Computational Neuroscience Homework

HW2: Poisson Spike Generator

# Exercise 1) Poisson spike generator

## A) Make a Poisson spike generator as described in your lecture note, using the Poisson point process. Generate spike trains of a constant firing rate and length . Make a raster plot of spike trains for 50 repeated trials.

Creating a Poisson spike generator is same as creating a spike generator which generates a spike which is independent to other spikes. Thus, we can generate a random number between 0 and 1 for each bin, (here, the bin size would be 1ms) and if the random number is lower the threshold, (the threshold depends on the spike rate) the spike is generated. Otherwise, the spike is not generated. Using this theory, we can easily create a Poisson spike generator and using plot function, we can generate a raster plot for all trials.

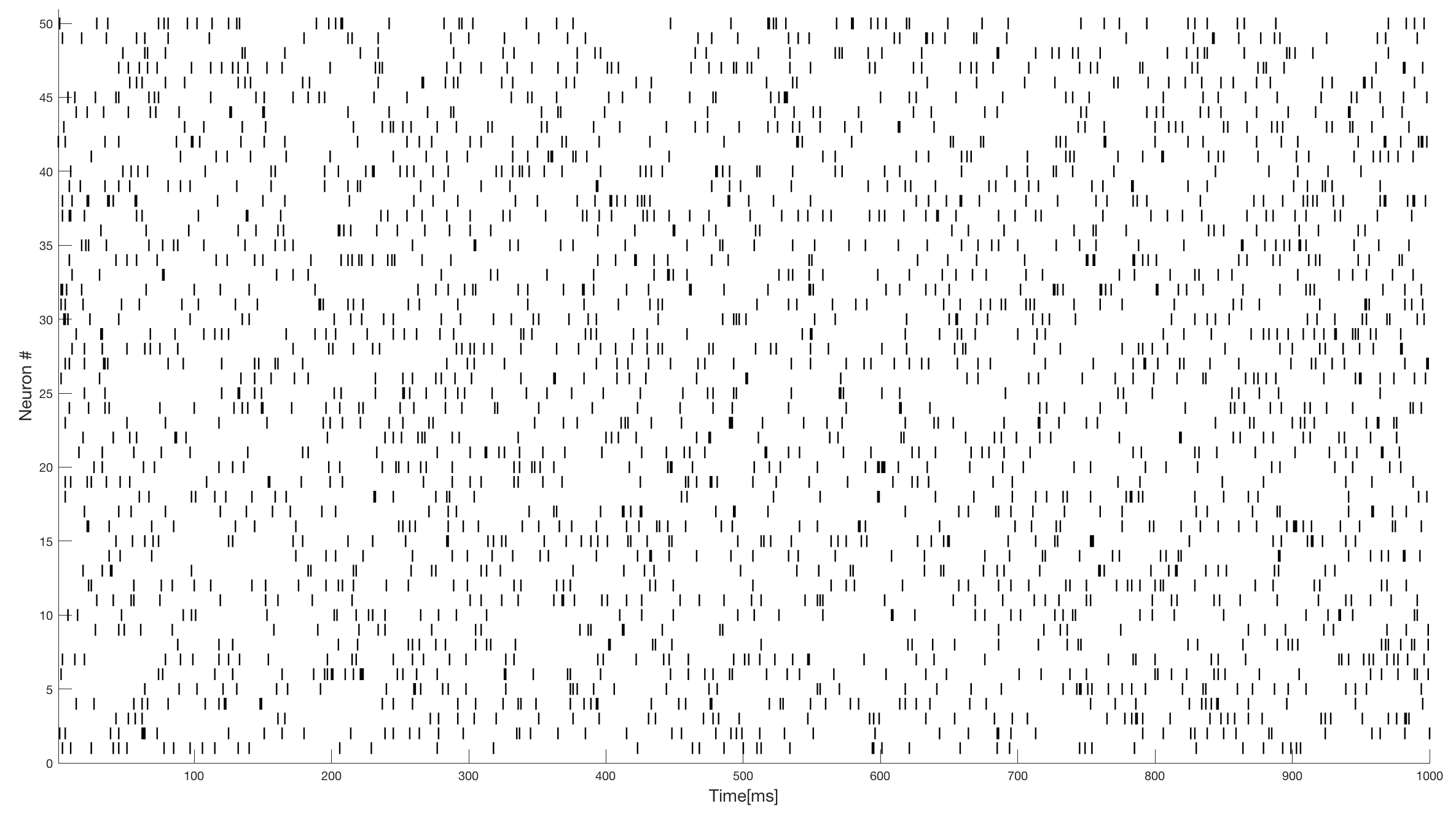


Figure . Raster plot result of problem 1a. (Number of trials: 50, firing rate=50Hz, T=1sec)

Then, for additional analysis, number of trials was changed to 1, firing rate was changed to 100Hz, and T was set to 100s. Then, the inter-spike-intervals were collected, and was plotted as a histogram.

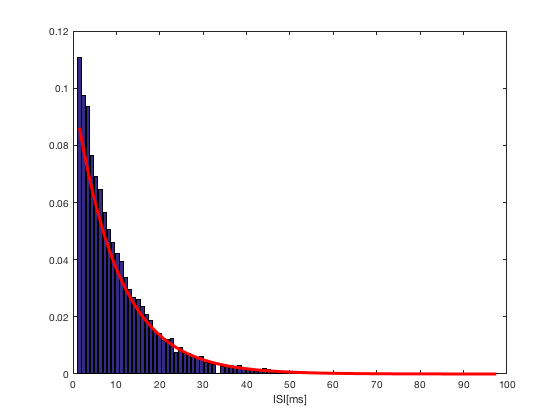


Figure . The ISI histogram for HW2\_1a. The red line is the theoretical curve of ISI.

As shown in Figure 2, the ISI histogram seemed to show a very similar shape with the exponential curve, which is driven theoretically. Thus, we can conclude that the Poisson spike generator constructed in HW2\_1a is constructed properly.

## B) Implement a refractory period of 5ms in the model. In this case, inter-spike-intervals (ISIs) cannot be smaller than 5ms. Explain your design to make this. Generate spike trains of a constant firing rate , and length . Make a raster plot of spike trains for 50 repeated trials.

Remember, a spike generated by Poisson spike generator are independent from other spikes that are generated in different time. Thus, whether we ‘blow-off’ the spikes in certain region won’t influence spikes in other regions. We can easily implement a refractory period of 5ms just by erasing spikes if it is generated after 5ms or less. In the code, this can be achieved by changing 5 values of ‘spk’ array to 0 if we find value 1, which means there is a spike, for a certain time.

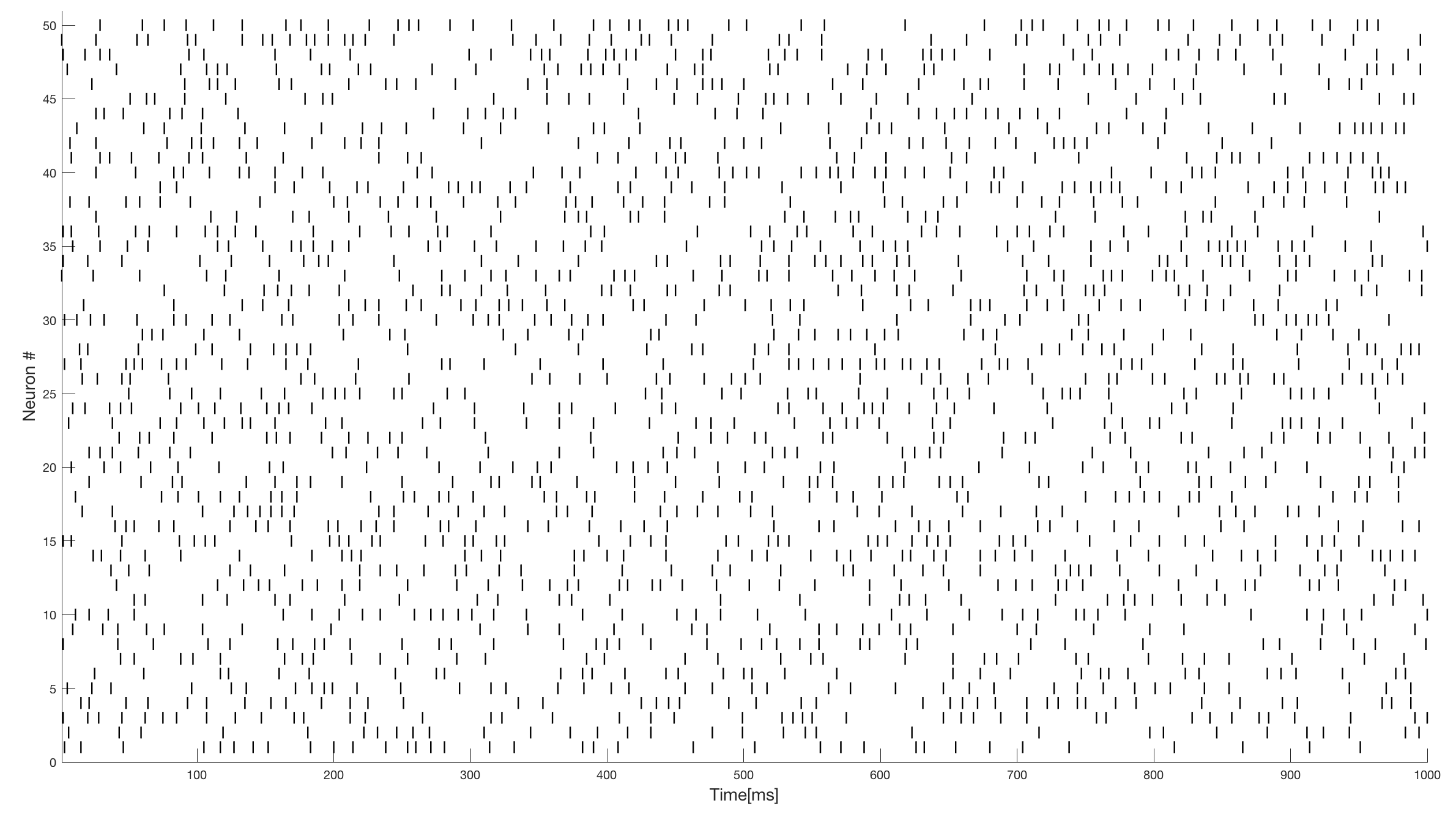


Figure . Raster plot result of problem 1b. (Number of trials: 50, firing rate=50Hz, T=1sec, refractory period=5ms)

In the case where refractory period exists, since ISI cannot be between 0 to 5ms, it means that all ISI larger than 5ms would have a higher frequency compared to when there is no refractory period. Thus, the original theoretical exponential curve would not be appropriate to fit the ISI histogram for this case. We can expect that simply shifting the exponential curve in x-direction (in the time axis as much as the refractory period) would might explain the ISI histogram: although we did not achieve our model by adding refractory period to the Poisson spike generator, we can intuitively expect that ISI would be just simply shifting the ISI curve of the Poisson spikes to the time axis as much as the refractory period. To confirm the result, we have plotted the ISI histogram with the original exponential curve and the curve that is shifted.

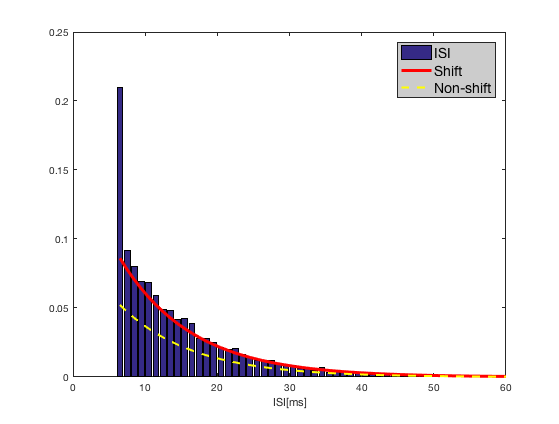


Figure . The ISI histogram for HW2\_1b. The yellow line is the original exponential curve which is the same one in Figure 2, where as the red line is the exponential curve shifted as much as the refractory period.

As in Figure 4, we can conclude that as we expected, the shifted exponential curve explains better than the original curve.

## C) Measure the average firing rate of models in a and b. Can you find any possible problem? If then, discuss your idea for solution. (The solution is suggested in HW2\_1d code.)

By measuring the average firing rate of models in 1a and 1b, we can conclude that the model in 1b had smaller average firing rate compared to model in 1a.

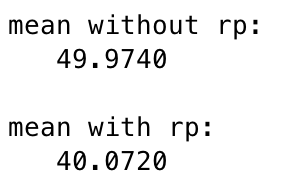


Figure . The average firing rate of models in 1a (without refractory period) and 1b (with refractory period). The average firing rate in 1a model is approximately 50Hz, as what we expected. However, the average firing rate in 1b model is approximately 40Hz, which is lower than what we expected.

This is quite obvious because the 1b model is obtained by ‘removing’ some of the spikes from 1a, which would definitely reduce the total number of spike in certain time range, in other words, the average firing rate. Thus, to obtain the firing rate same as what we ‘expected’, which is 50Hz, would be possibly obtained by increasing the firing rate that we ‘really’ set in the code. But what would be the exact firing rate? Let’s say the ‘observed’ spike rate is , where as the real firing rate which is set in the code is .

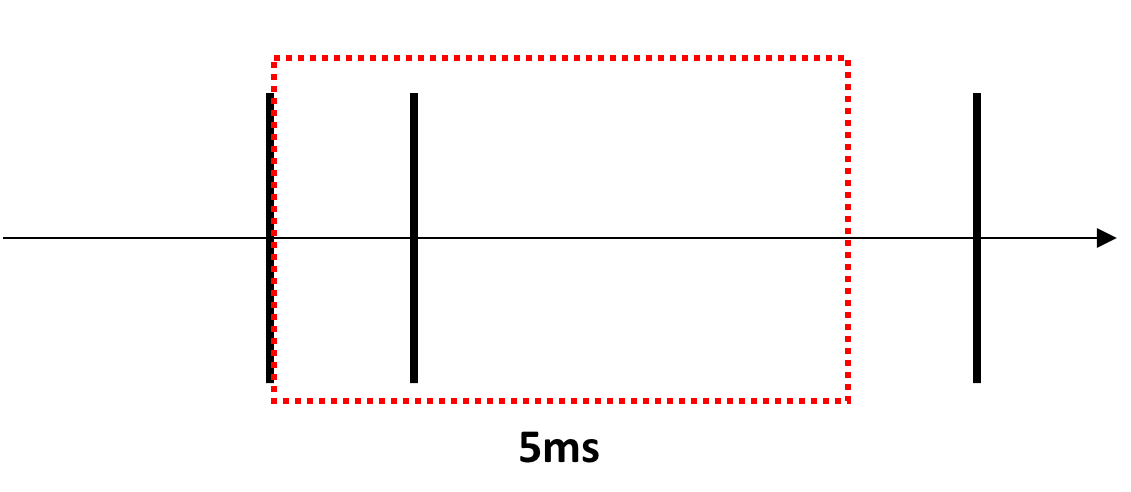


Figure . The brief diagram explaining how spikes are eliminated in HW2\_1b.

As shown in Figure 6, the we set a window of 5ms right after a spike. All spikes in this window would be eliminated, and the ‘expected’ number of spikes that can be observed in a 5ms window would be . The expected number of these 5ms windows in 1 second would be equivalent to the number of spikes that are generated in HW2\_1b model, is . Thus, we can derive an equation as below:

This equation can be re-written as

or,

What we want is to be 50Hz, and thus we can obtain this by setting the to 66.666.. Hz. To confirm whether this conclusion enables us to obtain 50Hz as the average firing rate for 1b model, we implemented the result above into the 1b model. Below is the result.

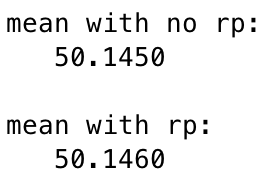


Figure . The average firing rate of models in 1a (without refractory period) and 1d (with refractory period, calibrated). The average firing rate in 1d model is approximately 50Hz, which is the value that we expected.

Thus, by solving a simple equation, we were able to calibrate the firing rate for model 1b.

# Exercise 2) Poisson spike generator II

## A) Make a Poisson spike generator, using the probability density function of ISI. Generate spike trains of a constant firing rate and length . Make a raster plot of spike trains for 50 repeated trials.

If the spike is generated through a Poisson process, the probability for the events to occur for a certain time, when events occur in average, is given as

Then, if events occur on the average rate of per unit of time – what we call the average firing rate, events will be likely to occur for time . Thus, the Poisson distribution can be re-written as

and would be the probability to have no events for time . Thus, the time for an event to occur before time is , and this is . In other words, the probability for a spike to occur after would be

has a range in [0, 1] where as has range in [0, ]. Thus, we can simply obtain the inter-spike-interval, , by simply using the equation below:

Since has a range in [0, 1], we can simply replace with ‘rand(1)’ to generate a random inter-spike-interval, following an exponential distribution. Using this theory, we can easily create a Poisson spike generator and using plot function, we can generate a raster plot for all trials.

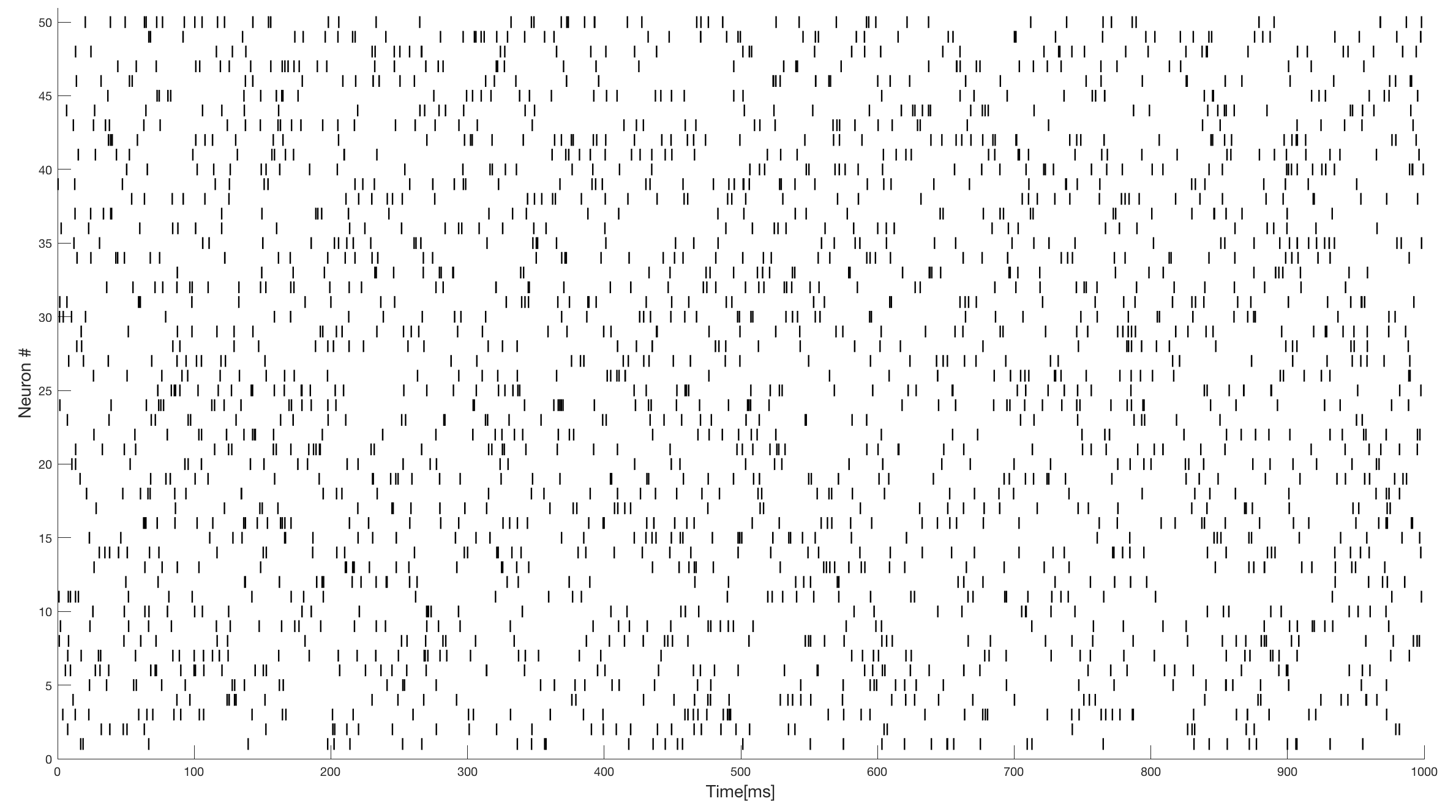


Figure . Raster plot result of problem 2a. (Number of trials: 50, firing rate=50Hz, T=1sec, refractory period=5ms)

## B) Implement a refractory period of 5ms in the model. Explain your design to make this. Generate spike trains of a constant firing rate , and length . Make a raster plot of spike trains for 50 repeated trials.

As we have seen in Figure 4, simply shifting the ISI exponential graph as much as the refractory period enabled us to obtain a curve that can explain the ISI histogram of a neuron model containing a refractory period. Thus, based on this fact, we can conclude that simply adding refractory period to the ISI obtained in the original code HW2\_2a would be an appropriate model to implement a refractory period. Based on this conclusion, we constructed a model, and obtained a raster plot for all trials as below.

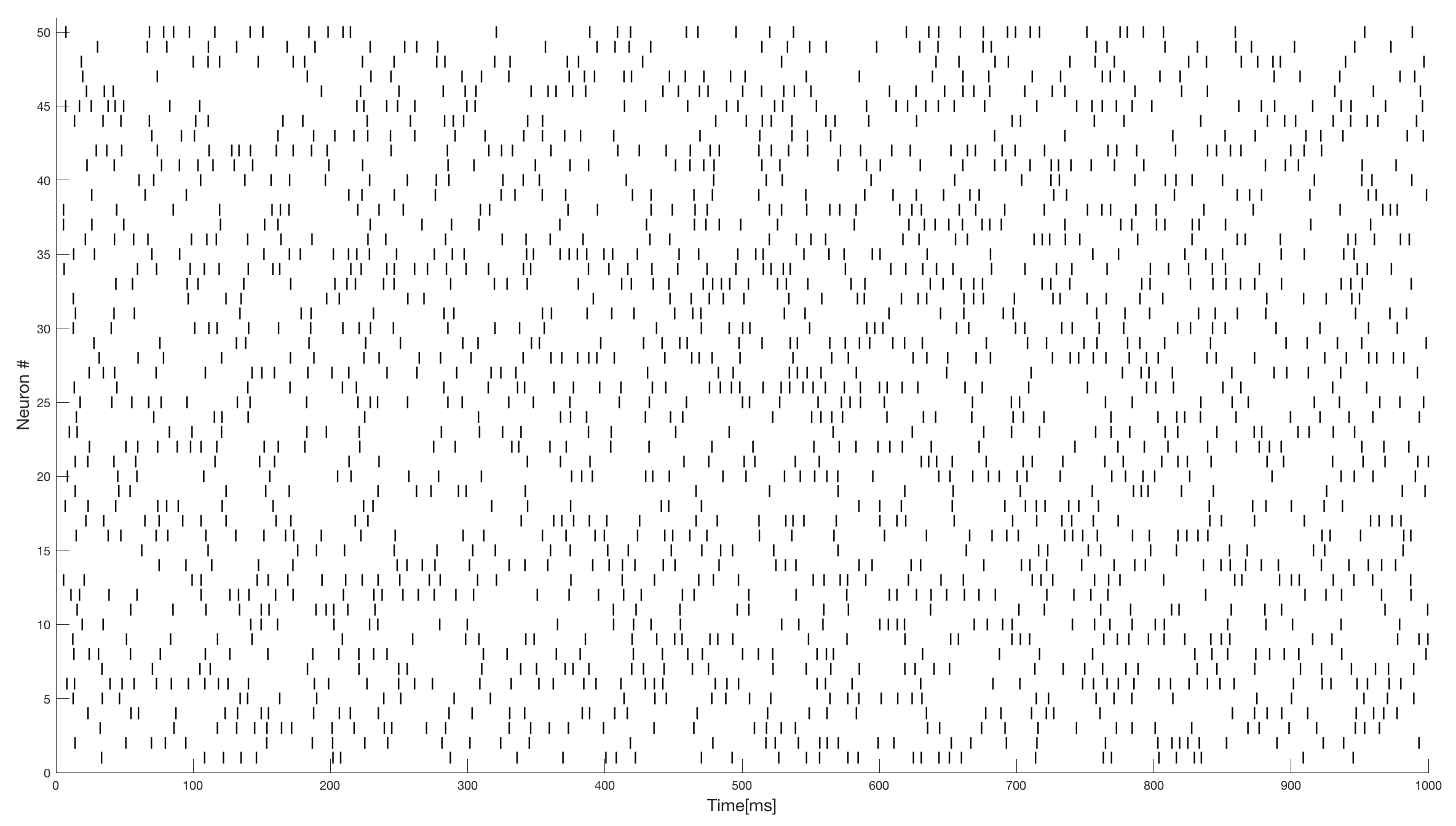


Figure . Raster plot result of problem 2b. (Number of trials: 50, firing rate=50Hz, T=1sec, refractory period=5ms)

## C) Measure the average firing rate of models in a and b. Can you find any possible problem? If then, discuss your idea for a solution. (The solution is suggested in HW2\_2d code.)

Since we simply added the refractory period in the original ISI, it is obvious that the average ISI in model 2b is longer than in model 2a, and this would reduce the average firing rate in model 2b.

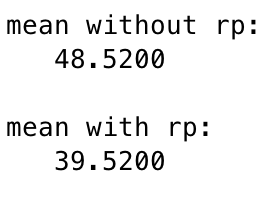


Figure . The average firing rate of models in 2a (without refractory period) and 2b (with refractory period). The average firing rate in 2a model is approximately 50Hz, as what we expected. However, the average firing rate in 2b model is approximately 40Hz, which is lower than what we expected.

When we set the refractory period as , we can say that the probability density function of ISI would be

Thus, we can calculate the expectation of the inter-spike-interval by calculating

Thus, if we say that the ‘observed’ spike rate is , whereas the real firing rate which is set in the code is , we can set a equation as below:

This can be re-written as

and this is identical to the result as in 1c. Thus, as same as in 1c, we obtain that should be 66.666…Hz, and based on this knowledge, we can calibrate the spike rate of the model 2b. The result was obtained as below:

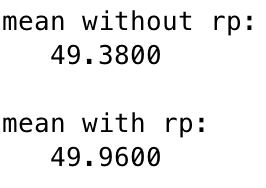


Figure . The average firing rate of models in 2a (without refractory period) and 2d (with refractory period, calibrated). The average firing rate in 2d model is approximately 50Hz, which is the value that we expected.

As what we expected, the average firing rate was approximately 50Hz also in the model with refractory period after calibration, thus we can conclude that the analysis was correct.

## E) Bonus! ISI distribution when stochastic refractory period occurs

After solving exercise 2, I got curious about what would happen if the refractory period was not a fixed value, but rather a random variable that varies following a normal distribution. I expected that this stochastic refractory period would explain the real neuron in a better manner, since neurons have noise, and the channels in the neurons are not ideal: sometimes the refractory period can be relatively shorter than expected, whereas sometimes it can be relatively longer. Thus, I set the refractory period to be a random number, having mean value as 5ms, and variation as 2ms. After obtaining a histogram of the ISI of this model, I noticed the fact that the ISI histogram seems to follow a ‘gamma distribution’, rather than an exponential distribution. To confirm whether it is truly following a gamma distribution, I used the ‘histfit’ function. Below is the result:

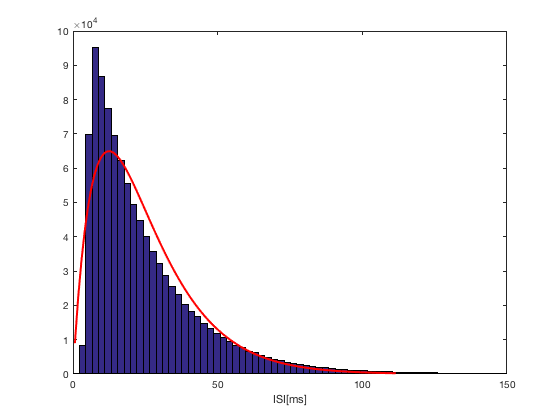


Figure . The histogram of the ISI in a neuron having stochastic refractory period. The ISI of the neuron seems to follow the gamma distribution (red line) as expected.

This ISI histogram seems to have more similar form than the original ISI histogram, with the real data obtained in experiments.[[1]](#footnote-1) Thus, as we expected, refractory period might not be a fixed value in real neurons, but rather a random number following a normal distribution. However, this conclusion is too ambitious to be obtained only using this single modeling result, and further experiments should be done.

# Exercise 3) Cross-correlation of input and output of neuron

## A) Set the peak value of your “open-and-decay” type EPSC() as . Find the value of such that a single spike cannot generate an output spike but two successive spikes of ISI< can. Put this value for your model neuron for next questions.

Here, we simply try three different spike trains as the input of the neuron: spike trains having 2.99ms as its ISI, 3ms as its ISI, and 3.01ms as its ISI. The value can vary from 0 to 3, and we found the case where only spike train with 2.99ms as its ISI can generate a spike. The approximate range of was given as below:

Result/%5BHW2_3a%5D%20Result.png

Thus, in problem 3b, was set as the median of the given range, which is approximately 0.5603.

## B) Suppose your model neuron receives inputs from the Poisson spike generater algorithm in problem 2a, but with friing rate Hz. Plot a cross correlation between input and output spikes (For better result, you need enough number of spikes.) Discuss your result.

We can reach this approach using two different ways: we can make a histogram of the relative input spike time when we set the output spike time as 0ms and stack them, or we can use the MATLAB ‘xcorr’ function and obtain the cross correlogram directly. Both approach has different pros and cons, so both were constructed in my code. (In HW2\_3b, the cross correlogram in a histogram shape, whereas in HW2\_3c, the cross correlogram is obtained by using ‘xcorr’ function.)

The advantage of the histogram version of cross correlogram is that we can set the bin size of the graph, and if we select the ‘appropriate’ bin size, we can clearly understand the data. (We can say, that detailed data that varies among each trial can be eliminated, and the core data, can be only obtained if we extend the bin size into an appropriate size.) However, to attain the cross correlogram as same as what we can obtain from ‘xcorr’, the bin size has to be set to a very small set, same as the time interval of each step in the for loop. In this case, I assume it would be a relatively inefficient algorithm, compared to the ‘xcorr’ function. The cross correlogram when we used the histogram method is plotted below:

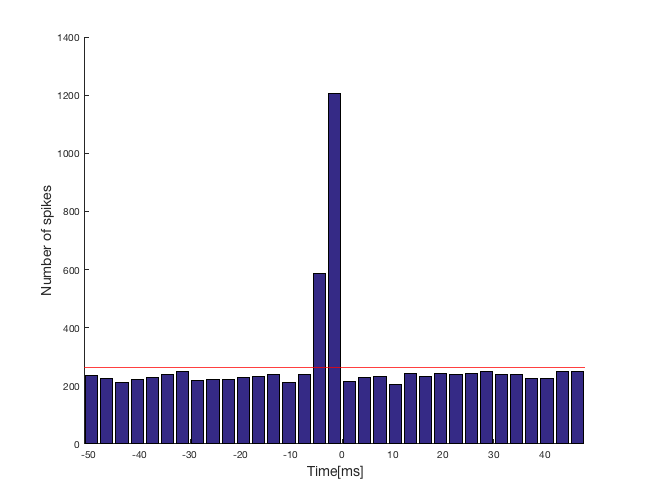


Figure . The cross correlogram histogram when the bin size is 3ms. 0ms is the time where an output spike is generated. As we expected, input spikes generated between 3ms to 0ms before the output spike has shown high correlation with the output spike. Also, input spikes generated between 6ms to 3ms before the output spike is has a high correlation, and this might be the case where the first spike is generated in this range, and the second spike is generated after 3ms or less. (However, the second spike has to be generated)

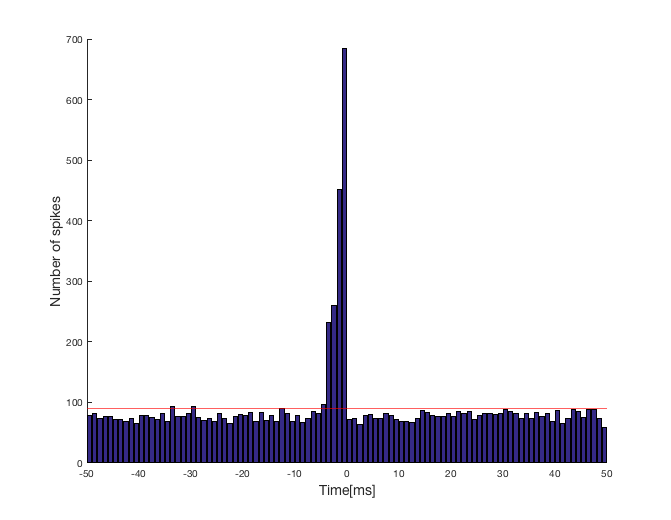


Figure . The cross correlogram histogram when the bin size is 1ms. 0ms is the time where an output spike is generated. For -4ms to -1ms, the correlation of the input spike is high with the output spike. Through this, we can infer the fact that the input spike is usually generated between 4ms to 1ms before the output spike is generated, and especially, input spike generated 1ms (0~1ms) before the output spike is crucial for generating a spike. This is quite an obvious result, since if the the second input spike is generated with an inter-spike interval less than 3ms, this would definitely generate a spike. What we have to focus on here is the fact that the input spikes generated in -4ms to -2ms, (relative to output spike time) which does not seem to induce a spike alone, had a high correlation with the output spike. Thus, we can conclude that spike trains having approximately 3ms ISI would generate an output spike.

As mentioned previously, obtaining the correlogram using xcorr would be more efficient way when we want to obtain the correlogram in a very high resolution. The result is plotted as below:

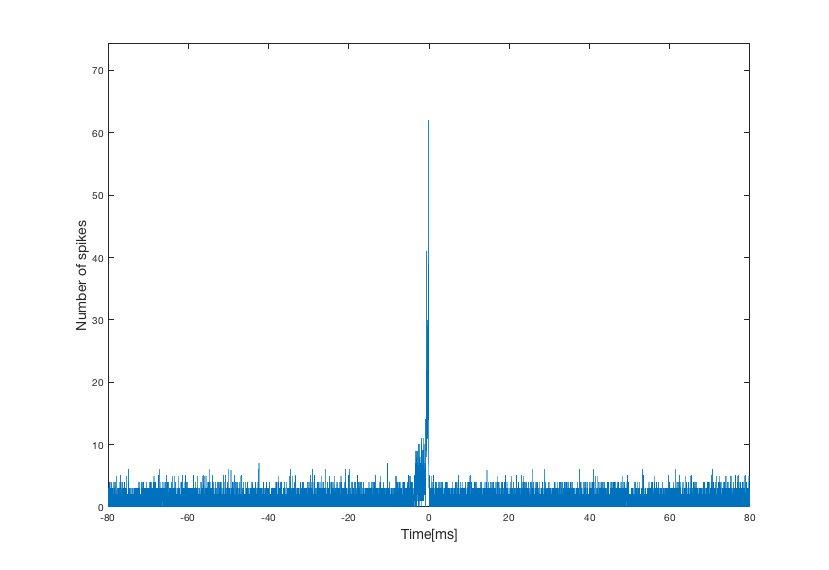


Figure . The cross correlogram obtained using ‘xcorr’.

According to Figure 13 and 14, we can conclude that spikes generated before the output spike is crucial for generating the output spike. However, besides the obvious fact that the histogram bar right before 0ms is highly correlated, (this is obvious because if some input source induces a spike directly, it is most likely that those input spikes would be found right before the output spike.) what we should focus is the fact that input spikes 4ms~2ms before the output spike also had relatively high correlation with the output spike generation. Based on this knowledge, we can infer the fact that those spikes which are relatively far to generate a spike alone, might somehow – in this model, this would be increasing the membrane voltage – influence the neuron to have a higher chance to generate a spike when another input spike occurs. However, this only holds for a certain time range, which is -4ms to -1ms. Thus, we can approximately predict that the ISI should be 3ms or smaller to generate an output spike. (Which is exactly what we modeled in Problem 3a.)

1. JS Taube, *Journal of Neurophysiology*, 104(3):1635-1648. (2010) [↑](#footnote-ref-1)