

XNBC: SIMULATING BIOLOGICAL NEURAL NETWORKS¹

Jean-François Vibert

CHU Saint-Antoine

Université Pierre et Marie Curie (Paris VI)

France

Abstract

XNBC is software to model and simulate biological neural networks. Four models of neurons are available: a phenomenological model, a leaky integrator, a conditional burster model, and a Hodgkin-Huxley type model with 14 different ionic currents. Inputs to the simulated neurons can be provided by data stored in files from actual experiments, allowing the making of 'hybrid' networks. The modelled neurons as well as the network are described using graphic tools. Neuron and network parameters can be modified during the simulation to mimic electrical stimulations and the action of drugs. After the simulation, the temporal behaviour of the network or of selected neurons can be visualised, and point process, frequential and dynamic analyses can be performed.

1. INTRODUCTION

Computational neuroscience aims to contribute to the understanding of living neural networks by using realistic models to predict behaviours, to test hypothesis, and to design experiments. Neural coding appears as a field where computational neuroscience can be particularly helpful. User-friendly simulation tools will help to develop this area. XNBC goes in this direction, since it is a fully featured application allowing the simulation of interacting living neural networks, designed for computer-naive neuroscientists. XNBC provides two simulation levels: one based on a phenomenological representation of the membrane potential dynamics, with three different models, and another based on the Hodgkin-Huxley formalism describing ionic channel conductance dynamics. XNBC allows the simulation of living networks formed by several interconnected neural nuclei connected through pathways of variable length, which can receive external inputs. Neuron parameters are selected and precisely tuned using a graphic editor which immediately displays the modification effects on neural behaviour. The network architecture, including the connection pattern and the inter-neuron transmission times, is also built using a graphic editor. Simulated neurons can receive inputs stored in a file, allowing 'hybrid' system simulations with inputs from experiments. During the simulation, electrical stimulations, periodic input, and background noise can be provided. The action of drugs can also be simulated when using the conductance-based model. Inputs, anatomical characteristics, and other properties of individual units and

synapses can be modified during the simulation to take into account changes in environmental conditions due to learning or pathological conditions (Vibert *et al* 1996).

The time evolution of network variables can be followed with several types of visualisations tools. XNBC provides a wide collection of analysis tools to study network behaviour, at both global and unitary levels, and in both frequency and temporal domains.

XNBC is mouse driven with a user-friendly XWindow/Motif interface and produces native colour PostScript high quality graphic outputs.

In this paper, we describe the general organisation of XNBC Version 8 (section 2), its components and objects (sections 3 and 4), how to use XNBC (section 5), its implementation (section 6) and comparison with others simulation tools (section 7).

2. PRESENTATION

XNBC is a simulation workstation for neurobiologists developed for research purposes. XNBC means Xwindow Neuro_Bio_Clusters. Xwindow is the Unix windowing system, Neuro_Bio is for biological neurons and Clusters is for the way neurons are grouped. This software is a package of different complementary cooperative tools created by neuroscientists who need for their own research to simulate some neural networks in order to test hypotheses.

XNBC allows one to build a neural network using neurons grouped at two levels: nuclei and clusters. A nucleus is a collection of neurons. A cluster is a group of neurons sharing the same membrane properties. The nuclei and clusters can be positioned in the network according to the Horsley Clarke coordinate system (in three dimensions), and the user can create intra and inter nuclei or cluster connections, according to his/her experimental protocol.

XNBC provides two neuron simulation levels: one level is called phenomenological, it is rather simple and fast, and includes membrane potential and threshold. At this modelling level, the temporal evolution of membrane potential and threshold are computed without considering the underlying phenomena leading to these evolutions. The other level is more sophisticated, but slower and based on conductance variations in different ionic channels. Here the temporal evolution of membrane potential is the consequence of the conductance variations, as proposed by Hodgkin and Huxley (1952). The notion of threshold is absent in this model.

Beside these explicitly simulated neuron models, previously recorded spike trains can be used and are considered as virtual neurons (because they are not simulated by XNBC). They could have been produced by either a

previous simulation or an actual experiment. The virtual model concept is powerful since it allows one to make hybrid networks out of simulated neurons receiving inputs from experimentally recorded neurons.

Neurons are grouped together into nuclei. One nucleus can include one or several neuron models (clusters). Perturbations (noise, stimulation, modulation, drugs), can be added either at the level of the nuclei or at the level of the clusters, allowing a high flexibility on the simulated experiment control.

XNBC computes the simulation results and provides different tools for visualisation and analysis of the results.

3. THE XNBC COMPONENTS

XNBC is made of several independent tools integrated by a program (xNBC) providing a control panel from which all the process can be controlled.

The XNBC simulation toolkit is thus made of several parts:

- a control program;
- two graphic neuron editors to adjust the neuron parameters, one for all the phenomenological models (PUM, LIM, BUM) and one for the conductance based model (CBM);
- two graphic network editors, one to build simple networks where one nucleus can contain only one cluster type and one to build complex and anatomically precisely tuned networks;
- the simulator itself allowing to mix PUM, LIM, BUM, CBM and virtual neurons;
- a visualisation tool to examine in several ways the simulation results;
- two analysis tools, a time series analysis tool with 31 different analyses and a network activity analysis tool with 8 different analyses.

A full manual of 150 printed pages is always available on-line by spawning a browser from each tool. The manual is written in html and hypertext links allow one to browse it easily. A second level of help is available on a topic basis.

4. THE XNBC OBJECTS

4.1 The neuron

The neuron is the basic element of XNBC. It represents a neural entity that has a specific physical type, and has a specified location. It is of course a real concept. The neurons are the basis of the neural activities. Each neuron is individually simulated and has its own evolution of parameters (membrane potential, ionic conductance, etc.). A neuron shares some basic properties with other neurons (we say that they pertain to the same cluster –

see below), but has its own life, different from other neurons. Each neuron can be anatomically positioned in the 3D space if needed.

4.2 The cluster

A cluster is an abstraction allowing one to describe simultaneously in one shot a large number of neurons. When describing the properties of a cluster, we describe the basic parameters of all the neurons (rest potential, rest threshold, membrane capacity, mean Na or K conductance, etc.) pertaining to this cluster. Many neurons can share the same basic properties, and then evolve on their own, as explained above. These neurons are said to pertain to the same cluster. The way the neuron is modelled (phenomenological or conductance based) is therefore a cluster property.

Five different ways of modelling can be chosen to describe the neuron and thus to constitute the clusters:

- the Phenomenological Unit Model of a neuron (PUM) is a phenomenological model with adaptation and post spike membrane shunt, and threshold fatigue;
- the Leaky Integrator Model of a neuron (LIM) is the classical simple leaky integrator;
- the Bursting Unit Model of a neuron (BUM) is a phenomenological model of a conditional burster with adaptation, post spike membrane shunt and two thresholds;
- the Conductance Based Model of a neuron (CBM) is a Hodgkin-Huxley like model with 14 different transmembranar currents;
- the Virtual (not simulated by the simulator) model is in fact spike firing dates stored in a file and coming from either a live experiment or a previous simulation.

The concept of a cluster has proven to be a very powerful concept to describe large sets of neurons and to group them. When only one cluster is used in a nucleus (see below), the cluster can be viewed as a nucleus.

4.3 The network

A network is made of one or several nuclei and/or one or several isolated neurons. Nuclei and neurons can be anatomically positioned if necessary. Nuclei and neurons are connected together by links representing the axons of constituting neurons (see below).

4.4 The nucleus

The nucleus is a concept introduced with XNBC V8.0. It is a convenience object to design a group of neurons (each belonging to a given cluster) that have the same location area, specified by a centre and a radius around this

centre. This concept introduces the spatial influence in the networks interactions and allows one to take into account:

- the anatomic location of neurons (the Horsley Clarke coordinates can be used);
- their connections according to the inter neuron distance;
- the connection pattern;
- the dissociation of anatomical location and unit characteristics;
- the neuromodulator or drug concentration according to the production or injection locus (in a future version of XNBC).

A nucleus is constituted by several neurons. These neurons can pertain to one or several clusters, and clusters can span several nuclei (since they are only a way to describe neuron behaviour, not the neuron location). When nuclei contain only one cluster, nucleus and cluster can be viewed as equivalent (in this case, the simple network editor can be used to build the network). Neurons inside nuclei can be connected together and to the other nuclei.

4.5 The connections or links

Neurons inside or between nuclei can be connected together. Connections can be either excitatory, inhibitory or with NMDA (long lasting excitation), or a mix of excitatory and inhibitory, called random connection. Inter-neural transmission of action potentials, called interneural delay (also referred as axon length) can be adjusted, as well as the number of synaptic boutons at the axon ending, called also synaptic weight. The connection matrix can be defined either globally or individually, neuron to neuron.

5. SIMULATING NEURAL NETWORKS USING XNBC

Efforts were made to provide an ergonomic and user friendly user interface in order to allow neuroscientists to use XNBC without any expertise in computer science. The XNBC program is run by launching the xNBC control panel, from which the user sequentially calls several programs using the control panel push-buttons. This control panel displays push-buttons arranged to guide the user to perform neuron and network definition, simulations and then analysis. Figure 1 shows the control panel. Arrows indicate the steps to follow.

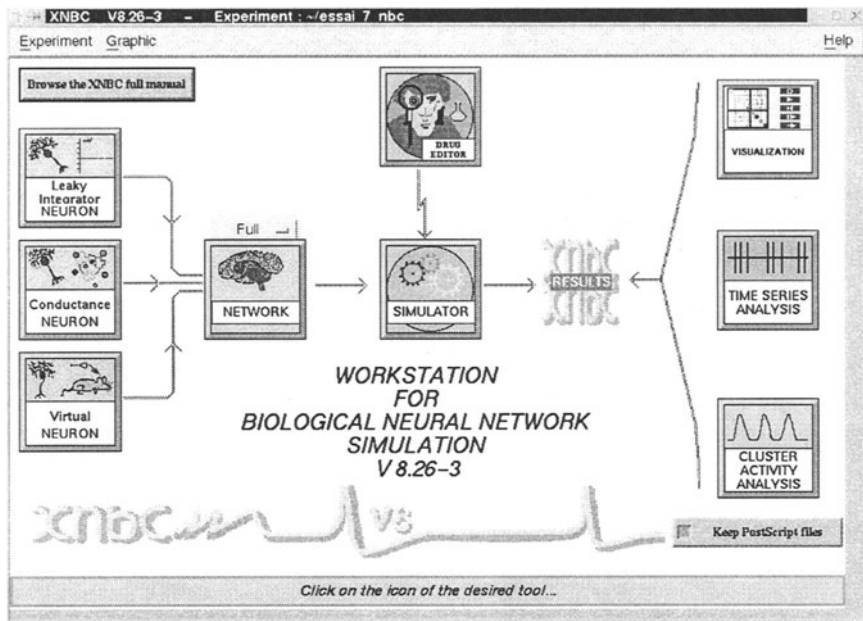


Figure 1. The XNBC V8 control panel

The control panel shows the steps to follow to perform a simulation. From left to right: the two neuron graphic editors (top: phenomenological model editor; bottom: conductance based model editor), then the network editor before the simulator itself. After the simulation, the user can choose to either visualise (right top), perform time series analysis (right middle) or cluster activity analysis (right bottom). A button allows one to launch the drug editor . Each square with an icon is a push-button that launches the corresponding tool. Menus visible at the top are tear-off menus.

The very first thing to do is to choose a name for the simulation. This name is used as a generic base name. Using this name, a directory is created in which are stored all files created during the simulation session. This allows one to isolate and keep together all related files. Then the user must sequentially:

- i) choose the parameters of the neuron models;
- ii) build a neural network;
- iii) define drugs and the currents on which they act (if CBM used);
- iv) run the simulation;
- v) visualise the simulation results;
- vi) analyse the simulated network behaviour.

The modelled neurons, as well as the network, are described using graphic tools, and the simulator itself also allows one, like the control panel, to

control all the simulation process using pull-down menus instead of graphic push-buttons.

These six steps are detailed below. A separate subsection is devoted to each tool.

5.1 Choose the parameters of the neuron models

The user must first choose the modelling level (phenomenological or conductance based) and the type of model to use (PUM, LIM, BUM, CBM) or even virtual for hybrid simulations. Neuron and network parameters can be modified during the simulation, to mimic electrical stimulations and the action of drugs. There exist two graphic editors to adjust the neuron parameters.

5.1.1 The phenomenological model graphic editor

The PUM model is derived from the classical leaky integrator, but implementing more properties such as threshold fatigue and post-spike membrane shunt (for a complete description of this model, see Vibert *et al* 1994a).

Figure 2 shows the user interface of this neuron editor. The user adjusts the parameters by sliding scale cursors on the right part of the screen, while the temporal evolution of the membrane potential changes in real time according to the parameter values on the left part of the screen.

In this model, a pacemaker behaviour can be obtained with a threshold below the resting potential. A small amount of noise can be added to the membrane potential. When the parameters are correctly tuned, the user saves the unit parameters and can print a copy of the graph.

The classical leaky integrator model (LIM) can be defined using this neuron editor, with the possibility of having threshold adaptation. The main difference with PUM resides in the way postsynaptic potentials (PSP) are modelled (see below).

The conditional burster, referred to as a Burster Unit Model (BUM), is also available from the same neuron editor

In PUM and BUM, the postsynaptic potential (PSP) parameters, namely their amplitude, rise and decay time constant, depend on the membrane on which the postsynaptic receptor is, and are consequently a neuron characteristic. The three parameters can be adjusted separately. In LIM, PSPs are modelled as pulses with a decay time constant identical to the membrane time constant.

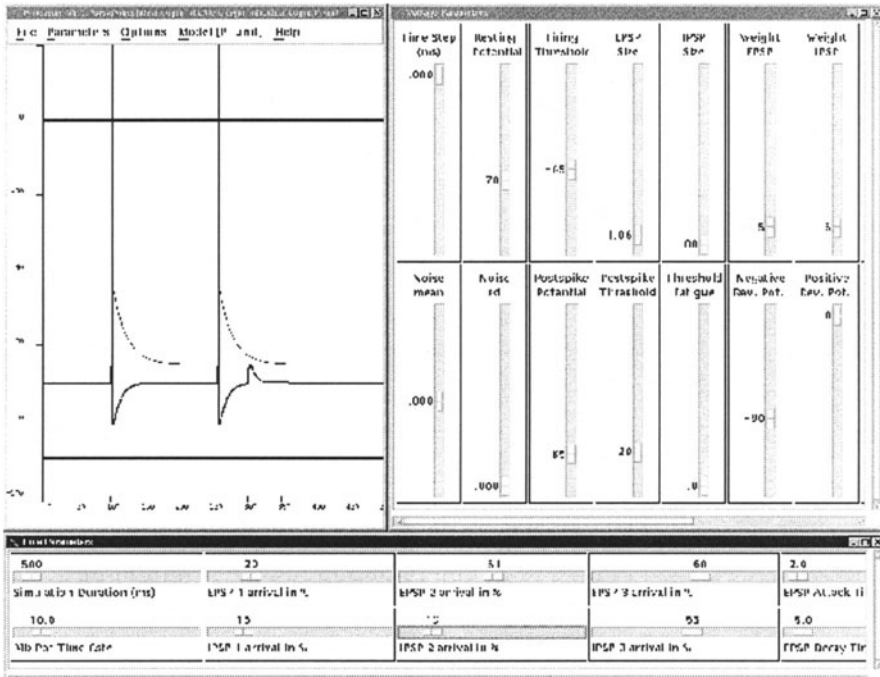


Figure 2. The neuron graphic editors: a) phenomenological model editor (left: the modelled neuron; right and bottom: the scales used to adjust the parameters).

5.1.2 The conductance based model graphic editor

XNBC's conductance model is derived from the Hodgkin-Huxley (HH) model, and incorporates 14 different transmembranar currents. This model explicitly takes into account the Na^+ , K^+ , Ca^{++} , and Mg^{++} ions. The following currents are implemented by XNBC: I_{Na} , $I_{\text{Na+}_\text{persistent}}$, I_{Ca} , I_{t} , I_{K} , I_{M} , I_{A} , I_{AHP} , I_{C} , I_{H} , I_{NMDA} , I_{leak} , $I_{\text{syn_epsp}}$ and $I_{\text{syn_ipsp}}$. Currents are modelled using the dynamics of their activation/inactivation constants. All parameters of the CBM can be individually adjusted. The NMDA receptor for glutamatergic synapses neuromodulation is also implemented in this model (for a complete description of this model, see Vibert et al 1994a). Figure 3 shows the user interface of this neuron editor. The user adjusts the parameters by moving the double dials cursors (or by typing the value), while the temporal evolution of the membrane potential and ionic current (left part of the screen) changes in real time according to the parameter values. Several integration methods can be used. A graphic submenu allows one to plot any parameter versus any other parameter in order to study the dynamics of neuron behaviour. Current and voltage clamp experiments can be simulated in order to adjust the conductance values.

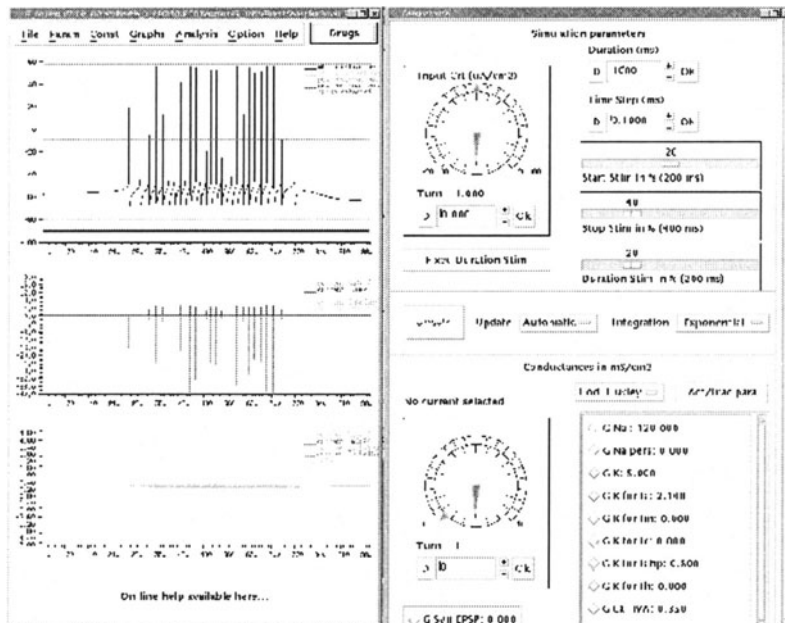


Figure 3. The conductance based model editor (left: the modelled neuron (upper part: potential, two lower parts: currents); right: some of the dials and input windows used to adjust the parameters).

Drugs such as TTX and TEA, or any drug the user wants to define using the drug editor, can be released to compare the neuron behaviour with and without the corresponding blocked channels.

The postsynaptic currents (I_{syn}) are defined by three parameters, namely their amplitude, rise and decay time constant, as the PSP in the phenomenological model.

5.1.3 The virtual model

The neuron model can also be a 'virtual model', that is, an actually recorded neuron, whose spike time arrivals are used as input on the simulated neurons or networks. This allows one to build hybrid networks.

Once the neuron models are defined, neurons can be grouped together into networks.

5.2 Build a neural network

The connection matrix of the modelled network can be described using one of the two graphic network editors. According to the push-button selected

above, the network editor push-button, the simple, or the full featured network editor is launched when the network editor push-button is pressed.

The units of one nucleus can be connected with the units of any other nucleus (including itself) through either all excitatory, all inhibitory or both excitatory and inhibitory synapses.

According to the type of network you want to build, the simple network editor can be enough, or the full featured network editor can be necessary. Figure 4 schematises networks and the best editor choice.

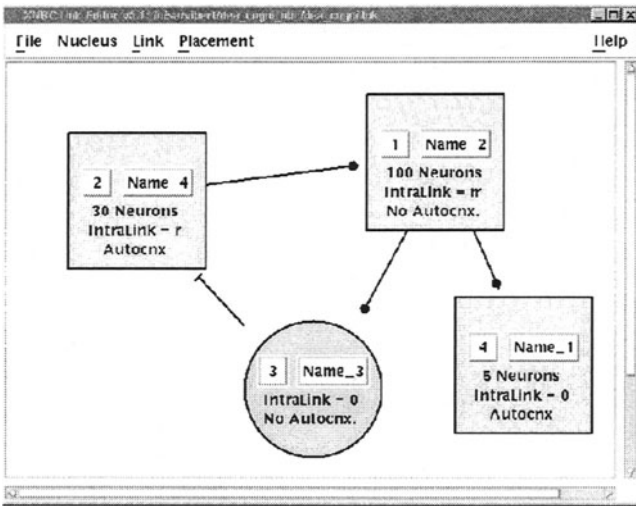


Figure 4. The simple network editor (squares represent nuclei and the disk a single neuron. Lines with an open circle = inhibitory connection; lines with a closed circle = excitatory connection. B: Intra nucleus connection editor: the user interface of one of the parts of the network editor, where all neurons and individual connections are represented, and can be modified.

5.2.1 The simple network editor

This allows one to build nuclei containing only one cluster each. It is made for rapidly building large networks made of one or several nuclei that can be mixed together (PUM, LIM, BUM, CBM or Virtual). It nevertheless allows one to finely edit the connection matrix, but not to see individually the neurons inside the clusters, nor to position the neurons geographically, nor to work in Horsley-Clarke coordinates. It does not allow one to describe the NMDA connections. Despite these limitation it is much easier to use than the fully-featured editor, and is generally chosen by users. Figure 4 displays its user interface with a network built within.

5.2.2 The fully-featured network editor

This allows one to build networks made of nuclei containing several clusters each. It is intended for building small (or large if time is not a problem...) networks made of several clusters that can be mixed together (PUM, LIM, BUM, CBM or Virtual). It allows one to finely edit the connection matrix and to choose the connection density around a given neuron. This rather sophisticated tool allows one to see individually the neurons and their connections inside the nuclei, and to work in Horsley-Clarke coordinates. Connections with glutamate release acting on NMDA receptors can also be specified. Figure 5 displays its user interface with a network built within.

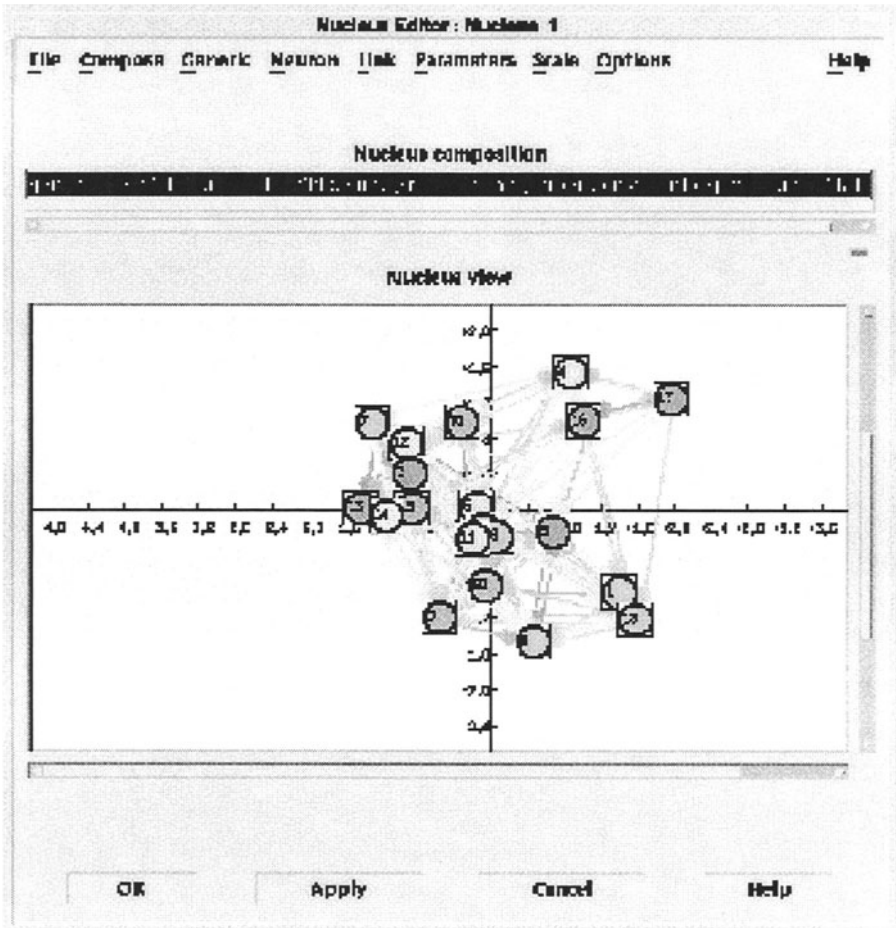


Figure 5. The fully-featured network editor (this panel is the nucleus editor, part of the full featured network editor where neurons are visible in their position in the 3D space. The colour is proportional to the synaptic weight. Neurons are here sparsely connected).

Once the network is built, the simulation can be run.

5.2.3 Run the simulation

After the iteration step and simulation duration are set, the simulation can be run. The integration method can be chosen between the exponential algorithm described in MacGregor (1987), the 4th order Runge-Kutta or the Von Euler method. During the simulation process, network behaviour can be observed on a graphic display representation if necessary. At any time, the simulation can be momentarily stopped in order to modify the external input to one or several nuclei, to provide stimulation, drug, or change any parameter. It is also possible to modify some anatomical characteristics, such as connections between two nuclei, mimicking a lesion, or the membrane properties, mimicking pharmacological effects of drugs. The simulation can be stopped at any time, and the network state kept for later simulation. Figure 6 shows the simulator during a simulation.

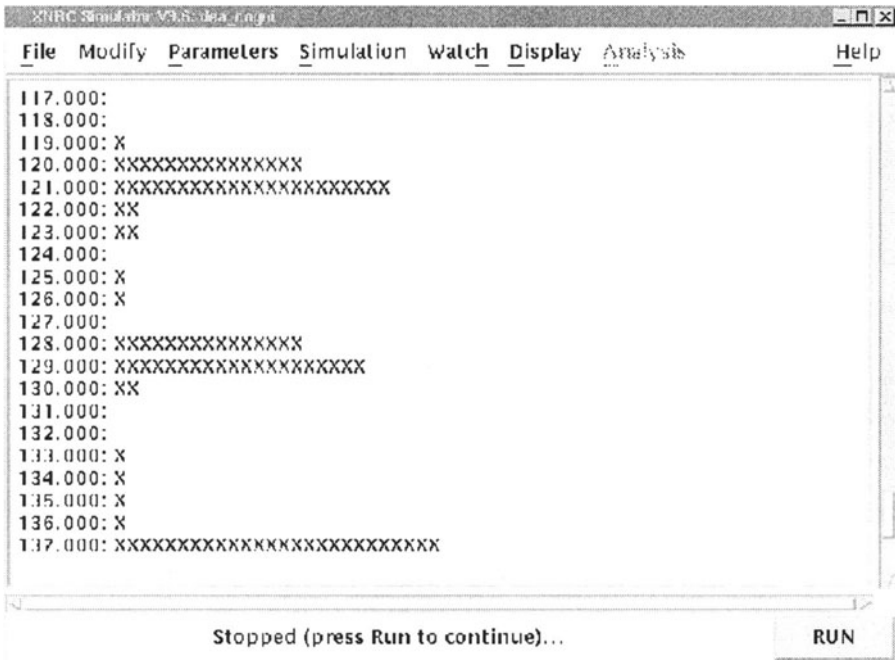


Figure 6. The simulation program during a simulation (Network activity is represented in arbitrary units by crosses. Time runs from top to bottom. This representation allows one to follow dynamically the network activity during the simulation.)

After the simulation, the visualisation tool can display the behaviour of the modelled network with respect to time, using several representations with adjustable speed, colour and time scale. Data have also to be analysed.

5.3 Visualise the simulation results

The visualisation tool is similar to a video tape player allowing one to run, stop, go to a given iteration, or change the display speed. It allows one to display:

- the joint colour coded temporal evolution of the membrane potential of simulated units (and of units from a virtual cluster);
- spikes travelling along connections between neurons. This representation is convenient for studying the synchronisation of spike trains;
- intracellular recordings of selected units;
- the global activity of the network.

Figure 7 displays examples of screens obtained with this tool.

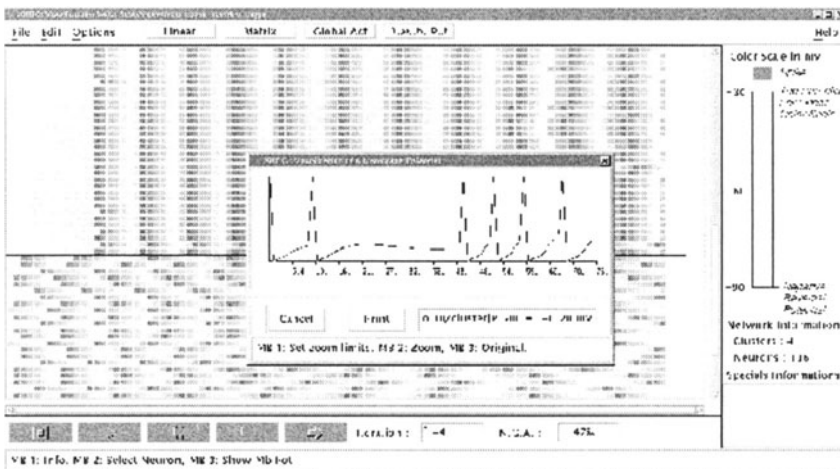


Figure 7. The visualisation tool. (Visualisation of membrane potential temporal evolution; Inset: Intra cellular recording from a selected unit. Membrane potential is colour coded. Red bars represent spikes.)

5.4 Analyse the simulated network behaviour

Two analysis tools are available.

- One is for time series analysis, and is mainly devoted to analysing the individual unit discharges (point process analysis). It offers 31 different analyses grouped into 8 submenus, such as instantaneous rate, interspike intervals, pre and post-stimulus histograms, Poincaré maps of intervals, auto and cross correlograms, etc.

- The other analysis tool is devoted to cluster activities and allows 8 different analyses such as FFT, Poincaré maps, amplitude correlograms, etc.

Their user interface is similar. Those of the time series analysis tool is shown in Figure 7.

6. HISTORY AND IMPLEMENTATION

The original name of XNBC was `Neuro_bio_clusters` (NBC), developed in 1988 on a MicroVax II, under Ultrix V2. Versions 2 and 3 of NBC implemented a PUM and few analysis tools. Subsequently, beginning with version 4, NBC evolved toward a tool devoted to the general simulation of neural networks. In version 5 the CBM was introduced, and version 6 saw the arrival of the notion of virtual cluster, of a graphic network editor and of a visualisation tool (Vibert et al 1994b). NBC versions 4 to 6 were menu-driven, with only some parts using an Xwindow interface. Version 7 was the first version where the control menu was replaced by an Xwindow interface, and was consequently renamed XNBC. With version 7 began also the possibility to mix within a single simulation the PUM and the CBM (Vibert et al 1994b). For XNBC V8, presented here, the interface was completely redesigned, including a second network editor, a new version of the ion-conductance based model editor, an Xwindow version of the simulator and the apparition of the concept of nucleus, allowing us to take into account the anatomy of the modelled network. XNBC development is still in progress through close collaboration with neuroscientists.

XNBC is written in portable ANSI C, and was compiled on Ultrix, Digital Unix, IBM AIX, SUN Solaris, HP Ux, Linux and DEC VMS and OpenVMS. XNBC runs on Xwindow workstations and needs the Motif library. When possible, the GNU C compiler (gcc) is preferred. XNBC produces generally simple ASCII data files that can be easily converted to any format required by common graphic programs or spreadsheets. It produces native colour PostScript files (that can be directly used to prepare figures). XNBC is a public domain software package available freely as open source under the GPL licence on Internet (<ftp://ftp.b3e.jussieu.fr/pub/XNBC>). Information about new versions and the full manual is available at URL <http://www.b3e.jussieu.fr/xnbc>.

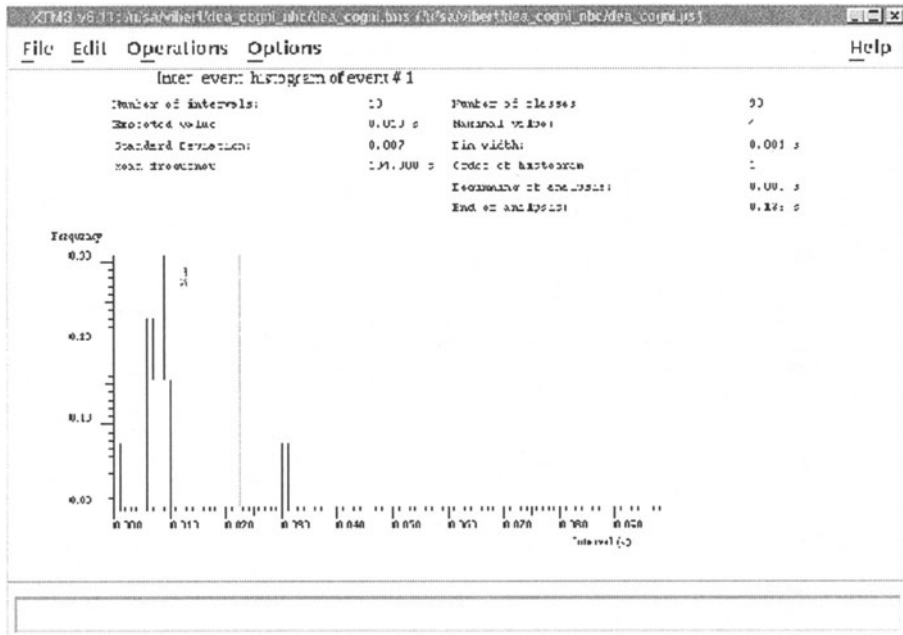


Figure 8. Analysis tools (Time series analysis tool window (interspike interval histogram))

7. DISCUSSION

XNBC was developed by an experimenter, to answer a specific question, and then enhanced to be used by computer-naive experimenters. Few simulation tools exist for neurobiologists. These are simulation tools devoted to multi-compartmental modelling of one or a few neurons possibly grouped into small to medium-sized networks. They are used to study biophysical models and interactions between parts of units, at the level of dendritic fields and when connected into small networks. A comparative study between several such simulation tools was made by De Shutter (1992) and showed that few were able to simulate thousands of networks. This is confirmed by analysing the simulators presented in the book of Skrzypek (1994) devoted to neural network simulation tools. Genesis (Bower and Beeman 1995) allows one to simulate multi-compartmental neurons and, when the number of compartments is not too large, to simulate networks of several hundreds of neurons. Nevertheless, a neurobiologist has to learn how to use it, since specific scripts must be written to build the neurons and the network. XNBC provides the neurobiologist with a turn-key tool to simulate simple-to-complex living neural networks, using graphic tools and real time visual control to adjust the numerous parameters of the different models (phenomenological or CBM). Graphic tools provide a natural way to build the network and to connect nuclei or neurons together, or to visualise the

simulation results. No programming language has to be learned to build the neuron or the network. Nevertheless, for compartmental modelling, Genesis or Neuron (Hines 1994) are more adequate than XNBC.

The possibility of building hybrid networks is not found in other general simulators, and provides a powerful way to study neuronal coding, since experimental data can feed simulated networks whose parameters can be modified at will to understand the way the message is coded as exemplified in Vibert et al (1997).

XNBC produces simple ASCII data files that can be easily converted into any format required by any common graph or number cruncher program. It produces native colour PostScript files (that can be directly used to prepare figures).

The analysis tools incorporated with the simulator represent a specific feature of XNBC. These tools are the tools daily used by an experimenter to analyse the experimental records. Moreover, these tools can also be used separately, independently from XNBC to analyse experimental data, giving the experimenter the interesting possibility to analyse in exactly the same way both experimental and simulated data. Several papers were published with data analysis performed with the tools included in xtms (in fact, these tools were primarily developed to analyse actual experimental data (Segundo et al 1995a, 1995b, 1998).

Simulation speed is an important factor, and allows one to increase the number of simulations, and consequently the number of parameter tests. Nevertheless, one must keep in mind that when setting a simulation, the time spent to adjust the neuron parameters and to connect the neurons, rather than just the time to run the simulation (De Schutter 1992), and that a good interface is almost as important to speed up the simulation process as the computer speed itself. Moreover, an ergonomic user interface relieves the experimenter from basic problems of computer language, and allows him/her to concentrate on the simulation process itself, not on the simulation tool. We think that XNBC can efficiently help the neuroscientist because of the carefully designed ergonomomy of the user interface.

NOTES

- 1 The XNBC project is supported by the DRET (Contract 91/1246A and 94/2526A).

REFERENCES

- Bower, J, Beeman D. (1995) *The Book of Genesis: exploring realistic neural models with the GEneral NEural Simulation System*. Springer-Verlag, New-York.
- De Schutter E. (1992) A consumer guide to neuronal modeling software. *TINS* **15** 462-464.

Geoffrois E, Edeline J-M., Vibert J-F. (1994) Learning by delay modifications. In F. H. Eeckman (ed.) *Computation in neurons and neural systems*. Kluwer Academic Publishers, Boston (USA), pp.133-138.

Hines M. (1994) The Neuron simulation program. In J. Skrzypek (ed.) *Neural Network Simulation Environments*. Kluwer Academic Publishers, Boston (USA), pp. 148-163.

Hodgkin, A.L., Huxley, A.F., (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. Lond.* 117, 500--544.

MacGregor, R.J., 1987, *Neural and Brain Modeling*. Academic Press, San Diego.

Segundo J. P., Vibert J-F, Stiber M., Hanneton S. (1995a) Periodically modulated inhibition and its post-synaptic consequences. I. General features. Influence of modulation frequency. *Neuroscience* 68 657-692.

Segundo J. P., Stiber M., Vibert J-F, Hanneton S. (1995b) Periodically modulated inhibition and its post-synaptic consequences. II. Influence of pre-synaptic slope, depth, range, noise and of post-synaptic natural discharges. *Neuroscience* 68 693-719.

Segundo J. P., Vibert J-F, Stiber M.. (1998) Periodically modulated inhibition and its post-synaptic consequences. III. The heterogeneity of the post synaptic spike train, and how control parameters affect it. *Neuroscience* 71.

Skrzypek J. (1994) *Neural network simulation environments*. Kluwer Academic Publishers, Boston, USA.

Vibert J-F, Pakdaman K., Azmy N. (1994a) Inter-neural delay modification synchronizes biologically plausible neural networks. *Neural Networks*, 7: 589-607.

Vibert J-F, Pakdaman K, Cloppet F, Azmy N. (1994b) NBC: a workstation for biological neural network simulation. In Skrzypek J (ed.) *Neural network simulation environments*. Kluwer Academic Publishers, Boston (USA), pp. 113-133.

Vibert J-F, Pakdaman K., Boussard E., Av-Ron E. (1996) Computational neuroscience and neurology. *Nature-Medecine* 1 1247-1248.

Vibert J-F., Pakdaman K., Boussard E., Av-Ron E. (1997) XNBC: a simulation tool. Application to the study of neural coding using hybrid networks. *BioSystems* 40 211-218.

Vibert J-F., Pham J., Pakdaman K, Azmy N. (1995) XNBC: A simulation tool for neurobiologists. In J. Bower (ed.) *The neurobiology of Computation*. Kluwer Academic Pub., Boston, USA, pp. 346-352.