2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Magnitude Comparison in Analog Spiking Neural Assemblies

José Rodrigues de Oliveira-Neto, Felipe Duque Belfort, Rafael Cavalcanti-Neto, and João Ranhel

Abstract—**Spiking neural systems can represent external stimuli and internal states by means of sets of neurons firing together, the so-called cell assemblies. Neural assembly computing (NAC) is an approach that investigates how spiking neural assemblies represent things and states of the world, how interaction among assemblies results in information processing, computation and behavior. Mainly, NAC deals with digital assemblies in which all-or-none cell members are firing. The notion of analog assemblies is introduced, describing sets of neurons that represent something proportionally to their driving stimuli. Interactions among digital and analog cell assemblies create a rich computational environment. In this paper a spiking neural network that compares the magnitude of two analog assemblies is presented.**

I. INTRODUCTION

SYNCHRONICITY is a property ubiquitously found in biological neural networks, such as the visual, olfactory and auditory systems, as well as in thalamus and other neural circuits. However, it remains unclear how neural synchrony results in information processing and neural computation (see [1] for review). For decades, it was thought that neurons represent information and compute by means of neural firing rate. Traditional artificial neural networks (ANN) are strongly based on such assumptions [2]. But evidences from empirical biology have shown that precise spike timing plays an important role in representation and computation in brains. In fact, "many behavioral responses are completed too quickly for the underlying sensory processes to rely on estimation of neural firing rates over extended time windows" [3]. As a consequence, many researchers have focused on Spiking Neural Networks (SNN), the third generation of ANN [4], [5]. SNN takes advantage of individual spike timing for encoding and processing information. The association of both spike-timing and synchronism in SNN may lead to a rich computational environment, capable of dealing with input selection, consolidation and combination of learned information, binding information in cell assemblies, among others [6].

In 1982 Abeles proposed that neurons may play the role of '*coincidence detectors*' instead of mean spike integrators [7]. This hypothesis put neurons in the scene as synchronous detectors, an idea comparable to the 'reader' notion proposed by Buzsáki [8]. Neurons operating in synchronism naturally give rise to cell assemblies, an idea proposed by Hebb in 1949 [9]. A feed-forward network with pools (sets) of neurons working together was proposed by Abeles as 'synfire chains' [10]. In synchronous operation synfire chains have all neurons firing at the same time. Afterwards, it was proposed the notions of 'synfire braid' [11], and later 'polychronous groups' [12]. These concepts consider that sets of spiking neurons do not necessarily need to fire synchronously for characterizing a cell assembly activity. Instead, they may fire in a time-locked fashion.

These concepts tried to explain how sets of spiking neurons compute. In this direction, the neural assembly computing (NAC) approach was proposed [13]. NAC investigates how neural assemblies represent things and states of the world, how interactions among assemblies lead to information processing, and how it results in computation and behavior.

NAC is strongly based on synchronicity among coalitions; in such way that almost all NAC topologies rely on internal rhythm generators, a bioinspired approach (see [8], [6]). So far, NAC assemblies were treated as operating in '*digital*' mode: either all members are firing or all members are silent. By using the all-or-none operational mode it was shown that neural assemblies can perform Boolean functions, realize bistable memory without demanding any plasticity mechanism, and sustain bistable rhythmic loops. Finally, by associating these functions NAC can perform finite-state automata (FSA) [14].

Nevertheless, it is important to note that external stimuli may lead a set of spiking neurons to behave not as all-or-none digital assembly. Instead, external stimuli may cause neurons to fire '*proportionally*' to its intensity, rate of change, duration, etc. How neural assemblies can proportionally represent external stimuli intensities? What results from '*analog*' coalitions interacting with the digital ones? These questions open new investigation fields on NAC.

In this paper we address an important issue concerning the interactions among analog and digital assemblies: how can a spiking network compare two externally stimulated analog assemblies X and Y and generate digital responses indicating whether they are equal or whether one is greater than the other. Neural circuits performing such task may be basal for more complex decision-making systems.

In section II a brief review of NAC concepts is presented. Then, the synchronization of external stimuli is evaluated, and a synchronizing circuit is presented. A network that solves the magnitude comparison problem is then introduced. In section III the simulations are described and the raster plots are interpreted. Section IV holds the discussions about the presented findings. Section V has the conclusions and the next steps in this line of investigation.

The authors are with the Universidade Federal de Pernambuco, UFPE / DES. Av Arquitetura SN 4th floor – Recife – Brazil. Phone: 55 81 2126 8217; e-mail: jranhel@ ieee.org.

This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) and by the Federal Agency for Support and Evaluation of Graduate Education (CAPES).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

II. NAC AND MAGNITUDE COMPARATOR

We use the term *neuroid* following Cruse's denomination for artificial computational unities [15], whereas we use neuron for biological cells. As a consequence, action potential is related to neurons whereas the term spike is related to artificial signals generated by neuroids.

Classical ANNs, such as multi-layer perceptrons and self-organizing map (SOM), heavily depend on single *neuroids*. For example, the SOM training algorithm may result in a specialized *neuroid* that is singly active when specific input patterns are presented to the network. Such dependency is less acceptable in biological neural networks and in SNN. Once a single spike lasts for one millisecond, it becomes necessary to devise a robust way for representing and processing information involving several unities. Single computational unities representing and processing information is an idea hardly accepted in biological terms, which resemble the idea of the Grandmother Cell [16]. Although in simple organisms single neurons may perform key computation, in complex animals representations are done sparsely, with many neurons participating in information retention [17], as well as realizing distributed parallel computation.

The idea that information might be represented by groups of neurons firing together was proposed by Donald Hebb in his book "The organization of behavior" [9]. Since then, there have been discussions about the nature of such assemblies: how they come about, how they represent things, how it results in computation and behavior, among other issues. One concern is over the coalition *spatiality*: do neurons need to be physically close to each other for firing as assemblies? Sometimes this physical aggregation is implicitly suggested, although other scientists do not impose such physical restriction [8], [18].

One important issue is that action potential propagate with finite and variable velocity, depending on the neuron types, the regions in the nervous systems, the type of synapses, the axon properties (its diameter, whether it is myelinated or not), among other issues [19]. It means that action potentials are likely to reach different cells in different time intervals. Hence, they are more likely to reach neighboring neurons, but once action potentials may radially propagate in the medium, several regions may be reached simultaneously. This fact reinforces the idea that cell assemblies may occur with members firing in different spatial location in the nervous tissue. In this work, analog and digital assemblies will be considered, without regards to whether the unities are close to each other or not.

Concerning the coalition response: must all members fire together or may some of them fail without taking the coalition apart? Sometimes it is considered that all members fire together because the cell members have mechanisms of triggering their neighbors [20], [10]. In this case, the all-or-none behavior suggests a '*digital*' form of operation in which the assembly is ON (the event happens, it is '1') or OFF (the event does not happen, it is '0'). However, when excited by external stimuli that vary in intensity, for instance, cell assemblies may behave differently. In this sense, it might be useful to consider the idea of '*analog*' assemblies, where a variable number of neurons or *neuroids* fire proportional to the intensity of the external stimuli.

A. NAC Review

As said before, NAC is an approach that tries to explain how computation is carried out *by* and *in* cell assemblies. Basically, when correctly excited, a set of neuroids might fire together, forming a coalition. Such assembly, in turn, might produce a stimulus strong enough to fire or inhibit other neuroid assemblies.

The combination of firing and inhibition is the cornerstone of NAC. One can easily spot its similarities with digital logic computation. Indeed, it has been shown how logical functions can be implemented using NAC concepts. As an example, suppose an assembly *A* can trigger another assembly *C* alone (denoted $C \leftarrow A$, and read *C* is caused by *A*). It means that the spikes generated by the *A* neuroids will arrive at *C* after some milliseconds. Let us denote Δt this mean propagation delay. Consider the synaptic weight among all members from *A* to all members *C* calculated by:

$$w_{j,k} \leftarrow \Theta/k \tag{1}$$

where *w* is the synaptic weight from neuroid *k* (pre-synaptic) to *j* (post-synaptic), *k* is the number of members in *A*, and Θ is the excitatory post-synaptic potential (EPSP) that makes a *C* member to fire.

Now consider that two assemblies *A* and *B* are connected to *C* by *w*/2 (half of the synaptic strength necessary for firing any neuroid in *C*), meaning that neither *A* nor *B* firing alone can trigger *C*. In this case, consider that *A* and *B* fire simultaneously and that the propagation delay Δt is equal for both; consequently, all spikes reach *C* coincidently. It is necessary that *A* AND *B* contribute with spikes for the EPSP in *C* be high enough for triggering its neuroids. In other words, the event *A* and the event *B* are necessary for triggering *C*, which is equivalent to the logical function AND. On the other hand, suppose *A* and *B* both connected to *C* with synaptic weights *w*. It means *A* can trigger *C* singly, but *B* can also do it. In other words, *A* OR *B* can trigger *C*, performing the OR logic function. All logical functions can be implemented in NAC [13].

Reverberating assemblies in NAC can perform bistable memories as well as pacemakers or internal rhythm (*clock* signals). A loop of reverberating assemblies is implemented when an assembly *D* triggers an assembly *E* that triggers an assembly *F* that triggers back the assembly *D* (denoted by $D \rightarrow E \rightarrow F \rightarrow D$). This loop can be active (ON) or inactive (OFF), so a loop memorizes one bit of information, which is equivalent to an electronic flip-flop circuit. Bistable assemblies may be dismantled by inhibitory coalitions performing NOT, NOR, or NAND functions.

These are the main components used for constructing computers. NAC can also implement FSA [14], the entry-level for computing machines.

B. External Stimuli

In real world, external stimuli cannot be predicted in terms of synchronism or in intensity. It means that outside stimuli may not occur synchronously to internal rhythms. But stimulus intensity may carry important information that cannot be unvalued by the nervous system. Thus, let us consider that assemblies may fire not only for representing an active (coalition='1') or inactive stimulus (coalition='0'). It becomes necessary to come up with new ways for representing external stimuli attributes, perhaps reflected into the assemblies' behavior.

In neuroscience literature, synchronism and timing on real neurons have been analyzed by many [21], [3], [6], [1]. Intensity, for instance, can be represented by the number of assembly's active members, which may be somehow proportional to the external stimuli. Coalitions that behave this way are called 'analog' assemblies.

In this work, two analog assemblies (X and Y) are deployed to mimic the neural response to two different input stimuli. Each assembly varies independently in its own pace. The goal is to devise a neural assembly magnitude comparator whose output indicates which input stimulus is stronger. In other words, a NAC network is constructed that outputs three digital assemblies: one firing only when $X > Y$, another one firing when $X<Y$, and the last one firing only when $X \cong Y$.

C. Topology

A spiking network that realizes magnitude comparison was developed. Fig. 1 shows the topology used for the comparator network. The Izhikevich's simple model of neuroid was chosen due to its capabilities on mimicking different types of real neurons, combining plausibility with computational efficiency. We have changed the parameters {a, b, c, d} in order to simulate different neuron types (see [22]).

The network has worked well with several neurons types even under severe noise situation. The results are described below and are presented in the Table 1.

The topology can be roughly divided in three blocks. One is responsible for the system rhythm, represented by the A_3, A_4, and A_5 digital assemblies. Another block simulates analog input stimuli, represented by the X and Y assemblies. Finally, there is a block that compares the number of firing members in X and Y. This block generates digital events as outputs, represented in the diagram by A_7, A_8 and A_9 digital assemblies.

1) Pacemaker: The *rhythm* is created with A_3 triggering A_4 that triggers A_5 that triggers A_3 back. The sequence defines how the assemblies are connected: all A_3 neuroids (pre-synaptic) are connected to all A_4 neuroids (post-synaptic); all A_4 neuroids are connected to all A_5 neuroids; and all A_5 neuroids are connected to all A_3 neuroids.

The synaptic weights (w) are calculated by (1). These assemblies create a time base ensuring synchronization.

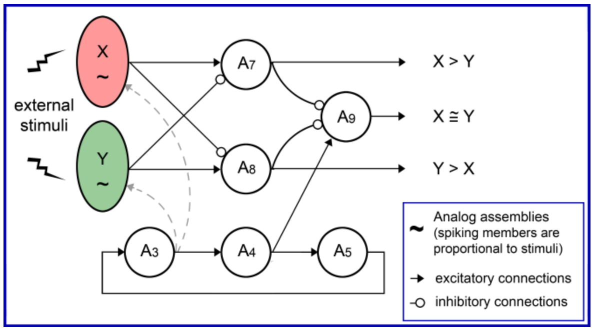

Fig. 1. Topology: X and Y are input analog assemblies excited by external stimuli. In response, they vary the number of firing members proportionally to the external stimuli. A_3, A_4, and A_5 form a loop responsible for synchronizing the process. A_7 compares if X is greater than Y whereas A_8 compares if Y is greater than X. The assembly A_9 detects when neither A_7 nor A_8 is firing, therefore $X \cong Y$.

As A_3 fires, its spikes take Δt_1 milliseconds to reach A_4. Afterwards, A_4 fires and it takes Δt_2 ms for its spikes to reach A_5. Subsequently, A_5 fires and it takes Δt_3 ms for its spikes to reach A_3. The cycle repeats indefinitely. The total propagation delay is obtained by the sum of all delays in the network:

$$t = \Delta t_1 + \Delta t_2 + \Delta t_3 \tag{2}$$

For simplicity, consider the delays are equal. It means the network has three well-determined events that cyclically repeat at regular time interval, which can be used to determine when information is processed. By doing so, different phase shifted operations may occur in NAC. Functions and information processing occur when '*events*' are triggered in NAC, and such events happen only when A_3, A_4, and A_5 fire. Note that assemblies are brief and ephemeral events, and most of the time information is travelling along axons.

2) Formulas: As stated before, each neuroid (j) in a post-synaptic assembly F needs a perturbation Θ_F in order to fire. Once a spike only assumes the values '1' (the pre-synaptic neuron fired) or '0' (it did not fired), such perturbation is expected to come from k connections to a triggered pre-synaptic coalition A. This can be denoted as:

$$\Theta_F \leftarrow \sum_{i=1}^{k_A} 1 \, . \, w_{i,j} \tag{3}$$

Note that this formula do not denote an equality, and the '$\leftarrow$' symbol indicates a causality, which means Θ_F is caused by the formula right hand term. Θ_F is the EPSP received by a neuroid (j) in F, i is the pre-synaptic neuroid index, k_A is the number or neuroids in pre-synaptic assemblies, and $w_{i,j}$ is the synaptic weight between a neuroid i and j.

Let us consider that all synaptic weights w are equal among all neurons from A to F, thus, the formula reduces to:

$$\Theta_F \leftarrow k_A \, . \, w \tag{4}$$

The stimulus that an assembly A induces in each neuroid of F after a delay Δt is given by (4). The same calculation can be used by inhibitory assemblies just considering to change the synaptic weight signal. For simplicity, all synaptic weights among coalition member are equal, as well as all propagation delays among the assemblies in the pacemaker. These assumptions make easy to calculate the strength connections as well as the network timing.

However, external stimuli are independent from such rhythmic events. In order to guarantee that information from X and Y are processed synchronous to the pacemaker, it was created a biased-gate input circuit, which is inspired in biological oscillation-gating functions (see [6]).

3) Analog Input Synchronization: Consider two coalitions Z and X with equal number of members. Each i member in Z (z_i) is connected to a single paired i neuroid in X. The synaptic weight for such connection is $w/2$, therefore, each neuroid in Z cannot fire its correspondent neuroid in X by itself. From (1) it is possible to see that each neuroid from Z can only cause half of the perturbation Θ required for firing X neuroids. However, if another signal simultaneously contributes with $\Theta/2$, neurons in X are able to fire. In fact, this additional signal is acting as an *enabler*, which allows X to fire only when it is active. This enable signal is provided by assembly A_3 in the network. By using (1) it is also possible to calculate the synaptic weight from A_3 to X.

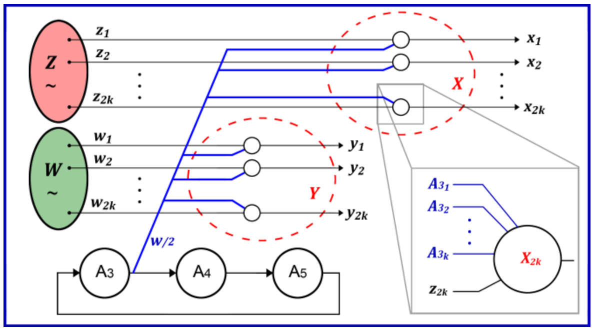

Fig. 2 Enabled-gated circuit: each input neuroid Z is connected to a single paired neuroid in X, and each neuroid in W is connected to a single paired neuroid in Y. Neuroids from A_3 are fully connected to both X and Y neuroids by half of the necessary synaptic weight. A_3 enables firing neuroids in Z and W to trigger their correspondent inter-neuroids in X and Y respectively.

In summary, the contribution of all neuroids firing in A_3 causes an excitatory *bias* in all X neuroids, but this perturbation is not enough for triggering X neuroids. Exactly when spikes from A_3 are reaching X neuroids, the spikes from Z can contribute with $\Theta/2$ to its correspondent X neuroid pair. It results that the only triggered neuroids in X are those whose equivalent neuroid in Z is firing when A_3 is active. The same reasoning can be applied to W and Y neurons.

It means that the network *samples* X and Y every $3\Delta t$. The enabled-gate circuit that synchronizes the analog inputs to A_3 is shown in Fig. 2. In Fig. 1, only the connections from A_3 to X and Y are represented by the dashed lines.

4) Magnitude Comparator: The previously presented formulas make it possible to determine the perturbation in the assemblies A_7 and A_8. Fig. 1 shows that these assemblies are connected by reciprocal excitatory $+w$ and inhibitory $-w$ connections. It can be denoted as:

$$\Theta_7 \leftarrow (k_X\,.w - k_Y.w) \quad (5)$$

$$\Theta_8 \leftarrow (k_Y\,.w - k_X.w) \quad (6)$$

These formulas can be reduced to:

$$\Theta_7 \leftarrow (k_X - k_Y).w \quad (7)$$

$$\Theta_8 \leftarrow (k_Y - k_X).w \quad (8)$$

It means that perturbations on A_7 neuroids depend on the difference between the number of excitatory firing member in X and the number of inhibitory firing members in Y. The opposite occurs in A_8. Consider multiplying w by a constant α (with $2 \leq \alpha \leq 10$). The factor α allows us to control how large the difference between X and Y must be. A small difference multiplied by a great α may cause the necessary perturbation Θ_7 *or* Θ_8. We have experimented some values for α and, in the described simulation we have used $\alpha=4$. As aforementioned, A_7 and A_8 are digital assemblies and they fire all or none of their members. Therefore, A_7 may fire when X is slightly greater than Y, and A_8 may fire when Y is slightly greater than X.

The coalition A_9 must fire when the difference of firing members in X and Y is not enough for triggering A_7 or A_8. In other words, if neither X is greater than Y nor Y is greater than X, then neither A_7 nor A_8 are active, so A_9 must fire.

The simplest way to create a network to solve this problem is to use A_7 and A_8 as inhibitory stimuli for a circuit that is always active. But it takes some time to obtain A_7 and A_8 responses. More precisely, after A_3 firing (sampling X and Y) the network spends Δt for X and Y spikes to reach A_7 and A_8. Then, it takes another Δt for spikes from A_7 and A_8 to reach A_9. Simultaneously, in the pacemaker, A_3 triggers A_4, and since A_4 is connected to A_9, then spikes from A_4 reach A_9 at the same time that spikes from A_7 and A_8 do. It can be denoted as:

$$\Theta_9 \leftarrow (k_4.w_4 - k_X.w_X - k_Y.w_Y) \quad (9)$$

This is how the equality indicator works: A_4 is always triggering A_9 but whether A_7 or A_8 are firing they inhibit the excitatory spikes from A_4 and the assembly A_9 becomes silent. Therefore, A_9 only fires when neither A_7 nor A_8 are active.

III. METHODS AND SIMULATIONS

In order to simulate the proposed topology we used Matlab (the codes can be obtained from the author upon request). The results can be seen in Fig. 3, which shows a raster plot with the distribution of spiking neuroids in time domain. Each dot means that the neuroid n fired at time t *ms*, where n is in Y-axis and t is in X-axis. As a mean to facilitate visualization, each assembly has their members assigned sequentially, e.g. neuroids from 1 to k belong to A_0 assembly, from $k+1$ to $2k$ belong to A_1, and so on.

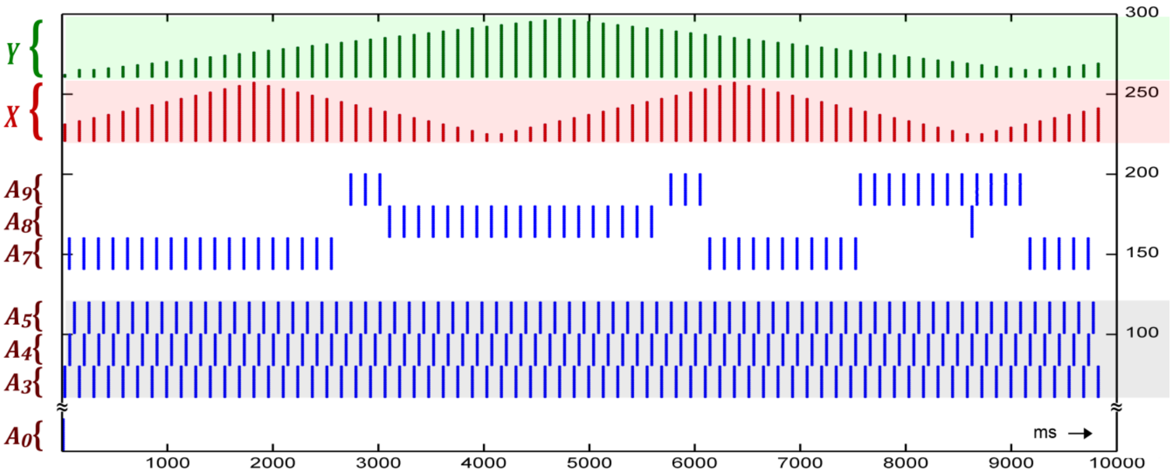

Fig. 3 Raster plot. Each assembly has k=20 neuroid members. A_3, A_4, and A_5 form pacemaker for the spiking network: A_3 triggers A_4 that triggers A_5 that triggers A_3, closing the loop. The assemblies X and Y has 2k variable firing members each, both excited by external stimuli. A_7 neuroids fire when X is greater than Y whereas A_8 neuroids fire when Y is greater than X. A_9 only fires when there is no winner, therefore $X \cong Y$.

In this simulation, the assemblies were composed by twenty neuroids each. The exceptions were the analog assemblies X and Y, with forty neuroids each. Assemblies operate in synfire mode. The synfire chain is a feed-forward network with multiple layers where spikes mainly propagate synchronously between layers [10]. In Fig. 3 the synfire effect can easily be observed because the neuroids of each assembly fire at the same time.

Back to the simulation, the assembly A_0 merely triggers the spiking neural network, A_1 and A_2 are not used. As A_0 fires, it starts the pacemaker loop formed by the assemblies A_3, A_4 and A_5. As mentioned before, this pacemaker is responsible for synchronizing the other processes, and this bistable loop fires indefinitely.

At the top of Fig. 3 it is possible to see the assemblies X and Y. They vary the number of firing members as time increases. Note that X and Y vary in different pace in order to cover all possible outcomes from the comparator.

When X is greater than Y, it is possible to observe that only the coalition A_7 fires. On the other hand, only the assembly A_8 fires when Y is greater than X. There is a 'gap' in which the circuit does not respond for $X>Y$ or for $Y>X$. Such interval can be acceptable and, sometimes, desirable. It shows that the network only indicates majority with a degree of certainty. The equality is indicated by the assembly A_9, which is expected to fire only when neither A_7 nor A_8 are firing.

Because NAC has a synchronous operational mode some events are triggered but its response comes with certain delay. That is why sometimes an overlapping situation may occur, when A_9 may firing indicating equality and at the same time A_8 or A_7 may fire indicating majority. Such behavior can be seen once in Fig. 3 around 8500 milliseconds.

A. Performance Under Noise

Several simulations were performed with different neuroid types under different noise situation.

Noise (ε) is injected through all pre-synaptic connections, generated by the function *randn()* in Matlab. When no noise is injected (ε=0) the membrane potential (signal) measured in all neuroids is the resting potential (e.g. −65 mV). As noise is injected neuroids may occasionally fire or be inhibited. Table 1 shows the network response for some chosen neuroid types under certain noise conditions.

Firstly, perturbation Θ is adjusted for each neuroid type in such way that, in absence of noise, the network performs the function shown in Fig. 3. Then, noise is injected and the network is tested for two conditions: (*i*) it is checked whether the comparator works well with a 3.52 dB noise (1.5 times the amplitude of resting membrane), (*ii*) it is checked whether the comparison function is no longer executed with 6 dB of noise (the noise with the double of amplitude compared to the resting potential).

It is possible to observe that the comparison function may be performed in NAC even in noisy networks. Only one of the chosen neuroid types did not work when 3.52 dB of noise is applied to the network. These chosen neuroids still worked with 6 dB noise. It was chosen neuroid types that respond for excitatory inputs, avoiding those that respond with bursts.

TABLE I
RESULTS FOR DIFFERENT NEUROID TYPES UNDER NOISE

Neuron Type	Θ min	No Failures ε(dB)	Total Failure ε(dB)
Tonic spiking	21.5	< 3.52	> 6
Phasic spiking	13.2	< 3.52	> 6
Mixed Mode	22.1	< 3.52	> 6
Class 2	5.5	< 2.00	> 6
Spike Latency	27.2	< 3.52	> 6
Regular Spiking	22.5	< 3.52	> 6
Fast Spiking	18.6	< 3.52	> 6

Θ min – minimal perturbation for stable comparison operation.

No Failures – obtained by using ε = 3.52 dB without any failure.

Total Failures – obtained after seven tries in which none of them the network presented successful comparison.

IV. DISCUSSIONS

Magnitude comparators play a key role in intelligent agents, for instance, an animal might face a situation in which it must compare the color coming from two nearby fruits in order to decide which one is better for eating. Moreover, such agent may put together results from different comparisons: which fruit smells more intensively, for instance. By associating the results of magnitude comparators it is possible to create sophisticated decision-making systems. In other words, despite being a simple decision making tool, it is worth to study and to implement this magnitude comparison network in order to build much larger and complex decision making systems in a bottom-up approach.

Some pendent issues on magnitude comparison in NAC include: can assemblies represent proportional (analog) stimuli by means other than the number of firing members? In this case, how could a network realize the comparison task in such new representation forms? Concerning this circuit, we shall now try to associate this comparator in more complex constructions, for instance, by associating this 'block' and FSA. By doing so we believe that it is possible to construct very complex algorithms.

V. CONCLUSION

The neural assembly computing approach is used in this work to create a spiking network that compares two variable and independent stimuli and determine whether they are equal or which one is the greatest. The notion of analog assemblies is introduced. Analog coalitions are synchronized to an internal pacemaker. Such pacemaker creates a rhythm for the spiking neural assembly network.

In order to represent the magnitude of the external stimuli, we choose to proportionally vary the quantity of firing members in analog coalitions. These inputs are applied to the network that correctly indicates when one assembly is greater than the other or when they are equal. This magnitude comparator may be used as a basal circuit (a block) for more complex decision-making systems.

REFERENCES

[1] R. Brette, "Computing with neural synchrony," *PLoS Comput Biol*, vol. 8, no. 6, p. e1002561, 06 2012.

[2] R. Rojas, *Neural Networks: a Systematic Introduction*. Berlin: Springer, 1996.

[3] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends in Neurosciences*, vol. 28, no. 1, pp. 1–4, 2005.

[4] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International Journal of Neural Systems*, vol. 19, no. 04, pp. 295–308, 2009.

[5] H. Paugam-Moisy and S. Bohte, "Computing with spiking neuron networks," in *Handbook of Natural Computing. 1st Edition*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds. Heidelberg, Germany: Springer-Verlag, 2010, vol. 1, pp. 1–47.

[6] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, 2004.

[7] P. König, A. K. Engel, W. Singer "Integrator or coincidence detector? the role of the cortical neuron revisited." *Trends in Neurosciences*, vol. 19, no. 4, pp. 130 – 137, 1996.

[8] G. Buzsáki, "Neural syntax: Cell assemblies, synapsembles, and readers," *Neuron*, vol. 68, no. 3, pp. 362–385, 2010.

[9] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. Mahwah, NJ: Lawrence Erlbaum Associated, Inc, 2002, originally published: New York: Wiley, 1949.

[10] M. Abeles, "Synfire chains," *Scholarpedia*, vol. 4, no. 7, p. 1441, 2009. [Online]: http://www.scholarpedia.org/article/Synfire_chains (accessed: 2014/16/01).

[11] E. Bienenstock, "A model of neocortex," *Network: Computation in Neural Systems*, vol. 6, no. 1, pp. 179–224, 1995.

[12] E. M. Izhikevich, "Polychronization: computation with spikes," *Neural Computation*, vol. 18, no. 2, pp. 245–282, 2006.

[13] J. Ranhel, "Neural assembly computing," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 916–927, 2012.

[14] J. Ranhel, "Neural assemblies and finite state automata," in *Proceedings of 1st BRICS Countries Congress (BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence*, vol. 1, 2013, pp. 1–6. [Online]. http://www.neuralassembly.org/download/-NAC_Finite_State_Automata_BRICS-CCI-2013.pdf (accessed: 2014/16/01).

[15] H. Cruse, *Neural Networks as Cybernetic Systems. 3rd and revised edition*. Bielefeld, Germany: Brains, Minds & Media, 2009.

[16] C. G. Gross, "Genealogy of the grandmother cell," *The Neuroscientist*, vol. 8, no. 5, pp. 512–518, 2002.

[17] P. Foldiak and D. Endres, "Sparse coding," *Scholarpedia*, vol. 3, no. 1, p. 2984, 2008.

[18] A. K. Engel, P. Fries, and W. Singer, "Dynamic predictions: oscillations and synchrony in top-down processing," *Nature Reviews Neuroscience*, vol. 2, no. 10, pp. 704–716, 2001.

[19] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, *Principles of Neural Science. 4th ed.* New York, NY: McGrall-Hill Health Prof. Division, 2000.

[20] J. Wickens and R. Miller, "A formalisation of the neural assembly concept 1.constraints on neural assembly size," *Biological Cybernetics*, vol. 77, no. 5, pp. 351–358, 1997.

[21] D. A. Butts, C. Weng, J. Jin, C.-I. Yeh, N. A. Lesica, J.-M. Alonso, and G. B. Stanley, "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.

[22] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.