EE416 Term Project

Berken Utku Demirel - 2166221

1 Introduction

In the project, the model of an excitable membrane of an axon using the Hudgkin-Huxley (H&H) network model is implemented according to the rate constants for ionic channel conductivities determined by Hudgkin and Huxley. In addition to these constants, simulation has three user input variables. First one is the amplitude and duration of the stimulus current, second is the number of stimuli to be applied and the last is the time delay between the successive stimuli.
The section 2 explains how the H&H model is implemented, and the section 3 gives the result of the model with different input stimuli. Lastly, in the section 4 the results are discussed.

2 Theory

While simulating the Hudgkin and Huxley model, the state variables are used. State variables are the set of variables whose values, when known at a particular time, allow the other variables in the problem to be calculated for that same time. In the simulation of membrane action potentials using Hodgkin–Huxley model, the state variables are v_m, n, m, and h. In other words, it means that all the other time-varying quantities, such as I_K and I_{Na}, can be found from the values of the state variables. However, in our project, the state variables are n, m, h and the membrane current, the calculation of the other time-varying quantities are given below.

Living membrane exists continuously throughout its lifetime, so it has no fixed starting points. However, for the project, I assume that the stimulus consists of a depolarizing current $I_s(t)$ that begins at $t = 0$ and lasts for the input value which the user determines.
In order to find ΔV_m, which is the incremental change of V_m during time step Δt, the following formula is used.

$$\begin{aligned} \Delta V_m^i &= \frac{\Delta t}{C_m}[I_m^i - I_{ion}^i] \\ &= \frac{\Delta t}{C_m}[I_m^i - I_K^i - I_{Na}^i - I_L^i] \end{aligned} \tag{1}$$

where the ionic current are given as below

$$\begin{aligned} I_K^i &= g_K^i(V_m^i - E_K) \\ I_{Na}^i &= g_{Na}^i(V_m^i - E_{Na}) \\ I_L^i &= g_L^i(V_m^i - E_L) \end{aligned} \tag{2}$$

where the conductivities are

$$
\begin{aligned}
g_K^i &= n_i^4(V_m^i - E_K) \\
g_{Na}^i &= m_i^3 h_i(V_m^i - E_{Na})
\end{aligned} \tag{3}
$$

The above equations have a superscript i. The presence of the indexing i identifies each one as a quantity that has a value that changes from one time to another; quantities without an index hold constant values with time. The fact that all indices are i (rather than, say, mixed with i+1) signifies that the equations hold when all the quantities are for the same time instant.Though time varies continuously, for numerical analysis time is discretized into a sequence of particular time instants. Each time in the sequence is separated from the next by a time interval Δt.

3 Results

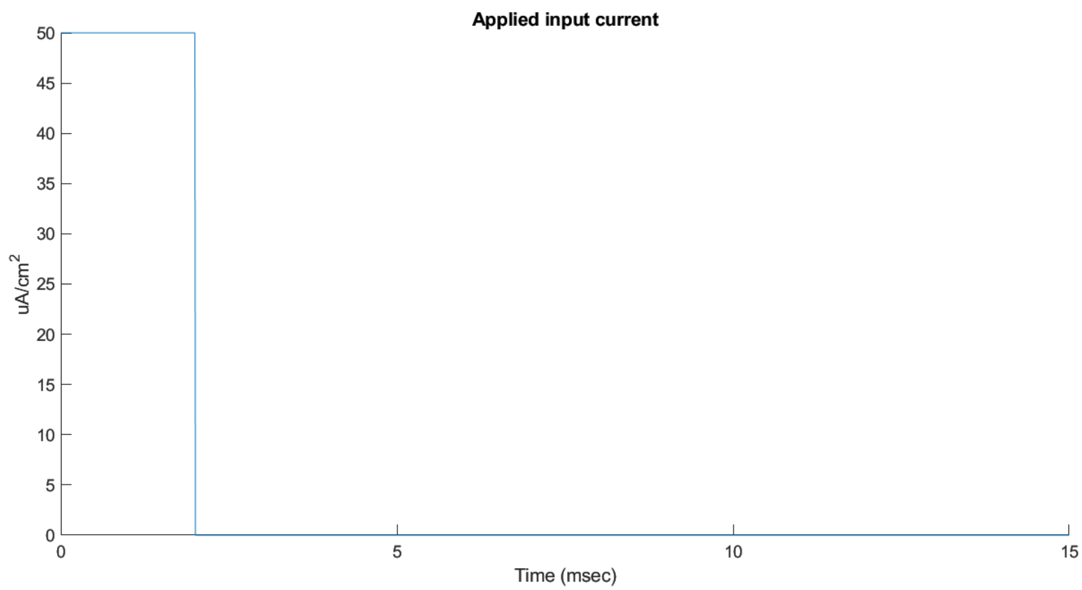

Figure 1: Applied stimulus

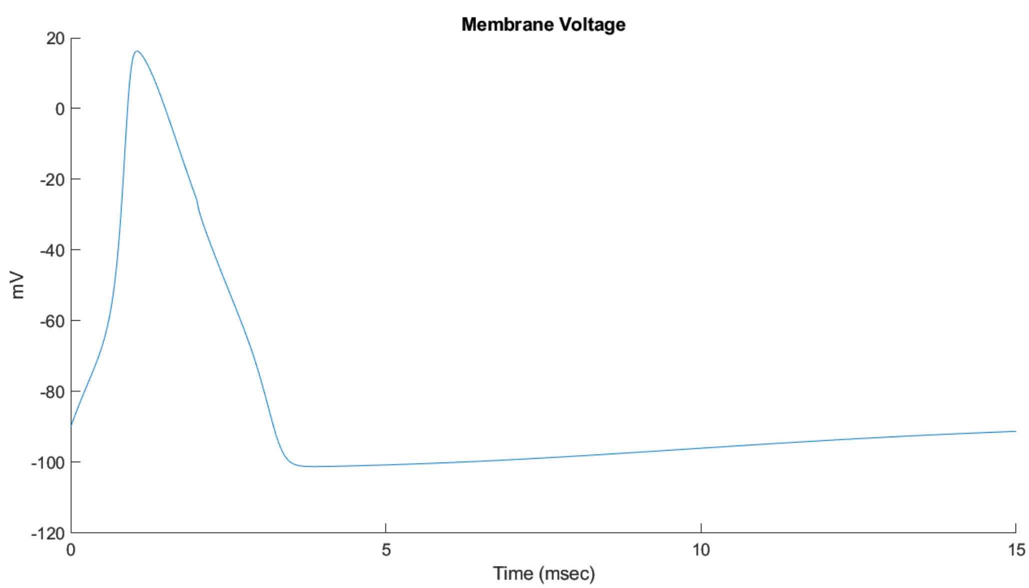

Figure 2: Membrane Potential

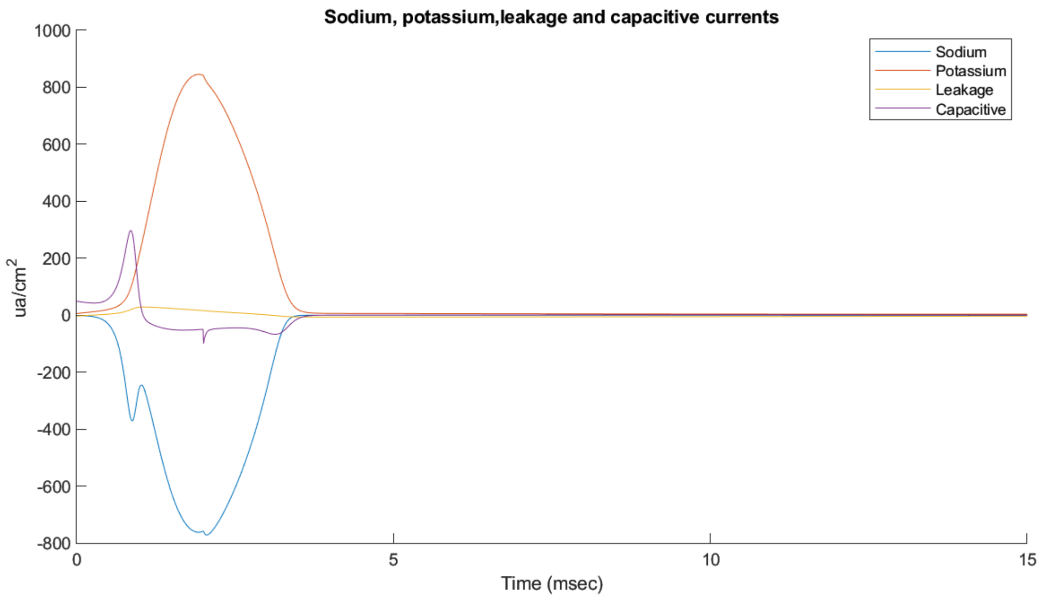

Figure 3: Currents

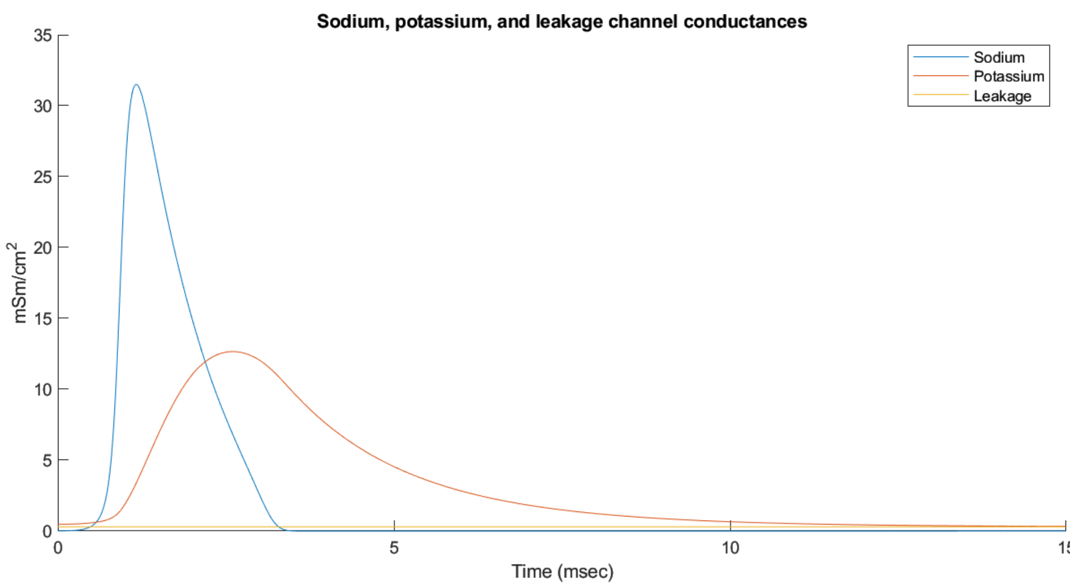

Figure 4: Conductances

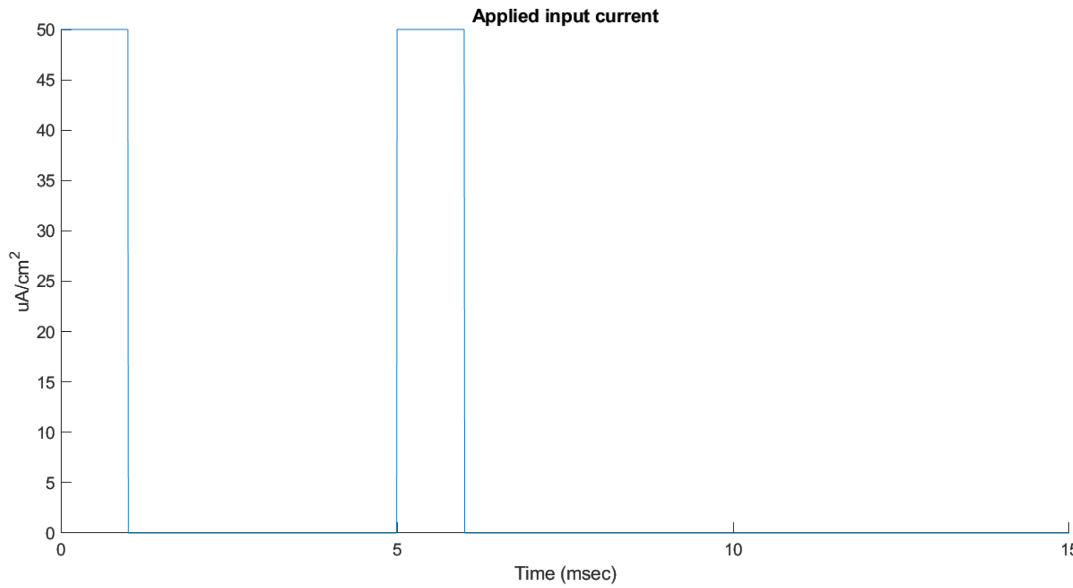

Figure 5: Applied Input

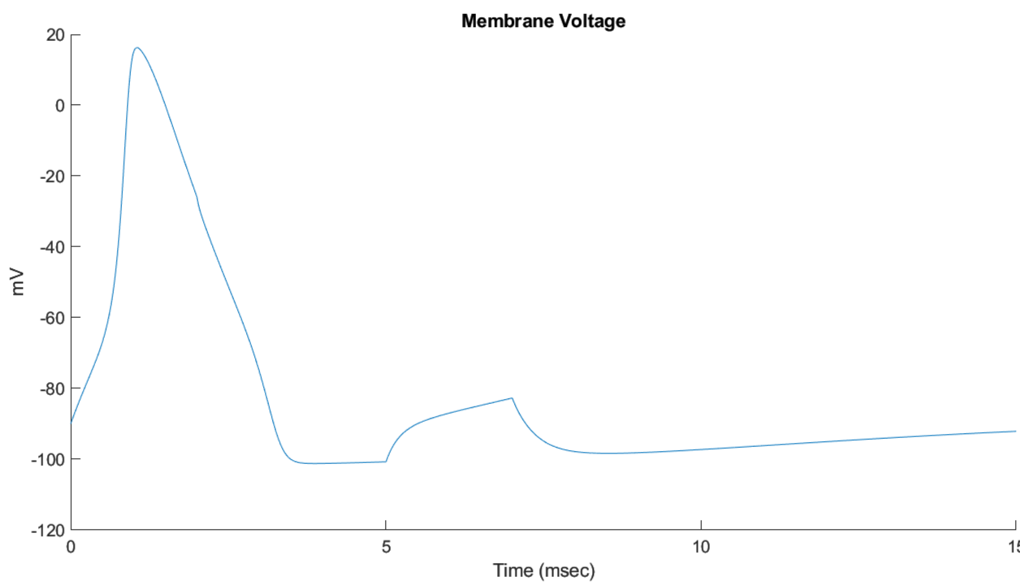

Figure 6: Membrane Potential

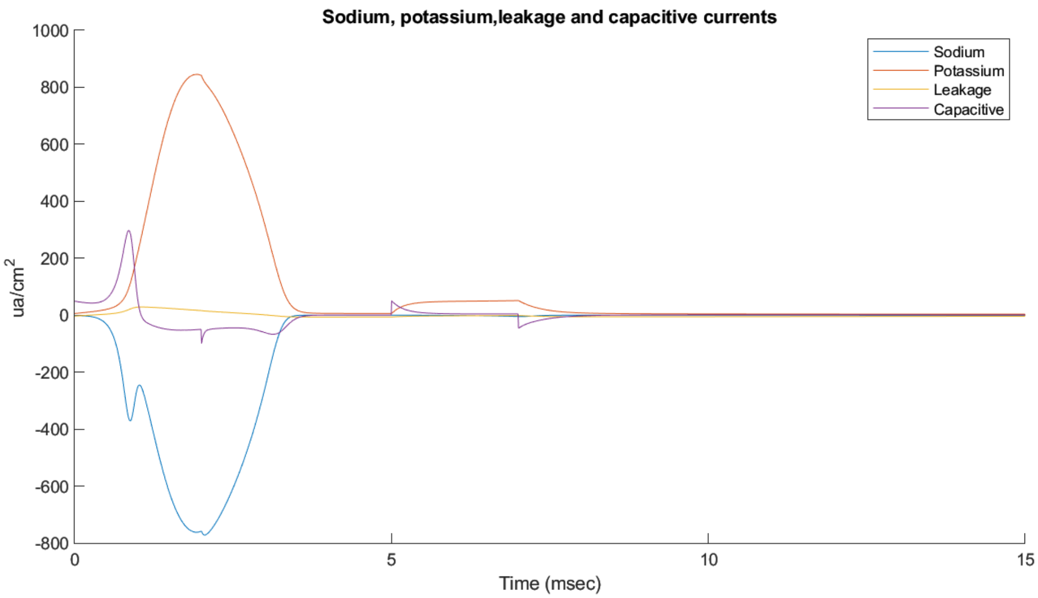

Figure 7: Currents

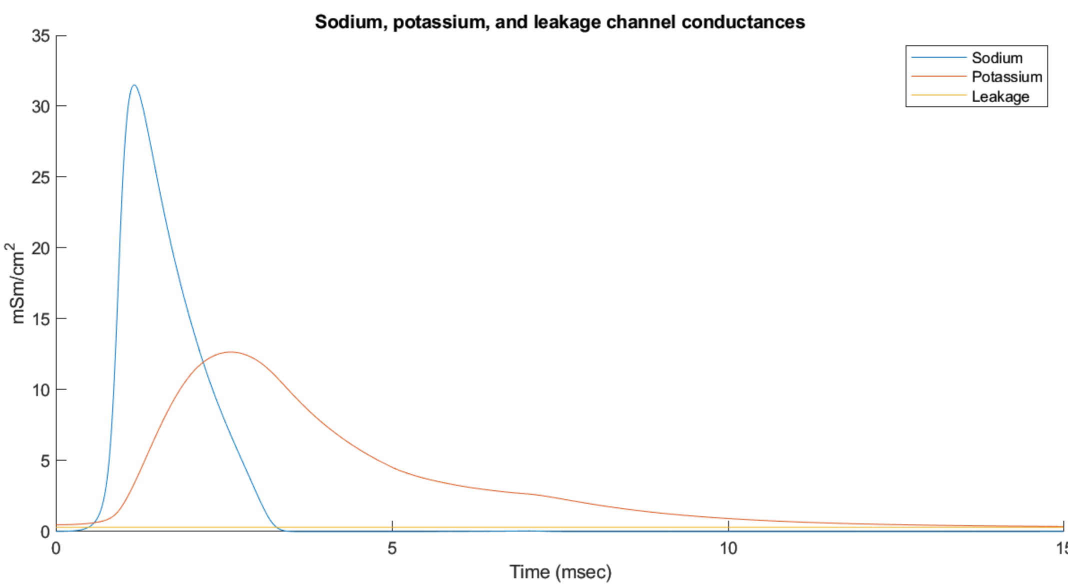

Figure 8: Conductances

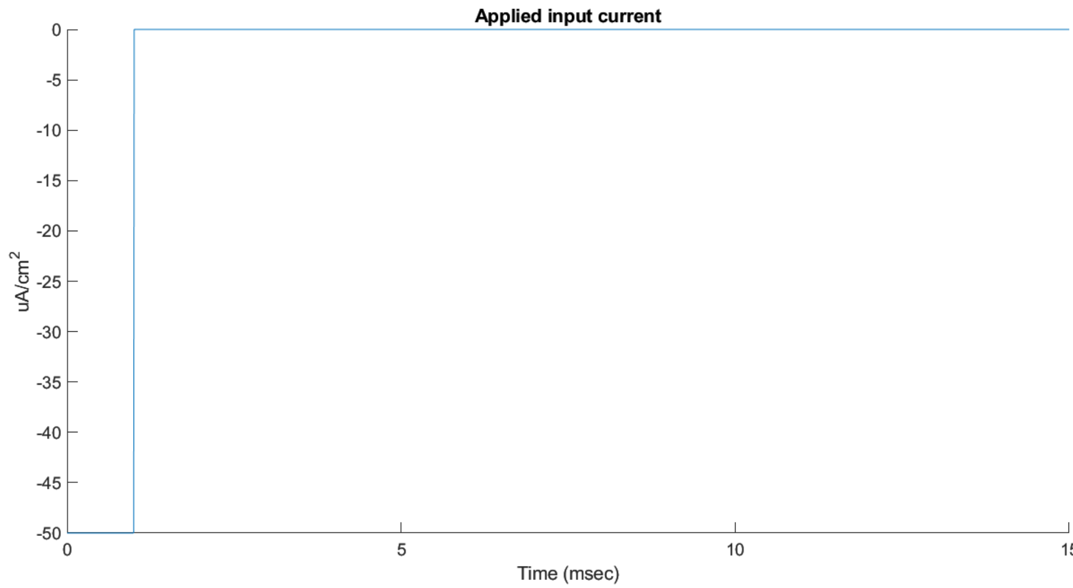

Figure 9: Applied Input

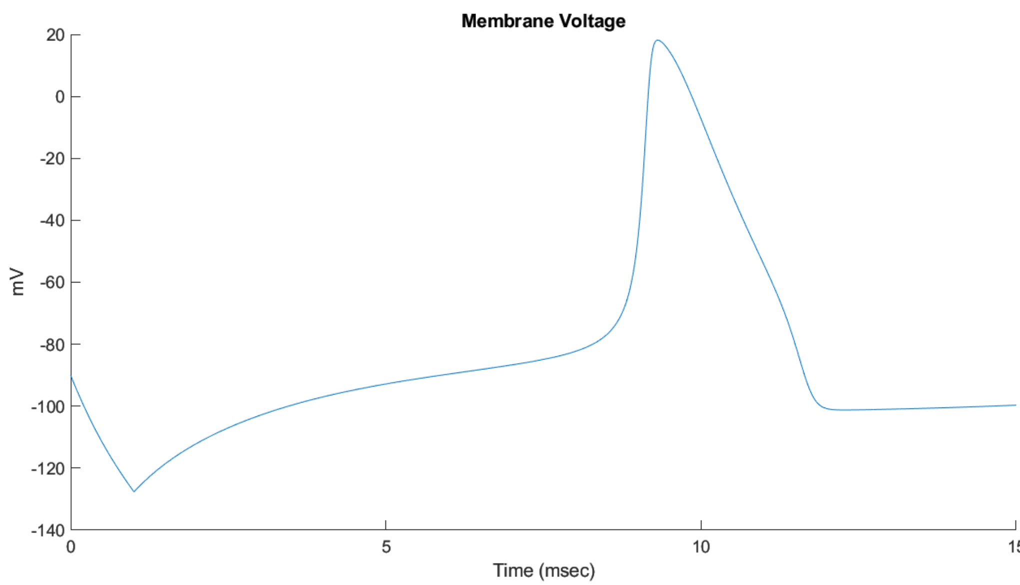

Figure 10: Membrane Potential

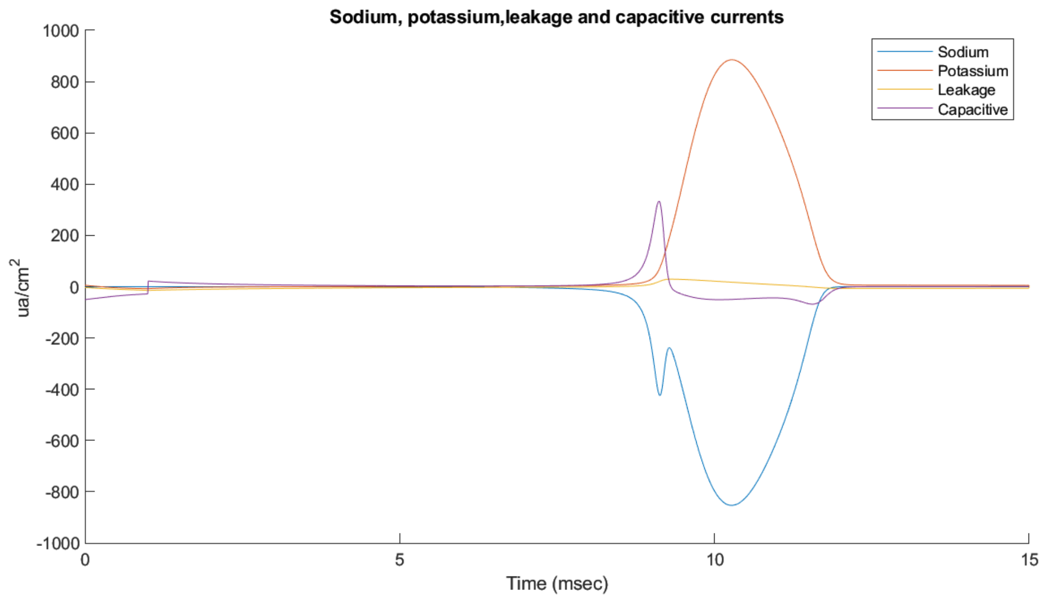

Figure 11: Currents

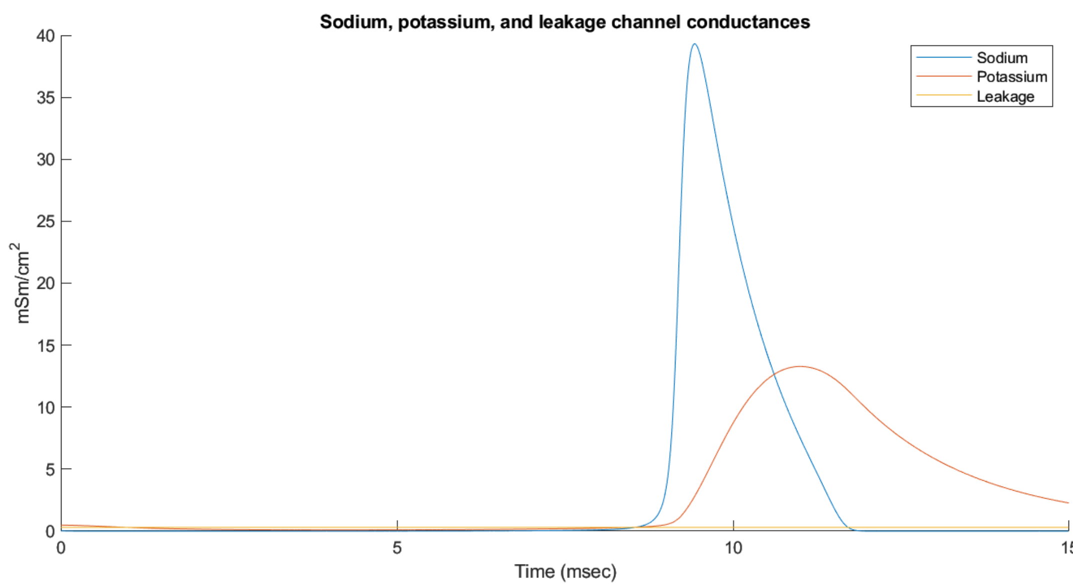

Figure 12: Conductances

4 Discussion and Conclusion

The stimulus is at the top (figure 1). Below it are the calculated changes in membrane potential (figure 2), sodium and potassium conductances (figure 4), and sodium and potassium currents (figure 3).
It is important that g_{Na} and g_K follow a different period, as is seen by the figure 4, as these differences give rise to the observed duration of V_m. It also is remarkable that at first glance,I_K and I_{Na} have wave shapes that are more or less identical except for opposite polarity. In figures, we can deduce that the rapid rise and decay of $g_{Na}(t)$. In contrast, $g_k(t)$ has a delayed rise and more lasting elevation in magnitude. This behaviour might have been expected as a result of what was learned from the voltage clamp measurement.

In the figure 5, the second stimulus is seen to elicit essentially no response even though it is of the same size and duration as the first (for which an action potential results, as is seen). Therefore, It identifies the condition as refractory. Since a larger stimulus would generate an action potential, this is a relatively refractory period. The stimulus amplitude is $50\mu A/cm^2$, and its duration is 1 msec.The second stimulus is same as in amplitude and duration and occurs after a delay of 5 msec.

The figures (10, 11 and 12) show that happens after the termination of a 1 msec hyperpolarization and the sudden restoration of normal transmembrane potential. This is because the release of the hyperpolarization, the value of h is increasing while m and n are decreasing. However, m reached its normal value rapidly. This accounts for the name given it: anode break excitation.

Appendix

Table of Contents


```
% Berken Utku Demirel - 2166221
clearvars
close all

% User inputs
current_amplitude = 50; % mA
current_duration = 1; % msec

flag = true;
if flag == true
    time_delay = 5; % msec
end

% Duration
duration = linspace(0,15,2001);  % 0 ms to 200 ms
durationStep = duration(2)- duration(1);

% Current definition
if flag == true
    current = current_amplitude * ones(1,length(duration));
    current((current_duration < duration)&(duration< time_delay)) = 0;
    current(duration < 0) = 0;
    current(duration > current_duration + time_delay) = 0;
else
    current = current_amplitude * ones(1,length(duration));
    current(duration > current_duration) = 0;
    current(duration < 0) = 0;
end

% Na
current_Na = ones(1,length(duration));
current_Na(duration < 0) = 0;
% K
current_K = ones(1,length(duration));
current_K(duration < 0) = 0;
% L
current_L = ones(1,length(duration));
current_L(duration < 0) = 0;
% C
current_C = ones(1,length(duration));
current_C(duration < 0) = 0;

%define constants
GNArest = 120;
```

```
GKrest = 36;
GL = 0.3;
Cm = 1;
Vrest = -90;
VNA = 25;
VK = -102;
VL = -79.387;

%definition of initials
Vmembrane = Vrest;
m = 0.05*ones(1,length(duration));
h = 0.54*ones(1,length(duration));
n = 0.34*ones(1,length(duration));
alpha_m = 2.237*ones(1,length(duration));
alpha_n = 0.0582*ones(1,length(duration));
alpha_h = 0.07*ones(1,length(duration));
beta_m = 4*ones(1,length(duration));
beta_n = 0.125*ones(1,length(duration));
beta_h = 0.0474*ones(1,length(duration));
```

Loop for simulation

```
for i = 2:length(duration)
alpha_m(i) = alphaM(Vmembrane(i-1),Vrest);
alpha_h(i) = alphaH(Vmembrane(i-1),Vrest);
alpha_n(i) = alphaN(Vmembrane(i-1),Vrest);
beta_m(i) = betaM(Vmembrane(i-1),Vrest);
beta_h(i) = betaH(Vmembrane(i-1),Vrest);
beta_n(i) = betaN(Vmembrane(i-1),Vrest);

%m h and n values
m(i) = m(i-1) + durationStep*(alpha_m(i-1)*(1-m(i-1))-
 beta_m(i-1)*m(i-1));
h(i) = h(i-1) + durationStep*(alpha_h(i-1)*(1-h(i-1))-
 beta_h(i-1)*h(i-1));
n(i) = n(i-1) + durationStep*(alpha_n(i-1)*(1-n(i-1))-
 beta_n(i-1)*n(i-1));
% ionic conductances
GNA(i-1) = GNArest * m(i-1)^3*h(i-1);
GK(i-1) = GKrest * n(i-1)^4;
GL(i-1) = 0.3;
% Currents
current_K(i-1) = GK(i-1) * (-VK + Vmembrane(i-1));
current_Na(i-1) = GNA(i-1) * (-VNA + Vmembrane(i-1));
current_L(i-1) = GL(i-1) * (-VL + Vmembrane(i-1));

Vmembrane(i) = Vmembrane(i-1) + durationStep * (current(i-1) -
 current_K(i-1) - current_Na(i-1) - current_L(i-1)) / Cm;
current_C(i-1) = current(i-1) - current_K(i-1) - current_Na(i-1) -
 current_L(i-1);
current_total(i-1) = current_K(i-1) + current_Na(i-1) + current_L(i-1)
 + current_C(i-1);
end
```

Plots

```
% Comment for publishing
% figure,
% plot(duration,current)
% title('Applied input current')
% ylabel('\muA/cm^2')
% xlabel('Time (msec)')
%
% figure,
% plot(duration(1:end-1), current_total)
% title('The total membrane current')
% ylabel('\muA/cm^2')
% xlabel('Time (msec)')
%
% figure,
% plot(duration(1:end-1),GNA)
% hold on
% plot(duration(1:end-1),GK)
% hold on
% plot(duration(1:end-1),GL)
% title('Sodium, potassium, and leakage channel conductances')
% ylabel('mSm/cm^2')
% xlabel('Time (msec)')
% legend('Sodium','Potassium','Leakage')
%
% figure,
% plot(duration,current_Na)
% hold on
% plot(duration,current_K)
% hold on
% plot(duration,current_L)
% hold on
% plot(duration,current_C)
% title('Sodium, potassium,leakage and capacitive currents')
% xlabel('Time (msec)')
% ylabel('\muA/cm^2')
% legend('Sodium','Potassium','Leakage','Capacitive')
%
% figure,
% plot(duration, Vmembrane)
% xlabel('Time (msec)')
% ylabel('mV')
% title('Membrane Voltage')
```

Alpha and Beta functions

```
function aM = alphaM(V,Vrest)
aM = (2.5-0.1*(V-Vrest)) ./ (exp(2.5-0.1*(V-Vrest)) -1);
end

function bM = betaM(V,Vrest)
```

```
bM = 4*exp(-(V-Vrest)/18);
end

function aH = alphaH(V,Vrest)
aH = 0.07*exp(-(V-Vrest)/20);
end

function bH = betaH(V,Vrest)
bH = 1./(exp(3.0-0.1*(V-Vrest))+1);
end

function aN = alphaN(V,Vrest)
aN = (0.1-0.01*(V-Vrest)) ./ (exp(1-0.1*(V-Vrest)) -1);
end

function bN = betaN(V,Vrest)
bN = 0.125*exp(-(V-Vrest)/80);
end
```

Published with MATLAB® R2019b