Neural Networks for AI

Lab 1

Fernanda Marana (s3902064)
Floris Cornel (s2724685)

April 24, 2019

1 Description

The aim of this project is to implement an artificial integrate-and-fire neuron based on the book "An Introduction to MATLAB for Neuroscience Resarch" (Sterratt, Auzinger, Fabianek, Holy, Pawlik, 2006). We analyse the influences of three parameters over the unit by relying our arguments on plots as a representative justification and by conducting a parameter sweep over them. There parameters analysed are the input resistance in MOhm (Rin), membrane time constant in milliseconds (tau) and threshold in mV (theta). On Figure 1, it is shown the default values determined by the group in order to lead the reader to make a comparison between them and the results of the later modifications.

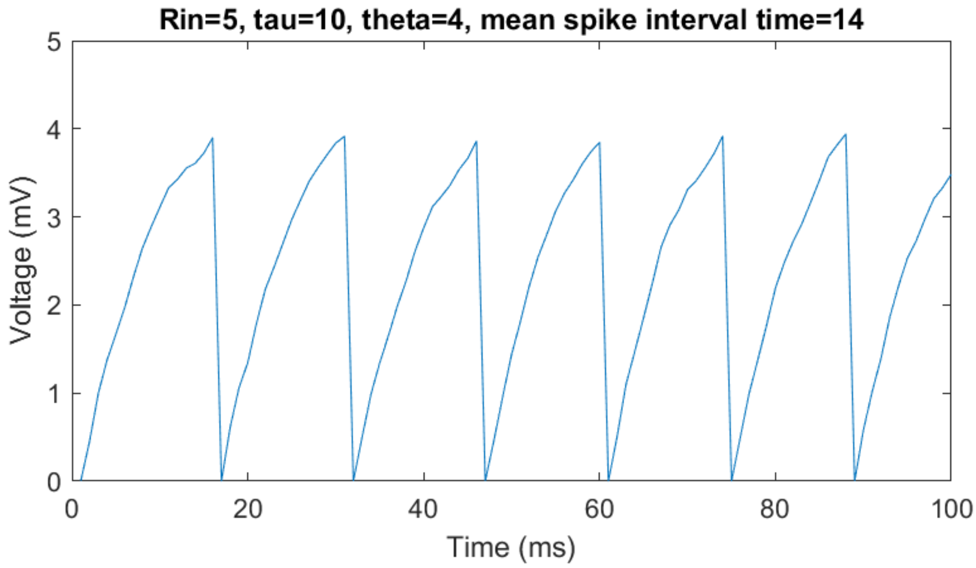

Figure 1: Plot of voltage (mV) over time (ms) of a firing neuron with parameter values: The input resistance is 5 MOhm, membrane time constant is 10 ms, threshold is 4 mV. The mean spike interval time is 14 ms.

On the last pages of this document, it is available the codes used on the project. It includes the noisyifneuron.m[1] and isi.m[2] of the book "An Introduction to MATLAB for Neuroscience Resarch" (Sterratt, Auzinger, Fabianek, Holy, Pawlik, 2006) and two others implemented for this exercise. The defaultPlot.m[3] gets the tspike of the noisyifneuron function and plots it while the parameterSweepPlot.m[4] does all the three parameters sweep and plots them. On the next section, the analysis of each parameter's influence is discussed by evaluating the spikes. On every graph is shown three different behaviors related to a low, medium or high setting.

2 Analysis

On Figure 2, the input resistance in MOhm (Rin) parameter sweep is presented. As we can see, when influenced by a low setting the mean value turns infinite and, as a result, the neuron doesn't fire. On contrast, a medium or high setting fires the neuron, but with different frequencies and different voltages. The higher the Rin the less is its voltage and its mean spike interval time.

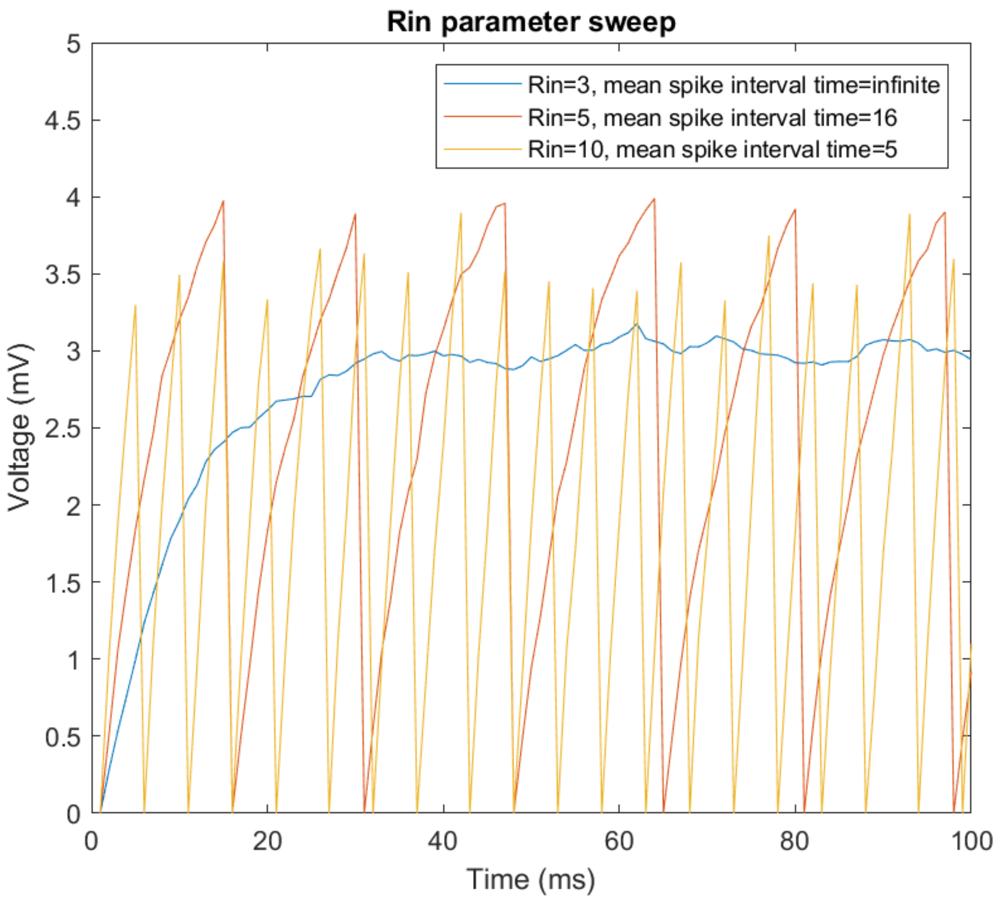

Figure 2: Plot of voltage (mV) over time (ms) of firing neurons with different values for the input resistance. The membrane time constant is 10 ms and the threshold is 4 mV. The mean spike interval time (in ms) varies based on the input resistance (in MOhm). Both be found in the legend.

On Figure 3, it is notable that the membrane time constant parameter does not influence if the firing of a neuron happens since no matter what setting it is on, the mean spike interval time does not turn infinite. However, this parameter still influences the spike interval time of the firing and a little on how much voltage is released.

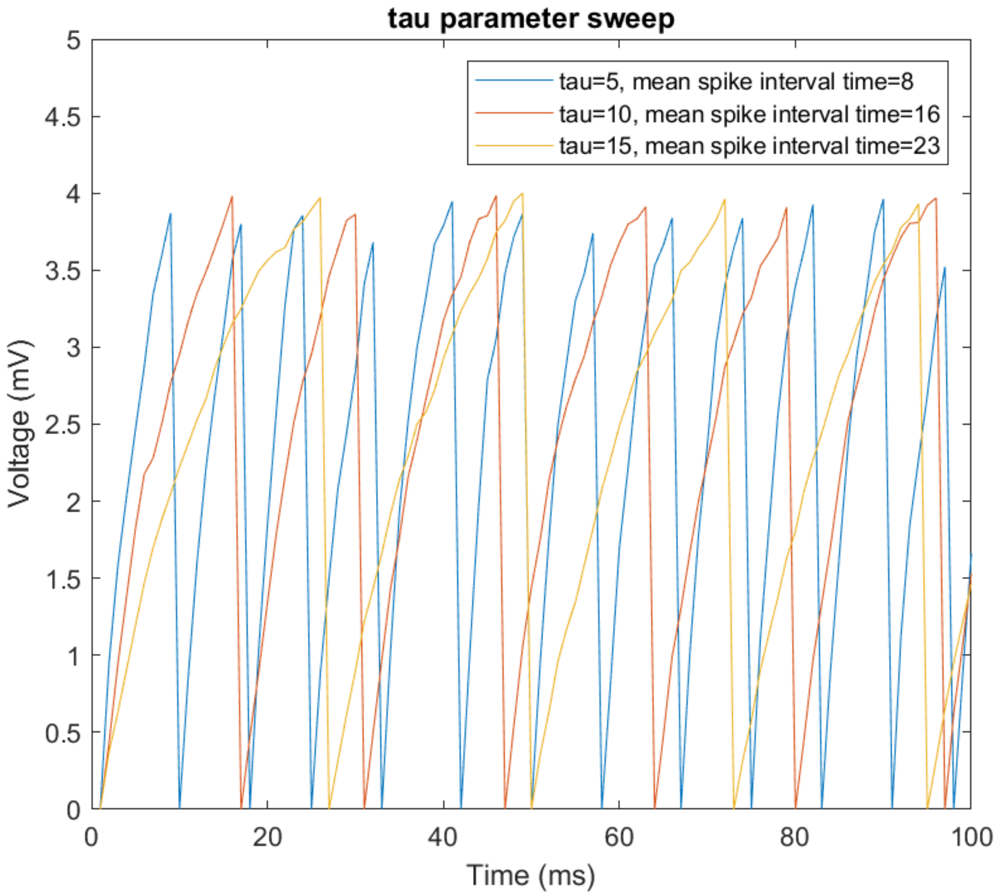

Figure 3: Plot of voltage (mV) over time (ms) of firing neurons with different values for the membrane time constant. The input resistance is 5 MOhm and the threshold is 4 mV. The mean spike interval time (in ms) varies based on the membrane time constant (in ms). Both be found in the legend.

Finally, Figure 4 presents the threshold (theta) in the three different settings. Unlike the input resistence (Rin), as this parameter's value increased, its mean spike interval time goes to infinite and the neuron does not fire. There is also a great difference between the voltage and the spike interval between the low and medium setting leading us to believe this parameter has the greatest influence over the integrate-and-fire neuron of the three.

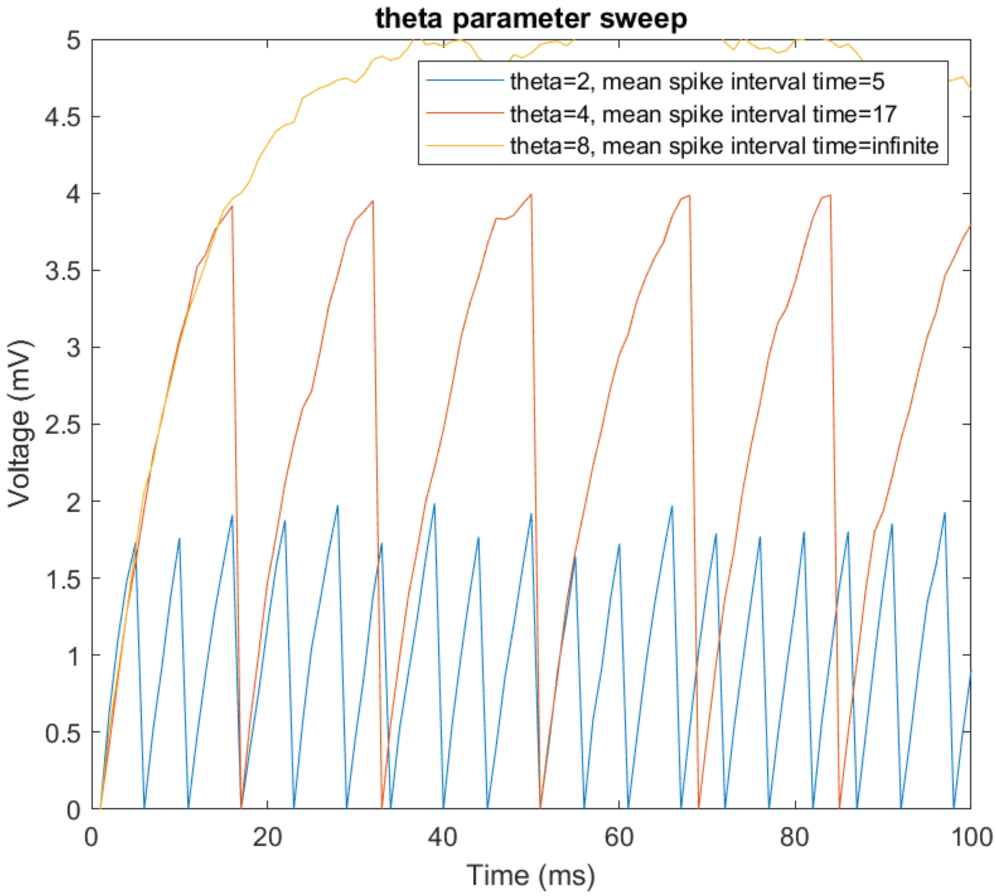

Figure 4: Plot of voltage (mV) over time (ms) of firing neurons with different values for the threshold. The input resistance is 5 MOhm and the membrane time constant is 10 ms. The mean spike interval time (in ms) varies based on the threshold (in mV). Both be found in the legend.

3 Code

```
function tspike = noisyifneuron(tau,theta,Rin)
  nstep = 100; % Number of timesteps to integrate over
  Inoise = 0.1;
  I0 = 1+Inoise*randn(1,nstep); % Input current in nA
  dt = 1; % time step in ms
  %tau = 10; % membrane time constant in ms
  %theta = 4; % threshold in mV
  v = zeros(1,nstep);
  %Rin = 5; % Input resistance in MOhm
  tspike = [];
  t = (1:nstep)*dt;
  for n=2:nstep
    v(n) = v(n-1) + dt*(- v(n-1)/tau + Rin*I0(n)/tau);
    if (v(n) > theta)
    v(n) = 0;
      tspike = [ tspike t(n) ];
```

```
    end
  end
  plot(t,v);
end
```

Listing 1: noisyifneuron.m

```
function isi_result=isi(spiketimes)
  % ISI produces interspike intervals from spike times
  % ISI(spiketimes) returns the interspike intervals
  % of SPIKETIMES
  if(length(spiketimes)>1)
    isi_result = diff(spiketimes);
  else
    isi_result = [];
  end
end
```

Listing 2: isi.m

```
Rin = 5; % Input resistance in MOhm
tau = 10; % membrane time constant in ms
theta = 4; % threshold in mV

tspike = noisyifneuron(tau,theta,Rin);

asp = round(mean(isi(tspike))); % mean spike interval time

% Create figure
fig = gcf;
fig.PaperUnits = "inches";
fig.PaperPosition = [0 0 6 3];

% Make it look decent
title("Rin="+num2str(Rin)+", tau="+tau+", theta="+theta+", mean spike interval time="+
    asp);
ylim([0 5]);
ylabel("Voltage (mV)"); % Set the y axis label
xlabel("Time (ms)"); % Set the x axis label

% Save the plot to image
saveas(gcf, "defaultValues.png");
```

Listing 3: defaultPlot.m

```
Rin = 5; % Input resistance in MOhm
tau = 10; % membrane time constant in ms
theta = 4; % threshold in mV

Rins = [3 5 10];
taus = [5 10 15];
thetas = [2 4 8];

% Create figure
fig = gcf;
```

```
fig.PaperUnits = "inches";
fig.PaperPosition = [0 0 6 5];

legends = [];
for n = 1 : 3
    tspike = noisyifneuron(tau,theta,Rins(n));
    hold on;
    asp = round(mean(isi(tspike))); % mean spike interval time
    if isnan(asp)
        asp = "infinite";
    end
    legendentry =  "Rin="+Rins(n)+", mean spike interval time="+asp;
    legends = [legends, legendentry];
end

% Make it look decent
title("Rin parameter sweep");
legend(legends);
ylim([0 5]);
ylabel("Voltage (mV)"); % Set the y axis label
xlabel("Time (ms)"); % Set the x axis label

% Save the plot to image
saveas(gcf, "rinValues.png");
clf(fig, 'reset') % reset plot
hold off;

% Create figure
fig = gcf;
fig.PaperUnits = "inches";
fig.PaperPosition = [0 0 6 5];

legends = [];
for n = 1 : 3
    tspike = noisyifneuron(taus(n),theta,Rin);
    hold on;
    asp = round(mean(isi(tspike))); % mean spike interval time
    if isnan(asp)
        asp = "infinite";
    end
    legendentry =  "tau="+taus(n)+", mean spike interval time="+asp;
    legends = [legends, legendentry];
end

% Make it look decent
title("tau parameter sweep");
legend(legends);
ylim([0 5]);
ylabel("Voltage (mV)"); % Set the y axis label
xlabel("Time (ms)"); % Set the x axis label

% Save the plot to image
saveas(gcf, "tauValues.png");
```

```
clf(fig, 'reset') %reset plot

% Create figure
fig = gcf;
fig.PaperUnits = "inches";
fig.PaperPosition = [0 0 6 5];

legends = [];
for n = 1 : 3
    tspike = noisyifneuron(tau,thetas(n),Rin);
    hold on;
    asp = round(mean(isi(tspike))); % mean spike interval time
    if isnan(asp)
        asp = "infinite";
    end
    legendentry =  "theta="+thetas(n)+", mean spike interval time="+asp;
    legends = [legends, legendentry];
end

% Make it look decent
title("theta parameter sweep");
legend(legends);
ylim([0 5]);
ylabel("Voltage (mV)"); % Set the y axis label
xlabel("Time (ms)"); % Set the x axis label

% Save the plot to image
saveas(gcf, "thetaValues.png");
clf(fig, 'reset') %reset plot
```
Listing 4: parameterSweepPlot.m