

INF5470, exercise 2: Exploiting Noise to Enhance Information Throughput

P. Häfliger & Jenny Olsson
Institute of Informatics
University of Oslo
e-mail: hafliger@ifi.uio.no

September 28, 2007

Abstract

This project introduces the surprising fact, that noise can be used to improve the quality of a signal encoding. Its use for decorrelation in population coding is explored. We base our analysis on looking at a sinusoidal input signal and we estimate, how well this signal is still present at the systems output by the signal to noise ratio (SNR). (Note that this is only one possible form of analysis. It is well established though and a rather general measure for signal quality. However, if one knows more about how information is encoded in a system, more dedicated methods to estimate information throughput will be more appropriate.)

1 Tools

The CSIM package for matlab allows you to simulate neuronal pulse networks with a variety of standard neuronal models. A complete manual can be found in `\\Samba\infneuro\CSIM\usermanual.pdf` (using Windows), or under `/hom/infneuro/CSIM/usermanual.pdf` (using Linux/Sun).

1.1 Getting ready to use CSIM

- start matlab
 - go to your working directory
 - add path with subfolders:
 - "for windows"
 - * add the path `\\Samba\infneuro\CSIM\` with subfolders with the MATLAB command
- ```
>> addpath(genpath('\\Samba\infneuro\CSIM'));
```

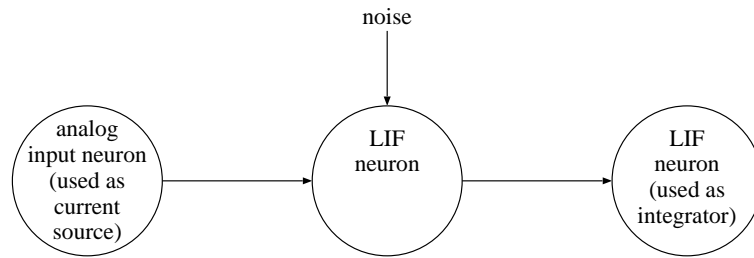


Figure 1: Encoding an analog input signal by a leaky integrate-and-fire neuron

It is possible that this produces a warning message concerning change notification handles, due to a Samba server limitation. This should not be of importance as you will not be making changes to the CSIM code.

– ”for UNIX”

\* add the path `/hom/infneuro/CSIM` with the MATLAB command

```
>> addpath(genpath('/hom/infneuro/CSIM'));
```

- Optionally you may save the changes in your working directory, using 'File' → 'Set path...' → 'Save'
- CSIM should be working now.

## 1.2 Example code

CSIM offers one form of function call, for all its functions:

```
csim(<function_name>,<parameter1>,...,<parameterN>)
```

There is an example program called `start_example.m` in the CSIM directory, that you should use as a starting point for the programs you are going to write in this exercise. Copy this to your working directory.

It defines a sequence of three neurons as illustrated in figure 1 and runs a simulation on them. The first neuron is stimulated with a 10Hz sinusoidal input. In order to estimate how much of the input signal can be reconstructed by looking at the output of the system, the signal to noise ratio (SNR) of the last neuron's membrane potential is computed in this program by the function 'snd'. It computes the frequency power spectrum, takes the frequency with maximum power as the assumed signal frequency (always check in the graph that this is actually your input frequency), and computes the ratio between its power and the power integrated over the other frequency bands (shown between the blue bands in the graph, exempting those between the red).

### 1.3 Hints for modifying the example code

- On line 44 - `csim('set',n2,'Inoise',5)` sets the amount of input noise of the first LIF neuron. This is the noise parameter we are interested in modifying in this exercise.
- On line 70 - `csim('set',syn12,'W',8.00)` is the weight of the synapse from the input neuron to the first LIF neuron. This is the weight parameter we are interested in modifying.
- On line 132 - `spikes.channel(1).data` summons the spiking data for the first LIF in the second layer. When using multiple LIFs in the second layer, you can access each of those by changing the channel number (i.e. `spikes.channel(i).data` for accessing spiking data from LIF number `i`).
- On line 156 - `plot(t,vm.channel(3).data,'b')` will plot the membrane potential of the 3rd layer LIF when the script is in its original state. If `n` 2nd layer LIFs are added to the simulation, the 3rd layer LIF's membrane potential will instead be accessed from `channel(n+2)`. Remember as well that we are interested in the frequency analysis of this 3rd layer LIF neuron.

## 2 Task

### 2.1 Noise to enhance response to weak signals

Use the setup described by figure 1 and start with the program `start_example`. If you run it with the original settings, you will notice that most power will not lie in the desired 10Hz frequency but somewhat lower. Why is that? Tip: run the simulation for 1 second only (change `Tsim` in the program) to see better what is happening in the figures plotted by the program. The frequency spectrum, though, will not be accurate with only so short a sample. Alternatively you can also zoom into the plot of a 100s simulation. (Also, note that because of the randomness of the noise, consecutive experiments can differ in their results.)

If you increase the weight of the synapse from the analog input neuron to the IF neuron, this situation changes and finally the 10Hz signal is stronger in the output. But alternatively you can leave the weight alone but slowly increase the noise input to the first LIF neuron instead. For a certain noise magnitude, the 10Hz signal will be strongest in the output. Do you have an intuitive explanation as to why noise actually helps here? Which is the 'optimal' noise level with the best SNR in this scenario? What SNR can you reach? Show a frequency spectrum for this noise level!

### 2.2 Noise for decorrelation in population coding

Extend the simulation network according to figure 2 to 50-100 LIF neurons. Make sure that you do this in a way, that all LIF neurons have their own noise

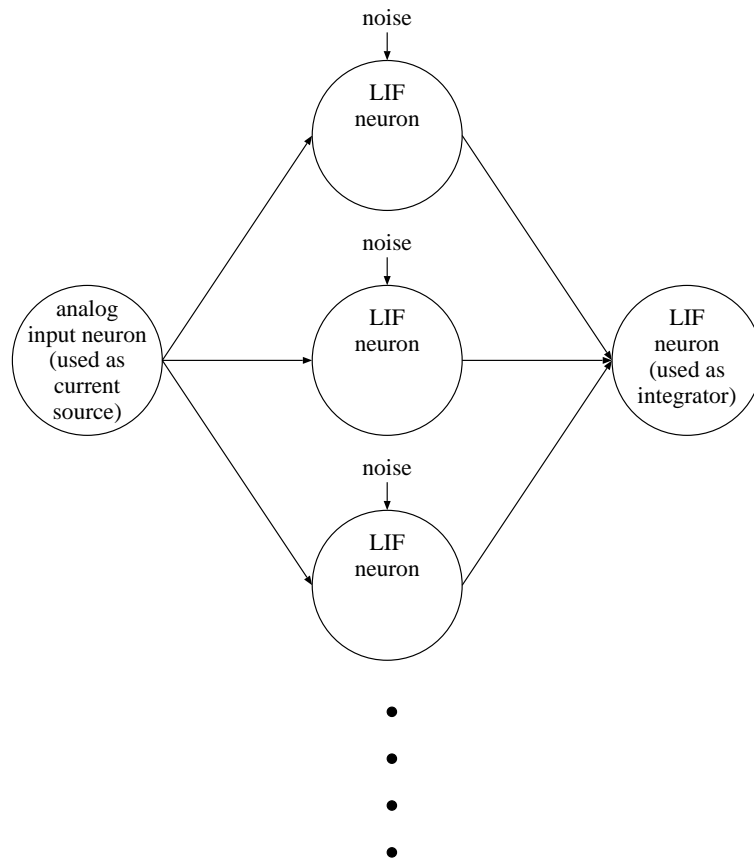


Figure 2: Encoding an analog input signal by a population of integrate-and-fire neurons

source. Set the noise to zero at first. Otherwise, use the original parameters from `start_example` again. You will notice that the result is still quite bad: still, the 10Hz is not the most powerful frequency component. But as you set the noise level back to its original value, the 10Hz suddenly pops out again, although, with just one LIF neuron, it did not. Why is that? What is the best SNR you can reach by tweaking the noise level? Show a frequency power spectrum of that solution, too!