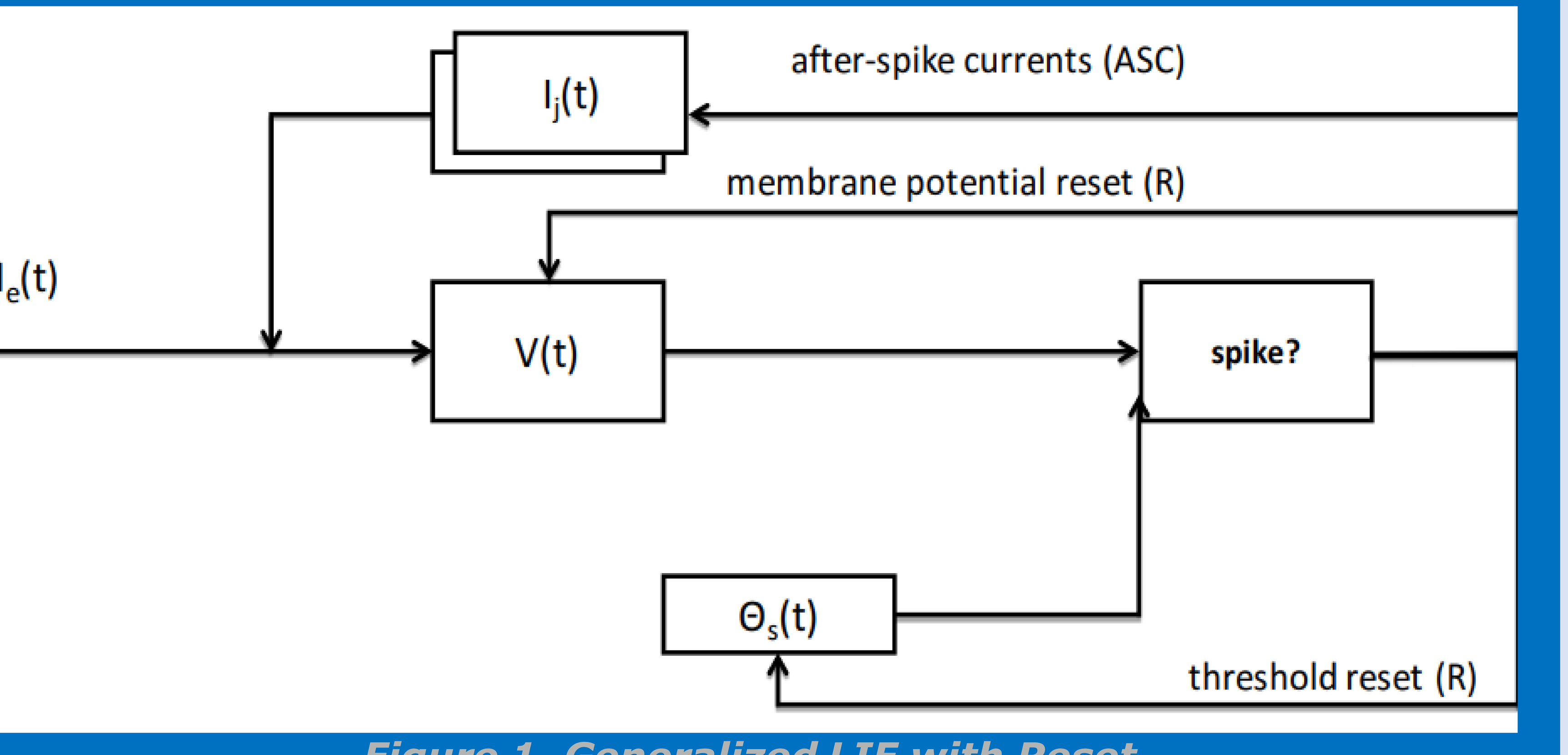


Figure 1. Generalized LIF with Reset

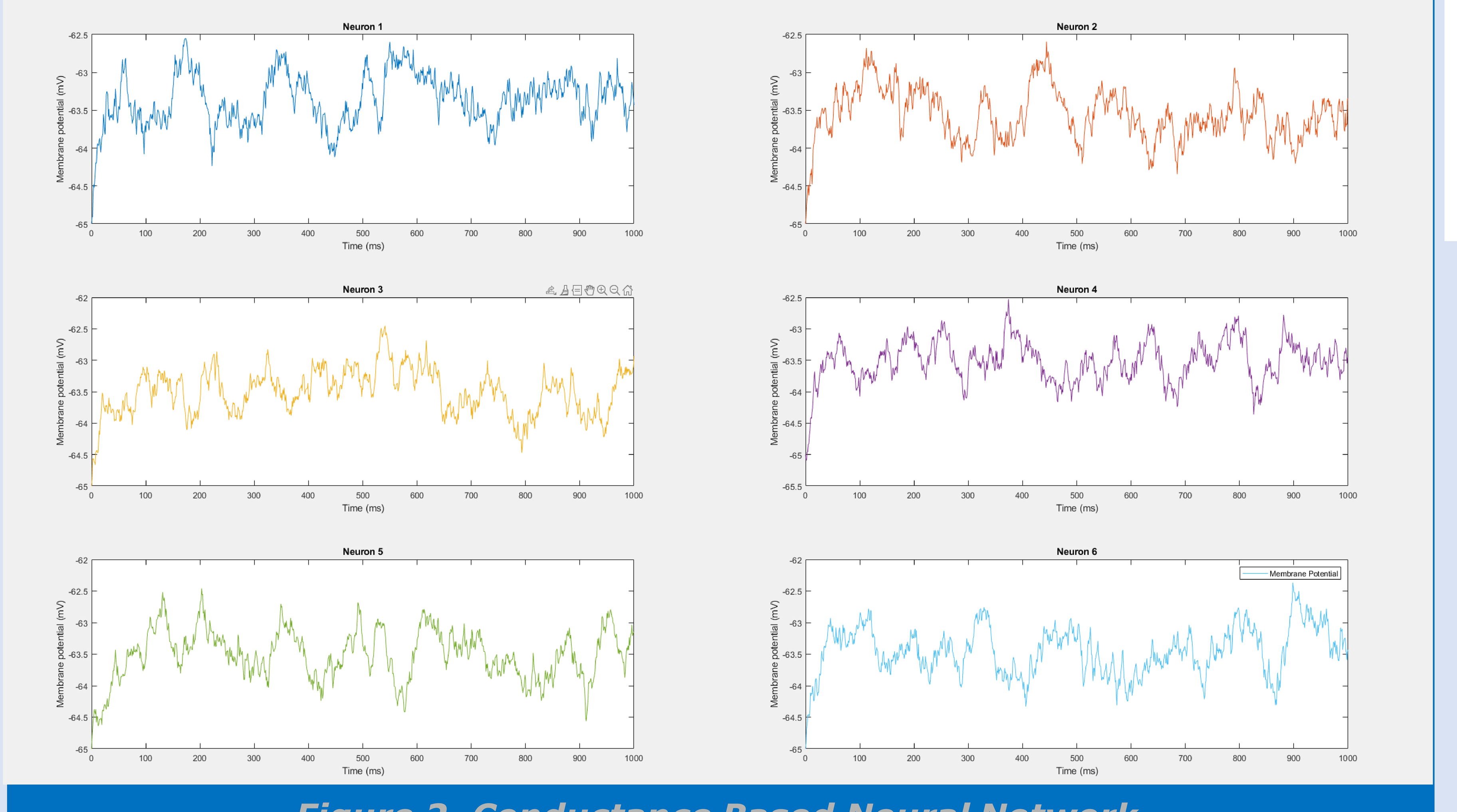


Figure 2. Conductance Based Neural Network

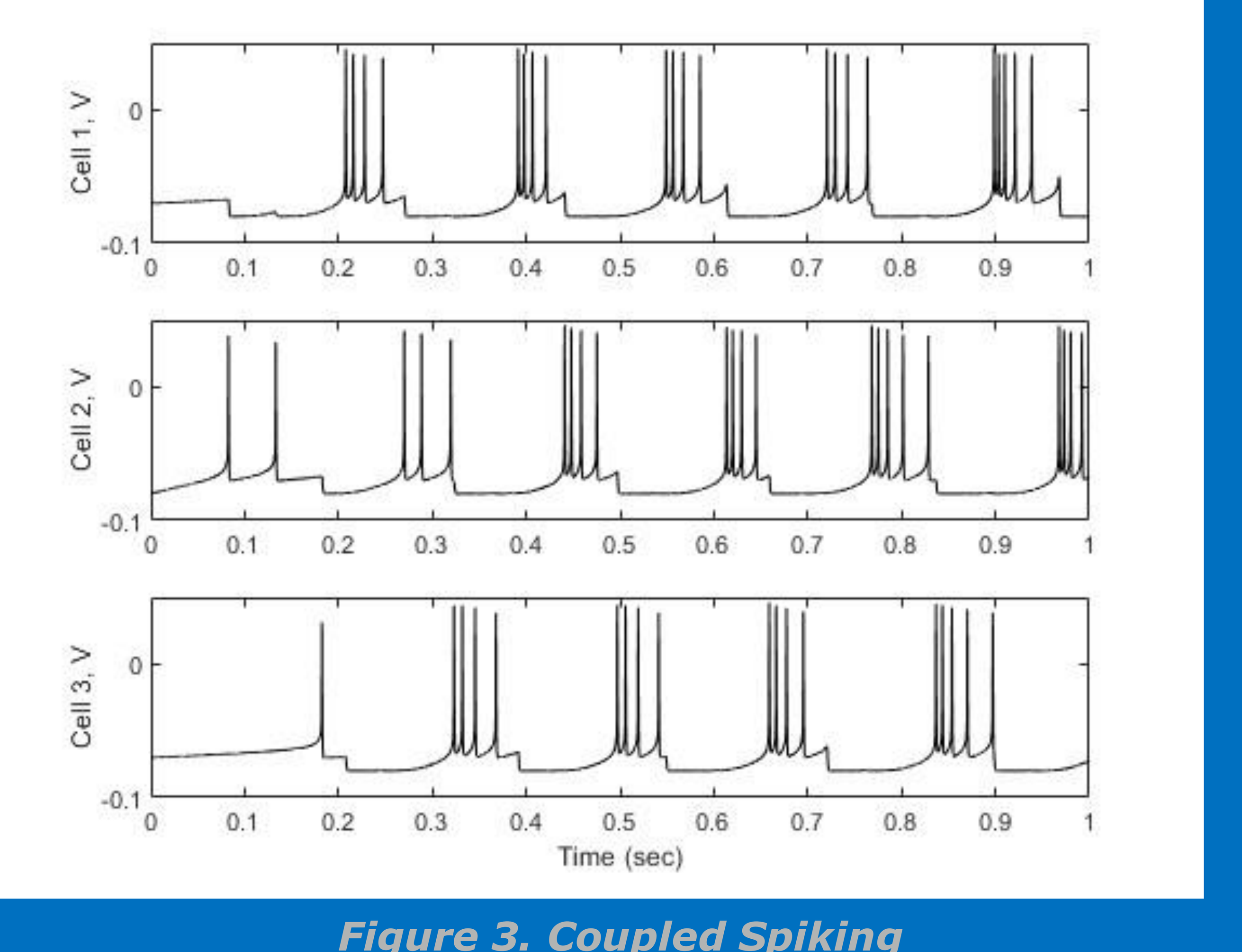


Figure 3. Coupled Spiking

Methods and Approach

- ➢ Define neuron properties and setup simulation framework.
- ➢ Create inter-neuron connections with randomized strengths.
- ➢ Simulate each neuron's response to inputs and noise.
- ➢ Detect and log spikes, then reset neuron potentials.
- ➢ Visualize dynamic neuron behavior and interaction over time.

```
% Simulation loop
for i = 2:length(t)
    for neuron = 1:n
        if V(neuron, i-1) == V_spike
            V(neuron, i) = V_reset; % Reset after spike
        else
            I_syn = sum(W(neuron, :) .* (V(:, i-1) > V_th)') * R_m; % Synaptic current
            I_noise = noise_sigma * randn; % Gaussian noise
            dV = ((E_L - V(neuron, i-1)) + R_m * I_ext + I_syn + I_noise) * dt / tau_m;
            V(neuron, i) = V(neuron, i-1) + dV;

            % Spike event
            if V(neuron, i) >= V_th
                V(neuron, i) = V_spike;
                spikeTimes{neuron} = [spikeTimes{neuron}, t(i)];
            end
        end
    end
end
```

Results and Future Works

- ➢ Synaptic conductance and Gaussian noise critically shape neuronal membrane potential dynamics.
- ➢ Neuronal excitability and action potential thresholds are essential for neural communication, determined by ionic conductance.
- ➢ The interplay between ionic channel conductance and synaptic activity, modulated by noise, drives the emergent behavior of neural networks.

Future works include exploring chimera states, using AI for model optimization, and leveraging parallel computing for faster simulations providing future directions in computing and medicine.