Sharif University of Technology
Electrical Engineering School

Advanced Neuroscience HW1
Based on Softky & Koch, 1993

NEURON SPIKING SIMULATIONS

Armin Panjehpour
arminpp1379@gmail.com

Supervisor(s): Dr. Ghazizadeh
Sharif University, Tehran, Iran

04/03/2022

Part.1 - Integrate & Fire Model

First, we will do some simulations on a Poisson spike train:

Part.1.a - Poisson process A Poisson processed spike train has a Poisson distribution for its spike count and its inter-spike intervals (ISI) follows an exponential distribution. So, to create a spike train, we will obtain ISIs using an exponential distribution with the rate of 100. In order to do this, I've used "exprnd" function of MATLAB. Here's the spike train for a trial of 1 second:

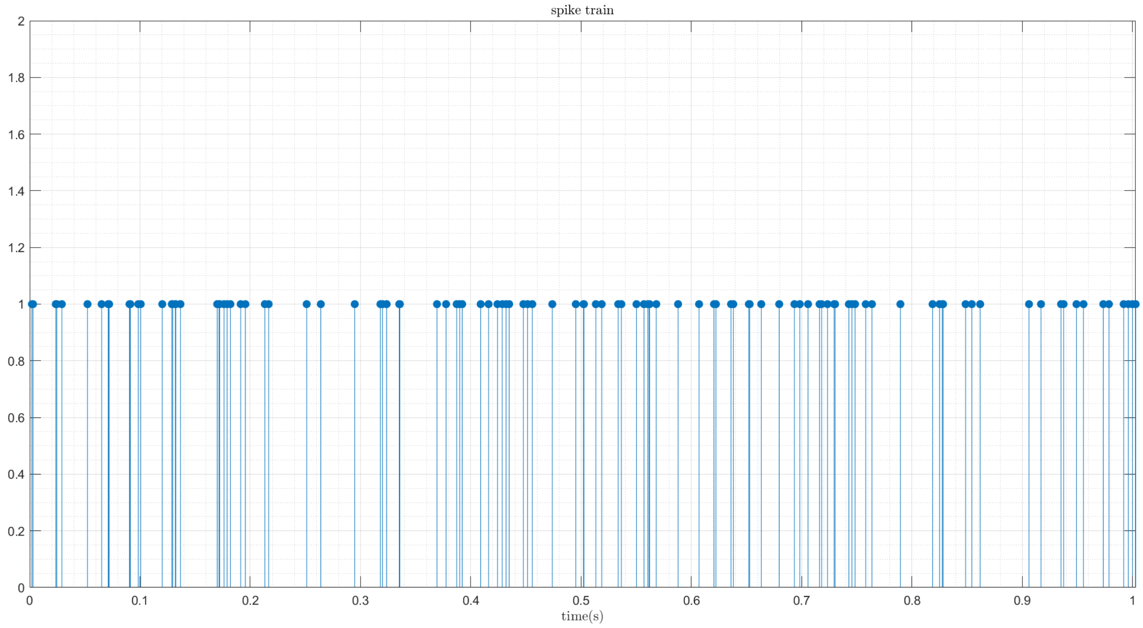


Figure 1: Poisson Spike Train

Part.1.b - Spike Count Distribution Spike count probability distribution histogram is calculated on 1000 trials each 1 second. At the end, a theorical Poisson PMF is plotted on the histogram which indicates the Poisson distribution of spike counts:

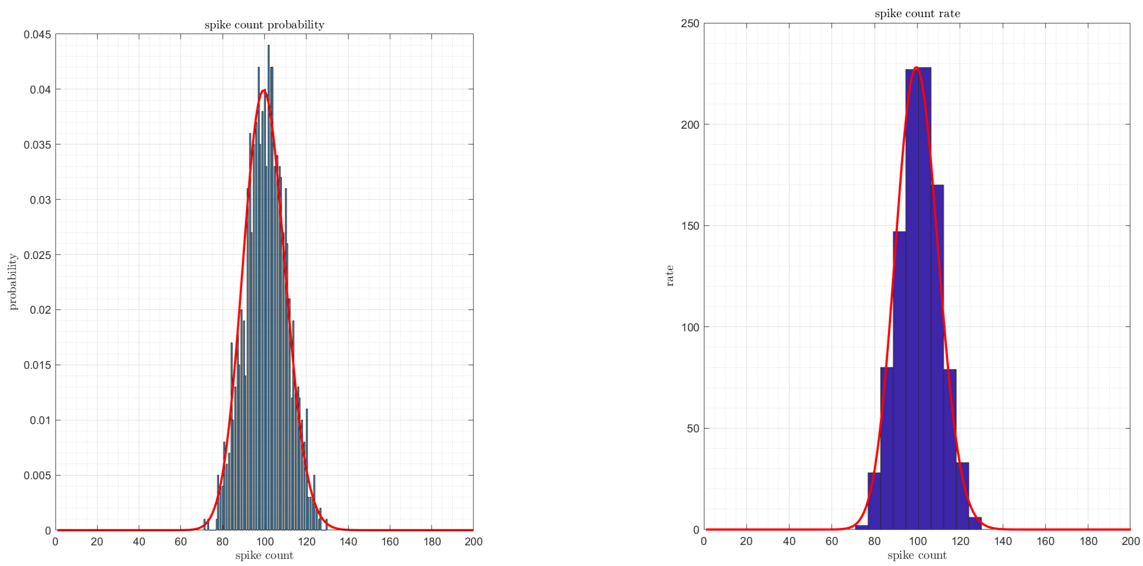


Figure 2: Spike Count Probability Distribution | | Spike Count Rate Distribution

Part.1.c - ISI Distribution In this part, we calculate time between each two spikes and we would have a vector of this time which is called ISI. We do this for the spike train in part.1.a and here's the distribution of these ISIs with a theorical exponential PDF plotted on it:

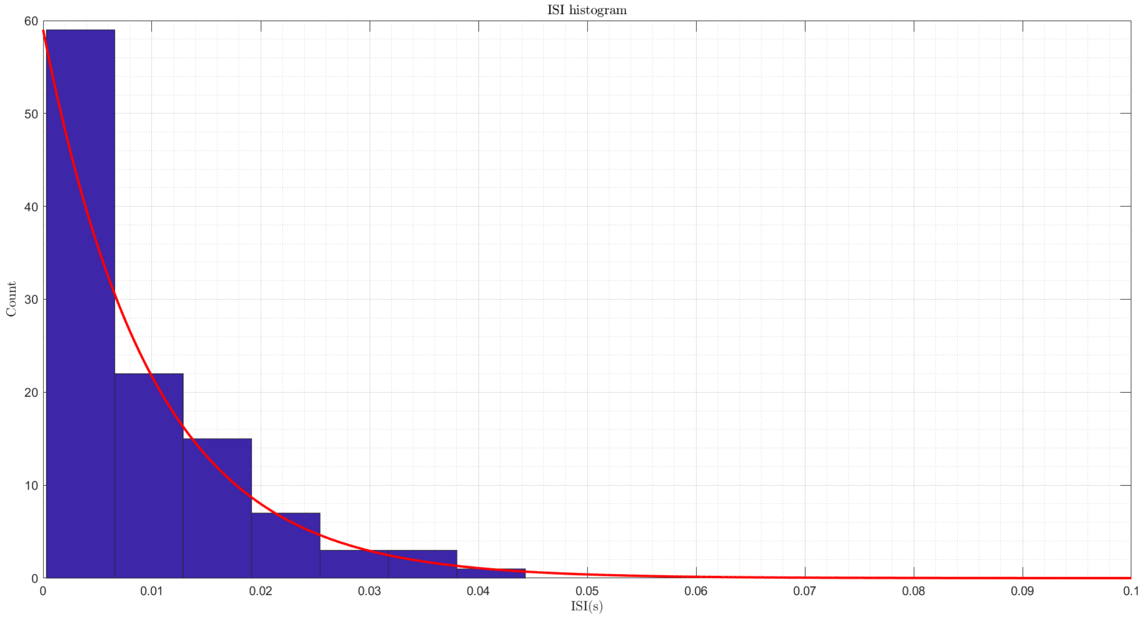


Figure 3: ISI Distribution

The distribution of ISIs follows an exponential distribution as expected.

Part.1.c - d - Integration over inputs.a We first, create a Poisson spike train and remove all spikes but every kth spike. This procedure is similar to integration over pre-synaptic inputs of a neuron which they have an ISI distribution of Poisson. Here we assume k to be 5. Here's the resulted spike train before and after removing operation (using down sample):

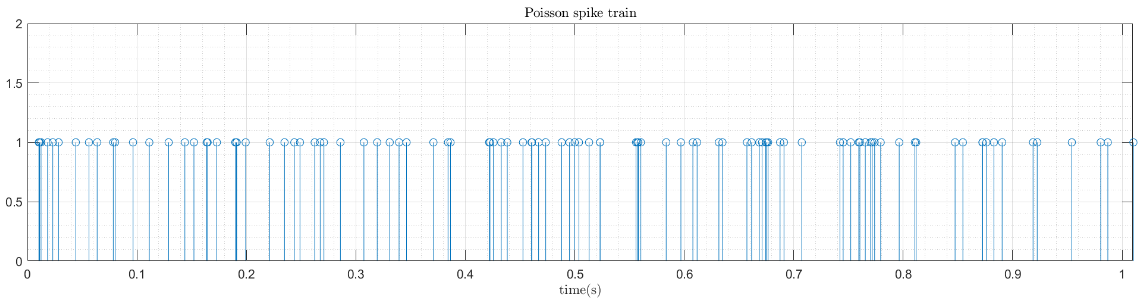


Figure 4: Initial Spike Train

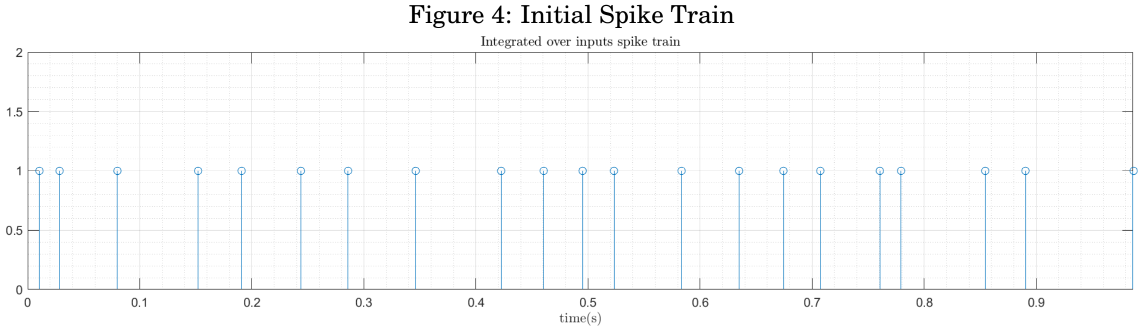


Figure 5: Down Sampled Spike Train

Part.1.c - d - Integration over inputs.b Here we repeat part 1.b for the resulted spike train for 4 different "k"s. You can see the spike count probability histogram below:

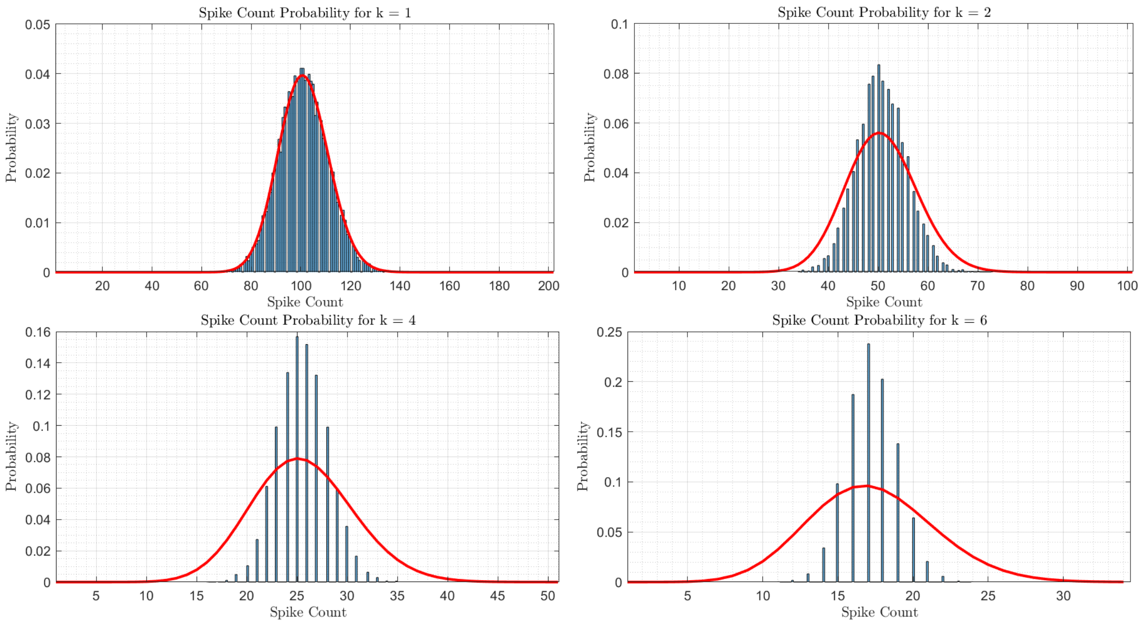


Figure 6: Spike Count Probability Distribution for different "k"s

It seems that by increasing k, the spike count probability distribution will move away from a Poisson distribution as the ISI histogram in the next part move away from exponential too:

Part.1.c - d - Integration over inputs.c ISI distribution histogram for spike train in part 1:

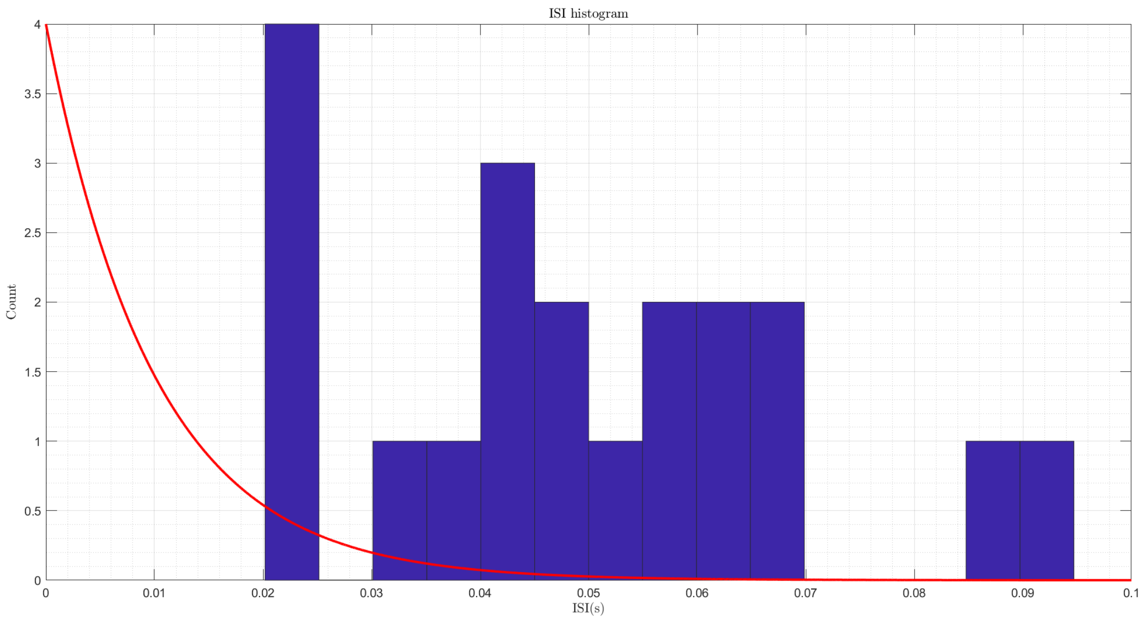


Figure 7: ISI Distribution

Part.1.d - Coefficient of Variation We calculate coefficient of variation (CV) for 100 simulations for two different spike trains and here's the result: As you can see, in all of the trials (100%), the CV of Poisson process is closer to 1 than the other spike train which shows we have more variability in

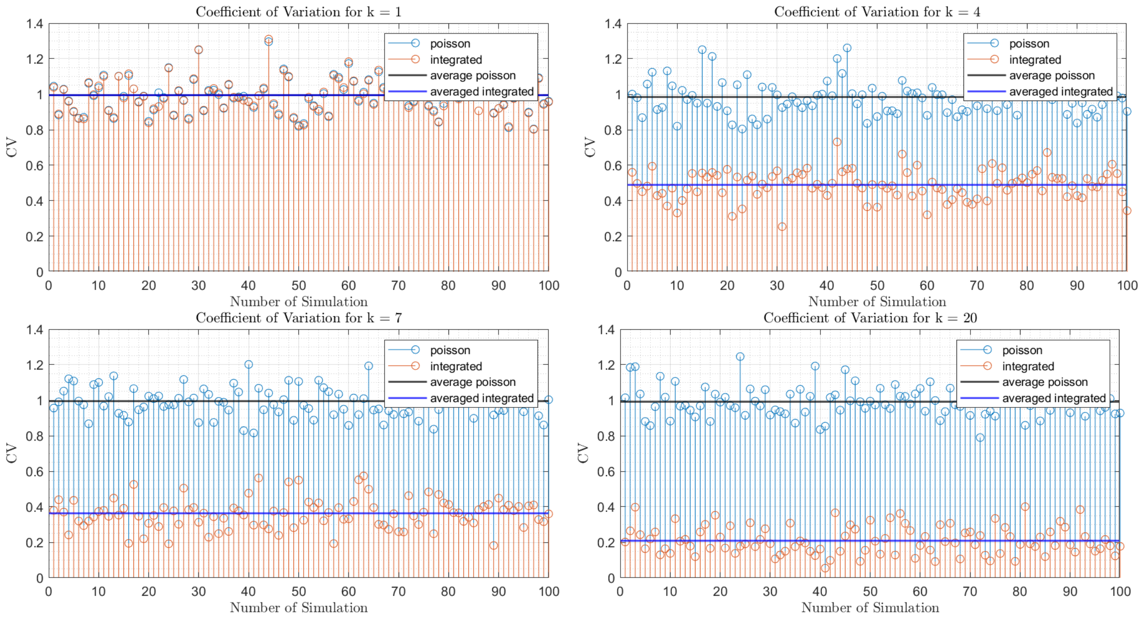


Figure 8: CV of Poisson process and Integrated inputs for different "K"s

Poisson process than when we integrate over the inputs. So, the conclusion is that, integration over inputs will decrease CV and variability as we expected. As you can see, for k = 1, the CVs for two spike trains are the same. Another result that we can get is that by increasing k, CV will decrease which is matched to the results of the paper:

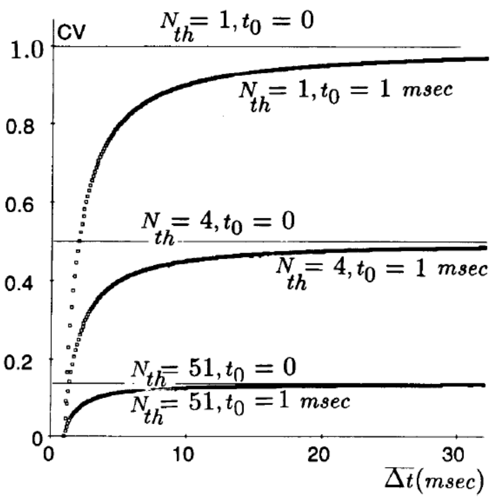


Figure 9: CV Integrated inputs for different "k"s

Part.1.e - Coefficient of Variation Assuming we need K Poisson EPSPs in input to reach the threshold then the ISIs will follow an Erlang distribution:

$$X_i \sim Exp(\lambda) \quad \rightarrow \quad \tau = \Sigma_{i=1}^{k} X_i \sim Erlang(k, \lambda) \quad | \quad Erlang(k, \lambda) = \frac{\lambda^k x^{k-1} e^{(-\lambda x)}}{(k-1)!}$$

$$E(\tau) = \int_{-\infty}^{\infty} x f_x(\tau; k, \lambda) = \frac{k}{\lambda} \int_{-\infty}^{\infty} f_x(\tau; k+1, \lambda) = \frac{k}{\lambda} \quad | \quad std(\tau) = \sqrt{E(\tau^2) - E^2(\tau)} = \frac{\sqrt{k}}{\lambda}$$

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

Part.1.f - Variability As we saw on the Softky & Koch, 1993, CV for neurons in V1 & MT areas are like this:

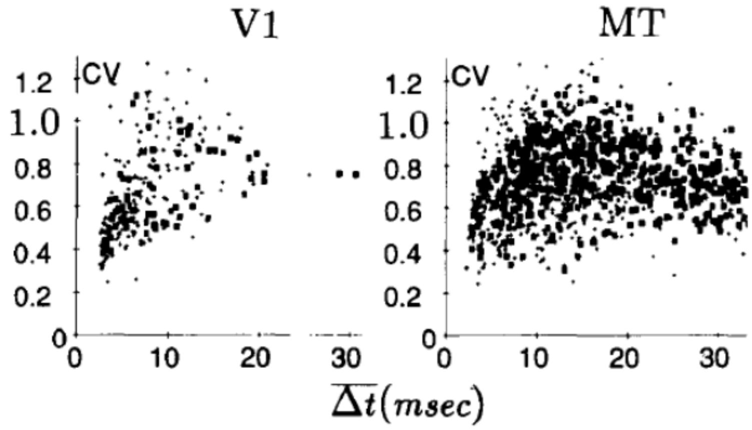


Figure 10: variability of neurons in area V1 & MT

This plot is saying, neurons with more firing rates have less variability (CV). But our simulations showed us that by integrating over inputs which leads to a new spike train with less firing rate, we will have less variability (CV). This is the exact thing that Softkey and Koch realised and they thought that this is some how a problem of integrate and fire so the neurons should be co-incident detectors but then Shadlen and Newsome told that by including inhibitory neurons in the simulation we will make this paradox will be resolved.

Part.1.g - Comparison of CV from integrate models Here we run the simulation for k = 1, 4, 7, 20 and refractory period of 0.000:0.001:0.011, Here are the results:

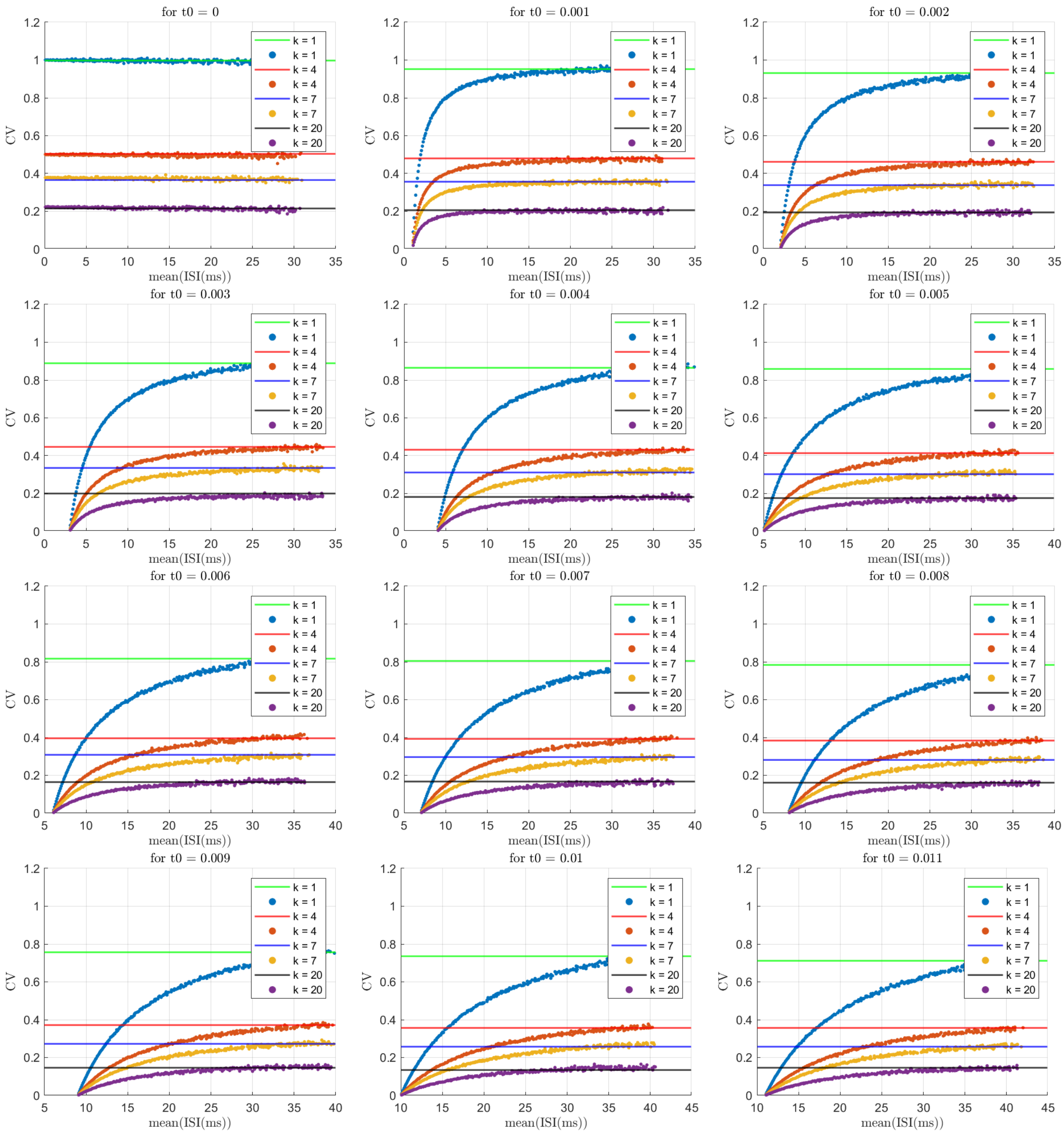


Figure 11: CV Integrated inputs for different "k"s

As you can see, all plots starts from their specific refractory period and are smaller than the $\frac{1}{\sqrt{Nth}}$.

Part.2 - Leaky Integrate & Fire Model

Leaky integrate & fire is the same as IF model plus the leakage of the pre-synaptic inputs:

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

Part.2.1 - Simulation of the Model - Constant Input Here, we did a 100ms simulation by a constant input (RI = 20mv), v_r = 0mv, vth = 15mv and τ_m = [1, 5] using numerical programming and here's the result:

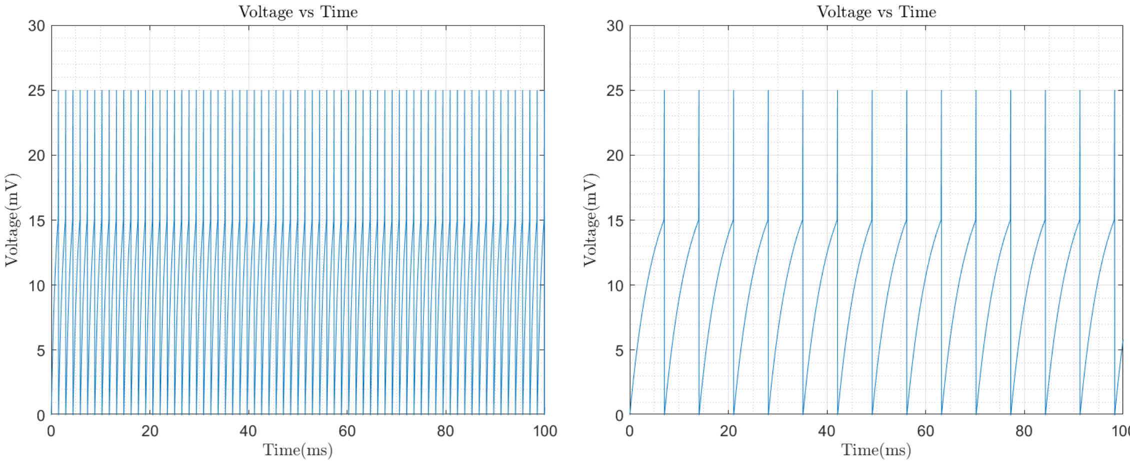


Figure 12: a) tau = 1 | | b) tau = 5

Part.2.2 - Mean Firing Rate of the Neuron - Constant Input Solve the equation below for v(t):

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

$$v(t) = c exp(-\frac{t}{\tau_m}) + RI \rightarrow v(t) = vth \rightarrow t = -\tau_m ln(\frac{vth-RI}{c}) \rightarrow$$

When v(t) reaches the threshold voltage, The Period:

$$\rightarrow T = -\tau_m ln(\frac{vth-RI}{c}) + \Delta\tau_r \rightarrow f = \frac{1}{T} \rightarrow f = \frac{1}{-\tau_m ln(\frac{vth-RI}{c}) + \Delta\tau_r}$$

As we saw in the first part, by increasing τ_m, the frequency of spiking decreases. Here the equation confirm this result.

Part.2.3.1 - Simulation of the Model - Time Varying Input Here we take τ_m to be 20 and we will create a k spike train with ISIs following exponential distribution with r = 250 for 0.1 seconds. Then we convolve the spike train with a EPSC kernel as in the paper and here's the result. As you can see, when we have spike trains, we have EPSCs and increase of the current which cause the neuron to spike. The threshold voltage was 25mV in here:

$$I_s(t) \sim t.exp(\frac{-t}{t_(peak)})$$

$$I(t) = \Sigma_{ik}\, \delta(t - t_i).I_s(t)$$

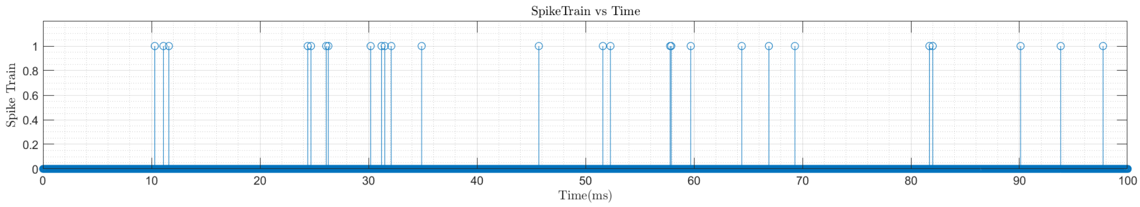


Figure 13: Spike Train

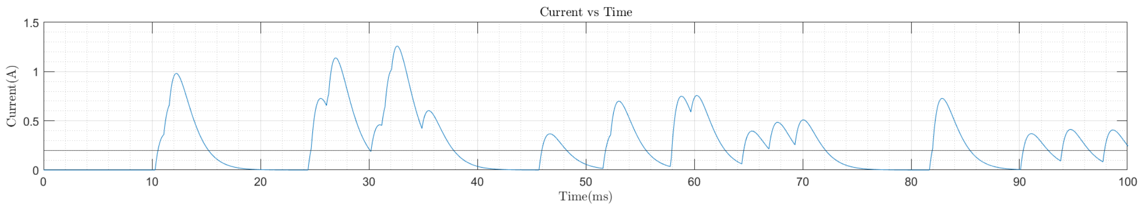


Figure 14: Input Current

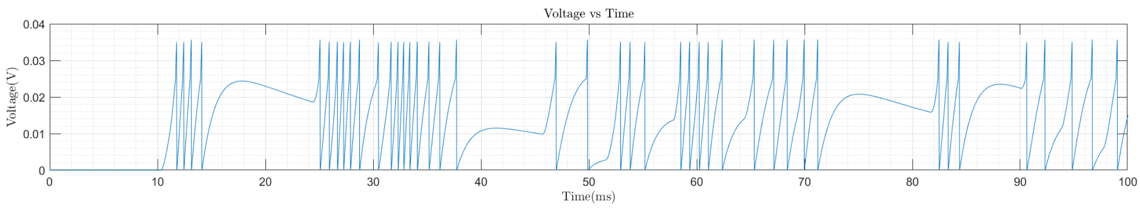


Figure 15: Voltage

Part.2.3.2 - CV Contour By changing τ_m, we can set the decay speed of the EPSPs. Also, we will add a refractory period of 1ms to the neurons in this part. Here's the result for Nth from 1 to 100, τ_m from 0.1ms to 10ms and r = 200:

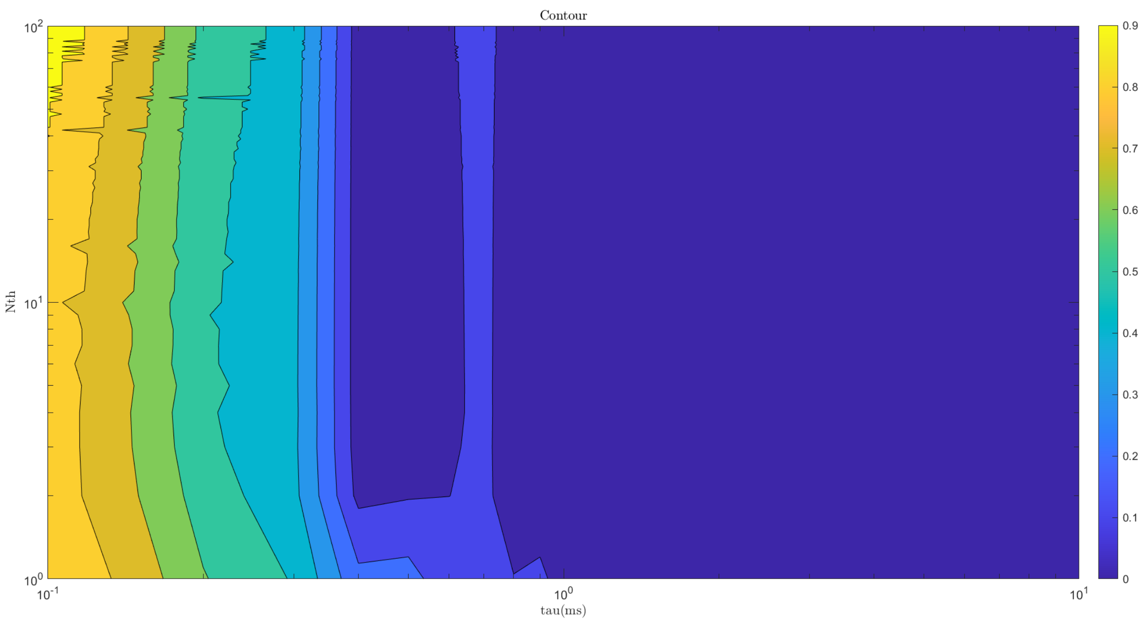


Figure 16: Voltage

Part.2.3.3 - CV vs Width and Amplitude of EPSCs We are expecting the CV to be decreased by both increasing width and amplitude of the EPSCs since we get closer to the periodic behavior in these state. Here are the results for mean over 100 trials:

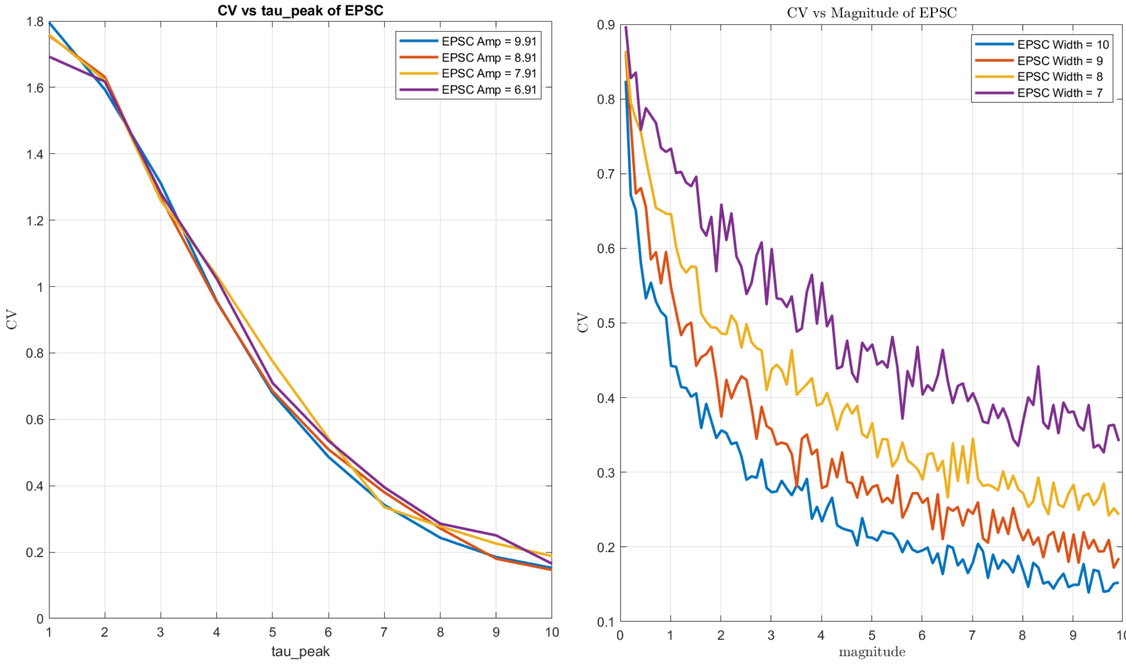


Figure 17: Voltage

Part.2.4 - Including Inhibitory Inputs We repeat part.2.3 by including inhibitory inputs to the simulation. We have a overall of 450 currents which a percentage of these 50 are inhibitory inputs and the others are excitatory. You can see the results for different percentages below:

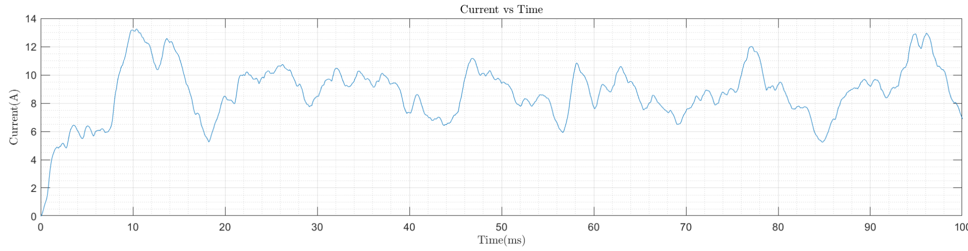


Figure 18: Current - Inhibitory Percentage = 0.1

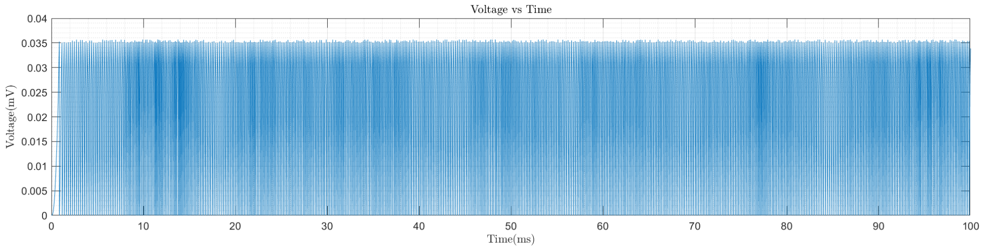


Figure 19: Voltage(V) - Inhibitory Percentage = 0.1

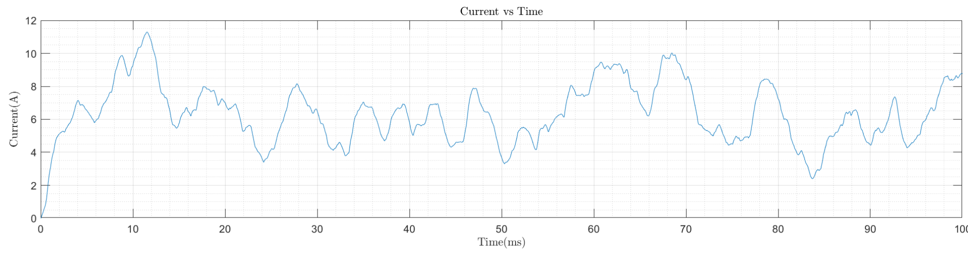


Figure 20: Current - Inhibitory Percentage = 0.2

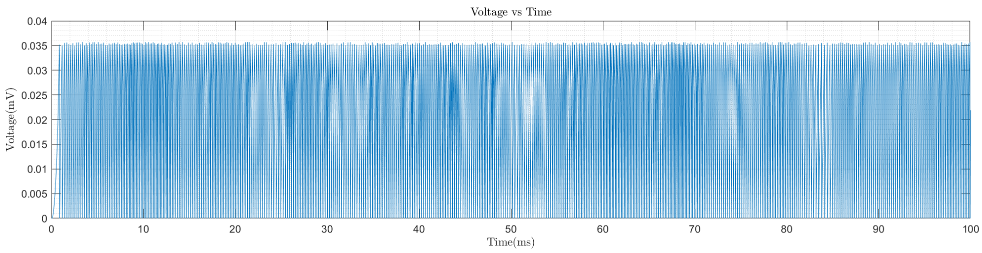


Figure 21: Voltage(V) - Inhibitory Percentage = 0.2

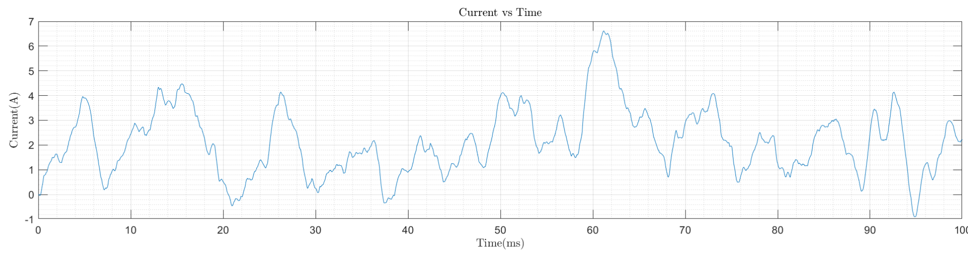


Figure 22: Current - Inhibitory Percentage = 0.3

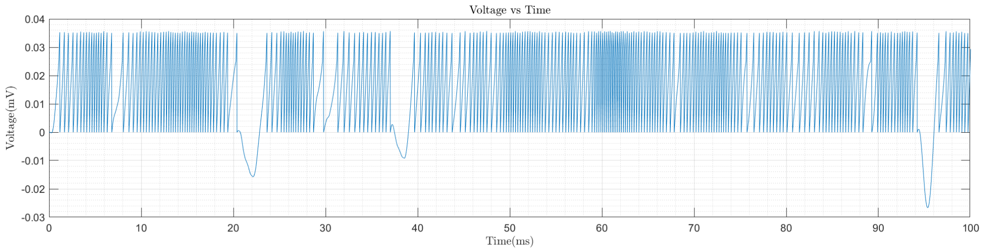


Figure 23: Voltage(V) - Inhibitory Percentage = 0.3

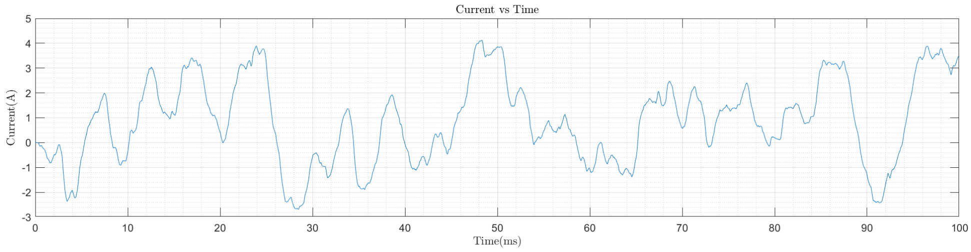


Figure 24: Current - Inhibitory Percentage = 0.5

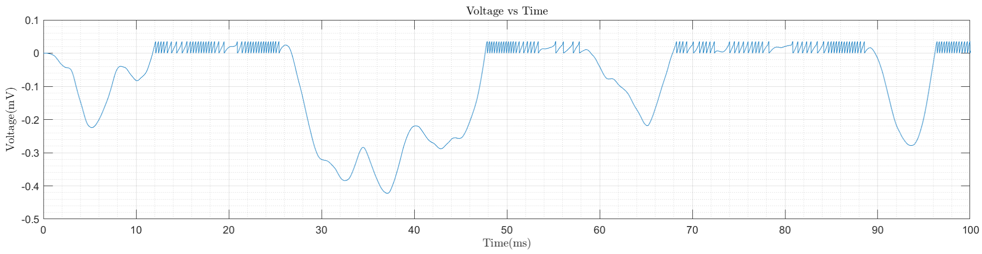


Figure 25: Voltage(V) - Inhibitory Percentage = 0.5

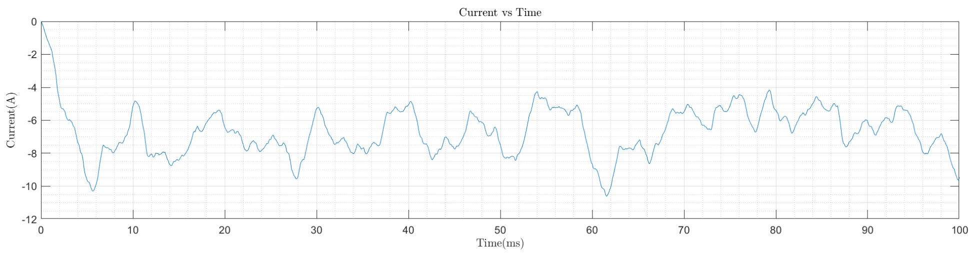


Figure 26: Current - Inhibitory Percentage = 0.7

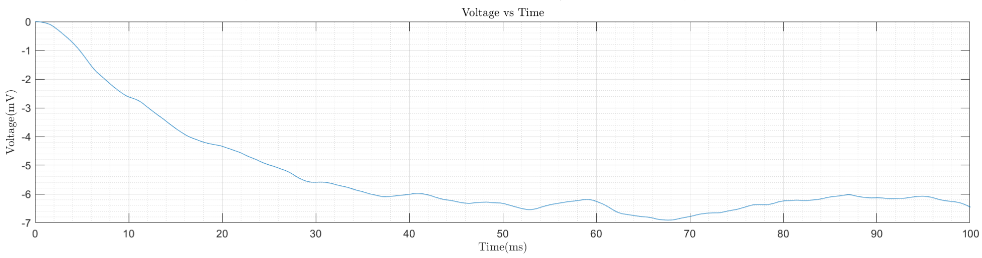


Figure 27: Voltage(V) - Inhibitory Percentage = 0.7

As you can see, by increasing the percentage of inhibitory inputs, the total input will be probably more negative and the firing rate will decrease. By adding inhibitory inputs we can have bigger CVs as Shadlen & Newsome said.

Part.2.5 - Coincidence Detection In this part, we calculate the CV for window lengths of 10:100, k = [1 4 7 20] and averaged over 100 trials and here are the results:

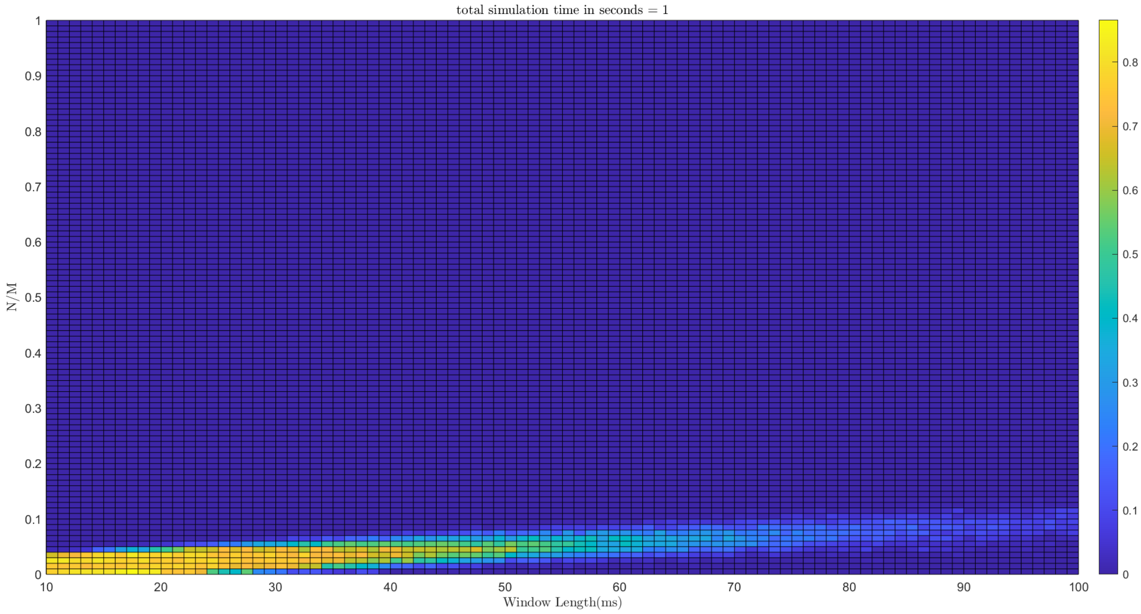
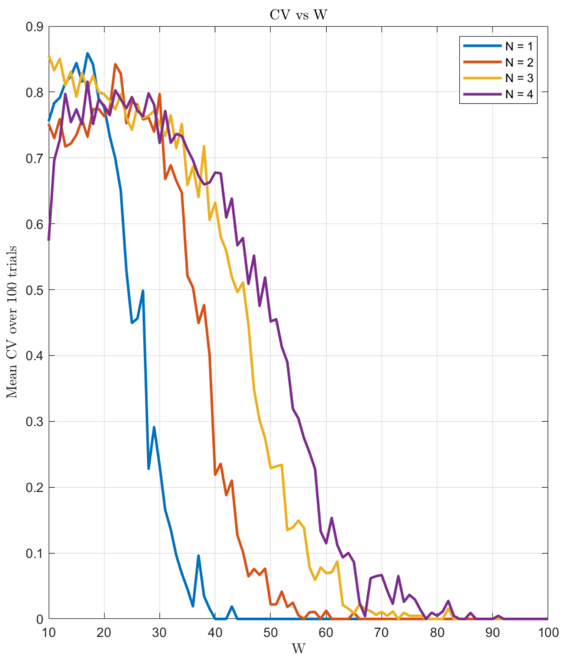
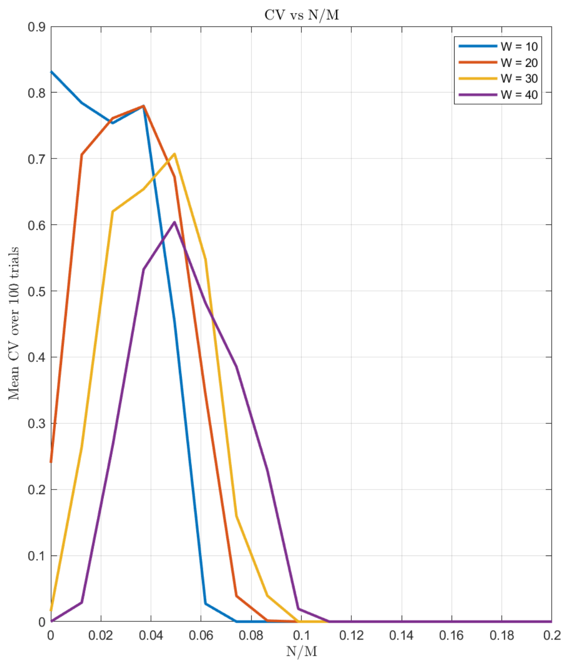


Figure 28: CV plot over Ws and N/Ms

As we can see, by increasing the window length, the CV value will decrease. It's logical because when the window length increases, the neuron gets far from having a random behavior, But when the window length is small, the neuron will have a it's Poisson behavior which results in big CV values. When N/M increases, the CV decreases again because the more the N is, the higher the threshold for spiking, the less the probability of spiking would be and so the neuron gets far from the random behavior and CV will decrease. If N and W both are small, our coincidence detector neuron will be exactly like a Poisson neuron and will have high CV value.





As you can see in the left plot, by increasing the W, CV will fall. In the right plot, when N is too small, the CV is low and by increasing the N, CV will rise and then fall as it is logical.

Part.2.6 - Coincidence Detection - Including Inhibitories Repeating the previous part but just adding 150 inhibitory neurons to 300 excitatory neurons, Here are the results:

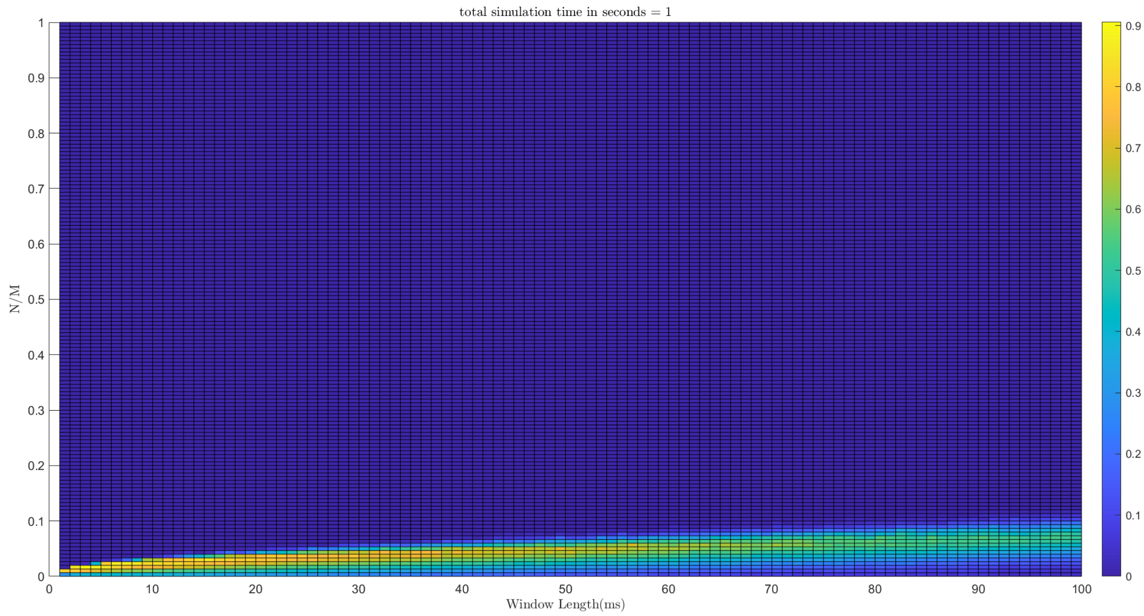
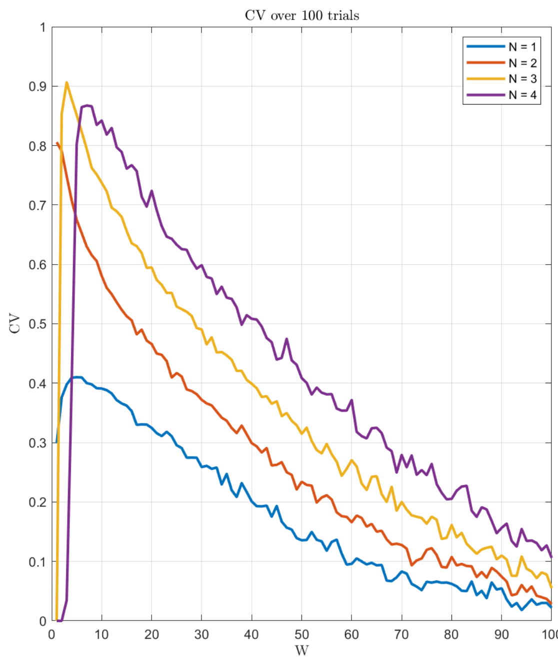
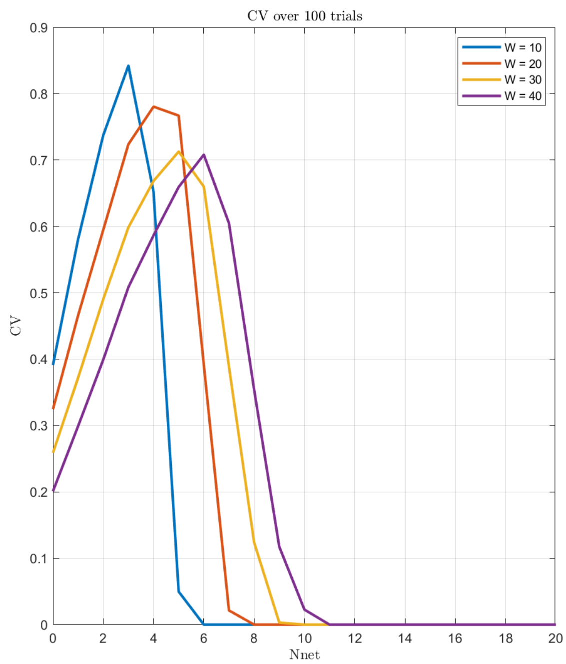


Figure 29: CV plot over Ws and Nnet





The results are the same as the previous part. By increasing the w, CV will decrease and by increasing Nnet, at first there is a rise then a fall.