Assignment 3

Neuromorphic Engineering
Bipin Rajendran

$\boxed{\text{Due Oct 3, 2014 at 5:00PM}}$

Your submission must include:

1. A .pdf or .doc file clearly documenting your code, figures, and results.

2. Your MATLAB code. The code must be saved in plain text files that can be immediately run in MATLAB; include a file called main.m that runs the functions you implemented and generates the figures described in your assignment write-up. You will lose credit if your code is absent or cannot be run.

Store your write-up and code in a single directory named hw3_yourID (for example hw3_12D423222) and submit it in moodle.

Do not wait till the last minute to start the Assignment as you will require at least at least 8 hours to complete all the questions.

Late submission policy:

Before the solution key is uploaded in moodle: If your original score is S and you submitted the HW X hours after the deadline, your score will be $S\exp(-X/24)$.
After the solution key is uploaded in moodle: 0 credit.

In this assignment, we will understand how populations of integrate and fire neurons with plastic synapses and axonal delays can generate interesting neuronal dynamics.

Problem 1: Representing synaptic connectivity and axonal delays
Spike transmission in biology is associated with finite transmission delays. One way to model this is to assume that if a pre-synaptic neuron issues a spike at time t^k, the current flowing into the post-synaptic neuron is given by the expression

$$I_{syn}(t) = I_0 w_e \left[e^{-(t-t^k-\tau_d)/\tau} - e^{-(t-t^m-\tau_d)/\tau_s} \right] h(t - t^m - \tau_d)$$

where $h(x)$ is the Heaviside function and τ_d is the axonal delay for that synapse. Hence, the connectivity information for a neuron with M synapses can be represented by three vectors, given as

$$Fanout = [n_1, n_2, \dots n_M] \qquad W = [w_1, w_2, \dots w_M] \qquad \tau_d = [\tau_{d1}, \tau_{d2}, \dots \tau_{dM}]$$

where $Fanout$ stores the list of ids of the post-synaptic neurons, w_i is the strength of the corresponding synapse, and τ_{di} is the axonal delay associated with that connection. Note that we only need to store the fan-out information, the fan-in information can be determined if necessary from this list.

(a) Since all neurons in a network may not have the same number of connections, using arrays to represent this connectivity information can be very inefficient. One efficient way to represent connectivity information is to use cell arrays in MATLAB.

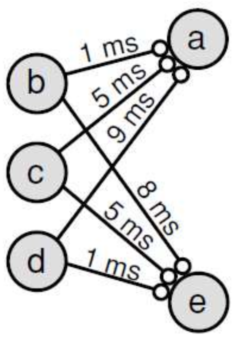

Figure 1: Small network with axonal delays

Create three cell arrays named $Fanout$, $Weight$ and $Delay$ to store the connectivity information and axonal delays of all the neurons for the network shown in Figure **??**. All synapses have equal strength, $w = 3000$. **2 points**

(b) Assume that the neurons can be modeled by the simple Leaky Integrate and Fire model discussed in Homework 1, with a minor modification to incorporate an artificial refractory period, R_p.

$$C\frac{dV(t)}{dt} = -g_L(V(t) - E_L) + I_{app}(t) + I_{syn}(t)$$

This is done by assuming that if a neuron spikes ($V \geq V_T$) at time t^k, then, the membrane potential is artificially held at E_L for all times $t^k \leq t < t^k + R_p$. As before, assume that C= 300 pF, $g_L = 30$ nS, $V_T = 20$ mV, $E_L = -70$ mV and $R_p = 2$ ms. Note that the synaptic current has to be calculated at every instant of time during the simulation, based on the spike time of the presynaptic current and the axonal delay.

Write a MATLAB code to simulate the behavior of the network in Figure **??** for the two situations described below:

Case 1: Neurons b, c, d received a square pulse input of duration 1 ms, starting at t=0, 4 and 8 ms respectively.

Case 2: Neurons d, c, b received a square pulse input of duration 1 ms, starting at t=0, 3 and 7 ms respectively.

Assume that the magnitude of the current injected during the 1 ms pulse is 50 nA. For the synaptic current equation, assume $I_0 = 1$ pA, and $\tau = 15$ ms represents the time constant of the membrane and $\tau_s = \tau/4$ is the synaptic time constant.

Plot the synaptic currents and the response of the neurons. **8 points**

Hint: It will also be useful to create cell arrays to store the time when a neuron spiked, time when that spike arrives at a post-synaptic neuron, the weight through which that signal was transmitted, and also the *id* of the pre-synaptic neuron. The following MATLAB code can be used to initialize such an array for the entire network of N neurons.

```
for i=1:N
        spike_time{i}=[ ];
        arrival_time{i}=[ ];
        strength{i}=[ ];
        pre_neuron{i}=[ ];
end
```

As an example, if neuron 3 spiked at 15 ms and and 23 ms, while neuron 4 spiked at 20 ms, these cell arrays should look like

```
        spike_time = {[ ] [ ] [15 23] [20] [ ]}
        arrival_time ={[20 29 28 ] [ ] [ ] [ ] [20 21 28] }
        strength ={[3000 3000 3000 ] [ ] [ ] [ ] [3000 3000 3000] }
        pre_neuron{i}={[3 4 3 ] [ ] [ ] [ ] [3 4 3]}
```

Problem 2: Dynamical Random network

We would like to use the framework you developed above to understand the dynamics of a random network of neuron interacting with each other through non-plastic synapses.

Assume we have a network made of $N = 500$ LEIF neurons from problem 1 with $R_p = 2\,\mathrm{ms}$. Assume that the first 80% of the neurons in your network are excitatory and the last 20% are inhibitory. The excitatory neurons can connect to any other neuron in the pool, while the inhibitory neurons synapse only to other excitatory neurons. Each neuron communicates to $N/10 = 50$ unique neurons in the network (a self-connection is also permitted, but not necessary). All excitatory synapses have a strength equal to $w_e = 3000$ and all inhibitory synapses have a strength equal to $w_i = -w_e$. The axonal delays for synapses from excitatory neurons are uniformly distributed integers in the range of $1 - 20\,\mathrm{ms}$, while the delay for all the inhibitory synapses is $1\,\mathrm{ms}$.

Assume that the first 25 excitatory neurons are receiving a Poisson stimulus, whose arrival rate is given by $\lambda = 100/\,\mathrm{s}$. A stimulus at time t^s results in a current flowing into the recipient neuron with no delay, according to the relation

$$I_{ext}(t) = I_0 w_s \left[e^{-(t-t^s)/\tau} - e^{-(t-t^s)/\tau_s}\right] h(t - t^s)$$

Assume $I_0 = 1\,\mathrm{pA}$, and $w_s = 3000$.

(a) Create a raster plot for the spikes of the $N = 500$ neurons for a total simulation time of $T = 1000\,\mathrm{ms}$. **6 points**

(b) Let $R_e(t)$ and $R_i(t)$ denote the total number of spikes issued by all the excitatory and inhibitory neurons in the interval $[t, t + 10ms]$. Plot how $R_e(t)$ and $R_i(t)$ vary during your simulation. **2 points**

(c) Qualitatively explain the mechanisms underlying the dynamical behavior. **2 points**

Problem 3: Dynamics of smaller networks

(a) Repeat the exercise in Problem 2, but now with $N = 200$ neurons. Assume similar statistical connectivity characteristics as in Problem 2. (i.e., the excitatory-inhibitory populations are in the $80 - 20$ ratio, and the fanout of each neuron is $N/10 = 20$). Assume $w_e = -w_i = 3000$, and the first 25 neurons are receiving Poisson stimulus. **1 point**

(b) Study the behavior of the network for other values of synaptic strengths, while maintaining $w_e = -w_i$. Are you able to obtain the behavior you saw earlier for some aptly chosen value for the synaptic magnitude. Explain on the basis of $R_e(t)$ and $R_i(t)$. **3 points**

(c) For smaller networks, $|w_e| = |w_i|$ can not give the network behavior observed earlier. Choose the correct alternatives in this sentence:
The net *excitation/inhibition* in the network should be *increased/decreased* in order to observe the network behavior observed in problem 2. **1 point**

(d) Propose a modification of the synaptic configuration, $w_e = -\gamma w_i$, where γ is a positive real number to achieve similar dynamical behavior (at least in the overall spike counts) in the smaller network and simulate the new network's dynamics for the same random external stimulii. γ could be lesser or greater than 1. **5 points**

Problem 4: Adjusting the weights dynamically

Now we will try to incorporate timing dependent plasticity in our synapses. This is done by the following algorithm.

When a neuron N_i spikes, say at time t_i^k

1. For each of its upstream neuron, N_j

   - Determine the time t_j^{last} at which its most recent spike arrived at N_i. Note that we are not looking for the most recent time at which the upstream neuron N_j spiked, but the most recent time at which a prior spike from N_j reached N_i. For example, if $t_i^k = 20\,\text{ms}$, and an upstream neuron had spiked at times $t_j = [2, 8, 13, 15]\,\text{ms}$ and if the axonal delay for that channel is $6\,\text{ms}$, $t_j^{last} = 13 + 6 = 19\,\text{ms}$.
   - Determine the new strength of the synapse according to the rule,

$$w_{new,ij} - w_{old,ij} = w_{old,ij} A_{up} \exp\left[\frac{-\left(t_i^k - t_j^{last}\right)}{\tau_L}\right]$$

2. For each of its downstream neuron, N_j

   - Determine the most recent time t_j^{last} at which each of them spiked.
   - Determine the time at which the spike from the current spiking neuron will reach them, $t_i^k + \tau_{d,ij}$
   - Determine the new strength of the synapse according to the rule,

$$w_{new,ij} - w_{old,ij} = w_{old,ij} A_{down} \exp\left[\frac{-\left(t_i^k + \tau_{d,ij} - t_j^{last}\right)}{\tau_L}\right]$$

Positive A_{up} and negative A_{down} implements STDP, while negative A_{up} and positive A_{down} implements anti-STDP.

(a) Write a program to implement timing dependent plasticity in a network of spiking neurons. Simulate the behavior of the small network in problem 3(a) if all the excitatory synapses are plastic and obey STDP, and the inhibitory synapses are non-plastic. Assume $A_{up} = 0.01$, $A_{down} = -0.02$ and $\tau_L = 20\,\text{ms}$. **9 points**

(b) Plot the variation in the average excitatory synaptic strength in the network as a function of time. **1 points**