Computer Science Journal of Moldova, vol.**9**, no.2(26), 2001

# Novel Principles and Methods for Computing with Attractors

Horia-Nicolai Teodorescu, Daniel Mlynek, Abraham Kandel

**Abstract**

We briefly analyze several issues related to the "computing with attractors" domain. We present a point of view on the topic and several new concepts, methods, and techniques for computing with attractors. We discuss applications where this method may prove useful. We answer several questions related to the usefulness of this computing paradigm.

## 1 Introduction

There are currently several research directions aiming to generalize and renew the computation principles and the computing machines. Among these directions, quantum computing [1-3], computing with attractors [4] and cellular automaton computing [5, 6] promise to revolutionize both the computation principles and the hardware. The first two methods go far beyond the typical trends in innovating computing hardware, as based on classic neural networks, fuzzy logic and GA (see [7] for the state of the art in these implementations that are more classical.)

Computing with attractors is a generalization in several directions of the digital computers and is based on neuro-biological findings [8-13]. Computing in the usual sense means to induce (follow) a successive string of transitions between states represented by strings of digits (binary numbers.) The basic element that performs the computation in a classic computer is a transistor - or a CMOS transistor pair, which represents the output stage of a logic gate, or of a flip-flop. The transistor

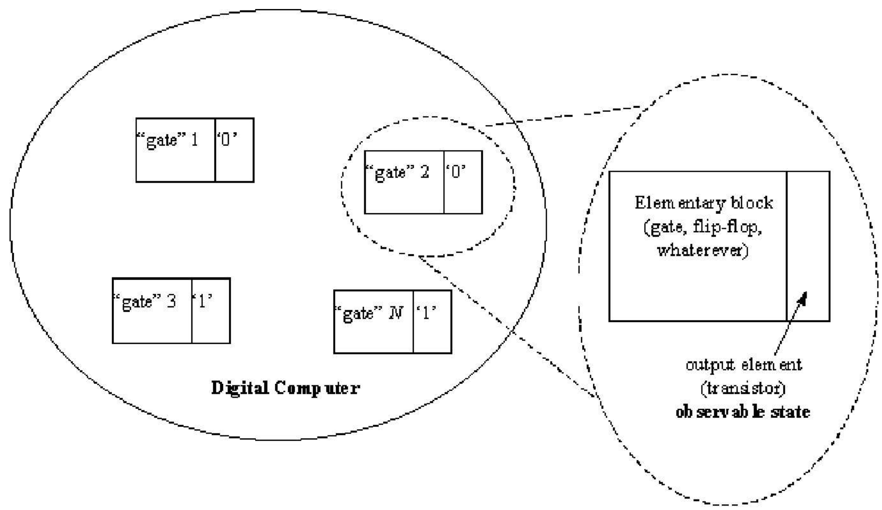


Figure 1.Conceptual scheme of a computer: systemic point of view emphasizing the state concept.

operates in two stable states, i.e. its operation, or dynamics is characterized by only two stable states, denoted by "0" and "1" respectively. It is immaterial, for the computation purpose, any transitory state of the transistor and any dynamics of it, between two clock-pulses. Actually, any dynamics is seen as "transition noise" and rejected. The state of the computer is represented, at some given time, by the vector of the states of its transistors (see Fig. 1). Computation is a string of transitions between state vectors in the state space of the digital computer.

A first generalization of computer principles comes with the generalization of its elementary blocks. Again, without loss of generality, we shall consider that some output devices (more precisely, some observable state parameters) of the elementary blocks are essential, and we disregard any other physical element in the system named "computer". We can allow the states of the elements span a larger set, for example a set of three or more states, allowing multi-valued logic to be implemented on that computer. This generalization has seen some success in the 1960s, but it has been largely abandoned due to its limited capabilities.

The second generalization, which is by far more powerful, is the quantum computing (QC). QC replaces the building elements by quantum elements, characterized by quantum states, named "quantum bits", or qubits. Apart from being extremely small, quantum elements bring some essential advances, namely the statistical (quanta) states and their specific properties, which allow us a new type of computation due to the so-called entanglement process [1-3]. This actually represents the generalization in another sense of the operation of the elementary computing elements: they are no more supposed to be independent, except the input-output relationship characterizing the elements, but they are considered entangled. Notice that either quantum or classic, these states are static: they are supposed in some respect constant (although the spin is seen generally as a rotation, in classic terms.)

Another radical way to generalize states is to allow them to be dynamic states, instead of static states [4]. In this case, we use elements that exhibit dynamics, and for whom the dynamics is relevant, not the instantaneous state. In this case, the instantaneous state, and whether the state is periodically or chaotically varying, is immaterial, as far as we can characterize and recognize various dynamics. In this case, the specific dynamics are characterizing the "state" of the elementary blocks. Basically, a set of oscillating elements can be used to build the computer, as far as these elements exhibit a well-defined set of oscillations "oscillatory states" and the oscillatory state can be controlled in a reliable way by the output states of other elementary blocks. If these conditions are fulfilled, each dynamic state can be labeled with symbols from a suitable alphabet, e.g. $0, 1, 2, \ldots, n$ and a multi-valued computation can be performed. Such computation has been dubbed "computing with attractors". We shall assume in this paper that we deal with a network of dynamic subsystems, with no specific restriction on the systems or on their dynamics. We may name such a system a chaotic network, or a "chaotic computer" (briefly, "cha-puter"), and we will denote it by the acronym ChaN (Chaotic Network).

## 2 Why "Computing With Attractors"?

From the discussion above, there is no apparent reason to implement and use the "computing with attractors" strategy. It looks that we need more transistors to implement an elementary block in a "cha-puter" than a classic logic gate. Moreover, it is not apparent how to determine the dynamical states of these elementary blocks. Most important, there is no apparent benefit in using "computing with attractors", because it seems that the same results can be obtained using other types of multi-valued computers, even binary, classic computers.

We first notice that *Nature* has not invented a binary computer, but did invented "computing with attractors" long before humans. Indeed, there is evidence that the nervous system, natural neural networks (NNNs), moreover most tissues are "computing with attractors" as a manner of "life performing". It is not clear why this way has been adopted in the natural systems, except maybe that static ways can not be supported - life equates dynamics.

### 2.1 What is chaos

Chaotic behavior occurs in systems described by nonlinear equations, e.g. NDEs or discrete nonlinear equations (maps). More precisely, there must be some non-monotonicity in the equations for a chaotic behavior can be produced. The existence of the chaotic behavior is tested by determining the Lyapunov coefficients (the way the space element is expanded and compressed during time by the system.) A stable system only compresses space; asymptotically unstable systems only expand space; periodically oscillating systems keep the space element unchanged. Only chaotic systems expand the space along some directions, while compressing it along other directions, the overall volume remaining essentially unchanged. Therefore, to test for chaos, we should find at least one Lyapunov coefficient larger than zero, $\lambda > 0$. Consequently, two trajectories of the system evolution starting from nearby points will (exponentially) diverge. This makes chaotic dynamics "unpredictable," although it remains deterministic. The system follows a "strange" path, neither oscillating, nor stabilizing, nor infinitely expanding. While bounded, the trajectory remains an *open* curve – a *strange attractor*. Other ways exist to recognize and characterize chaos, e.g. $1/f$ spectra, Poincaré sections, fractal dimensions, bifurcation diagrams, phase plots etc.

Basically, a set of self-oscillating elements (possibly elements that are able to generate a chaotic behavior by themselves) can be used to build the machine. These elements should exhibit a well-defined set of "oscillatory states" that can be controlled in a reliable way by the output states of the other elementary blocks, or by some other parameter. The *specific dynamics* are characterizing the "state" of the elementary blocks. Again, if these conditions are fulfilled, each dynamic state can be labeled with symbols from a suitable alphabet. Such a computation has been dubbed "computing with attractors." We need to characterize and recognize various *dynamics*.

## 2.2 First reason of using Cha-Ns: Modeling neuronal structures

Putting apart the question why nature uses "computing with attractors" – a question we probably are unable to satisfactory answer – we can find the first field where technical "computing with attractors" may have tremendous advantages: modeling living tissues, chiefly the nervous tissue, and the brain. Such models could be expanded and used in creating artificial intelligence at the hardware level by mimicking the nervous tissue and the brain. They might be useful in experimenting with sound and fast operating models of the nervous tissue and the brain, including performing psychological experiments. Subsequently, we detail this issue.

Consider we need to perform a realistic simulation of how a nervous tissue autonomously behaves under specific circumstances for, let us say, a period of time of one month (about $2'592'000$ sec.) Assume the tissue represents 0.01% of the brain. This amounts to about $10^7$

neurons, for a brain of $10^{11}$ neurons[1] The operation of each neuron is described in the simplest form by a set of at least three differential equations. This amounts to a set of $3 \times 10^7$ equations, that have to be solved discretely for a sampling rate of about 100 ms. The total number of equations times the total number of iterations is in the range of $2.610^6$ sec $\times 0.1/\text{sec}$ $\times 310^7 \approx 10^{13}$, that is about $10^{13} \times 10^2 \times 10^{-9} = 10^8$ seconds $> 300$ hours.

Compare the same problems in a system that mimics the brain with the same level of complexity as the computation discussed above. Assume that for every neuron we use an elementary hardware cell. For building an elementary chaotic system implementing a set of three differential equations, we need today about 10 to 20 transistors (including resistors and simulated capacitors). To model $10^7$ neurons, we need $10^9$ transistors, which is almost feasible today. The first apparent advantage of this model is that, in contrast to natural neurons – that operate at about 1 kHz, the modeled neurons can operate at much higher frequencies, in the range of 1 MHz or higher. This allows us modeling the *whole* brain operation 1 to 10 thousands faster that the brain operates. This shows us that, *with currently available technology*, we are able to perform a modeling of 1% of the brain in 0.1 to 1 ms for every second modeled. In other words, we can create a partial brain model that "lives" our "psychological life" in a few tens to thousands of hours. Such a psychological modeling could satisfy any psychologist or neuro-biologist today.

Notice that the use of analog circuitry instead of digital circuitry has a significant advantage in biological modeling: while the natural neurons are continuous-time and analog machines, digital computers are discrete-time and discrete-state machines. Therefore, the latter can only provide approximated results, whose behavior can significantly

[1] According to [14]: "The total number of neurons in the central nervous system ranges from under 300 for small free-living metazoans such as rotifers and nematodes" ..., "to well over 200 billion for whales and elephants. Estimates for the human brain range between 10 billion and 1 trillion."... "In a ... analysis of human cortex ..., Pakkenberg and Gundersen (1997) have shown that the number of neocortical neurons ranges from 15 to 31 billion and averages about 21 billion. ... Total neuron number in humans therefore probably averages 95–100 billion."

depart from the continuous solutions.

Moreover, we are now able to produce models of small nervous systems, such as the nervous systems of nematodes (300 neurons) in almost full detail, moreover to create populations of such models, and to analyze the interaction between their "brains". We are also able to create models of human ganglia (clusters of 20-30 neurons). Moreover, we are able to build models for complex nervous systems, such as that of "the complex nervous system of grasshoppers, which contain up to 200,000 neurons," that "may be made up almost entirely of identified neurons and identified neuron clusters" [14]. This will take a chip of around 5 million transistors to build a "fast behaving grasshopper" and to analyze in a few minutes its behavior along its entire life. I believe this is the experiment a biologist would like to conduct just now, instead of using a few electrodes inside an actual grasshopper.

The apparent hardware level difficulty of building all connections representing the synapses – about 10 to 1000 per neuron – can be overcome in present integrated circuits with multi-layer interconnections.

From the point of view of the (algorithmic) complexity theory, modeling a natural neuronal system is a polynomial problem. Indeed, for $n$ neurons with $q$ synapses each, there are $n^q$ entries in the system, to be modeled for $p$ time steps. Therefore, the complexity is of the order $pn^q$ and should be tractable by current machines in reasonable time. However, this time might be too large for practical problems, like simulating biological and psychological experiments. The difficulty arises from the large numbers $n$, $p$, and $q$ and from the fact that we try to model a highly parallel system by a sequential machine (classic computers). A specific architecture should be applied to this problem and such architectures and systems are at hand today. Using them, the complexity is reduced to $p$, or, for one instant, to constant complexity. Notice that using typical CAs or parallel digital machines does not alleviate the problem, because we apply the wrong machines to the problem.

### 2.3 Second reason: Making use of intrinsic properties

"Computing with attractors" may prove to be more than classic computing, in a way similar to quantum computing. The main intrinsic property that may benefit this type of computation relays on the hidden relationship that exists between the state of a group of subsystems and the inputs to another set of network elements. This allows building hidden dependencies that may act as *data fusing* operators. Also, ChaNs may act as associative memories and feature extraction systems, nonlinear transformers of an input space into a different output space etc. Most of these properties and possible applications are not specific to ChaNs, and can be found in CAs and usual recurrent neural networks, however the potential application field may differ. It is known that "the power of quantum computers comes from their ability to follow a coherent superposition of the computation paths" [3]. At another level, we may expect specific correlations between various dynamic regimes, and specific paths in the pattern space, for ChaNs.

## 3 Application: Intelligent interfaces

We have been interested in applying the power of networks of chaotic systems to derive new measuring concepts and technical means. Moreover, we have been interested in modeling natural sensing structures.

### 3.1 Current models for the sensing tissues

The state of the art in modeling biologic sensing structures, as established by Freeman and several others, is well summarized in a paper by Baird and Eeckman. They state in the section titled "1.1. Computing with attractors" that: "In the design of this system, we follow an approach inspired by a particular concept of the physical structure **required** of macroscopic computational systems **in general for reliable computation**..." (our underlying).

Next, these authors detail: "We view a computational medium as a set of **structurally stable** subsystems ..." and further on: "By 'structurally stable' we mean that the dynamical behavior of each subsystem is **to a large extent immune to small perturbations** due to noise **or parameter changes**."

This common wisdom has been the law in technology for a long time already. Notice that in this research, we have started by contradicting one of the above hypotheses, as subsequently explained - and we do believe that the new hypothesis we shall formulate may throw light on several biological sensing processes.

Coming back to the quoted article, the authors state, moreover, that: "We assume that the dynamics of each subsystem is organized into attractor basins. As the overall system evolves in time (...), each subsystem passes through a sequence of attractors. *These sequences of attractors constitute the 'computation' of the system.*"

Further on, the authors explain that digital computers include flip-flops that basically have two attractors, denoted by '0' and '1', respectively, while they propose using oscillating subsystems ("neurons" or populations of neurons) that generate several categories of slow or fast bursts of impulses (the basic attractors), like natural neuronal networks. They further hypothesize that the state (attractors) are sampled and "clocked to change" by the alpha rhythm at a 10 Hz rate in the brain, and suggests a similar procedure is followed in technical systems.

## 3.2 Analysis of the hidden hypotheses behind the current models for computing with attractors and sensing tissues

There are two hypotheses in the quoted paper that may hidden the operation of living neurons and sensing tissues. Both hypotheses are used to simplify simulations and hardware implementations of these models. We shall point them out and discuss the consequences of rejecting them.

**A.** The first hypothesis is that "the dynamical behavior of subsystems is **immune to small perturbations** due to noise or **parameter changes**."

In natural sensing living systems, this can not be true; instead, we have to account for the very high sensitivity of living sensing systems to specific perturbations. This sensitivity, which is typically orders of magnitude higher than that of technical sensors, is actually based on the sensitivity to small perturbations. A visual neuron in the retina can sense minute changes in light, while a hearing cell in the inner ear can sense small displacements of the otoliths in the cochlea. (An otolith is a calcareous concretion in the inner ear.)

Consider a model for the sensing tissue consisting in an artificially built dynamic sensing system [15]. Let us first assume, that the input to the dynamic sensing system is the measurement value. To account for the operation of the sensing tissues, we need to induce high sensitivity in the models of oscillating cells, and this equates:

i) agreeing either with very high sensitivity of the attractors in the dynamic behavior, meaning that there are attractors having similar "specific energies" (a fundamentally unstable behavior), or

ii) agreeing with the high sensitivity inside the boundary regions between attractor basins, meaning that the operation is based on the transitory regime of the system, rather than on the steady state regime (on the attractors).

We have used both approaches. We have noticed during the research that the second approach could be feasible, under certain conditions, still preserving some degree of operational stability.

**B.** The second, less obvious hypothesis is that natural neurons and living tissue in general separates internal parameters from the operation during their task accomplishment.

### 3.3 Relaxed hypotheses to accommodate high sensitivity

In contrast to the above hypothesis, it is well known that information traffic is chemically (not energetically) driven (mediated). Moreover, the chemical mediators do change the internal parameters of the synapse and neurons, while reactivity to the internal environment (e.g., blood O2 or CO2 content) is also due to change of internal (metabolic) parameters of the living cell. Simulating such changes on digital computers is a rather large computational extra burden, but modelling them with chaotic electronic circuits may prove simpler. (Hence, one of our research efforts has been directed toward this "side purpose": to create hardware "computing systems" specifically designed to solve the modelling problem in a faster, more flexible way.)

The second, less obvious hypothesis is that natural neurons and living tissue in general separates *internal parameters* from the *operation* during their task accomplishment. In contrast, information is chemically (not energetically) driven (mediated). Moreover, the chemical mediators do change the internal parameters of the synapse and neurons. Therefore, we may assume that the sensitivity is due to the change of the attractor when the system *changes* (time-dependent system, instead of time-independent system).

We have implemented these modified hypotheses by choosing circuitry that exhibit large regions corresponding to families of attractor basins. We define here a family of attractor basins *in the parameter space* a set of sets of attractors, each set including attractors that are characterized by largely similar dynamic regimes, while the dynamic regimes significantly differs from set to set. It is also possible in the parameter space to have a large region basically corresponding to a family of dynamic regimes, with small regions (islands) of different dynamic regimes (see Figure 3).

### 3.4 Intelligent interfaces based on dynamic sensing

Based on our previous researches and on results, we have proposed the concept of *intelligent interface using chaotic systems* [15-22]. A classic interface has the block diagram shown in figure 4.

The concept of interface we propose is much closer to that in living structures. It consists of merging sensors, sensor drivers, signal processing, and classification operations in a single process. To achieve this goal, the sensors are part of the physical system that performs the

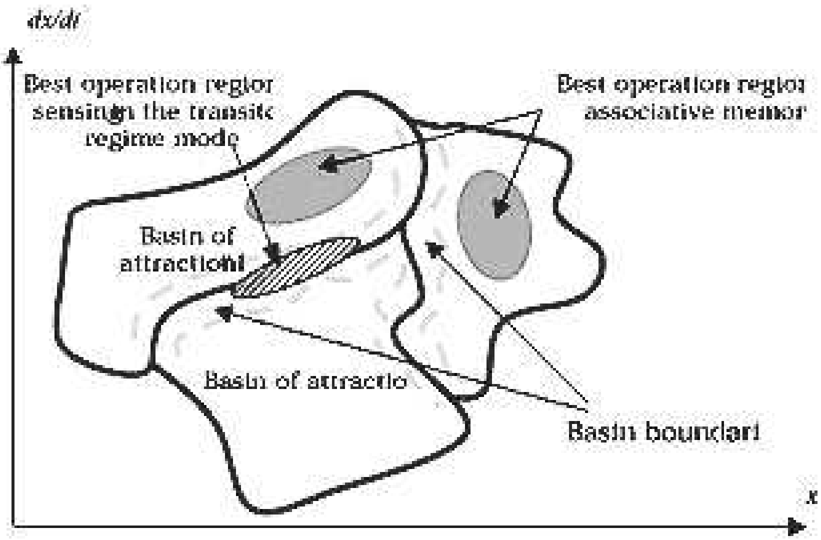


Figure 2.Possible manner of operation of a time-independent chaotic system, with sensing in the input space and detection in the transitory regime mode.

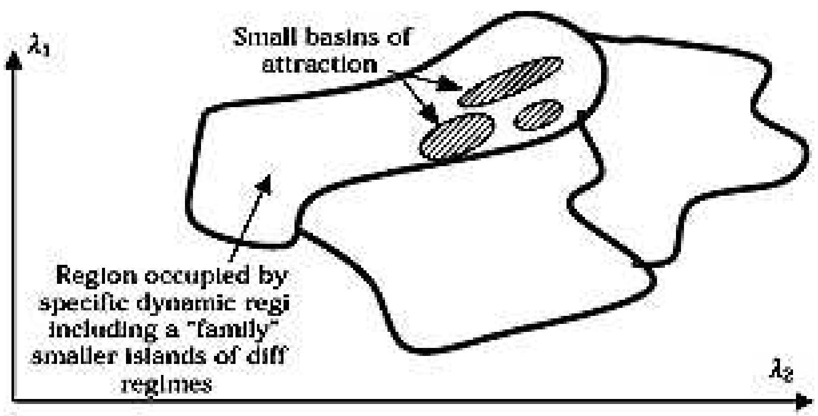


Figure 3.Possible manner of operation of a time-dependent chaotic system, with sensing in the parameter space and detection in the stationary regime mode.

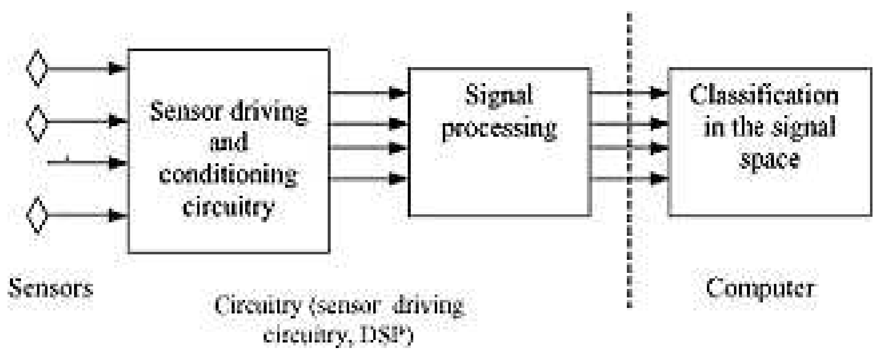


Figure 4.Block diagram of a complete interface.

signal generation and, at the same time, signal classification. Precisely, the sensors are parts in a nonlinear dynamic system, whose states are dependent on the sensed values and whose dynamical behavior can be classified according to the attractor type (basin of attraction). Another step forward was done by performing classification by means of *boundary regions between attraction basins*, rather than by the basins themselves. Moreover, these basins are not required to be basins that include strange attractors of the system. This new technique, based on boundary regions, allows us to produce higher sensitivity sensing system. However, the best performing method is to use *families of dynamic regimes*, in the parameter space, and with "inputs" in the parameter space.

Previous researches concentrated on obtaining associative memory operation for various types of neural networks, including oscillating NNs, but have not focused on the sensing process itself. The concept of the interface we propose is shown in figure 5:

Examples of various attractors, including strange attractors, through whom a hardware circuit can pass during its evolution when a single circuit parameter changes, for fixed initial conditions, are shown in figure 6.

Coupling several such chaotic cells into appropriate network configurations (see fig. 7) allows mimicking the natural sensing systems. A problem to solve is the characterization of the attractors, in a simple and reliable way that does not require complex circuitry. We have devised several methods and circuits to solve this problem, allowing us to avoid the requirement of complex computations as needed for the Lyapunov exponents or other classic features of the attractors. For example, the computation of the average curvature of the attractor is such a method, which proves as sensitive as the method based on Lyapunov coefficients in evidencing the change of the attractors (Fig. 8).

Regarding the technological aspects, the use of random Boolean networks (RBN), as presented in [23], is preferred to other network architectures, like CAs, because of the closeness to the architecture of NNNs. Other *ad hoc* interconnecting schemes, inspired from the

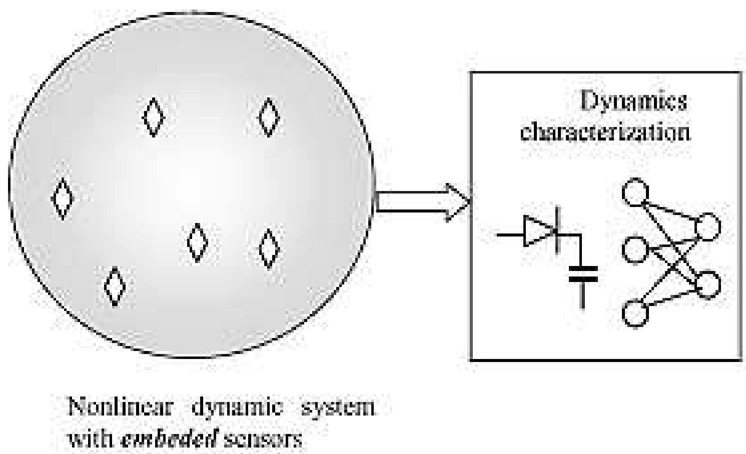


Figure 5.The sketch of the interface based on nonlinear dynamic systems, as proposed in this project. The dynamic characterization system may have a complexity that ranges from a very basic electronic rectifier to a complex neural network or a neuro-fuzzy system.

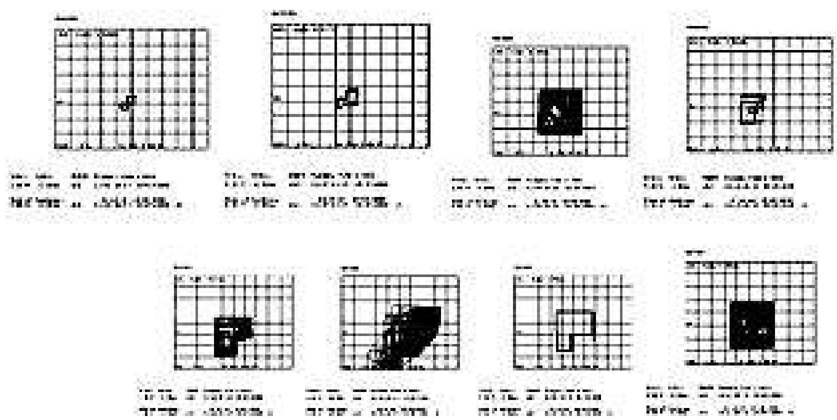

Figure 6.Examples of changes of attractors when a parameter value changes in a simple network of three oscillating systems. Every picture represents a different attractor that can be used as an output label.

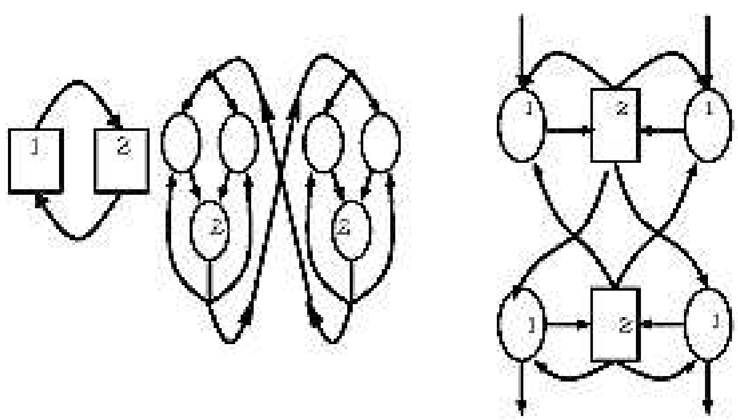

Figure 7.Examples of interconnections of chaotic systems to build elementary chaotic networks that mimic natural neural networks.

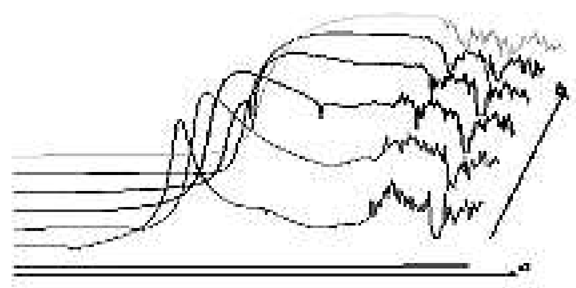

Figure 8.Change of the curvature, for a logistic process (step 0.1 for the second parameter). Computer simulation.

natural systems, are also considered. A large degree of flexibility in the hardware model, with respect to the parameters of the neurons and of their interconnections, should be provided to allow for a large enough range of models. We consider using some form of EHW (Evolvable Hardware) to insure higher flexibility to the system and its capability of learning. These technological aspects for the ChaNs are still in the study phase and will be reported elsewhere. It is our intention to develop several such models and to test them in the near future. We already have tested several models of 2 to 6 neurons of different types and we will connect such small neuronal clusters to form clusters that are more complex. The next steps are to build realistic ganglia and to automatically generate "environmental stimulation" to these neuronal clusters, to determine their behavior.

## 4 Conclusions

It is now recognized that we need to address different problems using different tools. The digital computers have proved their usefulness in several problem categories, and they definitely proved their limits in other problem categories, where they need algorithms that are NP. In some categories of problems, NNs are better suited, yet having their own limits. Quantum computing is another way to deal with several specific problems. Yet, no method is universal. Similarly, our brain is excellent in tasks like analog memory and associative memory, in pattern recognition and classification, in sensation formation, but it performs extremely poor in tasks involving computations. Looking around, we can see that computations are not essential for survival, while recognition of the predators and of the food are essential tasks. This can explain our brain abilities, as well as its limits. Even poorer is our brain ability to perform large number factorizations – and almost the same quality the digital computers have in this task. Human brain has developed an ability to use approximate reasoning, modeled today by fuzzy logic, ability that neither digital nor quantum computers have. It looks that neither our brains, nor our machines are able today to efficiently deal with large optimization problems, as our genes do. We have genetic algorithms, but do no have a specific hardware counterpart for them yet. Concluding these considerations, there is no reason to favor or even to compare on a general basis various types of computing paradigms and hardware approaches: We may expect to have in the future hybrid machines and systems of various types of machine to cope with a large pallet of problems. Computing with attractors may show, beyond promises, a benefits in fields as varied as medical diagnosis, intelligent interfaces, modeling in biology, robotics, modeling in psychology and cognitive or behavioral sciences. This research should be understood in this framework.

**Acknowledgments.** The first two authors acknowledge the partial support of a research Grant (#52664, 1997–2001) from Swiss National Founds. The first author also acknowledges a grant from Techniques & Technologies, Ltd. who supported part of the research.

## References

[1] Samuel L. Braunstein: *Quantum computation: a tutorial.* http://www.sees.bangor.ac.uk/ schmuel/comp/comp.html

[2] Henry P. Stapp: *On Quantum Theories of the Mind* (1997). http://citeseer.nj.nec.com/cache/papers2/cs/233/http:zSzzSz www-physics.lbl.govzSz stappzSz40369.pdf/stapp97quantum.pdf

[3] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, Umesh Vazirani: *Strengths and Weaknesses of Quantum Computing.* http://arXiv.org/abs/quant-ph/9701001

[4] M. W. Hirsch and B. Baird. *Computing with dynamic attractors in neural networks.* Biosystems Special Issue on Biological and Evolutionary Information Processing, 34(1,2) 173–195, 1995.

[5] S. Wolfram: *Cellular Automaton Supercomputing* (1988). www.stephenwolfram.com/publications/articles/computing/88-cellular/

[6] S. Wolfram: *High-Speed Computing: Scientific Applications and Algorithm Design*, ed. Robert B. Wilhelmson. University of Illinois Press, 1988

[7] Teodorescu, H.N., L. C. Jain, Abraham Kandel (Eds.): *Hardware Implementation of Intelligent Systems* (Studies in Fuzziness and Soft Computing, Vol. 74). (June 2001) Physica Verlag / Springer Verlag; ISBN: 3790813990

[8] Freeman, W. J. 1978. *Spatial properties of an EEG event in the olfactory bulb and cortex.* Electroencephalogr. Clin. Neurophysiol. 44, 586–605.

[9] B. Baird. *Bifurcation and learning in network models of oscillating cortex.* Physica D, 42:365–384, 1990.

[10] B. Baird. *Learning with synaptic nonlinearities in a coupled oscillator model of olfactory cortex.* In F. Eeckman, editor, Analysis and Modeling of Neural Systems, pp. 319–327, 1992.

[11] Zaoping Lee and John Hertz: *Odor recognition and segmentation by a model olfactory bulb and cortex.* Comput. Neural Syst. (2000) 83–102

[12] Zaoping Lee and John Hertz: *Odor recognition and segmentation by coupled olfactory bulb and cortical networks.* Journal Neurocomputing 26–27, pp.789–794 (1999).

[13] DeLiang Wang, David Terman: *Image Segmentation Based on Oscillatory Correlation.* Neural Computation, vol. 9, (1997), pp. 805–836 (this paper includes an extensive bibliography on the topic)

[14] Robert W. Williams and Karl Herrup: *The Control of Neuron Number.* http://www.nervenet.org/personnel/PAPERS/NUMBER_REV_1988.html#1

[15] Teodorescu, H.N.: *Nonlinear sensor.* RO Patent, 2000.

[16] Teodorescu, H.N.: *Modeling natural sensitivity: ALife sensitive, selective sensors.* J. Biomedical Fuzzy and Human Sciences – BMFSA Journal (Japan). Volume 6, Number 1, October 2000, pp. 29–34, http://www.kct.ac.jp/BMFSA/journal/06/

[17] Teodorescu, H.N., Abraham Kandel: *Fuzziness and Artificial Life: A New Paradigm of A–Life.* Proc. 6th International Conference on Soft Computing (IIZUKA2000), Japan, September 30-October 4, 2000

[18] Teodorescu, H.N., A. Kandel: *Analysis of a class of chaotic models for sensorial processes.* ECIT2000 Conference, 25–28 Sept. 2000 (CD Proceedings)

[19] Teodorescu, H.N., A., Stoica, D. Mlynek, A. Kandel, J.C. Iov: *Nonlinear Dynamics Sensitivity Analysis in Networks and Applications to Sensing.* Proc. 9th IFAC / IFORS / IMACS / IFIP/ Symposium on Large Scale Systems: Theory and Applications. July 18–20, 2001, Bucharest, Romania

[20] Teodorescu, H.N.: "Information processing using chaotic systems and interfaces". Seminar, July 11, 2001, Swiss Federal Institute of Technology, Lausanne, Switzerland

[21] Teodorescu, H.N., Jan-Catalin Iov: *Parametric Pattern Control for Cellular Automata with Self-Oscillating Neurons.* Proc. IEEE Symposium SCS2001, Iasi, July 2001

[22] Peillon, X., Teodorescu, H.N., Mlynek, D., Kandel, A.: *Analysis of chaos sensitivity in circuits based on Wien oscillators*, Int. J. Chaos Theory & Applic., vol. 4 (1999), nr. 4, pp. 21–34

[23] Wuensche, Andrew: *Discrete Dynamical Networks and their Attractor Basins.* Complexity International. Vol. 6 (1998), e–Journal. www.csu.edu.au.ci/vol06/wuensche/wuensche.html

H.-N. Teodorescu, D. Mlynek, A. Kandel,

Received August 10, 2001

Horia-Nicolai Teodorescu,
Technical University of Iasi,
Romania,

Institute of Information Science,
Romanian Academy,

Daniel Mlynek,
Swiss Federal Institute of Technology,
Lausanne, Switzerland

Abraham Kandel,
University of South Florida,
Tampa, USA