Attractor Systems and Analog Computation

Hava T. Siegelmann

Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel. iehava@ie.technion.ac.il

Shmuel Fishman

Department of Physics, Technion, Haifa 32000, Israel. fishman@physics.technion.ac.il

ABSTRACT

Attractor systems are useful in neurodynamics, mainly in the modeling of associative memory. This paper presents a complexity theory for continuous phase space dynamical systems with discrete or continuous time update, which evolve to attractors. In our framework we associate complexity classes with different types of attractors. Fixed points belong to the class BPP_d, while chaotic attractors are in NP_d. The BPP=NP question of classical complexity theory is translated into a question in the realm of chaotic dynamical systems. This theory enables an algorithmic analysis of attractor networks and flows for the solution of various problem such as linear programming. We exemplify our approach with an analysis of the Hopfield network.

I. INTRODUCTION

A digital computer can be viewed as a dynamical system. It starts at some input state, follows a trajectory in state space and may converge to an output state. Classical physical systems can also be represented by dynamical systems; if a system is dissipative, flow to an attractor takes place. The fundamental difference is that for physical systems the state space is continuous, while for computers the state space is discrete. In addition the attractors of physical systems are not limited to fixed points but may be chaotic as well. This establishes a correspondence between computers and physical systems through the common description by dynamical systems. We interpret the attractor to which a system flows as the output, the initial condition plays the role of the input, and the flow is the process of computation. What determines the output is the location of the initial condition relative to the basin boundaries of the various attractors. These boundaries may be fractal even if the attractors are regular [1,2]. Generally, there is no way to predict analytically the basins and their boundaries, and thus the general evolution is unpredictable and can be very complicated.

The goal of this research (an extended version will be published elsewhere [3]) is twofold: to develop an interpretation of the physical world as performing a processes of computation, and provide a theoretical background for an algorithmic and complexity analysis of computation in an analog computation framework. There seems to be an algorithmic advantage in using analog computers for specific problems. A good example is the linear programming problem. In another paper we demonstrate that a flow based on the principle of Karmarkar's interior point algorithm has optimal performance [4]. This exemplifies the power of using a continuous rather than discrete phase space.

In the rest of the introduction we show the connection of our work to other lines of research. In the field of neurodynamics both content addressable memory and the associative memory are frequently modeled by dissipative dynamical systems, for which an energy (Lyapunov) functional can be defined. Recalled patterns reside in the location of local minima of the energy. The most popular model is probably the Hopfield network [5-7]. Consequently, the computational capabilities of dynamical systems that are relevant to neurodynamics - non-linear dissipative systems - are of great interest.

A similar view of the process of computation from an initial condition to an attractor has been used to design continuous time algorithms. Brockett introduced a set of ODE's that perform various tasks such as sorting and solving linear programming problems [8]. In the book of Helmke and Moore [9] one can find numerous other applications and references. Our theory is a continuation of their work, in that it provides a natural framework for the complexity analysis of continuous time algorithms. Furthermore, we allow for attractors which are not fixed points or limit cycles.

We differentiate our work from the following seemingly related work. In an effort to bound the computational power of continuous systems, one line of work was done on performing step by step simulation of discrete computational models [10]. Whereas this type of work does not express the inherent computational capabilities of continuous systems, we take continuous systems as is, and interpret their evolution as a process of computation.

Analog computation can be utilized to test possible

theoretical limitations of the "physical Church-Turing thesis" [11] that states that the computational capabilities of any physical device should not exceed (in idealization) that of a Turing machine [12]. If a device that computes problems that cannot be computed by the Turing machine (and therefore digital computers) is found, it will challenge the "physical Church-Turing thesis" and will therefore be of great interest. Some theoretical analog models of computation have the capability of computing beyond the Turing limit [13,14], but no realizable super-Turing system has been noted. We do not suggest the current work as providing a step towards the identification of super-Turing natural systems. We rather concentrate on perceiving physical systems as efficient special purpose computers.

I.1 The Dynamical Model

A dynamical system is defined by a set of equations that allow to predict the evolution. Such systems may be continuous in time, e.g.

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}(t)) \qquad (I.1)$$

where $\mathbf{x}(t)$ is a d-dimensional vector and $\mathbf{F}$ is a d-dimensional vector function of $\mathbf{x}$. They can also take the form of a map:

$$\mathbf{x}_{n+1} = \mathbf{T}(\mathbf{x}_n). \qquad (I.2)$$

Maps can be related to physical systems defined by continuous equations like (I.1) by a *Poincaré map* [15], by the *stroboscopic map* and in various other ways. For the stroboscopic map, the position is monitored in intervals of length τ, namely at times $t = n\tau$, where n is an integer. Our results will be presented for continuous systems but they hold for maps as well [3].

Dynamical systems can be classified into two classes. *Hamiltonian* systems which preserve the volume of their phase space. For *dissipative systems*, on the other hand, the phase space volume decreases in time. As a consequence, dissipative systems typically are characterized by the presence and the nature of attractors.

An attractor may be either regular, for example a fixed point or a limit cycle; it may also be chaotic. The region of phase space that flows to a certain attractor is called the basin of attraction [15-17]. These basins are separated by basin boundaries; the basin boundaries are either smooth or fractal.

An attractor is called chaotic if the following two conditions exist: (1) Locally, the separation between most nearby pairs of points that reside on the attractor grows apart exponentially; (2) The attractor is

bounded in phase space. Because of this exponential growth, prediction becomes exponentially difficult. Their irregularity is part of the intrinsic dynamics as opposed to the case of stochastic systems.

Our interest is in physically realizable, dissipative dynamical systems with the extra feature that *the convergence to the attractor is exponentially fast.* This last requirement does not constrain our theory much because such systems are abundant in nature, although other possibilities exist (see [3]). These require vanishing Lyapunov exponent, that is an additional constraint, and therefore these possibilities are scarce. We also assume that the initial conditions are not in the vicinity of the boundaries. This is true for most initial conditions in presence of isolated attractors. In this simplified model the richness of computation stems from the physical complexity of the attractors. More general cases are of great interest and should be considered in future work.

I.2 The Computational Model

The following computational question will be examined: starting from some point in phase space, to which attractor will the trajectory flow? Our purpose is to evaluate the computability and to estimate the computational difficulty (e.g. P, NP, etc.), involved in answering this question for various types of dynamical systems.

We define a computation for general dissipative dynamical systems as follows. The initial condition corresponds to the input. The system evolves until approaching an attractor. Because the actual convergence to an attractor takes an infinite time, we do not require complete convergence but rather define that the calculation is completed when an ϵ-vicinity of the attractor is approached and the system is verified to be confined there. Deciding to stop the computation after it is found that the trajectory is attracted to the ϵ-vicinity of the attractor may sound obscure because the ϵ-vicinity may contain several attractors that can be resolved only for smaller values of ϵ. This is actually not a problem but rather a manifestation of the richness of the corresponding computation. As the resolution is increased, new and more refined results are found.

I.3 Computational Complexity

In discrete time computational models, the computation time is quantified by the number of steps until halting, and this is used to measure the complexity of the process. This definition can be adapted for maps; it is however meaningless for continuous time systems. A different definition is required, and is considered an important issue in computer science (e.g., [18-21]).

To see why the problem of definition of a computation time scale is challenging, assume we wish to

describe computation by a system of the type (I.1). If **F** is multiplied by a constant a, the computation time changes but the complexity should stay unchanged. Our suggestion is to introduce a concept related to a natural physical time scale: this is the **characteristic computation time scale** that will be defined by the rate of convergence of the underlying physical system. The complexity of the process will be then measured as the time — quantified in multiples of the time scale — required to complete the process. This definition is applied to both continuous and discrete time systems. For the discrete case the relation to the classical definition of computation time is transparent and the difference between these definitions is clear.

We next wish to emphasize the importance of having the input and output in form of bits, that is, the input to the machine can be thought of as a string of binary digits and the output adheres to the same restriction. The finiteness of the input and output is a crucial requirement in the theory of computing, assuring that the power of models is purely based on the internal structure rather than on higher input/output precision. The complexity of computation is defined with respect to the length of the input. We take a similar approach: to comply with the finiteness of input and output, and with the decision of ending a computation, *the computation in a dynamical system is defined relatively to a grid*. The spacing of the grid is of the order of the precision in the initial conditions. If the flow converges to some fixed point $\mathbf{x}^*$, the output will be taken as the closest point on the grid. The convergence time t_c will be so that for all $t > t_c$ the distance from the fixed point satisfies $|\mathbf{x}(t) - \mathbf{x}^*| < \epsilon$. The connection between the precision in the dynamical evolution and the number of bits in the input is given by the log ratio; that is, for grid spacing of size ϵ the number of "input bits" is $|\log \epsilon|$.

We next define the characteristic time scale for our class of exponentially converging dissipative systems: assume first that the relevant stable attractor is a fixed point x^*, one can linearize (I.1) around this point, to obtain

$$\dot{\delta \mathbf{x}} = M \ \delta \mathbf{x} \qquad (I.3)$$

where $\delta \mathbf{x} = \mathbf{x} - \mathbf{x}^*$ and M is the *stability matrix*. The eigenvalues of the stability matrix are the Lyapunov exponents (the fixed point is assumed to be stable, hence all these are negative). The rate of convergence is determined by the largest of them, that will be denoted by $-\lambda$. In the vicinity of the fixed point $|\mathbf{x}(t) - \mathbf{x}^*| \sim e^{-\lambda t}$, and $1/\lambda$ is the characteristic time scale for convergence. Since the time spent near a fixed point dominates the elapsed time (unless the initial point is very close to the boundaries), we can

choose $1/\lambda$ as the formal characteristic time of the computation. It depends only on the nature of the fixed point and not on any details of the physical system. Obviously the numerical value of λ depends on the units used to measure time. The change of the time scale is equivalent to the multiplication of **F** in (I.1) by a scalar constant. Such a characteristic time scale will be found any time exponential convergence to an attractor takes place. It may be a fixed point, a periodic orbit or even a strange attractor. A similar definition can be introduced for maps. If there are several attractors the computation time will be defined as the smallest characteristic time.

Having this time scale, we can calculate the time complexity of the dynamical system. The *convergence time t_c* is what it takes for the trajectory to flow from the initial point to the point where it terminates. It is

$$t_c = t_\epsilon + t_f + t_B \qquad (I.4)$$

where t_ϵ is the contribution from the linear regime (see (I.3)), t_B is the contribution from the vicinity of the boundary, if the initial condition is there, while t_f is the contribution from the other regions. The convergence time t_c does not constitute the whole computation; one also has to verify that the computation has indeed ended. Here the differences between the complexity of the attractors come to sculpt the different computation classes in dynamical systems.

In this paper we illustrate how the theory can be used for flows to fixed points and to isolated chaotic attractors. For more applications see [3].

II. COMPUTATION FOR SYSTEMS WITH A LYAPUNOV (ENERGY) FUNCTIONAL

First we calculate explicit bounds on the computation time for simple systems for which a Lyapunov or energy functional satisfying

$$\frac{dE(\mathbf{x}(t))}{dt} < 0 \qquad (II.1)$$

can be defined. In this case the flow (if it is bounded) is to fixed points. The fixed points of the continuous systems and of the corresponding discrete maps are exactly identical, since these are the zeros of **F**.

When the phase space velocity of the trajectory $\frac{d\mathbf{x}}{dt}$ is found to be smaller than ϵ (in appropriate units) for some time, it is likely that it is in the ϵ-vicinity of a stable fixed point. For some problems it is known that the flow is only to stable fixed points [4]. For other situations, one may want to verify that it is indeed in the vicinity of such a point and not near the

stable manifold of a hyperbolic point. Our suggestion is to conduct a probabilistic check. Assume we want to validate convergence of the point $\bar{x}$ that is ϵ-close to a suspected fixed pint. We create k new points $\bar{x}_1, ... \bar{x}_k$ by adding to $\bar{x}$ independent noise of strength up to ϵ. Now the system is called repetitively on each of these points; each time it is stopped after time t_v to be specified later. Consequently a cluster of trajectories in a sphere of radius ϵ around the fixed point is generated. If the fixed point is stable the sphere shrinks, while if it is hyperbolic it gets stretched in the unstable direction at the rate $e^{\lambda_1 t}$, where λ_1 is the corresponding Lyapunov exponent. After a time of the order $|ln\frac{\epsilon}{\delta}|$ it reaches a distance $\delta \gg \epsilon$ from the fixed point. For a hyperbolic point, there is a finite probability p to find one of the k trajectories at a distance between δ and $\frac{1}{2}\delta$ from the fixed point. If one of the trajectories is found at such a distance we conclude that the fixed point is unstable and continue the computation from $\bar{x}$ until it runs away to another fixed point. The probability that non of the trajectories reaches the distance between δ and $\frac{1}{2}\delta$ from an unstable fixed point is $(1-p)^k$, that is exponentially small in k. Note that $t_v = O(|log\epsilon|)$ and therefore the verification time t_v is polynomial. It was assumed here that both ϵ and δ are in the linear regime where (I.3) holds.

We turn now to estimate the total time of computation for such systems. The various contributions to convergence time (I.4) are evaluated. The time it takes to flow through the linear region from a distance δ of the stable fixed point to its ϵ-vicinity is

$$t_\epsilon = \frac{1}{\lambda}|ln\frac{\epsilon}{\delta}| \qquad (II.2)$$

where $-\lambda$ is the largest Lyapunov exponent (note that all of them are negative). If the initial point is assumed to be in a narrow region of width η in the vicinity of the basin boundary, $t_B \sim O(|ln\eta|)$. If $\epsilon \ll \eta$ the computation time is dominated by t_ϵ. We focused in this work on these cases only. Since at the fixed point $\frac{dE}{dt}$ vanishes, there exists a δ so that when the distance from the fixed point is larger than δ, $|\frac{dE}{dt}| > v_\delta$, where v_δ is a positive number. Consequently if ΔE is the difference between the maximal and minimal values of E, the time of flow t_f from the initial point to the point of distance δ from the fixed point satisfies $t_f < \frac{\Delta E}{v_\delta}$. The exact value v_δ depends on the particular system at hand.

We next estimate v_δ for two examples: the gradient flow and the Hopfield neural network. By the chain rule we find

$$\frac{dE}{dt} = \sum_i \frac{\partial E}{\partial x_i}\frac{dx_i}{dt} \qquad (II.3)$$

that will be useful in what follows. We start with the gradient flow (such as in the case of a particle rolling down hill) which is defined by

$$F_i \equiv \frac{dx_i}{dt} = -\frac{\partial E}{\partial x_i}. \qquad (II.4)$$

In this case $\frac{dE}{dt} = -|\mathbf{F}|^2$ and v_δ is just the value of $|\mathbf{F}(x_\delta)|^2$ at some point x_δ at a distance δ from the fixed point in its linear regime.

A more sophisticated example is the Hopfield network [7]. The continuous version of the Hopfield network is defined by

$$\frac{dx_i}{dt} \equiv F_i = -x_i + \sum_j W_{ij}\sigma_j(x_i) \qquad (II.5)$$

with symmetric weights $W_{ij} = W_{ji}$ and $\sigma_j(x_j) = tanh x_j$. Our analysis allows a generalized Hopfield model with the functions σ_j satisfying:
(i) $|\sigma_j| \leq 1$;
(ii) $\frac{\partial \sigma_j(x)}{\partial x} > 0$;
(iii) $\frac{\partial^2 \sigma_j(x)}{\partial x^2}$ is negative for positive x, positive for negative x and tends to zero monotonically in the limits $\pm\infty$.
The energy E is defined here by: $F_i = -\frac{1}{c}\frac{\partial E}{\partial \sigma_i}$, where c is a positive constant, leading to

$$\frac{dE}{dt} = -c\sum_i \left(\frac{\partial \sigma_i}{\partial x_i}\right)\left(\frac{dx_i}{dt}\right)^2 \qquad (II.6)$$

Because of condition (ii), the differential of the energy is indeed negative. One finds that v_δ is proportional to $|\mathbf{F}(x_\delta)|^2$. The convergence time of the generalized Hopfield network is thus

$$t_c \leq t_\epsilon + t_B + C\frac{\Delta E}{|\mathbf{F}(x_\delta)|^2} \qquad (II.7)$$

where x_δ is some point in the linear regime of the fixed point and C is a constant.

We conclude that for systems with a decreasing energy functional the computation approaches a fixed point within a small precision ϵ in time $t_\epsilon = \frac{1}{\lambda}|ln\frac{\epsilon}{\delta}|$. The verification time is of the same order of magnitude. These systems are thus belonging to the computation class P_d of polynomial-dynamics, that correspond to the class P in the standard computation theory, if verification is not required [4]. If it is required, these are belonging to BPP_d corresponding to BPP.

III. COMPUTATION FOR CHAOTIC SYSTEMS

Often flow to chaotic strange attractors takes place. The behavior of a chaotic system can be very rich, exhibiting structures on all scales. These increasingly rich structures are revealed as the grid of resolution is decreased. Systems where the number of chaotic attractors is arbitrarily large can be found [22,23] and thus many possible output responses are possible in the associated computation. Such a situation can be found when weak dissipation is applied to a chaotic Hamiltonian system exhibiting complicated structures that consist of elliptic and hyperbolic fixed points as well as of chaotic layers. The Kolmogorov-Arnold-Moser and Poincare-Birkhoff theorems assure the existence of such structures [15,24].

As in the case of regular attractors our focus is still on systems for which attractors are approached exponentially in time, namely as $e^{-\lambda t}$. The verification, though, is much harder here, causing longer total computation time: determining the location of the attractor within a precision ϵ, may require exponentially long time for chaotic systems (it is polynomial for fixed points).

For many dynamical systems strange attractors are multifractal: their measure on various points is not uniform. Let the measure in a hypersphere of radius ϵ on the attractor be $\mu_i = \epsilon^{\alpha_i}$. The measure is smallest in regions where $\alpha_i = \alpha_{max}$. The time it takes to reach the region of the smallest measure is inversely proportional to it. Therefore, the time required to compute the location of the attractor is

$$t_A \sim \epsilon^{-\alpha_{max}} = e^{\alpha_{max}|\ln \epsilon|} \qquad \text{(III.1)}$$

Hence: if the attractors are known (for example, from previous calculations), and this exponential amount of information is kept, the computation time required to determine the attractor to which the flow takes place is proportional to $|\ln \epsilon|$. Otherwise, it is **exponential in** $|\ln \epsilon|$.

However we can do better than exponential time when the strange attractors are chaotic: consider the question "Does a given point x approach a typical isolated attractor Φ ? " The attractor is specified by one of its fixed points, we abuse the notation, using Φ both to denote the fixed point and its associated attractor. In addition to Φ another real constant ν is provided so that any periodic orbit that passes in the ν neighborhood of Φ is on the attractor as well. Now the problem is defined as follows:
Given an initial point with precision ϵ decide whether this point is attracted to the ϵ vicinity of the attractor Φ.

We next show that this problem is in the class NP_d for our model. This is the class of the dynamical systems corresponding to the class NP in the standard computational theory. Given an initial point x, there is a guess point y so that y is in the ν neighborhood of Φ and also y belongs to a periodic orbit of length $O(|\log \epsilon|)$. These two properties are easy to verify and hence it is easy to know that y is on the attractor. Furthermore, the periodic orbit of the point y is going to pass in the ϵ vicinity of the trajectory of x in time $O(\log \epsilon)$, provided that x converges to the attractor associated with the point Φ. If such a point y is guessed, the affirmative answer can be indeed verified polynomially. For this purpose we proved the following lemma:

Lemma III.1 There is a constant ν such that for any initial point x in the basin of attraction of an isolated chaotic attractor, that is denoted by one of its fixed points Φ, there is a point y_x so that:

1. y_x is in the ν-vicinity of the fixed point Φ.

2. y_x is on a periodic orbit of length $O(|\log \epsilon|)$.

3. The periodic orbit of (2) passes in the ϵ vicinity of the trajectory starting from x (towards the attractor) in time up to $O(|\log \epsilon|)$.

The correctness of this lemma will validate the existence of such y and the fact that the above problem is indeed in NP_d. The justification of the lemma relies on the exponential proliferation of periodic orbits [3].

Is it possible to verify general chaotic attractors in deterministic polynomial time? For specific cases where a well defined structure is given by a simple rule an affirmative answer is possible. It is believed, however, that usually it is not the case.
Conjecture: Unless *all* chaotic attractors can be described by simple rules, $P_d \neq NP_d$

So far it was assumed in our discussion that the attractors are isolated. Also the computational properties of intermingled attractors and attractors in crisis were studied [3]. If the chaotic attractors are intermingled (riddled basins) only probabilistic computation can be defined. Chaotic systems which exhibit crisis demonstrate natural undecidable computation.

REFERENCES

[1] S.W McDonald, C. Grebogi, E. Ott, and J.A. Yorke. Fractal basin boundaries. *Physica D* 17, 125-153, 1985.

[2] G.H. Hsu, E. Ott, and C. Grebogi. Strange saddles and the dimensions of their invariant manifolds. *Phsics Letters*, 127(4), 199-204, 1988.

[3] H.T. Siegelmann and S. Fishman. Analog Computation and Dynamical Systems, *Physica D*, to appear.

[4] A. Ben-Hur, H.T. Siegelmann and S. Fishman. An optimal algorithm for linear programming in an analog computation framework. In preparation.

[5] J.J. Hopfield. Neurons with graded responses have collective computational properties like those of two-state neurons. In *Proc. of the Natl. Acad. of Sciences*, volume 81, pages 3088-3092, USA, 1984.

[6] J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141-152, 1985.

[7] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, 1991.

[8] R.W. Brocket. Dynamical systems that sort lists. *Linear Algebra and Its Applications*, 146:79-91, 1991.

[9] U. Helmke and R. Moore. *Optimization and Dynamical systems* Springer, 1995.

[10] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138: 67-100, 1995.

[11] R. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21, 1982, 467-488.

[12] R. Penrose. *Shadows of the Mind*. Oxford University Press, Oxford, England, 1994.

[13] H. T. Siegelmann. Computation beyond the Turing lim *SCIENCE*, 268(5210), 545-548, 1995.

[14] H. T. Siegelmann and E. D. Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131, 1994. 331-360.

[15] E Ott. *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge, 1993.

[16] C. Grebogi, E. Ott, and J. A. Yorke. Chaos, strange attractors, and fractal basin boundaries in nonlinear dynamics. *SCIENCE*, 238:632-637, October 1987.

[17] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer Verlag, New York, 1983.

[18] C.E. Shannon. Mathematical theory of the differential analyzer. *Journal of Mathematics and Physics of the Massachusetts Institute of Technology*, 20:337-354, 1941.

[19] M.B. Pour-El. Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers). *Transactions of the American Mathematical Society*, 199:1-29, 1974.

[20] L. A. Rubel. Digital simulation of analog computation and Church's thesis. *The Journal of Symbolic Logic*, 54(3):1011-1017, September 1989.

[21] K. Ko. On the computational complexity of ordinary differential equations. *Information and Control*, 58, 1983. 157-194.

[22] L. Poon and C. Grebogi. Controlling complexity. *Phys. Rev. Lett*, 75:4023-4026, 1995.

[23] U. Feudel, C. Grebogi, B. Hunt, and J.A. Yorke. A map with 100+ low period periodic attractors. *in Phys. Rev. E*, 54(1):71-81, 1996.

[24] V.I. Arnold and A. Avez. *Ergodic Problems of Classical Mechanics*. Adddison-Wesley, NY, 1989