. In the name of GOD .

A study on

Simulate the action potentials generation of a neuron

Tarbiat Modares University

Presented to

Dr. Zahra Bahmani

Submitted by

Poorya Aghaomidi

9961391001

Jan 2021

Q1

1.1. Link :

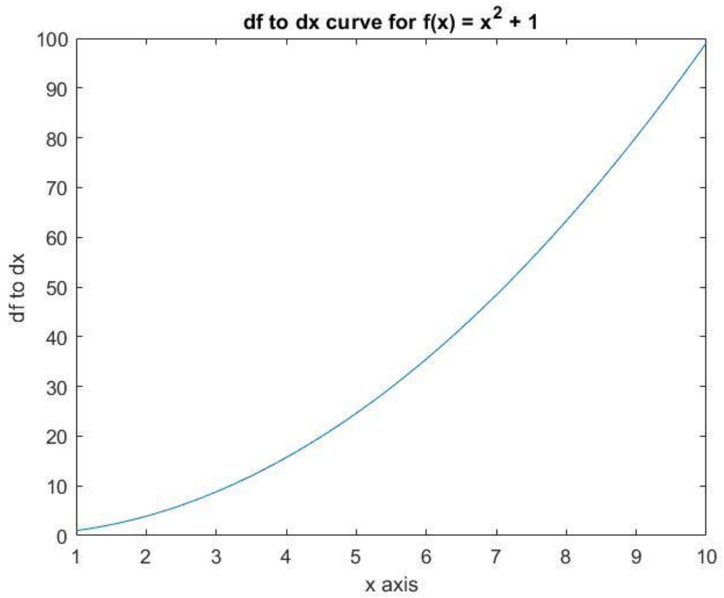
[..\1](..\1)

1.2. Code :

```
1.2.1 :


function f = ode_euler(func, x, f_o)
% function arguments : x and f_o
% x   : a vector that specifies the time points that the f should be
approximated for
% f_o : the initial condition
% f   : representing the approximate solution to the differential
equation with f(0)=f_o

delta_x=x(2)-x(1);
% delta_x : as the difference between successive x values

l_x=length(x);
% Determine how many points we need to approximate by finding the length
of vector x

f=zeros(1, l_x);
% Initialize f by creating a vector of the right length
% We will reset the elements to the correct values in the for loop below

f(1)=f_o;
% Set the initial value of f to f_o

%Use a for-loop to implement Equation :
for i = 1:(l_x-1)
f(i+1) = f(i) + delta_x * feval(func,x(i));
end;
```

```
1.2.2 :

function df = f_prime(x)
% This function takes a point x and calculates the derivative of f at
the point x

% For example f(x) = x^2 + 1    :
df=2*x;
f(1)=f_o;
% Set the initial value of f to f_o

%Use a for-loop to implement Equation :
for i = 1:(l_x-1)
f(i+1) = f(i) + delta_x * feval(func,x(i));
end;
```

1.3. Figure :



1.4. Explanation :

در این سوال تابعی تعریف کردیم که با توجه به قضیه اویلر و روابط آن، مشتق یک تابع دیگر را در نقاط داده شده محاسبه کند.

برای بررسی صحت این کد، تابع f(x) = x^2 + 1 را به برنامه دادیم و نمودار مشتق آنرا رسم کردیم.

جزئیات کدها و نحوه عملکرد هر خط در تمامی سوال ها بصورت کامنت زیر هر خط کد نوشته شده است.

Q2

2.1. Link :

..\2

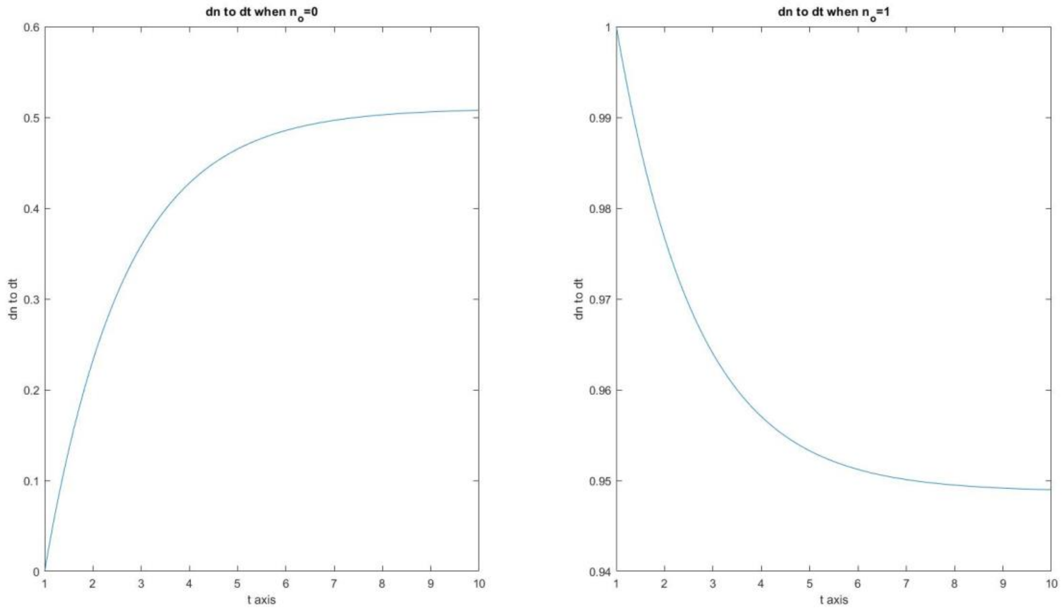
2.2. Code :

```
2.2.1 :


function dn_dt = n_prime(time, n_o, V)
% function arguments : t and n_o and V
% t      : a vector that specifies the time points that the n should be
approximated for
% n_o    : the initial condition
% V      : membrane potential
% dn_dt : representing ( dn / dt )

Vrest = -85 ;
% Assume that the amount of rest potential is -85

[ alfa_n , beta_n ] = transition_rate_n(V , Vrest) ;
% Calling transition_rate_n function to calculate alfa and beta for n

TAW_n = 1 / (alfa_n + beta_n) ;
INF_n = alfa_n / (alfa_n + beta_n) ;
% Calculating required parameters to achieve dn/dt value

syms n(t);
% Define n as a function of t

n(t) = INF_n - (INF_n - n_o)*exp((-t)/TAW_n) ;
% Define the equation for n(t)

diffn = diff(n,t) ;
% Derivative of the function n(t)

% Calculate dn/dt :
dn_dt = diffn(time) ;
```

```
2.2.2 :

function [ alfa_n , beta_n ] = transition_rate_n(Vm , Vrest)
% function arguments : Vm and Vrest
% Vm      : membrane potential
% Vrest   : resting potential
% alfa_n : transition rate between open and close states ( close to open
)
% beta_n : transition rate between open and close states ( open to close
)

v = Vm - Vrest ;
% Calculate v as required in alfa and beta formula

% Calculate alfa and beta for n
alfa_n = (0.01 * (10 - v)) / (exp((10 - v) / 10) - 1) ;
beta_n = 0.125 * exp((-v)/80) ;
```

```
2.2.3 :


function n = ode_euler_modified(nprime, t, n_o, V)
% function arguments : t and n_o and V
% t   : a vector that specifies the time points that the n should be
approximated for
% n_o : the initial condition
% V   : membrane potential
% n   : representing the approximate solution to the differential
equation with n(0)=n_o

delta_t=t(2)-t(1);
% delta_t : as the difference between successive t values

l_t=length(t);
% Determine how much time we need to approximate by finding the length
of vector t

n=zeros(1, l_t);
% Initialize n by creating a vector of the right length
% We will reset the elements to the correct values in the for loop below

n(1)=n_o;
% Set the initial value of n to n_o

%Use a for-loop to implement Equation :
for i = 1:(l_t-1)
n(i+1) = n(i) + delta_t * feval(nprime ,t(i) ,n_o ,V);
end;
```

2.3. Figure :



2.4. Explanation :

در این سوال دو تابع تعریف کردیم که یکی از آنها مقدار ولتاژ غشا و پتانسیل استراحت را می گیرد و با توجه به روابط گفته شده در اسلاید های درس، آلفا و بتا را برای n محاسبه می کند. تابع دیگر با فراخوانی تابع قبلی، آلفا و بتا را در یک ولتاژ مشخص می گیرد و مشتق n نسبت به t را در هر لحظه داده شده محاسبه می کند. برای اینکه خود n را در هر لحظه داشته باشیم از تابع نوشته شده در سوال 1 استفاده می کنیم، اما در آن تغییراتی ایجاد می کنیم تا مقدار ولتاژ را به عنوان ورودی بگیرد و به زیر تابع های خود بدهد.(زیرا مقادیری مانند آلفا و بتا به مقدار ولتاژ وابسته هستند)

در نتیجه می توانیم با دادن زمان، ولتاژ و مقدار اولیه n به تابع n_prime مقدار n را در هر لحظه داشته باشیم.

همچنین نمودار تغییرات n را در مدت زمان 10 ثانیه و در ولتاژ منفی 20 میلی ولت به ازای دو مقدار اولیه صفر و یک، رسم کنیم.

Q3

3.1. Link :

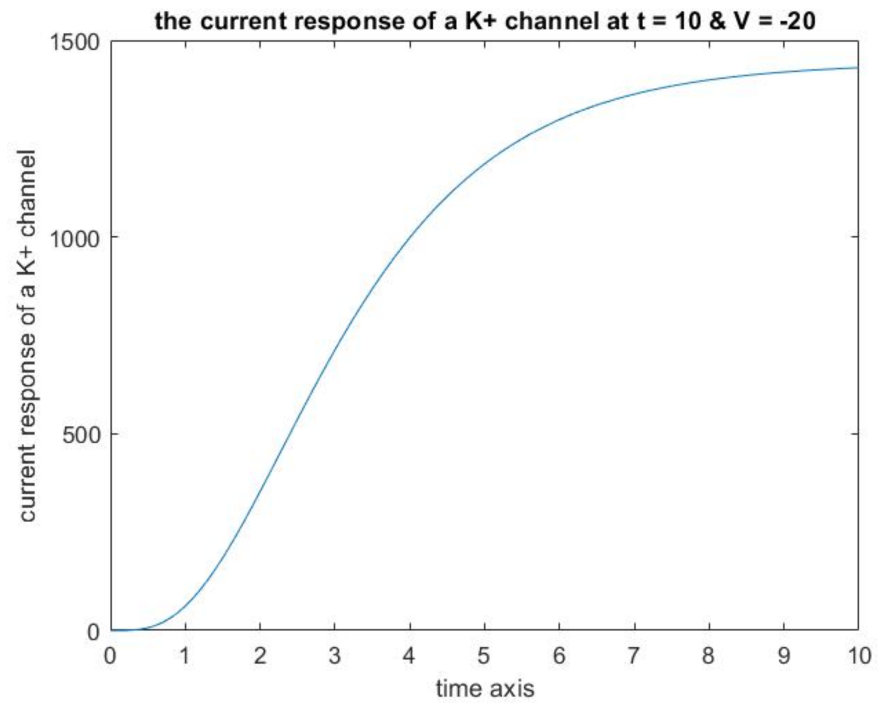
..\3

3.2. Code :

```
3.2.1 :

function Ik = K_v(t , V)
% function arguments : t and V
% t  : a time interval to study the current response
% V  : a holding potential that causes the current response
% Ik : the current response of a K+ channel

g_k_avg = 36 ;
E_k = -72.1 ;
% Attribute defult values to required parameters
% Source : BIOELECTRICITY:AQUANTITATIVE APPROACH _ Table 13.2

time = [0:0.1:t] ;
% Set the time array

n_o = 0 ;
% Set the initial condition

n = ode_euler_modified(@n_prime, time, n_o, V);
% Call ode_euler_modified function to calculate the dn

n4 = n.^4 ;
% Calculate n^4

% calculate the current response of a K+ channel :
Ik = g_k_avg * n4 * (V - E_k);
```

```
3.2.2 :

function n = ode_euler_modified(nprime, t, n_o, V)
% function arguments : t and n_o and V
% t : a vector that specifies the time points
% n_o : the initial condition
% V   : membrane potential
% n : representing the approximate solution to the differential equation

delta_t=t(2)-t(1);
% delta_t : as the difference between successive t values

l_t=length(t);
% Determine how much time we need to approximate by finding the length
of vector t

n=zeros(1, l_t);
% Initialize n by creating a vector of the right length
% We will reset the elements to the correct values in the for loop below

n(1)=n_o;
% Set the initial value of n to n_o

%Use a for-loop to implement Equation :
for i = 1:(l_t-1)
n(i+1) = n(i) + delta_t * feval(nprime ,t(i) ,n_o ,V);
end;
```

3.3. Figure :



3.4. Explanation :

در این سوال یک تابع تعریف کردیم که با گرفتن n از توابع نوشته شده، جریان کانال پتاسیم را در ولتاژ و زمان داده شده محاسبه می کند.

در این تابع فرض کردیم که همه کانال ها در لحظه اول بسته می باشند. (یعنی مقدار اولیه n برابر صفر فرض شده) جریان بدست آمده با جریان گفته شده به عنوان late current مطابقت دارد و تغییر جریان در نتیجه ولتاژ کلمپ را به ما نشان می دهد.

همچنین نمودار تغییرات جریان کانال پتاسیم در حالتی که غشا فقط به +K نفوذپذیر باشد را در مدت زمان 10 ثانیه و در ولتاژ منفی 20 میلی ولت رسم کنیم.

Q4

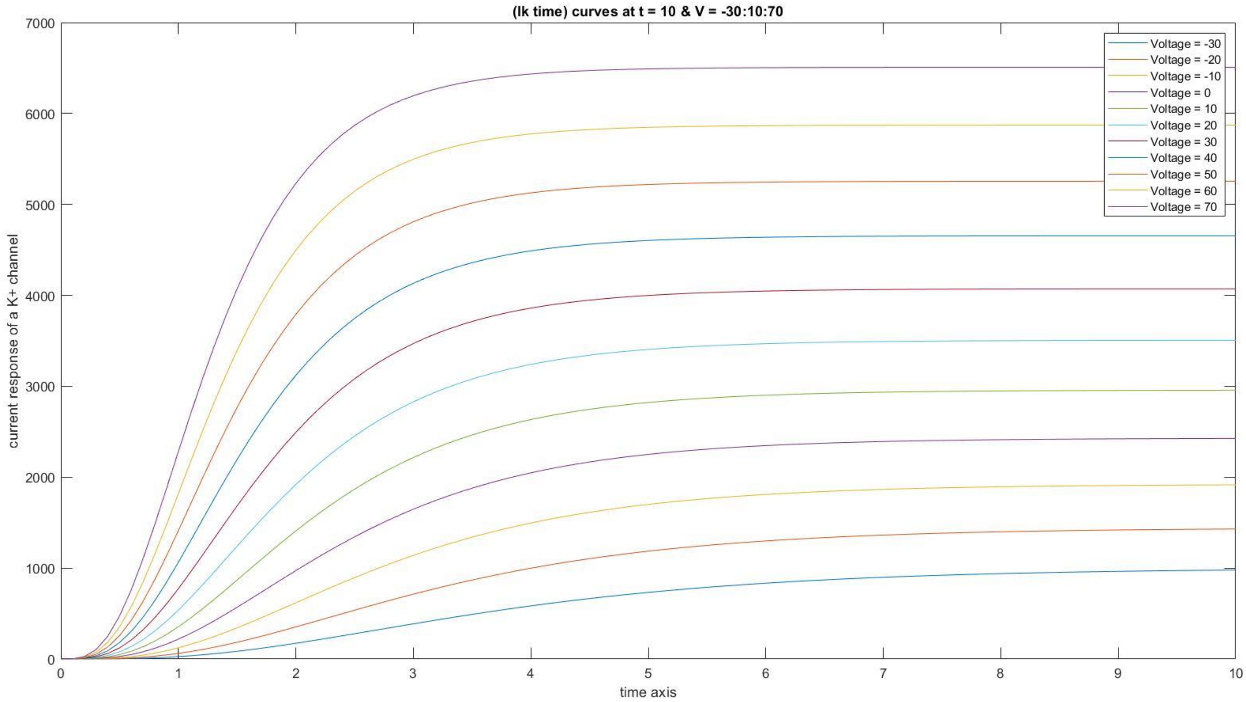
4.1. Link :

[..\4](..\4)

4.2. Code :

```
4.2.1 :

% Poorya Aghaomidi
% 9961391001
% Question_4 , main
% Goal : display Ik_time curves at t = 10 & V = -30:10:70

clear all; close all; clc;

t = 10 ;
time=[0:0.1:t] ;
Lt = length(time) ;
% Set a constant time duration and its array and length

V = [-30:10:70] ;
Lv = length(V) ;
% Initialize voltage array and calculate its length

Ik = zeros(Lv,Lt) ;
% Initialize current matrix
% Row : number of given voltage values ( V = [-30:10:70] )
% Column : number of given time values ( t = [0:0.1:10] )

% A for loop to display Ik_time curves for all given voltag values :
for i = 1:Lv

    Ik(i,:) = K_v( t , V(i)) ;
    % Call K_v function to calculate the current response of a K+
channel

    lgnd = ['Voltage = ',num2str(V(i))] ;
    % Set information to display legend

    plot(time,Ik(i,:),'DisplayName',lgnd), title('(Ik time) curves at t
= 10 & V = -30:10:70')
    xlabel('time axis'),ylabel('current response of a K+ channel');
    % Display Ik_time curve

    % Check the end of the plots :
    if i ~= Lv
        hold on
    else
        hold off
        legend ;
        % Show legend
    end

end

savefig('Ik_time_diff_V.fig');
% Save the curves in a single figure in current folder
```

4.3. Figure :



4.4. Explanation :

در این سوال با توجه به تابعی که برای بدست آوردن جریان پتاسیمی در سوال قبل نوشته شد، نمودار تغییرات جریان کانال پتاسیم در حالتی که غشا فقط به K+ نفوذپذیر باشد را در مدت زمان 10 ثانیه و در ولتاژ های متفاوت رسم کردیم.

جریان های بدست آمده با جریان گفته شده به عنوان late current مطابقت دارد و تغییر جریان در نتیجه ولتاژ کلمپ های متفاوت را به ما نشان می دهد. همانطور که میبینیم با افزایش ولتاژ، جریان کانال های K+ نیز افزایش می یابد.

Q5

5.1. Link :

[..\5](..\5)

5.2. Code :

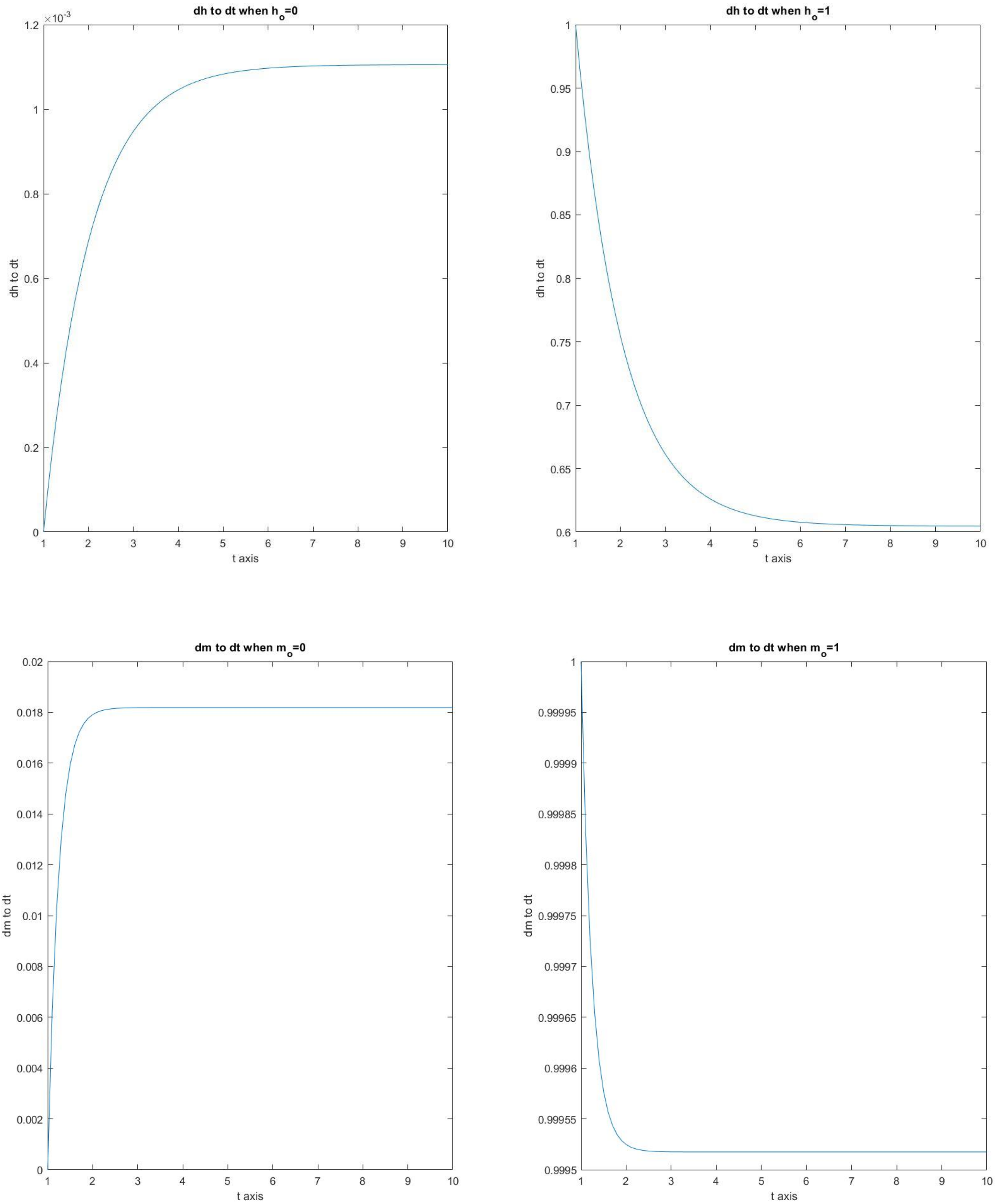
```
5.2.1 :

function [ alfa_m , beta_m ] = transition_rate_m(Vm , Vrest)
% function arguments : Vm and Vrest
% Vm      : membrane potential
% Vrest   : resting potential
% alfa_m : transition rate between open and close states (close to open)
% beta_m : transition rate between open and close states (open to close)

v = Vm - Vrest ;
% Calculate v as required in alfa and beta formula

% Calculate alfa and beta for m
alfa_m = (0.1 * (25 - v)) / (exp(0.1*(25-v)) - 1) ;
beta_m = 4 * exp((-v)/18) ;
```

```
5.2.2 :

function dm_dt = m_prime(time, m_o, V)
% function arguments : t and m_o and V
% t      : a vector that specifies the time points that the m should be
approximated for
% m_o    : the initial condition
% V      : membrane potential
% dm_dt : representing ( dm / dt )

Vrest = -85 ;
% Assume that the amount of rest potential is -85

[ alfa_m , beta_m ] = transition_rate_m(V , Vrest) ;
% Calling transition_rate_m function to calculate alfa and beta for m

TAW_m = 1 / (alfa_m + beta_m) ;
INF_m = alfa_m / (alfa_m + beta_m) ;
% Calculating required parameters to achieve dm/dt value

syms m(t);
% Define m as a function of t

m(t) = INF_m - (INF_m - m_o)*exp((-t)/TAW_m) ;
% Define the equation for m(t)

diffm = diff(m,t) ;
% Derivative of the function m(t)

% Calculate dm/dt :
dm_dt = diffm(time) ;
```

```
5.2.3 :

function [ alfa_h , beta_h ] = transition_rate_h(Vm , Vrest)
% function arguments : Vm and Vrest
% Vm      : membrane potential
% Vrest   : resting potential
% alfa_m : transition rate between open and close states ( close to open
)
% beta_m : transition rate between open and close states ( open to close
)

v = Vm - Vrest ;
% Calculate v as required in alfa and beta formula

% Calculate alfa and beta for h
alfa_h = 0.07 * exp((-v)/20) ;
beta_h = 1 / (exp((30-v)/10) + 1) ;
```

```
5.2.4 :

function dh_dt = h_prime(time, h_o, V)
% function arguments : t and h_o and V
% t      : a vector that specifies the time points that the h should be
approximated for
% h_o    : the initial condition
% V      : membrane potential
% dm_dt : representing ( dh / dt )

Vrest = -85 ;
% Assume that the amount of rest potential is -85

[ alfa_h , beta_h ] = transition_rate_h(V , Vrest) ;
% Calling transition_rate_h function to calculate alfa and beta for h

TAW_h = 1 / (alfa_h + beta_h) ;
INF_h = alfa_h / (alfa_h + beta_h) ;
% Calculating required parameters to achieve dh/dt value

syms h(t);
% Define h as a function of t

h(t) = INF_h - (INF_h - h_o)*exp((-t)/TAW_h) ;
% Define the equation for h(t)

diffh = diff(h,t) ;
% Derivative of the function h(t)

% Calculate dh/dt :
dh_dt = diffh(time) ;
```

5.3. Figure :



dh to dt when h_o=0
dh to dt
t axis
dh to dt when h_o=1
dh to dt
t axis
dm to dt when m_o=0
dm to dt
t axis
dm to dt when m_o=1
dm to dt
t axis

5.4. Explanation :

در این سوال برای هر کدام از مقادیر m و h دو تابع تعریف کردیم که یکی از آنها مقدار ولتاژ غشا و پتانسیل استراحت را می گیرد و با توجه به روابط گفته شده در اسلاید های درس، آلفا و بتا را محاسبه می کند. تابع دیگر با فراخوانی تابع قبلی، آلفا و بتا را در یک ولتاژ مشخص می گیرد و مشتق m و h نسبت به t را در هر لحظه داده شده محاسبه می کند. برای اینکه خود m و h را در هر لحظه داشته باشیم از تابع نوشته شده در سوال 1 استفاده می کنیم، اما در آن تغییراتی ایجاد می کنیم تا مقدار ولتاژ را به عنوان ورودی بگیرد و به زیر تابع های خود بدهد.(زیرا مقادیری مانند آلفا و بتا به مقدار ولتاژ وابسته هستند)

در نتیجه می توانیم با دادن زمان، ولتاژ و مقدار اولیه m و h به تابع m_prime و h_prime مقدار m و h را در هر لحظه داشته باشیم.

همچنین نمودار تغییرات m و h را در مدت زمان 10 ثانیه و در ولتاژ منفی 20 میلی ولت به ازای دو مقدار اولیه صفر و یک، رسم کنیم.

Q6

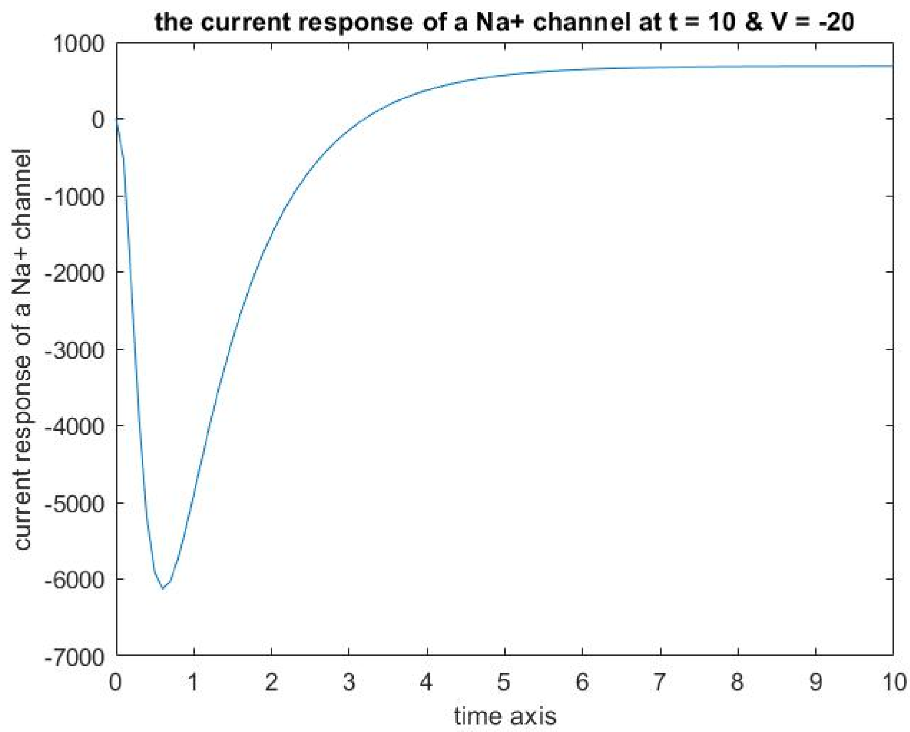
6.1. Link :

[..\6](..\6)

6.2. Code :

```
6.2.1 :

function Ina = Na_v(t , V)
% function arguments : t and V
% t   : a time interval to study the current response
% V   : a holding potential that causes the current response
% Ina : the current response of a Na+ channel

g_Na_avg = 120 ;
E_Na = 52.4 ;
% Attribute defult values to required parameters
% Source : BIOELECTRICITY:AQUANTITATIVE APPROACH _ Table 13.2

time = [0:0.1:t] ;
% Set the time array

m_o = 0 ;
h_o = 1 ;
% Set the initial condition

m = ode_euler_modified(@m_prime, time, m_o, V);
% Call ode_euler_modified function to calculate the dm

h = ode_euler_modified(@h_prime, time, h_o, V);
% Call ode_euler_modified function to calculate the dh

n3 = m.^3 ;
% Calculate n^4

% calculate the current response of a K+ channel :
Ina = g_Na_avg * ( n3 .* h ) * (V - E_Na);
```

6.3. Figure :



6.4. Explanation :

در این سوال یک تابع تعریف کردیم که با گرفتن m و h از توابع نوشته شده، جریان کانال سدیم را در ولتاژ و زمان داده شده محاسبه می کند.

در این تابع فرض کردیم که همه کانال ها در لحظه اول بسته می باشند. جریان بدست آمده با جریان گفته شده به عنوان early current مطابقت دارد و تغییر جریان در نتیجه ولتاژ کلمپ را به ما نشان می دهد.

همچنین نمودار تغییرات جریان کانال سدیم در حالتی که غشا فقط به +Na نفوذپذیر باشد را در مدت زمان 10 ثانیه و در ولتاژ منفی 20 میلی ولت رسم کنیم.

Q7

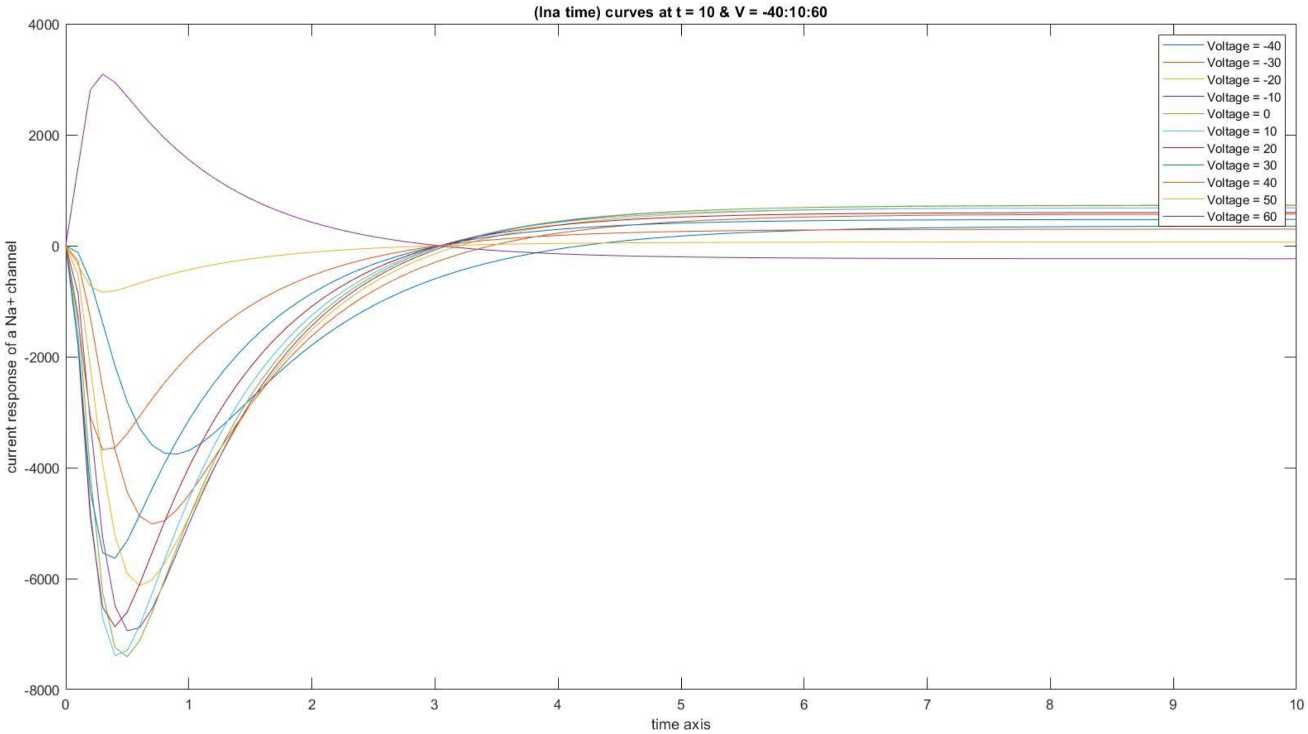
7.1. Link :

[..\7](..\7)

7.2. Code :

```
7.2.1 :

t = 10 ;
time=[0:0.1:t] ;
Lt = length(time) ;
% Set a constant time duration and its array and length

V = [-40:10:60] ;
Lv = length(V) ;
% Initialize voltage array and calculate its length

Ina = zeros(Lv,Lt) ;
% Initialize current matrix
% Row : number of given voltage values ( V = [-40:10:60] )
% Column : number of given time values ( t = [0:0.1:10] )

% A for loop to display Ina_time curves for all given voltage values :
for i = 1:Lv

    Ina(i,:) = Na_v( t , V(i)) ;
    % Call Na_v function to calculate the current response of a Na+
channel

    lgnd = ['Voltage = ',num2str(V(i))] ;
    % Set information to display legend

    plot(time,Ina(i,:),'DisplayName',lgnd), title('(Ina time) curves at
t = 10 & V = -40:10:60')
    xlabel('time axis'), ylabel('current response of a Na+ channel');
    % Display Ina_time curve

    % Check the end of the plots :
    if i ~= Lv
        hold on
    else
        hold off
        legend ;
        % Show legend
    end

end

savefig('Ina_time_diff_V.fig');
% Save the curves in a single figure in current folder
```

7.3. Figure :



7.4. Explanation :

در این سوال با توجه به تابعی که برای بدست آوردن جریان سدیمی در سوال قبل نوشته شد، نمودار تغییرات جریان کانال پتاسیم در حالتی که غشا فقط به +Na نفوذپذیر باشد را در مدت زمان 10 ثانیه و در ولتاژ های متفاوت رسم کردیم.

جریان های بدست آمده با جریان گفته شده به عنوان early current مطابقت دارد و تغییر جریان در نتیجه ولتاژ کلمپ های متفاوت را به ما نشان می دهد. همانطور که میبینیم با افزایش ولتاژ، پیک جریان کانال های +Na ابتدا کاهش یافته و سپس افزایش می یابد. این موضوع بدلیل تغییر نیروی driving force نسبت به ولتاژ های مختلف است.

Q8

8.1. Link :

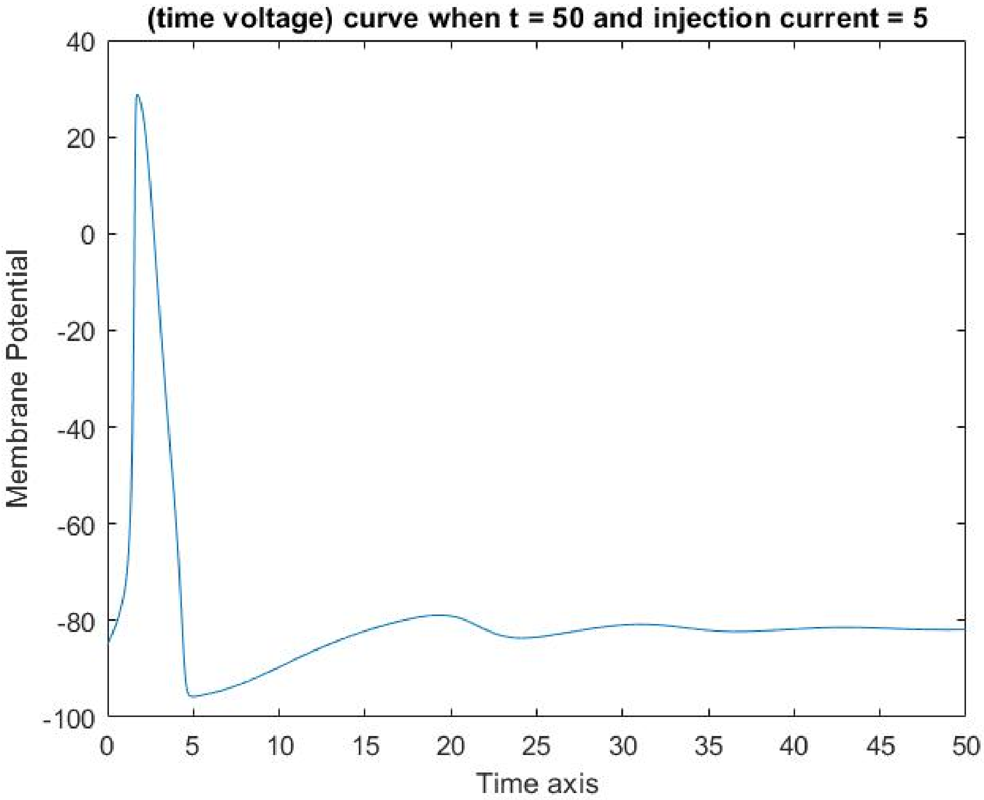
..\8

8.2. Code :

```
8.2.1 :

function V = hodgkin_huxley(t, I_inj)
% function arguments : t and I_inj
% t       : given time serie
% I_inj   : representing injected current
% V       : voltage at every point

Cm    = 1     ;
gL    = 0.3   ;
EL    = 10.6  ;
gK    = 36    ;
n     = 0.378 ;
EK    = -12   ;
gNa   = 120   ;
m     = 0.417 ;
h     = 0.477 ;
ENa   = 115   ;
% Specify required parameter values
% Source : BIOELECTRICITY:AQUANTITATIVE APPROACH - Table 13.2 & Table
13.3

time = [0.01:0.01:t] ;
% Set the time array

delta_t = time(2) - time(1) ;
% Calculate time step size

Lt = length(time) ;
% Calculate time array length

V = zeros(1,Lt) ;
m = zeros(1,Lt) ;
h = zeros(1,Lt) ;
n = zeros(1,Lt) ;
% Initialize required arraies for voltage , m , h & n

V(1)   = -85  ;
Vrest  = -85  ;
m(1)   = 0    ;
h(1)   = 1    ;
n(1)   = 0    ;
% Set  resting potential , m , h & n before current injection

% Define a for loop to calculate voltage at every time in given time
array:
for i=1:Lt-1

    iK  = gK*n(i)^4*(EK-(V(i)+85)) ;
    % Calculate current for K+ channels at every given voltage

    iNa = gNa*m(i)^3*h(i)*(ENa-(V(i)+85)) ;
    % Calculate current for Na+ channels at every given voltage
```

```
    V(i+1) = V(i) + delta_t*(iNa + iK + gL*(EL-(V(i)+85)) + I_inj);
    % Calculate voltage value with respect to previous voltage value

    [ alfa_n , beta_n ] = transition_rate_n(V(i) , Vrest) ;
    % Call transition_rate_n to calculate alfa & beta for n at given
voltage

    [ alfa_m , beta_m ] = transition_rate_m(V(i) , Vrest) ;
    % Call transition_rate_m to calculate alfa & beta for m at given
voltage

    [ alfa_h , beta_h ] = transition_rate_h(V(i) , Vrest) ;
    % Call transition_rate_h to calculate alfa & beta for h at given
voltage

    m(i+1) = m(i) + delta_t*(alfa_m*(1-m(i)) - beta_m*m(i));
    % Calculate m value with respect to previous m value at given
voltage

    h(i+1) = h(i) + delta_t*(alfa_h*(1-h(i)) - beta_h*h(i));
    % Calculate h value with respect to previous h value at given
voltage

    n(i+1) = n(i) + delta_t*(alfa_n*(1-n(i)) - beta_n*n(i));
    % Calculate n value with respect to previous n value at given
voltage

end
```

8.3. Figure :



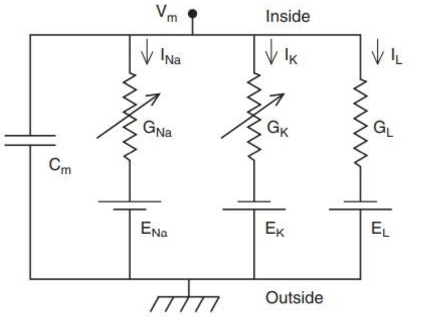
8.4. Explanation :

در این سوال یک تابع تعریف کردیم که با توجه به جریان تزریق شده، ولتاژ غشا را در مدت زمان داده شده محاسبه می کند.

برای این کار ابتدا جریان ها کانال های سدیم و پتاسیم را در هر لحظه حساب کردیم و با توجه به مدل و رابطه زیر، تغییرات ولتاژ را به دست آوردیم. اکنون برای محاسبه ولتاژ کافیست ولتاژ اولیه را با تغییرات ولتاژ جمع کنیم. این کار را به ازای تمام لحظات خواسته شده در یک حلقه for محاسبه می کنیم.

همچنین نمودار ولتاژ به ازای جریان تزریق شده 5 آمپر را رسم کردیم که نشان می دهد با تزریق این مقدار جریان، پتانسیل عمل تولید می شود.

$$C_m \frac{dV_M}{dt} = -\bar{g}_{Na} mh(V_M - E_{Na}) - \bar{g}_K n(V_M - E_K) - g_L(V_M - E_L) + I_{inj}$$
$$\frac{dn}{dt} = k_{1n} - (k_{1n} + k_{-1n})n$$
$$\frac{dm}{dt} = k_{1m} - (k_{1m} + k_{-1m})m$$
$$\frac{dh}{dt} = k_{1h} - (k_{1h} + k_{-1h})h$$



Q9

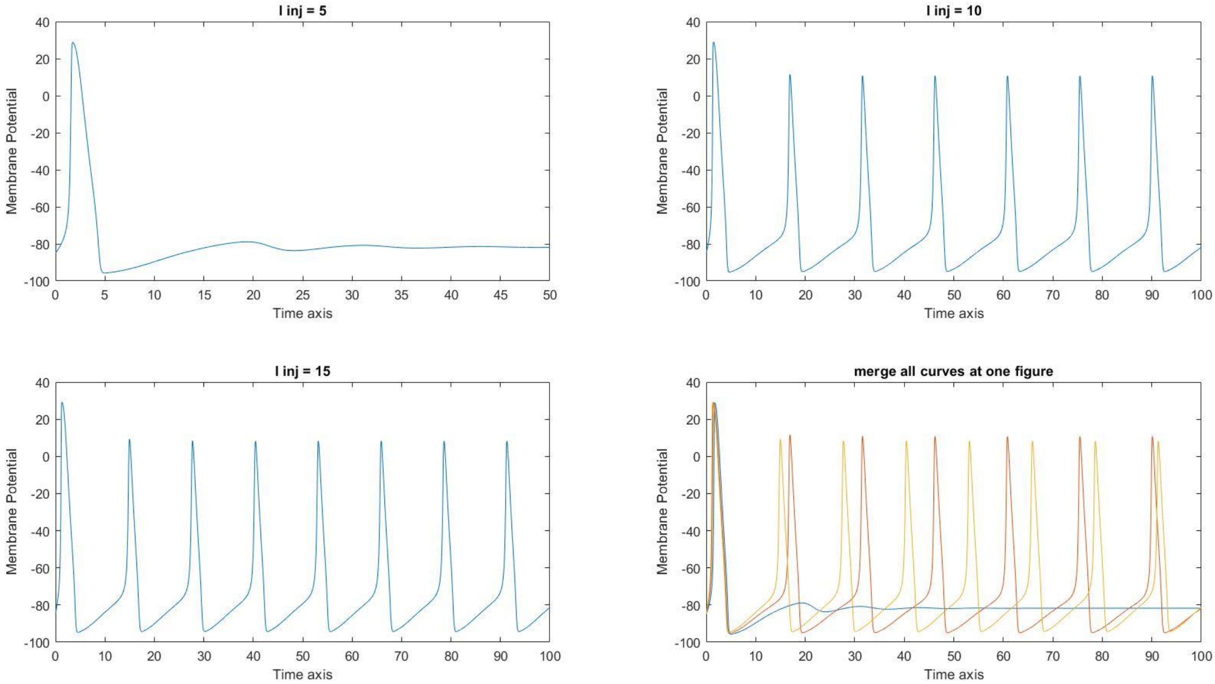
9.1. Link :

..\9

9.2. Code :

```
9.2.1 :

clear all; close all; clc;

v1 = hodgkin_huxley(50, 5);
% Call hodgkin_huxley function to calculate voltage when I_inj = 5

v1plus = hodgkin_huxley(100, 5);
% Call hodgkin_huxley function to calculate voltage when I_inj = 5

v2 = hodgkin_huxley(100, 10);
% Call hodgkin_huxley function to calculate voltage when I_inj = 10

v3 = hodgkin_huxley(100, 15);
% Call hodgkin_huxley function to calculate voltage when I_inj = 15

time1 = [0.01:0.01:50];
% Set the time array when t = 50

time2 = [0.01:0.01:100];
% Set the time array when t = 100

figure ;
subplot(2,2,1),plot(time1,v1),title('I inj = 5'),xlabel('Time
axis'),ylabel('Membrane Potential');
subplot(2,2,2),plot(time2,v2),title('I inj = 10'),xlabel('Time
axis'),ylabel('Membrane Potential');
subplot(2,2,3),plot(time2,v3),title('I inj = 15'),xlabel('Time
axis'),ylabel('Membrane Potential');
subplot(2,2,4),plot(time2,v1plus,time2,v2,time2,v3)
title('merge all curves at one figure'),xlabel('Time
axis'),ylabel('Membrane Potential');
% Display time_voltage curve with axis details

savefig('time_voltage.fig');
% Save time_voltage curve in current folder
```

9.3. Figure :



9.4. Explanation :

در این سوال با توجه به تابع به دست آمده در سوال قبل، جریان های 5،10 و 15 آمپر را به سلول تزریق کردیم و تشکیل پتانسیل عمل در هر کدام را به نشان دادیم.

همچنین هر سه نمودار را در یک شکل رسم کردیم که تغییرات فرکانس در آن مشخص شود.

Q10

10.1. Link :

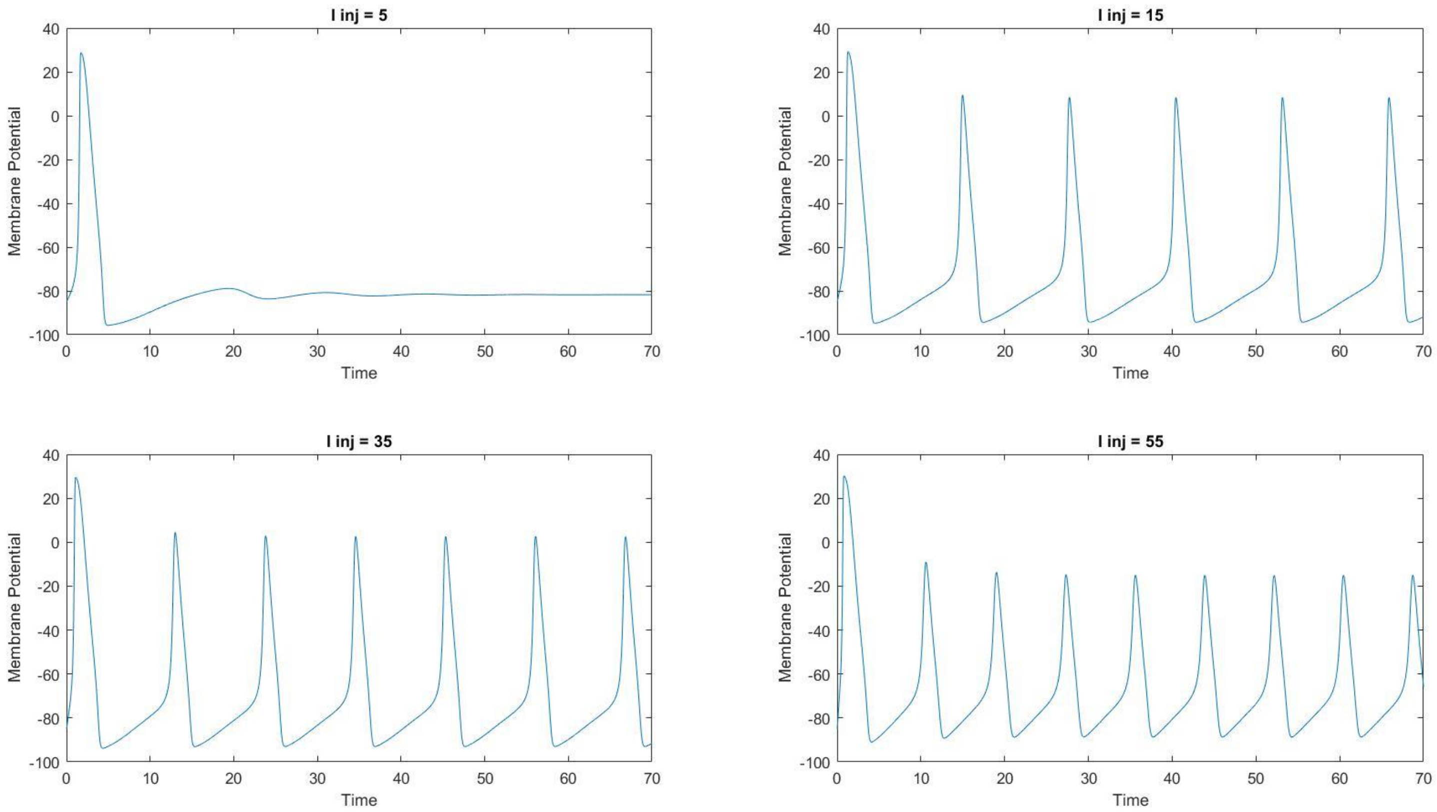
..\10

10.2. Code :

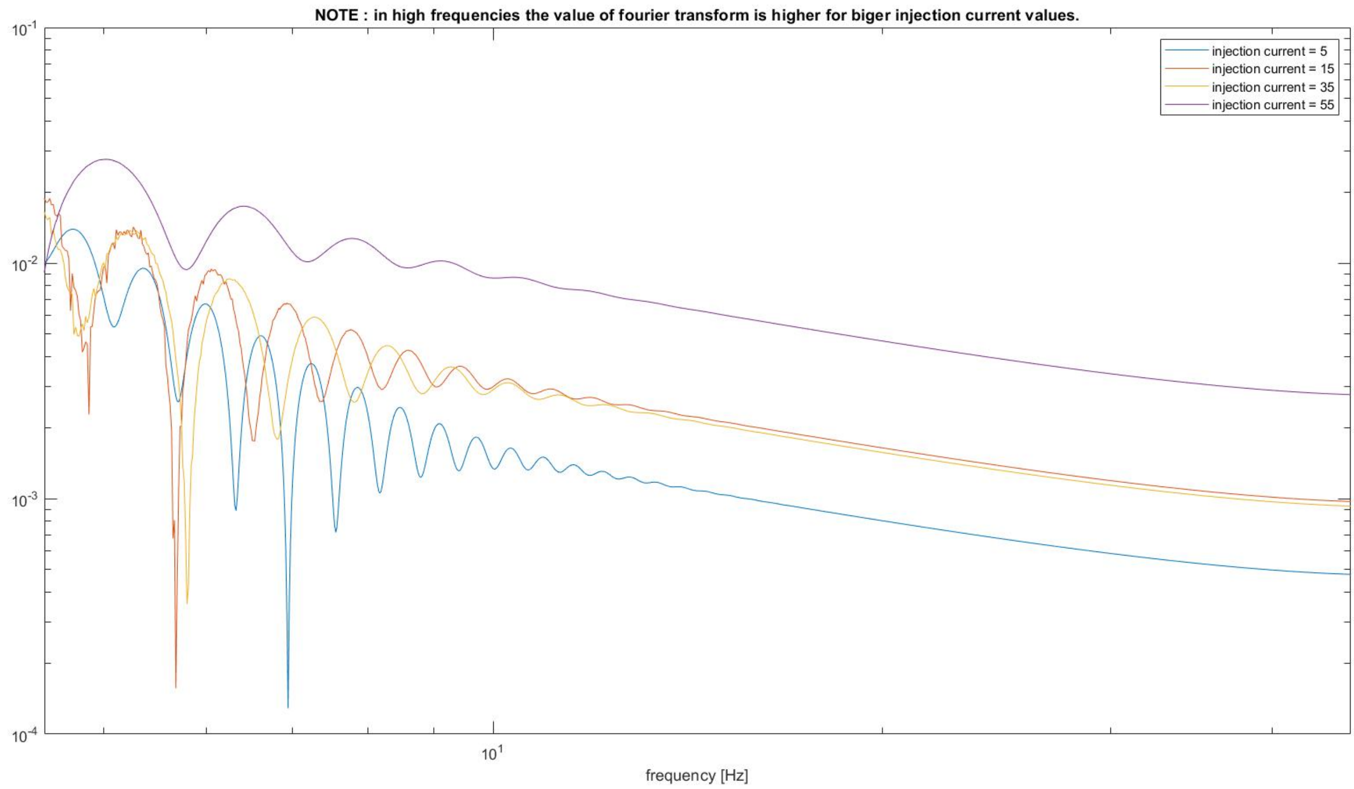
```
10.2.1 :

clear all; close all; clc;

v1 = hodgkin_huxley(70, 5);
% Call hodgkin_huxley function to calculate voltage when I_inj = 5

v2 = hodgkin_huxley(70, 15);
% Call hodgkin_huxley function to calculate voltage when I_inj = 15

v3 = hodgkin_huxley(70, 25);
% Call hodgkin_huxley function to calculate voltage when I_inj = 35

v4 = hodgkin_huxley(70, 55);
% Call hodgkin_huxley function to calculate voltage when I_inj = 55

time = [0.01:0.01:70];
% Set the time array when t = 70
sampling_frequency = 1 /0.01 ;
% Calculate sampling frequency

ydft1 = fft(v1);
ydft1 = 2*ydft1(1:ceil((length(v1)+1)/2));
freq1 = 0:sampling_frequency/length(v1):sampling_frequency/2 ;
ydft1 = ydft1/(2*length(freq1));
% Calculate fast fourier transform for voltage when I_inj = 5
% and determine frequency changes in a limited range in a logarythmic
space

ydft2 = fft(v2);
ydft2 = 2*ydft2(1:ceil((length(v2)+1)/2));
freq2 = 0:sampling_frequency/length(v2):sampling_frequency/2 ;
ydft2 = ydft2/(2*length(freq2));
% Calculate fast fourier transform for voltage when I_inj = 15
% and determine frequency changes in a limited range in a logarythmic
space

ydft3 = fft(v3);
ydft3 = 2*ydft3(1:ceil((length(v3)+1)/2));
freq3 = 0:sampling_frequency/length(v3):sampling_frequency/2 ;
ydft3 = ydft3/(2*length(freq3));
% Calculate fast fourier transform for voltage when I_inj = 35
% and determine frequency changes in a limited range in a logarythmic
space

ydft4 = fft(v4);
ydft4 = 2*ydft4(1:ceil((length(v4)+1)/2));
freq4 = 0:sampling_frequency/length(v4):sampling_frequency/2 ;
ydft4 = ydft4/(2*length(freq4));
% Calculate fast fourier transform for voltage when I_inj = 55
% and determine frequency changes in a limited range in a logarythmic
space
```

```
figure ;
subplot(2,2,1),plot(time,v1),title('I inj =
5'),xlabel('Time'),ylabel('Membrane Potential');
subplot(2,2,2),plot(time,v2),title('I inj =
15'),xlabel('Time'),ylabel('Membrane Potential');
subplot(2,2,3),plot(time,v3),title('I inj =
35'),xlabel('Time'),ylabel('Membrane Potential');
subplot(2,2,4),plot(time,v4),title('I inj =
55'),xlabel('Time'),ylabel('Membrane Potential');
% Display time_voltage curve with axis details at different injection
current

savefig('time_voltage.fig');
% Save time_voltage curve in current folder

figure ;
loglog(freq1,abs(ydft1),'DisplayName','injection current = 5'),xlim([4.5
46]), xlabel('frequency [Hz]');
hold on
loglog(freq2,abs(ydft2),'DisplayName','injection current =
15'),xlim([4.5 46]), xlabel('frequency [Hz]');
hold on
loglog(freq3,abs(ydft3),'DisplayName','injection current =
35'),xlim([4.5 46]), xlabel('frequency [Hz]');
hold on
loglog(freq4,abs(ydft4),'DisplayName','injection current =
55'),xlim([4.5 46]), xlabel('frequency [Hz]');
title('NOTE : in high frequencies the value of fourier transform is
higher for biger injection current values.');
hold off
% Demonstrate increasing frequency with increasing injected current

legend ;
% Show legend

savefig('frequency.fig');
% Save frequency curve in current folder
```

10.3. Figure :



I inj = 5
I inj = 15
I inj = 35
I inj = 55
Membrane Potential
Time



NOTE : in high frequencies the value of fourier transform is higher for biger injection current values.
injection current = 5
injection current = 15
injection current = 35
injection current = 55
frequency [Hz]

10.4. Explanation :

در این سوال می خواهیم اثبات کنیم که با افزایش جریان تزریق شده، فرکانس پتانسل عمل نیز افزایش می یابد. برای این کار جریان های 5،15،35 و 55 آمپر را به سلول تزریق کردیم و تشکیل پتانسیل عمل در هر کدام را به نشان دادیم.

همانطور که مشاهده می شود با افزایش جریان تزریق شده و در مدت زمان ثابت، تعداد پتانسل های عمل بیشتری تولید می شود.

برای نمایش بهتر ارتباط جریان با فرکانس، از نمودار های بدست آمده تبدیل فوریه گرفتیم و در فرکانس های مورد نظر بازه ای را تعیین کردیم که تبدیل فوریه را با توجه به نرخ نمونه برداری نشان دهد.

با رسم این نمودار میبینیم که با افزایش فرکانس، نمودار مربوط به جریان 55 آمپر مقدار بیشتری دارد و هرچه جریان بیشتر باشد فرکانس های بیشتری در نمودار تغییرات ولتاژ آن حضور دارند و در نتیجه پتانسل عمل با فرکانس بیشتری تولید می شود.

Q11

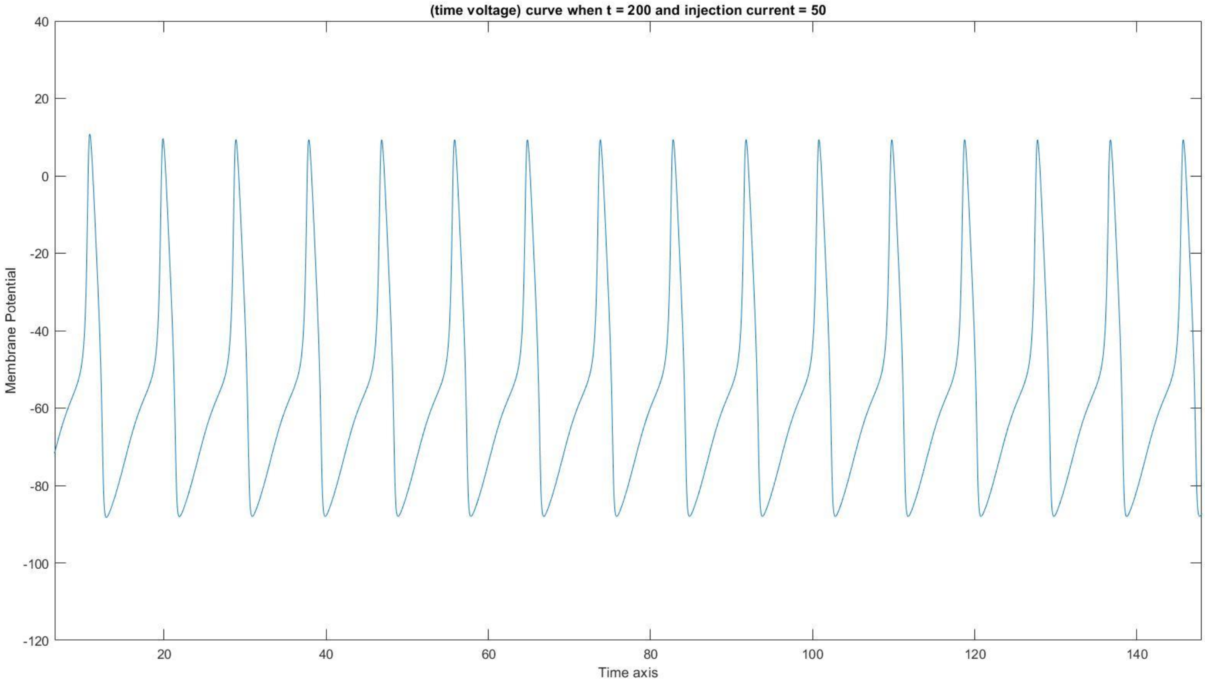
11.1. Link :

[..\11](..\11)

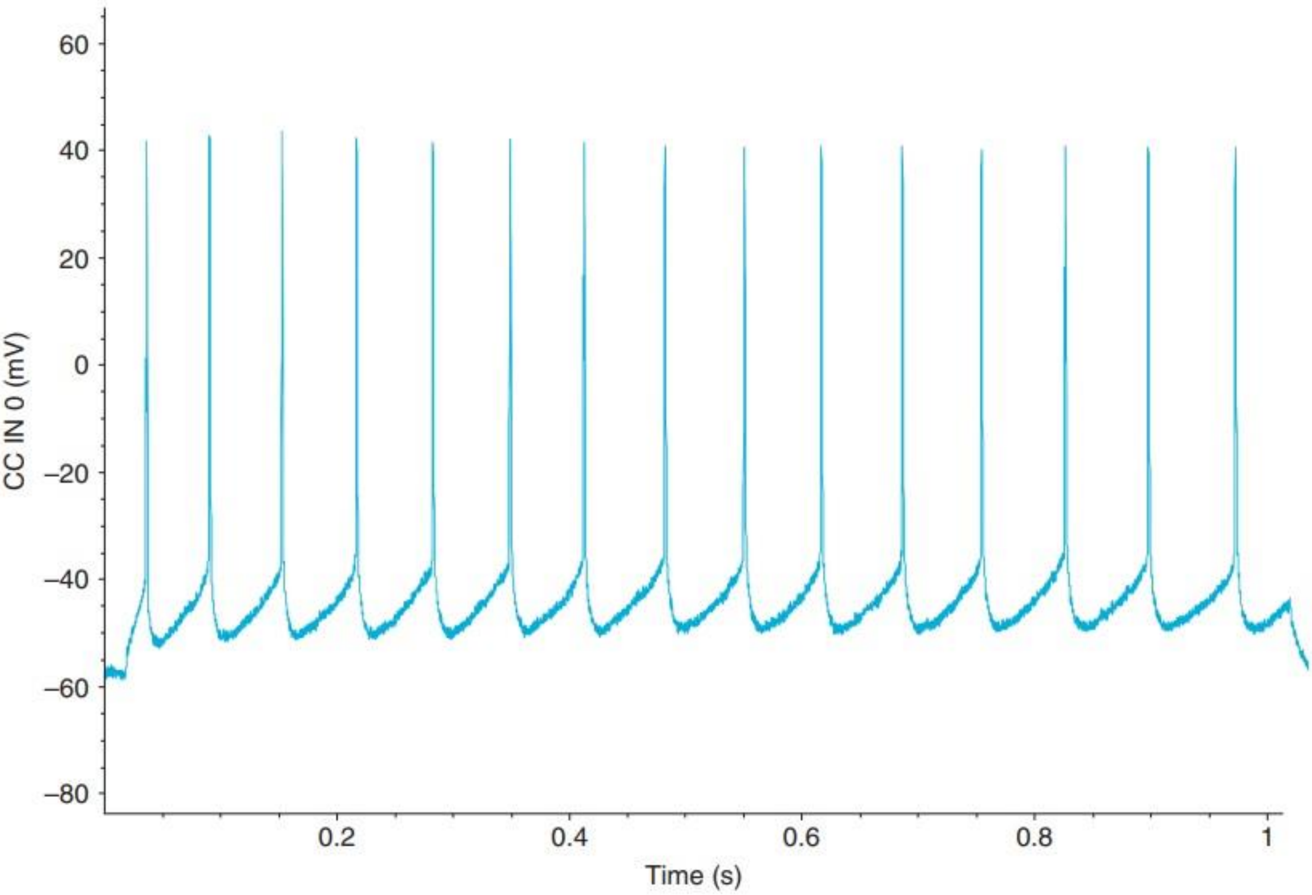
11.2. Code :

```
11.2.1 :

clear all; close all; clc;

img = imread('Figure 20.1.jpg');
% Insert Figure 20.1 in 'img' variable

v = hodgkin_huxley(200, 50);
% Call hodgkin_huxley function to calculate voltage when t = 200 and
I_inj = 50

time = [0.01:0.01:200];
% Set the time array when t = 200

figure ;
subplot(2,2,1),plot(time,v),axis([6.5 148 -120 40])
xlabel('Time axis'),ylabel('Membrane Potential');
title('(time voltage) curve when t = 200 and injection current = 50');
% Display time_voltage curve with axis details

subplot(2,2,4),imshow(img)
% Display Figure 20.1

savefig('Comparison figure.fig');
% Save the Comparison figure in current folder
```

11.3. Figure :



(time voltage) curve when t = 200 and injection current = 50
Membrane Potential
Time axis
40
20
0
-20
-40
-60
-80
-100
-120
20
40
60
80
100
120
140



CC IN 0 (mV)
Time (s)
60
40
20
0
−20
−40
−60
−80
0.2
0.4
0.6
0.8
1

11.4. Explanation :

در این سوال می بینیم که مدل ما به خوبی کار می کند و پتانسل عمل های تولید شده توسط مدل هاچلین و هاکسلی با تقریب خوبی مشابه پتانسل عمل های تولید شده در واقعیت هستند.

این موضوع نشان می دهد که روابط هاچلین و هاکسلی از دقت بالایی برخوردار هستند و عملکرد بیوالکتریکی سلول ها را به خوبی مدلسازی می کنند.

اما تفاوتی که بین این دو وجود دارد این است که در واقیعت موارد دیگری نیز در این امر دخیل هستند (مانند پمپ ها) که در مدل ما فعالیت آنها در مظر گرفته نشده است. همچنین به دلیل عدم بررسی تغییرات کلسیم که باعث متوقف شدن قطار پتانسل عمل می شود، در این مدل تا زمانی که تزریق جریان داریم پتانسل عمل اتفاق می افتد که در واقیت این گونه نیست.

Q12

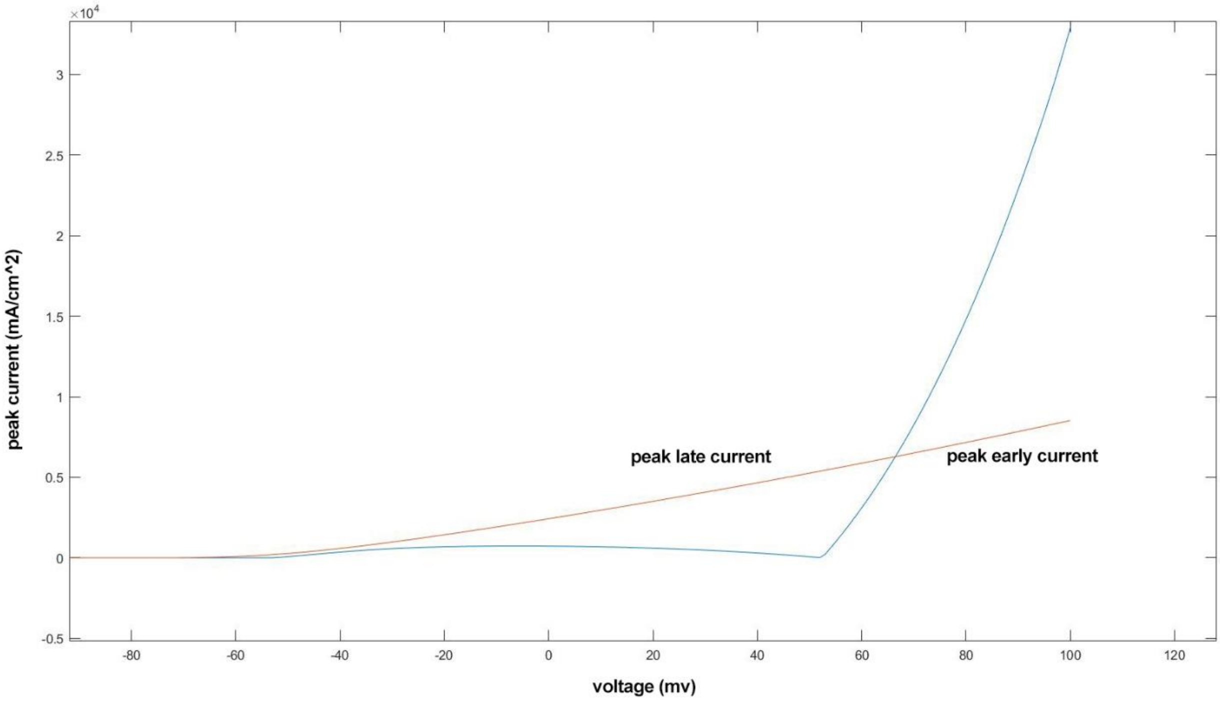
12.1. Link :

[12]

12.2. Code :

```
12.2.1 :

clear all ; close all ; clc ;

Namax = zeros(1,201);
Kmax = zeros(1,201);
% Initialize arraies for peaks

i = 1 ;
% Initialize counter

% Define a for loop to calculate peak changes for late and
early current
for v = -100:1:100
    iNa = Na_v(10,v);
    % Call Na_v function to calculate early current for
every given
    % voltages

    Namax(i) = max(iNa) ;
    % Calculate peak for early current

    iK = K_v(10,v);
    % Call K_v function to calculate late current for every
given voltages

    Kmax(i) = max(iK) ;
    % Calculate peak for late current

    i = i+1 ;
    % Add 1 to counter

end

v = -100:1:100 ;
% Set x axis

% Display peak changes for late and early current with
respect to voltage changes
plot(v,Namax,v,Kmax)
```

12.3. Figure :



12.4. Explanation :

در این سوال با توجه به توابعی که برای جریان بدست آوردیم، تغییرات پیک آنها را نسبت به تغییرات ولتاژ محاسبه کردیم.