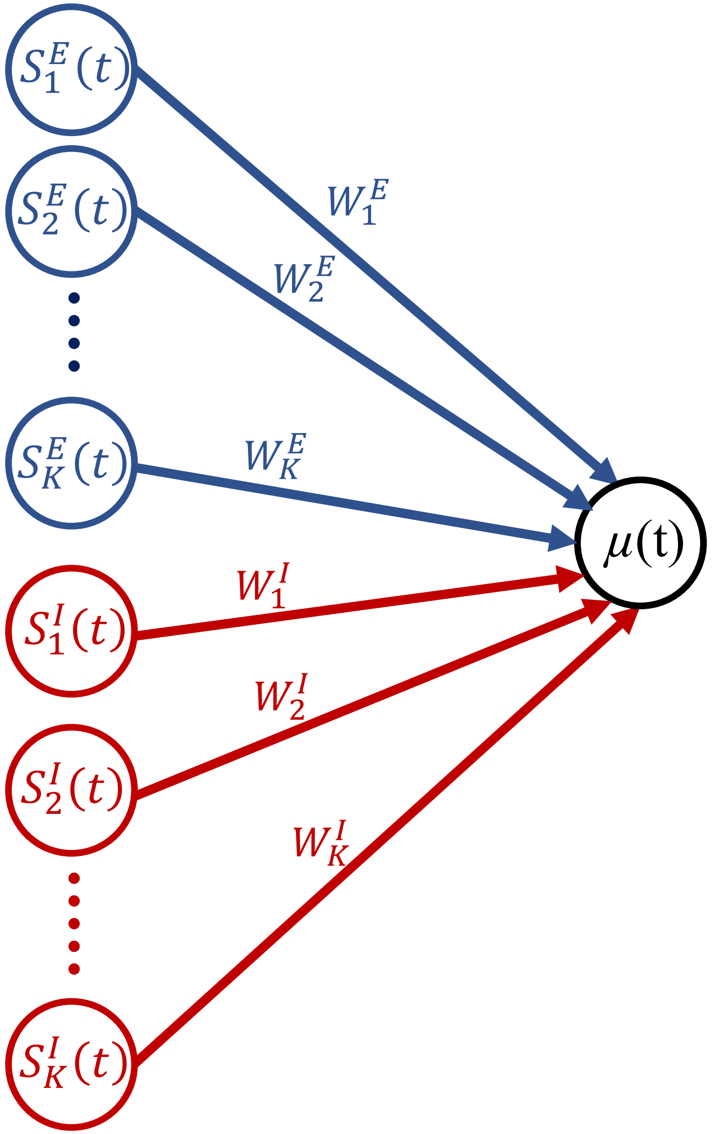
HW2 Prob. 2 Noise and "Balanced" Network



$$\mu(t) = \sum_{j=1}^{K} W_j^E S_j^E(t) + \sum_{j=1}^{K} W_j^I S_j^I(t), \qquad S_j^{I,E} = \{0,1\}\ (\text{Binary})$$

Explore the mean $\langle \mu \rangle$ and $\mathrm{Var}(\mu)$ as a function of number of inputs, K.

- Prob. 2.1 $W_j^E = \frac{1}{K}$, $W_j^I = 0$
- Prob. 2.2 $W_j^E = \frac{1}{K}$, $W_j^I = \frac{-1}{K}$
- Prob. 2.3 $W_j^E = \frac{1}{\sqrt{K}}$, $W_j^I = \frac{-1}{\sqrt{K}}$

W19 PHYS 178/278 Prob. 2 Noise and 'Balanced' Network

`MATLAB code demo, Feb. 6th 2019`

- You are welcome to use any programming language you are familier with!
- Please use "Run Section" when you read and run the code step-by-step.
- You need to fill in the missing section for Prob. 2.2 and Prob 2.3.

Contents

Step-by-step demo of Prob 2.1

STEP 1

Generate K excitatory inputs occurring in one time step (dt) as a one-dimensional binary vector.

```
Hint: use MATLAB command (rand(K,1) <= m*dt) to generate a K-by-1 column
vector, where m*dt is the spiking probability of one time bin (dt).
You can choose the spiking probability (m) between 0 < m < 1.
-------------------------------------------------------------------------
```

Spiking rate = Probability to have a spike per unit of time. You may choose any m between 0 < m < 1.

```
m = 0.2;
```

Number of presynatic connection. Let's set K = 20 for testing.

```
K = 20;
```

The time for each bin.

```
dt = 1;
```

K-by-1 column vector

```
rand(K, 1) <= m*dt
```

```
% which is equivalent to:  rand(K, 1) >= (1-m)*dt
```

```
ans =

  20×1 logical array

   0
   1
   1
   0
   1
   0
   0
   0
   0
   1
   0
   1
   0
   0
   0
   0
   0
   0
   0
   0
```

Q: Why does this code mean?

Type the following two lines in the Command Window and see what it returns

```
v = rand(10, 1)
```

```
v <= m*dt
```

STEP 2

Extend the K excitatory inputs (K-by-1) from one-time step to a sequence over a period of time T >> dt (ex: you may set dt = 1 and T = 10000).

```
Hint: how many dt bins (n) are there in the time interval T?
Find the number of bins (n) and use (rand(K,n) <= m*dt) to generate a
binary K-by-n matrix. Each column represents K inputs at each time step,
while each row is the input sequence sending from one of the presynaptic
neurons.
--------------------------------------------------------------------------
```

```
% Set the total time T.
% Let's try a small time period fist. You are going to set T = 10000 latter
T = 10;
```

```
% Number of bins over time period T.
n = T/dt;          % Please define n in terms of T and dt!!!

% K-by-n column vector.
S = rand(K, n) <= m*dt
```

```
S =

  20×10 logical array

   0   0   0   1   0   0   0   0   0   0
   0   0   1   0   0   1   1   0   0   0
   0   0   0   1   0   0   0   0   0   0
   0   1   0   1   0   0   0   0   1   0
   0   1   0   0   0   1   0   0   1   0
   0   0   0   1   0   1   0   0   0   0
   0   0   0   0   0   0   1   0   0   0
   0   1   0   0   0   0   0   0   1   0
   0   1   0   0   1   1   1   1   1   0
   0   1   0   0   0   1   0   1   0   0
   1   0   1   0   1   0   0   0   0   0
   0   0   0   0   0   0   1   0   1   0
   1   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   1   0   1
   0   0   0   0   0   0   1   0   0   0
   0   0   0   1   0   0   1   0   0   0
   0   0   0   0   0   0   0   1   1   0
   0   0   0   1   0   0   1   0   0   0
```

STEP 3 Compute the total input received at each time step.

From the K-by-n matrix, how would you get a total input $\mu(t)$ received at each time step?

`Hint :(1/K)*sum(rand(K,n)<= m*dt,1) and it will be a 1-by-n row vector.`
`--------------------------------------------------------------------------`

Input $\mu(t)$ is

```
input = (1/K)*sum(S, 1)   % For Prob. 2.1, we set the weight as (1/K).
```

```
input =

  Columns 1 through 7

    0.1000    0.2500    0.1500    0.3000    0.1000    0.3000    0.3500

  Columns 8 through 10
```

```
    0.2000    0.3000    0.0500
```

STEP 4

Now you have a input sequence (1-by-n row vector), you will be able to calculate the mean $\langle\mu\rangle$, and the variance $\mathrm{Var}\,(\mu)$.

```
Hint: You may need some built-in MATLAB functions. Try to look them up.
------------------------------------------------------------------------
```

Mean

```
input_mean = mean(input)
```

```
input_mean =

    0.2100
```

it's equivalent to

```
sum(input)/length(input)
```

```
ans =

    0.2100
```

Variance

```
input_var = var(input)
```

```
input_var =

    0.0110
```

it's equivalent to

```
sum((input-input_mean).^2)/(length(input)-1)
```

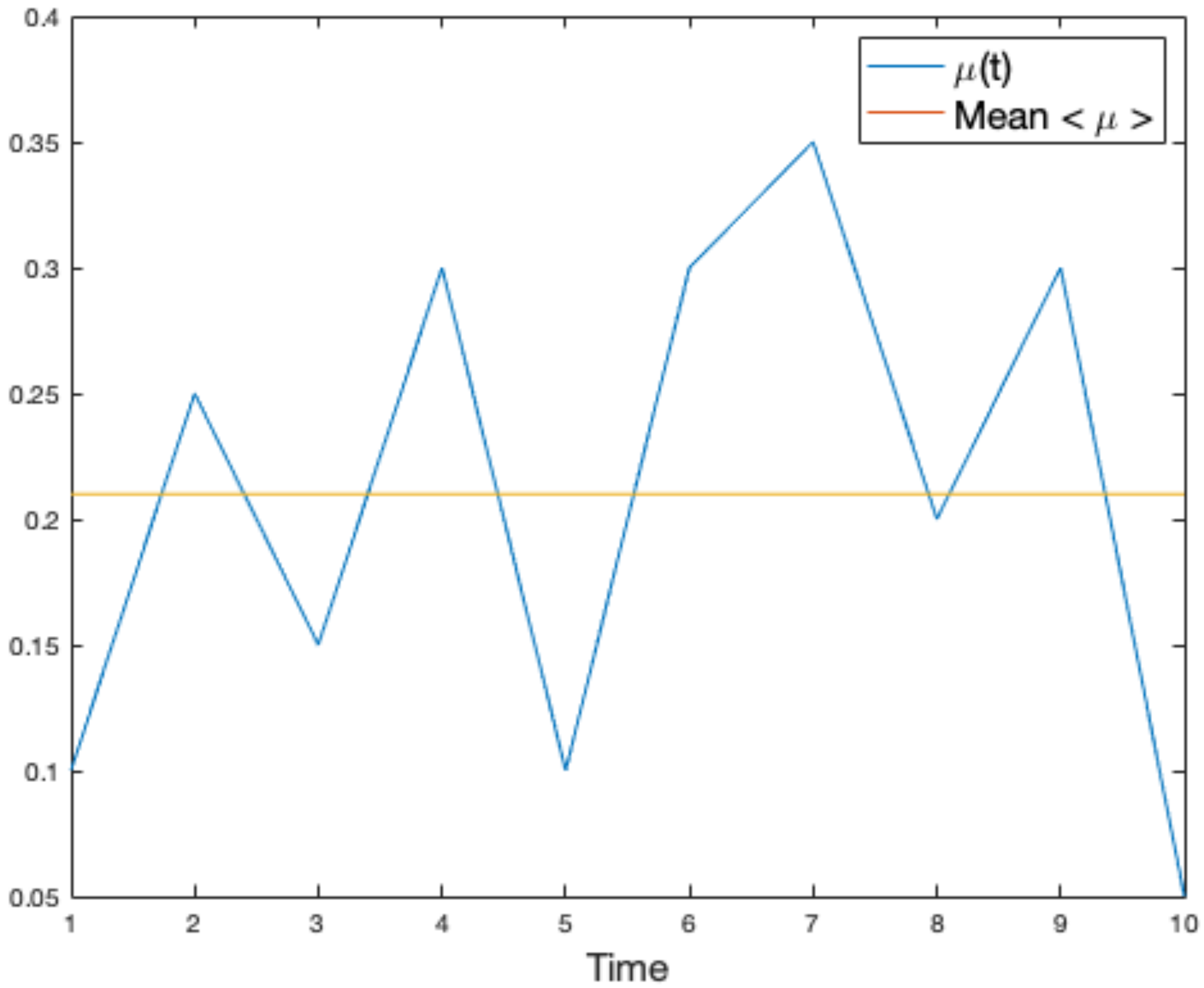
```
ans =
```

```
0.0110
```

- Visualizing the input as a function of time, $\mu(t)$.

```
You don't need to include this plot in your HW2.
```

```
figure(1)
plot(input); hold on
plot([1, T], input_mean*ones(2))
xlabel('Time', 'FontSize', 15)
legend({'\mu(t)', 'Mean < \mu >'},'FontSize',15)
hold off
```



Summary of STEP 1 to 4

A postsynaptic neuron receives purely excitatory inputs from K presynaptic neurons over a period of time T. The goal is to compute the mean and variance of total input (sum over K) across time.

```
clear all
close all

% Spiking rate = Probability to have a spike per unit of time.
% Choose any m between 0 < m < 1 !!!
m = 0.2;
```

```
% Number of presynatic connection. Now, we set a larger K = 200.
K = 200;

% The time for each bin.
dt = 1;

% Set the total time T = 10000.
T = 10000;

% K-by-n matrix. (K inputs, n time steps)
n = T/dt;
input = (1/K)*sum(rand(K, n) <= m*dt, 1);

% Mean
input_mean = mean(input)  % Q: Is the mean close to the m value you set?

% Variance
input_var = var(input)
```

```
input_mean =

    0.1998


input_var =

   7.9511e-04
```

STEP 5 & 6

Write a for-loop that calculate the mean $\langle\mu\rangle$, and variance $\mathrm{Var}\,(\mu)$ with different connections K = 200, 400, 600, 800, 1000, 1500, 2000, 3500, 5000. For each K, you may try to run multiple trials and take the average.

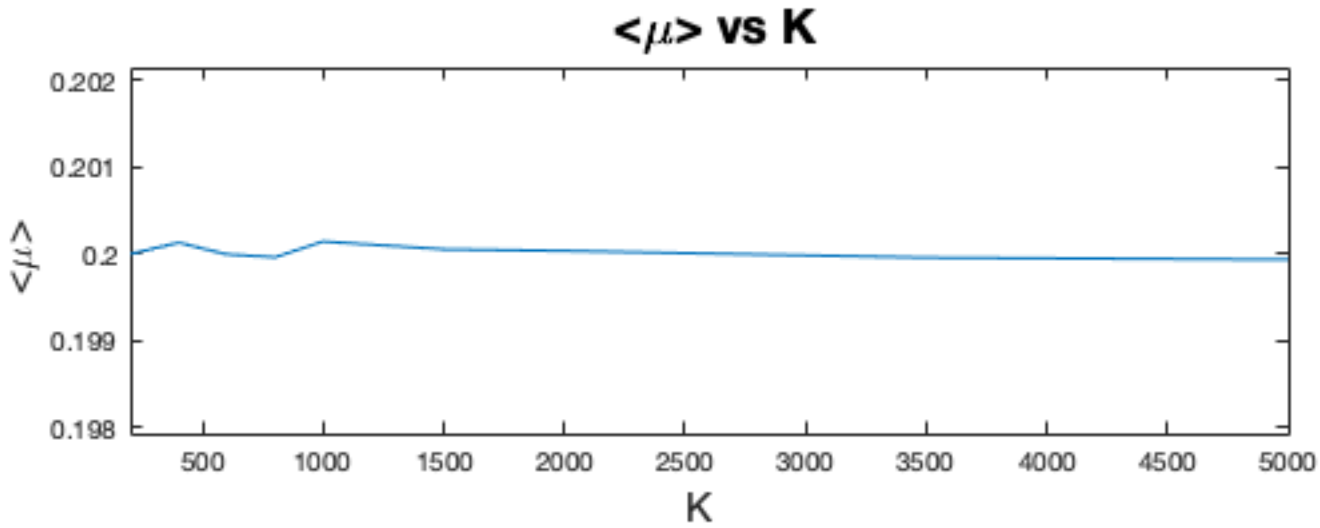
Plot $\langle\mu\rangle$ vs K, and $\mathrm{Var}\,(\mu)$ vs K. Compare each curve with its own zero (horizontal line y = 0, label it on the plot would be helpful), and briefly describe the trend for each.

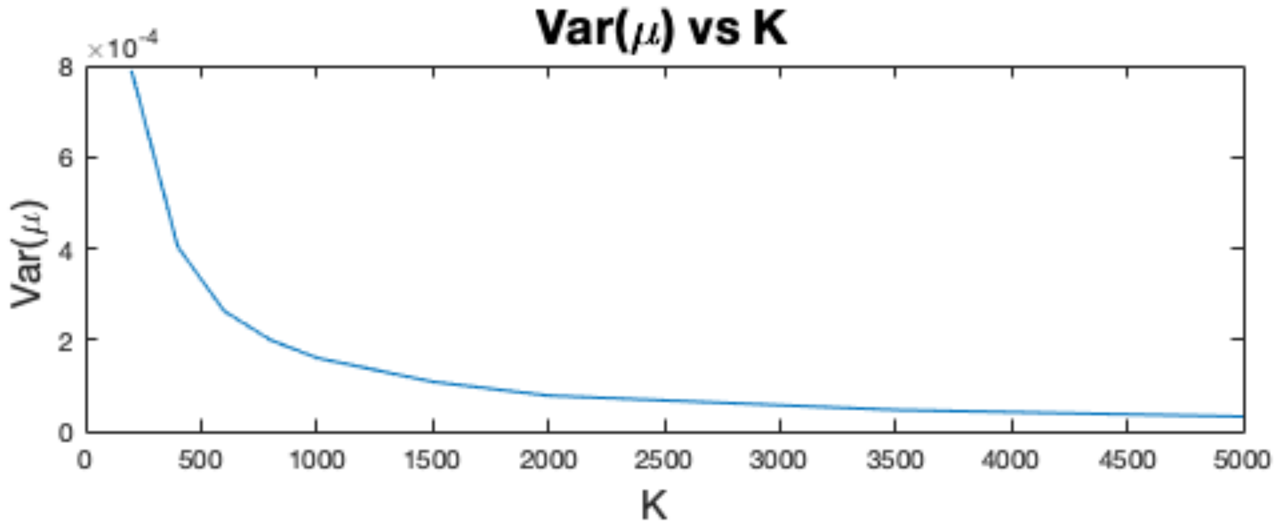
Prob 2.1 Excitatory Synaptic Input, with the weight (1/K)

```
clear
% Spiking rate = Probability to have a spike per unit of time!!!
m = 0.2;

% Number of presynatic connection
K_list = [200, 400, 600, 800, 1000, 1500, 2000, 3500, 5000];

% Set the total time T = 10000.
T = 10000;
dt = 1;
t = 0+dt:dt:T;
n = T/dt;

% Preallocation a vector to store the values.
```

```
input_Mean = zeros(size(K_list));
input_Var = zeros(size(K_list));

% For-loop
ind = 0;

for K = K_list

    input = (1/K)*sum(rand(K, n) <= m*dt, 1);
    % Please modified the code as needed!!!

    ind = ind + 1;
    input_Mean(ind) = mean(input);
    input_Var(ind) = var(input);

end
```

```
figure(2)

% Plot <input> vs K
subplot(2,1,1)
plot(K_list, input_Mean); hold on
ylabel('<\mu> ', 'FontSize', 15)
xlabel('K  ', 'FontSize', 15)
axis([min(K_list) max(K_list) 0.99*min(input_Mean) 1.01*max(input_Mean)])
title(' <\mu> vs K', 'FontSize', 18)

% Plot Var(input) vs K
subplot(2,1,2)
plot(K_list, input_Var); hold on
ylabel('Var(\mu) ', 'FontSize', 15)
xlabel('K  ', 'FontSize', 15)
title('Var(\mu) vs K', 'FontSize', 18)
```





Note that $\mathrm{Var}\,(\mu)$ vs K has a power-law relationship

$$\mathrm{Var}(\mu) = \mathrm{a_2 K^{a_1}}$$

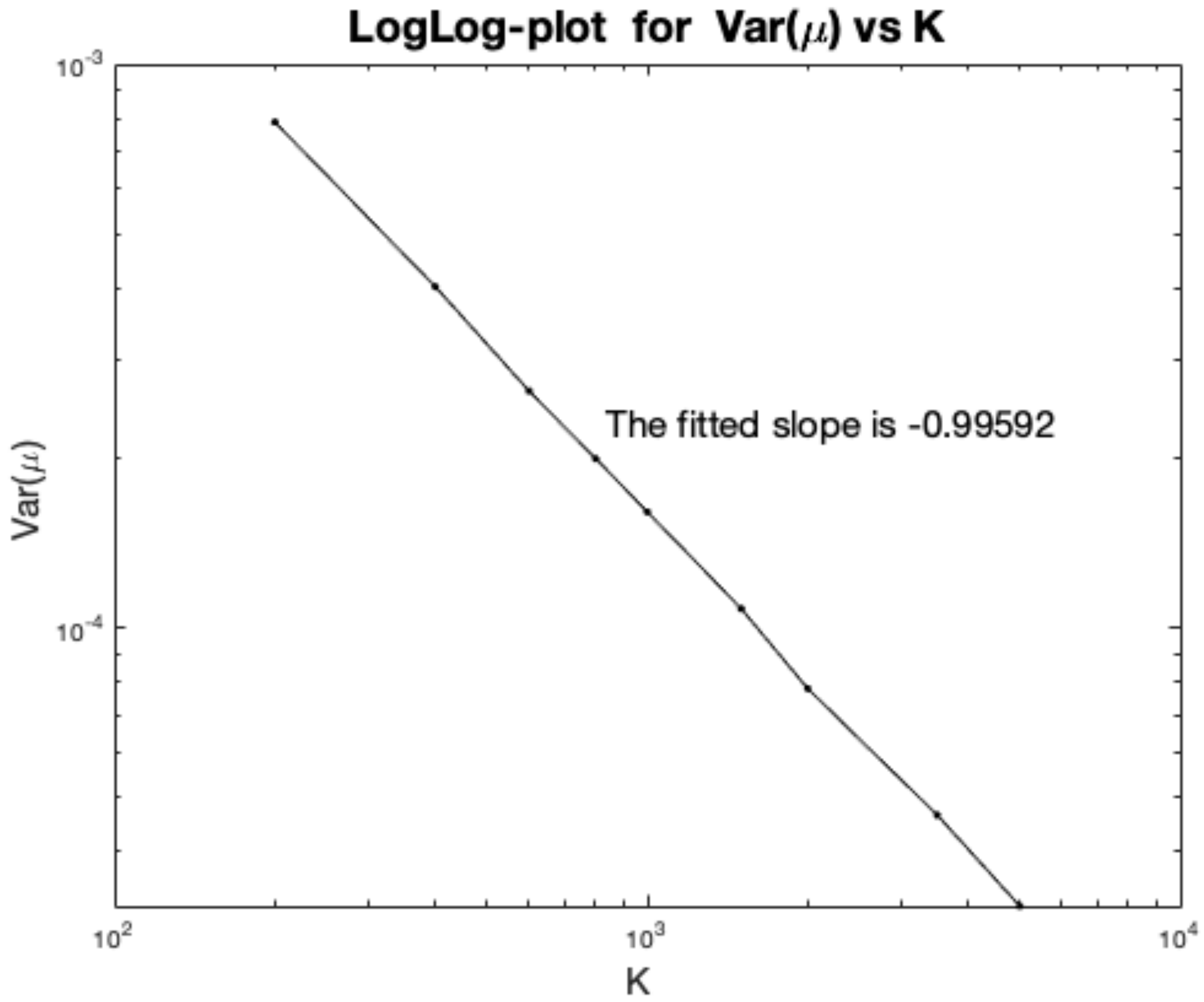
$$\log \mathrm{Var}(\mu) = \mathrm{a_1} \log(\mathrm{K}) + \mathrm{a_2}$$

We can fit the data with Log(input_Var) = a1*Log(K_list) + a2

```
% Plot Var(input) vs K in log-log scale
figure(3)
loglog(K_list, input_Var, 'k.-')
ylabel('Var(\mu) ', 'FontSize', 15)
xlabel('K  ', 'FontSize', 15)
title('LogLog-plot  for  Var(\mu) vs K', 'FontSize', 18)


fun = @(a,x) a(1)*x + a(2);
p = lsqcurvefit(fun,[-0.8, 0], log10(K_list),log10(input_Var));
disp(['The slope is ', num2str(p(1))])

txt = ['The fitted slope is ', num2str(p(1))];
text(mean(K_list)/2, mean(input_Var), txt, 'FontSize', 15)
```

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.
```

```
The slope is -0.99592
```



Prob 2.2 Excitatory & Inhibitory Synaptic Input, with the weight (1/K)

```
Fill in by yourself!!!
```

Prob 2.3 Excitatory & Inhibitory Synaptic Input, with weight 1/sqrt(K)

```
Fill in by yourself!!!
```

As you may notice, the variance is a constant regardless of K. How does the value of variance come? Check Eq. (1.11) in Week 3 notes.

Published with MATLAB® R2018a