ELSEVIER

Journal of Neuroscience Methods 136 (2004) 23–32

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

A computational tool to simulate correlated activity in neural circuits

F.J. Veredas[a,*], F.J. Vico[a], J.M. Alonso[b]

[a] *Departamento de Lenguajes y Ciencias de la Computación, ETSI Informática, Universidad de Málaga, Bulevar de Louis Pasteur s/n, E-29071 Málaga, Spain*

[b] *State University of New York (SUNY-Optometry), Department of Biological Sciences, 33 West 42nd Street, New York City, NY 10036, USA*

Received 30 June 2003; received in revised form 12 December 2003; accepted 22 December 2003

Abstract

A new computational approach to study correlated neural activity is presented. Simulating Elementary Neural NEtworks for Correlation Analysis (SENNECA) is a specific-purpose simulator oriented to small circuits of realistic neurons. The model neuron that it implements can reproduce a wide scope of integrate-and-fire models by simply adjusting the parameter set. Three different distributions of SENNECA are available: an easy-to-use web-based version, a Matlab (Windows and Linux) script, and a C++ class library for low-level coding. The main features of the simulator are explained, and several examples of neural activity analysis are given to illustrate the potential of this new tool.

Keywords: Correlated firing; Synchrony; Correlogram; Monosynaptic; Connection

1. Introduction

Most neural networks simulators are general-purpose environments designed for a wide range of modeling prerequisites. This generality is convenient, since the needs of the neuroscientific community are wide, but it also makes some of these simulators inappropriate for studies where the computational load is high, or a big effort has to be invested in learning a language or a complex environment.

Some simulators offer the neuroscientist the possibility of implementing highly realistic models of the membrane electrical behavior, with multicompartmental neuron models like in GENESIS (Bower and Beeman, 1998; The GENESIS Simulator, 2003), NEURON (Hines and Carnevale, 1997; Neuron web site at Duke, 2003), NODUS (Nodus software info, 1999), SNNAP (SNNAP—Simulator for Neural Networks and Action Potentials, 2003), XNBC (XNBC, v8, 2001) and NEMOSYS (Eeckman et al., 1993), among the most popular tools. However, these models are computationally very costly, and networks of several neurons simulated for long periods demand an extraordinary computational power, not available to a wide scope of the neuroscientific community. In particular, tasks that involve analysis of correlated activity with cross-correlograms imply long simulations, since the neurons under study must generate enough spikes for the correlogram shape to emerge from background activity. In these cases, dedicated tools clearly outperform popular simulators. On the other hand, some simulators model the neuron at a high level of abstraction [like Neurograph (Neurograph home page, 2003), and NSL (Weitzenfeld et al., 2002; Neural Simulation Language, 2003], but they are not realistic enough for correlation studies. Other simulators, like the SYNOD/NEST project (Diesmann and Gewaltig, 2002) or the software described in Morrison et al. (2003), focus on large networks of spiking neurons and correlation studies thereof. While these simulators can reproduce the type of experiments we address here, their usage requires considerable training and no graphical interface is provided.

In this paper we describe SENNECA (The SENNECA Simulator, 2003), a specific-purpose neural simulator dedicated to the study of pairwise correlated activity. This user-friendly tool is distributed in three different formats: web-based, for quick, installation-free experiments, Matlab® script, for specific, time-consuming simulations (run on the user's computing set up), and as a C++ class library, when massive simulations are required (e.g. searching of connectivity or physiological settings that fit specifically shaped correlograms). SENNECA is based on a highly parameterized neuron model to allow the independent study of a wide number of physiological factors, and also to account

[*] Corresponding author. Tel.: +34-952-137155; fax: +34-952-131397.
E-mail address: fvn@geb.uma.es (F.J. Veredas).

0165-0270/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2003.12.026

for most variants of realistic neuron models (leaky versus non-leaky, instantaneous integration versus EPSP-based, etc.

The description starts with an explanation of the formal neuron, then goes to a detailed presentation of the tool features, and concludes with a representative number of examples that illustrate the adequacy of this simulator in analyzing correlated activity.

2. Methods

The structure of a simulation has been divided in four sets of parameters that allow the experimenter to flexibly define (1) the network topology, (2) the cellular electrical behavior, (3) the synaptic transmission properties, and (4) the simulator environmental setting.

2.1. Network parameters

A network of neurons is defined by specifying the number of neurons, and the synaptic strength of the connections among them [given by ω_{ij} in Eq. (2)], that can take positive or negative values for excitatory and inhibitory synapses, respectively.

2.2. Neuronal parameters

The model neuron embodied in SENNECA is a general-purpose integrate-and-fire model (Stein, 1967). It has been designed according to two main objectives: (1) to allow the simulation of a wide range of models by setting parameters to proper values, and (2) to provide the user with enough degrees of freedom to independently study the influence of physiological characteristics in pairwise analysis of neural activity.

The electrical properties of the neuron membrane are expressed in Eq. (1), parameterized by the following constants:

- resting potential (υ): constant value in millivolts where the membrane potential converges due to the leaky effect,
- threshold potential (θ): membrane potential in millivolts for spike initiation,
- afterhyperpolarization (ν): it determines the membrane potential after the spike, given as a percentage of the underthreshold membrane potential just before spike initiation,
- membrane time constant (τ): it determines how long a postsynaptic potential will last, given in tenths of a millisecond,
- refractory period (ρ): time in tenths of a millisecond after the spike during which the threshold does not operate (although the neuron can integrate synaptic inputs).

The behavior of the dendritic tree depends on four parameters:

- background noise: the dendritic input is modeled as a current [referred to term *r* in Eq. (1)] obtained from a distribution selected from a list of different noise sources: uniform noise, Gaussian noise, 1/f noise, Poissonian noise (implemented by synaptic failures as in Zador, 1998), and custom noise (previously generated and stored in a file). Each noise source has different parameters. Uniform noise follows a uniform random distribution in an interval given by a minimum and a maximum value. Gaussian noise is based on a normal random distribution parameterized by the mean and standard deviation. 1/f noise has been implemented using the Voss–McCartney pink noise generation method, and the scale and shift values must be specified. For Poissonian noise five parameters are necessary: number of axons reaching a neuron; number of contacts per axon; inter-spike interval mean rate (Poissonian distribution); probability of neurotransmitter release in a synaptic contact (Zador, 1998); and maximum amplitude of each post-synaptic current. Finally, the simulator can read noise from a file, that can also be scaled and shifted. (The file generated by the user should have a time length equal to or longer than the simulation time. It should be in ASCII format with each entry representing the current (mA) at one simulation step (0.1 ms). The name of the file cannot have more than 255 characters),
- post-synaptic current rise time (α): time in tenths of a millisecond for the post-synaptic current to reach its maximum amplitude, modeled as an alpha function. It directly affects the rise time of the PSP (postsynaptic potential),
- post-synaptic current decay time (β): time in tenths of a millisecond for the post-synaptic current to decay. It is also modeled as an alpha function with an independent decay factor. It influences the PSP duration time,
- post-synaptic current amplitude (ω): modulates the amount of the dendritic PSC (postsynaptic current). It is also related with the synaptic efficacy between pairs of neurons and the connection strength. It is defined in milliamps,
- axonal transmission delay and synaptic jitter: spike latency and jitter have been modeled as the mean and the standard deviation of a gaussian random distribution, respectively. They are defined in tenths of a millisecond.

The model that interrelates all these magnitudes is expressed by three equations that govern the injected synaptic current [see Eqs. (1) and (2)], and the membrane potential (Eq. (3)).

$$I_i(t) = r_i(t) + \sum_j E_{ji}(t - \hat{t}_j - D_{ji}) \tag{1}$$

$$E_{ij}(u) = \begin{cases} \omega_{ij}\dfrac{u}{\alpha_j}e^{-u/\alpha_j} & \text{if } u \leq \alpha_j \\ \omega_{ij}\dfrac{\beta_j - \alpha_j + u}{\beta_j}e^{-(\beta_j-\alpha_j+u)/\beta_j} & \text{if } u > \alpha_j \end{cases} \tag{2}$$

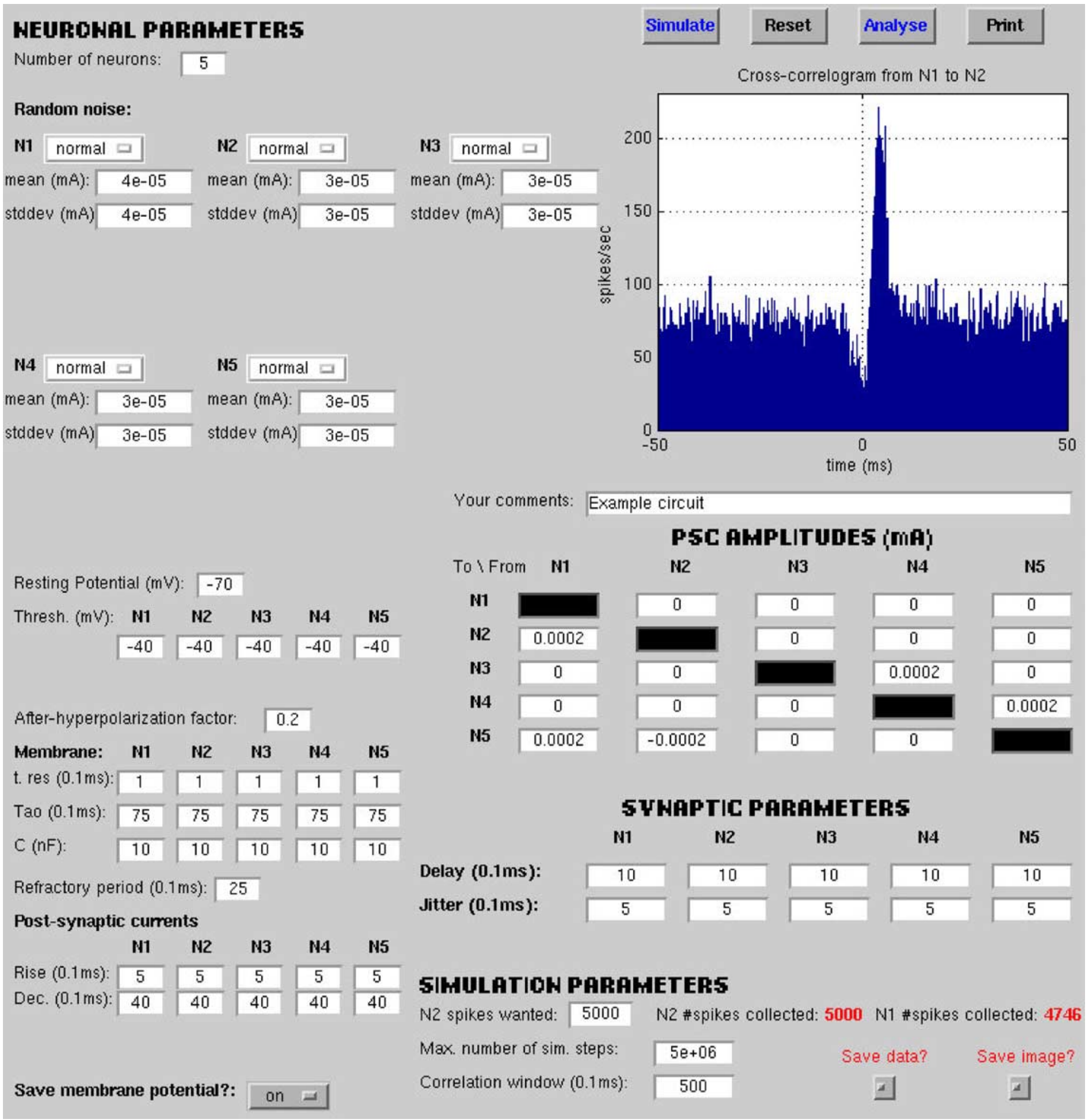
$$V_i(t+\Delta t) = \begin{cases} \nu V_i(\hat{t}_i) & \text{if } t \le \rho + \hat{t}_i \\ \left(1-\dfrac{\Delta t}{\tau_i}\right) V_i(t) + \dfrac{\Delta t}{C} I_i(t) + \dfrac{\Delta t}{\tau_i}\upsilon & \text{if } t > \rho + \hat{t}_i \end{cases} \tag{3}$$

where $\hat{t}_j$ is the simulation step where neuron j fired last time, and D_{ji} is the overall delay of the last spike in neuron j to reach neuron i, where both, constant transmission delay and variable synaptic jitter, contribute. C is de capacity of the membrane. Some parameters are not expressed in this formulation for the sake of simplicity. Just a couple aspects are worth mentioning: the axonal delay and jitter are implemented by a buffer (with a length equal to the maximum delay) where previous membrane states are stored, and the duration of the injected current is implemented with a counter.

Eq. (1) has been formulated in a time discrete manner, by solving the current based membrane equation on a fixed time grid of 0.1 ms (Rotter and Diesmann, 1999; Diesmann et al., 2001), since this is the highest temporal resolution imposed by the jitter.

2.2.1. Simulation parameters

The environment is also parameterized to define the conditions under which the simulation will take place. Two halting conditions are provided: the number of spikes in the target neuron for a correlogram to be significant, and the maximum simulation time, given in simulation steps. The output shape is also affected by a different parameter, the correlogram window size.



NEURONAL PARAMETERS

Number of neurons: 5

Random noise:

N1 normal, mean (mA): 4e-05, stddev (mA) 4e-05

N2 normal, mean (mA): 3e-05, stddev (mA) 3e-05

N3 normal, mean (mA): 3e-05, stddev (mA) 3e-05

N4 normal, mean (mA): 3e-05, stddev (mA) 3e-05

N5 normal, mean (mA): 3e-05, stddev (mA) 3e-05

Resting Potential (mV): -70

	N1	N2	N3	N4	N5
Thresh. (mV):	-40	-40	-40	-40	-40

After-hyperpolarization factor: 0.2

Membrane:	N1	N2	N3	N4	N5
t. res (0.1ms):	1	1	1	1	1
Tao (0.1ms):	75	75	75	75	75
C (nF):	10	10	10	10	10

Refractory period (0.1ms): 25

Post-synaptic currents

	N1	N2	N3	N4	N5
Rise (0.1ms):	5	5	5	5	5
Dec. (0.1ms):	40	40	40	40	40

Save membrane potential?: on

Your comments: Example circuit

PSC AMPLITUDES (mA)

To \ From	N1	N2	N3	N4	N5
N1	■	0	0	0	0
N2	0.0002	■	0	0	0
N3	0	0	■	0.0002	0
N4	0	0	0	■	0.0002
N5	0.0002	-0.0002	0	0	■

SYNAPTIC PARAMETERS

	N1	N2	N3	N4	N5
Delay (0.1ms):	10	10	10	10	10
Jitter (0.1ms):	5	5	5	5	5

SIMULATION PARAMETERS

N2 spikes wanted: 5000 N2 #spikes collected: 5000 N1 #spikes collected: 4746

Max. number of sim. steps: 5e+06

Correlation window (0.1ms): 500

Save data? □ Save image? □

Fig. 1. Main interface window for the Matlab version of SENNECA. Parameters can be fed in the corresponding cells, and the correlogram from N1 to N2 is displayed.

2.3. Description of the software tool

The web-based and the Matlab versions are equipped with friendly graphical interfaces for data feeding and output inspection. We provide next some general guidelines of this interface, since many of the tools are self-explanatory.

2.3.1. Data input

We illustrate the setting of parameters with the Matlab version of the simulator, since everything is compiled on a single window (the web-based version shows the same functionality, but it is structured in several pages). Fig. 1 shows such window for a specific experiment. Network size and background activity are defined in the left side of the interface, specifying the number of neurons (up to five neurons) and the average and standard deviation of the dendritic noise for each neuron. Physiological parameters follow, like resting potential, afterhyperpolarization and refractory period (parameters that are shared by all neurons), threshold, decaying potential and synaptic injected current for each individual neuron. Spike transmission is detailed at the middle-right side of the interface, given by the axonal delay and the limits of the uniformly distributed synaptic jitter. On top of this set of parameters, the strengths of the connections among the neurons are specified. The simulation parameters are fed at the bottom-right corner of the window: number of postsynaptic spikes to build the correlogram, maximum time length for the simulation, and correlogram window size. In addition to these features, a general comment of the nature of the experiment can be stored in an annotation field.

Apart from the main sets of parameters that specify the simulated experiment, a number of buttons are also associated to different tasks: simulation start (Simulate), extended report of the simulation (Analysis), print the interface window (Print), reset the parameter values (Reset), save to file the configuration and numeric results (Save data), save a Postscript image of the window (Save image), and store membrane potential traces in ASCII format for further analysis (Save membrane potential).

2.3.2. Simulation results

The window in Fig. 1 also allows the visualization of the main results of the simulation: the correlogram from neuron N1 (trigger) to neuron N2 (target), as well as the number of

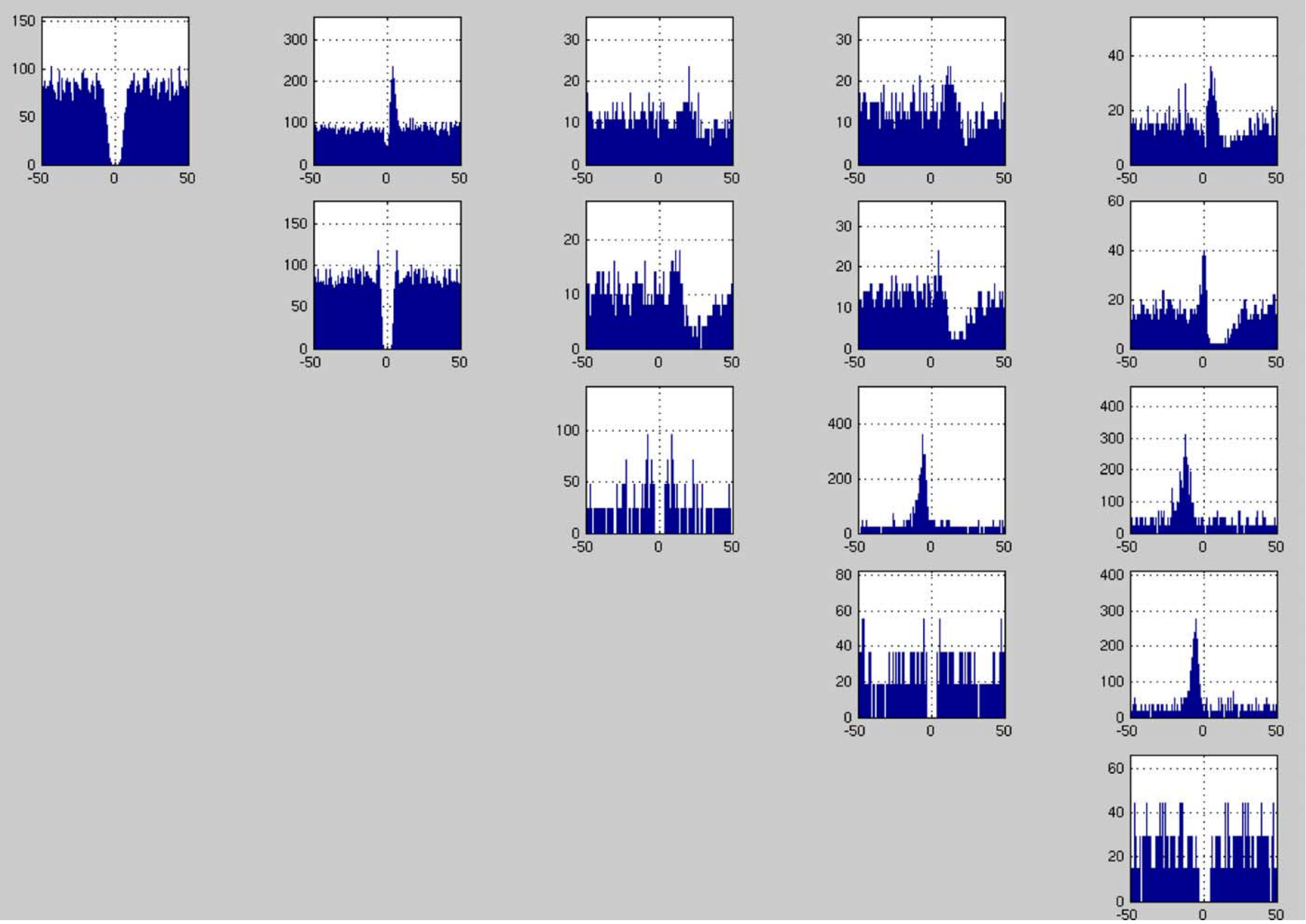


Fig. 2. Correlograms for all pairs of neurons in the experiment of Fig. 1. Each correlogram represents the combined activity of a trigger neuron (row) with a target neuron (column). The central bin of auto-correlograms (figures in the diagonal) has been deleted for visualization purposes.

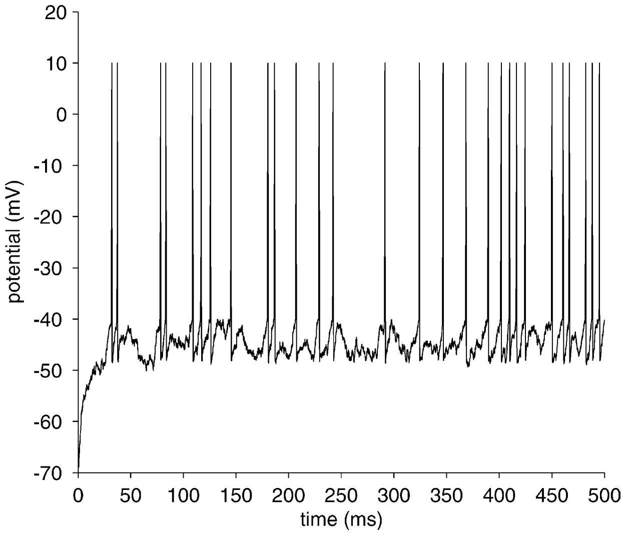


Fig. 3. Membrane potential for neuron N2 in the experiment of Fig. 1 as simulated by Eq. (1).

spikes collected during the simulation (bottom-right). Also, the Analysis button triggers the generation of an extended report to better understand the experiment. This report includes: (1) the visualization of the correlogram for each pair of neurons in the network (Fig. 2), (2) the evolution of membrane potential for each neuron or group of neurons in a single chart (Fig. 3), and (3) a graph that schematizes the network connectivity (Fig. 4).

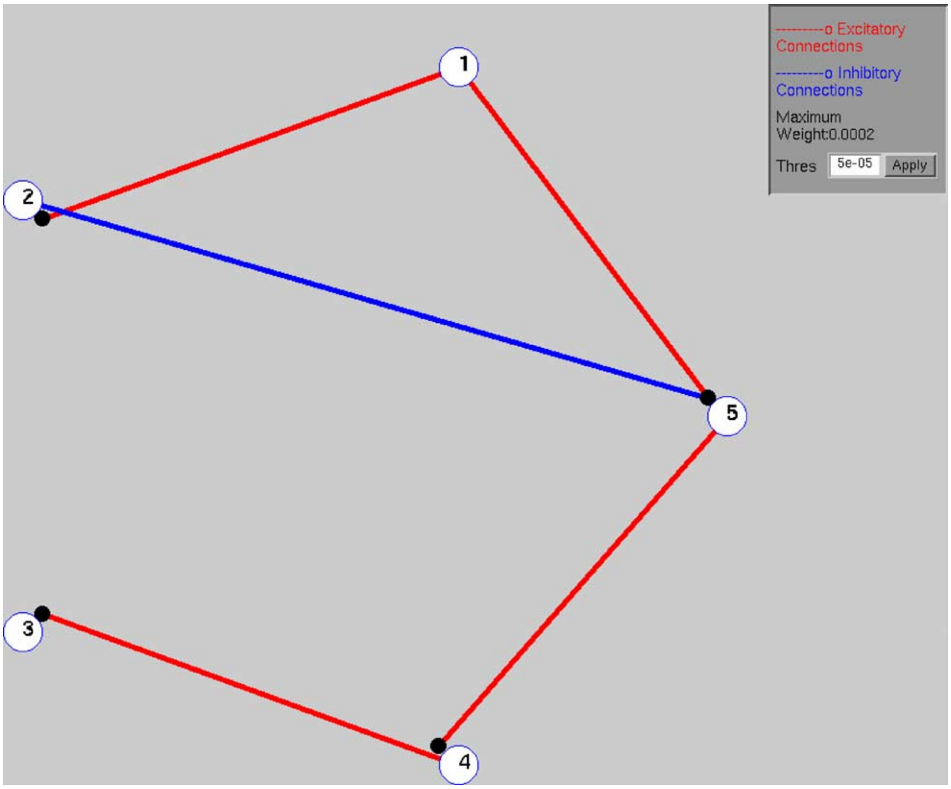
All this information helps to understand the shape of the reference correlogram (N1 to N2) in a specific simulation. In addition, the web version produces a detailed report of the parameters and correlograms obtained, accessible from the simulations database, and also attached to the message that is sent to the e-mail address given by the experimenter, including the correlograms (ASCII files) as well as the figures (Portable Network Graphics format).

As an additional feature, the web version allows interactive control of the simulation state (convenient for time consuming experiments), where the time left to finish is displayed, and the simulation state can be altered by the experimenter (hold, stop, and continue simulation). Also, the simulations are stored in a database that can be easily accessed.

2.3.3. *Software/hardware requirements*

SENNECA has been implemented in ANSI/ISO C++. The simulator is freely distributed (http://SENNECA.geb.uma.es) in three different formats: (1) as downloadable opened code, (2) with a graphical user interface in Matlab®, and (3) as a web-based program.

The simulator library has been developed in C++ and can be compiled with GCC (GNU general public license) version 2.95. It can be done under Linux, and also Microsoft


---------o Excitatory Connections
---------o Inhibitory Connections
Apply
1
2
3
4
5

Maximum Weight:0.0002

Thres 5e-05

Fig. 4. Network scheme of the experiment in Fig. 1. Labeled circles represent neurons. Excitatory connections are represented by solid lines and inhibitory connections are represented by broken lines. The relative strength of the connection is represented by the width of the line. A threshold can be applied to show only connections that are stronger than a specified value. Synapses are represented by solid dots.

Windows® using MinGW with GCC 2.95. GNU Scientific Library (GSL) version 1.0 is required to import the routines for random numbers generation.

The simulator is equipped with two different graphical user interfaces. Parameter setting, running simulations, and visualization of resulting data are the main features of the interface. One version of the interface runs under Matlab© 5.3 or higher (with Signal Processing Toolbox 4.2 or higher). Also, a web version is provided in order to ease program portability. Both show the same functionality, although, of course, the Matlab version runs on the experimenter's computer, while the web version runs remotely on a heterogeneous cluster of computers supported by Mosix 7.0. The web version is compatible with most web browsers (Explorer 5.0, Mozilla 5.0, Netscape 7.0 and Opera 5.0). Access to the simulator resources requires opening an account by the experimenter, what can be done simply by contacting the site administrator from the home page.

3. Results

Since its introduction in the 1960s (Moore et al., 1966; Perkel et al., 1967), cross-correlation analysis has been used extensively to study functional connectivity in different parts of the brain (for reviews see Kirkwood, 1979; Vaadia and Aertsen, 1992; Reid, 2001). Although the technique is not free of limitations (e.g. Aertsen et al., 1989,1994; Brody, 1999), its use has become increasingly popular. In particular, there have been a series of recent papers from separate laboratories that have used this technique for the study of thalamocortical connections in different sensory systems (e.g. visual: Alonso et al., 2001; auditory: Miller et al., 2001; somatosensory: Bruno and Simons, 2002; Roy and Alloway, 2001; Swadlow and Gusev, 2002). This renaissance of the technique reflects a general belief in the scientific community that cross-correlation analysis can be reliably used to identify certain type of monosynaptic connections that are strong and excitatory. It is important to notice, however, that inhibitory connections are harder to identify with cross-correlation analysis (Aertsen and Gerstein, 1985) and the same is true for weak excitatory connections, particularly within the cortex. Aertsen et al. (1994) made an important distinction between structural (or anatomical) connectivity and functional coupling (or effective connectivity). Structural connectivity includes strong excitatory connections that can be reliably measured with cross-correlation analysis under different stimulus conditions. In contrast, effective connectivity refers to polysynaptic pathways or weak monosynaptic connections that can be only identified by cross-correlation analysis under certain stimulus/behavioral conditions (e.g. Vaadia et al., 1995; Aertsen et al., 1996). The computational tool presented here should help to better distinguish between these two types of connectivity and make more precise inferences about the physiological properties of strong structural connections.

To illustrate how pairwise activity can be studied with this computational tool we have performed six examples, where changes in neural physiology and network connectivity reproduce noticeable differences in the correlograms obtained. The purpose of these examples is to illustrate the properties of the simulator (the range of parameters used is not always within physiological values). The figures that illustrate each experiment have been imported from the graphical simulation results (correlograms in Portable Network Graphics format) produced by SENNECA, and included with little make up.

3.1. *Experiment #1: the correlogram peak can be shifted in position by changing the synaptic delay of a connection*

The network consists of two neurons (namely N1 and N2) with a monosynaptic excitatory connection from N1 to N2. Both neurons are given equal parameter values ($\upsilon = -70$ mV, $\theta = -40$ mV, $\tau = 7.5$ ms, $\rho = 2.5$ ms, $\nu = 0.2$, and $\alpha = 0.5$ ms, what gives an approximate EPSP rise time of 7.8 ms, and $\beta = 4.0$ ms, yielding an approximate EPSP duration at half amplitude of 16.5 ms). The connection strength (ω_{12}) is set to 2×10^{-4} mA, and the jitter is 0.5 ms. Both neurons were provided with normally distributed background activity with a mean and standard deviation of 4×10^{-5} mA for N1 and a mean and standard deviation of 3×10^{-5} mA for N2. Fig. 5 shows the resulting correlograms from N1 to N2 with increasing transmission delays (1 ms in **A**, 5 ms in **B**, and 10 ms in **C**). The simulation results show how the monosynaptic peak shifts to the right as the synaptic delay increases.

The time window size for the correlograms in Fig. 5 is 50 ms and each simulation generates 10 000 spikes in N2. The mean firing rates obtained are 62 spikes/s for N1 and 66 spikes/s for N2.

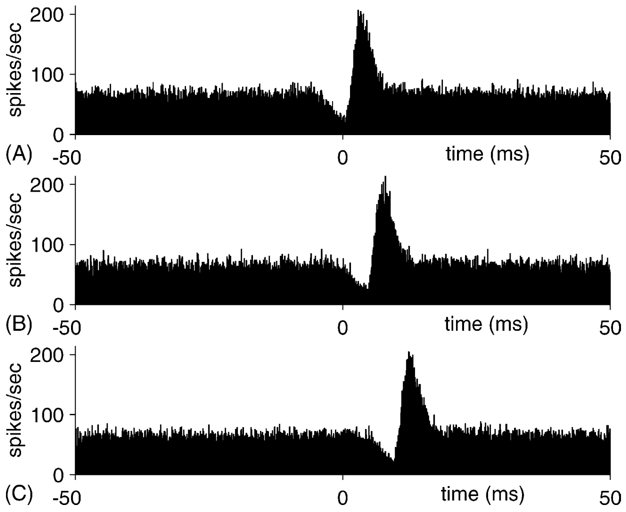


Fig. 5. Three excitatory monosynaptic correlograms from N1 to N2 are shown with different transmission delays. Simulations are run for (**A**) 1 ms latency, (**B**) 5 ms, and (**C**) 10 ms. As can be observed, different delays produce increasingly larger shifts in the position of the correlogram peak.

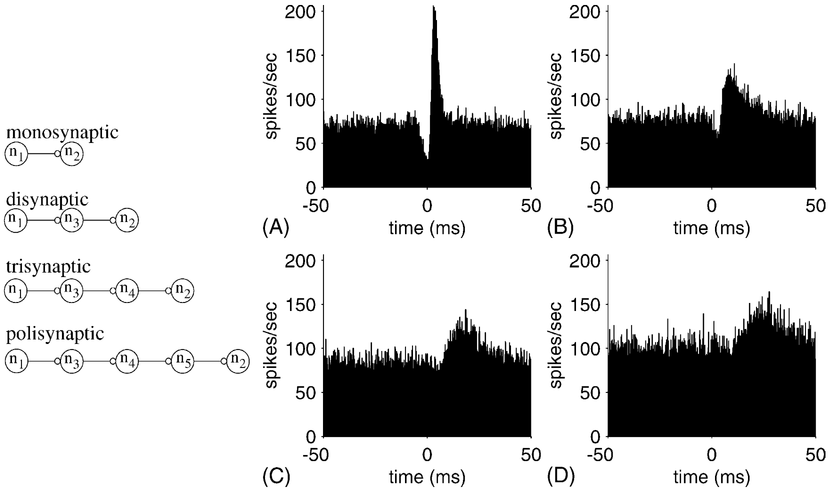


Fig. 6. Resulting correlograms (from N1 to N2, see circuit schemes on the left) of four simulated circuits with different number of interconnecting neurons: (**A**) monosynaptic, (**B**) disynaptic, (**C**) trisynaptic, and (**D**) polysynaptic. The amplitude correlates inversely with the number of interneurons, while peak's width and shift are directly correlated.

3.2. Experiment #2: the number of serial synaptic connections that separate two neurons affects the amplitude, width and delay of the correlogram peak

This experiment is concerned with the number of excitatory interneurons that separate N1 from N2. The simulated neurons were given the same basic parameters used in experiment #1 (synaptic strength = $2 \times 10^{-}4$ mA, delay = 1 ms, and jitter = 0.5 ms). Background activity also resembles that of experiment #1: neuron N1 is fed with a mean (and standard deviation) noise of 4×10^{-5} mA, while neurons N2 to N5 were provided with 3×10^{-5} mA mean and the standard deviation. Four circuits were simulated, starting with the monosynaptic connection and adding new relay neurons, up to three interneurons (polysynaptic connection). Fig. 6 shows the correlograms obtained from N1 to N2, after 10 000 spikes were collected in N2. **A** reproduces the stereotyped correlogram of a monosynaptic connection. Correlograms **B–D** are obtained for the circuits on the left (disynaptic, trisynaptic and polysynaptic, respectively). As the number of serial interconnected neurons increases, the correlogram peak decreases in amplitude, becomes wider and shifts towards the right.

3.3. Experiment #3: synaptic jitter affects the peak amplitude and width

In another experiment we modified the latency jitter of a monosynaptic connection. Again, the circuit (N1 excites N2) and parameters are the same as in experiment #1. Fig. 7 shows the correlograms (correlogram window size of 30 ms) from N1 to N2 (with 10 000 spikes in N2) simulated with different levels of synaptic jitter (0 ms in **A**, 1 ms in **B**, 2 ms in **C** and 3 ms in **D**). As the jitter increases, the monosynaptic peak becomes wider and smaller in amplitude. Also, there is a slight shift in the peak towards the right.

3.4. Experiment #4: inhibitory connections alter the correlogram symmetry

In experiment 4 a single neuron (N3) provided common input to neurons N1 and N2 (connection strength from N3 to N1 and N2: 3×10^{-4} mA). The setting again reproduced that of experiment #1, although here N2 inhibits N1 and all synaptic jitters have been set to 0.1 ms. The parameter under study in this experiment is the strength of the inhibitory connection (amplitude of PSC from N2 to N1, ω_{21}). Fig. 8 shows the resulting correlograms, after N2 fired 10 000 times, for a PSC amplitude of 0 mA (**A**) (reference common-input circuit), -1×10^{-4} mA (**B**), -2×10^{-4} mA

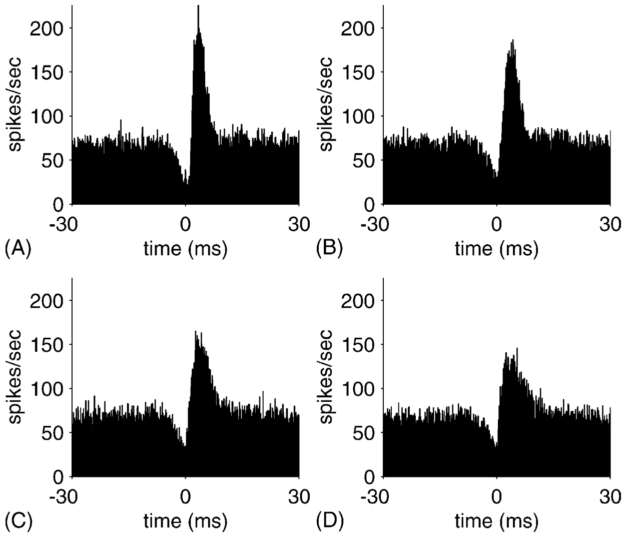


Fig. 7. Four excitatory monosynaptic connections were simulated with different latency jitter. Resulting correlograms are shown for a synaptic jitter of (**A**) 0 ms, (**B**) 1 ms, (**C**) 2 ms, and (**D**) 3 ms. As can be observed, when the jitter increases the peak becomes wider and with less amplitude.

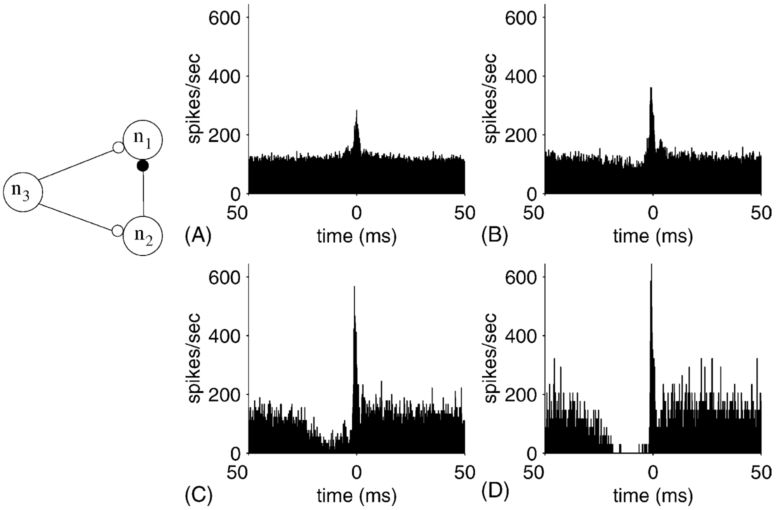


Fig. 8. Correlograms from N1 to N2 (see circuit scheme on the left) from simulations of a circuit with a common input (N3) and an inhibitory connection from N2 to N1. The PSC amplitude of the inhibitory synapse increases from 0 mA in (**A**), to -1×10^{-4} mA in (**B**), -2×10^{-4} mA in (**C**) and -3×10^{-4} mA in (**D**). The correlogram becomes more asymmetric and the amplitude of the peak generated by common input increases.

(**C**), and -3×10^{-4} mA (**D**). The correlogram becomes more asymmetric as inhibition becomes stronger. Also, the amplitude of the central peak increases.

3.5. Experiment #5: the correlogram peak becomes higher in amplitude as the EPSP rise time increases

In this experiment two monosynaptically connected neurons are simulated to study the influence of a variable EPSP rise time. Both neurons resemble the setting of experiment #1, although their membrane time constants (τ) have been reduced to 1 ms to produce realistic EPSPs in N2; moreover, the PSC decay time of N2 (β_2) has been set to 0.5 ms and the PSC from N1 to N2 amplitude is set to 4×10^{-4} mA. Also, the background synaptic noise currents have been adjusted for this simulation, using normal noise sources with mean 2.5×10^{-4} mA for N1 and 2.3×10^{-4} mA for N2 and standard deviation of 1×10^{-4} mA for both neurons. The post-synaptic current rise time in neuron N2 has been increased progressively (0.1 ms in **A**, 1 ms in **B**, 2 ms in **C**, and 3 ms in **D**) to obtain different EPSP rise times (1.1, 1.6, 2.4, and 3.3 ms, respectively). The correlograms reveal that the monosynaptic peak and baseline amplitudes become higher as the injected current rise time increases (Fig. 9).

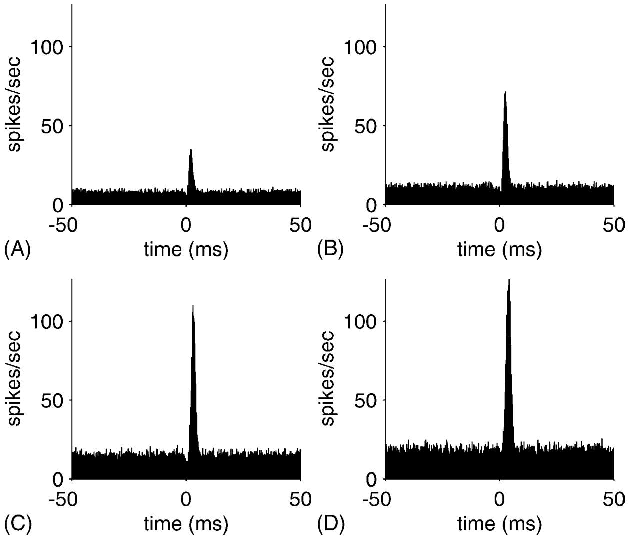


Fig. 9. Four excitatory monosynaptic correlograms from N1 to N2 are shown with different EPSP rise times. Simulations are run for (**A**) 1.1 ms EPSP rise time (PSC rise time = 0.1 ms), (**B**) 1.6 ms EPSP rise time (PSC rise time = 1 ms), (**C**) 2.4 ms EPSP rise time (PSC rise time = 2 ms), and (**D**) 3.3 ms rise time (PSC rise time = 3 ms). Correlograms' peak and baseline amplitudes increases with higher EPSP rise times.

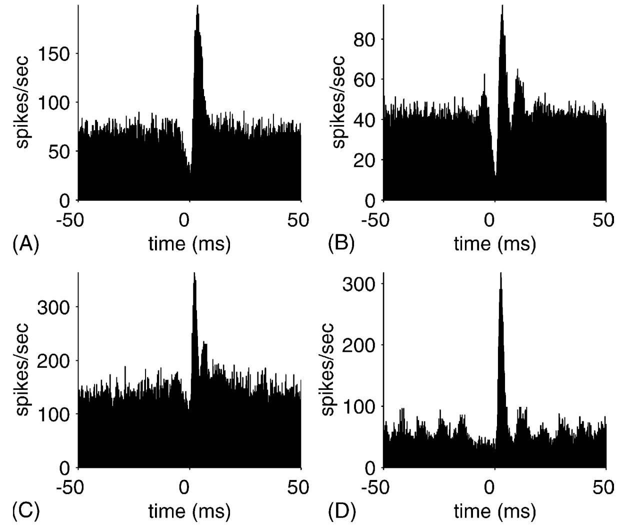


Fig. 10. Four excitatory monosynaptic correlograms obtained with different types of synaptic noise: Gaussian distribution (**A**); uniform distribution (**B**); 1/f noise (**C**); and Poissonian distribution of inter-spike interval with synaptic failures (**D**). The shape of the correlograms varies for the different noise classes.

Table 1
Simulation benchmarks for the web-based version of SENNECA simulator over a heterogeneous Mosix-based cluster of computers

# Simultaneous simulations	5		10		20		50		100	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Time per one simulation (sec)	14.76	1.71	23.24	3.03	40.93	5.86	59.32	25.14	107.04	42.85

Excitatory monosynaptic circuits (with the physiological parameters of experiment #1) have been simulated for 500 000 simulation steps each one, and have simultaneously been launched with different simulations pool sizes (5, 10, 20, 50, and 100 simultaneous simulations). Times for correlograms calculation have not been considered here.

3.6. Experiment #6: the shape of the correlogram is affected by the type of synaptic noise

In this experiment, two monosynaptically connected neurons are simulated to evaluate the influence of four different types of background synaptic noise on the shape of the correlogram. The physiological parameters of N1 and N2 are those of experiment #1, except for background synaptic noise settings. In Fig. 10, the correlograms represented have been obtained from simulations of the same monosynaptic circuit but using different noise parameters. In **A**, normal distributed random noise has been used, with mean and standard deviation of 4×10^{-5} mA for N1 and mean and standard deviation of 3×10^{-5} mA for N2. In **B**, a uniform distribution was used for noise sources to N1 and N2, with minimums of -1×10^{-5} mA for N1 and -2×10^{-5} mA for N2 and maximums of 1×10^{-4} mA for N1 and 6×10^{-5} mA for N2. In **C**, N1 and N2 was injected with 1/f noise sources (see 1/f implementation details above) with scale 7.905×10^{-6} and no shift for both neurons. Finally, in **D**, both neurons were injected with noise source currents following a poisson random distribution of inter-spike intervals with synaptic failures (Zador, 1998); the parameters of this noise type for both neurons were: 10 afferent axons, 100 contacts per axon, inter-spike interval rate of 9 ms, probability of neurotransmitter vesicles release of 0.85 and maximum amplitude of PSCs of 1×10^{-6} mA. As can be observed in Fig. 10, the shape of the correlograms varies for the four different classes of noise.

4. Conclusions

The drawbacks of general-purpose simulators in computing large real time experiments can be overcome by dedicated software. SENNECA is a computational tool that has been optimized to easily specify the neuron model (no scripts, nor new languages) by parameterizing a highly flexible model, that incorporate those physiological features most relevant to correlated activity. The different formats of graphical user interface allow the experimenter a straightforward input of the simulation parameter, as well as a complete report of the results. The examples related in the results section illustrate the competence of SENNECA to emulate, in a computational way, the influence of membrane physiology and network connectivity in correlated firing.

SENNECA subscribes to the trend of current programs dedicated to intense computation, by providing a front-end that allows the operation on remote machines. Like in webMathematica (WebMathematica: Add Dynamic Computations and Visualizations to Your WebSite, 2002) the web-based version of SENNECA is intended to support high cost computation, freeing the user's computer. The cluster that runs SENNECA can perform the simulations of about 30 users in real time (see Table 1), what makes it suitable for courses of Physiology or Computational Neuroscience where groups of students simulate correlated activity in an interactive way. In case the computational load increases, a scheme of cluster reservation would apply, so the groups can organize in advance the use of the simulator.

The Matlab-script version of SENNECA simulator has been tried locally over a modern PC (Intel® P IV, 1.8 GHz) giving a simulation execution mean time (averaged over 100 independent simulations) of 12.3 s for 500 000 simulation steps (with the physiological parameters of experiment #1).

Future versions of this simulator will incorporate other analytical tools that can be useful to better understand how neural networks generate correlated firing.

Acknowledgements

We specially thank a detailed review on the manuscript by Markus Diesmann, who made us understand many aspects of the work. Likewise the contribution of two anonymous referees is acknowledged. The web-version has benefited much from the help of our web-designer Juan Miguel Taboada. This research has been supported by NIH-EY 05253, and by the Spanish Ministerio de Ciencia y Tecnología (project PTR95.0734.OP).

References

Aertsen AM, Gerstein GL. Evaluation of neuronal connectivity: sensitivity of cross-correlation. Brain Res 1985;340(2):341–54.

Aertsen AMHJ, Gerstein GL, Habib MK, Palm G. Dynamics of neuronal firing correlation: modulation of "effective connectivity". J Neurophysiol 1989;61:900–17.

Aertsen A, Erb M, Palm G. Dynamics of functional coupling in the cerebral cortex: an attempt at a model-based interpretation. Physica D 1994;75:103–28.

Aertsen A, Erb M, Haalman I, Vaadia E, Coherent dynamics in the frontal cortex of the behaving monkey – Experimental observations and model interpretation. In: Gath I, Inbar GF, editors, Advances in Processing and Pattern Analysis of Biological Signals. New York, London: Plenum Press, 1996, pp. 205–24.

Alonso JM, Usrey WM, Reid RC. Rules of connectivity between geniculate cells and simple cells in cat primary visual cortex. J Neurosci 2001;21:4002–15.

Bower JM, Beeman D. The Book of GENESIS. Exploring Realistic Neural Models. Springer-Verlag: New York, 1998.

Brody CD. Correlations without synchrony. Neural Comput 1999;11:1537–51.

Bruno RM, Simons DJ. Feedforward mechanisms of excitatory and inhibitory cortical receptive fields. J Neurosci 2002;22:10966–75.

Diesmann M, Gewaltig M-O. NEST: An environment for neural systems simulations. In: Macho V, editor, Forschung und wissenschaftliches Rechnen, GWDG-Bericht. Göttingen: Ges. für Wiss. Datenverarbeitung. ISSN 0176-2516.

Diesmann M, Gewaltig, MO. NEST: An environment for neural systems simulations. In: Macho V, editor, Forschung und wissenschaftliches Rechnen, GWDG-Bericht. Göttingen: Ges. für Wiss. Datenverarbeitung, 2002.

Diesmann M, Gewaltig MO, Rotter S, Aertsen A. State space analysis of synchronous spiking in cortical neural networks. Neurocomputing 2001;38:565–71.

Eeckman FH, Theunissen FE, Miller JP. NeMoSys: A neural modeling system. In: Schwetman HD, Walrand JC, Bagchi KK, DeGroot D, editors. Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, 1993, p. 365–6.

Hines ML, Carnevale NT. The NEURON simulation environment. Neural Comput 1997;9:1179–209.

Kirkwood PA. On the use and interpretation of cross-correlations measurements in the mammalian central nervous system. J Neurosci Methods 1979;1:107–32.

Miller LM, Escabi MA, Read HL, Schreiner CE. Functional convergence of response properties in the auditory thalamocortical system. Neuron 2001;32:151–60.

Moore GP, Perkel DH, Segundo JP. Statistical analysis and functional interpretation of neuronal spike data. Annu Rev Physiol 1966;28:493–522.

Morrison A, Mehring C, Diesmann M, Aertsen A, Geisel T, 2003. Distributed simulation of large biological neural networks. In Proceedings of the Twenty Ninth Göttingen Neurobiology Conference, p. 590.

Neural Simulation Language. Retrieved 2000 June 18 from http://www.neuralsimulationlanguage.org, n.d.

Neurograph home page. Retrieved 2003 June 18 from http://www2.informatik.uni-erlangen.de/IMMD-II/Research/Activities/NeuroGraph/, n.d.

Neuron web site at Duke. Retrieved 2003 June 18 from http://kacy.neuro.duke.edu/, n.d.

Nodus software info. Retrieved 2003 June 18 2003 from http://bbf-www.uia.ac.be/NODUS/NODUS_info.html, 1999.

Perkel DH, Gerstein GL, Moore GP. Neuronal spike trains and stochastic point processes. Part II. Simultaneous spike trains. Biophys J 1967;7:419–40.

Reid RC. Divergence and reconvergence: multielectrode analysis of feed forward connections in the visual system. Prog Brain Res 2001;130:141–54.

Rotter S, Diesmann M. Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. Biol Cybernet 1999;81:381–402.

Roy SA, Alloway KD. Coincidence detection or temporal integration? What the neurons in somatosensory cortex are doing. J Neurosci 2001;21:2462–73.

SNNAP—Simulator for Neural Networks and Action Potentials. Retrieved 2003 June 18 from http://nba19.med.uth.tmc.edu/snnap/, 2003.

Stein RB. Some models of neuronal variability. Biophys J 1967;7:37–68.

Swadlow HA, Gusev AG. Receptive-field construction in cortical inhibitory interneurons. Nat Neurosci 2002;5:403–4.

The GENESIS Simulator. Retrieved 2003 June 18 from http://www.genesis-sim.org/GENESIS/, 2003.

The SENNECA Simulator. Retrieved 2003 June 18 from http://SENNECA.geb.uma.es/, 2003.

Vaadia E, Aertsen A, Coding and computation in the cortex: single-neuron activity and cooperative phenomena. In: Aertsen A, Braitenberg V, editors, Information Processing in the Cortex: Experiments and Theory. Berlin, Heidelberg, New York, Tokyo: Springer, 1992, pp. 81–121.

Vaadia E, Haalman I, Abeles M, Bergman H, Prut Y, Slovin H, Aertsen A. Dynamics of neuronal interactions in monkey cortex in relation to behavioral events. Nature 1995;373:515–8.

WebMathematica: Add Dynamic Computations and Visualizations to Your WebSite Retrieved 2003 June 30 from http://www.wolfram.com/products/webMathematica/, 2002.

Weitzenfeld A, Arbib M, Alexander A. The Neural Simulation Language A System for Brain Modeling. MIT Press, 2002.

XNBC, version 8. Retrieved 2003 June 18 from http://www.b3e.jussieu.fr/xnbc/, 2001.

Zador A. Impact of synaptic unreliability on the information transmitted by spiking neurons. J Neurophysiol 1998;79(3):1219–29.