AMATH 342 HW 3

Due Monday, Feb 26

Chongyi Xu, Univeristy of Washington

I. Filtering of inputs

- Choose $t_{now} = 30ms$, $V(t_{now}) = 10mV$, and $R * C = 10ms$. Find two input currents $I_1(t)$ and $I_2(t)$.

  First, set the values we have chosen.

```
deltat=0.1 ; %timestep
Tmax=50;

%circuit parameters
R=10;
C=1;
```

And plug these values into the given file `euler_illustrateRC.m` .

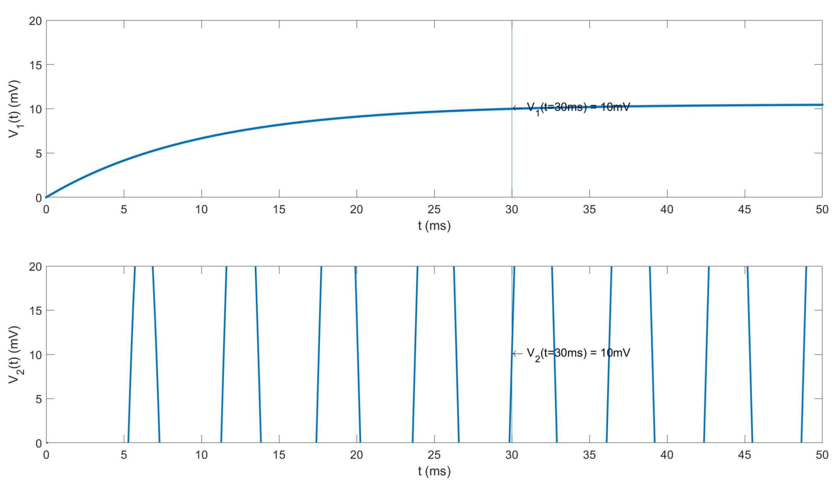
```
deltat=0.01 ; %timestep
Tmax=50;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
V1=zeros(1,length(tlist));
V2=zeros(1,length(tlist));

%initialize
V0=0;
V1(1)=V0;
V2(1)=V0;

%circuit parameters
R=10;
C=1;
```

And then we would like to find the input current that could give $V(t_{now}) = 10mV$ when $t_{now} = 30ms$. With testing for a several number of different input currents, the $I_1(t)$ and $I_2(t)$ I got are $I_1(t) = 1.05157$ and $I_2(t) = -64.681 * sin(t)$, which are obviously very different from each other. However, they results in same $V(t_{now}) = 10mV$ at $t_{now} = 30ms$.

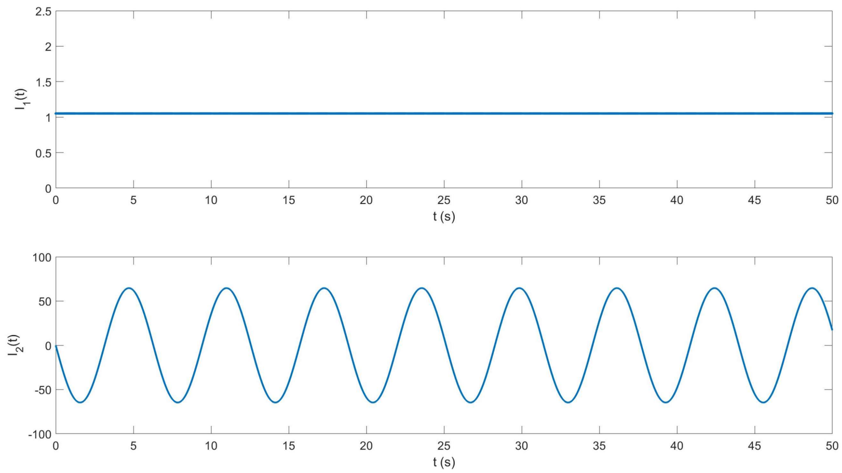
```
%define input currents
I1=1.05157 * ones(1,length(tlist));
I2=-64.681* sin(tlist);

%caculate for V1 and V2 for every t
for n=1:length(tlist)-1
    t=tlist(n);
    V1(n+1)=V1(n) + (-V1(n)/(R*C) + I1(n)/C )*deltat;
    V2(n+1)=V2(n) + (-V2(n)/(R*C) + I2(n)/C )*deltat;
end
```

Then we would like to see the difference between I_1 and I_2. So I plotted V_1 and V_2 in the time interval $[0, 500ms]$.

```
figure(1)
subplot(211)
plot(tlist,V1,'.-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (ms)'); ylabel('V_1(t) (mV)');
axis([0 50 0 20])
line([30 30], [0 30]);
txt1 = '\leftarrow V_1(t=30ms) = 10mV';
text(30, V1(tlist==30), txt1);

subplot(212)
plot(tlist,V2,'-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (ms)'); ylabel('V_2(t) (mV)');
axis([0 50 0 20])
line([30 30], [0 30]);
txt2 = '\leftarrow V_2(t=30ms) = 10mV';
text(30, V1(tlist==30), txt2);
```



It can be seen that they have completely different plots of relavant voltage during this time interval. And the current traces are much more different.



The second current curve is sine curve but the first curve is just a constant.

- What is it about the explicit solution for $V(t)$?

From class, we have learned that the explicit form of $V(t)$ is

$$V(t) = V_0 e^{-t/RC} + \int_0^t k(t')I_A(t-t')dt'$$

$$k(t') = \frac{e^{-t'/RC}}{c}$$

Therefore, $k(t')$ indicates that it is possible to have more than one current trace. And due to the principal of superposition, different current inputs will have same $V(t_{now})$.

II. Summation of simultaneous impulses

- Consider an incoming current impulse. What is the peak voltage achieved over time in response to this impulse?

  I chose $\overline{I} = 0.5\mu A$, $\Delta t = 0.1ms$.

```
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));

Ialist=zeros(1,length(tlist));
width=30; t1=30; t2=t1+width;
Ialist(tlist>=t1&tlist<t2) = 0.5;


%initialize
```

```
V0=0;
Vlist(1)=V0;

%circuit parameters
R=10;
C=1;
```

Then I tried to find the voltage $V(t)$

```
for n=1:length(tlist)-1
    t=tlist(n);
    Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) +Ialist(n)/C)*deltat;
end
```

And the peak voltage I found is

```
>> max(Vlist)

ans =

    4.7548
```

- If the threshold for spike generation is 10 mV, what fraction of the way to threshold does this impulse drive the voltage response?

```
>> 1 - max(Vlist) / 10

ans =

    0.5245
```

So, $f \approx 0.5245.$

- Next consider the case in which N such impulses arrive simultaneously. What is the lowest value N that will drive the voltage over threshold?

```
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));
Ialist=zeros(1,length(tlist));
width=30; t1=30; t2=t1+width;
Ialist(tlist>=t1&tlist<t2) = 0.5;
```

```
%initialize
V0=0;
Vlist(1)=V0;
threshold = 10;
N = 1;

%circuit parameters
R=10;
C=1;

while max(Vlist) < threshold
    for n=1:length(tlist)-1
        t=tlist(n);
        Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) +Ialist(n)/C)*deltat;
    end
    N = N + 0.00001;
    Ialist(tlist>=t1&tlist<t2) = N * 0.5;
end
figure
plot(tlist,Vlist,'.-','LineWidth',2); hold on; grid on;
xlabel('t'); ylabel('V(t)');
```

The lowest value N I found is

```
>> N

N =

    2.1032
```

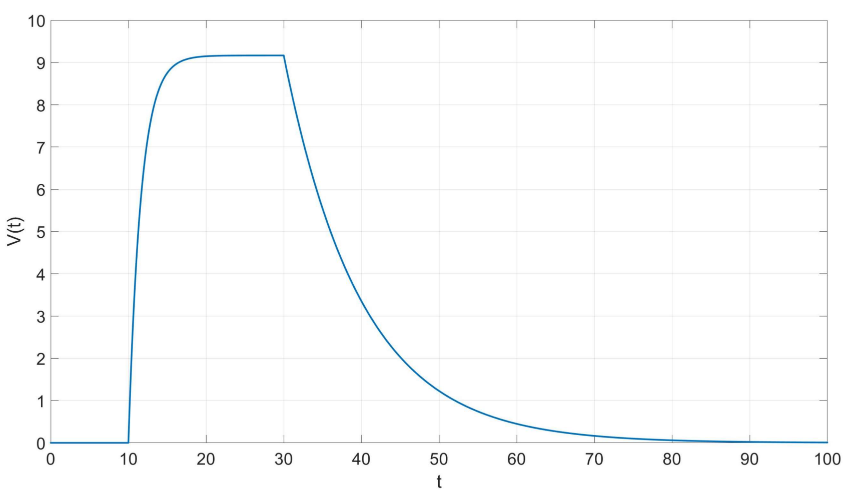
This implies that $f * N = 1.0$, which is the same result as I got from solving the explicit solution of integral form. And for RC circuit, impulses summate linearly.

- Now study for a conductance-based input model.

First, I tried $g = 0.5$ as I tried with I in the previous part.

```
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));

%initialize
V0=0;
Vlist(1)=V0;
```

```
%define input conductance
gapplist=zeros(1,length(tlist));
t1 = 10; t2 = t1 + 20;
gapplist(tlist>=t1&tlist<t2) = 0.5;


%circuit parameters
R=10;
C=1;
E=11;

for n=1:length(tlist)-1
    t=tlist(n);
    Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) + gapplist(n)*(E-Vlist(n)) )*deltat;
end

figure
plot(tlist,Vlist,'.-','LineWidth',2); hold on
xlabel('t'); ylabel('V(t)');
```

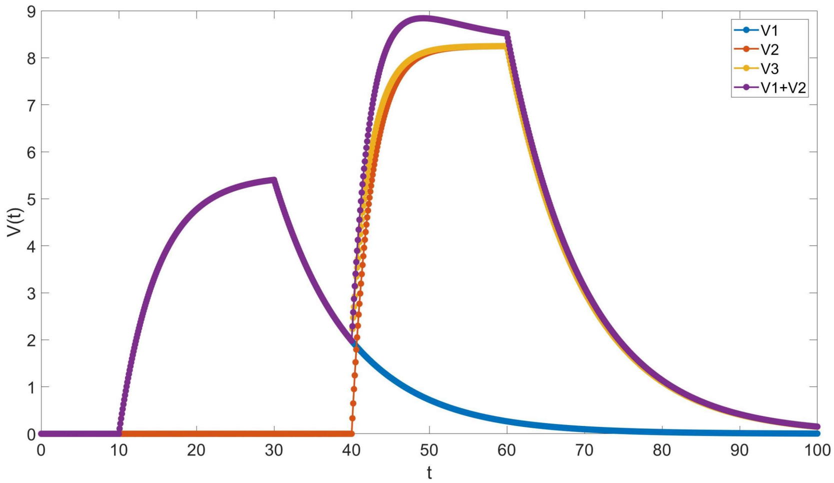


And then I tried $\overline{g} = 0.1, 0.3, 0.4$ to test the linearity.

```
deltat=0.1 ; %timestep
Tmax=100;
width = 20;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
V1=zeros(1,length(tlist));
V2=zeros(1,length(tlist));
V3=zeros(1,length(tlist));
V4=zeros(1,length(tlist));

%initialize
V0=0;
```

```
V1(1)=V0;
V2(1)=V0;
V3(1)=V0;
V4(1)=V0;

%define input conductance
g1=zeros(1,length(tlist));
t1 = 10; t2 = t1 + width;
g1(tlist>=t1&tlist<t2) = 0.1;

g2=zeros(1,length(tlist));
t1 = 40; t2 = t1 + width;
g2(tlist>=t1&tlist<t2) = 0.3;

g3=g1+g2;


%circuit parameters
R=10;
C=1;
E=11;
for n=1:length(tlist)-1
    t=tlist(n);
    V1(n+1)=V1(n) + ( -V1(n)/(R*C) + g1(n)*(E-V1(n)) )*deltat;
    V2(n+1)=V2(n) + ( -V2(n)/(R*C) + g2(n)*(E-V2(n)) )*deltat;
    V3(n+1)=V3(n) + ( -V3(n)/(R*C) + g3(n)*(E-V3(n)) )*deltat;
end

figure(3);
plot(tlist,V1,'.-','LineWidth',2, 'MarkerSize', 26); hold on
xlabel('t'); ylabel('V(t)');
plot(tlist,V2,'.-','LineWidth',2, 'MarkerSize', 26);
plot(tlist,V3,'.-','LineWidth',2, 'MarkerSize', 26);
plot(tlist,V1+V2,'.-','LineWidth',2, 'MarkerSize', 26);
legend('V1','V2','V3','V1+V2')
```

And the plot I got is



It can be seen that the pulses summate sublinearly $f(I1+I2) < f(I1)+f(I2)$. From class notes, the reason why is that $g(t)$ is a nonlinear function that $g(t1+t2) = g(t1)+g(t2)$ but $e^{g(t1+t2)} \neq e^{g(t1)} + e^{g(t2)}$.

III HH model

- Use appropriate codes provided in HH directory to plot the $firing\ rate - current$ tuning curve for HH model.

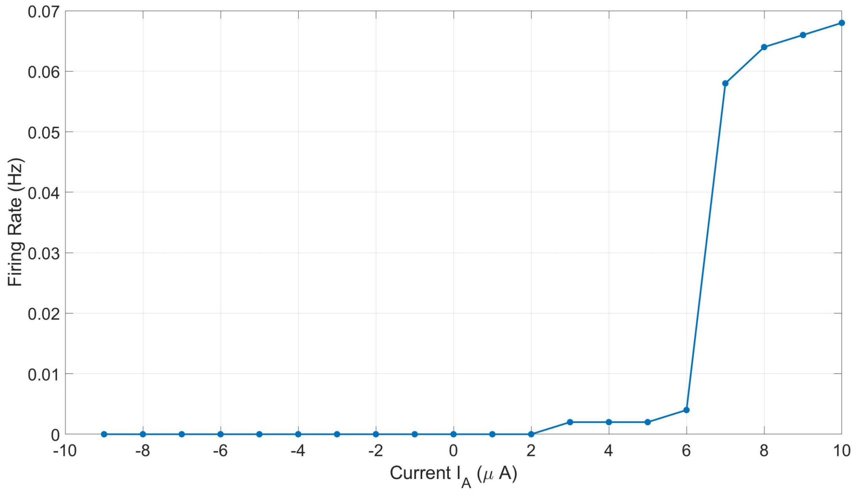
I used the code `HH.m` and modified it to count for spikes in order to calculate firing rate over time interval (500 ms). I assumed that there will be a spike at the peak value that is greater than my threshold 0.

```
function firingrate = hhmodel(I)
    vna=50;  %set the constants
    vk=-77;
    vl=-54.4;
    gna=120;
    gk=36;
    gl=.3;
    c=1;

    v_init=-65;  %the initial conditions
    m_init=.052;
    h_init=.596;
    n_init=.317;

    npoints=500000;  %number of timesteps to integrate
    dt=0.001;        %timestep


    m=zeros(npoints,1); %initialize everything to zero
    n=zeros(npoints,1);
```

```
    h=zeros(npoints,1);
    v=zeros(npoints,1);
    time=zeros(npoints,1);

    m(1)=m_init; %set the initial conditions to be the first entry in the vectors
    n(1)=n_init;
    h(1)=h_init;
    v(1)=v_init;
    time(1)=0.0;
    numpeak = 0;
    thresh = 0;

    tic
    for step=1:npoints-1
        v(step+1)=v(step)+((I - gna*h(step)*(v(step)-vna)*m(step)^3 ...
                   -gk*(v(step)-vk)*n(step)^4-gl*(v(step)-vl))/c)*dt;
        m(step+1)=m(step)+ (alpha_m(v(step))*(1-m(step))-beta_m(v(step))*m(step))*dt;
        h(step+1)=h(step)+ (alpha_h(v(step))*(1-h(step))-beta_h(v(step))*h(step))*dt;
        n(step+1)=n(step)+ (alpha_n(v(step))*(1-n(step))-beta_n(v(step))*n(step))*dt;
        time(step+1)=time(step)+dt;

        % spike detection: decreasing now and increasing before
         if ((step>1) && (v(step+1)<v(step)) && (v(step)>v(step-1))) && (v(step) > thresh)
            numpeak = numpeak + 1;
            peaktime(numpeak) = time(step+1);
         end
    end
    toc

    firingrate = numpeak / (npoints * dt);
end
```

Then I tried I over the interval $[-10, 10]$ to get a brief result.

```
rate = zeros(20, 1);
I = zeros(20, 1);
for k=1:20
    I(k) = -10 + k;
    rate(k) = hhmodel(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
xlabel('Current I_A (\mu A)'); ylabel('Firing Rate (Hz)');
```



And from the plot, we could see that the firing rate changes from < 0.01 to ≈ 0.06 at $\overline{I} = 7\mu A$.

- Now repeat but with a sinusoidal background current of frequency ωkHz.

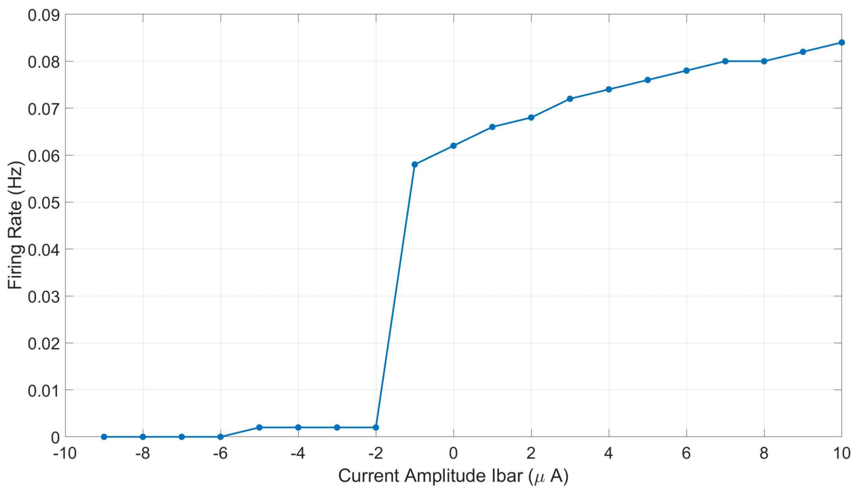
I modified the previous function code to change $I_A(t)$ to a sinusoidal current. I chose my $\epsilon = 2$ and $\omega = 4$. So $I_A(t) = \overline{I} + 2sin(8\pi t)$

```
epsilon = 2;
omega = 4;


tic
for step=1:npoints-1
    I = I + epsilon * sin(2*pi*time(step)*omega);
    v(step+1)=v(step)+((I - gna*h(step)*(v(step)-vna)*m(step)^3 ...
               -gk*(v(step)-vk)*n(step)^4-gl*(v(step)-vl))/c)*dt;
    m(step+1)=m(step)+ (alpha_m(v(step))*(1-m(step))-beta_m(v(step))*m(step))*dt;
    h(step+1)=h(step)+ (alpha_h(v(step))*(1-h(step))-beta_h(v(step))*h(step))*dt;
    n(step+1)=n(step)+ (alpha_n(v(step))*(1-n(step))-beta_n(v(step))*n(step))*dt;
    time(step+1)=time(step)+dt;

    % spike detection: decreasing now and increasing before
     if ((step>1) && (v(step+1)<v(step)) && (v(step)>v(step-1))) && (v(step) > thresh)
        numpeak = numpeak + 1;
        peaktime(numpeak) = time(step+1);
     end
end
toc
```

And then similarly, I plot the firing rate vs current amplitude curve.

```
rate = zeros(20, 1);
I = zeros(20, 1);
for k=1:20
```

```
    I(k) = -10 + k;
    rate(k) = hhmodel(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
symbol = 'I';
xlabel('Current Amplitude Ibar (\mu A)'); ylabel('Firing Rate (Hz)');
```



It can be seen that the "responsive" current dropped to $-1\ \mu A$. The reason is that the peak value for $I_A(t) = -1 + 2sin(8\pi t)$ is greater or equal to the previous "responsive" current.

- Finally, add noise to the applied current.

The noise I added is a uniformly distributed random number between $[-0.05, 0.05]\ \mu A$

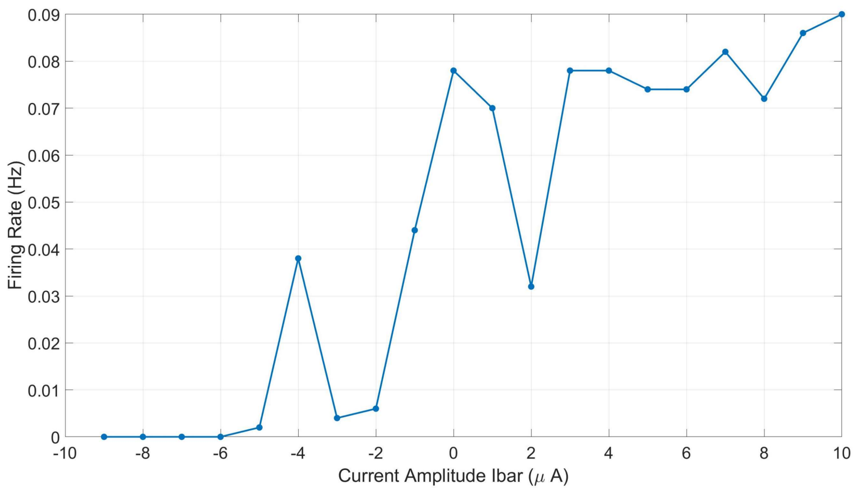
```
function [firingrate, fano] = hhmodel_with_noise(I)
    vna=50;  %set the constants
    vk=-77;
    vl=-54.4;
    gna=120;
    gk=36;
    gl=.3;
    c=1;

    v_init=-65;  %the initial conditions
    m_init=.052;
    h_init=.596;
    n_init=.317;

    npoints=50000;  %number of timesteps to integrate
    dt=0.01;        %timestep
```

```
    m=zeros(npoints,1); %initialize everything to zero
    n=zeros(npoints,1);
    h=zeros(npoints,1);
    v=zeros(npoints,1);
    time=zeros(npoints,1);
    noise = zeros(npoints, 1);

    m(1)=m_init; %set the initial conditions to be the first entry in the vectors
    n(1)=n_init;
    h(1)=h_init;
    v(1)=v_init;
    time(1)=0.0;
    numpeak = 0;
    thresh = 0;
    epsilon = 2;
    omega = 4;

    tic
    for step=1:npoints-1
        % noise
        noise(step) = -0.05 * rand() + 0.05 * rand();
        % current with noise
        I = I + epsilon * sin(2*pi*time(step)*omega);
        v(step+1)=v(step)+((I - gna*h(step)*(v(step)-vna)*m(step)^3 ...
                   -gk*(v(step)-vk)*n(step)^4-gl*(v(step)-vl))/c)*dt;
        m(step+1)=m(step)+ (alpha_m(v(step))*(1-m(step))-beta_m(v(step))*m(step))*dt;
        h(step+1)=h(step)+ (alpha_h(v(step))*(1-h(step))-beta_h(v(step))*h(step))*dt;
        n(step+1)=n(step)+ (alpha_n(v(step))*(1-n(step))-beta_n(v(step))*n(step))*dt;
        time(step+1)=time(step)+dt;

        % spike detection: decreasing now and increasing before
         if ((step>1) && (v(step+1)<v(step)) && (v(step)>v(step-1))) && (v(step) > thresh)
            numpeak = numpeak + 1;
            peaktime(numpeak) = time(step+1);
         end
    end
    toc

    firingrate = numpeak / (npoints * dt);
    fano = var(noise) / mean(noise);

end
```

And when I plot the firing rate vs current plot, the noise affects a lot.

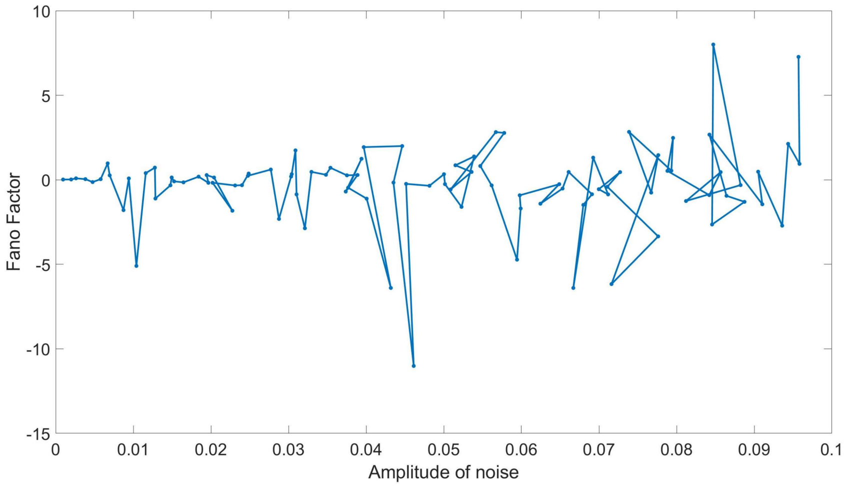
```
rate = zeros(20, 1);
I = zeros(20, 1);
fano = zeros(20, 1);
```

```
for k=1:20
    I(k) = -10 + k;
    [rate(k), fano(k)] = hhmodel_with_noise(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
symbol = 'I';
xlabel('Current Amplitude Ibar (\mu A)'); ylabel('Firing Rate (Hz)');
```



Then we would like to know that how is the fano factor related with the amplitude of the noise. I chose $\overline{I} = 5$.

```
I = 5;
fano = zeros(10, 1);
a = zeros(10, 1);
i = 1;
for k = 0.001:0.001:0.1
    noise = zeros(500, 1);
    for n = 1:500
        noise(n) = -k* rand() + k * rand();
    end
    fano(i) = var(noise) / mean(noise);
    a(i) = max(noise);
    i = i + 1;
end

plot(fano, a, '.-', 'LineWidth', 2, 'MarkerSize', 16);
xlabel('Amplitude of noise'); ylabel('Fano Factor');
```



And it can be seen that as the amplitude of noise rises, the fano factor would varied more around 0 as it does for small amplitudes.

Appendix

```
%% PART 1
%euler method simulator

deltat=0.01 ; %timestep
Tmax=50;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
V1=zeros(1,length(tlist));
V2=zeros(1,length(tlist));

%initialize
V0=0;
V1(1)=V0;
V2(1)=V0;

%circuit parameters
R=10;
C=1;

%define input currents
I1=1.05157 * ones(1,length(tlist));
I2=-64.681* sin(tlist);


for n=1:length(tlist)-1
    t=tlist(n);
    V1(n+1)=V1(n) + (-V1(n)/(R*C) + I1(n)/C )*deltat;
    V2(n+1)=V2(n) + (-V2(n)/(R*C) + I2(n)/C )*deltat;
end
%% voltage
```

```
figure(1)
subplot(211)
plot(tlist,V1,'.-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (ms)'); ylabel('V_1(t) (mV)');
axis([0 50 0 20])
line([30 30], [0 30]);
txt1 = '\leftarrow V_1(t=30ms) = 10mV';
text(30, V1(tlist==30), txt1);

subplot(212)
plot(tlist,V2,'-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (ms)'); ylabel('V_2(t) (mV)');
axis([0 50 0 20])
line([30 30], [0 30]);
txt2 = '\leftarrow V_2(t=30ms) = 10mV';
text(30, V1(tlist==30), txt2);
%% current
figure(1)
subplot(211)
plot(tlist,I1,'.-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (s)'); ylabel('I_1(t)');


subplot(212)
plot(tlist,I2,'-','LineWidth',2, 'MarkerSize', 10); hold on
xlabel('t (s)'); ylabel('I_2(t)');
%% ===============================================================
%% PART 2
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));
Ialist=zeros(1,length(tlist));
width=30; t1=30; t2=t1+width;
Ialist(tlist>=t1&tlist<t2) = 0.5;

%initialize
V0=0;
Vlist(1)=V0;

%circuit parameters
R=10;
C=1;

for n=1:length(tlist)-1
    t=tlist(n);
    Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) +Ialist(n)/C)*deltat;
end

figure(2);
```

```
plot(tlist,Vlist,'.-','LineWidth',2,'MarkerSize',10); hold on; grid on;
axis([0 100 0 10]);
xlabel('t (ms)'); ylabel('V(t) (mV)');

%% PART 3
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));
Ialist=zeros(1,length(tlist));
width=30; t1=30; t2=t1+width;
Ialist(tlist>=t1&tlist<t2) = 0.5;

%initialize
V0=0;
Vlist(1)=V0;
threshold = 10;
N = 1;

%circuit parameters
R=10;
C=1;

while max(Vlist) < threshold
    for n=1:length(tlist)-1
        t=tlist(n);
        Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) +Ialist(n)/C)*deltat;
    end
    N = N + 0.00001;
    Ialist(tlist>=t1&tlist<t2) = N * 0.5;
end
figure
plot(tlist,Vlist,'.-','LineWidth',2); hold on; grid on;
xlabel('t'); ylabel('V(t)');
%% PART 4
deltat=0.1 ; %timestep
Tmax=100;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
Vlist=zeros(1,length(tlist));

%initialize
V0=0;
Vlist(1)=V0;

%define input conductance
gapplist=zeros(1,length(tlist));
t1 = 10; t2 = t1 + 20;
gapplist(tlist>=t1&tlist<t2) = 0.5;
```

```
%circuit parameters
R=10;
C=1;
E=11;
for n=1:length(tlist)-1
    t=tlist(n);
    Vlist(n+1)=Vlist(n) + ( -Vlist(n)/(R*C) + gapplist(n)*(E-Vlist(n)) )*deltat;
end

figure
plot(tlist,Vlist,'.-','LineWidth',2); hold on; grid on;
xlabel('t'); ylabel('V(t)');
%%
deltat=0.1 ; %timestep
Tmax=100;
width = 20;

tlist=linspace(0,Tmax,Tmax/deltat +1) ;
V1=zeros(1,length(tlist));
V2=zeros(1,length(tlist));
V3=zeros(1,length(tlist));
V4=zeros(1,length(tlist));

%initialize
V0=0;
V1(1)=V0;
V2(1)=V0;
V3(1)=V0;
V4(1)=V0;

%define input conductance
g1=zeros(1,length(tlist));
t1 = 10; t2 = t1 + width;
g1(tlist>=t1&tlist<t2) = 0.1;

g2=zeros(1,length(tlist));
t1 = 40; t2 = t1 + width;
g2(tlist>=t1&tlist<t2) = 0.3;

g3=g1+g2;


%circuit parameters
R=10;
C=1;
E=11;
for n=1:length(tlist)-1
    t=tlist(n);
    V1(n+1)=V1(n) + ( -V1(n)/(R*C) + g1(n)*(E-V1(n)) )*deltat;
    V2(n+1)=V2(n) + ( -V2(n)/(R*C) + g2(n)*(E-V2(n)) )*deltat;
    V3(n+1)=V3(n) + ( -V3(n)/(R*C) + g3(n)*(E-V3(n)) )*deltat;
```

```
end

figure(3);
plot(tlist,V1,'.-','LineWidth',2, 'MarkerSize', 26); hold on
xlabel('t'); ylabel('V(t)');
plot(tlist,V2,'.-','LineWidth',2, 'MarkerSize', 26);
plot(tlist,V3,'.-','LineWidth',2, 'MarkerSize', 26);
plot(tlist,V1+V2,'.-','LineWidth',2, 'MarkerSize', 26);
legend('V1','V2','V3','V1+V2')
%%
rate = zeros(20, 1);
I = zeros(20, 1);
for k=1:20
    I(k) = -10 + k;
    rate(k) = hhmodel(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
xlabel('Current I_A (\mu A)'); ylabel('Firing Rate (Hz)');

%%
rate = zeros(20, 1);
I = zeros(20, 1);
for k=1:20
    I(k) = -10 + k;
    rate(k) = hhmodel(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
symbol = 'I';
xlabel('Current Amplitude Ibar (\mu A)'); ylabel('Firing Rate (Hz)');
%%
rate = zeros(20, 1);
I = zeros(20, 1);
fano = zeros(20, 1);

for k=1:20
    I(k) = -10 + k;
    [rate(k), fano(k)] = hhmodel_with_noise(I(k));
end

figure(4);
plot(I, rate, '.-','LineWidth', 2, 'MarkerSize', 26)
grid on;
symbol = 'I';
xlabel('Current Amplitude Ibar (\mu A)'); ylabel('Firing Rate (Hz)');
```

```
%%
I = 5;
fano = zeros(10, 1);
a = zeros(10, 1);
i = 1;
for k = 0.001:0.001:0.1
    noise = zeros(500, 1);
    for n = 1:500
        noise(n) = -k* rand() + k * rand();
    end
    fano(i) = var(noise) / mean(noise);
    a(i) = max(noise);
    i = i + 1;
end

plot(fano, a, '.-', 'LineWidth', 2, 'MarkerSize', 16);
xlabel('Amplitude of noise'); ylabel('Fano Factor');


%%====================================================
function firingrate = hhmodel(I)
    vna=50;  %set the constants
    vk=-77;
    vl=-54.4;
    gna=120;
    gk=36;
    gl=.3;
    c=1;

    v_init=-65;  %the initial conditions
    m_init=.052;
    h_init=.596;
    n_init=.317;

    npoints=50000;  %number of timesteps to integrate
    dt=0.01;        %timestep


    m=zeros(npoints,1); %initialize everything to zero
    n=zeros(npoints,1);
    h=zeros(npoints,1);
    v=zeros(npoints,1);
    time=zeros(npoints,1);

    m(1)=m_init; %set the initial conditions to be the first entry in the vectors
    n(1)=n_init;
    h(1)=h_init;
    v(1)=v_init;
    time(1)=0.0;
    numpeak = 0;
    thresh = 0;
```

```
    epsilon = 2;
    omega = 4;


    tic
    for step=1:npoints-1
        I = I + epsilon * sin(2*pi*time(step)*omega);
        v(step+1)=v(step)+((I - gna*h(step)*(v(step)-vna)*m(step)^3 ...
                   -gk*(v(step)-vk)*n(step)^4-gl*(v(step)-vl))/c)*dt;
        m(step+1)=m(step)+ (alpha_m(v(step))*(1-m(step))-beta_m(v(step))*m(step))*dt;
        h(step+1)=h(step)+ (alpha_h(v(step))*(1-h(step))-beta_h(v(step))*h(step))*dt;
        n(step+1)=n(step)+ (alpha_n(v(step))*(1-n(step))-beta_n(v(step))*n(step))*dt;
        time(step+1)=time(step)+dt;

        % spike detection: decreasing now and increasing before
         if ((step>1) && (v(step+1)<v(step)) && (v(step)>v(step-1))) && (v(step) > thresh)
            numpeak = numpeak + 1;
            peaktime(numpeak) = time(step+1);
         end
    end
    toc

    firingrate = numpeak / (npoints * dt);
end

%% ========================================
function [firingrate, fano] = hhmodel_with_noise(I)
    vna=50;  %set the constants
    vk=-77;
    vl=-54.4;
    gna=120;
    gk=36;
    gl=.3;
    c=1;

    v_init=-65;  %the initial conditions
    m_init=.052;
    h_init=.596;
    n_init=.317;

    npoints=50000;  %number of timesteps to integrate
    dt=0.01;        %timestep


    m=zeros(npoints,1); %initialize everything to zero
    n=zeros(npoints,1);
    h=zeros(npoints,1);
    v=zeros(npoints,1);
    time=zeros(npoints,1);
    noise = zeros(npoints, 1);
```

```matlab
    m(1)=m_init; %set the initial conditions to be the first entry in the vectors
    n(1)=n_init;
    h(1)=h_init;
    v(1)=v_init;
    time(1)=0.0;
    numpeak = 0;
    thresh = 0;
    epsilon = 2;
    omega = 4;

    tic
    for step=1:npoints-1
        % noise
        noise(step) = -0.05 * rand() + 0.05 * rand();
        % current with noise
        I = I + epsilon * sin(2*pi*time(step)*omega) + noise(step);
        v(step+1)=v(step)+((I - gna*h(step)*(v(step)-vna)*m(step)^3 ...
                   -gk*(v(step)-vk)*n(step)^4-gl*(v(step)-vl))/c)*dt;
        m(step+1)=m(step)+ (alpha_m(v(step))*(1-m(step))-beta_m(v(step))*m(step))*dt;
        h(step+1)=h(step)+ (alpha_h(v(step))*(1-h(step))-beta_h(v(step))*h(step))*dt;
        n(step+1)=n(step)+ (alpha_n(v(step))*(1-n(step))-beta_n(v(step))*n(step))*dt;
        time(step+1)=time(step)+dt;

        % spike detection: decreasing now and increasing before
         if ((step>1) && (v(step+1)<v(step)) && (v(step)>v(step-1))) && (v(step) > thresh)
            numpeak = numpeak + 1;
            peaktime(numpeak) = time(step+1);
         end
    end
    toc

    firingrate = numpeak / (npoints * dt);
    fano = var(noise) / mean(noise);

end
```