Natural Signal Classification by Neural Cliques and Phase-Locked Attractors

I. Raichelgauz, K. Odinaev, Y.Y. Zeevi, Technion

Abstract — **Cortical neural networks are responsible for identification, recognition and classification of natural signals mediated by various sensory channels. These tasks are still too complex to be accomplished by state-of-the-art engineering systems. There is, therefore, a great deal of interest in the development of suitable biologically-motivated architectures which are based on a realistic model of generic neural ensembles. We present a computational architecture for classification of natural signals, such as physiological signals, based on the emergence of instant neural cliques and phase-locked attractors in liquid architectures. The emergence of instant neural cliques enables mapping of complex classes of signals onto specific spatio-temporal firing patterns. The convergence of neural cliques onto attractors, along phase-locked pathways, reveals a new type dynamic behavior of neural ensembles, which lends itself to simple discrete-output computational systems.**

I. INTRODUCTION

Cortical networks are, perhaps, the most computationally-powerful class of dynamical systems. Cortical networks process in real-time complex temporal inputs and capture complex structures and relationships in massive quantities of low precision, ambiguous, noisy data in rapidly varying environments. Such cortical networks differ in most of the fundamental aspects from computational systems and models used by conventional technologies. In particular, components of cortical networks, neurons and synapses, are very slow (ms. scale), multi-parallel, heterogeneous, highly recurrent and sparsely connected. It is obvious that the conventional computational paradigms, such as Turing-machine or Hopfield-type attractor-neural-networks, are not applicable to such cortical-type systems.

Early, well-known, theoretical investigation of computational properties of recurrent neural networks with limited dynamics [1], revealed fixed-point attractors and enabled computation of complex problems, such as the traveling salesman problem [2]. Theoretical studies [3] discovered that the number of Hopfield attractors (fixed-point states) may be as large as the number of neurons (i.e. given a Hopfield network of N neurons, with an arbitrary matrix of connectivity W, the number of state vectors that can be made stable is upper-bounded by N). However, experimental results indicated that the number of attractors attainable with such networks may be too small for practical applications. Hopfield networks are composed of symmetric weights, and thus the dynamics of the system is limited. Since the activity of the elements always evolves to one of a set of stable states which is then kept forever, only fixed-point attractors can emerge in the dynamics of the system.

Recently, T. Natschläger, W. Maass, and H. Markram have developed the framework of Liquid State Machine (LSM), which is based on a more biologically realistic and dynamically rich model of spiking neural network [4]. LSM consists of a filter L^M, implemented by a complex recurrent network of spiking neurons. It maps input streams $u(t < t_0)$ onto static liquid-states $x(t_0)$, in a nonlinear manner. Then, memoryless function f^M maps at any time t_0 the liquid-states onto some target output. To generate this readout function, a layer of linear perceptrons is trained to find the required classes of equivalences in the liquid-state space, dictated by the given task. Obviously, cortical modules are not composed of these two functionally different components – liquid-states generators and readouts. However, by this simplification and by emphasizing that cortical networks, rather than individual neurons, should be viewed as basic computational units, the LSM computational framework suggests a radically different paradigm for neural computation. Moreover, the LSM framework enables the application of real cortical networks in real-world tasks by embodiment of cortical neural culture in artificial environments [5].

Following the concept of LSM, liquid-state was extended to spatio-temporal firing structure in a limited time-window, called neural clique [6], [7]. In the sequel, we implement neural cliques for mapping classes of signals to specific firing patterns. We then present a recently revealed dynamic behavior of neural ensembles in form of limit-cycle phase-locked attractors. This dynamics is sensitive to initial conditions injected into the network, and is computationally effective in mapping complex cliques onto discrete outputs. Finally, both concepts are implemented in a computational system and tested in voice recognition benchmark.

II. NEURAL CLIQUES

A computational model of generic neural microcircuits is inherently endowed with powerful and versatile information processing capabilities. We use a model similar to [8], composed of a 3-dimentional recurrent network of 300

Leaky-Integrate-and-Fire (LIF) neurons [9] with random connectivity of statistics similar to generic cortical microcircuit -- 20% of the neurons chosen to be inhibitory and 80% excitatory. The probability of interconnection depends on the distance between two neurons,

$$C \cdot \exp(-D(i, j) / \lambda^2), \tag{1}$$

where λ and C are parameters that determine the average number of connections for a certain Euclidean distance D between neurons i and j. This connectivity characterization by primary local connections and a few longer connections is biologically realistic. Random, heterogeneous parameters of neural-microcircuit (NM) model fit cortical data [4]. Synaptic short-term plasticity of the NM is implemented by dynamic synapses in which the amplitude of each post-synaptic current depends on the spike train that is impinging on the synapse [10], [11], and causes facilitation and depression processes. The model is implemented using CSIM simulator [8].

The concept of "Cliques" within neural ensemble was presented in [12] and implemented in computational system of interaction between two neural ensembles and learning-by-dispersion. Neural clique is a spatio-temporal structure determined by a firing-pattern of cluster of neurons. These spatio-temporal patterns are sensitive to input signals and are indicative of network states. The concept of "Synfire-chains", originally introduced by M. Abeles [13] and recently confirmed in neural culture activity by R. Yuste and associates [14], [15], emphasizes the importance of correlated spatio-temporal firing patterns generated by neural ensembles and their relevance to connectivity characteristics. Synfire-chains highlight the importance of interrelated spatio-temporal firing patterns across a relatively wide range activity and of its relevance to neurobiological functions. However, there is a missing link towards understanding the code of spatio-temporal firing patterns in the context of task-dependent activity and perceptual elements. For this reason, we address the concept of functionally higher spatio-temporal patterns, produced by neural ensembles, "Cliques". A clique may be composed of coherent interrelated activity of several synfire-chains active in the same or even different brain loci and structures. As such as it does not directly depend on the connectivity characteristics of the firing/non-firing neurons. The meaning of the clique as a spatio-temporal cube of activity of finite temporal length [12] is determined by short-term dynamics of the neural ensemble. Synfire-chains serve as the carriers of cliques, whereas the latter are composed of subsets of neurons participating in the activities of several synfire-chains.

Mapping onto cliques has different properties than the mapping onto liquid-states, applied in LSM framework. A clique has an important temporal structure within a limited time-window, characterized by short-term dynamics of the ensemble (~50ms in our model). Moreover, LSM framework is based on separation property, suggesting that injections of any two sufficiently different inputs $u(\cdot)$ and $v(\cdot)$ will drive the liquid into sufficiently different liquid-states, represented by spatial constant vectors at a given time t_0 -- X_u and X_v [4]. In contrast to LSM, cliques perform a selectivity property, mapping an entire certain class of inputs onto the same clique. This property leads to computational capability of identification and classification of certain classes of natural signals. Moreover, natural signals processed by different sensory systems, may be mapped onto cliques by different neural ensembles. Then, convergence of different sensory sources towards a single clique is possible by mapping multiple cliques onto a higher-level clique in subsequent layers. Specifically, we examine the emergence of cliques through interaction of two neural ensembles. As indicated in the framework of Fig. 1, the activities within the two ensembles are coupled by means of the cliques.

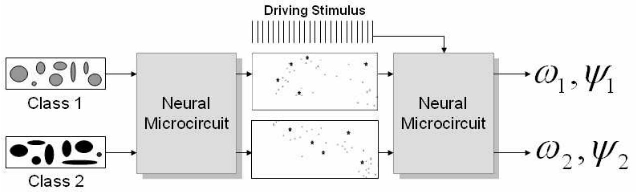

Fig.1. Block diagram of the computational framework. Inputs from 2 classes are injected into first microcircuit, which maps them onto cliques. The output of first microcircuit (including the cliques) is injected as initial condition to the second microcircuit. Second microcircuit is driven by a driving stimulus to phased-locked-attractors according to the initial conditions it is received. The phased-locked-attractors are characterized by frequency (ψ), and phase (ω).

In this study we focus our exposition on the functional aspects of cliques. Mechanisms that enable generation and transformation of cliques through pipeline of neural ensembles are macro "liquid-currents" within neural ensembles [12]. We suggest that on the macro-level of neural ensemble, the discussed mapping is implemented by existing specific liquid-currents from the group of input-neurons to the group of output-neurons, which in turn read out by the next layer of neural ensembles. To generate the appropriate subsets of cliques, a neural ensemble is required to be sensitive, in the process of cliques' generation, to specific class of inputs, and insensitive to other inputs. Capturing of the intrinsic dimensions of the data in the context of a given computational task is characterized by redundancy [12].

The intensity of liquid-current (*LCI)* generated by a certain class of inputs k to a spike j, in imaginary spatio-temporal layer l, over the subset of neurons identified by index i, is defined by the following recursive algorithm:

$$LCI_{k,j,l} = \frac{\sum_i e^{\frac{-((t_0-\tau_i)-t_i)}{\tau_a}} w_i LCI_{k,i,l-1}}{K_{\max} w_{\max}} \quad \Big| t_0 - \tau_i > t_i > t_0 - t_w \tag{2}$$

$$LCI_{k,j,l=l_{input}} = 1,$$

wherein, t_0 is the time of the spike *j*, and t_i is the time of the spike *i*, in preceding imaginary spatio-temporal layer, *l-1*. w_i is the strength of the synapse connecting neuron producing spike *j* and neuron producing spike *i*, and τ_i is the delay of this synapse. The algorithm scans neuronal spikes at layer *l-1,* which have led to the spike of the neuron at the next layer, *l*. The concerned time window is equal to the time lapsed between the activation of the synapse and the firing of the spike *j*, t_w precedes the time of spike *j* by 10ms. The probability of spike *i* to lead to spike *j* is infinitely small. In other words, the *LCI* is defined by the tree of neurons along which the signal "back-propagates" from the target spike, *j*, to the input *k*. The *LCI* is normalized by the product of the maximal synaptic strength, $w_{\max}$ and the maximal number of synaptic connections per neuron, $K_{\max}$. The *LCI* at the input layer, l_{input}, is set to *1*. This determines in turn is the stopping condition of the recursive algorithm. Note that, the terminology of layers, depicted in Fig. 2, refers to neuronal spikes at different time steps for a certain input. The fact that the same neuron may receive input either from the external environment or from interneurons highlights the recurrent connectivity characteristics of the neural ensemble. Note that the definition of *LCI* is valid also for non-firing spikes, i.e. the absence of spike for a certain neuron at certain required time-window. Since cliques are spatio-temporal sequences of neuronal spikes, we define the strength of clique *q* for input *k* by,

$$C_{k,q} = \prod_{n} \sum_{m} LCI'_{k,n,m} \quad \left| \begin{array}{l} LCI' = LCI_{firing} \\ LCI' = 1 - LCI_{non-firing} \end{array} \right. , \qquad (3)$$

where, *n* refers to the number of constraints on spikes expected by the environment or the subsequent neuronal ensemble, and *m* identifies accordingly the number of degrees of freedom within each constraint. Here we assume that there is no statistical dependency between the spikes, since spikes defining a clique are not connectivity-dependent and may exist in different and far brain loci and structures. The strength of cliques, based on definition of *LCI* (2), refers to the immunity of the cliques, generated by neural ensemble, as a function of variations in the presented signals. Thus, for example, longer liquid-currents are less reliable, while parallel liquid-currents increase the immunity of the neural ensemble to variations in signal, and result in more stable cliques.

III. PHASE-LOCKED ATTRACTORS

So far, we discussed the mapping of complex classes of inputs onto spatio-temporal signatures by neural ensembles in form of cliques. These cliques are the code which propagates along the neural pathways, and leads to fusion of the information from multiple sources. In this section, we are concerned with the following issue – what kind of mechanism is responsible for translating complex spatio-temporal cliques to simple discrete outputs, which can be read-out by static, non-adaptive and non-learning actuating system, such as some motor neural circuits.

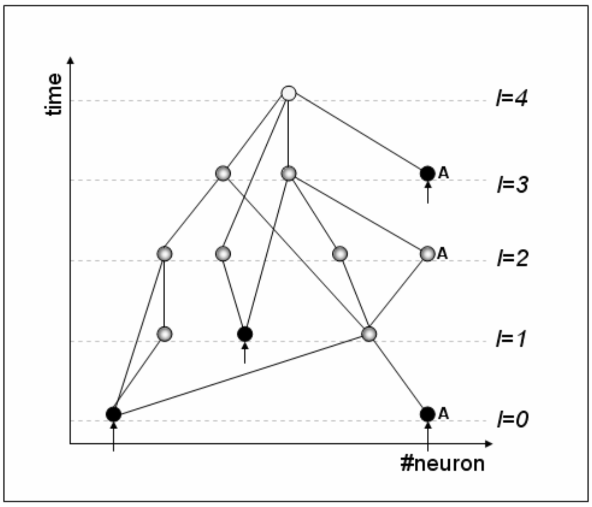

Fig. 2. Example of liquid-currents from a given input to a spike at the output layer. Spikes from input neurons, indicated by black arrows, may appear at several spatio-temporal layers since the injected input has temporal structure. Spikes from one layer can excite several next layers.

Here we implemented another, functionally different, computational framework of cortical network's model, based on concepts adopted from dynamical-systems theory. Cortical networks are dissipative systems. Their fading memory property requires perpetual inputs for the network to maintain activity. Otherwise, the network relaxes to a single resting state. In this setup, cliques are injected to the second neural ensemble (Fig. 1) as initial conditions. Then, a periodic external input (driving stimulus) drives the network dynamics. Three types of behavior are possible: (Type-I) - network dynamics does not converge to a periodic steady-state. (Type-II) - all initial conditions relax to a single steady state, no matter how far they are from each other. (Type III) – the network reaches several different periodic steady-states which are highly dependent on, and sensitive to, the initial conditions (cliques).

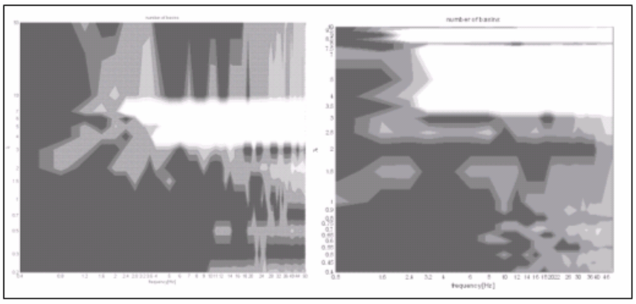

Fig. 3. Possible network dynamics vs. different parameters and its scattering for two different networks. Driving stimulus frequency is shown in x-axis and network connectivity-radius in y-axis. Black areas depict dynamics with a single steady state for all initial conditions; white areas depict dynamics in which no convergence to a periodic steady-state has occurred; gray areas are intermediate states in which the network converged to discrete number steady-states dependent on the initial-conditions.

The systems characterized by Type-II dynamics do not posses the selectivity property, since all initial conditions converge to the same steady state. Thus, systems with this type of dynamics can not be used for computational purposes. Systems with Type-I dynamics, which do not converge to a steady-state, have the selectivity property and therefore may be used for computational tasks. However, since no limit-cycle is obtained, information-extraction regarding the current network's state is very complicated, and requires a complex readout. Systems with Type-III dynamics are driven to limit-cycles, characterized by a certain frequency and phase (shown abstractly in Fig. 5). Since these limit-cycles are selective to classes of inputs injected as the initial conditions and easily detectable, they are the best candidate for definition of system's output in classification computational tasks. Dynamics of such a computational system as a function of specific system's parameters (driving stimulus frequency and the radius of network's connectivity) is shown in Fig. 3. As the figure implies, the values of system's parameters determine the type of network's dynamics. The computationally interesting areas which are represented by the limit-cycle dynamics tend to vary for different network realizations. Nevertheless, some consistency of dynamics in the space of system's parameters can be indicated. It is clear that low connectivity and/or low frequency of the driving stimulus, yields Type-II dynamics, whereas, higher connectivity and high frequency of the driving stimulus lend itself to Type-I dynamics. In the transition between the two types of dynamics, there is a region characterized by Type-III dynamics, implying a certain tendency of the system to exhibit dynamical behavior which is on the verge of chaos in the space of its parameters.

Revealing the computational principle of this model requires thorough understanding of all three types of dynamics. As a consequence of their non-linear nature with numerous degrees of freedom, cortical neural networks have a high tendency for chaotic behavior, as indicated by Type-III behavior. The universality of chaos provides a general theory and offers a well-defined framework for analysis of all types of networks dynamics. However, here we focus on a limit-cycle behavior (Type-III), wherein a periodic external force entrains and synchronizes the system's activity, and makes it computationally effective for classification tasks.

Similarly to other types of dynamical neural networks (e.g. Hopfield type), the behavior of the proposed computational system is also characterized by attractors. The periodic driving-stimulus pushes the network into an attractor, determined by the periodicity of the driving stimulus and the initial condition. This facet of the dynamical behavior of this computational system is quite complex and will, therefore, be dealt with elsewhere in greater detail. However, the role played by the initial condition is fundamental to the understanding of the computational power exhibited by our system, and we would therefore like to discuss briefly some facts related to the role played by the initial condition. An important finding is that the same periodic driving-stimulus can map different initial conditions into different attractors, by entraining the network into sub-, or super-harmonic of the driving frequency (Fig. 4). More interesting, and important computationally, is the structure of the basins of attraction and the mapping of initial conditions into such basins. A basin of attraction is divided into several sub-basins, each being identified by a specific phase of the attractor's limit-cycle, corresponding to a specific class of inputs. In other words, an attractor characteristic of our system exhibits much more complex and richer topology of its dynamics, in that it may be considered as a cluster of attractors characterized by the same limit-cycle with different phases (Fig. 5).

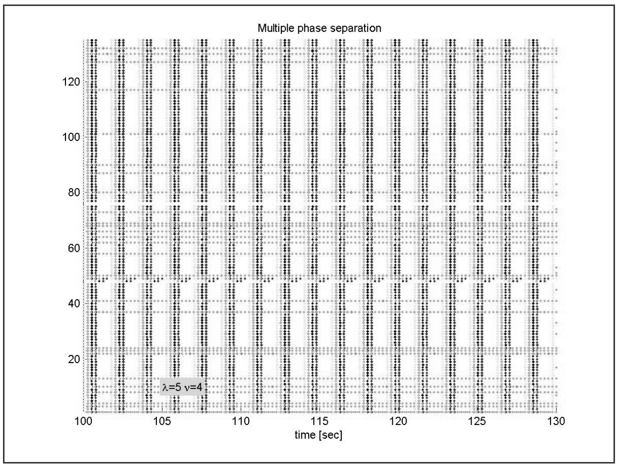

Fig.4. Raster plot of microcircuit response. Shown starting from second 100 of the simulation. The microcircuit is at phase-locked-attractor. Different colors indicate the response of the microcircuit for three different initial conditions.

Thus, as far as neural-computation is concerned, the capacity of the proposed computational system exceeds by far the one realized by Hopfield-type dynamical neural networks [3].

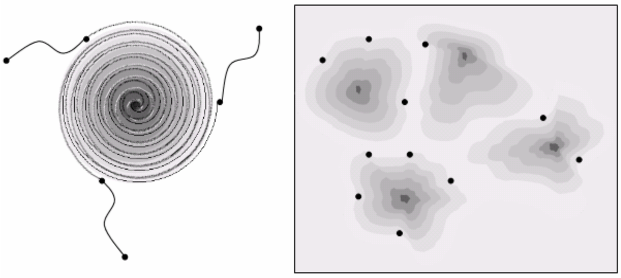

Fig.5. Left figure shows a schematic depiction of an attractor (which represents a frequency), with several possible points of entrance to the attractor (representing the phase). The right figure shows schematically several attractors (i.e. several frequencies), black dots indicate possible entrance points to each of the attractors.

To test the computational power of the proposed system we selected speaker identification task as a benchmark for natural signal classification. The database includes 1-minute phone conversations of ten speakers, 30 conversations for train and 30 for test. The voice signals are pre-processed based on MFCC pre-processing [16] and are injected to the first neural ensemble. Then, an external frequency is applied to the second neural ensemble until it reaches its steady-state. The specific dynamic attractor identified by its phase and frequency is associated with a certain speaker based on the training data-set. Based on the matches between the attractors and input signals in test data-set, the system performed the task with an error rate of 12%.

IV. DISCUSSION

Convergence processes drive neural ensemble into limit-cycle attractors characterized by certain frequencies and phases. A new phenomenon of phased-locked limit-cycle attractor, with multiple entry phases, characteristic of dynamic neural network (and most likely other dynamic systems) was discovered in the course of this study. This new type of dynamic neural behavior can serve as a powerful computational paradigm and mechanism. Further, the generation of large number of attractors, which depend on the initial conditions imposed by the inputs on to the system, enables the mapping of inputs with complex temporal structure onto discrete simple outputs. The proposed computational system for classification of natural signals, exclusively composed of biologically-plausible components, provides also insight into the function of neurobiological processes and structures. We utilize the phase-splitted attractors for recognition and classification of several types of signals (patterns), each one being associated with a specific phase.

V. REFERENCES

[1] Hopfield, J., Neural networks and physical systems with emergent collective computational properties. Proceedings of the National Academy of Sciences of the USA, 79:2554 – 2588, (1982).

[2] Hopfield, J. and Tank, D., Neural computation of decisions in optimization problems. Biological Cybernetics, 52:141 – 152, (1985).

[3] Y. Abu-Mostafa and J. St. Jacques, Information capacity of the Hopfield model.. *IEEE Trans. on Information Theory*, IT-31(4):461-464, July 1985.

[4] Maass W., Natschläger T., Markram H., Computational models for generic cortical microcircuits. In J. Feng, editor, Computational Neuroscience: A Comprehensive Approach, chapter 18, pages 575-605. Chapman & Hall/CRC, Boca Raton, (2004)

[5] DeMarse, T. B., Wagenaar, D. A., Blau, A. W. and Potter, S. M., The Neurally Controlled Animat: Biological Brains Acting With Simulated Bodies Autonomous Robots , Vol. 11, No. 3, 305-310 (2001)

[6] Abeles M., Neural Circuits of the Cerebral Cortex, Cambridge University Press (1991)

[7] Unger H., Zeevi Y. Y., Blind Separation of Spatio-temporal Data Sources. ICA 2004: 962-969 (2004)

[8] Natschläger T., Markram H., Maass W., Computer models and analysis tools for neural microcircuits, In R. Kötter, editor, A Practical Guide to Neuroscience Databases and Associated Tools, chapter 9. Kluver Academic Publishers, Boston, (2002 in press), (http://www.lsm.tugraz.at)

[9] Gerstner W., Kistler W. M., Spiking Neuron Models. Cambridge University Press, (2002)

[10] Markram H., Wang Y., Tsodyks M., Differential signaling via the same axon of neocortical pyramidal neurons. Proc. Natl. Acad. Sci., 95, 5323-5328. (1998)

[11] Tsodyks M, Pawelzik K., Markram H., Neural networks with dynamic synapses, Neural Computation 10, 821-835 (1998)

[12] K. Odinaev, I. Raichelgauz, Y.Y. Zeevi, Cliques in Neural Ensembles as Perception Carriers, to be published in IJCNN2006 proceedings

[13] Aviel Y., Pavlov E., Abeles M., and Horn D. Synfire chain in a balanced network. Neurocomp. 44:285-292 (2002)

[14] Ikegaya Y., Aaron G., Cossart R., Aronov D., Lampl I., Ferster D., Yuste R., Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity. Science. 304 (5670): 559-564, (2004)

[15] Yuste, R., Lanni, F. and Konnerth, A, Imaging Neurons: a Laboratory Manual, Cold Spring Harbor Press, (1999)

[16] M. J. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," Proceedings of the 1980 ICASSP, Denver, CO, pp. 880-883, 1980.