In The Name of God

Negin Esmaeilzade 97104034

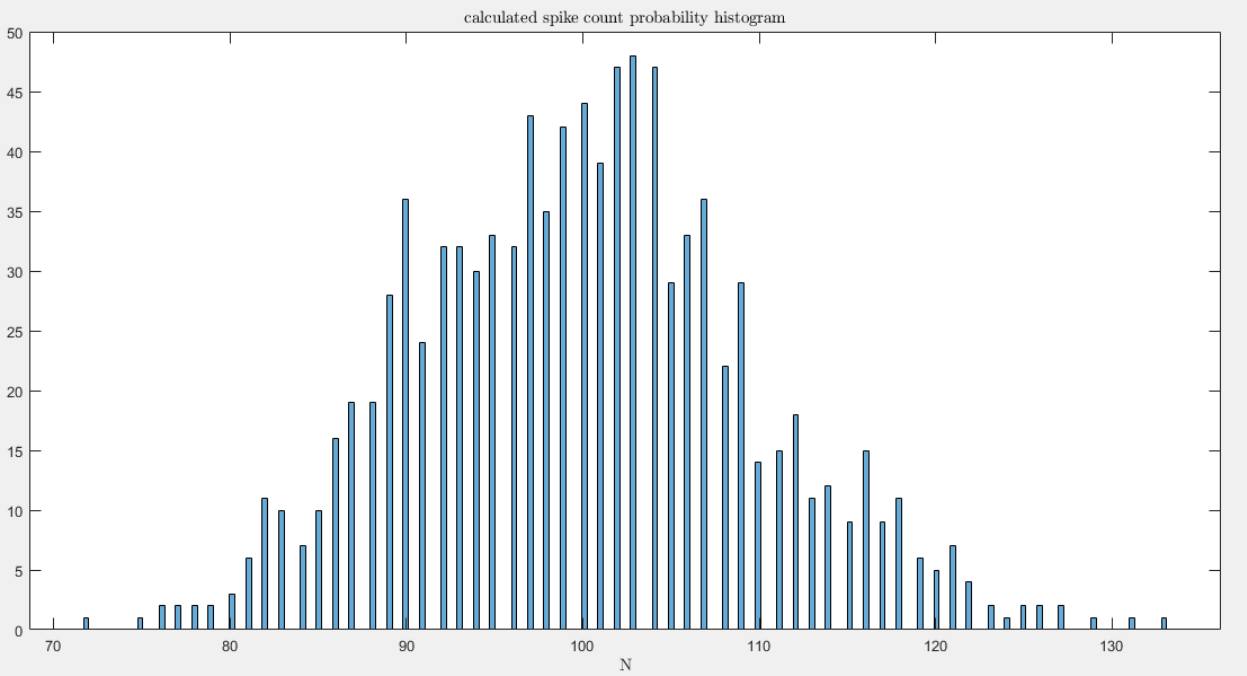
Advanced Neuroscience HW1

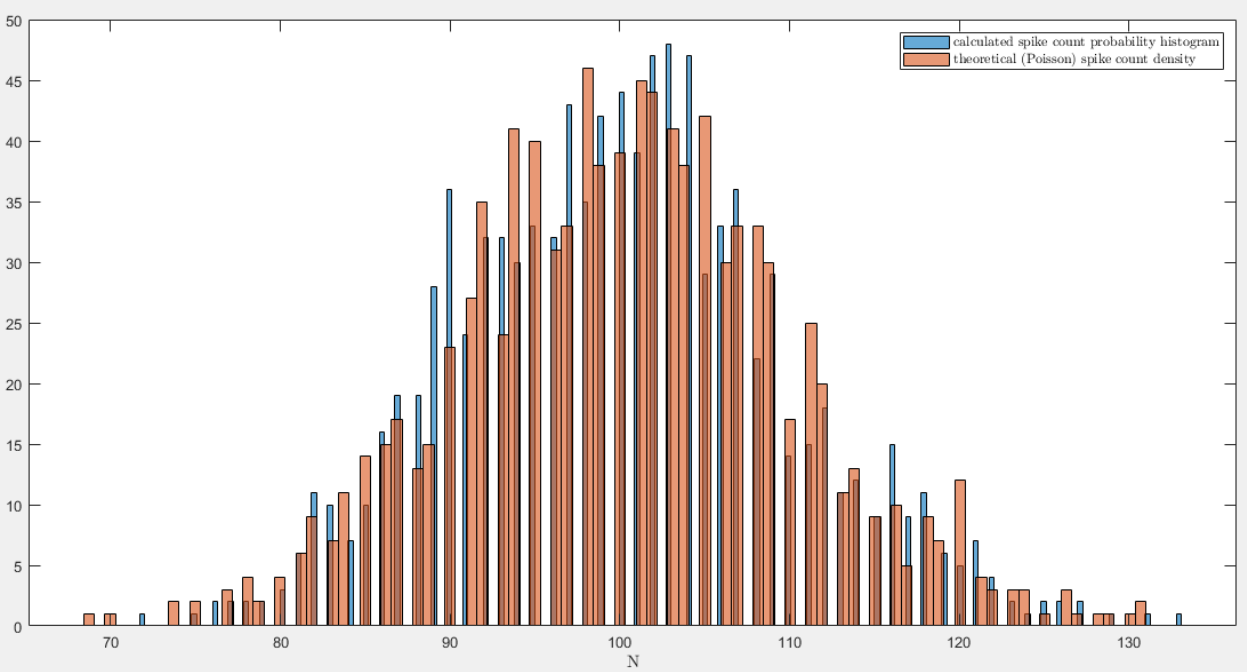
Dr. Ali Ghazizade

Integrate and Fire Neuron

A) In this part, 20 Poisson random process with $r = 100$, $\tau = 1\ ms$ were made each during $t_{sim} = 1s$ with the threshold method using uniform random process which was mentioned in the class. Here we can see the raster plot of these 20 processes:

B) To plot this histogram, 1000 trials are used. As we expected from theory the spike count density has Poisson distribution.





C) In this step, ISI's histogram is calculated considering all 100 trials. As we see the result is mostly like an exponential distribution. To make sure, the mean and the standard variation are also calculated for the result. We know for an exponential distribution:

$$f(x) = \lambda e^{-\lambda x} \rightarrow \quad \mu = \frac{1}{\lambda} \ , \sigma = \frac{1}{\lambda}$$

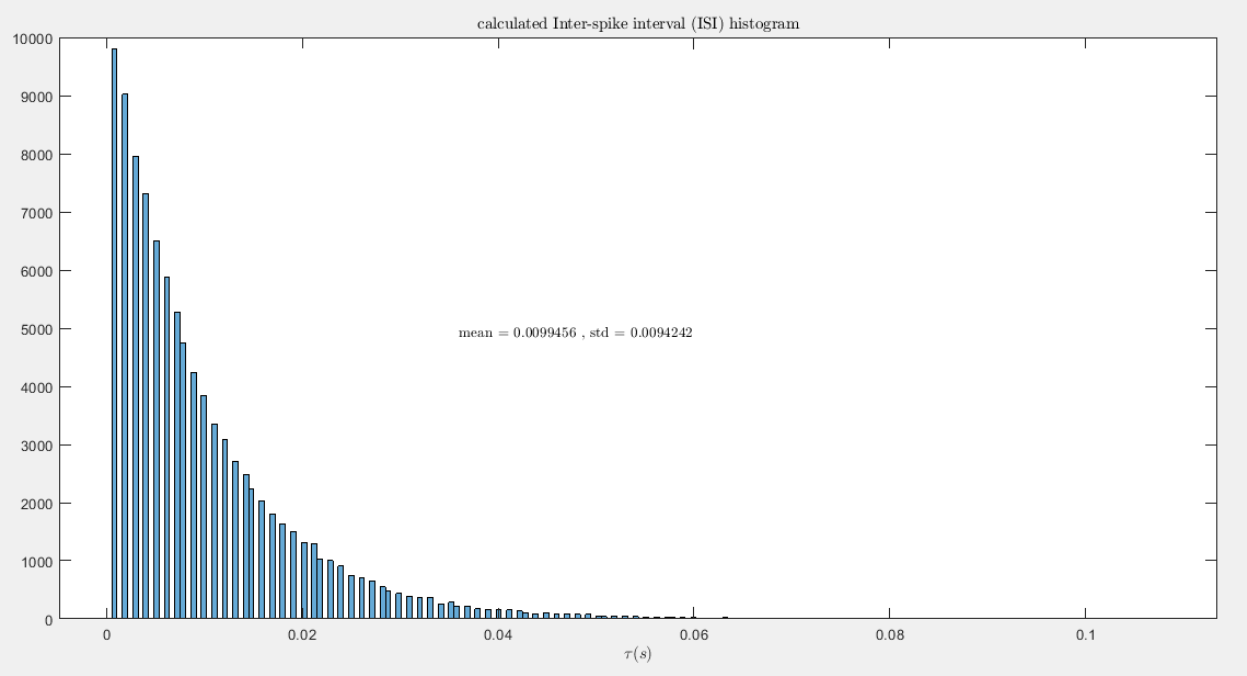
Here we have:

$$\lambda = 100 \rightarrow \ \mu \cong 0.01 \ , \sigma \cong \ 0.01$$

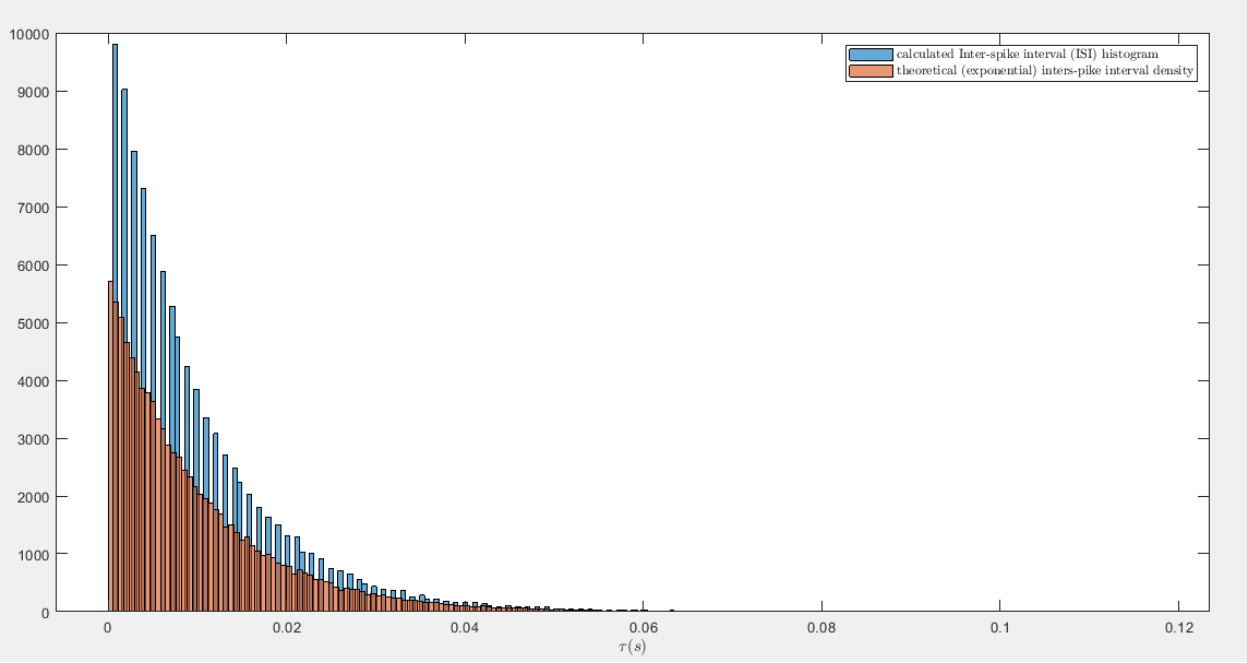
As we can see in the figure below, the calculated histograms' means and standard variations are:

$$\mu = 0.0099456 \ , \sigma = \ 0.0094242$$

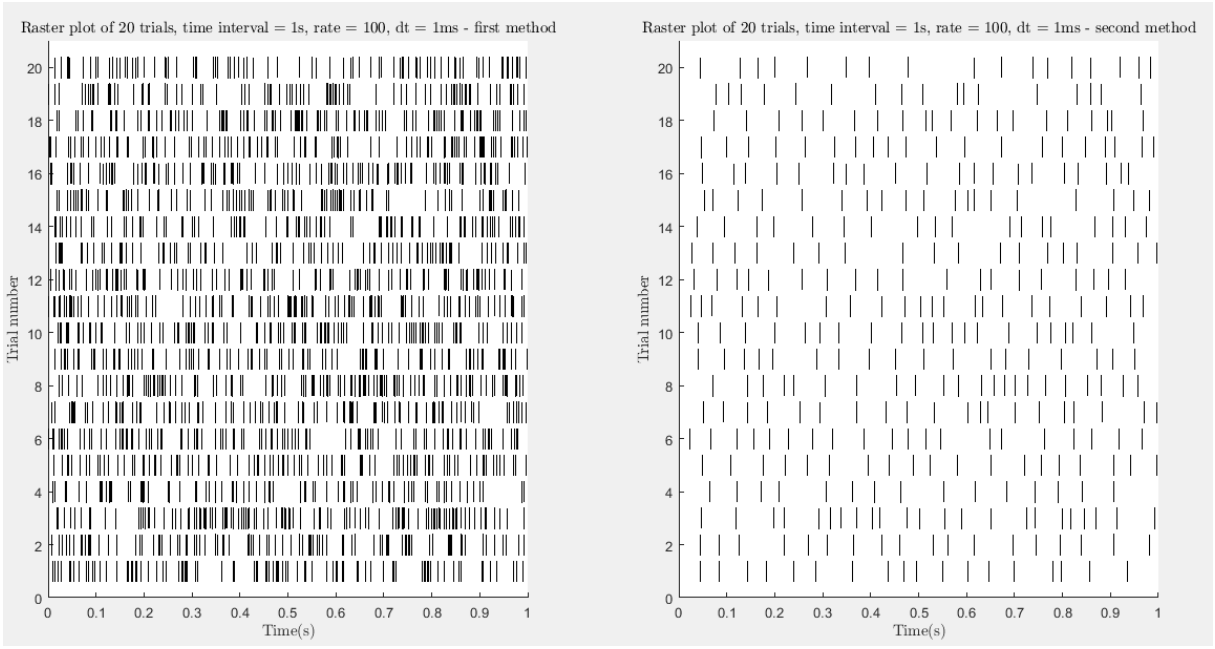
Which are almost about the amounts predicted by theory.



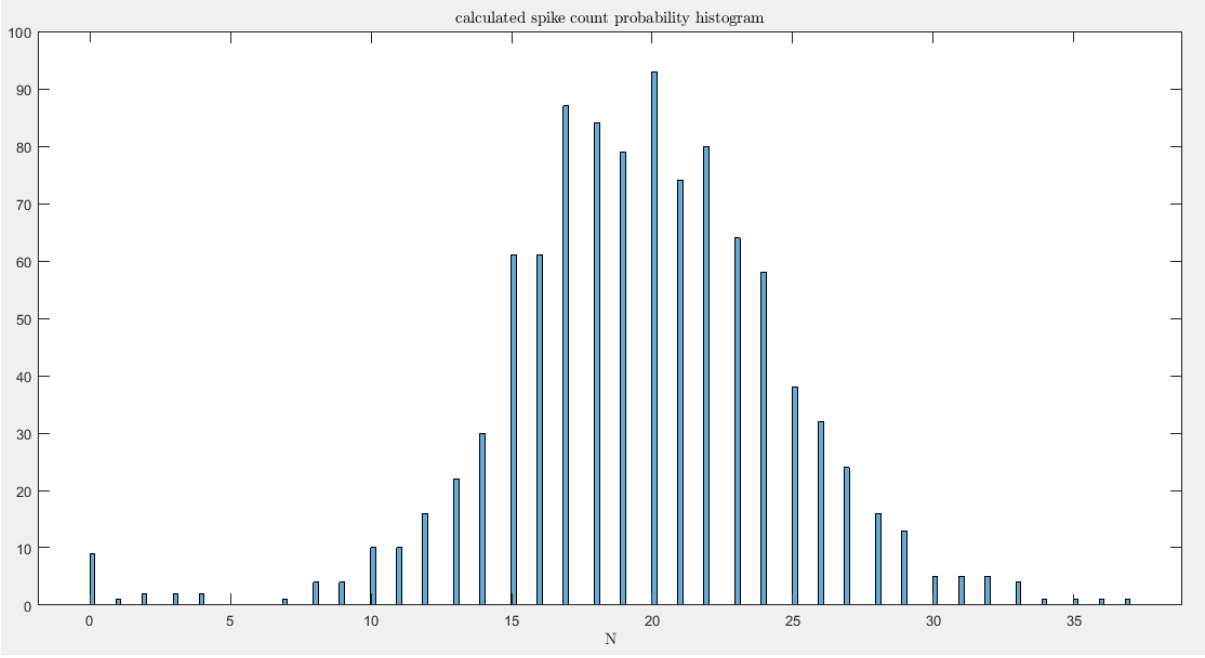
To compare the results better, the ISI histogram calculated from the simulated Poisson spike trains is plotted superimposed with the theoretical (exponential) inters-pike interval density and we can see that they are almost the same.

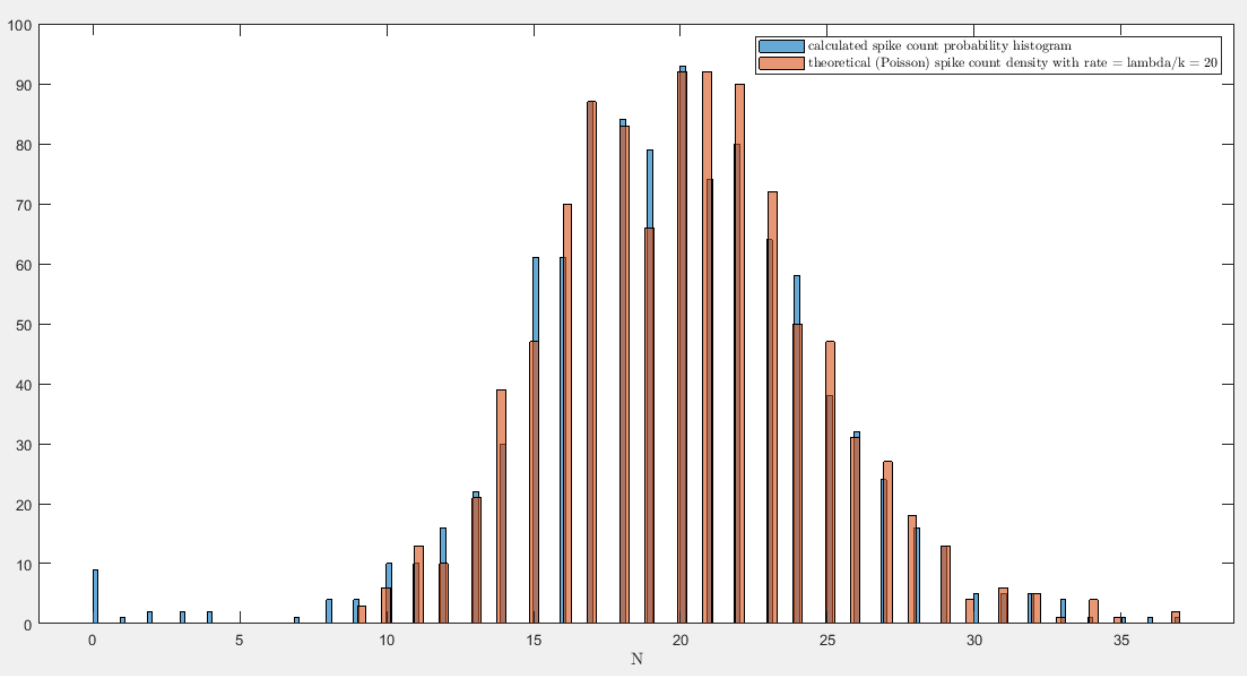


D) When we have a spike only after k spike, we actually have a spike counting system that fires only if can detect k spikes. This means that this system is calculating the input spike train's integration, evenly weighting the spikes, and whenever the integration grows higher than k it will fires and clears the memory. This is a way for modeling a neuron's firing with a system with some postsynaptic neurons' firings for input. Now, part A is repeated and here is the raster plot of 20 spike trains before and after using this kth-spike-detecting method. Here we assumed k = 5.

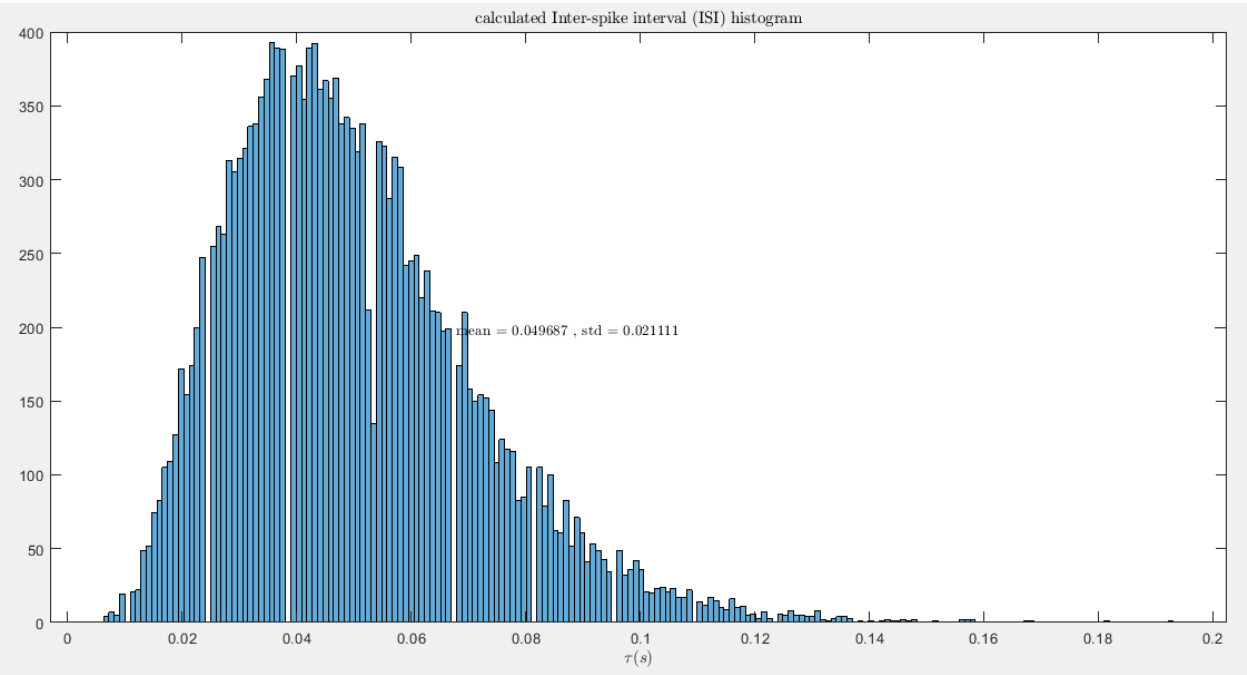


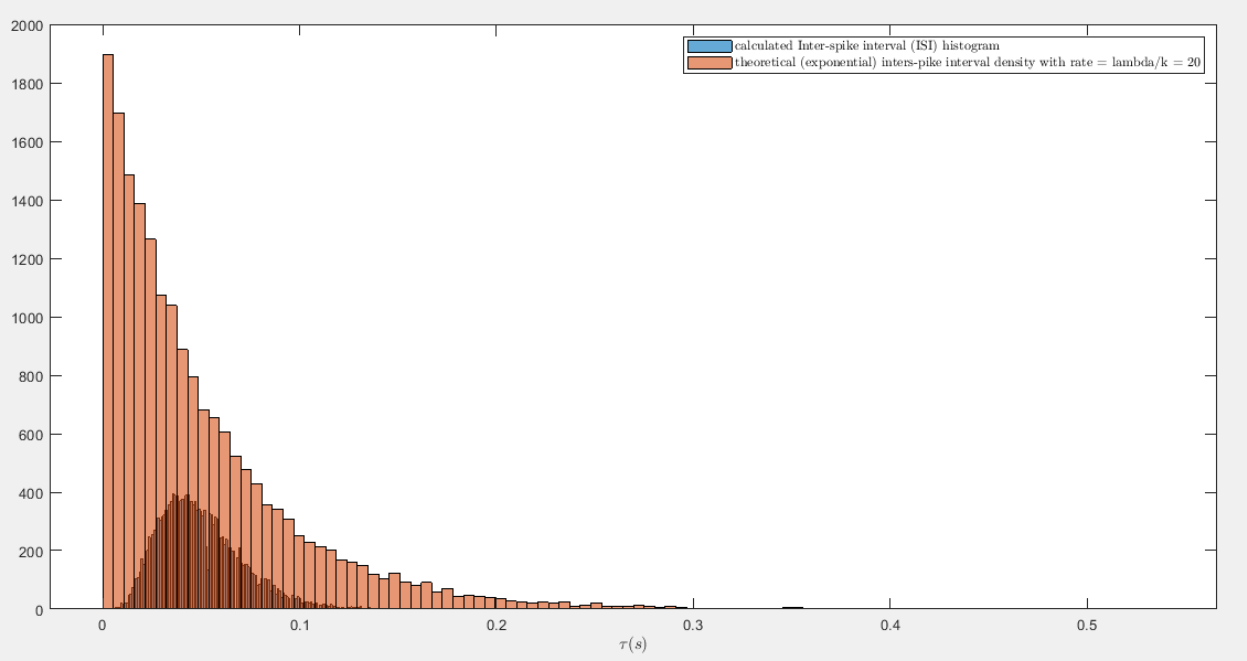
Part B is repeated with the newly reached data (the renewal spike trains) and here are the results for the spike count density distribution.





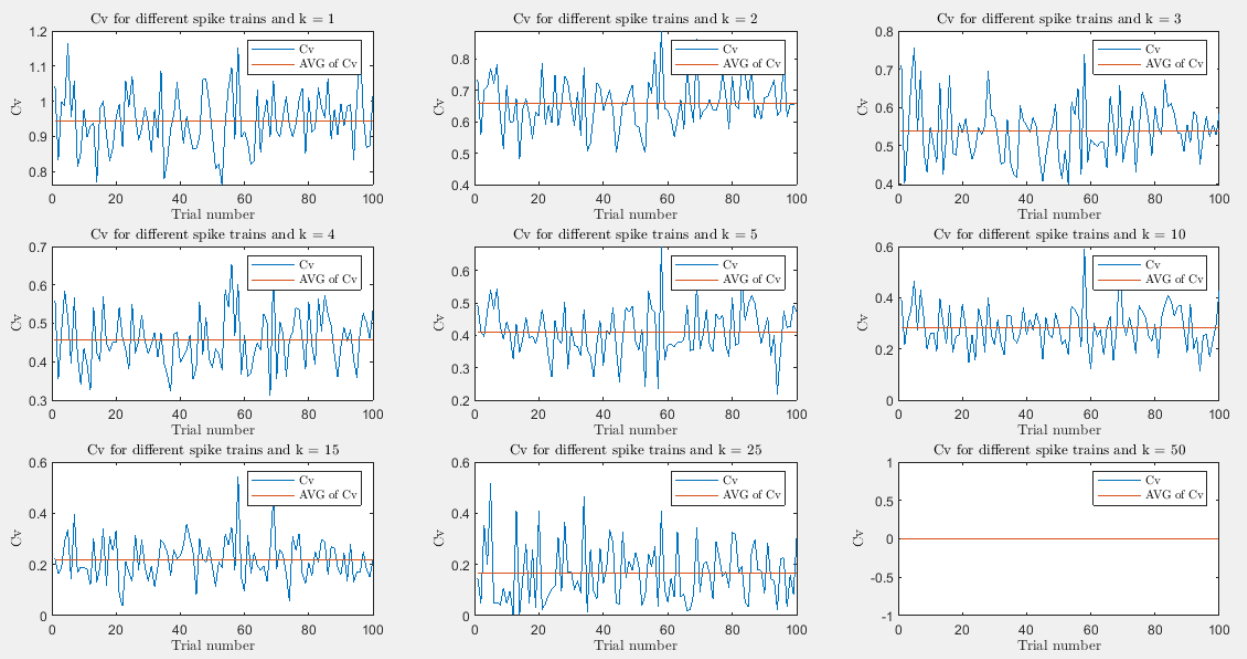
Part C is repeated with the newly reached data (the renewal spike trains) and here are the results for the ISI histogram.





According to the above results we can say that in renewal process the spike count density distribution is remained Poisson, however the ISI distributions also lead to Poisson and no longer exponential.

Now we try to compare the Cv parameter for the renewal process and Poisson process. We can expect that when k = 1, we have the main Poisson process rather than a renewal process. These figures are the Cv amounts for 1000 trials of a renewal process, plotted for k = 1, 2, 3, 4, 5, 10, 15, 25 and 50. In each plot the average Cv amount is shown by a red line.



As we see, when k = 1 we have the most amounts of Cv on average, which refers to the main Poisson process. Also, it can be seen that as k increases, the renewal processes have less amounts of Cv.

E) Consider X_i is an exponential random variable.

$$\tau = \sum_{i=1}^{k} X_i$$

We know that the sum of k exponential random variables with λ parameter leads to an erlang random variable with k and λ parameters. So, we have: $\tau \sim E(k, \lambda)$

$$f_x(k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \quad x, \lambda \geq 0$$

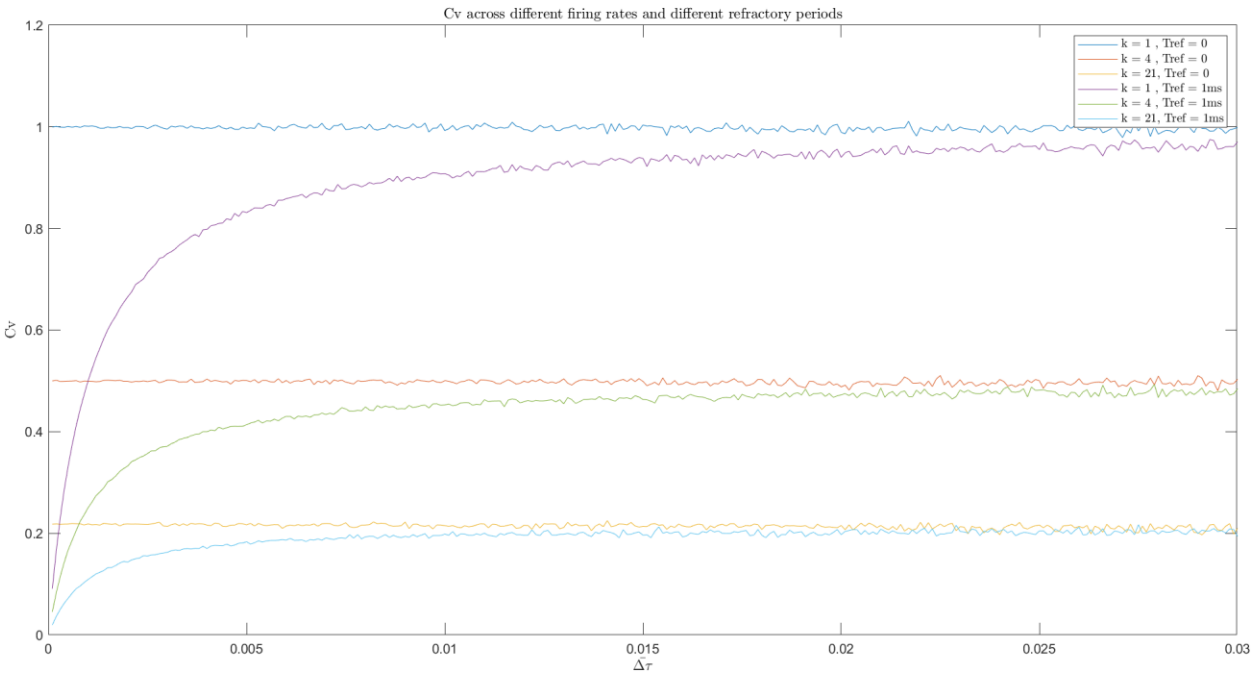
$$\rightarrow \quad E(\tau) = \frac{k}{\lambda} \quad , \qquad std(\tau) = \frac{\sqrt{k}}{\lambda}$$

$$\rightarrow \ C_v = \frac{std(\tau)}{E(\tau)} = \frac{\frac{\sqrt{k}}{\lambda}}{\frac{k}{\lambda}} = \frac{1}{\sqrt{k}}$$

F) The resulting ISIs and spike trains are similar to the real data which Softkey and Koch mentioned in their paper. However, the Cv amounts have a meaningful difference. For the real data Softkey and Koch reached to higher Cv amounts (most of the times more than 0.5) while we saw that as k increases, the Cv amount falls to zero in the way that even for $k > 4$, Cv amounts are almost always less than 0.5. Thus, it is concluded that simulating the neuron with this model lacks information and is not making enough variation same as a real neuron.

G) Contradicting with generated spike trains, real neural data contain less spikes within small intervals. The biological support for this claim is that each neuron has a refractory period which limit neuron to generate spikes within this period, so there is an upper limit for firing frequency in small intervals. Similar to equation [13] of paper, a refractory period is considered for the

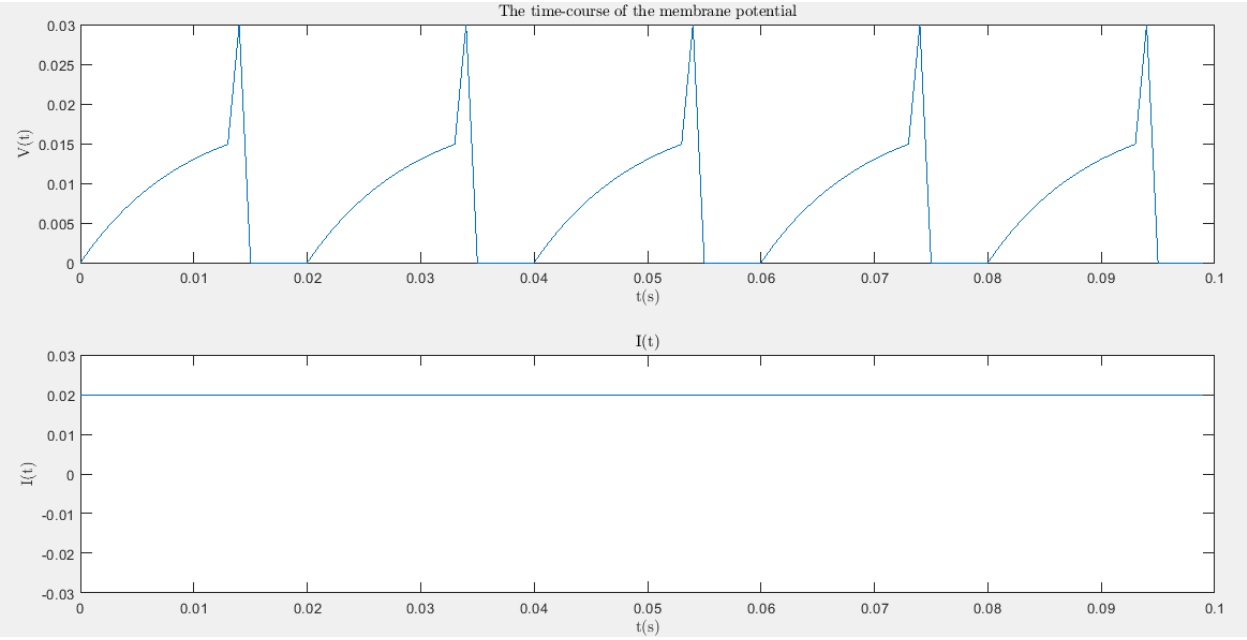
spike trains and similar to figure 6 of paper Cv amount is plotted across different firing rates and different refractory periods. Here is the result:



As we can see Cv amounts decrease (specially for less firing rates) when increasing refractory period and higher firing rates leads to higher Cv amounts until saturating.

Leaky Integrate and Fire Neuron

A) The time-course of the membrane potential is stimulated for $t_{sim} = 100\ ms$, considering a constant input current of $IR = 20\ mV$, resting potential of $v_r = 0\ mV$, and the threshold voltage of $v_{th} = 15\ mV$. Here is the result considering $\tau = 10ms,\ t_{ref} = 5ms$:



B) We know the LIF equation is:

$$\tau_m \frac{\partial v}{\partial t} = -v(t) + RI(t)$$

$I(t)$ is considered to be constant, so we actually have:

$$\tau_m \frac{\partial v}{\partial t} = -v(t) + RI$$

Let's find the answer of this equation. Using Laplace transform we have:

$$\tau_m V(S) + V(S) = RI \quad \rightarrow \quad V(S) = \frac{RI}{S + \frac{1}{\tau_m}}$$

$$\rightarrow \quad v(t) = RI(1 - e^{-\frac{t}{\tau_m}})$$

Know we should find the firing rate. The firing rate has an inverse relationship with the average time it takes for a neuron to fire. Thus, we only need to see how much time it takes on average for the membrane voltage to reach the threshold voltage, v_{th}.

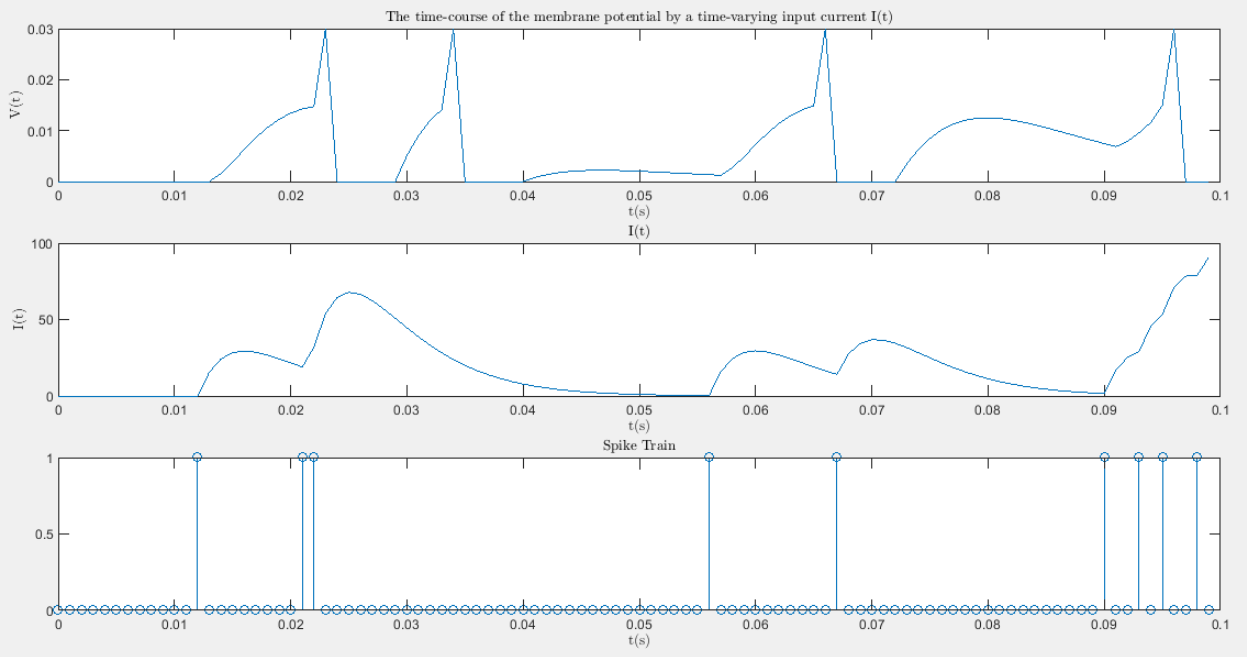
$$v(t) = v_{th} \rightarrow RI\left(1 - e^{-\frac{t}{\tau_m}}\right) = v_{th} \rightarrow e^{-\frac{t}{\tau_m}} = 1 - \frac{v_{th}}{RI}$$

$$\rightarrow t = -\tau_m \ln\left(1 - \frac{v_{th}}{RI}\right) = \tau_m \ln\left(\frac{RI}{RI - v_{th}}\right)$$

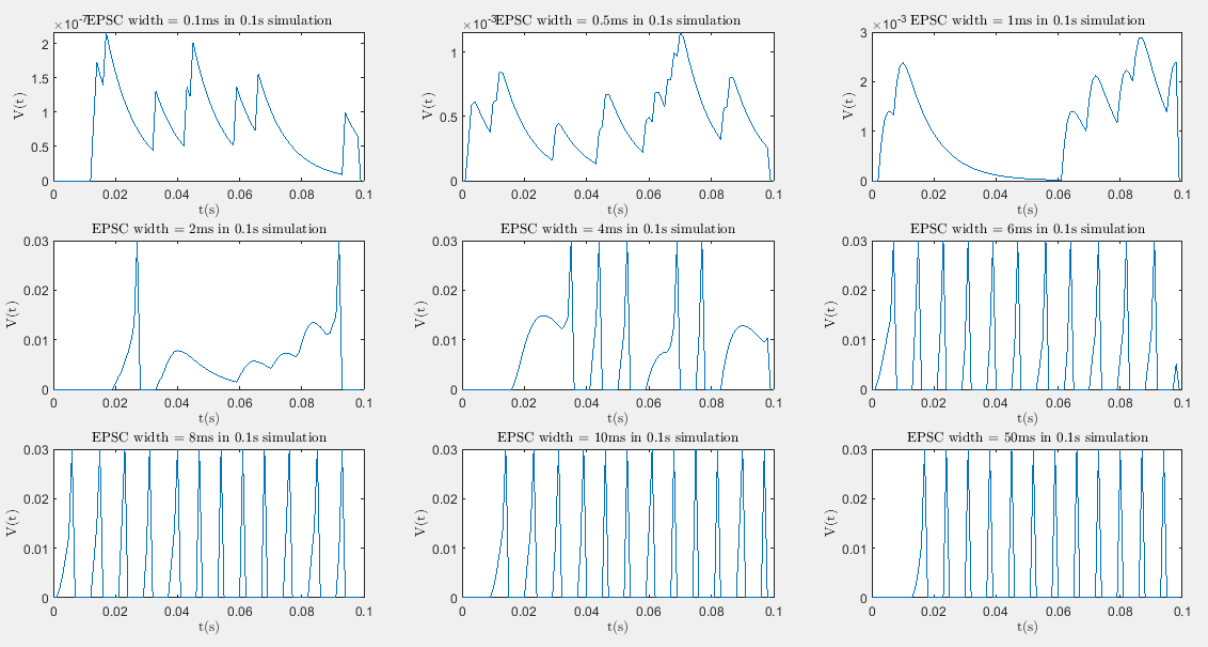
Considering refractory period of Δt_r , we have:

$$Average\ Firing\ Rate = r = \frac{1}{t + \Delta t_r} = \frac{1}{\tau_m \ln\left(\frac{RI}{RI - v_{th}}\right) + \Delta t_r}$$

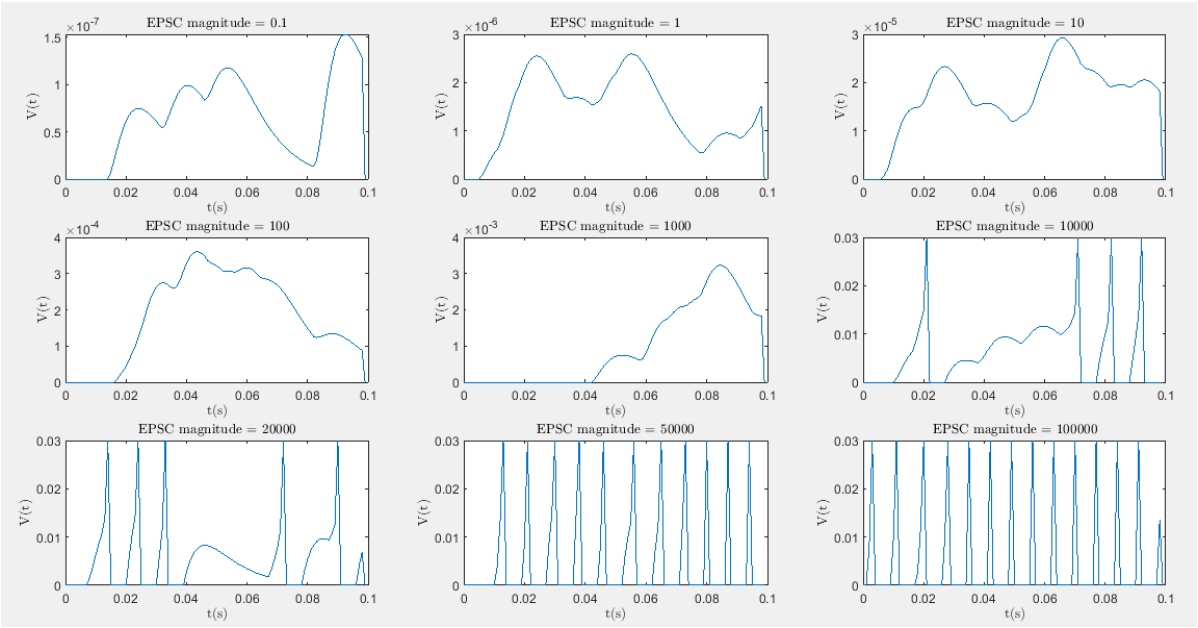
C) Section A is repeated, this time with a time-varying input current $I(t)$. To generate a realistic $I(t)$, a Poisson spike train is convolved with an EPSC kernel, similar to what Softky and Koch did in their paper. Here is the result of the spike train, the current $I(t)$, and the membrane potential plotted in a single figure considering $\tau = 10ms,\ t_{ref} = 5ms$:



In order to see the effect of EPSCs' width on the membrane potential, the addition graphs below are plotted. Here we can see the membrane potential considering the EPSCs' width = 0.1ms, 0.5ms, 1ms, 2ms, 4ms, 6ms, 8ms, 10ms and 50ms:

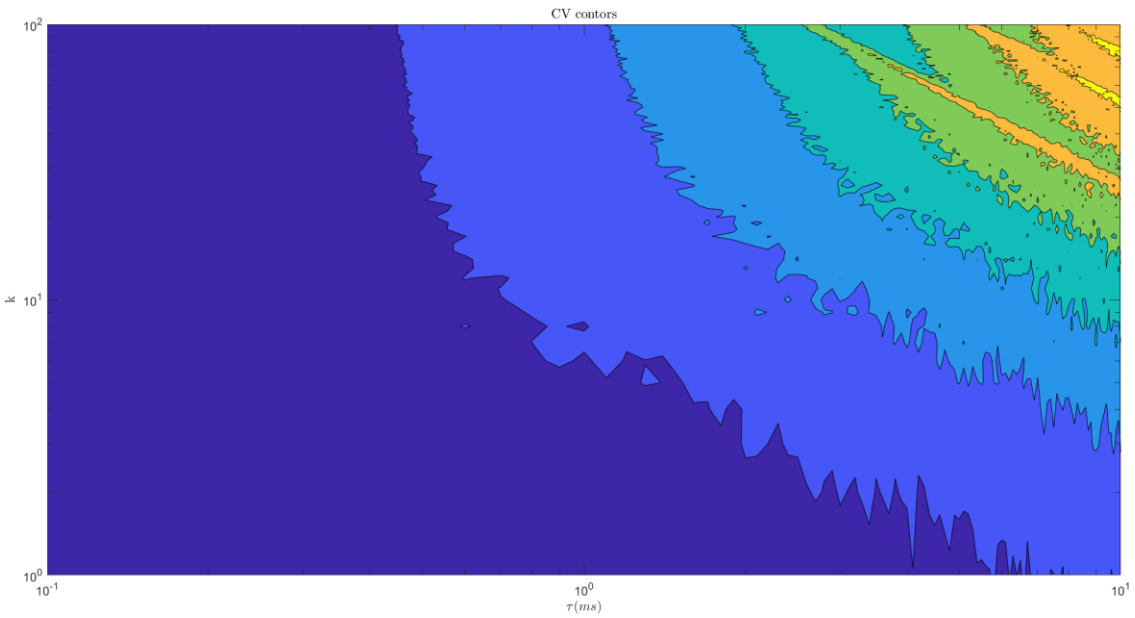


In order to see the effect of EPSCs' magnitude on the membrane potential, the addition graphs below are plotted. Here we can see the membrane potential considering the EPSCs' magnitude = 0.1, 1, 10, 100, 1000, 10000, 20000, 50000 and 100000:



As it is shown in the figures above, the number of spikes of a neuron, during a specific time, is increased both when the EPSCs' width or EPSCs' magnitude increases.

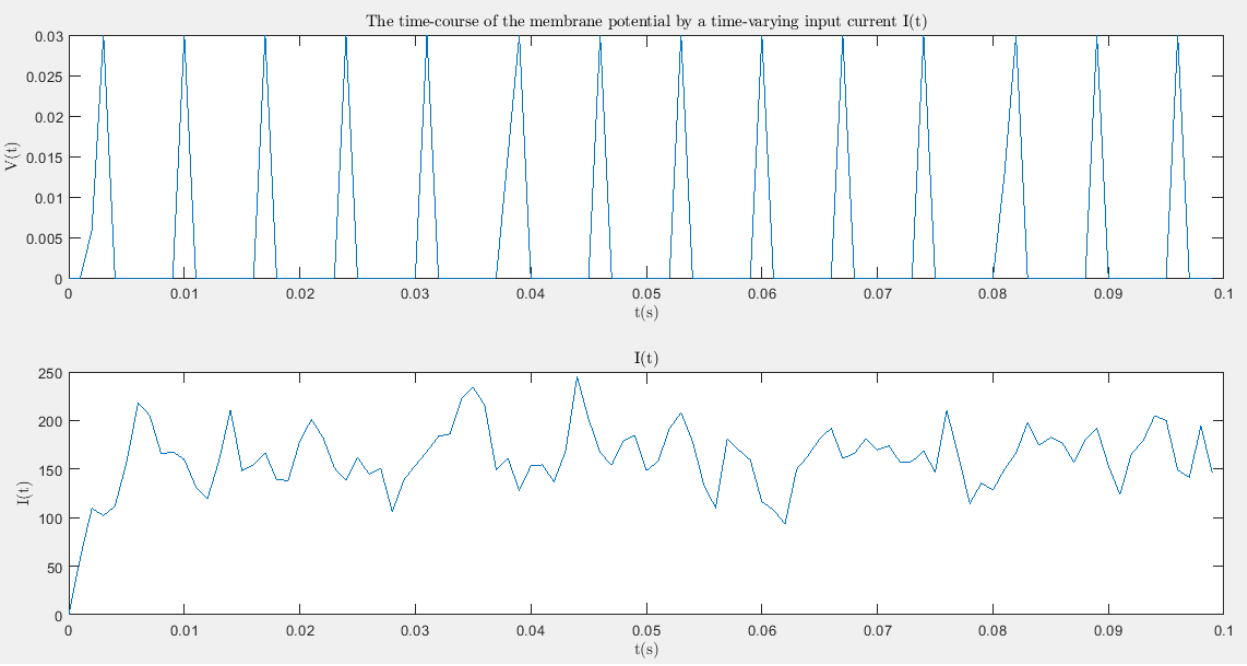
Now we try to generate a contour plot similar to figure 8 of paper which shows the CV contours in terms of τ and k and here is the result:



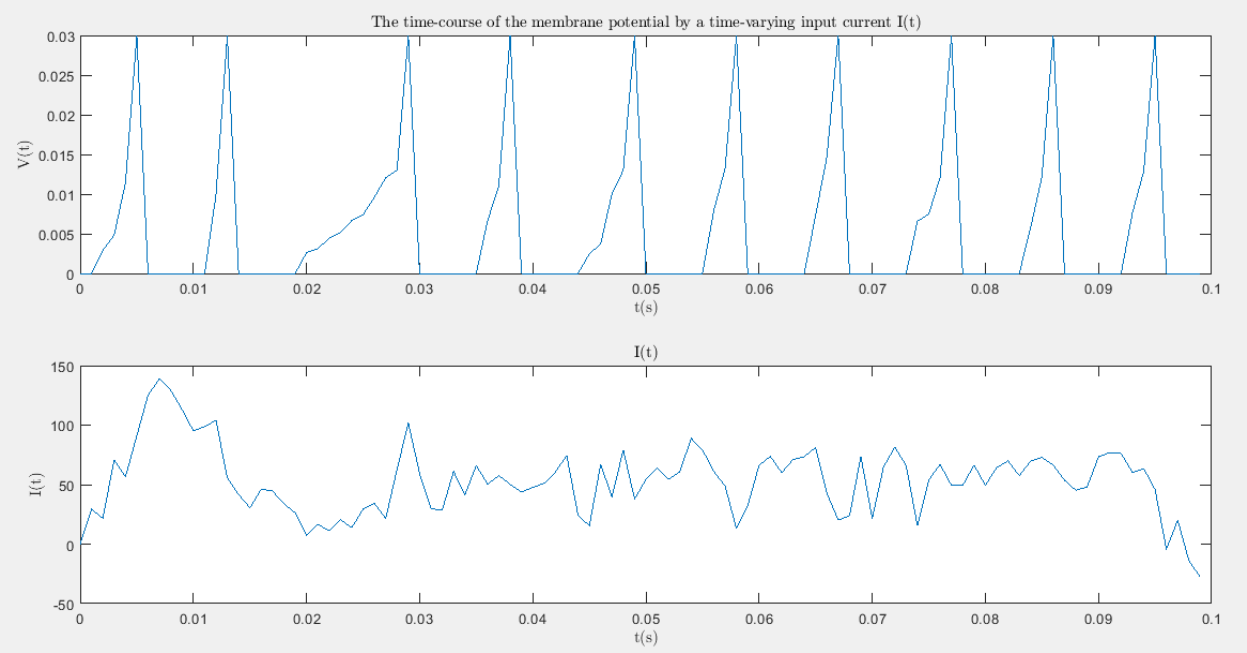
Although I made a lot of effort, the figure is not even near to the paper's figure!!

D) In this section, a percentage of synaptic inputs are selected to be inhibitory and assigned negative kernels. Then section C is repeated with different percentage of inhibitory synaptic inputs (p = 10%, 20%, 40%, 45%, 50%, 70%). Here are the results:

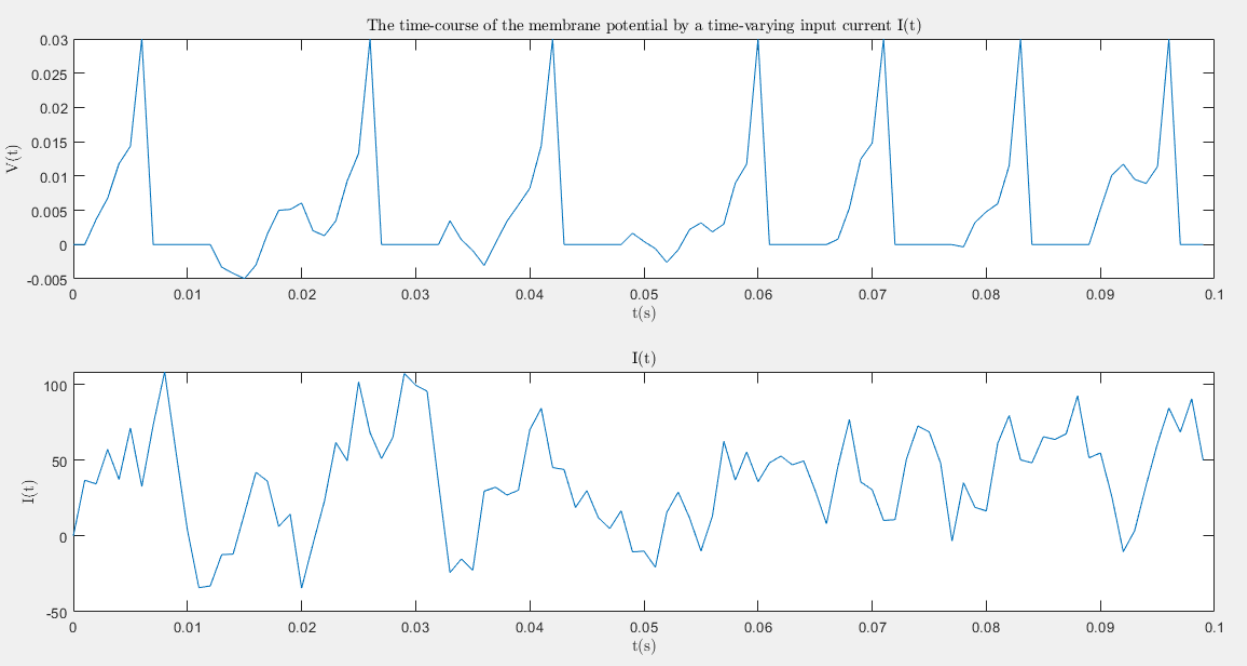
P = 10%



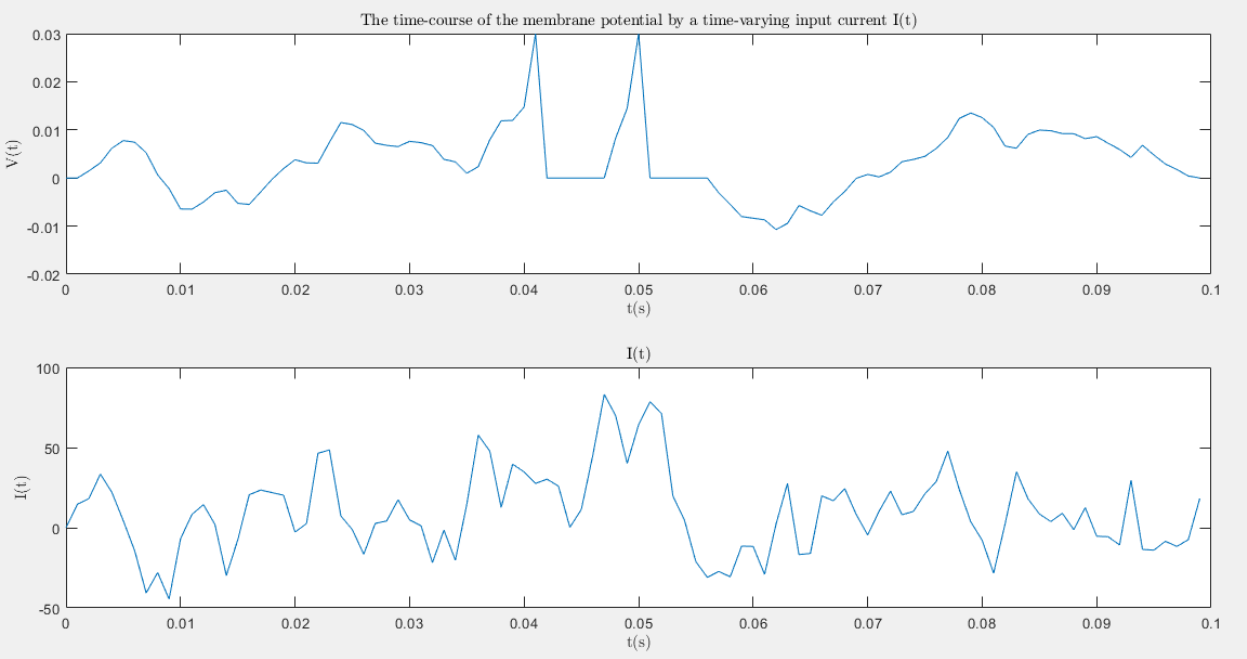
P = 30%


The time-course of the membrane potential by a time-varying input current I(t)
V(t)
t(s)
I(t)
I(t)
t(s)

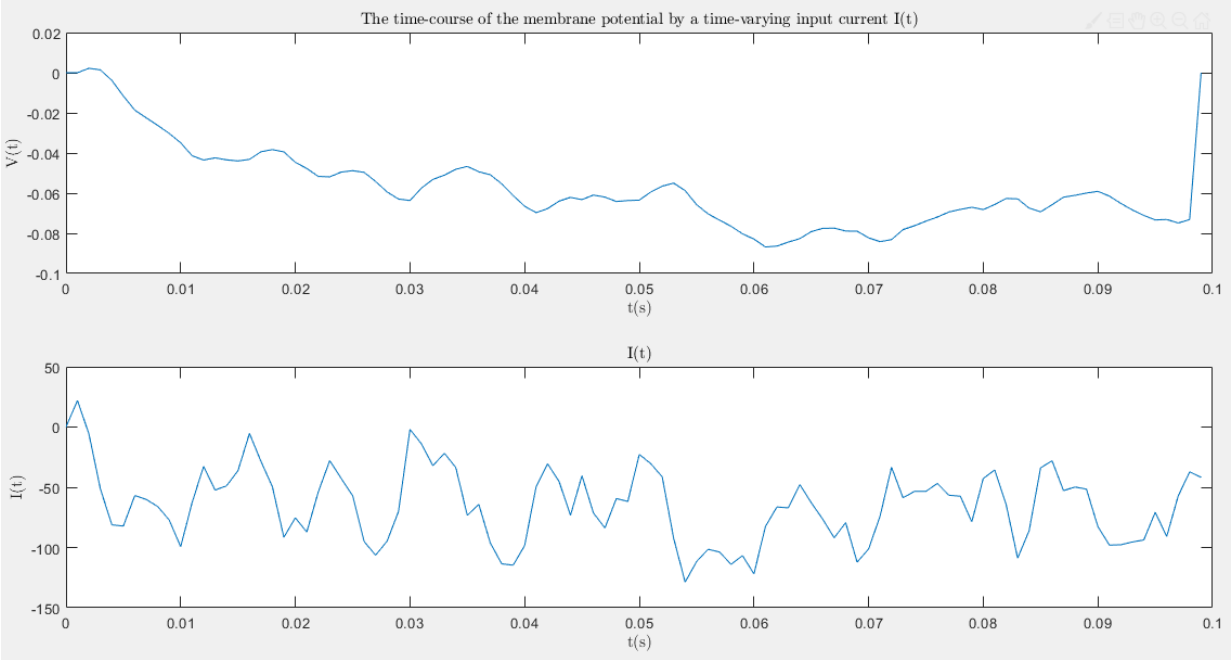
P = 45%


The time-course of the membrane potential by a time-varying input current I(t)
V(t)
t(s)
I(t)
I(t)
t(s)

P = 50%

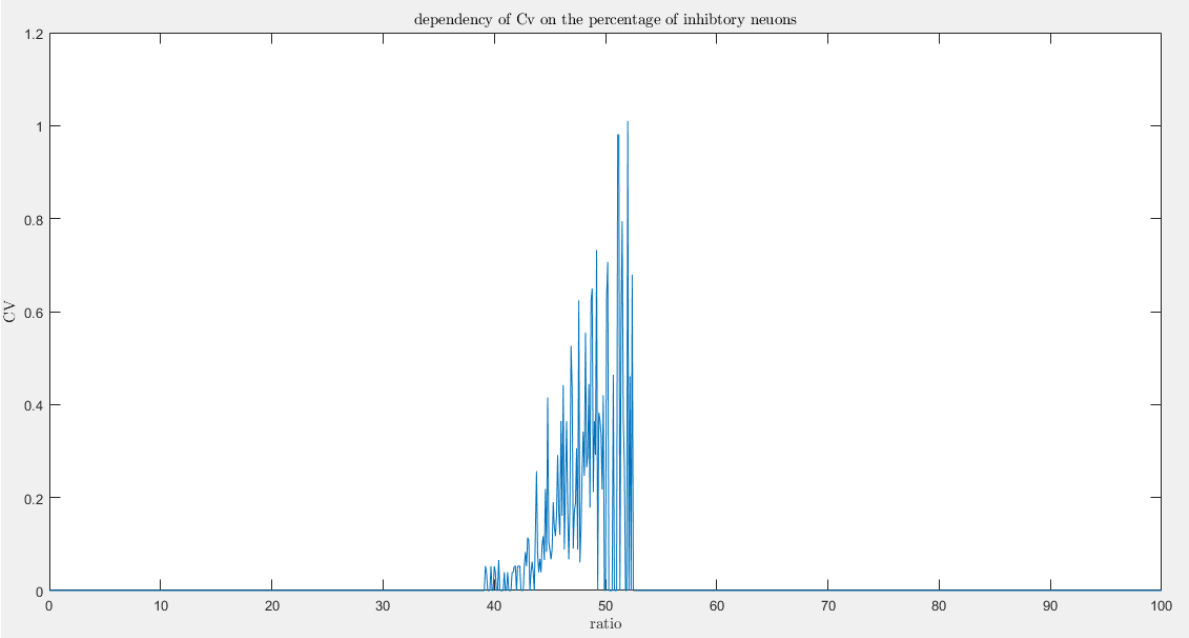


P = 70%



It can be seen that the number of spikes decrees in a specific time period, when the ratio of inhibitory synaptic inputs increases.

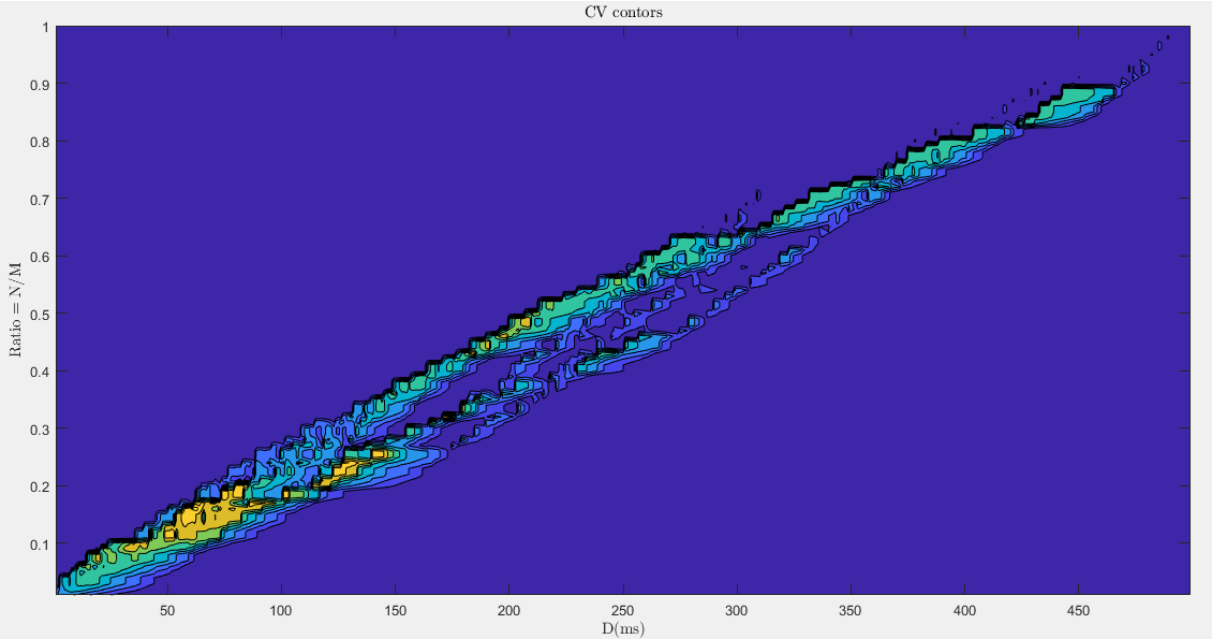
Now, we plot the Cv in terms of the inhibitory inputs' ratio:



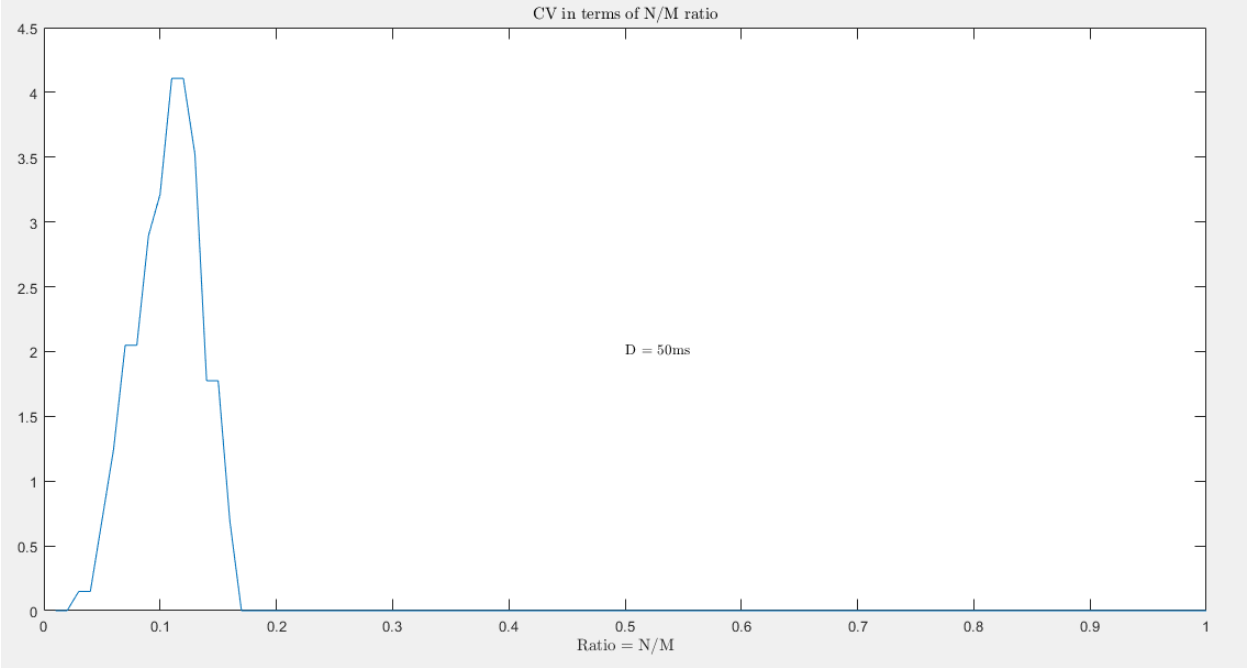
We can see that the maximum Cv is almost when 50% inputs are inhibitory. This points to the

fact that adding inhibitory inputs leads to the increase of Cv and consequently is a better model for real neurons' data.

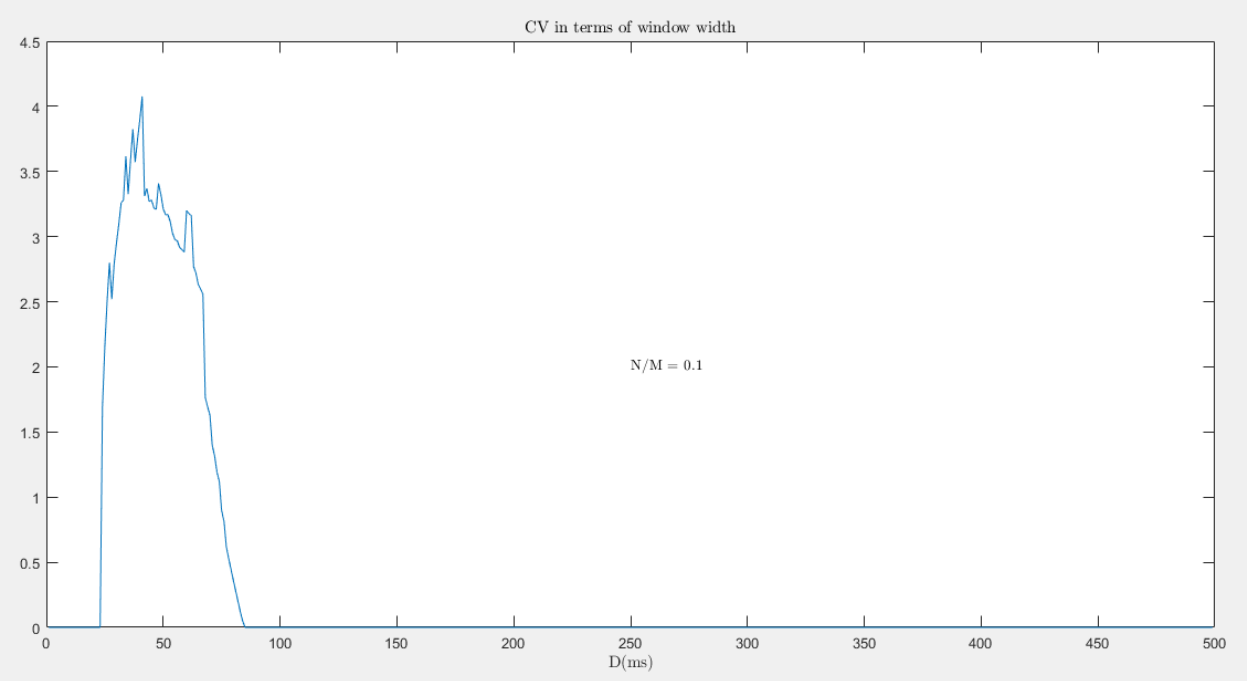
E) Assume the neuron is doing coincidence detection of it excitatory inputs which have a Poisson distribution and neuron requires N out of M inputs to be active in a short D ms time window. Here is the Cv contour in terms of N/M and D for this simulation:



Considering D = 50ms and plotting the Cv in terms of N/M ratio we have:



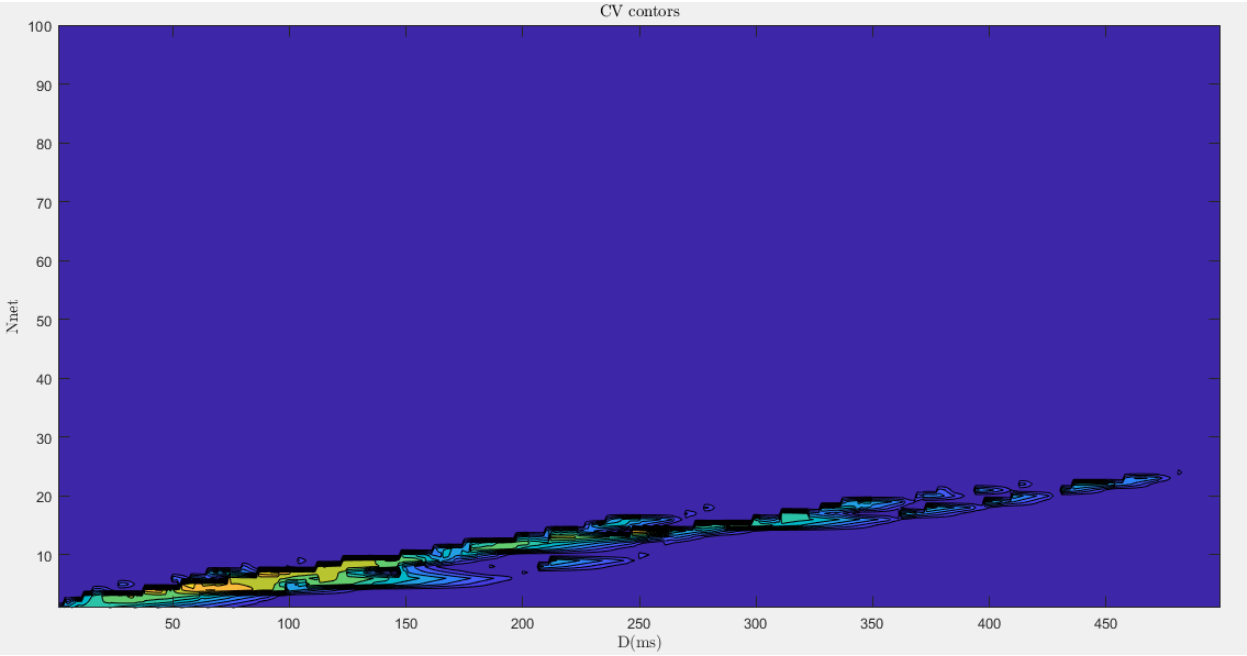
Considering N/M = 0.1 and plotting the Cv in terms of D we have:



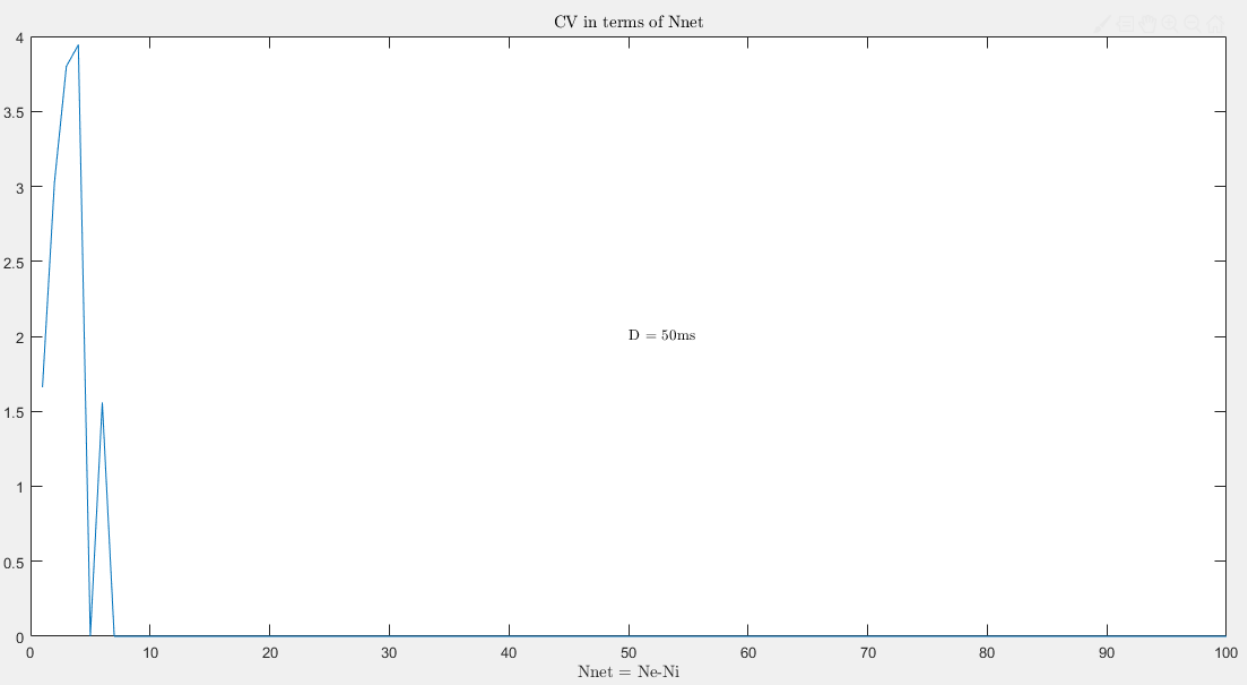
As we can see the Cv amount increases both when the window width (D) or spike detection threshold (N/M) increase. However, the relation is not always ascending. In fact, when the

window width increases a lot, all spikes can be detected, so it will be like when D is near to zero. This is also the same for the threshold.

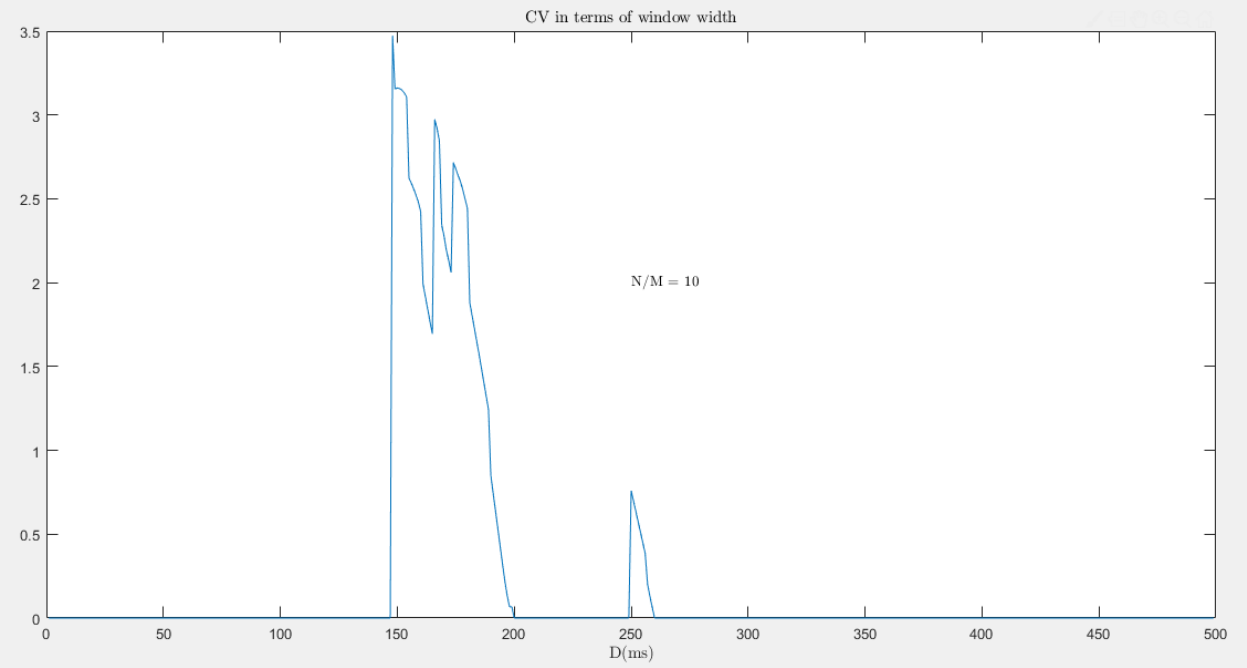
F) Now, inhibitory neurons are added to the simulation. all other conditions are similar to previous part, except the inhibitory inputs percentage which is set to 45%. Here is the Cv contour in terms of the threshold (Nnet) and the window width (D) for this simulation:



Considering D = 50ms and plotting the Cv in terms of Nnet threshold we have:



Considering Nnet = 10 and plotting the Cv in terms of D we have:



The dependences are almost like the previous section, but adding inhibitory inputs caused the neuron to have higher amounts of Cv rather than before and this is better modeling a real neuron.