

A Tour of NeuraLOG and Spike

Tools for Simulating Networks of Spiking Neurons

Version 1.034

Lloyd Watts

Synaptics, Inc.
San Jose, California

California Institute of Technology
Pasadena, California

The copyright for the source code and documentation for AnaLOG is held by John Lazzaro and David Gillespie. The copyright for the source code and documentation for Cview is held by David Gillespie. The copyright for the source code and documentation for Gnuplot is held by Thomas Williams and Colin Kelley. AnaLOG, Cview, and Gnuplot are distributed under the GNU General Public License. Mathematica is a registered trademark of Wolfram Research, Inc.

NeuraLOG is a customization of AnaLOG by Lloyd Watts. The copyright for the source code and documentation of Spike is held by Lloyd Watts. Spike is distributed by Lloyd Watts under the GNU General Public License. The GNU General Public License is included with Spike software distribution.

Copyright ©1993 by Lloyd Watts. All rights reserved.

Lloyd Watts makes no representations, express or implied, with respect to the documentation or the software it describes, including without limitations, any implied warranties of merchantability or fitness for a particular purpose, all of which are expressly disclaimed.

Lloyd Watts gratefully acknowledges helpful discussions with Carver Mead, Sylvie Ryckebusch, Misha Mahowald, John Lazzaro, David Gillespie, Mike Vanier, Brad Minch, Rahul Sarpeshkar, and Kwabena Boahen. Thanks to Federico Faggin and Carver Mead for actively encouraging this project, and to Al Crawford for long-standing support and encouragement.

For information on NeuraLOG and Spike, contact:

Lloyd Watts

Synaptics, Inc.
2698 Orchard Parkway
San Jose CA 95134
email: lloyd@synaptics.com

Computation and Neural Systems
California Institute of Technology
CALTECH 139-74
Pasadena CA 91125
email: lloyd@pcmp.caltech.edu

A Tour of NeuraLOG and Spike

NeuraLOG is a schematic entry tool, which allows you to conveniently and quickly enter a “neural” circuit diagram, consisting of neurons, synapses, test inputs, and your own custom symbols. You also specify the parameters of the neural elements, such as the refractory period, duration and intensity of the post-synaptic current pulse following an action potential, saturation value of summing post-synaptic currents, tonic input currents, etc. NeuraLOG is a customization of the program AnaLOG, by John Lazzaro and Dave Gillespie.

Spike is a fast event-driven simulator, optimized for networks of spiking neurons and synapses. Spike reads the network specification file created by NeuraLOG, and reads a simple instruction file, in which you specify how long you want the simulation to run, and the timing of spikes you wish to supply to your test inputs. Several input formats exist, including periodic spike trains, periodic bursts, poisson random spike trains, and gaussian-jittered periodic spike trains. The output of your simulation is a set of traces, one for each object in your schematic. You may instruct Spike to report only particular traces for large designs. You may view the output traces using several popular plotting programs, including Mathematica, Cview, and Gnuplot.

The software is distributed at no charge with installation instructions, and several working tutorial examples with associated descriptions. New users who already know how to use AnaLOG have reported that they learned the tools and had valuable results within 15 minutes of sitting down at their computer. For non-AnaLOG users, a simple textual input format is supported for entering the neural circuit diagram.

A Simple Example

A simple neural circuit, called the adapting tonic burster, is shown in Figure 1.

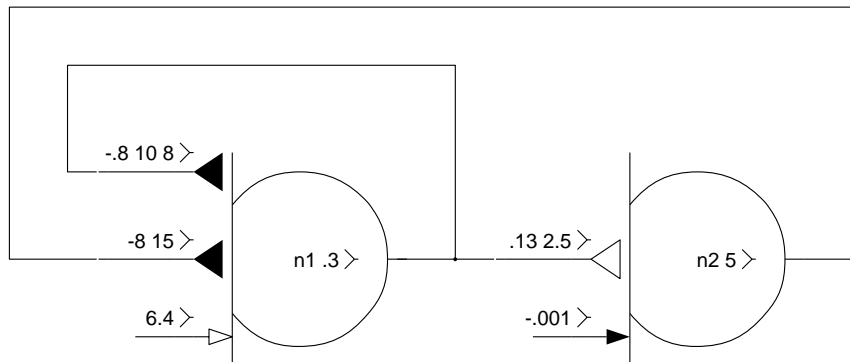


Figure 1 Graphical input representation of the adapting tonic burster circuit, as entered in NeuralOG.

This circuit consists of two two neurons (the large disks), several synapses (the large triangles), and two tonic inputs (the small arrows). There is a text string inside each neuron symbol which specifies the neuron’s identifier label (for example, `n1`), and the neuron’s absolute refractory period in milliseconds (ms). The neuron has a default membrane capacitance of 1 picoFarad (pF) and a default threshold for firing of 1 Volt (V). On the left side of the neuron symbol is a line that represents the input, or “dendrite” of the neuron; synapses and tonic inputs communicate to the neuron via its dendrite.

At the bottom left of the diagram is an arrow pointing into the dendrite of neuron `n1`; this arrow represents a constant excitatory tonic input to neuron `n1`, the value of which is specified to be 6.4 nanoAmperes (nA). Neuron `n2` has an inhibitory tonic input of -0.001 nA; the filled arrowhead symbol indicates inhibitory input.

Neuron `n2` has an excitatory synapse from neuron `n1`. The first synapse parameter is the value of the post-synaptic current in nA. The second parameter is the duration of the post-synaptic current pulse duration in ms. As with the tonic input, a filled synapse symbol indicates an inhibitory input. When many spikes arrive in rapid succession at a synapse, the default behavior is to re-start the current pulse, but not to increase the intensity of the current pulse. Such a synapse could be called an “all-or-nothing” synapse, or a binary synapse.

However, it is possible to specify that successive input spikes should increase the post-synaptic current, up to some saturation value. The inhibitory synapse in the upper left of the diagram has a third optional parameter, indicating the number of post-synaptic current pulses that will be summated before saturation. In this case, up to 8 post-synaptic current pulses will superpose; after this limit is reached, additional spikes on the input do not cause an increase in the current flowing into the post-synaptic cell. This type of synapse is called

a “summing” synapse. Note that the default “all-or-nothing” synapse is the special case of the summing synapse when the saturation value is set to unity.

The command file for simulating this circuit consists of the following instructions:

```
start_runtime 5
end_runtime 70
```

These instructions indicate that the simulation should begin at time 5 ms and end at time 70 ms. The Spike simulation output for this circuit is shown in Figure 2.

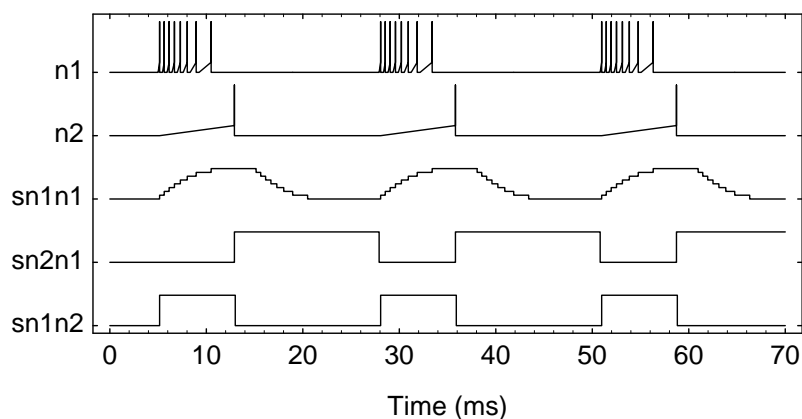


Figure 2 Simulation results, as plotted in Mathematica.

The output consists of 5 time traces, one for each neuron and synapse in the schematic diagram. The first two traces are for the neurons **n1** and **n2**. The remaining three traces are for the synapses; the label **sn1n2** indicates a synapse from neuron **n1** to neuron **n2**. (If there are several synapses between two neurons, a unique identifier is generated by appending a suffix.)

The operation of the circuit is as follows: the strong excitatory tonic input causes neuron **n1** to fire repeatedly. The weakly excitatory synapse from neuron **n1** causes neuron **n2** to fire after several spikes from **n1**. The strongly inhibitory synapse from neuron **n2** causes neuron **n1** to stop firing for 15 ms. The bursting behavior of neuron **n1** can easily be seen in the top simulation trace.

The weakly inhibitory summing synapse **sn1n1** from neuron **n1** to itself causes the firing rate of neuron **n1** to decrease after the beginning of the burst. In this case, the amount of inhibition grows gradually with each spike, leading to a gentle adaptation of the firing rate during the burst. This synapse behavior is reminiscent of a calcium-dependent potassium channel. For this reason, the “synapse” symbol is often called a “generalized synapse”, since it can be used to model other useful behaviors that are not strictly synaptic in origin.

A Complex Example

Central Pattern Generators (CPGs) are groups of neurons that generate rhythmic firing patterns during repetitive motor tasks. Sylvie Ryckebusch at Caltech has used NeuroLOG and Spike in her modeling of the central pattern generator that controls locust flight. One of her early circuits is shown in Figure 3. The simulation of this circuit took 9 seconds on a Sun Sparc II workstation.

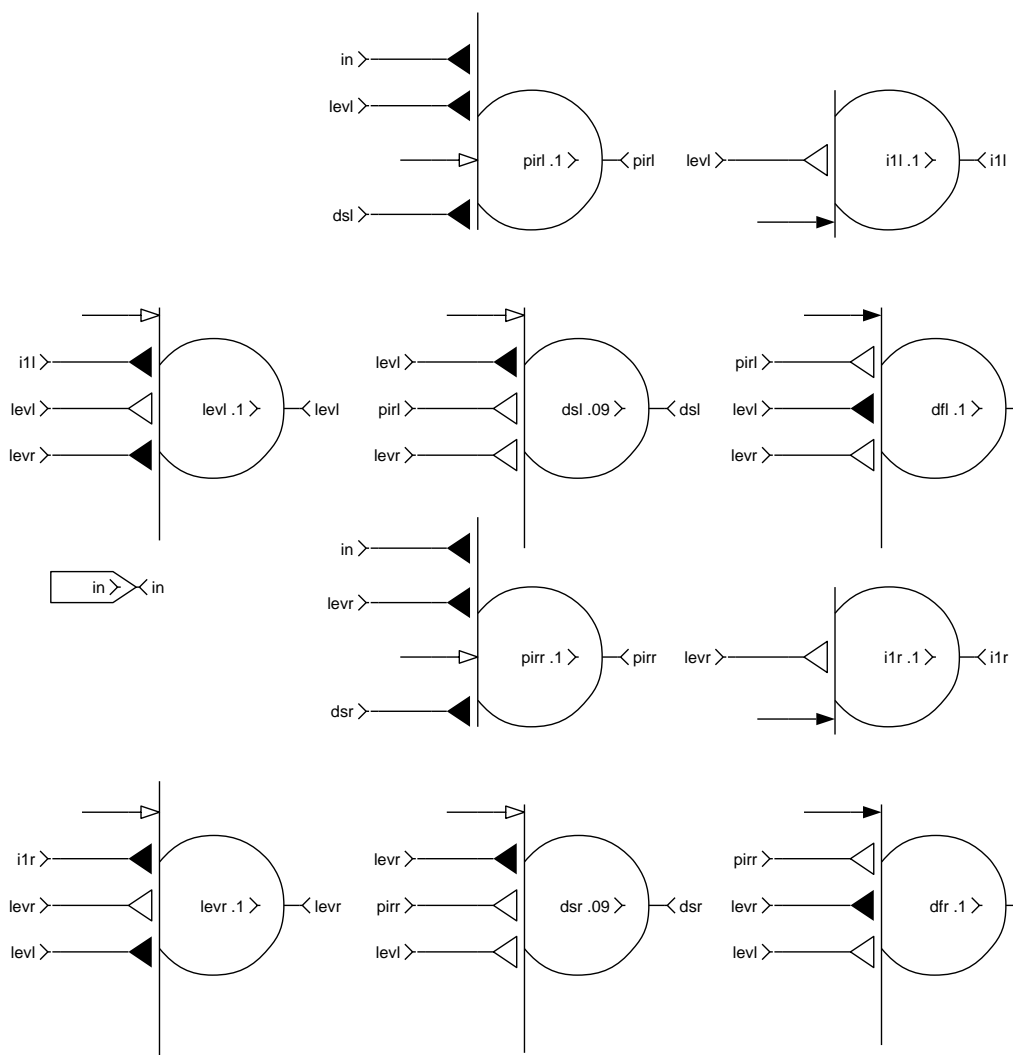


Figure 3 Sylvie Ryckebusch's locust CPG circuit. For clarity, the synapse parameters have been omitted.

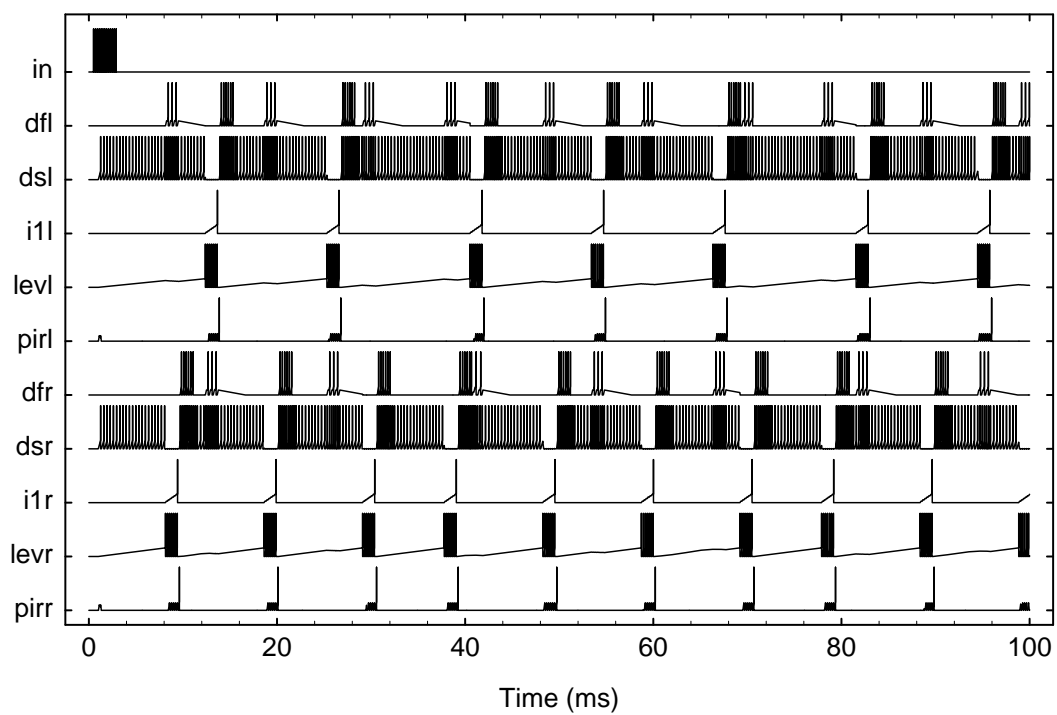


Figure 4 Simulation results for Sylvie Ryckebusch's locust CPG circuit. Only the neuron and input traces are shown.

How to Get NeuraLOG and Spike

NeuraLOG and Spike are distributed at no charge under the GNU General Public Licence. They are distributed in the hope that they will be useful, but *without any warranty*. The GNU General Public Licence is included in the distribution.

It is assumed that you already have AnaLOG and either Mathematica, Cview, or Gnuplot running on your system, and that you are familiar with their use. If not, you may obtain AnaLOG and Cview by anonymous ftp to `hobiecat.pcmp.caltech.edu`. The files reside in the directory `/usr/ftp/pub/chipmunk`. Look at the file README for instructions on how to install AnaLOG and Cview. AnaLOG and Cview are graciously supported and maintained by John Lazzaro. AnaLOG and Cview are provided with on-line documentation and tutorial examples.

Mathematica can be obtained by contacting Wolfram Research, Inc., at `info@wri.com`. Gnuplot can be obtained by anonymous ftp to `gnu.ai.mit.edu`, as can gcc, the recommended compiler for AnaLOG and Cview.

AnaLOG and Cview are supported on a wide variety of platforms, including HP and Sparc workstations. Spike is known to compile with cc on both of these workstation types, and probably will compile on others. I will be glad to provide support in porting Spike to other platforms.

For non-AnaLOG users, Spike will accept input in a convenient textual form.

If you would like to receive the software release and installation instructions, please send email to `lloyd@pcmp.caltech.edu`.

Lloyd Watts