

Practical 4: The Integrate & Fire neuron

2014 version by Mark van Rossum

2018 version by Matthias Hennig and Theoklitos Amvrosiadis

October 22, 2019

1 Introduction to MATLAB

If this is your first time working with MATLAB or you need a quick refresher, please refer to the instructions of the previous lab (NClab3). There you will find a short introduction to the basics of programming with MATLAB, as well as the functions that are used in this practical. For anything that is not addressed there, you can use MATLAB's "help" or the Mathworks online forum.

2 Simulating a noisy input to Integrate & Fire neurons

2.1 Model Setup

First, we will construct an array of I&F neurons. To do that, we have to give them some basic properties to make them functional (resting membrane potential, V_{rest} , reset voltage, V_{reset} , spike generation threshold, V_{thr} , membrane time constant, τ_m , input resistance. R_m is considered to be 1 for simplicity).

We also specify the time parameters of our simulation (duration of the simulation, timestep size, and number of timesteps).

Moreover, we will provide our neurons with an input current and two noise inputs, one that is common to all cells (can be seen as part of the stimulus, or as part of the global network activity¹), and one that is independent for each neuron (can be seen as reflecting the unique set of connections that each neuron receives).

Finally, we create some arrays to hold our data for the duration of the simulation.

Please look through the first section of the code and make sure you understand what each variable stands for.

2.2 Running the simulation

Once we have all the relevant variables specified and our data structures ready to store data, we can run the simulation. This simply consists of going one timestep at a time and:

- providing each neuron with an input current
- adding the common and independent noise components
- calculating the new membrane potential for the current timestep
- determining whether any cell has reached the AP threshold
- resetting the membrane potential of all the cells that have fired to the reset voltage

The values of the common input current, the membrane voltage of each cell, and a binary value for the spikes are all stored in the corresponding arrays we created during the model setup.

Look carefully through this part of the code, identify each of the steps detailed above, and make sure you understand what is going on.

¹Can you guess as to what might influence a cortical network as a whole?

2.3 Plotting the results

What we want to do after the end of the simulation is to look at the spikes our cells fired in response to the input we provided. We can do this by making a *raster* plot, which you have encountered in the lectures. While here we are looking at the spikes for many different neurons at the same time, you can imagine that the different neurons are equivalent to recordings of the *same* neuron in different *trials*, as is usually the case with raster plots from experiments. What do you observe in the raster plot compared to the input amplitude? Discuss with your neighbour.

You should end up with something like Figure 1. We are using the data structure we used to hold our spiking data for the raster plot, and we are also visualizing the input current as a separate subplot. The two plots are aligned in terms of simulation time. Look through the code used to generate the plots and try to understand the commands. Most of the commands are used to make the figure more presentable.

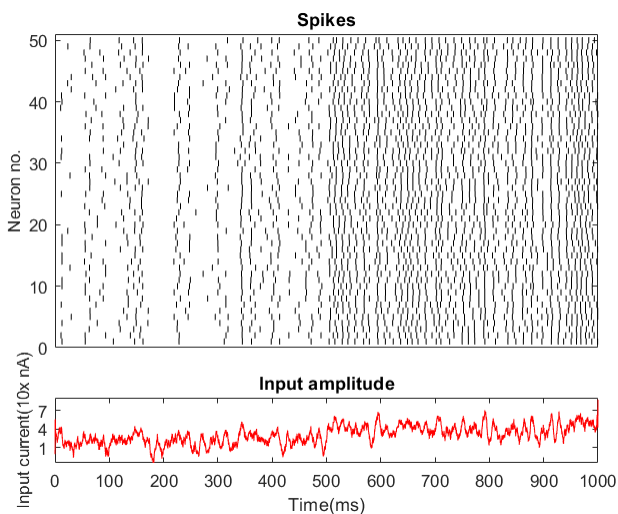


Figure 1: Raster plot and input current

Finally, we are computing the InterSpike Interval (ISI) distribution and visualizing it with the *histogram* function. It should look like Figure 2. Make sure you understand its generation. Discuss your thoughts with your neighbour. More below.

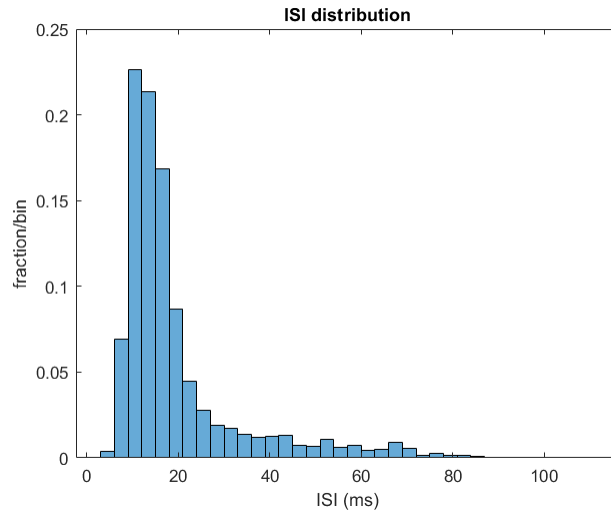


Figure 2: ISI distribution

3 Exploring the model's parameters

3.1 Neuron parameters

Try changing the parameters that characterize the neurons (voltages, time constant). Before each run of the simulation try to predict the outcome qualitatively. Check the correspondence of your intuitions with the results. For visualization purposes, also try varying the number of the neurons in the simulation (beware of increasing runtime!).

3.2 Input parameters

- What role does the input current play? How would you expect the spike trains to change if you vary its amplitude? Check your predictions in the raster plot but also the ISI distribution. Try incorporating different inputs at different time points.
- What role does each noise parameter play? What is the difference between them? What will happen if you set either one, or both, to zero? Make predictions before looking at the results! Discuss with your neighbours. What is stimulus locking? Recall the lectures and try to identify in the graphs.

4 ISI statistics

- What information does the ISI distribution provide? Plot the histogram for the different input conditions separately (i.e. first and second half of simulation, or as many different input conditions as you have included). Are they different? Why/why not?
- (Optional) You can look at the fit to the distribution with the function *histfit*. Look up how it works first, there are various options.
- Calculate the coefficient of variation (CoV)² for the ISI distribution. What does CoV signify in general? What does it mean for a neuron? Try doing this for the different input conditions separately and try different window sizes as well. How do you expect its value to change?

²Remember that the CoV for a distribution is defined as $\frac{\sigma}{\mu}$, the ratio of standard deviation over the mean of that distribution.

- (Optional) Do the above for the Fano factor as well ³. Plot the resulting Fano factor with different window sizes.

5 (Extra) Refractory period

Our neurons currently have no refractory period. However, that does not correspond to what happens in reality. Implement refractory periods of different durations and examine the effects on neuronal firing statistics.

6 (Extra) Adding adaptation

We can add adaptation to the model as follows:

- If the neuron spikes, increase the Ca^{2+} concentration by 1.
- At every timestep, the Ca^{2+} concentration decays exponentially back to zero with a time-constant of $50ms$.
- The Ca^{2+} activates a K_{Ca} current as $I_{K_{Ca}} = [Ca^{2+}] * g_{K_{Ca}} (V_{K_{Ca}}^{rev} - V)$. Take $V_{K_{Ca}}^{rev} = V_{rest}$ and $g_{K_{Ca}} = 2$.

Apply step currents of different strengths. How much and how quickly does the neuron adapt?

³The Fano factor is defined as $F(T) = \frac{\sigma^2(T)}{\mu(T)}$, for a time window with size T .