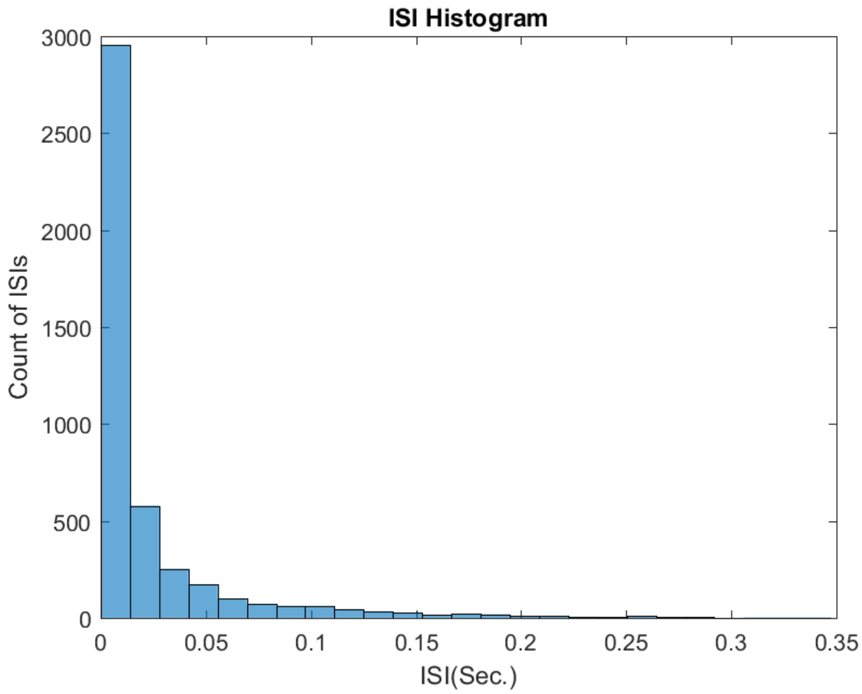
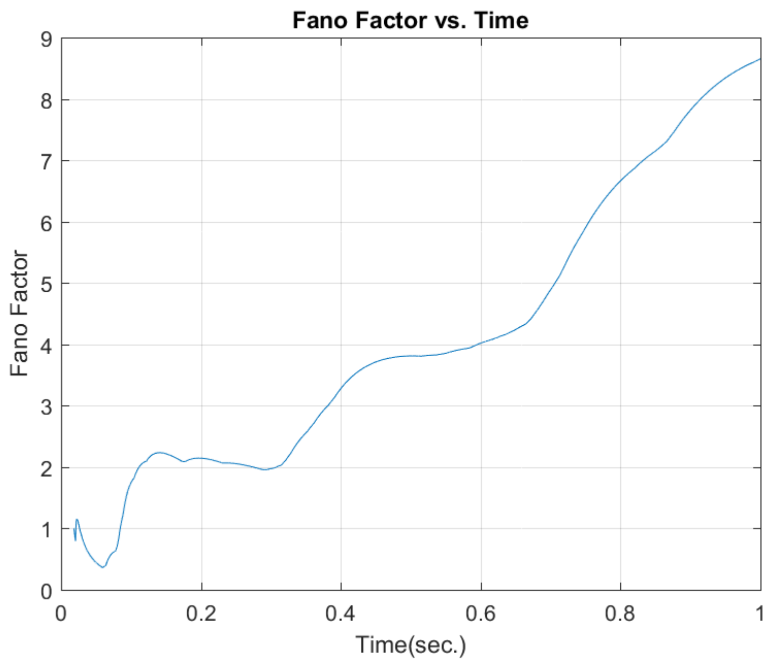
```matlab
%%
close all
clear
clc

dt = 0.01E-3;
t_max = 100;
time_vector = [0:dt:t_max];
sigma =50E-12;
I_app = randn(size(time_vector))*sigma/sqrt(dt);
tsteps = 250;
%define parameters
G_L = 8e-9;                 % Leak conductance (S)
leak_potential = -70E-3;
V_threshold = -50E-3;
reset_potential = -80E-3;
%R_m = 100E6;
R_m = 1/10E-9;
C_m = 100E-12;

tau_m = R_m*C_m;
a = 2E-9;
b = 0;
tau_SRA = 150e-3;
delta_th = 2e-3;

I_sra = zeros(size(time_vector));
I_sra(1) = 0;
V_m = zeros(size(time_vector));
V_m(1) = leak_potential;
spikes = zeros(size(time_vector));
for n = 2:length(time_vector)
V_m(n)= V_m(n-1) + (G_L*(leak_potential - V_m(n-1) + delta_th*exp((V_m(n-1) -
V_threshold)/delta_th))/tau_m - I_sra(n-1)/C_m + I_app(n-1)/C_m)*dt;
I_sra(n) = I_sra(n-1) + (a*(V_m(n-1) - leak_potential)/tau_SRA - I_sra(n-
1)/tau_SRA)*dt;
if V_m(n) > V_threshold
    V_m(n) = reset_potential;
    I_sra(n) = I_sra(n) + b;
    spikes(n) = 1;
end
end
spike_times = find(spikes)*dt;
ISI = diff(spike_times);
f1 = figure;
histogram(ISI, 25)
title('ISI Histogram');
xlabel('ISI(Sec.)');
ylabel('Count of ISIs');
saveas(f1, sprintf('f1_ISIHistogram.png'));
```

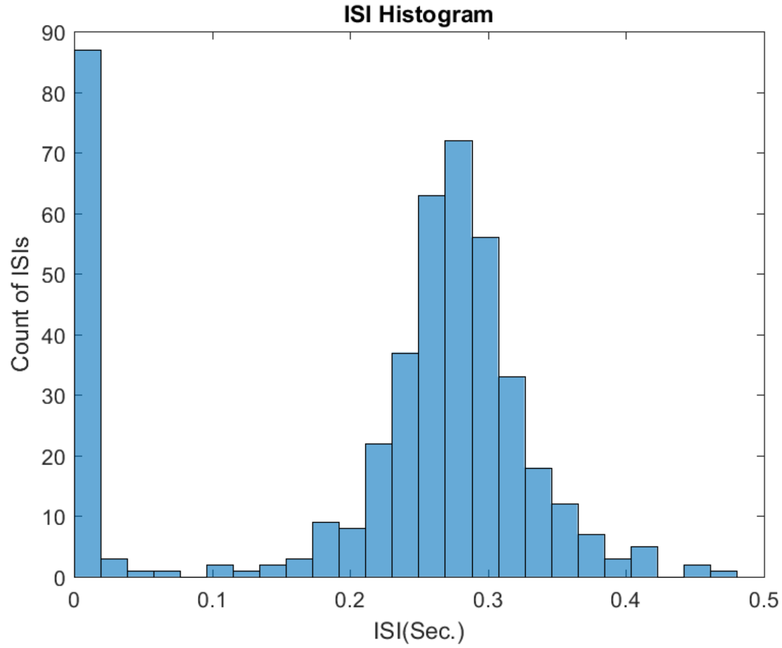



```
CV=std(ISI)/mean(ISI)
CV=1.8010
dT=0.1;
N=dT/dt;
No_spikes_100ms_window=zeros(size(1:t_max/100e-3));
for k=1:length(time_vector)/N
    No_spikes_100ms_window(k)=sum(spikes((k-1)*N+1:k*N));
end
No=No_spikes_100ms_window
mean_=(mean(No_spikes_100ms_window))
mean_  =4.4560
std_=((std(No_spikes_100ms_window))^2)
std_ =12.5506
fano = ((std(No_spikes_100ms_window))^2)/(mean(No_spikes_100ms_window))
fano= 2.8166
window=time_vector(1000:1e5);
spike=zeros(size(1000:1e5));
spike(1)=0;
l=1;
fano_1=zeros(size(window));
for k=1000:1e5
    s=spikes(k);
    spike(l+1)=spike(l)+s;
    std(spike(l+1));
    mean(spike(l+1));
    fano_1(l) = ((std(spike(1:l+1)))^2)/(mean(spike(1:l+1)));

    l=l+1;

end
f2=figure
plot(window,fano_1(1:end))
grid on
```

```
title('Fano Factor vs. Time');
xlabel('Time(sec.)');
ylabel('Fano Factor');
saveas(f2, sprintf('f2_FanoFactor.png'));
```

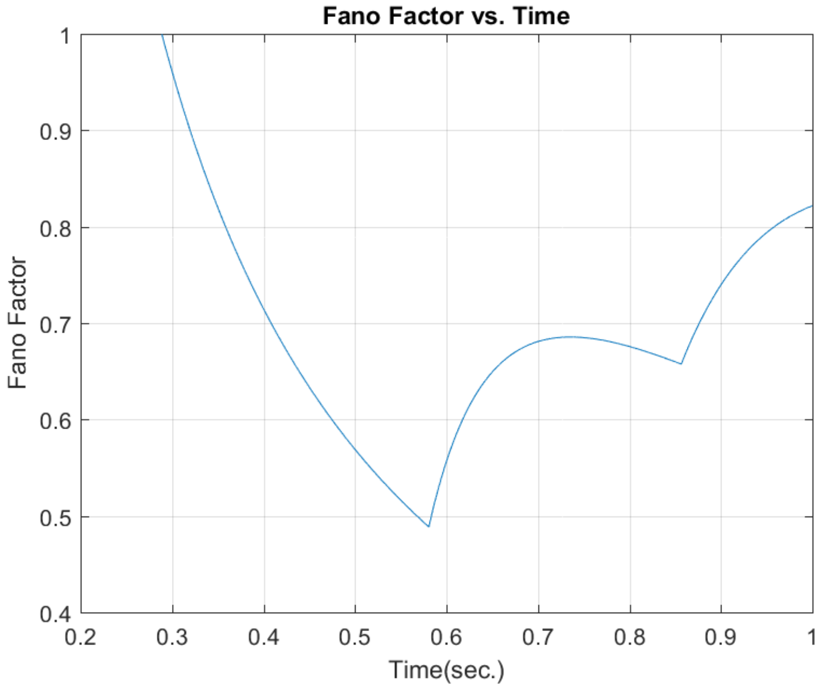



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%1.b
b=1e-9;
I_sra_b = zeros(size(time_vector));
I_sra_b(1) = 0;
V_m_b = zeros(size(time_vector));
V_m_b(1) = leak_potential;
spikes_b = zeros(size(time_vector));
for n = 2:length(time_vector)
V_m_b(n)= V_m_b(n-1) + (G_L*(leak_potential - V_m_b(n-1) +
delta_th*exp((V_m_b(n-1) - V_threshold)/delta_th))/tau_m - I_sra_b(n-1)/C_m +
I_app(n-1)/C_m)*dt;
I_sra_b(n) = I_sra_b(n-1) + (a*(V_m_b(n-1) - leak_potential)/tau_SRA -
I_sra_b(n-1)/tau_SRA)*dt;
if V_m_b(n) > V_threshold
    V_m_b(n) = reset_potential;
    I_sra_b(n) = I_sra_b(n) + b;
    spikes_b(n) = 1;
end
end
spike_times_b = find(spikes_b)*dt;
ISI_b = diff(spike_times_b);
f3 = figure;
histogram(ISI_b, 25)
title('ISI Histogram');
xlabel('ISI(Sec.)');
ylabel('Count of ISIs');
saveas(f3, sprintf('ISIHistogram_2.png'));
```




```
CV_2=std(ISI_b)/mean(ISI_b)
CV_2 =0.5322
dT=0.1;
N=dT/dt;
No_spikes_100ms_window_b=zeros(size(1:t_max/100e-3));
for k=1:length(time_vector)/N
    No_spikes_100ms_window_b(k)=sum(spikes_b((k-1)*N+1:k*N));
end
No_b=No_spikes_100ms_window_b
mean_b=(mean(No_spikes_100ms_window_b))
mean_b =0.4490
std_b=((std(No_spikes_100ms_window_b))^2)
std_b =0.4218
fano_b = ((std(No_spikes_100ms_window_b))^2)/(mean(No_spikes_100ms_window_b))
fano_b =0.9395
window=time_vector(1000:1e5);
spike_b=zeros(size(1000:1e5));
spike_b(1)=0;
l=1;
fano_1_b=zeros(size(window));
for k=1000:1e5
    s=spikes_b(k);
    spike_b(l+1)=spike_b(l)+s;
    std(spike_b(l+1));
    mean(spike_b(l+1));
    fano_1_b(l) = ((std(spike_b(1:l+1)))^2)/(mean(spike_b(1:l+1)));

    l=l+1;

end
f4=figure
plot(window,fano_1_b(1:end))
grid on
title('Fano Factor vs. Time');
```

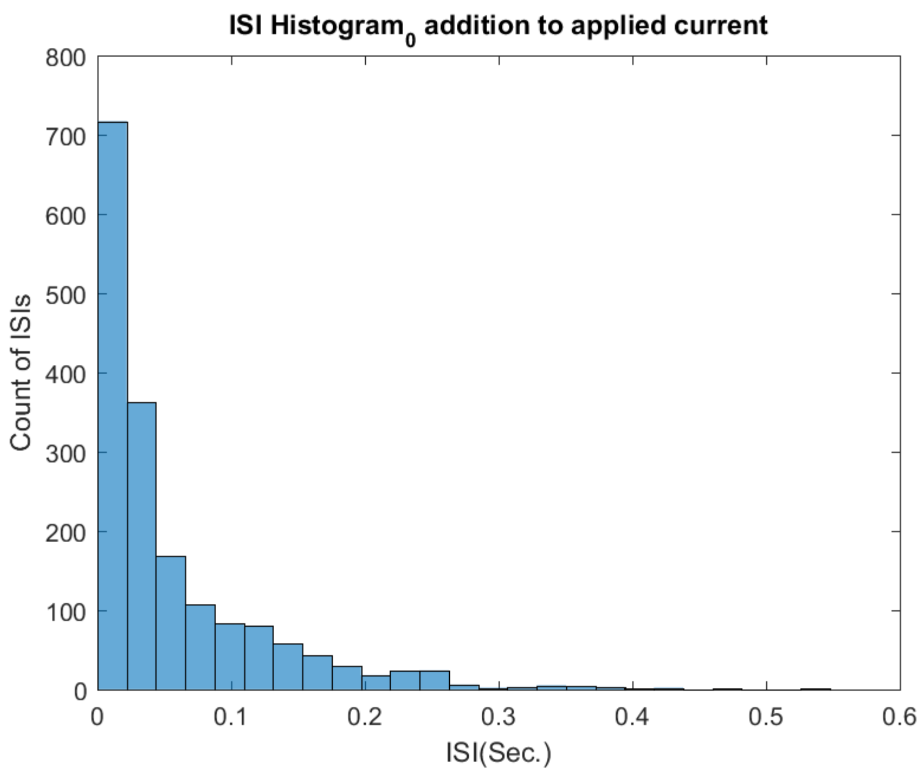
```
xlabel('Time(sec.)');
ylabel('Fano Factor');
saveas(f4, sprintf('f4_FanoFactor_b.png'));
```



- With a higher adaptation strength, the after hyperpolarization following a spike will be deeper and longer, making it harder for the neuron to reach the threshold potential for the next spike. This will lead to an increase in the average ISI, resulting in a shift of the ISI histogram towards longer ISIs.
- The ISI distribution is widen, indicating increased variability in spike timing.
- The Fano Factor measures the variability in the number of spikes within a specific time window. Since increasing b reduces the variability in ISIs, it will also lead to a decrease in the Fano Factor. This implies the spiking becomes more regular and less Poisson-like.
- The Fano Factor gets closer to 1 Depending on the original Fano factor, it gets closer to 1 indicating purely Poisson-like spiking, where the variance equals the mean, suggesting high randomness.

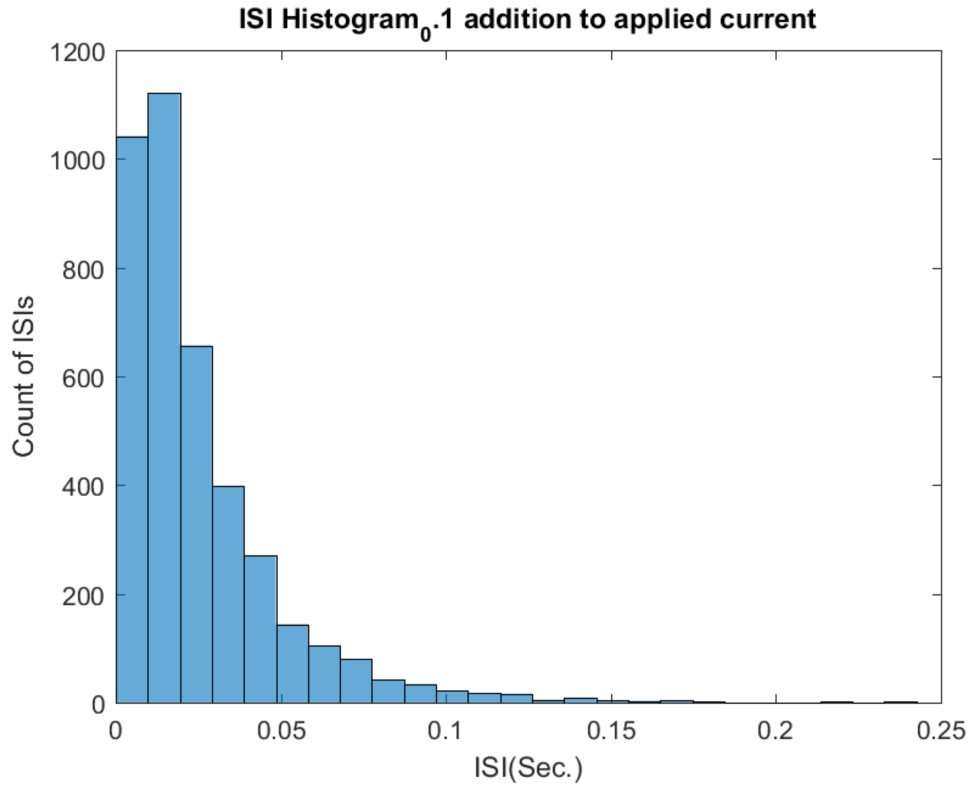
```
%%

%%%%%%%%%%%%%%%%%%%%%%%%
%1.c
b=0;
sigma =20E-12;

I_sra = zeros(3,length(time_vector));
I_sra(:,1) = 0;
V_m = zeros(3,length(time_vector));
V_m(:,1) = leak_potential;
spikes = zeros(3,length(time_vector));
for w=1:3
I_app(w,:) = randn(size(time_vector))*sigma/sqrt(dt)+0.1e-9*(w-1);
for n = 2:length(time_vector)
```

```
V_m(w,n)= V_m(w,n-1) + (G_L*(leak_potential - V_m(w,n-1) +
delta_th*exp((V_m(w,n-1) - V_threshold)/delta_th))/tau_m - I_sra(w,n-1)/C_m +
I_app(w,n-1)/C_m)*dt;
I_sra(w,n) = I_sra(w,n-1) + (a*(V_m(w,n-1) - leak_potential)/tau_SRA -
I_sra(w,n-1)/tau_SRA)*dt;
if V_m(w,n) > V_threshold
    V_m(w,n) = reset_potential;
    I_sra(w,n) = I_sra(w,n) + b;
    spikes(w,n) = 1;
end
end

end

spike_times_0 = find(spikes(1,:))*dt;
spike_times_1 = find(spikes(2,:))*dt;
spike_times_2 = find(spikes(3,:))*dt;
ISI_0 = diff(spike_times_0);
ISI_1 = diff(spike_times_1);
ISI_2 = diff(spike_times_2);
f5 = figure;
histogram(ISI_0, 25)
title('ISI Histogram_0 addition to applied current');
xlabel('ISI(Sec.)');
ylabel('Count of ISIs');
saveas(f5, sprintf('f5_ISIHistogram_1c_0.png'));
```

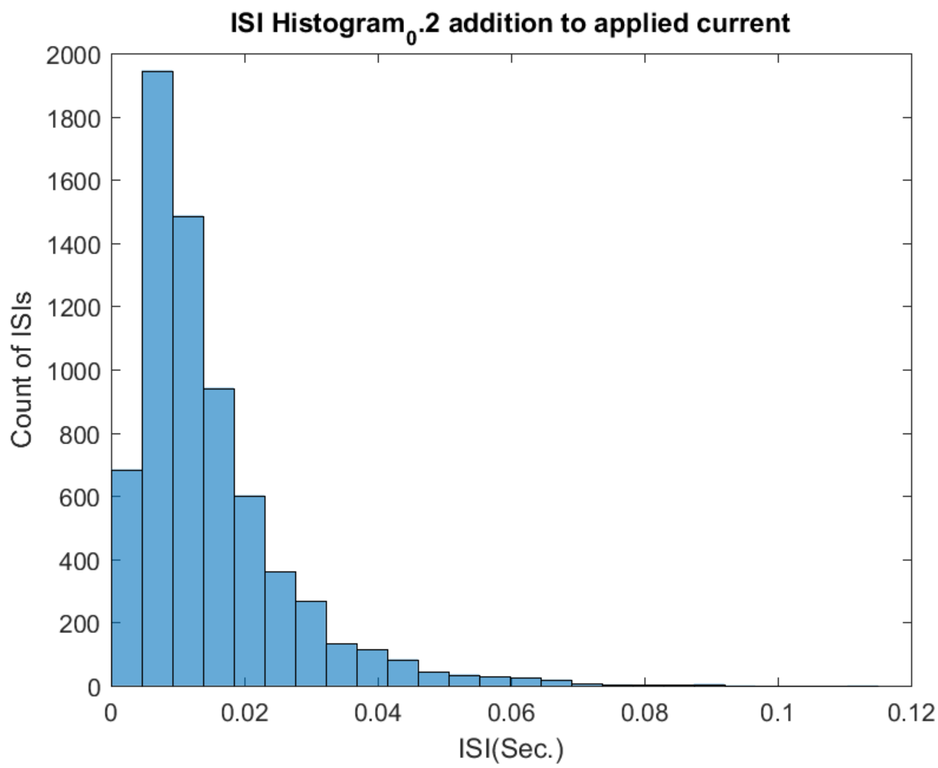



```
CV=std(ISI_0)/mean(ISI_0)
CV= 1.1925
f6=figure;
histogram(ISI_1, 25)
title('ISI Histogram_0.1 addition to applied current');
```

```
xlabel('ISI(Sec.)');
ylabel('Count of ISIs');
saveas(f6, sprintf('f6_ISIHistogram_1c_01.png'));
```




```
CV=std(ISI_1)/mean(ISI_1)
CV= 0.9556
f7=figure;
histogram(ISI_2, 25)
title('ISI Histogram_0.2 addition to applied current');
xlabel('ISI(Sec.)');
ylabel('Count of ISIs');
saveas(f7, sprintf('f7_ISIHistogram_1c_02.png'));
```

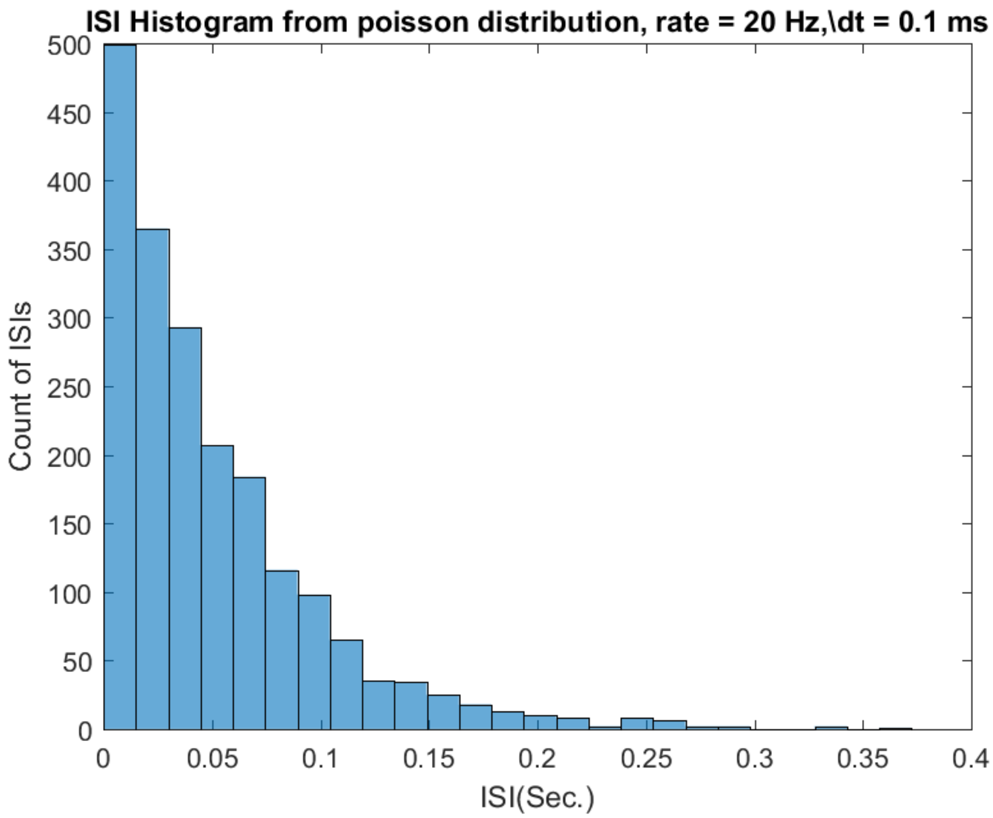



```
CV=std(ISI_2)/mean(ISI_2)
CV= 0.7872
dT=0.1;
N=dT/dt;
No_spikes_100ms_window=zeros(3,length(1:t_max/100e-3));
fano_c=zeros(1,3)
for w=1:3
for k=1:length(time_vector)/N
    No_spikes_100ms_window(w,k)=sum(spikes(w,((k-1)*N+1:k*N)));
end
fano_c(w) =
((std(No_spikes_100ms_window(w,:)))^2)/(mean(No_spikes_100ms_window(w,:)))
end
```

- The ISI distribution will be sharper with higher number of ISI counts in the left, meaning shorter ISIs will become more frequent. This is because higher current brings the membrane potential closer to the threshold, making it easier for the neuron to reach and cross the threshold, leading to more frequent spiking.
- The ISI distribution becomes narrower, indicating less variability in spike timing. This is because the increased current reduces the role of intrinsic noise and randomness in determining the timing of spikes.
- The Fano factor, which measures the variance in spike count relative to the mean, decrease because the stronger current reduces the randomness in spiking, leading to smaller fluctuations in spike count within windows.

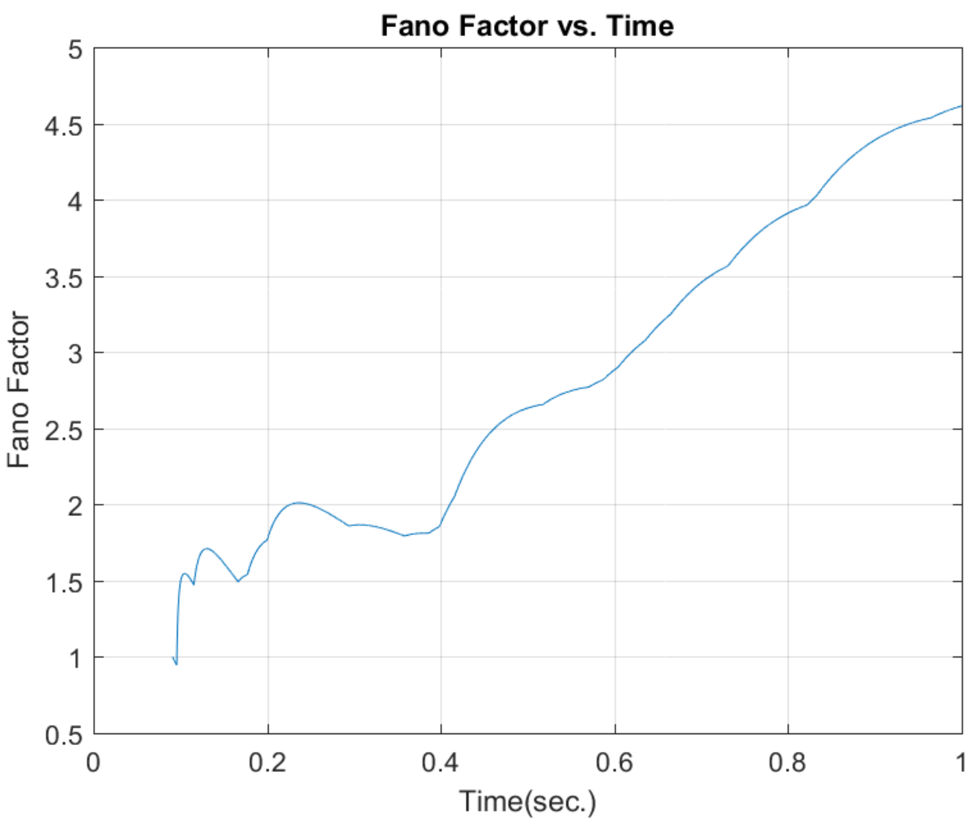
```
%%
```

```
%Part B

dt = 0.1e-3; % s
rate = 20; % Hz
t_max=100;
time_vector_B = 0:dt:100-dt; % initializing time
SPIKES = rand(1,length(time_vector_B))<rate*dt;
ISI = diff(find(SPIKES))*dt; % calculating ISI
f8=figure
histogram(ISI,25); title('ISI Histogram from poisson distribution, rate = 20
Hz,\dt = 0.1 ms'); xlabel('ISI(Sec.)'); ylabel('Count of ISIs');
saveas(f8, sprintf('f8_PoissonHistogram.png'));
```



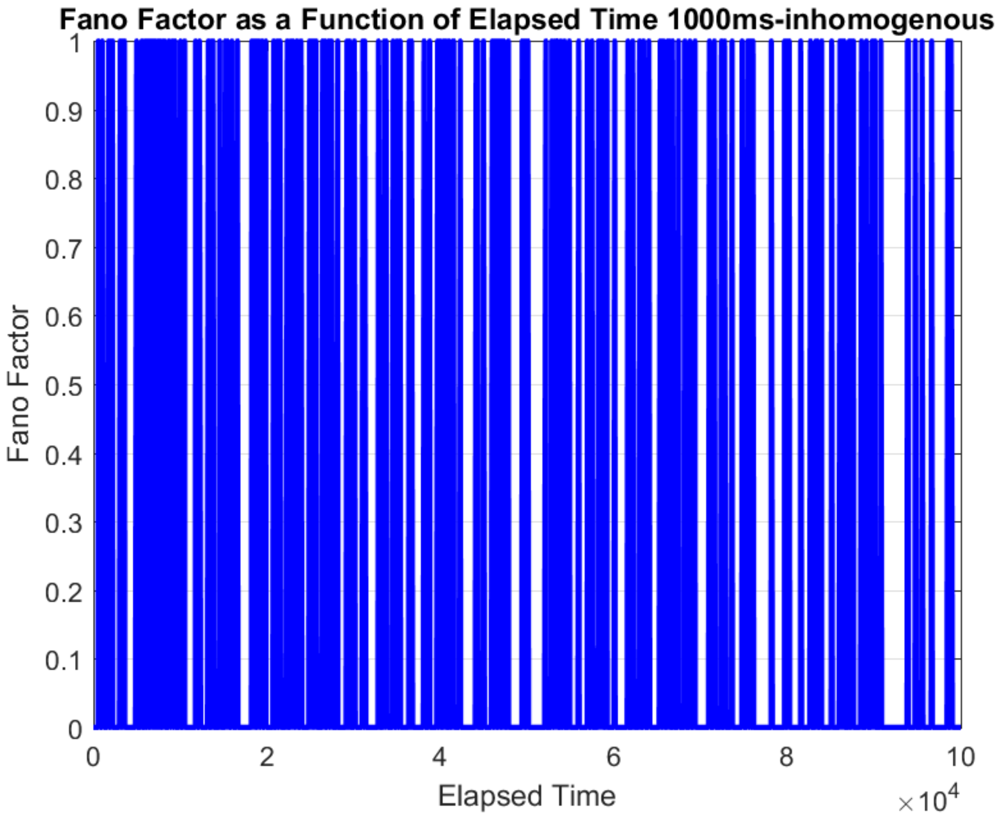

```
CV=std(ISI)/mean(ISI)
CV= 0.9740
dT=0.1;
N=dT/dt;
No_spikes_100ms_window=zeros(size(1:t_max/100e-3));
for k=1:length(time_vector_B)/N
    No_spikes_100ms_window(k)=sum(SPIKES((k-1)*N+1:k*N));
end
No9=No_spikes_100ms_window
mean9=(mean(No_spikes_100ms_window))
std9=((std(No_spikes_100ms_window))^2)
fano = ((std(No_spikes_100ms_window))^2)/(mean(No_spikes_100ms_window))

window=time_vector_B(100:1e4);
spike=zeros(size(100:1e4));
spike(1)=0;
```

```
l=1;
fano_1=zeros(size(window));
for k=100:1e4
    s=SPIKES(k);
    spike(l+1)=spike(l)+s;
    std(spike(l+1));
    mean(spike(l+1));
    fano_1(l) = ((std(spike(1:l+1)))^2)/(mean(spike(1:l+1)));

    l=l+1;

end
f9=figure
plot(window,fano_1(1:end))
grid on
title('Fano Factor vs. Time');
xlabel('Time(sec.)');
ylabel('Fano Factor');
saveas(f9, sprintf('f9_FanoFactor.png'));
```

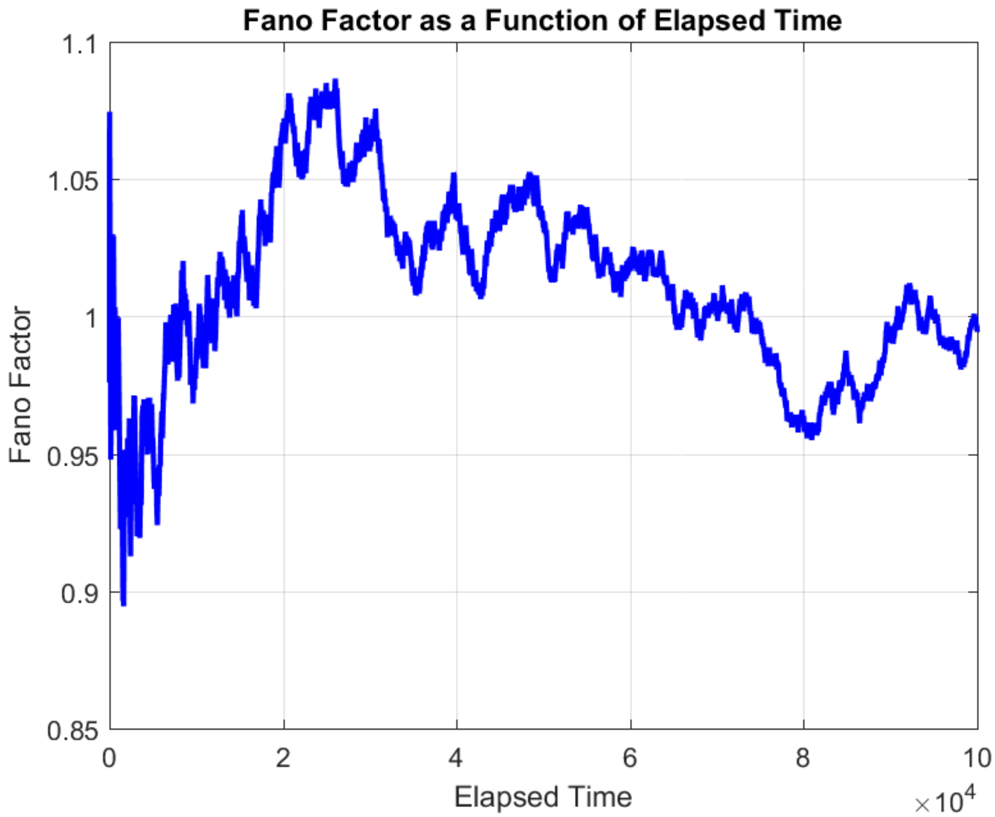



```
%%2b-B
t=0:dt:10-dt;
for k=1:1000
   spikemat(k,:)=rand(size(t))<rate*dt;
end
f10=figure
plot(spikemat(1,:), 'b-', 'LineWidth', 2);
xlabel('Elapsed Time');
ylabel('Fano Factor');
```

```
title('Fano Factor as a Function of Elapsed Time 1000ms-inhomogenous');
grid on;
saveas(f10, sprintf('f10_FanoFactor.png'));
```

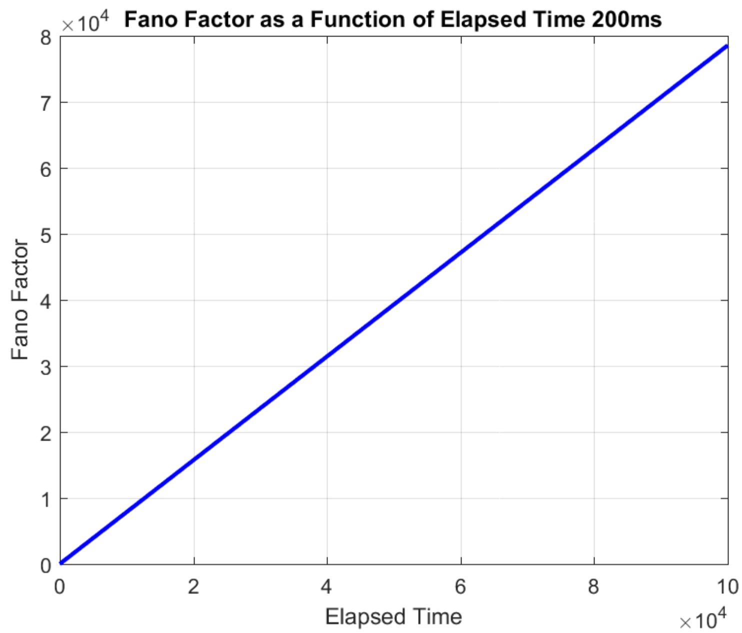



```
N_T = cumsum(spikemat,2)
variance_across_rows = var(N_T,0, 1)
mean_across_rows = mean(N_T, 1)
% Calculate Fano factor
fano_factor = variance_across_rows ./ mean_across_rows

% Plot the Fano factor as a function of elapsed time
elapsed_time = 1:size(N_T, 2);  % Assuming each row corresponds to a specific
time point
f11=figure;
plot(elapsed_time, fano_factor, 'b-', 'LineWidth', 2);
xlabel('Elapsed Time');
ylabel('Fano Factor');
title('Fano Factor as a Function of Elapsed Time');
grid on;
saveas(f11, sprintf('f11_FanoFactor.png'));
```

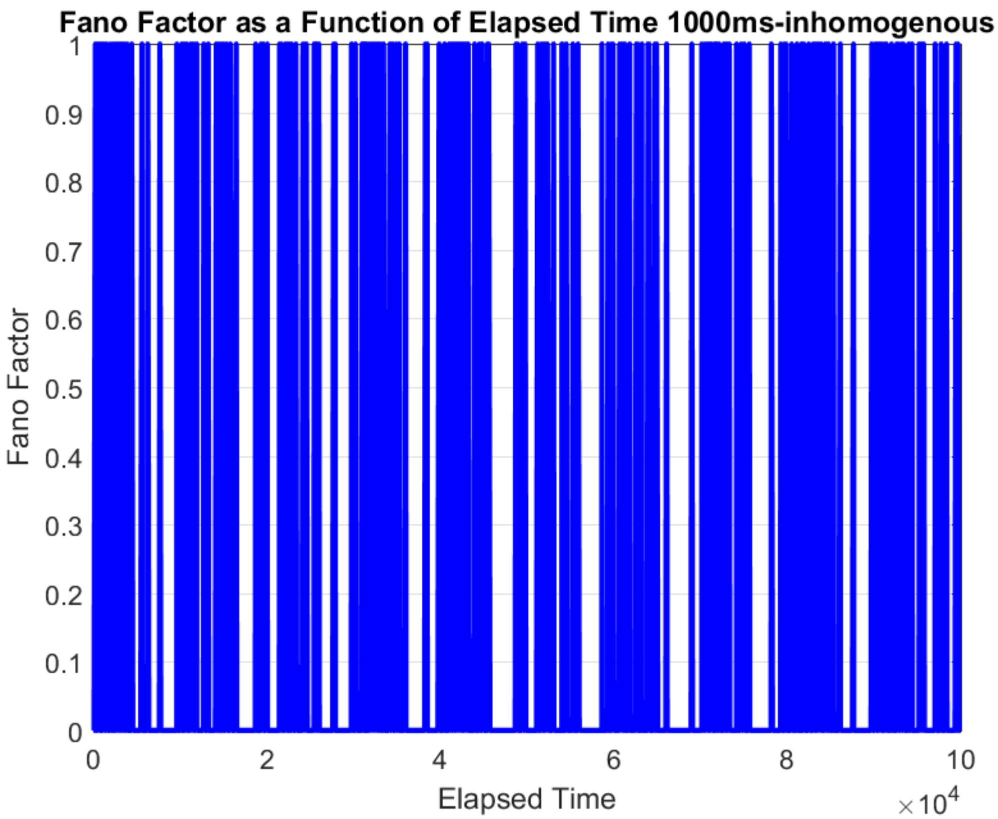



```
% Assuming you have spike times stored in the variable 'spike_times' and the
duration of each trial is known

window_size = 0.2; % 200 ms window size
trial_duration = 10; % Duration of each trial (in seconds), provide the
actual value
% Calculate the number of time windows
num_trials =(trial_duration / window_size);
num_windows=window_size/dt;
% Initialize arrays to store spike counts for each window and trial

spike_counts = zeros(num_trials,length(t));

% Loop over each trial
for trial = 1:num_trials
    % Loop over each time windowt
    for window = 1+(trial-1)*num_windows:num_windows*trial
        % Define the start and end time of the current window

         % Count the number of spikes within the current window for the
current trial
        spike_counts(trial,:) = spikemat(trial,window)+spike_counts(trial,:)
;
    end
end
N_T = cumsum(spike_counts,2)
variance_across_rows = var(N_T, 0,1)
mean_across_rows = mean(N_T, 1)
```

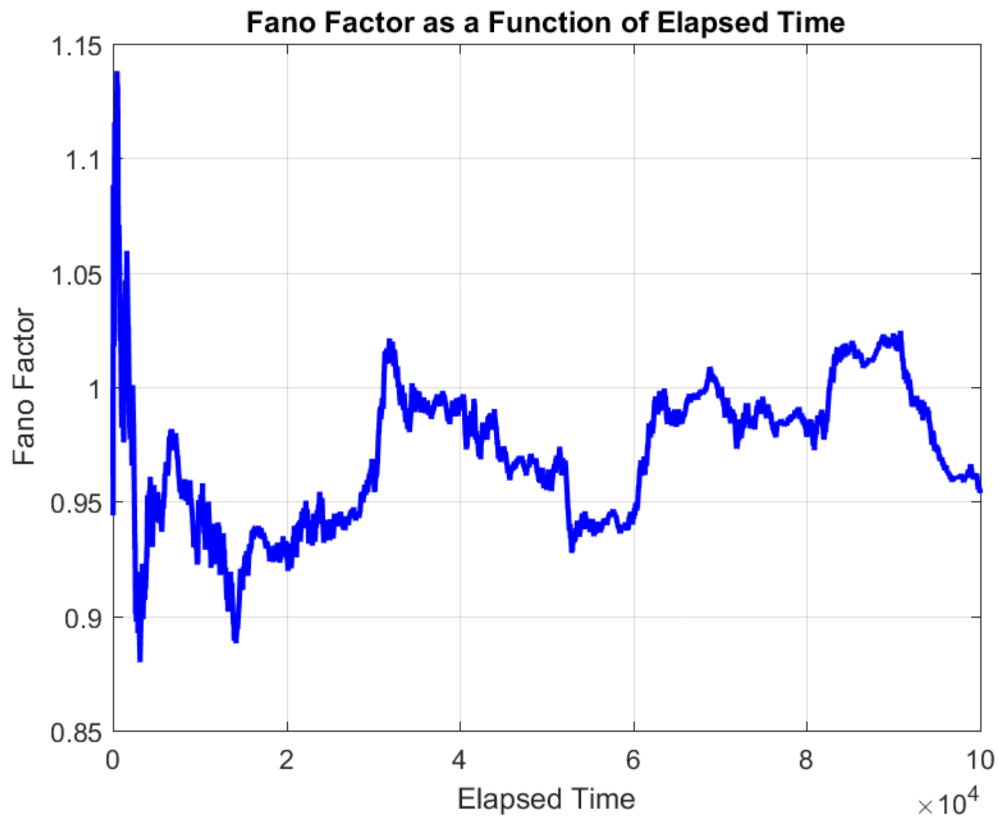
```
% Calculate Fano factor
fano_factor = variance_across_rows ./ mean_across_rows

% Plot the Fano factor as a function of elapsed time
elapsed_time = 1:size(spike_counts, 2);
f12=figure
plot(fano_factor, 'b-', 'LineWidth', 2);
xlabel('Elapsed Time');
ylabel('Fano Factor');
title('Fano Factor as a Function of Elapsed Time 200ms');
grid on;
saveas(f12, sprintf('f12_FanoFactor_200ms.png'));
```




```
%%%%%
t=0:dt:10-dt;
rate = 25 + 20*sin(2*pi*t);

for k=1:1000
   spikemat(k,:)=rand(size(t))<rate*dt;
end


f13=figure
plot(spikemat(1,:), 'b-', 'LineWidth', 2); %one trial
xlabel('Elapsed Time');
ylabel('Fano Factor');
title('Fano Factor as a Function of Elapsed Time 1000ms-inhomogenous');
grid on;
saveas(f13, sprintf('f13_FanoFactor_1000ms.png'));
```




```
N_T = cumsum(spikemat,2);
variance_across_rows = var(N_T,0, 1)
mean_across_rows = mean(N_T, 1)
fano_factor = variance_across_rows ./ mean_across_rows
elapsed_time = 1:size(N_T, 2);
f14=figure;
plot(fano_factor, 'b-', 'LineWidth', 2);
xlabel('Elapsed Time');
ylabel('Fano Factor');
title('Fano Factor as a Function of Elapsed Time');
grid on;
saveas(f14, sprintf('f14_FanoFactor-1000.png'));
```




```
window_size = 0.2; % 200 ms window size
trial_duration = 10; % Duration of each trial (in seconds), provide the
actual value
% Calculate the number of time windows
num_trials =(trial_duration / window_size);
num_windows=window_size/dt;
% Initialize arrays to store spike counts for each window and trial

spike_counts = zeros(num_trials,length(t));

% Loop over each trial
for trial = 1:num_trials
    % Loop over each time windowt
    for window = 1+(trial-1)*num_windows:num_windows*trial
        % Define the start and end time of the current window

         % Count the number of spikes within the current window for the
current trial
        spike_counts(trial,:) = spikemat(trial,window)+spike_counts(trial,:)
;
    end
end
N_T = cumsum(spike_counts,2);
variance_across_rows = var(N_T, 0,1);
mean_across_rows = mean(N_T, 1);
% Calculate Fano factor
fano_factor = variance_across_rows ./ mean_across_rows
Columns 99947 through 99960
```

```
    0.9545    0.9542    0.9543    0.9543    0.9543    0.9548    0.9548    0.9551    0.9551
0.9551    0.9549    0.9551    0.9548    0.9550

  Columns 99961 through 99974

    0.9552    0.9551    0.9551    0.9552    0.9552    0.9551    0.9551    0.9553    0.9552
0.9554    0.9553    0.9553    0.9553    0.9553

  Columns 99975 through 99988

    0.9550    0.9550    0.9547    0.9550    0.9550    0.9550    0.9548    0.9546    0.9547
0.9546    0.9547    0.9546    0.9548    0.9545

  Columns 99989 through 100000

    0.9547    0.9548    0.9549    0.9549    0.9549    0.9550    0.9552    0.9545    0.9544
0.9544    0.9540    0.9536
```

```
% Plot the Fano factor as a function of elapsed time
elapsed_time = 1:size(N_T, 2);
f15=figure
plot(fano_factor, 'b-', 'LineWidth', 2);
xlabel('Elapsed Time');
ylabel('Fano Factor');
title('Fano Factor as a Function of Elapsed Time 200ms');
grid on;
saveas(f15, sprintf('f15_FanoFactor_200ms.png'));
```