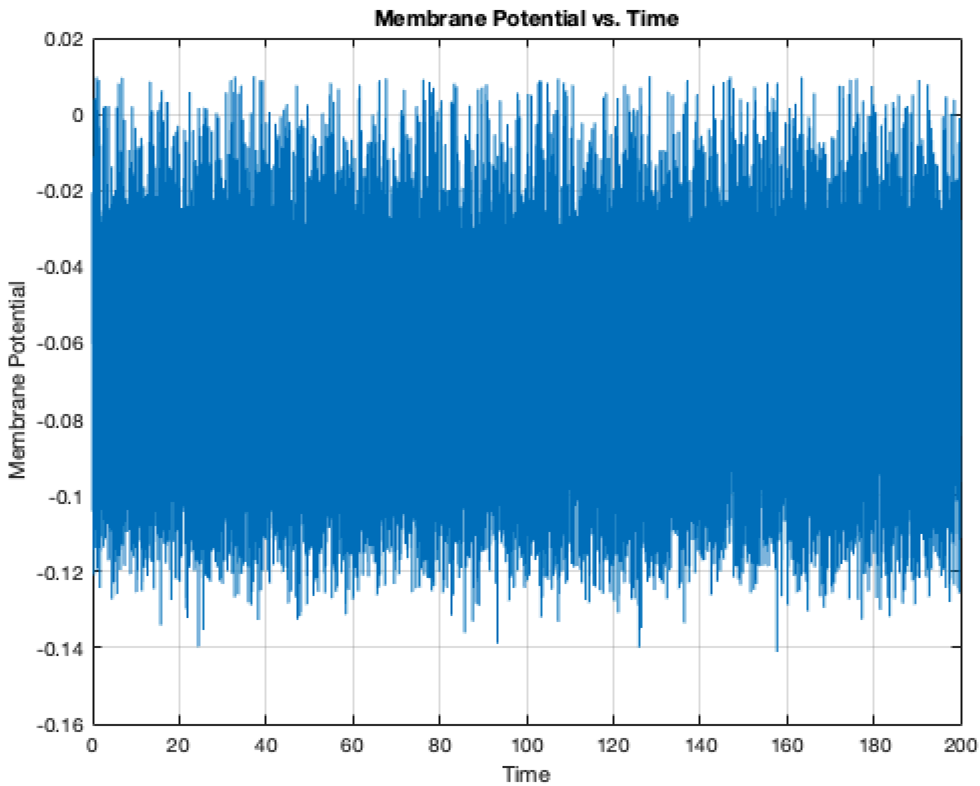
```matlab
% Tutorial 3.1
% Created by Jonathan Lindbloom, 3/2/2019
% Email: jlindbloom@smu.edu

clear all
close all

% Define parameters.
global e_leak r_membrane c_membrane v_threshold v_reset e_k
 delta_g_sra tau_sra g_leak a delta_th b v_max
e_leak = -60e-3;
r_membrane = 100e6;
c_membrane = 100e-12;
g_leak = 8e-9;
v_threshold = -50e-3;
v_reset = -80e-3;
v_max = 10e-3;
e_k = -80e-3;
delta_g_sra = 1e-9;
tau_sra = 50e-3;
a = 10e-9;
delta_th = 2e-3;
b = 0.5e-9;


% 1.a. Produce vector of 40000 random values for the applied current.

I_rand = (rand(1, 40000)-0.5)*(1e-9);

global dt
dt = 0.02e-3;
t = 0:dt:(40000*5e-3);
I_app = zeros(1, length(t));


% 1.b. Fill vector for applied current.

index = 1;
nums = 1;
for n = 1:length(t)
    if nums ~= 251
        I_app(n) = I_rand(index);
        nums = nums + 1;
    else
        index = index + 1;
        I_app(n) = I_rand(index);
        nums = 1;
    end
end


% 1.c. Simulate the AELIF neuron.
```

```
i_sra = zeros(1, length(t));
v = zeros(1, length(t));
v(1) = e_leak;
[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

f1 = figure;
figure(f1);
plot(t, v_sim)
grid on
title('Membrane Potential vs. Time');
xlabel('Time');
ylabel('Membrane Potential');

% 2. Downsample the stimulus and spike vectors (see end of script for
% function definition).

new_dt = 1e-3;
t2 = 0:1e-3:(40000*5e-3);

I_app = expandbin(I_app, dt, new_dt);
spikes = expandbin(spikes, dt, new_dt);

spikes(find(spikes)) = 1;

f2 = figure;
figure(f2);
plot(t2, I_app)
grid on
title('I_{app} vs. Time');
xlabel('Time');
ylabel('I_{app}');

% 3. Compute spike-triggered average.
[sta, tcorr] = STA(I_app, spikes, new_dt, 75e-3, 25e-3);

f3 = figure;
figure(f3);
plot(tcorr, sta)
grid on
title('Spike-triggered Average vs. Time');
xlabel('Time (ms)');
ylabel('Spike-triggered Average');

%{

Comments on parameters:

Parameter a: increasing the parameter a causes the spike-triggered
 average
curve to exhibit a short upsweep to a local maximum, prior to the
 curve's
decay before its large spike. As a is decreased towards zero, all of
 the
```

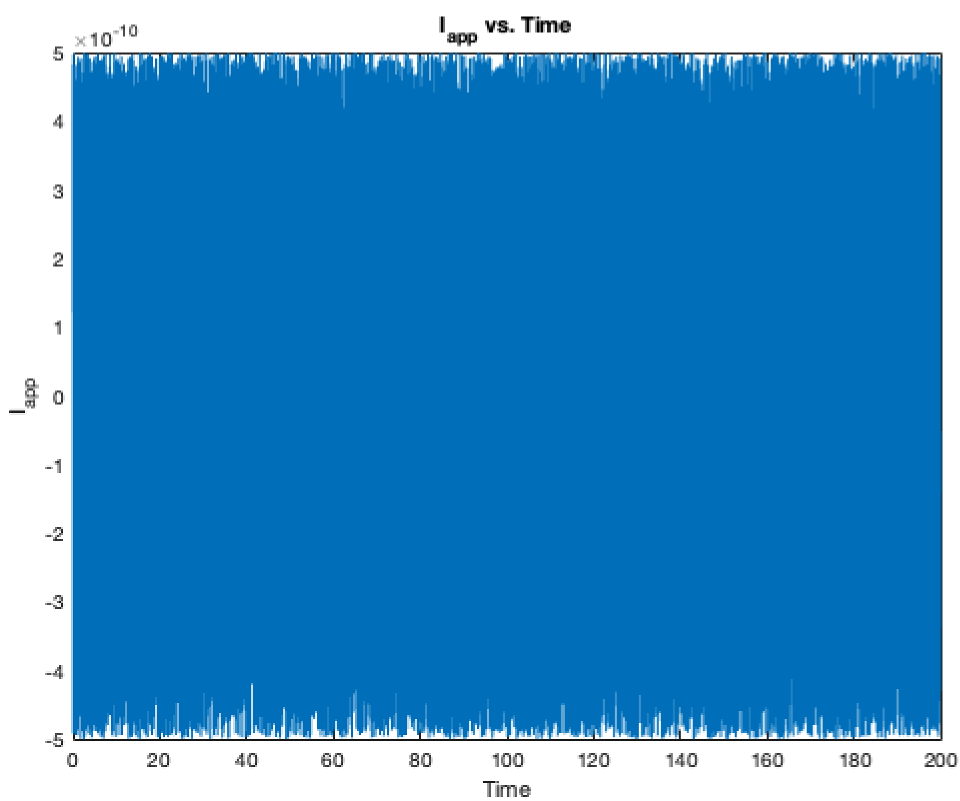
```
local maxima/minima prior to the large spike appear to disappear and
 the
curve is relatively constant before the large spike.

Parameter b: as b is increased, the spike-triggered average
takes a longer time to decay to its minimum prior to the large spike
in the curve. Similarly, as b is decreased, the spike-triggered
 average
reaches its minimum earlier, and remains at this minimum until the
 large
spike in the curve.

Parameter tau_sra: as tau_sra is increased, the local maxima/minima
 prior
to the large spike appear to disappear and the curve is relatively
constant before the large spike. Also, the maximum achieved by the
 large
spike decreases. As tau_sra is increased, the maximum achieved by the
 large
spike increases slightly, and the local maxima/minima prior
to the large spike appear to disappear and the curve is relatively
constant before the large spike.

Parameter delta_th: as delta_th is increases, the spike-triggered
 average
curve reaches its local minimum prior to the large spike quicker and
 stays
constant around that value longer - it also slightly decreases the
 maximum
achieved by the large spike. Decreasing the parameter doesn't seem to
affect the curve very much.

%}


%%%%%%%%%%%%%%%%%%%%%%%
% Function Definitions:

function [v_simulated, spike_vec, i_sra_simulated] = aelifsim(v,
 i_sra, t, i_applied)
% Simulates the membrane potential with given inputs.
global v_threshold v_reset dt b v_max c_membrane delta_th e_leak
 g_leak a tau_sra
v_simulated = zeros(1, length(v));
v_simulated(1) = v(1);
spike_vec = zeros(1, length(v));
i_sra_simulated = zeros(1, length(i_sra));
i_sra_simulated(1) = i_sra(1);
temp = length(v)-1;
for n = 1:temp
    if ( v_simulated(n) > v_max )
        v_simulated(n) = v_reset;
        i_sra_simulated(n) = i_sra_simulated(n) + b;
        spike_vec(n) = 1;
```

```
    end

    step1 = dt*( g_leak*(e_leak-v_simulated(n) +
 delta_th*exp((v_simulated(n)-v_threshold)/delta_th) ) ...
       - i_sra_simulated(n) + i_applied(n))/c_membrane;

    step2 = dt*( a*(v_simulated(n) - e_leak) - i_sra_simulated(n) )/
tau_sra;

    v_simulated(n+1) = v_simulated(n) + step1;
    i_sra_simulated(n+1) = i_sra_simulated(n) + step2;

end

end

function [result] = expandbin(initial_vector, initial_width,
 final_width)
% Downsamples the input vector.
old_length = length(initial_vector);
ratio = round(final_width/initial_width);

new_length = ceil(old_length/ratio);
result = zeros(1, new_length);

for n = 1:new_length-1
    result(n) = mean(initial_vector( (n-1)*ratio+1 : n*ratio) );
end

result(new_length) = mean(initial_vector( (new_length-1)*ratio :
 new_length ) );
end

function [sta, tcorr] = STA(Iapp, spikes, dt, tminus, tplus)
% Computes the spike-triggered average.

if (~exist('tminus'))
    tminus = 75e-3;
end
if (~exist('tplus'))
    tminus = 25e-3;
end

% Set nminus and nplus.
nminus = ceil(tminus/dt);
nplus = ceil(tplus/dt);

% Define tcorr.
tcorr = -dt*nminus:dt:dt*nplus;

% Setup sta vector.
sta = zeros(1,nminus+nplus+1);

% Find spikes.
```

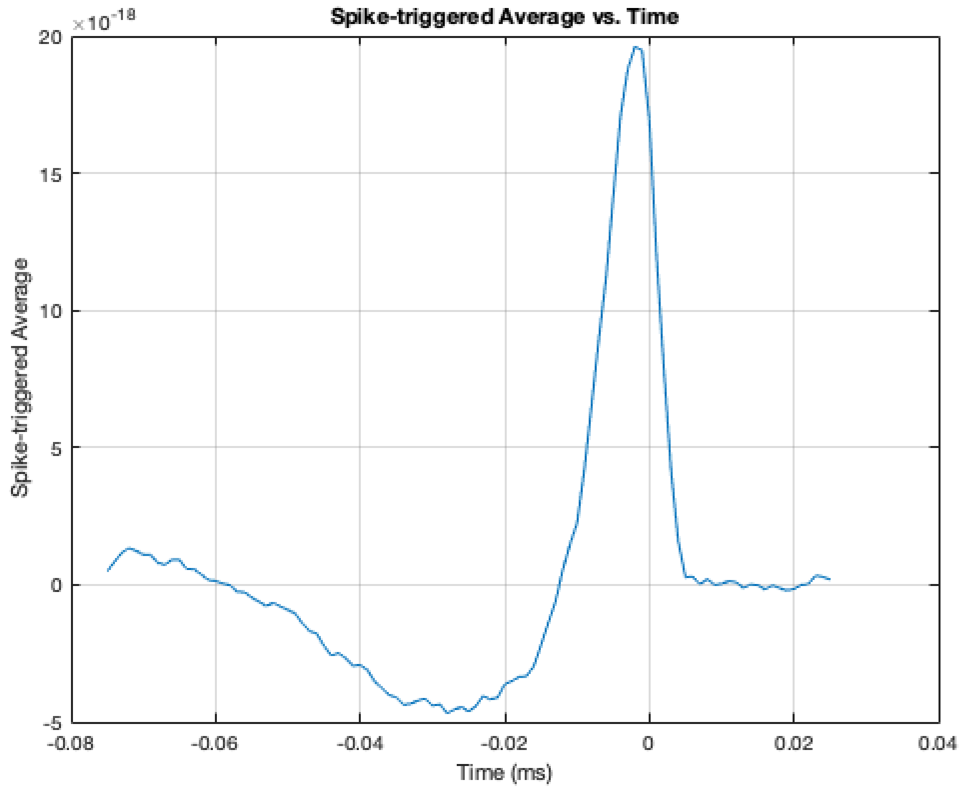
```
spike_indices = find(spikes);
nspikes = length(spikes);

% Loop over spike vector.
for n=1:length(spike_indices)

    index = spike_indices(n);

    % Check that indices never step outside of the bounds.

    if (index-nminus) < 1
        left = 1;
    else
        left = index-nminus;
    end

    if (index+nplus) > length(Iapp)
        right = length(spike_indices);
    else
        right = index+nplus;
    end

    % Add Iapp to sta vector.
    for i=left:right
        sta(i-index+nminus+1) = sta(i-index+nminus+1) + Iapp(i)/
nspikes;
    end
end

% Divide sta vector by the number of spikes.
sta = sta/nspikes;

end
```



Membrane Potential vs. Time
Membrane Potential
Time
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
-0.16
0
20
40
60
80
100
120
140
160
180
200



I_{app} vs. Time
$\times 10^{-10}$
I_{app}
Time
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
20
40
60
80
100
120
140
160
180
200



Published with MATLAB® R2018a