

Emergence of Computational Chaos in Asynchronous Neurocomputing

Sarit Barhen^{†*} Vladimir Protopopescu[†] Jack Wells[†] Neena Imam[†] Jacob Barhen[†]

[†]Center for Engineering Science Advanced Research
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6016, USA

*Emory University
Atlanta, GA 30322, USA
barhenj@ornl.gov

ABSTRACT

One of the most important features of artificial neural networks in emerging, brain-inspired, nanoarchitectural design is their inherent ability to perform massively parallel, nonlinear signal processing. When operating in a system-wide asynchronous regime, such networks may exhibit a phenomenon referred to as “computational chaos”, which impedes the efficient retrieval of information usually stored in the system’s attractors. In this paper, we illustrate the emergence of computational chaos from fixed point and limit cycle attractors for node communication delays in a widely used neural network model. In particular, the complete Lyapunov spectrum associated with the network dynamics is computed, and conditions that prevent the emergence of chaos are derived.

Keywords: asynchronous computing, computational chaos, neural networks, Lyapunov spectrum.

1. INTRODUCTION

Artificial neural networks are massively parallel, adaptive dynamical systems [1]. Their models are inspired by the general features of biological networks, where asynchronous behavior is prevalent. Biological asynchronicity arises from delays in nerve signal propagation, refractory periods, and adaptive thresholding [2], and has long been recognized as one of the most robust mechanisms through which natural systems cope with change, learn, and eventually adapt. Recently, there has been considerable interest in better understanding and exploiting the computational properties of asynchronous neurocomputing models (see [3-6] and references therein) to achieve similar robustness, adaptability and performance in artificial systems, with growing emphasis on emerging nanoscale architectures.

From a computational perspective, two algorithmic schemes have traditionally been considered for updating the state of a network [7-9]. In the *synchronous* regime, all nodes simul-

taneously update their state variables. This implies that each node must receive, within the interval Δ characterizing the resolution of the discrete-event network dynamics, all the necessary information it needs for its computation from all the nodes to which it is connected. In what is traditionally referred to as the *asynchronous* regime, only one node (usually determined randomly) is allowed to update its state on the basis of its inputs, and only after state information has been received from all required nodes. Clearly, this *conventional* type of asynchronicity limits the ability of a network to perform massively parallel, distributed information processing. Hereafter, we will refer to this regime as *sequentially asynchronous*. These two paradigms have provided, to date, the algorithmic foundation of most computational models reported in the open literature [e.g., 3, 5, 6].

However, true computational *asynchronicity* implies an uncoordinated, system-wide activity. We shall refer to it as *concurrent* asynchronicity. There is a strong motivation to develop algorithms that can fully exploit such a behavior. One of the main reasons progress in this direction has been slow is that concurrent asynchronous relaxation algorithms usually give rise to an aperiodic oscillatory behavior. This long known phenomenon was originally referred to as *chaotic relaxation* [10] or *computational chaos*.

In the sequel, we first discuss the limitations of previous approaches and some implications and promises of concurrently asynchronous neurocomputing. Then we characterize the chaotic behavior of our concurrently asynchronous model by estimating the complete Lyapunov spectrum associated with its dynamics. Finally, in order to ensure stable, accurate computations, and thereby enable efficient retrieval of information stored in attractors of the network, we derive conditions that prevent the emergence of computational chaos. We illustrate our results in terms of the well established model attributed to Grossberg and Hopfield [8].

2. LIMITATIONS OF PREVIOUS APPROACHES

The lack of capabilities for concurrently asynchronous computing in present models of artificial neural nets has several important limitations. These include, e.g., problems encountered in: (1) VLSI, optical, or opto-electronic computations; (2) discrete time simulations on large-scale, high performance computers; and (3) emulation of biological systems. For instance, complex global synchronization circuitry is typically needed in VLSI circuits and optoelectronic devices to neutralize the clock skew effects arising from variation in the physical [11] and/or optical [12] path lengths of the actual synaptic interconnections. Not only does this circuitry lack biological basis, but it also limits the overall network performance to operate at the rate of the slowest neuron, and enforces rigid firing sequences that would be difficult to sustain due to signal leakages and component instability. It is well known [13] that in large-scale networks such self-induced pathological activation may destabilize the entire neuromorphic system.

From a purely algorithmic perspective, let us consider a synchronous algorithm implemented on an MIMD parallel computer. The processors associated to sets of neurons must communicate their partial results to each other, at every instance of time specified by the precedence-constrained task graph [14] obtained from the problem decomposition. Such almost sequential algorithms produce overheads in the form of load imbalance due to processor inactivity. The potentially lower processor utilization then enhances resource contention due to communication and coordination requirements, and may lead to severe performance degradation in a real-time network environment.

Finally, from a biological emulation perspective, the current synchronous paradigm implies that neurons are not allowed to evaluate a firing threshold without having to wait to receive excitatory or inhibitory input signals from *all* other neurons to which they are connected. Thus, failure to receive input from some inoperating node in the sequentially asynchronous case could lead to blocking of the entire network.

These limitations make it imperative to develop a fully operational, stable, system-wide asynchronous computation paradigm.

3. BASIC CONCEPTS

To this end, we begin by defining more precisely what we mean by concurrent asynchronous computation. Let N denote the total number of nodes (neurons) in the network. A quantity of interest, $x_n(t)$, is being estimated at each node n , where t indexes a discrete temporal sequence. Let φ be a nonlinear operator from \mathcal{R}^N to \mathcal{R}^N , whose network components are expressed as $\varphi_n(x_1, x_2, \dots, x_N)$. Also, let $\tau_n(t)$ index the availability of the most recently updated state of node n . The successive temporal configurations of the network are $\psi = \{\tau_1(t), \dots, \tau_N(t) \mid t = 1, \dots\}$.

Definition. A concurrently asynchronous system iteration, denoted by the tuple $\{\varphi, \mathbf{x}(0), \xi, \psi\}$, is a sequence of state

iterates $\mathbf{x}(t)$ of vectors in \mathcal{R}^N , obtained by the following recursion starting from a given vector of initial node states $\mathbf{x}(0)$:

$$x_n(t) = \begin{cases} x_n(t-1) & \text{if } n \notin S_t \\ \varphi_n(x_1(\tau_1(t)), \dots, x_N(\tau_N(t))) & \text{if } n \in S_t. \end{cases} \quad (1)$$

Here S_t denotes the set of nodes that carry out an update at the t -th time grid point. The set $\xi = \{S_t \mid t = 1, 2, \dots\}$ is the sequence of nonempty subsets of nodes that performed an update at each t .

Assumptions. Three operational assumptions are made. Two refer to the set ψ , and one constrains the set ξ . Specifically, we require that:

- ▶ Each consecutive update uses only state information previously available at the node under consideration, i.e., $\tau_n(t) \leq t - 1$.
- ▶ Conservation of temporal logic: evermore recent state information must be used in evolving each node.
- ▶ Node n is not starved in ξ , i.e., there exists a finite natural number $s \in \mathcal{N}$, such that each node updates its estimate at least once in every s successive time intervals.

The above definition provides a formal framework for algorithms that implement concurrent, asynchronous network dynamics. Such a dynamics is capable of updating the nodes in an uncoordinated manner, where the neurons are seen as a collection of functionally cooperating processes, with no explicit dependencies to enforce waiting at synchronization points for the purpose of swapping partially computed results. Such a framework encompasses both the concept of random node delays introduced by Baudet [15], and a newly developed alternative paradigm [16], which explicitly models random delays in the interconnection (communication) network.

4. LYAPUNOV SPECTRUM

We will illustrate our discussion in terms of the temporal evolution of a fully connected Grossberg – Hopfield (GH) network [8]. Such a system is modeled by the following system of coupled, nonlinear differential equations:

$$\frac{dx_n}{dt} + a_n x_n = \sum_l T_{nl} g_l(\gamma_l x_l) + I_n. \quad (2)$$

Here x_n represents the internal state of the n th neuron. The strength of the synaptic coupling from neuron l to neuron n is denoted by T_{nl} , and the external bias is denoted by I_n . The sigmoidal function g_n modulates the neural response, γ_n denotes the gain of the transfer function of the n th neuron, and a_n represents the inverse of a characteristic time constant or a decay scaling term.

In order to quantitatively characterize the behavior of the network dynamics, consider a point \mathbf{x}_e on the trajectory of

the autonomous system described by the vector field \mathbf{f} corresponding to the RHS of Eq (2). The local behavior of the flow near \mathbf{x}_e is determined by linearizing the vector field at \mathbf{x}_e , i.e.,

$$\delta\dot{\mathbf{x}} = \mathcal{D}\mathbf{f}(\mathbf{x}_e)\delta\mathbf{x} \quad \delta\mathbf{x}(0) = \delta\mathbf{x}_0 \quad (3)$$

The linear vector field governs the time evolution of a perturbation $\delta\mathbf{x}_0$ in the neighborhood of \mathbf{x}_e . Let the eigenvalues and eigenvectors of $\mathcal{D}\mathbf{f}$ at \mathbf{x}_e be $\lambda_i \in \mathbb{C}$, and $\xi_i \in \mathbb{C}^N$, for $i = 1, 2, \dots, N$. We know from linear systems theory that (assuming that the eigenvalues are distinct) the trajectory with initial conditions $\mathbf{x}_e + \delta\mathbf{x}_0$ evolves as

$$\begin{aligned} \mathbf{x}(t, \mathbf{x}_e + \delta\mathbf{x}_0) &= \mathbf{x}_e + e^{\mathcal{D}\mathbf{f}(\mathbf{x}_e)t} \delta\mathbf{x}_0 \\ &= \mathbf{x}_e + c_1 e^{\lambda_1 t} \xi_1 + \dots + c_N e^{\lambda_N t} \xi_N \end{aligned} \quad (4)$$

where $c_i \in \mathbb{C}$ are constants determined from the initial conditions. If λ_i is real, then ξ_i and c_i are also real. It is clear that λ_i corresponds to the rate of contraction ($\lambda_i < 0$) or expansion ($\lambda_i > 0$) near \mathbf{x}_e in the direction ξ_i . Since the matrix $\mathcal{D}\mathbf{f}$ is real, if the eigenvalues λ_i are complex, they occur in complex conjugate pairs, and the real part of λ_i gives the rate of expansion or contraction.

Thus, Lyapunov exponents characterize the behavior of a dynamical system which may include equilibrium points, periodic solutions, as well as quasi-periodic and chaotic regimes. To find all N Lyapunov exponents, a set of N linearly independent perturbation vectors $\delta\mathbf{x}^{(k)}$ is repeatedly integrated and orthonormalized [17]. Here, a modified Gram-Schmidt (GS) procedure [18] is used for improved numerical stability. After each integration stage l of duration T , the GS generates two sets of vectors, $\mathbf{v}^{(k)}(l)$ and $\mathbf{r}^{(k)}(l)$, such that $\mathbf{v}^{(1)}(l) = \delta\mathbf{x}^{(1)}$ and

$$\begin{aligned} \mathbf{r}^{(k)}(l) &= \delta\mathbf{x}^{(k)} - \sum_{i=1}^{k-1} \langle \delta\mathbf{x}^{(k)}, \mathbf{v}^{(i)}(l) \rangle \mathbf{v}^{(i)}(l) \\ \mathbf{v}^{(k)}(l) &= \mathbf{r}^{(k)}(l) / \|\mathbf{r}^{(k)}(l)\|, \quad k = 2, \dots, N. \end{aligned} \quad (5)$$

Note that the set of vectors $\{\mathbf{v}^{(k)}\}$ spans the same subspace as $\{\delta\mathbf{x}^{(k)}\}$ for $k = 1 \dots N$. Then, at the L -th stage (for L sufficiently large), the k -th Lyapunov exponent is computed as

$$\lambda_k \approx \sum_{l=1}^{L-1} \text{Log}_e(\|\mathbf{r}^{(k)}(l)\|) / LT. \quad (6)$$

We now consider asynchronous interconnections, i.e., networks with *random delays*. The variational equation corresponding to Eq (2) takes the form [16]:

$$\begin{aligned} \frac{d}{dt} U_{nm} &= -a_n U_{nm} + T_{nn} (1 - g^2(\gamma_n x_n)) \gamma_n U_{nm} \\ &+ \sum_{l \neq n}^{l=N} T_{nl} (1 - g^2(\gamma_l x_l^{\eta(n,m,v,\theta)})) \gamma_l U_{lm}^{\eta(n,m,v,\theta)} \end{aligned} \quad (7)$$

where

$$U_{nm}(t) = \delta x_n(t) / \delta x_m(0), \quad \mathbf{U}(0) = \mathbf{I}. \quad (8)$$

Note that the vector $\delta\mathbf{x}^{(k)}$ in (5) refers to the k -th column of \mathbf{U} . In deriving Eq (7), a distribution of one neuron per processor was assumed. Here θ denotes a random delay

experienced by the data packet x_m^η sent at time η on a path from m to n and used by n at v . We now apply these concepts to highlight the emergence of computational chaos.

5. EMERGENCE OF COMPUTATIONAL CHAOS

In this study, we have considered a low-dimensional model made up of 4 fully interconnected neurons. We show that, even in such a small network, asynchronous distributed computing gives rise to a variety of complex behaviors.

In a previous study, we examined emergent behaviors under random *node* delays [19]. Here, the focus is on random *network* delays. All simulations were performed with the recently developed *A♦NET* code [16]. This code is entirely written in *Intel Visual FORTRAN 95*. The figures correspond to a direct screen dump of a simulation. The *A♦NET* visualization software creates, in real time, these graphics displays.

Each figure comprises three regions. The upper region displays the evolution of the complete Lyapunov Spectrum. The color-coded magnitude of each exponent is plotted versus consecutive time intervals.

The lower left region displays the signal output of each neuron versus integration time. This time-series plot is a basic observational tool for dynamical systems. As integration time progresses, one is able to monitor the output of each neuron, to ascertain whether it converges to a single (fixed) point, follows a cyclic path, or wanders chaotically.

The lower right region displays a Poincaré plot. This is a phase-space diagram where the signal output of one neuron is plotted versus the signal output of another neuron over time. From a dynamical system perspective, the Poincaré plot depicts the trajectories (orbits) of two particular components (neurons) of the network. Trajectories that enter the domain of a *point attractor* will approach and remain at that point. Then a stable equilibrium solution or fixed point has been reached. Trajectories that enter the domain of a *limit cycle attractor* will approach and generate a periodic solution. Finally, trajectories that enter the domain of a *strange attractor*, will exhibit divergence from one another, and are usually in a state of chaos.

Our *first case* addresses a situation, whereby in the *synchronous* regime the dynamics converges to a fixed point attractor. In the *asynchronous* regime, as delays are introduced, we observe first a transition to a quasi limit cycle. Then, as delays become larger, a chaotic regime emerges. The following parameters are used: $\forall n, a_n = \gamma_n = 1, I_n = 0$. The elements of the synaptic matrix T_{nl} are shown below:

0.850D+00	-2.000D+00	1.100D -00	0.500D+00
1.800D+00	1.150D+00	0.600D+00	0.300D+00
1.100D+00	2.500D+00	2.500D+00	0.500D -01
0.100D+00	-0.400D+00	-1.441D -00	1.450D+00

Table 1. Base model data for synaptic interconnection matrix

Synchronous dynamics is an idealized situation in which no network delays are assumed to occur in information propagation between neurons. This is a convenient simulation assumption, which is essentially equivalent to the requirement of synchronization blocking (and possible processor idling) in an actual distributed system. Of course, in a real-life system, synchronization requirements reduce the overall efficiency of the information processing throughput.

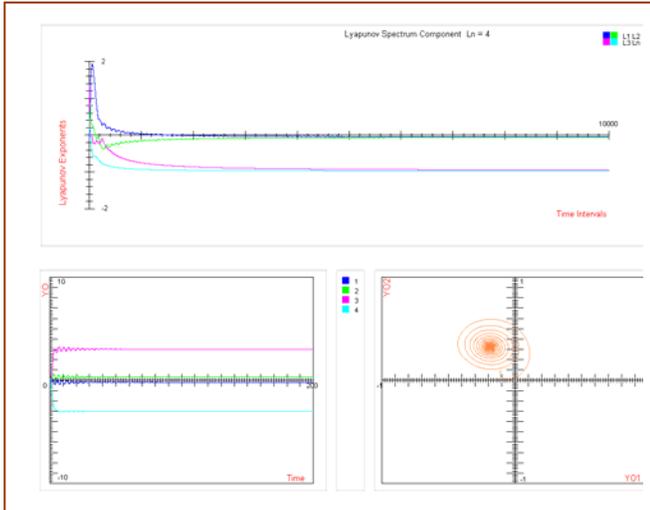


Figure 1. Synchronous dynamics for fixed point attractor

As can be observed in Fig. 1, the network dynamics converges to a *fixed point* attractor. All Lyapunov exponents are accordingly negative. The actual spectrum calculated by the $A\heartsuit NET$ code is:

$$(\lambda_1 = -0.047, \lambda_2 = -0.059, \lambda_3 = -0.948, \lambda_4 = -0.975).$$

For random communication delays of up to 5Δ , the Poincaré plot in Fig. 2 appears to indicate a quasi limit cycle. This is supported by the output signal from each neuron, which exhibits almost periodic motion.

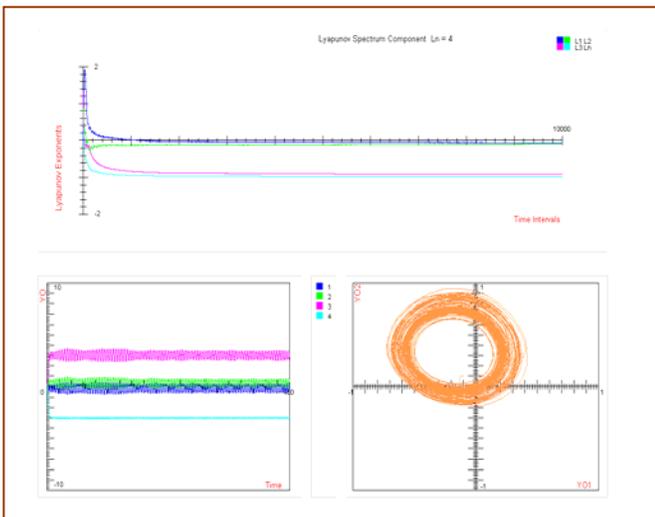


Figure 2. Asynchronous dynamics: small delays

As the bound on random communication delays increases, aperiodic oscillations arise. For a maximum allowable communication delay of 200Δ (where Δ is the integration step size), the emergence of computational chaos (Fig. 3) is confirmed by the existence of positive components in the Lyapunov spectrum. The exponents calculated by the $A\heartsuit NET$ code are:

$$(\lambda_1 = +0.162, \lambda_2 = +0.026, \lambda_3 = -0.303, \lambda_4 = -0.608).$$

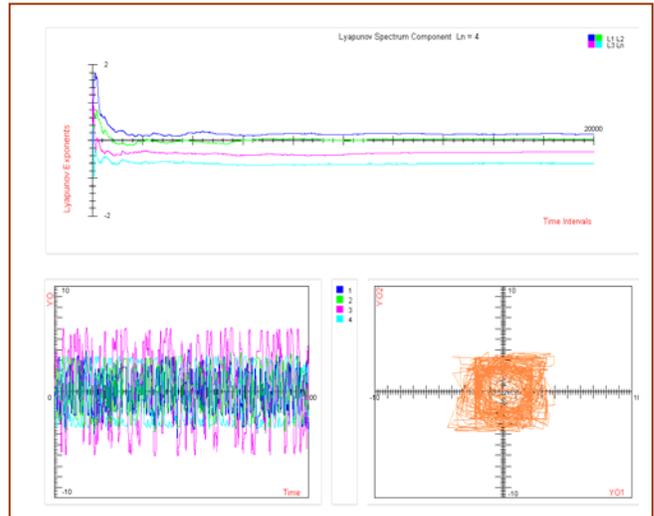


Figure 3. Asynchronous dynamics: evolution from fixed point attractor under large communication delays

Our *second case* addresses a situation whereby in the synchronous regime the dynamics converges to a *limit cycle* (see Fig. 4). In the asynchronous regime, as delays are introduced, we observe (Fig. 5) transition to chaos. The limit cycle was obtained by changing three synaptic parameters in Table 1:

$$T_{1,2} = -6.00; T_{1,3} = -0.55; T_{1,4} = +2.08.$$

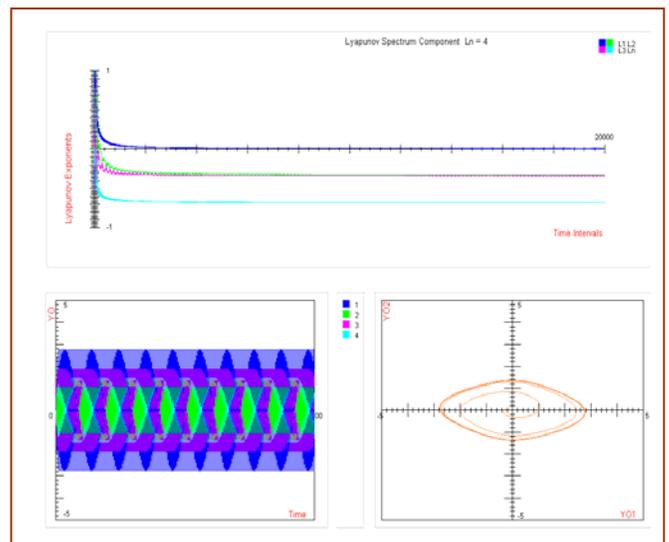


Figure 4. Synchronous dynamics for limit cycle attractor.

The lower left region in Fig 4 clearly indicates periodic behavior. The leading Lyapunov exponent is zero. In presence of network communication delays, Fig 5 illustrates the emergence of computational chaos. The Lyapunov spectrum exhibits two positive exponents:

$$(\lambda_1 = +0.216, \lambda_2 = +0.008, \lambda_3 = -0.139, \lambda_4 = -0.607).$$

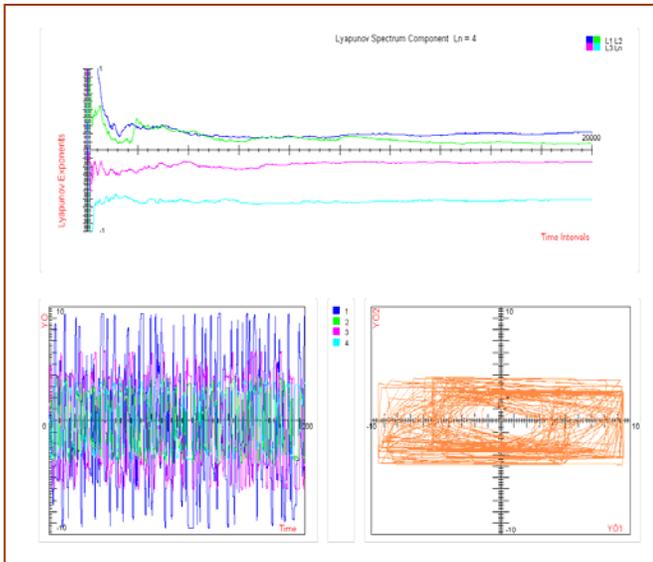


Figure 5. Asynchronous dynamics: evolution from limit cycle attractor under large communication delays

Remark: We note that the system considered in Section 4 is more general than a typical nonlinear delayed ODE system, since here the delays are not fixed, but *random*. Therefore, strictly speaking, the dynamics is in a perpetual transient state, without ever reaching an asymptotic behavior. However, if the average delay is sufficiently large with respect to the intrinsic time-scale of the dynamics (as measured by the inverse of a_n), then a certain average stabilization can be reached. This is indeed the case, as illustrated in Figures 3 and 5, where the Lyapunov exponents do not stabilize to fixed values, but instead slightly fluctuate, while maintaining a clearly discernible average.

6. TAMING COMPUTATIONAL CHAOS

Since asynchronous neurodynamics may become chaotic, additional tools are needed to guarantee that correct results are ultimately obtained. The tools we are proposing are based on the concept of contraction [20]. Contraction plays a fundamental role in the iterative solution of nonlinear equations. It is most useful to express contraction in terms of vector norms, defined as [20] $\|\mathbf{x}\| = (|x_1|, \dots, |x_N|)$. This norm induces a partial ordering on \mathcal{R}^N .

Definition. An operator $\varphi: \mathcal{D} \subset \mathcal{R}^N \rightarrow \mathcal{R}^N$ is called a Φ -contraction on a set $\mathcal{D}_0 \subset \mathcal{D}$, if there exists a linear operator $\Phi \in L(\mathcal{R}^N)$ with the following properties:

- $\|\varphi(\mathbf{x}) - \varphi(\mathbf{y})\| \leq \Phi \|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{D} \quad (9-a)$
- $\Phi \geq 0 \quad (9-b)$
- $\rho(\Phi) < 1. \quad (9-c)$

The first property implies Lipschitz continuity. Indeed, Φ is often referred to as the Lipschitz *matrix* of φ . The latter requirements, namely non-negativity and spectral radius of Φ , generalize the typical specification of the contractive constant used in conjunction with the usual norm on \mathcal{R}^N .

Baudet's Theorem. If $\varphi: \mathcal{R}^N \rightarrow \mathcal{R}^N$ is a Φ -contraction on the closed subset $\mathcal{D} \subset \mathcal{R}^n$, and if $\varphi(\mathcal{D}) \subset \mathcal{D}$, then any concurrent asynchronous iteration corresponding to φ and starting with a vector $\mathbf{x}(0) \in \mathcal{D}$, converges to a unique fixed point of φ on \mathcal{D} [15].

These concepts can be applied to study the convergence of concurrently asynchronous time-evolving processes in general, and neural networks in particular. We have previously presented results for *node delays* [19]. A similar derivation applies to communication delays as well, since both fall under the general definition (1), even though different variational equations are needed for estimating the Lyapunov spectrum [16].

Let $\varphi_n(\mathbf{x})$ denote the n th component of the discretized GH operator obtained from Eq. (2). We have

$$\varphi_n(\mathbf{x}) = x_n + \Delta(-a_n x_n + \sum_m T_{nm} g_m(\gamma_m x_m) + I_n). \quad (10)$$

Let us seek convergence to fixed-point attractors of (2). For any two phase-space points \mathbf{x} and \mathbf{y} in a domain of attraction, we can write

$$\begin{aligned} \varphi_n(\mathbf{x}) - \varphi_n(\mathbf{y}) &= (1 - \Delta a_n)(x_n - y_n) \\ &\quad + \Delta \sum_m T_{nm} [g_m(\gamma_m x_m) - g_m(\gamma_m y_m)]. \end{aligned} \quad (11)$$

After taking the vector norm and applying the Mean Value Theorem, assuming that, for each neuron the transfer function $g_n: \mathcal{R} \rightarrow [-1, +1]$ is of class \mathcal{C}^1 , and that $|g'_n| \leq 1$ (which is indeed the case for the sigmoid response functions usually considered), we obtain

$$\begin{aligned} \|\varphi_n(\mathbf{x}) - \varphi_n(\mathbf{y})\| &\leq |1 - \Delta a_n| \cdot \|x_n - y_n\| \\ &\quad + \Delta \sum_m |T_{nm}| \cdot |\gamma_m| \cdot \|x_m - y_m\|. \end{aligned} \quad (12)$$

Let us now define a matrix Φ in the following manner:

$$\Phi_{nm} = |1 - \Delta a_n| \delta_{nm} + \Delta |T_{nm}| \cdot |\gamma_m|. \quad (13)$$

We see that, by definition, Φ is non-negative. Considering Eqs. (12–13), we observe that

$$\|\varphi(\mathbf{x}) - \varphi(\mathbf{y})\| \leq \Phi \cdot \|\mathbf{x} - \mathbf{y}\|. \quad (14)$$

Thus, the GH operator φ is Lipschitzian with Lipschitz matrix Φ . For φ to converge to a fixed point in an appropriate basin of attraction, the spectral radius of Φ must be less than one. Our basic idea, here, is to use this requirement to establish constraints on the model parameters. In order to produce an operational statement, we

use an inequality immediately deduced from the definition of ρ . For any vector \mathbf{y} with positive components, we can write:

$$\min_{1 \leq n \leq N} \frac{\sum_{m=1}^{m=N} \Phi_{nm} y_m}{y_n} < \rho(\Phi) < \max_{1 \leq n \leq N} \frac{\sum_{m=1}^{m=N} \Phi_{nm} y_m}{y_n}. \quad (15)$$

In particular, we can choose all vector components y_n to be equal. The contraction requirement, $\rho(\Phi) \leq 1$, then translates into

$$\max_{1 \leq n \leq N} \sum_{m=1}^{m=N} \Phi_{nm} < 1. \quad (16)$$

The above inequality induces constrained interrelationships between the values of the model parameters a_n , Δ , γ_n , and T_{nm} (with $n, m = 1 \dots N$). In particular,

$$\max_n \left\{ |1 - \Delta a_n| + \Delta \sum_m (|T_{nm}| \cdot |\gamma_m|) \right\} < 1 \quad (17)$$

We note that conditions (17), which ensure the stabilization of the chaotic behavior are *sufficient* conditions obtained under certain simplifying assumptions. At this point, they offer a proof of principle rather than an easily and directly implementable criterion. Our future effort will focus on eliminating certain unnecessary assumptions and making these conditions more general. It is important, at this stage, to also emphasize that the reader should not confuse arbitrarily large (but finite) delays with large values of Δ . In the latter case, Eq. (17) would imply arbitrarily small values for γ_m , the system would become almost linear, and chaos would be excluded by definition. In our paradigm, delays are handled in the framework of a set-theoretic formalism (they appear implicitly in the set ξ). Thus, we allow for arbitrarily large delays, without affecting the structure of the dynamical system.

7. CONCLUSIONS

The biggest promise of artificial neural networks as computational tools lies in their potential capability to emulate information processing of biological systems. Their paradigmatic advantages (i.e., their inherent ability to perform distributed, massively parallel, asynchronous information processing) can not be fully realized under existing neurodynamics relaxation schemes. In particular, concurrent asynchronicity has not been used to date, since it often engenders computational chaos. In this paper, we have illustrated the emergence of computational chaos and derived conditions that ensure that it does not occur. The new methodology was illustrated on a recurrent network modeled via the Grossberg–Hopfield formalism.

ACKNOWLEDGEMENTS. This research was performed at the Center for Engineering Science Advanced Research (CESAR) in the Computer Science and Mathematics Division, Oak Ridge National Laboratory. Funding was provided by the Division of Materials Sciences and

Engineering, DOE Office of Basic Energy Sciences, and by the Missile Defense Agency under contract DE-AC05-00OR22725 with UT - Battelle, LLC.

REFERENCES

- Hassoun, M., (1995). *Fundamentals of Artificial Neural Networks*, MIT Press.
- Marcus, C. and M. Westervelt, (1989). "Stability of analog neural networks with delay", *Phys. Rev. A* **39**, 347-359.
- Yamanaka, K., M. Agu, and T. Miyajima, (1997). "A continuous-time asynchronous Boltzmann machine", *Neural Networks*, **10**, 1103-1107.
- VanRullen, R. and S. Thorpe, (1999). "Spatial attention in asynchronous neural networks", *Neurocomputing*, **26-27**, 911-918.
- Benson, M. and J. Hu, (2000). "Asynchronous self-organizing maps", *IEEE Trans. Neural Networks*, **11**(6), 1315-1322.
- Matsuoka, J., Y. Sekine, K. Saeki, and K. Aihara, (2002). "Analog hardware implementation of a mathematical model of an asynchronous chaotic neuron", *IEICE Trans Fundamentals*, **E 85-A**, 389-394.
- Cheung, K., L. Atlas, and R. Marks, (1987). "Synchronous versus asynchronous behavior of Hopfield's CAM", *Applied Optics*, **26**, 4808-4813.
- Hopfield, J., (1984). "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Nat. Acad. Sci.*, **91**, 3088-3092.
- Toomarian, N. and J. Barhen, (1992). "Learning a trajectory using adjoint functions and teacher forcing", *Neural Networks*, **5**, 473-484. *ibid*, "Fast temporal neural learning using teacher forcing", *U.S. Patent No. 5,428,710* (June 1995)
- Chazan, D. and W. Miranker, (1969). "Chaotic relaxations", *Linear Algebra & Applic.*, **2**, 199-222.
- Mead, C., (1989). *Analog VLSI and Neural Systems*, Addison-Wesley.
- Shamir, J., (1987). "Fundamental speed limitations on parallel processing", *Applied Optics*, **26**, 1567-1568.
- Macukow, B. and H. Arsenault, (1987). "Modification of the threshold condition for a content addressable memory based on the Hopfield model", *Applied Optics*, **26**, 34-36.
- Barhen, J. and E. C. Halbert, (1986). "ROSES: An efficient scheduler for precedence-constrained tasks on a hypercube multiprocessor", in *Hypercube Multiprocessors '86*, **3**, 123-147, M.T. Heath, ed., SIAM Publishers.
- Baudet, G. M., (1978). "Asynchronous iterative methods for multiprocessors", *Jour. ACM*, **25**, 226-244.
- Barhen, S., (2004). "Asynchronous computation in artificial neural networks", ORNL/TM-2004/106.
- Wolf, A., J. Swift, H. Swinney, and J. Vastano, (1985). "Determining Lyapunov exponents from a time series", *Physica* **16D**, 285-317.
- Golub, G. and C. Van Loan, (1996). *Matrix Computations*, Johns Hopkins University Press.
- Barhen, J. V. Protopopescu, S. Barhen, and J. Wells, "Asynchronous Computation and Emergence of Computational Chaos", *Proceedings of ADHS'03s*, pp. 123-128, IFAC Press (2003).
- Ortega, J. M. and W. C. Rheinboldt, (1970). *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press.