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