Simulation of neural networks: a preliminary report on two computational strategies.

Daiana Simone and Alfredo Colosimo
Dept. of Anatomy, Histology, Forensic Medicine and Orthopaedics
Sapienza, University of Rome

Keywords: Computer-aided Models, Neural Networks, Simulation Software,

Abstract: The basic assumptions of the present contribution are the following: i) a satisfactory mechanistic knowledge of the higher human cognitive abilities will be reached after understanding the concerted and cooperative behavior of brain regions usually identified by peculiar morphofunctional features; ii) it is really hard to overestimate the importance of computer-aided modeling and simulation in the study of the global and local network(s) connecting the above regions; iii) the related computational problems are better faced by exploiting different software tools, each endowed with excellent performance in specific problems, than focussing on a single programming environment. In this frame, we tested the steepness of the learning curve of two popular neural simulation environments, namely NEURON and BRIAN, in modeling the following cases: a) a self-sustained, reciprocal activation of a small number of ring-chained neurons, and b) the spiking activity of a small network possibly endowed with a random set of connections. We confirm that the NEURON and the BRIAN environments appear well suited in dealing with case a) and b), respectively, and, in particular, in studying the biophysical / morphological features of single neural cells or in exploring the functional features even of complicated or large-size network topologies.

1 INTRODUCTION

Neuronal (mis)functions are closely related to cell morphology changes, particularly in the case of functional degradation, as demonstrated by alterations occurring in severe neurodegenerative diseases such as Alzheimer's disease, amyotrophic lateral sclerosis and multiple sclerosis [1, 2]. At the subcellular level, even the finest features are based on dynamical changes in the membrane potential difference linked to the opening and closing of specific ion channels[1]. However, if in relatively simple functions such as the control of rhythmic, automatic movements and generation of circadian rhythm, individual neurons may continue to perform their functions after being separated from the corresponding networks, in complex cognitive functions, such as declarative memory, language processing, and face recognition, neurons architectures are likely to be driven by the properties of groups of highly specialized neurons organized in networks [3]. As a matter of fact, current understanding of the highest brain functions suggests that the role of individual neurons is limited to generating electrical signals and releasing neurotransmitters, while neural networks play a dominant role due to the complex interactions emerging within and between functionally specialized areas.

It is worth stressing that recent studies showed how the functional and anatomical features of these networks reflect a so called *small-world* (SM) architecture, characterized by high and low values, respectively, of topological descriptors like *clustering coefficient* (CL) and *characteristic path lengths* (CPL) [4]. A high CL value is indicating a high level of local neighborhood clustering, responsible for an efficient local information processing. A low CPL value, on the other hand, characterizes a high level of global communication efficiency. SM neural models have been used to study Short Term Memory (STM) as well as Working Memory (WM) performances, which demonstrate crucial for daily life, such as reading comprehension, planning, problem-solving and learning of new skills [5, 6]. An alternative STM mechanism is the storage of information in the form of ongoing neural activity maintained by excitatory feed back connections between neurons. Such an activity could reverberate within the closed loop of neurons and be sustained for a period of time. The model of reverberating circuits is interesting because it does not involve any enduring change in nerve cells: the short term memory for the event is maintained simply by ongoing neuronal activity.

In the first part of this contribution by means of NEURON [7] , one of the best known and most used neural simulation environment, we worked out relatively simple models inspired by hippocampal pyramidal neurons and analyzed their action potential trains as a function of the number of stimuli and synapses [8]. In the same software environment we also realized a model of a reverberating circuit and checked different topologies of SM networks to investigate whether individual differences in WM abilities could be related to a specific small-world network organization.

A major virtue of neural computational modeling is the high realism of the simulations they provide, even more so thanks to the continuously increasing power and sophistication of the software tools. This is well exemplified by BRIAN, a neural simulation tool designed as a Python plug-in and able to fully exploit the advantages of such a powerful and flexible programming environment. In the final part of our paper we used the BRIAN tool to study the influence of random and ordered connections in recording the spiking activity within a small neural architecture.

.

2 Methods.

2.1 the *NEURON* programming environment.

NEURON [9], is a quite powerful software for modelling neurons, both as individual cells or within structural and/or functional networks, allowing to represent excitable cells features like branching morphology, type and

[1] The ensuing electrical signals are of two types: 1) Graduate Potentials (GP), namely modifications of the resting potential determined by the opening/closing of ion channels, which can depolarize (decrease intracellular negative potential), hyperpolarize (increase Intracellular negative potential) or ripolarize (recover the resting potential values) neurons. Such modifications can be modulated in amplitude; however, they rapidly decrease as they move away from the generation point. 2) Action Potentials (AP), that is rapid modifications of the resting potential generated in response to a depolarizing stimulus above a threshold. AP propagate without loss in amplitude and are characterized by the inversion of the membrane polarity and modulated only in frequency.

distribution of channels, active transport, and synaptic plasticity.

The essential morphofunctional features of complex neural cells are defined by a friendly graphical user interface: as an example, Figure 1 shows how pyramidal neurons belonging to the CA1 region of the hippocampus, as described in [10], can be reduced to a basic and functional scheme where only cell body, axon, dendrites and synaptic spines are visible.

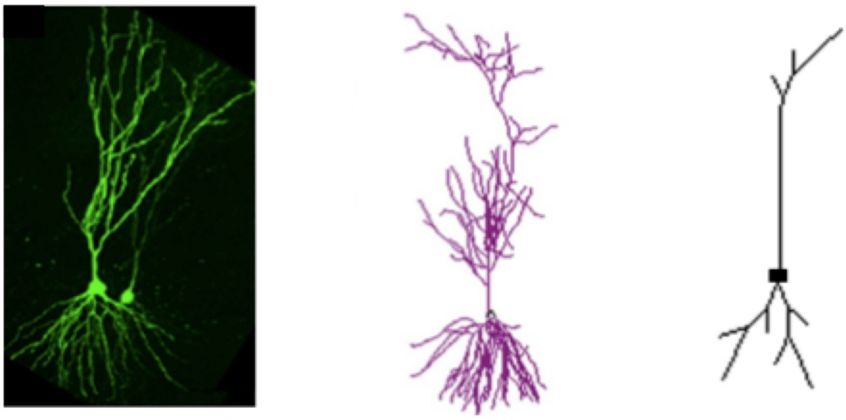

Figure 1: *Morphofunctional scheme of a pyramidal neuron.* The morphology of the hippocampal neuron on the left panel (London and Segev, http;//senselab.mad.yale.edu/ModelDB/) is reproduced with reasonable realism in the middle panel. The right panel shows the essential morphofunctional features of the cell: the axon and the dendritic arborization develop from the bottom and the top side, respectively, of the small black rectangle representing the cell body.

2.1.1 Modular programming in NEURON.

A most appreciated NEURON's feature is the modular programming style, which allows to separately deal with the morphological, biophysical, and dynamical aspects of the simulation. Very simple macroinstructions of the type shown in CodeSegment 1, after the activation of the graphical user interface (line 1), invoke and execute specialized modules each responsible for one of the three above mentioned aspects (lines 2 - 4).

CodeSegment 1: a typical macro in NEURON.

```
1)   load_file ("nrngui.hoc")
2)   load_file ("morpho.hoc")
3)   load_file ("linear_stim.hoc")
4)   load_file ("graphic.hoc")
```

Each line in the macro calls for a specialized module in charge of setting up, in the order: the Graphical User Interface (GUI), the morphological details of neurons and of their synaptic connections, the analytical model of their functional features and, finally, the graphical output of their dynamical response to stimulation.

The biophysical properties essential for a reliable multi-compartmental description of neurons and of their electrical activity are elucidated in a number of studies [3, 10]. The algorithms used by NEURON to account for the generated electrical signals are based upon a cable theory which depicts cell bodies, dendrites and axons as small cylinders connected to each other with capacitance and resistance combined in parallel. NEURON calculates the time and space dependent continuous variables representing action potentials or ionic currents as a finite and predefined number of discrete points in space at given time instants.

In the NEURON environment we analyzed first the very simple case shown in Figure 3 (left panel), where electrical responses in the form of action potentials are generated in the postsynaptic neuron through a single synapse by a presynaptic neuron receiving, in turn, electrical pulses of current. Thus, Figure 4 shows the action potentials generated by electrical pulses of different number and intensity. As expected, below an arbitrary

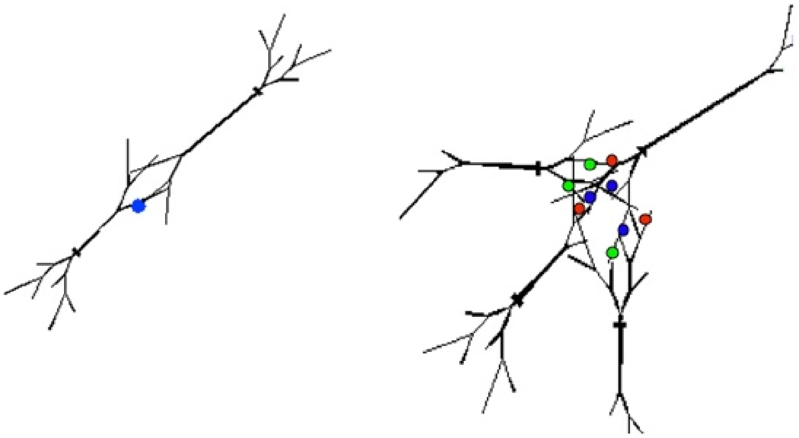

Figure 2: *Modeling functional connections.* In both panels the neuron in the upper right corner can be stimulated under various conditions by one (left panel) or three (right panel) neurons on the bottom right corner. The stimulating neuron(s) act(s) through a single synapse (left panel) or three synapses (right panel) each, indicated by the colored circles. The response (action potential) of the stimulated neuron are reported in Figure 3 and Table 1. The meaning of the morphological details is the same as in Figure 1, right panel.

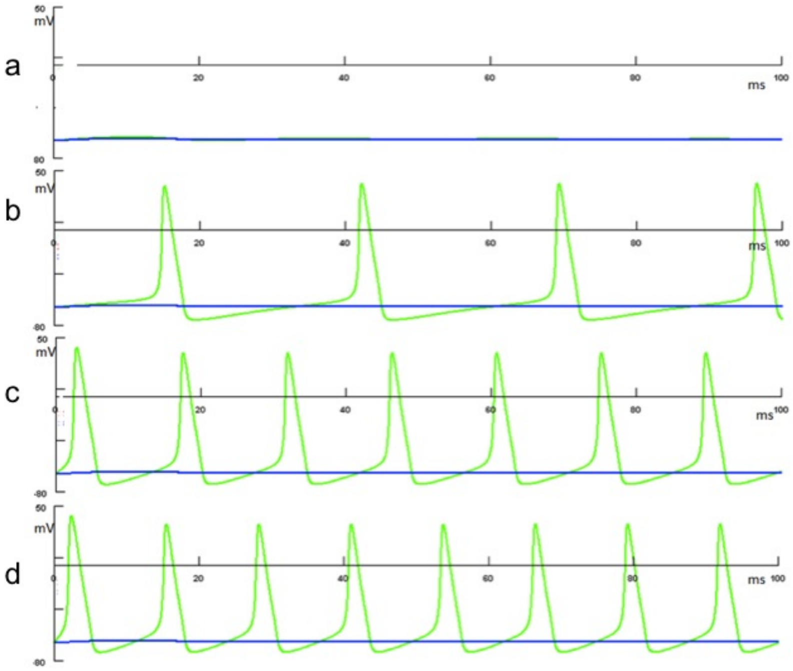

Figure 3: *Modeling action potentials.* Simulations were obtained by the NEURON software and refer to the monosynaptic connection shown in Figure 2 (left panel). The current stimulation on the presynaptic neuron in panels a,b,c,d is 0.010, 0.014, 0.20 and 0.30 nA, respectively. The threshold value for the onset of action potentials is 0.014 nA. Below the threshold value (panel a) the action potential is not produced. If the stimulus amplitude exceeds the threshold value (panel b, c, d) the action potentials appear and their frequency scales with the stimulus intensity. The blue track indicates the absence of activity in a reference neuron not subject to stimulation.

threshold value of current (0.014 nA) the action potential does not appear, while above the threshold the train of action potentials is modulated in frequency (panels c,d).

Then, we generalized the simulation set-up including more than one synapse and more than one presynaptic neuron (Figure 3, right panel), and summarized in Table 1 the observed activation in terms of firing frequency of the postsynaptic neuron. The same firing frequency was generated, irrespective of the synaptic weight, either by a triplet of active synapses associated to a single active presynaptic neuron (Table 1, row 1), or by three active synapses each associated to a single active presynaptic neuron (Table 1, row 2). Notice that in the any case each of the three potentially active synapses on each presynaptic neuron receive one third of the stimulating current. Thus, a clear scaling between firing frequency and synaptic weighting only appears if the three synapses associated to each presynaptic neuron ar all fully active(Table 1, row 3), although the nonlinear trend of such scaling is not easily predictable [7].

# Active Presynaptic Neuron(s)	**# Active Synapse(s) / Presynaptic Neuron**	**Synaptic Weight**	**Firing frequency** (# Spikes / 200 msec)
1	**3**	**0.1**	17
”	”	**0.5**	17
”	”	**1**	17
3	**1**	**0.1**	17
”	”	**0.5**	17
”	”	**1**	17
3	**3**	**0.1**	17
”	”	**0.5**	18
”	”	**1**	20

Table 1: *Firing frequencies generated using the connection scheme of Figure 3, right panel.* The values in the three rows refer to different (active synapses) / (active presynaptic neurons) ratios. Rows 1 and 2 refer, respectively, to three active synapses over a single active presynaptic neuron, and to one active synapse in each of three presynaptic neurons. In all cases the presynaptic neuron(s) was (were) activated by 0.1 nA current (threshold = 0.014 nA) and the AP spikes recorded in a 200 msec time window. The synaptic weight can be interpreted as the fraction of the maximal energy transmissible through the synaptic connection.

2.2 The *BRIAN* programming environment.

As compared to NEURON, BRIAN is a more recent and more specialized software primarily addressing single-compartment neural models. It was introduced in 2008 as a simulator for spiking neural networks [11], and is entirely written in Python using the NumPy and SciPy numerical and scientific computing packages, so that the models at hand can be directly represented through differential equations in standard mathematical form. As an example, CodeSegment 2 contains the instructions for drawing the raster plots in Figure 4.

CodeSegment 2: generating random spikes in BRIAN.

```
12)   tau = 10 * second            # membrane time constant
13)   Vt = -50 * mvolt             # spike threshold
14)   Vr = -60 * mvolt             # reset value
15)   El = -49 * mvolt             # resting potential (same as the rest)
16)   psp = 0.5 * mvolt            # postsynaptic potential sizes
17)   G = NeuronGroup(N=12, model=’dV/dt = -(V-El)/tau : volt’, threshold=Vt, reset = Vr)
18)   C = Connection(G, G)
19)   C.connect_random(sparseness=0.01, weight=psp)
```

Notice in line 17 the definition of a NeuronGroup G including 12 neurons with given threshold and reset values . Line 18 defines a fully-connected framework, within which the probability of finding an active connection in a random location is defined through the ”sparseness” parameter (line 19).

2.2.1 Dealing with random events in BRIAN.

The direct access to the bulk of powerful numerical and graphical macros available in the NumPy and SciPy packages is an important feature of the BRIAN environment and greatly simplifies the management of random events. An example is provided by Figure 4 showing, in the form of *raster plot* (namely, a graphic representation of events in a given temporal relation), the effect of a random, drastic reduction of the fraction of internal links on the spiking activity of an initially fully connected 12-neurons net. Both panels in Figure 4 are generated by the transparent, english-like code in CodeSegment 2: switching from the synchronous (left panel) to the randomic (right panel) spiking regime is obtained by just lowering from 1.0 to 0.01 the *sparsness* parameter in the *random* function (code line 19). It is difficult to overestimate the utility of a straightforward management of random events, in getting the null reference needed to assess the statistical significance of measurements in complex topological networks [12].

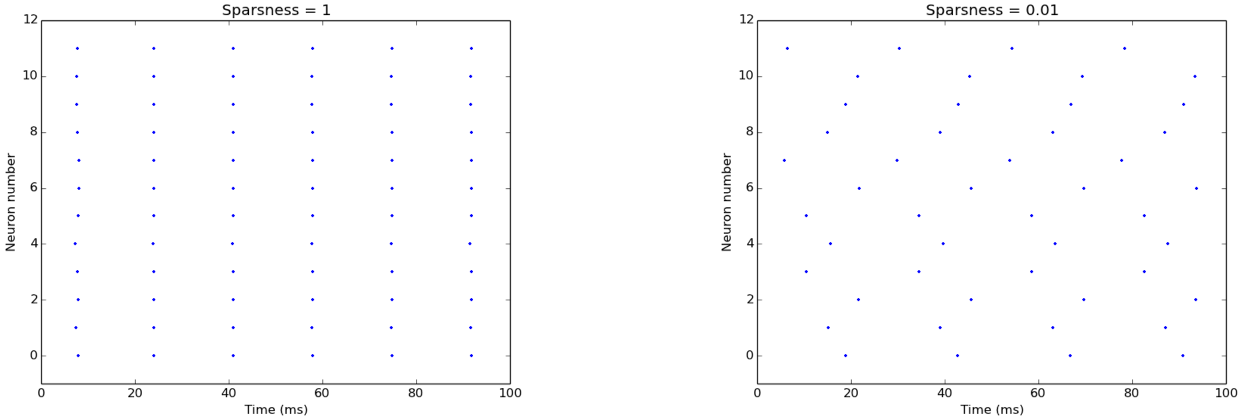

Figure 4: *Spiking activity in a 12-nodes neural net with random links.* The raster plots are generated by the code reported in CodeSegment 2. Left panel: synchronous activity of nodes connected by 66 identical binary links in a fully connected 12-node net. The timing of the activity bursts is defined by the average membrane relaxation time-constant which is identical for all nodes (see Figure 5 and [11] for details). Right panel: lowering the number of binary links to 1/100 of the full connectivity value destroys the synchronous spiking and reduces the global activity of about 33%. Notice, however, that the cycling activation of the asynchronous pattern is preserved.

3 RESULTS

3.1 Modeling memory mechanisms.

Reverberating circuits were proposed long time ago by Lorente de No and Hebb [13, 14] and proposed again more recently [15] as plausible mechanistic models of the *working* memory, since they allow the storage of information with no structural modifications of the type supposedly occurring in the *long term* memory. The idea can be tested by checking the spiking activity within a neural net where the arrangement of nodes and of their connections reproduces a closed loop. Setting up the toy net shown in Figure 5, left panel, is relatively simple by means of the NEURON programming environment, as well as producing the associated spiking activity, also shown in Figure 5 (right panel), by means of the code listed in Appendix A.

It should be noticed that the essential feature of the reverberating circuit hypothesis, namely the emergence of a self-sustaining, stable and regular pattern of activity involving the whole network as a consequence of the initial stimulation of a node, is independent of the number of nodes, and preserved even in nets of minimal size. As a matter of fact, the single frequency signal in figure 5 can be easily modulated by additional frequency components if the underlying net is appropriately worked out, as shown in Figure 6. It is worth noticing that the net sketched in Figure 6 (left panel) actually includes three nodes which are topologically aligned, the central node (A) being linked to both node (B) and (C).

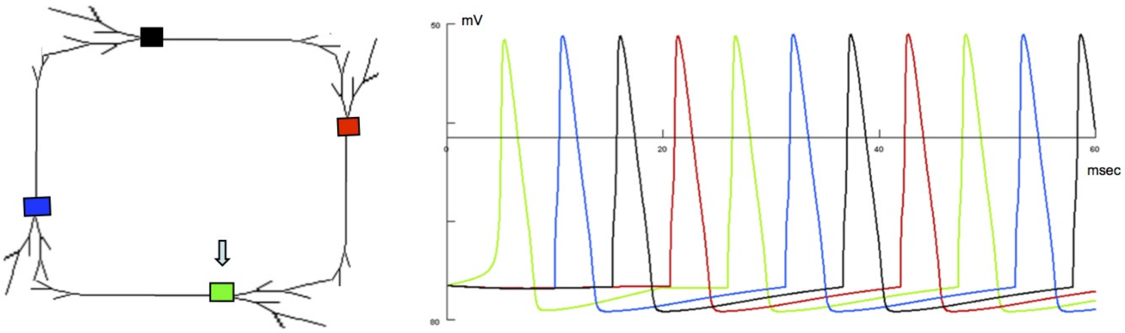

Figure 5: *Computational modeling of a simple reverberating circuit.* The left and right panels contain, respectively, a neural net whose nodes are arranged in a closed loop, and the corresponding train of Action Potentials (APs) generated by the code listed in Appendix A. APs of different color are associated to nodes of corresponding color in the net. The arrow in the left panel indicates the node which is stimulated first. Notice the theoretically endless rhythmical activation of the net's elements stemming from their mutual stimulation.

Thus, the spiking activities associated to (B) and (C), namely red and green APs (Figure 6, right panel), have similar frequencies, while the frequency of the blue APs, associated to node (A), is about double.

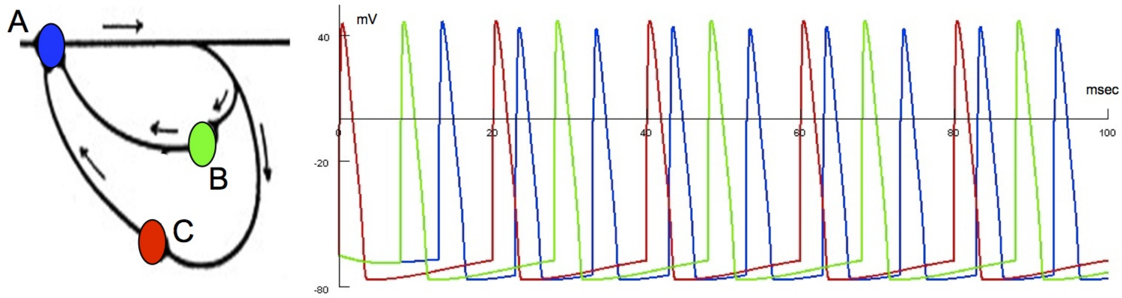

Figure 6: *A minimal net generating multiple frequencies.* Neurons A, B, C in the left panel produce specific APs in the right panels, identifiable on the basis of their color. Notice that the frequency of the blue signals is about twice than that of red and green signals, due to the central location of the (A) node in the linear arrangement: $(B) \iff (A) \iff (C)$.

3.2 Including random events in the net design.

Functional flexibility, and hence prompt responses to unexpected stimulation patterns, can be enormously enhanced by allowing the presence of random events in the underlying mechanisms. We took advantage of the facilitated dealing with randomness typical of the BRIAN environment in order to simulate the patterns of spiking activities arising from a three-neuron loop (see Figure 7) isomorphous to a fully connected network where links can be made active by a bimodal on/off probabilistic switch. In our simulations the code listed in Appendix B was used to explore a number of different combinations of delay times and strength of excitatory/inhibitory synapses, and the most interesting ones are summarized in Table 2.

In this preliminary report the results concern:

- an oversimplified condition in which, in the absence of delay times and of differences in the synaptic strength, only the inhibitory synapses are left active (Fig. 8, top left);
- the same condition as before, except for the activation of the excitatory synapses (Fig. 8, top right);
- the modulation of the time delay of subsequent excitatory (*del_e*) and inhibitory (*del_i*) signals (Fig. 8, bottom left) even in each single loop (Fig. 8, bottom right);

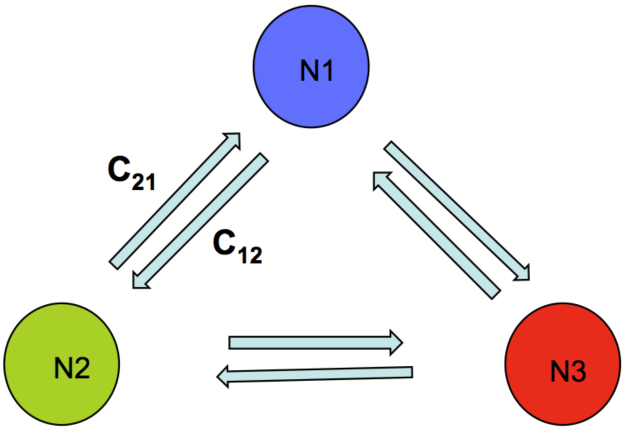

Figure 7: *A three-neuron loop endowed with bidirectional functional connections.*
The activating and inhibitory synapses are localized at the end of, respectively, anticlockwise (C_{12}, C_{23}, C_{31}) and clockwise (C_{21}, C_{32}, C_{13}) cycling connections (see also Table 2).

Connection		**Fig. 8** **top left** **ge ; gi**	**Fig. 8** **top right** **ge ; gi**	**Fig. 8** **bottom right** **ge ; gi**	**Fig. 8** **bottom leftl** **ge; gi**	**Appendix C** **ge ; gi**
C_{12} ; C_{21}						
	weight (we ; wi)	0 ; 3	3 ; 3	3 ; 5	3 ; 5	3 ; 3
	sparsness	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	0.5; 1.0
	delay (del_e ; del_i)	0 ; 0	0 ; 0	30 ; 0	30 ; 0	0 ; 0
C_{23} ; C_{32}						
	weight (we ; wi)	0 ; 3	3 ; 3	3 ; 5	3 ; 5	3 ; 3
	sparsness	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	0.5; 1.0
	delay (del_e ; del_i)	0 ; 0	0 ; 0	60 ; 0	50 ; 0	0 ; 0
C_{31} ; C_{13}						
	weight (we ; wi)	0 ; 3	3 ; 3	3 ; 5	3 ; 5	3 ; 3
	sparsness	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	1.0 ; 1.0	0.5 ; 1.0
	delay (del_e ; del_i)	0 ; 0	0 ; 0	90 ; 0	85 ; 0	0 ; 0

Table 2: *Connection parameters related to the three-neuron loop of Figure 7.*
The parameters are used in the code of Appendix B to produce the signals in Figure 8 and Appendix C. ’ge’ and ’gi’ indicate excitatory and inhibitory connection, respectively.

- the fraction of active links (through the *sparsness parameter* on the background of a fully connected adiacency matrix (Appendix C).

In this context, it is worth mentioning that both topological and dynamical parameters have been used to simulate even cognitive performances by means of neural nets of realistic size, designed according to the so called ”small world” architectures [4, 16–18]. In our case, we would like to stress that in spite of its minimal size, the three-neuron loop in Figure 7 includes, as a significant similarity to physiological counterparts, the presence on each node of both positive and negative inputs, potentially different both in weight and delay time (see Appendix B). Such differences within an essentially symmetric topological structure are responsible for the variety of time-dependent spiking patterns shown in Figure 8 and Appendix C (Figure 10). In the last column of Table 2 a sparsness value = 0.5 is associated to each excitatory link, which constrains the activation to a random event at 50% probability. Under these conditions, the number of active links may assume any integer value from 0 to 3, and the 0 and 3 active links conditions will be identical to those corresponding to the first two columns in Table 2. As expected, Figure 8 top left and Figure 8 top right will be the same as Figure 10 top left and Figure 10 bottom right, respectively.

In the Discussion section a further example of the huge power of the BRIAN environment in dealing with network of realistic size and topology is provided.

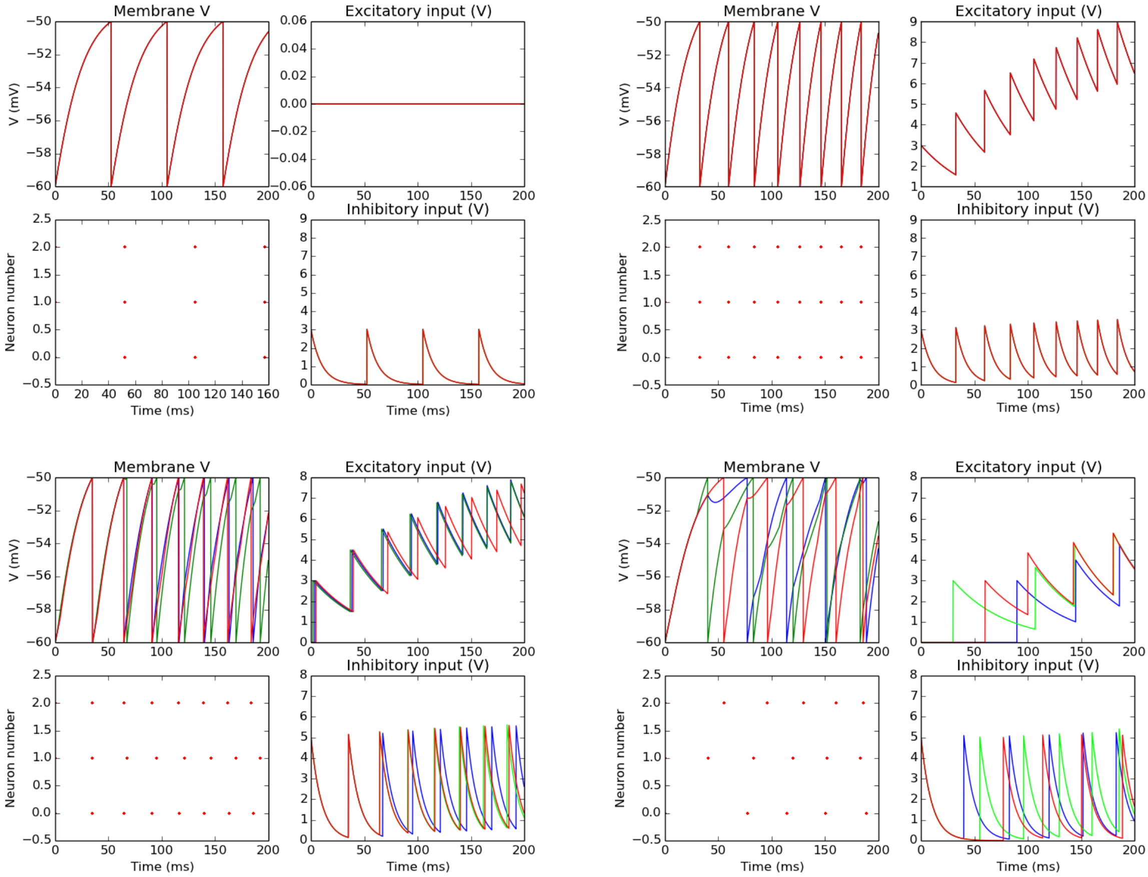

Figure 8: *Spiking Activity of the three-neuron loop in Figure 7.*
Simulations were carried out on the basis of the BRIAN code in Appendix B according to the four sets of parameters listed in Table 2. In each of the main panels the four sub-panels refer to: membrane potentials (top left) and raster plot (bottom left) of the spiking activity of each neuron; the corresponding excitatory (top right) and inhibitory (bottom right) signals received as an input by each neuron. Notice, in the bottom (left and right) panels, the progressive alteration in the synchronous occurrence due to the delay introduced in the Excitatory signals.

4 DISCUSSION

The reductionist paradigm, namely the cornerstone of many great scientific achievements in the last two centuries, assumed that any complicated system is better studied if reduced to its constituents. Such a paradigm, however, showed inappropriate to complex systems, whose most interesting behavior cannot be deduced from the behavior of isolated parts [19]: the consequences in physiological investigations, particularly in human neurophysiology, are enormous and still far from being completely appreciated. Only relatively recent studies, in fact, showed that the brain consists of spatially distributed, but functionally linked regions that continuously share information [16, 20].Thus, they form a complex integrative network where information is processed and transported between and within regions relatively homogeneous from the structural and/or functional viewpoints.

The 'holistic' paradigm, however, pointing to models of the human brain based on topological nets inspired by the Graph Theory and associated to fully connected adiacency matrices[2], implies facing many critical issues among which:

(1) the technical problems of recording reproducible patterns of synchronous, correlated responses from well defined brain regions[3];

(2) the need to single out, among the high number of connections potentially occurring within and between different cortical and subcortical regions, those more clearly related to some specific, cognitive activity;

(3) the desirable account for 'flexible' and 'adaptive' patterns of neuronal activities by dynamical models endowed with descriptive and, possibly, predictive abilities.

In spite of the continuous improvements in the space and time resolution of fMRI techniques, type (1) problems still appear formidable. This, in turn, provides serious limitations also to solving type (2) problems, since any convincing model of the observed phenomena needs confirmation against experimental data. Whenever such complications arise in getting real data to validate descriptive/predictive models, it is difficult to overestimate the importance of appropriate simulation techniques, that is adopting a somehow 'softer' approach where the main virtue of an artificial (computational) model lies in the reproduction of the performance of a natural system, not of detailed physiological mechanisms.

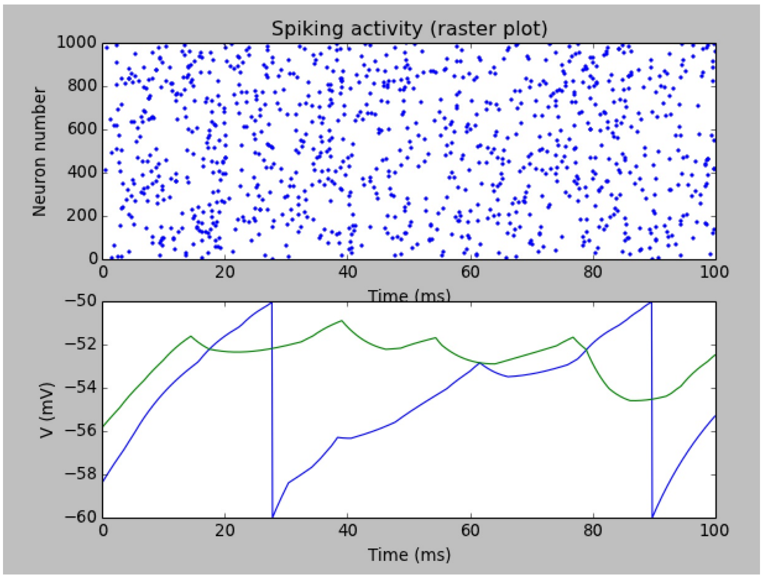

Figure 9: *Randomic links in a realistic neural net.* (modified from ref. [21]).

Following such an approach, we set up some minimal neural models based on two powerful and noncommercial neural simulation environments, NEURON and BRIAN, in order to explore two typical features of natural networks, namely the self-sustaining activities and the high sensitivity in responding to tiny stimulation changes. In our hands, each of the above mentioned environments appeared very efficient in specific simulation tasks; thus, to deal with quite complex problems, the best foreseeable perspective is a well balanced exploitation of their respective virtues. In particular, the ambitious type (3) problems require both an accurate account of the biophysical properties of single cells in a large ensamble of cooperative units , as well as a highly dynamical

[2] characterized by N*(N-1)/2 connections, where N is the number of nodes

[3] e.g. Brodmann areas

and time dependent arrangement of excitatory and inhibitory connections, which are obtained at best within the NEURON and BRIAN environment, respectively.

An appealing feature of one of those neural simulation environments, BRIAN, appears in Figure 9, showing the global spiking activity of a neural net including 1000 neurons fully connected by random links, together with the changes, along the same time window, of the membrane potential of two particular neurons. In this case, however, any subtle biophysical difference between (groups of) cells, of the type considered in the NEURON environment, is not considered, which provides a good feeling of the complexity level, and hence of the euristic power, potentially available to appropriate *hybrid* simulation strategies.

Along this pathway, and including at various stages random events in their computational counterparts, reproducing the intriguing adaptive features of physiological neural nets appears less intimidating.

REFERENCES

[1] Pilato F., Profice P., Ranieri F., Capone F., Di Iorio R. Florio G., and Di Lazzaro V. Synaptic plasticity in neurodegenerative diseases evaluated and modulated by in vivo neurophysiological techniques. *Mol Neurobiology*, 46:563–571, 2012.

[2] Dekosky S.T., Scheff S.W., and Styren S.D. Structural correlates of cognition in dementia: quantification and assessment of synapse change. *Neurodegeneration*, 5:417421, 1996.

[3] Arshavsky Y. Cellular and network properties in the functioning of the nervous system: from central pattern generators to cognition. *Brain Research Review*, 41:229267, 2003.

[4] Watts D.J. and Strogatz S.H. Collective dynamics of small-world networks. *Nature*, 393:440–502, 1998.

[5] Engle R. W. and M. J. Kane. Executive attention, working memory capacity, and a two-factor theory of cognitive control. *The Psychology of Learning and Motivation*, 44:145199, 2010.

[6] Shelton J. T., Metzger R. L., and Elliott E. M. A group-administered lag task as a measure of working memory. *Res. Methods*, 39:145199, 2007.

[7] Carnevale N.T. and Hines M. *The NEURON Book*. Cambridge, Departments of Psychology and Computer Science Yale University, New Haven, CT, 2004.

[8] Howland J. G. and Wang Y.T. Synaptic plasticity in learning and memory: Stress effects in the hippocampus. *Progress in Brain Research*, 169:145158, 2008.

[9] Hines M. and Carnevale N.T. The neuron simulation environment. *Neural Computation*, 9(6), 1997.

[10] Migliore M, Hoffman DA, Magee JC, and Johnston D. Role of an a-type k+ conductance in the back-propagation of action potentials in the dendrites of hippocampal pyramidal neurons. *J Comput Neurosci.*, 7:5–15, 1999.

[11] Goodman D. and Brette R. Brian: a simulator for spiking neural networks in python. *Front. Neuroinform*, 2:5, 2008.

[12] Rubinov M. and Sporns O. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, 52:1059–1069, 2010.

[13] De No' Rafael. Transmission of impulses through cranial motor nuclei. *Journal of Neurophysiology*, 2:402–464, 1939.

[14] Klein R.M. The hebb legacy. *Canadian Journal of Experimental Psychology*, 53(1):1–3, 1999.

[15] Tegner J., Compte A., and Wang X.J. The dynamical stability of reverbatory neural circuits. *Biol. Cybern.*, 87:471–481, 2002.

[16] Strogatz S.H. Exploring complex networks. *Nature*, 410:268–277, 2001.

[17] Bassett D.S. and Bullmore E. Small-world brain networks. *Neuroscientist*, 12:512–523, 2006.

[18] Latora V. Reti Small World : l'architettura di un sistema complesso. In *LXXXVIII Congresso SIF*. Dipartimento di Fisica e Astronomia, universit di Catania, 2002.

[19] P.W.Anderson. More is different. *Science*, 177:393421, 1972.

[20] Heuvel M.P. and Hulshoff Pol H.E. Exploring the brain network: a review on resting-state fmri functional connectivity. *Eur Neuropsychopharmacology*, 20:519–534, 2010.

[21] Brette R., Rudolph M., Carnevale T., Hines M., and D. Beeman et al. Simulation of networks of spiking neurons: a review of tools and strategies. *J. Comput. Neurosci.*, DOI 10.1007/s10827-007-0038-6, 2006.

5 Appendix A: Generating Figure 5, right panel, in the NEURON programming environment (”linear_stim.hoc” macro in CodeSegment 1).

```
objref stim[nNeurons]

i=0
Neuron[0].soma{
stim=new IClamp(0.5)
stim.del=0
stim.dur=20  // [ms]duration of pulse
stim.amp=0.1 // set to .1
}
maxsyn =  30
objref syn[maxsyn], nclist

Neuron[0].treeA[4]syn[9] = new ExpSyn(1)
Neuron[0].treeA[3]syn[10] = new ExpSyn(1)
Neuron[0].treeA[2]syn[11] = new ExpSyn(1)

Neuron[1].treeA[4]syn[6] = new ExpSyn(1)
Neuron[1].treeA[3]syn[7] = new ExpSyn(1)
Neuron[1].treeA[2]syn[8] = new ExpSyn(1)

Neuron[2].treeA[4]syn[0] = new ExpSyn(1)
Neuron[2].treeA[3]syn[1] = new ExpSyn(1)
Neuron[2].treeA[2]syn[2] = new ExpSyn(1)

Neuron[3].treeA[4]syn[3] = new ExpSyn(1)
Neuron[3].treeA[3]syn[4] = new ExpSyn(1)
Neuron[3].treeA[2]syn[5] = new ExpSyn(1)

//new NetCon(&source_v,synapse,threshold,delay,weight)

Neuron[0].soma Neuron[2].nclist.append(new NetCon(&v(0),syn[0],-20,6,0.5))
Neuron[0].soma Neuron[2].nclist.append(new NetCon(&v(0),syn[1],-20,6,0.5))
Neuron[0].soma Neuron[2].nclist.append(new NetCon(&v(0),syn[2],-20,6,0.5))

Neuron[2].soma Neuron[3].nclist.append(new NetCon(&v(2),syn[3],-20,6,0.5))
Neuron[2].soma Neuron[3].nclist.append(new NetCon(&v(2),syn[4],-20,6,0.5))
Neuron[2].soma Neuron[3].nclist.append(new NetCon(&v(2),syn[5],-20,6,0.5))

Neuron[3].soma Neuron[1].nclist.append(new NetCon(&v(3),syn[6],-20,6,0.5))
Neuron[3].soma Neuron[1].nclist.append(new NetCon(&v(3),syn[7],-20,6,0.5))
Neuron[3].soma Neuron[1].nclist.append(new NetCon(&v(3),syn[8],-20,6,0.5))

Neuron[1].soma Neuron[0].nclist.append(new NetCon(&v(1),syn[9],-20,6,0.5))
Neuron[1].soma Neuron[0].nclist.append(new NetCon(&v(1),syn[10],-20,6,0.5))
Neuron[1].soma Neuron[0].nclist.append(new NetCon(&v(1),syn[11],-20,6,0.5))

access Neuron[0].soma
```

6 Appendix B. Generating Figure 8 (right panel) in the BRIAN programming environment.

Section 1 contains the biophysical parameters and the differential equations common to all the nodes, as well as the size and the connection structure of the loop. Section 2 includes the default values of synaptic weight (we, wi) and delay time (del_e, del_i) of excitatory and inhibitory links, as well as the actual weight and delay time of each directed link; Section 3 contains the variables to be calculated for each node, namely V, ge and gi; Sections 4 and 5 define, respectively, the time span of the calculation and the plot drawing details. Notice that Section 2 contains the critical parameters for the different activation patterns shown in Figure 8 and Figure10 (Appendix C).

SECTION 1

```
from brian import *
taum = 20 * ms # membrane time constant
taue = 50 * ms # excitatory synaptic time constant
taui = 10 * ms # inhibitory synaptic time constant
Vt = -50 * mV # spike threshold
Vr = -60 * mV # reset value
El = -49 * mV # resting potential

eqs = Equations(’’’
dV/dt = (ge-gi-(V-El))/taum : volt
dge/dt = -ge/taue : mvolt
dgi/dt = -gi/taui : mvolt
’’’)

G = NeuronGroup(3, model=eqs, threshold=Vt, reset=Vr)
G1 = G.subgroup(1) # loop member neuron
G2 = G.subgroup(1) # loop member neuron
G3 = G.subgroup(1) # loop member neuron
```

SECTION 2

```
we = 0 * mV # excitatory synaptic weight (default)
wi = 0 * mV # inhibitory synaptic weigh (default)
del_e = 0 * ms # delay for excitatory links (default)
del_i = 0 * ms # delay for inhibitory links (default)
C12 = Connection(G1, G2, ’ge’, sparseness=1, weight=we, delay=del\_e*1)
C21 = Connection(G2, G1, ’gi’, sparseness=1, weight=wi, delay=del\_i*1)
C23 = Connection(G2, G3, ’ge’, sparseness=1, weight=we, delay=del\_e*2)
C32 = Connection(G3, G2, ’gi’, sparseness=1, weight=wi, delay=del\_i*2)
C31 = Connection(G3, G1, ’ge’, sparseness=1, weight=we, delay=del\_e*3)
C13 = Connection(G1, G3, ’gi’, sparseness=1, weight=wi, delay=del\_i*3)
```

SECTION 3

```
M = SpikeMonitor(G)
MV0 = StateMonitor(G, ’V’, record=0)
MV1 = StateMonitor(G, ’V’, record=1)
MV2 = StateMonitor(G, ’V’, record=2)
Mge0 = StateMonitor(G, ’ge’, record=0)
Mge1 = StateMonitor(G, ’ge’, record=1)
Mge2 = StateMonitor(G, ’ge’, record=2)
Mgi0 = StateMonitor(G, ’gi’, record=0)
Mgi1 = StateMonitor(G, ’gi’, record=1)
Mgi2 = StateMonitor(G, ’gi’, record=2)
```

SECTION 4

```
   run(200 * ms)
```

SECTION 5

```
subplot(221, title=’Membrane V’)
plot(MV0.times / ms, MV0[0] / mvolt)
plot(MV1.times / ms, MV1[1] / mvolt)
plot(MV2.times / ms, MV2[2] / mvolt)
xlabel(’’)
ylabel(’V (mV)’)
subplot(224, title=’Inhibitory input (V)’)
plot(Mgi0.times / ms, Mgi0[0] / mvolt)
plot(Mgi1.times / ms, Mgi1[1] / mvolt)
plot(Mgi2.times / ms, Mgi2[2] / mvolt)
xlabel(’Time (ms)’)
ylabel(’’)
subplot(223)
raster_plot(M, title=’Spiking activity’, newfigure=False)
subplot(222, title=’Excitatory input (V)’)
plot(Mge0.times / ms, Mge0[0] / mvolt)
plot(Mge1.times / ms, Mge1[1] / mvolt)
plot(Mge2.times / ms, Mge2[2] / mvolt)
xlabel(’’)
ylabel(’’)
show()
```

7 Appendix C. Additional spiking activity patterns generated by the three-neuron loop in Figure 7.

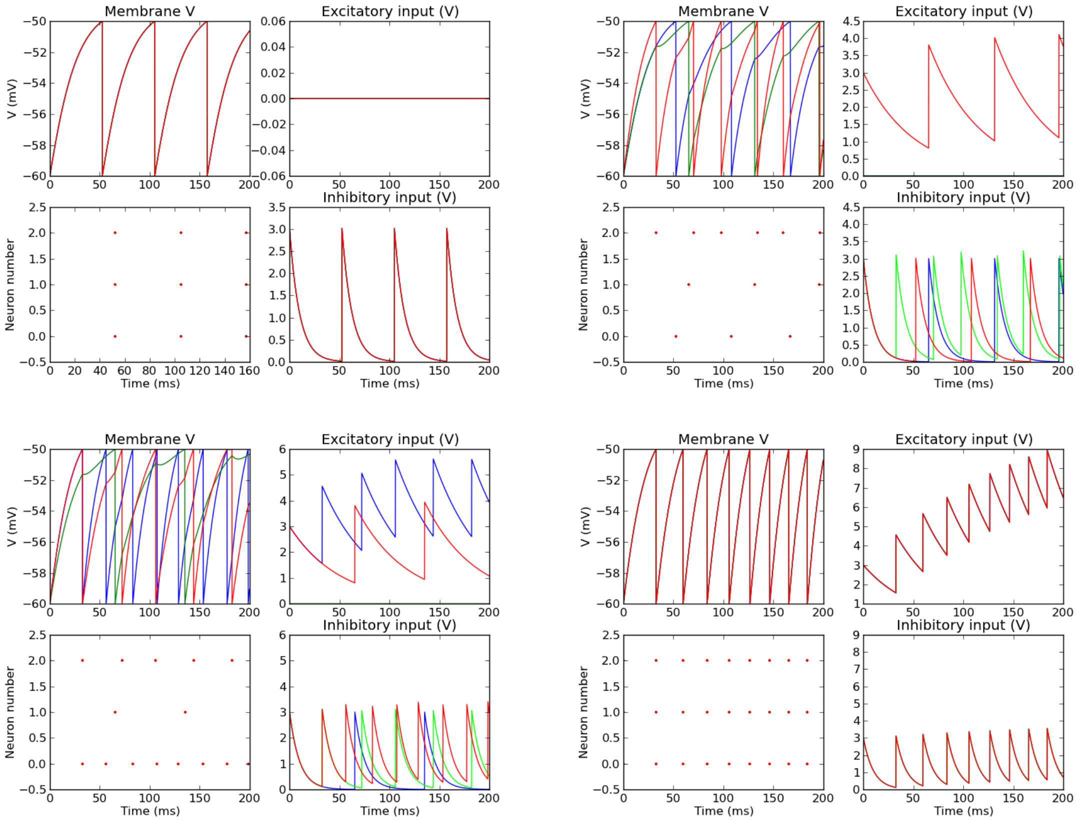

Figure 10: *Random link activation in a three-neuron loop.*
In each main panel the meaning of the four subpanels is the same as in Figure 8 (see the text for details). The four main panels are all produced by the same set of parameters, listed in the last column of Table 2. Each panel, however, refers to the activation of 0 (top left), 1 (top right), 2 (bottom left) or 3 (bottom right) excitatory inputs, and in each case the input is activated with the probability 0.5, as defined by the sparsness value = 0.5.