Lab Assignement 0

Karamoulas Eleftherios - S3261859
Tzafos Panagiotis - S3302148

February 15, 2017

1 Answers for Lab 0

1. After our experiments keeping 2 constants and sweeping 1 of the factors we came to the conclusion that when we sweep Rin low values give as spare spikes and when the value gets increased spikes occur more often until Rin reaches a high value that zeros the v(n) and in this case we have very few spikes. When we sweep tau because he is the divisor in our equation low values of tau give frequent spikes and the increase of tau has as a result to reach a point where we have none spikes. Last the sweep of theta as its the factor that limits our v(n) when its gonna become zero again when we have small theta the spikes are often and while theta rices the frequency of spikes drops down.

2. Something more i guess.

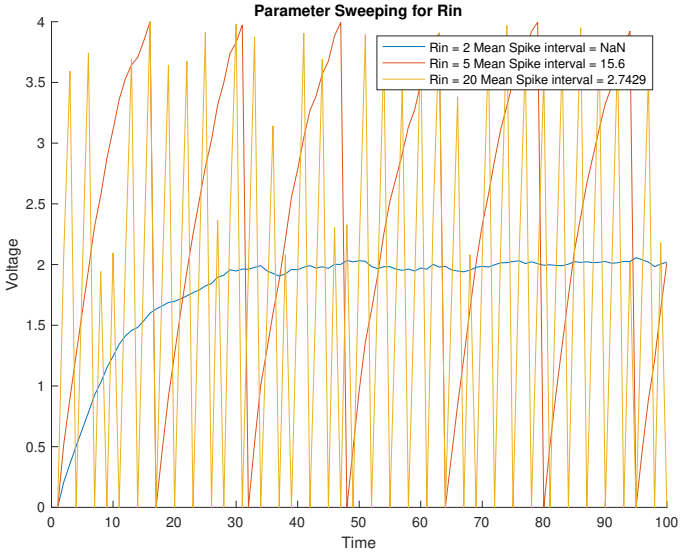

Figure 1: R-sweep

Listing 1: integrator.m

3. ```
   nstep =100; %Number of timesteps to integrate over
   Inoise = 0.1;
   IO = 1+Inoise*randn(1,nstep); %input current in nA
   dt = 1; % time step in ms
   tau_vector = [3 10 30] ; %membrane time constant in ns
   ```
   ```

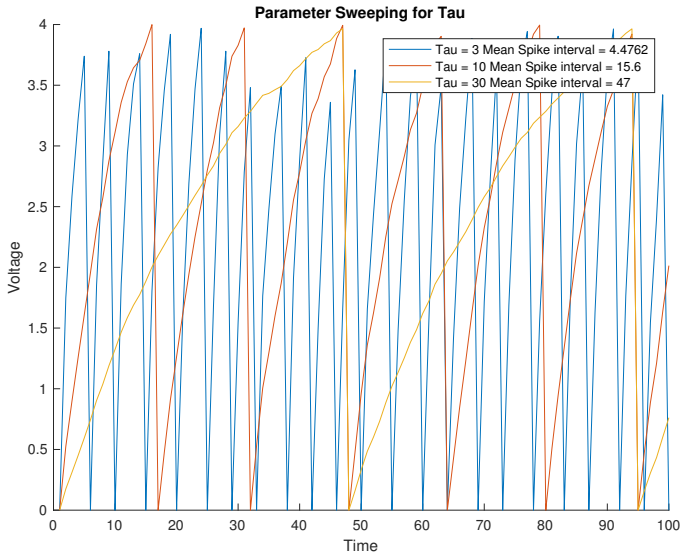

Figure 2: Tau-sweep

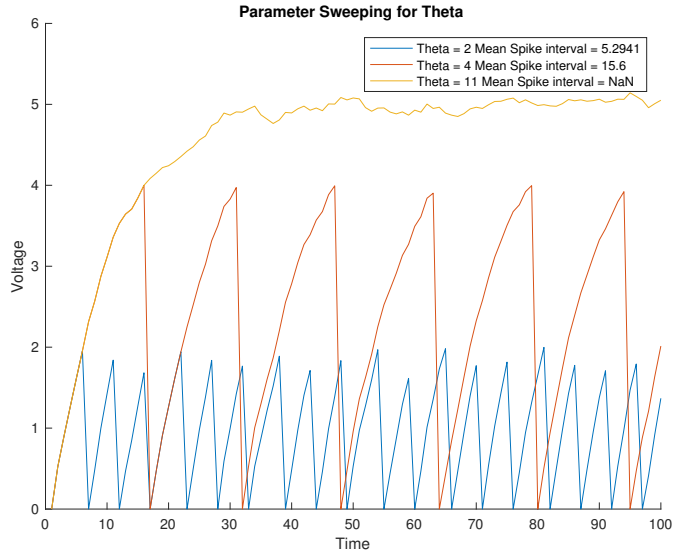

Figure 3: Theta-sweep

```
theta_vector = [2 4 11]; % threshold in mV
Rin_vector = [2 5 20]; %Input resistance in M??hm
v = zeros(1,nstep);
tspike = [];
t = (1:nstep)*dt;

figHandleTau = figure; %figure for Tau
figHandleTheta = figure; %figure for Theta
figHandleRin = figure; %figure for Rin
```

```
Rin = Rin_vector(2); %assigning starting values (middle) for the constants
theta = theta_vector(2);

legendInfo = cell(1,3); %initialization of legend cell for the figures
i=1; %initialization of i that counts the legends that we are gonna have in our fi
for tau=tau_vector %sweep of tau value while keeping the other 2 variables constar
        for n=2:nstep
                v(n)=v(n-1) + dt*(-v(n-1)/tau + Rin*IO(n)/tau);
                if (v(n) > theta)
                        v(n) = 0;
                        tspike = [tspike t(n)];
                end
        end
        figure(figHandleTau) %use of figure command so that our plots for the diff
        legendInfo{i} = ['Tau = ' num2str(tau) ' Mean Spike interval = ' num2str(r
        tspike = []; %initialization of tspike table for the next sweep
        hold all %hold all command so that all plots of this iteration are represe
        title('Parameter Sweeping for Tau');
        xlabel('Time');
        ylabel('Voltage');
        plot(t,v)
        i=i+1; %increment of legend count
end
% downwards we have 2 same iterations for Rin and theta

theta = theta_vector(2);
tau = tau_vector(2);
i=1;
legend(legendInfo)
for Rin=Rin_vector
         for n=2:nstep
                 v(n)=v(n-1) + dt*(-v(n-1)/tau + Rin*IO(n)/tau);
                 if (v(n) > theta)
                         v(n) = 0;
                         tspike = [tspike t(n)];
                 end
         end
         figure(figHandleRin)
         legendInfo{i} = ['Rin = ' num2str(Rin) ' Mean Spike interval = ' num2str
         tspike = [];
         hold all
         title('Parameter Sweeping for Rin');
         xlabel('Time');
         ylabel('Voltage');
         plot(t,v)
         i=i+1;
end

legend(legendInfo)

Rin = Rin_vector(2);
tau = tau_vector(2);
```

```
i=1;
legend(legendInfo)
for theta=theta_vector
        for n=2:nstep
                v(n)=v(n-1) + dt*(-v(n-1)/tau + Rin*IO(n)/tau);
                if (v(n) > theta)
                        v(n) = 0;
                        tspike = [tspike t(n)];
                end
        end
        figure(figHandleTheta)
        legendInfo{i} = ['Theta = ' num2str(theta) ' Mean Spike interval = ' num
        tspike = [];
        hold all
        title('Parameter Sweeping for Theta');
        xlabel('Time');
        ylabel('Voltage');
        plot(t,v)
        i=i+1;
 end
 legend(legendInfo)
```

Listing 2: isi.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
```