Introduction to Engineering Systems II
Section 09- Professor Goodrich

The Effects of STDP Learning of Hodgkin-Huxley Neurons

Group 06
Justin Skycak
William Boeschenstein
Luke Song
Paul Lee
Gbemisola Okunzua

04/28/2015

X_____

X_____

X_____

X_____

X_____

Abstract

A prominent goal of computational neuroscience is to understand how brain neuronal networks can represent sequences of stimulus patterns and learn new sequences of stimulus patterns. Learned stimulus patterns are believed to be stored in network connectivity, which can change over time, depending on various aspects of network activity. We built a GUI which allows a researcher to stimulate and evolve a biologically realistic 3-neuron network by means of an experimentally observed neural learning rule.

To computationally implement the 3 neurons, we chose to use the Hodgkin-Huxley (HH) (1952) neuron model because it gives a quantitatively accurate description of an action potential, a large voltage fluctuation or "spike" which is believed to be the fundamental mechanism of neural communication. The model consists of several coupled ordinary differential equations that describe the relationship between a neuron's voltage and its sodium and potassium ion channels.

The learning rule, spike-timing dependent plasticity (STDP) (Song et al. 2000), describes how a connection between two neurons is strengthened or weakened according to relative timing of neurons' action potentials or spikes. It can be implemented by updating connection weights (scalar values which encode the strength of a connection) as a function of spike delay time.

In our GUI, the user would enter initial connection strengths (values between -1 and 1) in an initial weight matrix. Then, the user would stimulate any or all of three neurons with an analog stimulus of arbitrary magnitude, duration, and period. The stimulus and each neuron's activity was then plotted on a graph, and the evolved weight matrix was displayed. Our model was verified for by its ability to reproduce stereotypical neural spikes, show expected weight evolutions for simple scenarios, and display characteristics such as adaption to prolonged stimuli.

Overall Concept

We built a platform that can be used to investigate STDP learning in a network of 3 biologically realistic HH neurons. The platform allows researchers to initialize network connectivity as desired and observe the resulting network evolution in response to a chosen input stimulus. Figure 1 illustrates the general idea of the platform.

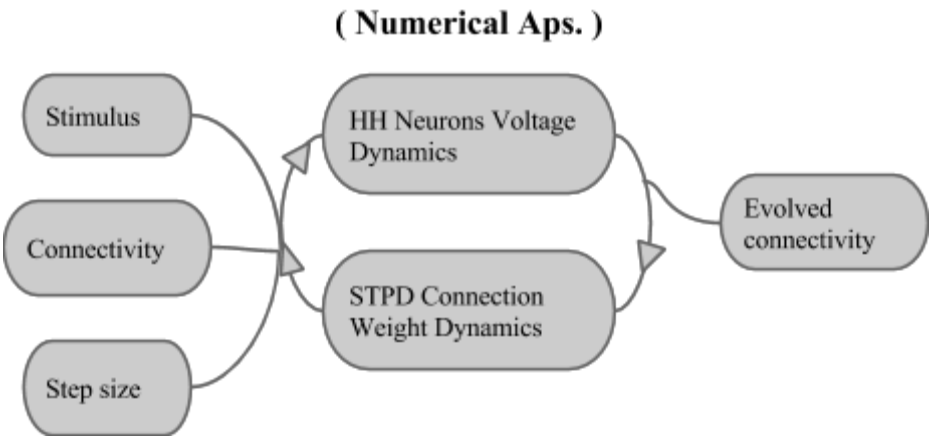

Figure 1: General Idea of the Platform

Initially, we sought to allow the user to choose the number of neurons in the network. However, during GUI construction, we realized that since the number of connections increases as the square of the number of neurons, initial weight matrix input for a network of 4 or more neurons would be very time-consuming. Thus, we chose to fix our network at 3 neurons, with 9 connections (self-connections, or autapses, have been empirically observed), to avoid excessive dimensionality. Additionally, in our final design we decided to let the user change the step size used for numerical approximation of the HH equations. Although using a timestep of ~0.01 was sufficient to accurately solve the HH equations on most network connectivities, it is plausible that some network connectivities that give rise to especially complex behavior may require more

precise approximation to yield accurate results. To inform the user that the chosen timestep was insufficient for accurate approximation, we had the GUI throw an error "ABORT" whenever a neuron's voltage exceeded biological constraints.

Underlying Concept

At time t, the neuron has membrane potential $V(t)$. An input current stimulus $s(t)$ is distributed over the membrane current $I(t) = C\frac{dV_i(t)}{dt}$ and each of several ion channel currents $I_k(t) = g_k(t)(V - V_k)$ having conductance $g_k(t)$ and reversal potential V_k. Then

$$I(t) = s(t) - \sum_k I_k(t)$$

Hodgkin and Huxley (1952) considered an activating ($m(t)$) and inactivating ($h(t)$) sodium current $I_{Na}(t) = \overline{g_{Na}}m(t)^3h(t)(V(t) - V_{Na})$, an activating ($n(t)^4$) potassium current $I_K(t) = \overline{g_K}n(t)^4(V_(t) - V_K)$, and a leak current $I_L(t) = \overline{g_L}(V(t) - V_L)$. The mean conductances (mS/cm^2) $\overline{g_{Na}} = 120, \overline{g_K} = 36, \overline{g_L} = 0.3$ and reversal potentials (mV) $V_{Na} = 115, V_K = -12, V_L = 10.6$ were empirically chosen.

Each (in)activation variable $x(t)$ was modeled as a mapping to $[0, 1]$ according to $\frac{dx(t)}{dt} = \alpha_x(V(t))(1 - x(t)) - \beta_x(V(t))x(t)$, where α_x and β_x were chosen to fit empirical data:

$$\alpha_n(V(t)) = 0.1 * \frac{10-V(t)}{\exp\{\frac{10-V(t)}{10}\}-1} \quad , \quad \alpha_m(V(t)) = 0.1 * \frac{25-V(t)}{\exp\{\frac{25-V(t)}{10}\}-1} \quad , \quad \alpha_h(V(t)) = 0.07 * \exp\{\frac{-V(t)}{20}\}$$

$$\beta_n(V(t)) = .125 * \exp\{\frac{-V(t)}{80}\} \quad , \quad \beta_m(V(t)) = 4 * \exp\{\frac{-V(t)}{18}\} \quad , \quad \beta_h(V(t)) = \frac{1}{\exp\{\frac{30-V(t)}{10}\}+1}$$

In expanded form, then,

$$C\frac{dV(t)}{dt} = s(t) - \left[\overline{g_{Na}}m(t)^3h(t)(V(t) - V_{Na}) + \overline{g_K}n(t)^4(V(t) - V_K) + \overline{g_L}(V(t) - V_L)\right].$$

To find suitable ICs, we rewrite $\tau_x(V(t))\frac{dx(t)}{dt} = x_\infty(V(t)) - x(V(t))$, where $x_\infty(V(t)) = \frac{\alpha_x(V(t))}{\alpha_x(V(t))+\beta_x(V(t))}$ and $\tau_x(V(t)) = \frac{1}{\alpha_x(V(t))+\beta_x(V(t))}$, and we choose $x(V(0)) = x_\infty(V(0))$ with $V(0) = 0$.

Although various neurons can include additional specific ion channels for specialized functions (Gerstner and Kistler, 2002), the Hodgkin-Huxley model implements the universal basic mechanism for an action potential. First, an increase in voltage causes sodium (m) and potassium (n) channel activation, which can (since $\overline{g_{Na}} > \overline{g_K}$) lead to positive excitatory feedback and result in a sharp depolarization. However, the sodium channel soon inactivates (h), allowing the potassium channel to bring the voltage back down and cause a hyperpolarization before returning to resting state.

A limitation of the HH model is that although it implements the universal basic mechanism for an action potential, it does not include specialized ion currents that are found exclusively in some brain regions. For example, it does not include a calcium current, which is essential for neurons to demonstrate bursting behavior, as shown below (Gerstner and Kistler, 2002):

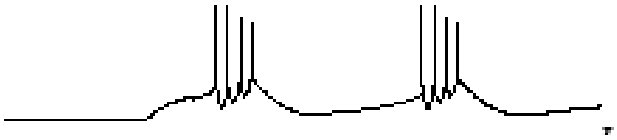

Figure 2: Bursting Behavior of Neurons

To couple the HH neurons, we incorporated currents from the other neurons, linearly weighted with the respective connection strength, into the stimulus term s(t). To implement STDP, we used a piecewise weight update function consisting of a linear realm. This function took the form of the graph below ("STDP"), with change in weight on the vertical axis and difference in spike times on the horizontal axis.

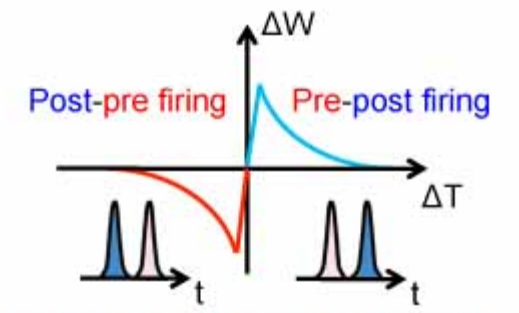

Figure 3: Graph of Piecewise Weight Update Function

GUI Tool

A screenshot of the GUI, as it first appears upon running the program, is shown below:

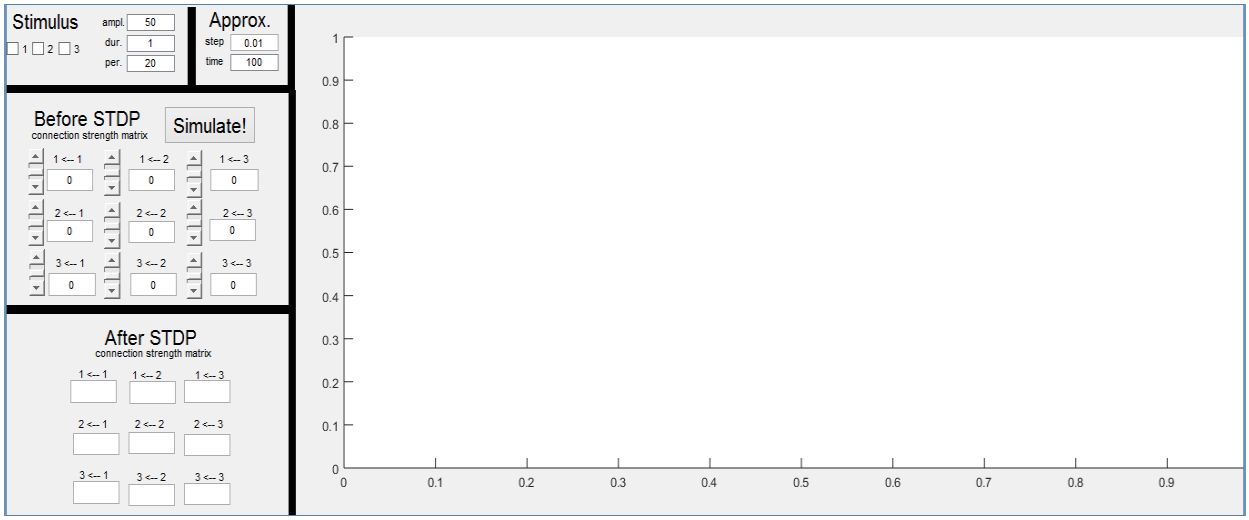

Figure 4: Screenshot of the GUI at Start of the Program

The GUI was split into five sections.

1. The stimulus section (top, left) included stimulus parameters. The user could set the amplitude, duration, and period (all in ms) of the analog stimulus by typing any number (must be nonnegative for duration and period) into text boxes. The user would also check the box(es) corresponding to the neuron(s) (labeled 1, 2, or 3) the user wished to stimulate.
2. The approximation section (right of stimulus section) allowed the user to choose the numerical approximation step size and the total time they wished to simulate the network (both in ms) by typing nonnegative numbers into the corresponding text boxes.
3. The "before STDP" section (below the stimulus and approximation sections) allowed the user to input dimensionless connection strengths into the weight matrix by entering any number between -1 and 1 into the corresponding text box or adjusting the corresponding

slider. Weights between 0 and +/- 1 corresponded to excitatory (+) and inhibitory (-) connections, with the magnitude encoding the strength of the connection. A weight of 0 meant there was no connection. As labeled, the input to the ij entry of the weight matrix corresponded to the connection that neuron i received from neuron j. A push button to start the simulation was included in this section so that the user could begin the simulation after defining the initial connectivity. Once the button was pushed, a simulation progress weight bar appeared to let the user know the progress of the simulation.

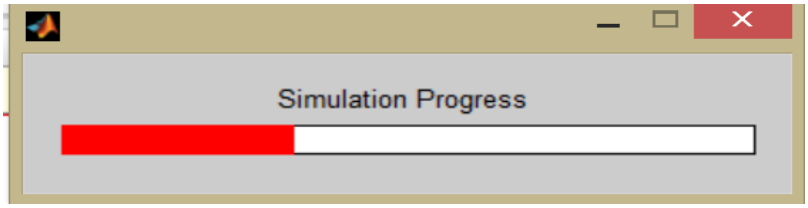

Figure 5: Stimulation Progress Weight Bar

If a user saw that the bar was moving extremely slowly, (s)he could adjust the approximation step size to speed up the simulation.

4. The plot section (right) would graphically display the simulation results, which consisted of the stimulus (black) and the neuron voltages (solid if stimulated, broken if not). Above the stimulus and voltage plots, we included a dimensionless spike train, a binary abstraction of the signal that encoded only spike times.

5. Lastly, after simulation, the STDP-evolved connections were shown in the "after STDP" section.

   A screenshot after a complete simulation is shown below:

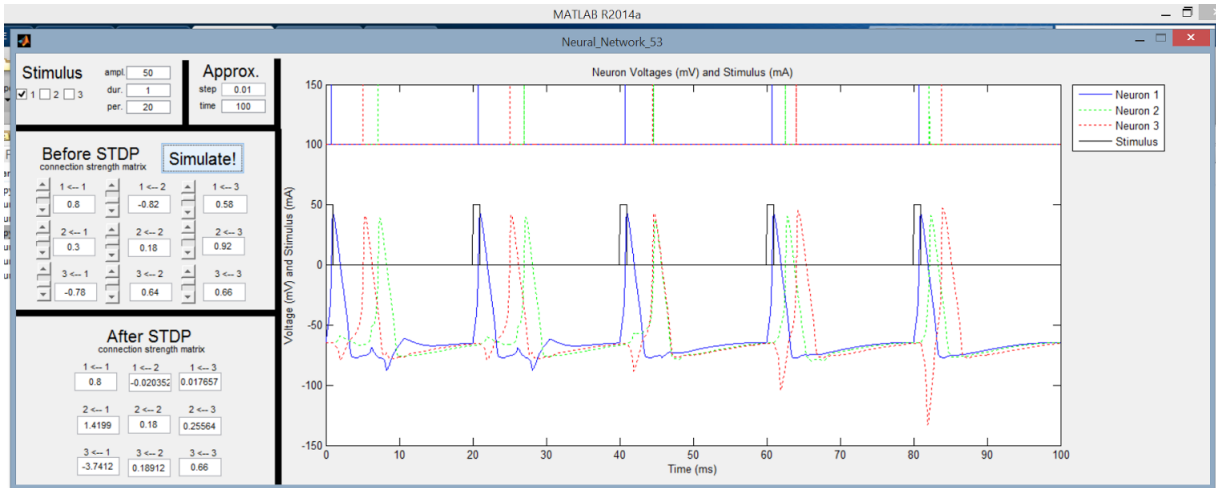

Figure 6: Screenshot of GUI after a Complete Simulation

To run another simulation, a user could simply adjust whatever parameters (s)he wanted, and then press simulate again. So that a user could keep track of which weights were adjusted, whenever a user adjusted a "before STDP" weight, the corresponding text box in the "after STDP" section would become blank.

Discussion of performance

Our GUI was able to take input values in our text boxes and assess checkbox values in a way that allowed for the accurate computation of network simulation and connectivity evolution. To demonstrate model verification for a nonconnected, we compare an action potential from a neuron (left, blue) in our program to a HH action potential given by Gerstner and Kistler (2002) (right):

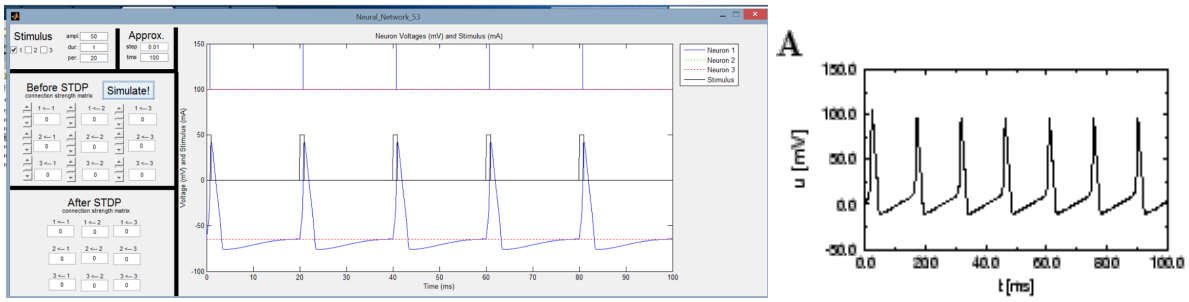

Figure 7: Graphs of an Action Potential from the GUI (left) and from Gerstner and Kistler (right)

Note that the vertical shift is due to our shifting the neuron's rest voltage to its experimentally-supported value of about -65 mV, and the horizontal stretch is due to graph size and the slightly lower-frequency stimulus that we used. Indeed, we see that our model reproduces the stereotypical HH action potential.

To verify our model for connected neurons, we first considered a 1 → 2 scheme (neuron 1 excites neuron 2). By stimulating neuron 1, we would cause spikes in neuron 1, and a strong connection would cause neuron 2 to spike directly after neuron 1. According to STDP, then, since neuron 2 is spiking directly after neuron 1, we would expect to see the 1 → 2 connection increase. As shown in the simulation screenshot below, this is indeed what we observed.

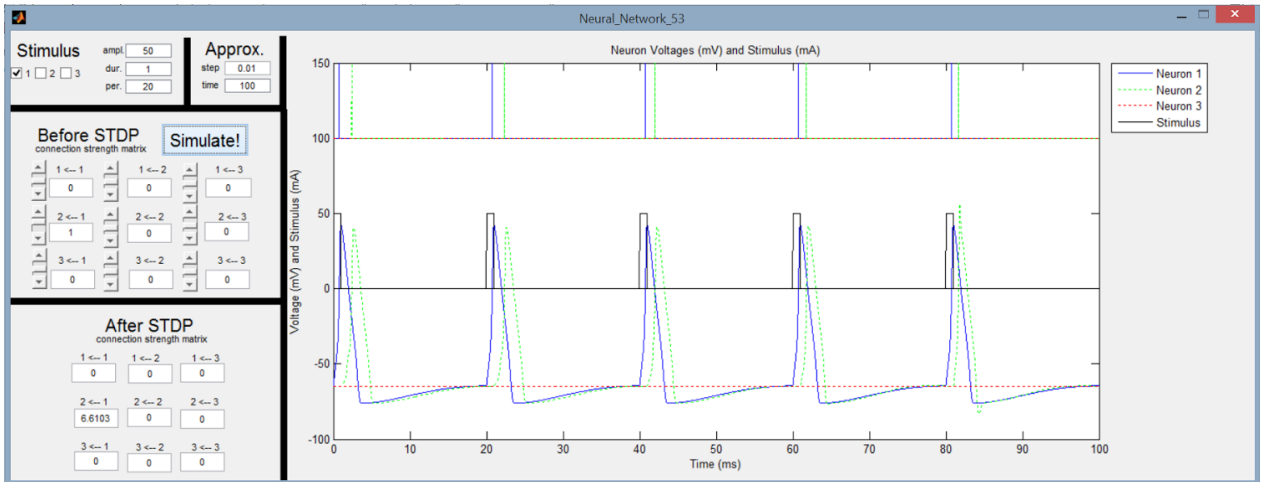

Figure 8: A Simulation Screenshot from the GUI

Lastly, we observed the evolution of a 1 → 2 → 3 synfire chain (neurons which consecutively excite each other). We expected to see both the 1 → 2 and 2 → 3 connections increase, and indeed, they did:

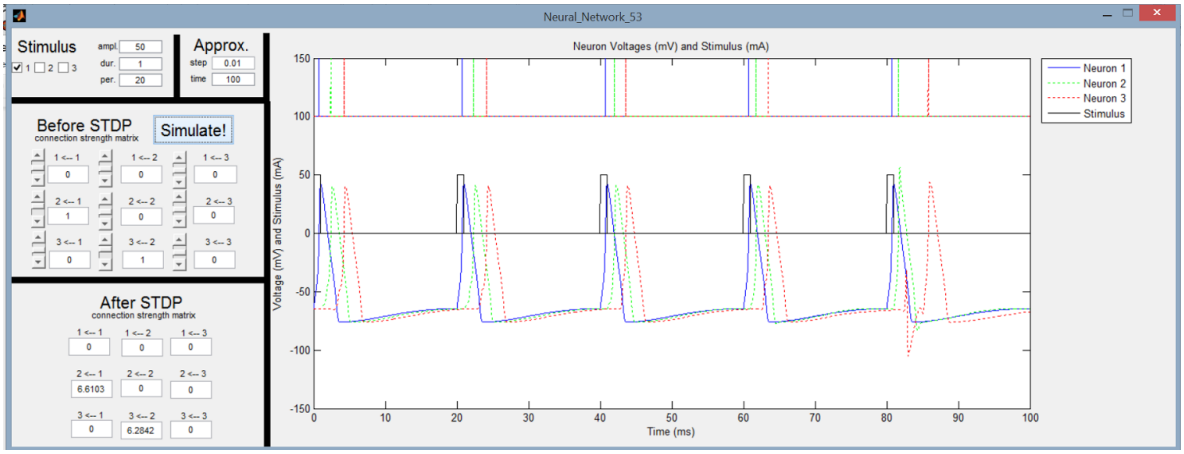

Figure 9: Screenshot of a 1 → 2 → 3 Synfire chain from the GUI

We emphasize that although the HH and STDP learning rules are not entirely complete (neurons and their plasticity rules are current research topics in the scientific community), our GUI's accuracy does not suffer from inaccuracies, such as missing parameters, because our goal was to provide a framework in which the HH and STDP models could be explored.

Discrepancies between experimentally observed neurons and our simulation results would stem from the HH and STDP models themselves.

However, we will elaborate on some weaknesses of the HH and STDP models. The HH model for neuron dynamics assumes that the internal state of a biological neuron is accurately described by its voltage and its voltage-dependent sodium and potassium channels. Indeed, this is sufficient to generate stereotypical action potentials, but it does not include properties of specialized neurons (such as bursting neurons, described earlier). Similarly, the STDP learning rule is ambiguous with regards to fast-spiking configurations such as bursting neurons (i.e. should each fast-spike be counted separately, or should all the fast-spikes be counted collectively as a single event?).

This GUI could be improved by including a variable number of neurons (in a feasible manner) and user-drawn stimulus. We wanted to make our GUI take on as many neurons as the user desired; however, due to computation time, it seemed best to limit the user to 3 neurons. However, although some researchers study small networks of 1, 2, or 3, neurons, other researchers study large networks of thousands or millions of neurons. Similarly, we wanted to include a feature of the GUI on which a user could draw the desired stimulus with mouse clicks, but this proved too complicated. Because the majority of experimental neural network studies involve excitation with analog stimuli, we believe our stimulus setup is sufficient for most researchers' needs; however, to cater to researchers who are interested in the effects of complex stimuli on neurons, greater stimulus capability would need to be included.

References

Gerstner, Wulfram, and Werner M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Hodgkin, Alan L., and Andrew F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve."*The Journal of physiology* 117.4 (1952): 500-544.

Song, Sen, Kenneth D. Miller, and Larry F. Abbott. "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity." *Nature neuroscience* 3.9 (2000): 919-926.

"STDP." *Lightwave and Microwave Photonics Laboratory.* Web. 27 Apr. 2015