NASA-UCLA Institute for Flight Systems Research

REAL-TIME SUPPORT FOR HIGH PERFORMANCE AIRCRAFT OPERATION

NASA GRANT NAG 2-302

Principal Investigator: Jacques J. Vidal

UCLA Computer Science Department and Brain Research Institute
3531 Boelter Hall
UCLA
Los Angeles, CA 90024-1596

FINAL REPORT 1986-88

Mr. Eugene L. Duke
NASA Technology Officer
Vehicle Technology Branch D-OFV
Dryden Flight Research Facility
P.O. Box 273, Edwards Ca., 93523

January, 1989

University of California, Los Angeles

REAL-TIME SUPPORT FOR HIGH PERFORMANCE AIRCRAFT OPERATION

FINAL REPORT 1986-88

Jacques J. Vidal, Principal Investigator
UCLA Computer Science Department

TABLE OF CONTENT

REAL-TIME SUPPORT FOR HIGH PERFORMANCE AIRCRAFT OPERATION

FINAL REPORT 1986-88

Jacques J. Vidal, Principal Investigator
UCLA Computer Science Department & Brain Research Institute

INTRODUCTION

This document describes the research conducted under this NASA Grant at the UCLA Distributed Machine Intelligence Laboratory and under the direction of the Principal Investigator. The central research theme for the laboratory is the exploration, development and eventual prototyping of **novel methodologies and architectures** for **real-time control** and **pattern processing**.

The **target applications** are **control systems** that are difficult for sequential machines to handle in real-time. Many applications, in the realm of high performance aircraft are characterized by extremely high and rising demands for processing speed, combined with an ever increasing volume of the relevant information.

Examples include craft or missile **interception** (and the inverse problem of **collision avoidance**), the **real-time guidance** of **smart munitions**, real-time **optimal path routing**, **autonomous vehicle control**, tactical mission control and malfunction management. (Applications in other fields would include high throughput combinatorial processes such as encrypting and decrypting images or text.)

Another issue of concern is that of the mismatch between hardware and software in the problem domain. There is a critical need for real-time systems that could provide **decision assistance to controllers and pilots during flight tests or missions** for **status display, failure monitoring** or to **avert or recover from dangerous conditions**. These problems also require advanced **human-machine interfaces** and **sophisticated visualization**. Requirements for such systems are often well beyond the performances of the current AI software running on von Neumann machines.

In search of alternate approaches to real-time control, we have been investigating the feasibility of real-time processing schemes using **Artificial Neural Networks (ANNs)** to increase throughput.

We have been making rapid progress toward the specification of an **integrated architecture model** using **purely digital adaptive networks integrated with conventional sequential processors**.

RATIONALE FOR DIGITAL NEURAL NETS

We feel that the development of entirely **digital but truly parallel network implementations** is of extreme importance. The only truly parallel neural network designs pursued elsewhere require an implementation based on analog electronic circuitry. The usual alternative, suggested in particular in a recent DARPA report on Neural Network perspectives, is to use **simulation** with "a relatively small number of high speed specialized processors".

Both alternatives will fall short of meeting the requirements of real-time control, although for different reasons.

Analog neural networks are not supported by a mature technology. Distributed process control mechanisms, crucial to the implementation of adaptive networks are conspicuously absent in most models. Generally speaking, analog computers have all but disappeared from the contemporary scene. The reasons have been a general lack of flexibility, a narrow domain of application, and problems in scaling up to large systems. Whether analog networks can be realized in VLSI on any realistic scale remain very speculative. To this day, the marriage of analog systems with large scale integrated technology has also been uneasy and analog computing has not in fact made the transition into the VLSI age.

The sequential processor simulation alternative has been the most popular option. The quest for massive parallelism is even dismissed as a "major fallacy" in the report mentioned earlier. This is curious to say the least, when even toy optimization problems take from minutes to hours to converge on high performance workstations. Parallel value-passing floating point procedures are inefficient to simulate on conventional or even "highly specialized" digital processors. Furthermore, the complete serialization of constraint relaxation presents convergence problems unrelated to the intrinsic dynamics of the problem. Above all, **this view simply ignores the distinction between processing and learning**. It fails to recognize that, if one treat the learning tasks off-line, the **network dataflow** paradigm is a **choice candidate for massive "intelligent arrays" of adaptive logic.**

A GENERAL PROCESSOR ARCHITECTURE FOR CONTROL APPLICATIONS

The concept is illustrated on Figure 1. The model is **hierarchical** and **divides labor between parallel and sequential hardware resources.**

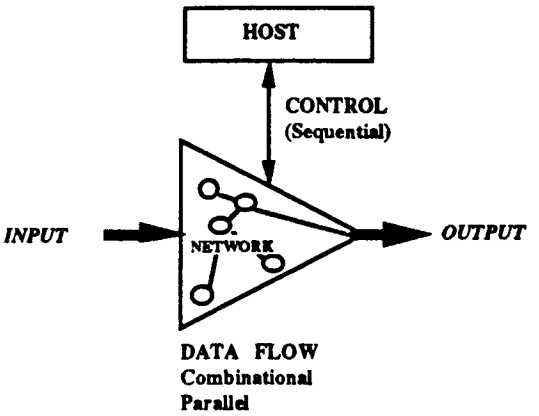


Figure 1

This is a natural division that exploits the dichotomy between ***execution*** i.e., the processing of input data which can be implemented as a feedforward and parallel dataflow (possibly pipelined) from input sensors to effectors and network ***learning*** or ***loading*** which consists of transferring knowledge into the networks Learning requires some inherently sequential global procedures and should be orchestrated by sequential processors. Furthermore these learning functions which include initialization, programming and parameter adjustments during empirical training, operate as ***control*** **processes** that are architecturally **orthogonal to the processing of data** proper. Control functions are inherently sequential, require memory and are best performed by conventional processors, centralized or distributed. A **central host** is needed to provide the necessary **human interface.**

A more detailed view of the structure is illustrated in Figure 2. It provides a realistic mechanism for integrating artificial neural networks with massive parallelism into an end-to-end processing architecture.

The intrinsically parallel elements are concentrated into a **low-level kernel of artificial neural networks** that are responsible for the most computation intensive operations of the execution phase. These operations are *local* within the networks and operate directly on the flow of sensor data. The networks operate entirely on digital principles and interface seamlessly with the sequential processors.

The parallel dataflow traverses two main *stages* dominated by two different types of network elements.

On the **input side,** a **Pattern Processing Stage** groups one or several networks that execute distributed processes such as **coarse encoding of analog data,**

feature extraction, pattern recognition and, wherever applicable, **constraint relaxation** and **optimization**. All these operations are natural problems for Artificial Neural Networks using thresholding elements. The match is owed to the nearest-neighbor and generalization properties afforded by threshold logic .

On the **output side,** a **Decision Stage** converts intermediate data into effector actions and provides an insertion point for human supervision. The networks there are performing as **flexible Boolean decision trees** with multiple inputs and outputs. They can be implementation in hardware with networks of **Programmable Logic Elements**

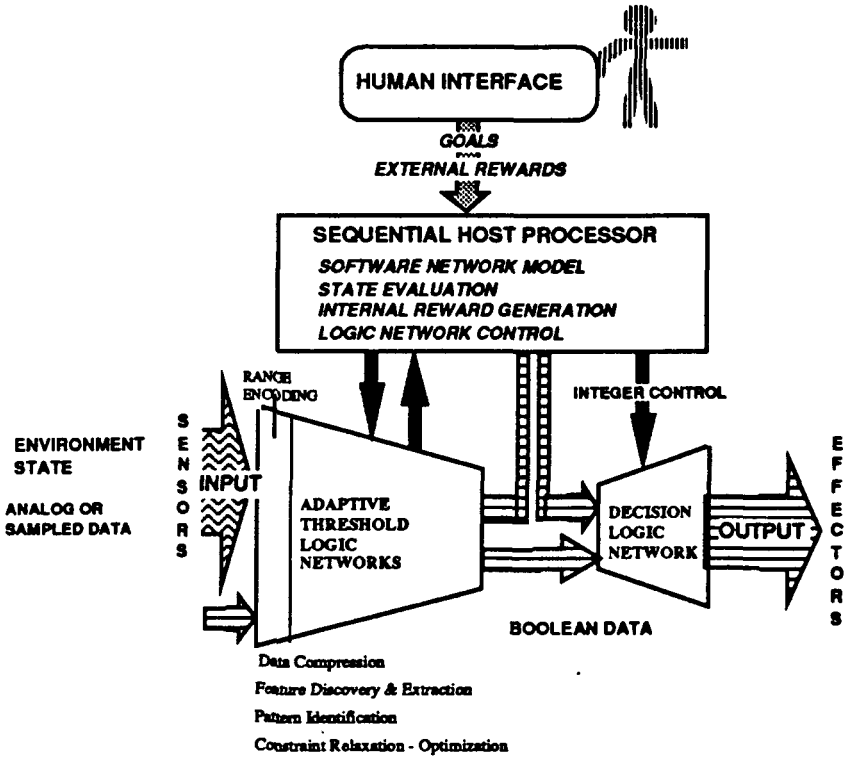


FIGURE 2

Overview of Neural Network Architecture

The constituent processors or ***nodes*** are therefore a mix of discrete **Adaptive Threshold Logic** and of **Programmable Logic** elements. They **do not follow** the usual **"value passing"** paradigm favored by most current research on Neural Networks.. Indeed, with the exception of analog sensors, inputs and outputs as well as intermediate variables appearing on internal connections are **Boolean logical variables.** Hence, and at least when viewed from outside, the network nodes are performing **combinational logic operations.** Internally, they require simple integer arithmetic and logic operations. Local memory requirements are minimal. Because of maximum parallelism and

absence of floating point operation, the network kernel would be capable of extremely fast computation.

The entire system relies on **digital rather than analog** machinery. The hardware components needed for assembling a prototype are beginning to appear commercially. In addition, and because effective network topologies may be achieved with **modular** and **testable,** arrangements of elements, VLSI and perhaps wafer-scale integration of the network sub-systems appear very promising.

RESEARCH RESULTS ON ANN STRUCTURES FOR REAL-TIME APPLICATIONS

The Pattern Processing Stage: Adaptive threshold logic

We have developed a "weight-free" approach to adaptive threshold logic networks that provides a flexible building block for feature extraction and relaxation function in the integrated system. The model was developed around a general problem of constraint satisfaction and successfully illustrated by a classical example. Its extension to the *execution* phase of feedforward networks that learn is the object of the current effort. For such applications, the processing of activation values needed for learning would be decoupled from the network and delegated to the host process.

From a practical point of view weight-free networks are parallel arrays of extremely simple processors that should lead to very efficient VLSI implementations. Two specific node designs have been formulated and are being evaluated. Local relaxation is maximally parallel, involving for each situation only a small and predetermined number of internally controlled iterations. Overall data propagation could be synchronous or asynchronous.

When a random bias is introduced as an additional term in the node activation, the Weight-Free network becomes a digital engine for stochastic relaxation that resembles the Boltzmann machine but that, unlike the latter, is internally and externally digital and executes in constant time. The general weight-free model has been introduced in

Le, N. and **Vidal J.J.** *"Weight-Free Relaxation"* IJCNN 1989. This paper is attached to the present report. See also [LE88].

The Decision Stage: Programmable Logic

In the decision stage, the adaptive logic network functions as an embodiment of logic propositions. This stages serves to insert human decisions transmitted from the host into the data flow.

We have published several papers on programmable logic. An overview appeared in the IEEE transaction and is attached in the appendix.

Vidal, J.J. "Implementing Neural Nets with Programmable Logic", IEEE Trans. ASSP, Vol. ASSP-36, no. 7, July 1988 .

Other papers are found in conference proceedings and in particular to the Annual International Conference on Neural Networks and the IEEE Neural

Network Conference [VIDA83, 87a]. Application of the programmable logic methodology to problems in machine vision is found in [VIDA 85b, 87a, b ,d]
We have embedded programmable logic networks in several designs for self-maintaining logic arrays. The arrays can be prescriptively and incrementally programmed with Prolog like expressions. This technology is directly application to the decision stage in the integrated architecture
The concept was first presented in a journal paper (attached to the present report):
Martinez, T. and Vidal, J.J., *"Adaptive Parallel Logic Networks," Journal of Parallel and Distributed Computing, Vol. 5, no. 1, (February 1988).*
Subsequently, a variant of one the algorithms was studied for VLSI realizability resulting in an experimental chip which is currently being tested.
Chang, J. and Vidal J. J., *"Inferencing in Hardware", MCC-University Symposium, Austin, Texas, July 87, Revised 1989..*

RESEARCH RESULTS ON ANN ALGORITHMS FOR REAL-TIME CONTROL

The learning protocols and algorithms developed around neural networks fall into two very general classes. The first is that of **explicitly supervised** protocols that rely on **error correction**. This approach is found in a substantial part of the published work on neural nets. The other class is often called **unsupervised** even though **implicitly supervised** would be more accurate. Supervised learning, narrowly defined means that explicit target values are known and used to calculate error signals. However, there can be no learning without some *feedback* , a reward or penalty to enable adjustments and produce "better" results in the future. One variant of this type of implicitly supervised paradigm is called **drive reinforcement.** Under this type of learning rule, network elements independently pursues a goal based on the current values and recent history of locally observable variables
We have been exploring an original drive reinforcement protocol for learning empirical control laws for unstable platforms, referred to as **state recurrence learning.** The procedure takes as input a subset of the state variables from the platform and from the environment, pools compact sets of points in the state space into "boxes" using range coding and adaptively binds output decisions (i.e. control actions on effectors) to each box. In the integrated architecture, the procedure would control subnetworks in both the pattern processing and the decision stages. Preliminary results have demonstrated the validity of the algorithm in comparison to a classical benchmarks in the neural control of unstable dynamic objects. Currently, a realistic computer model of the F-15 aircraft dynamics serves as experimental platform for the research.
(**Rosen, B., Goodwin J.M.** and **Vidal J.J.** *"Learning by State Recurrence Detection" Proc. IEEE Conf. on Neural Information Processing Systems Natural and Synthetic, Denver, Co. AIP Press, 1988)*
This paper is attached in the appendix.

A detailed discussion of Unsupervised Learning in the context of real-time control will be found in the next section.

VISUALIZATION: PARALLEL COORDINATES VERSUS RETURN MAPS

Human interface issues are intimately bound to proper visualization of phenomena. This is a particularly challenging problem with multidimensional and highly nonlinear systems since visually understandable renderings must remain simple enough to be grasped at once and since practical displays are limited to two dimensions.

With the F-15 simulation, we initially experimented with ***parallel coordinates*** , two-dimensional nomograms where each state variable amplitude is displayed side by side in some arbitrary order in bar graph form.

Considerations of projective geometry show that in this representation, each point in state space becomes a broken line spanning the entire width of the graph.

Considerations of projective geometry show that when two adjacent variables are linearly related, the fascia of lines in that segment all cross in one point. This features facilitates the detection of pairwise linearity between state variables. Unfortunately the advantage breaks down even with mild nonlinearity, leaving only a drawback, namely the large number of pixels claimed by each single point in the trajectory. **Parallel Coordinate representations** appear to be **of limited use** for the **monitoring of complex trajectories.**

An alternate approach is to use variants of **Poincare's Return Maps .**

Return maps are loci of points of intersection of the trajectory with a plane. The position of the plane is arbitrary but chosing its position is akin to choosing a point of view. Some positions can produce vastly more informative displays than others.

In contrast with parallel coordinates, **Poincare maps compact information** and replaces the **continuous time trajectory** with a **discrete-time mapping** of individual points. The depicted figure share topological properties with the flow that generates it. Dissipative epochs show contracting maps and vice-versa. Global conservation of flow in the sense of state recurrence, although not Hamiltonian, would reveal bounded trajectories and characteristic boundary shapes.

DIRECTIONS FOR CONTINUING RESEARCH
UNSUPERVISED LEARNING SYSTEMS FOR REAL-TIME CONTROL
THE STABILITY DILEMMA

Overview

The new demands made by real time control of high performance vehicles require a radical revision in control methodology. Traditional methods of control rely on the *maintenance of stability,* using known system dynamics.

The objective, holding selected system variables within small ranges in the presence of statistically random noise, has been well served by these methods. If there are large, unpredictable, and rapidly varying inputs, results are less satisfactory. Control of unstable mobile systems and response to hostile action are rarely addressed. We suggest that methods based on unsupervised learning and on the chaotic or fractal behavior of nonlinear systems may be essential in addressing these issues.

This section discusses our intended continuing research on real-time control methodologies appropriate for the architectures presented earlier. The material presented will be part of a forthcoming book chapter:

"Unsupervised Control Exploiting Instability and Chaos" by *James M. Goodwin, Bruce E. Rosen and Jacques J. Vidal.*

Evolution from analytical to empirical control methods

The traditional methods for designing controllers require a mathematically tractable model of the controlled system and, to some extent, of its environment. Traditional control theory has dealt mostly with damping dominated systems with known equations and slowly varying or small inputs. Small errors in the model could often be neglected. However, with the emergence of high performance systems performing demanding missions in hostile and/or rapidly varying environments, control must often be maintained at (or beyond) the margin of stability. Errors are less tolerable, and can have catastrophic results. Yet, at the same time, it is much more difficult to obtain tractable equations that accurately predict the behavior of systems of this type.

The missions have also become more demanding in both civilian and military use. High performance vehicles, such as aircraft subject to hostile actions by adversaries, often would require evasive actions to be formulated and executed at a rate beyond the capabilities of human pilots. Often, the use of predicable dynamic trajectories must be avoided to escape detection. Automatic control systems operable under these conditions are needed, and their man-machine interface must be augmented dramatically. However most of these crucial issues are outside the scope of traditional approaches.

While there is often a requirement need for fast response, there can be another need, almost antithetical to it. Since control actions may be irrevocable and may not be correctable for long (compared to system characteristic times) periods of time, control strategies consistent with minimal intervention are needed.

New methods, empirical rather than analytic, have been considered since the appearance of artificial intelligence on the computer science scene. Despite the empirical nature of these methods, the mathematical theory of nonlinear systems, currently an extremely active field, is providing new tools for their study. The methods do not rely on analytic solutions, and do not require models that are known *a priori*. Instead, they involve automatic learning techniques and automated decision making.

There are, currently, two divided directions in implementing empirical approaches to automated decision making. The *first* uses efficient rules dictated from outside the system by some assumed all-knowing source and they are epitomized by the so-called rule based *expert systems*. This is the traditional artificial intelligence approach that relies on directed search through a precompiled rule base. The expert system can also be embodied in hardware, reducing the actual speed of the search.

The *second* approach resides in the field of Artificial Neural Networks. There are still rules, but they are deduced by the system, based on experience or learning. These method rely on distributed computing and can be embodied in part by parallel architectures. Only general principles direct learning, through the use of immediate feedback, as well as delayed rewards and penalties to influence future actions. In turn, these actions may result in the acquisition of new rules.

These methods also have valuable assets other than adaptability to partially unpredictable conditions. Distributed processing can show graceful degradation rather than catastrophic failure when components are damaged, thus being intrinsically fault tolerant. In addition, they can generalize and categorize, to extract the essential features of the inputs, thus correctly reacting to novel inputs.

NEW DIRECTIONS FOR CONTROL METHODOLOGY

These new problem areas require a radical revision in our attitudes and approaches. The former objective of stability about fixed control points may have to be replaced by *control in the midst of instability*. For instance, using *maximal acceptable instability* may enhance performance, agility and stealth. New or improved control methods incorporating these concepts as well as new mathematical analysis and techniques are needed to support improved approaches. Finally, massively parallel processing and improved architectures will be essential. Prescriptively programmed control will no longer be sufficient but will have to be augmented by *adaptive learned control* and at least in part, *unsupervised learning* due to the lack of *a priori* solutions that supervisory learning would require. We will discuss this new generation of control systems in more detail along with new problems to be faced and some suggestions for its implementation.

Methodology for Control

Stability Concepts

The stability of a system state is evaluated by determining its response to small perturbations. A system is in a *stable* state when it responds to a perturbation with a small displacement of its output trajectory followed by an approximate return to the original state. If the system remains there after perturbations the state is stable.

If the stable state is characterized by a single value for each of its variables (degrees of freedom), it is referred to as an attracting point, an absorbing point,

or an attracting fixed point. If the stable state produces periodic repetitions of the values of a state variable, it is referred to as a limit cycle.

According to Lyapunov, a system is considered stable if for every small arbitrary deviation $e > 0$ from a given state there exists a sufficiently small range $d > 0$ of initial conditions such that all trajectories that deviate by less than e from the desired state have initial conditions that are bounded by d.

This definition is general and applies to perturbed systems with or without forcing inputs. Lyapunov's analysis was extended later to systems whose solutions were not known by showing that they are stable if mathematical functions satisfying appropriate criteria could be found.

A system is *unstable* state, on the other hand, when it responds to perturbation by transitions to new states or escaping "to infinity". If, however, we regard infinity as a state, the system can be described as having an attractor at infinity. This attractor is generally stable.

A compact set of states traversed by a system *without repetition* is called a *strange attractor* or *chaotic attractor.* Many if not most nonlinear physical systems are characterized by the presence of chaotic attractors and can produce chaotic behavior for wide ranges of parameters.

Traditional methods of control rely on the *maintenance of stability,* and depend on *a priori* solution of known dynamical equations describing the system, without particular reference to the environment in which the system is embedded. The control strategy is to modify some *control parameters* to hold selected system variables at fixed values, or within small ranges. This is not always appropriate. This approach is rooted in the assumptions and methods of linear system theory. However, nonlinearities, are becoming the dominant characteristic of advanced systems, defeating superposition based methods.

Practical stability

It is useful to consider systems which operate over *finite,* as opposed to the infinite times used theoretically, since, in any real system, "practical stability" during a finite time is sufficient for the completion of any desired task. A finite time analog of Lyapunov stability can be defined for a case without perturbations where, with bounded initial conditions, all trajectories remain bounded during the required period. What constitutes acceptable bounds must of course be decided operationally; for example, walking across a canyon on a tightrope requires more stringent boundaries than using a wider bridge. In such systems, the control task must simply to *maintain the practical stability* between interventions that may be widely separated in time.

Inherently, for some applications, a trajectory may *never* remain very close to the desired state. In particular, if the goal state is unstable (such as *avoidance* of attractors) and fluctuations about that state are not too large, the trajectory may still be acceptable Conversely, if a system moves gradually toward a desired stable state but undergoes large excursions during the approach, it may exceed acceptable limits.

Practical stability is a common control objective. For instance, controlling the temperature of a fluid in a thermal bath with a heating element causes perturbations of the bath temperature and thus of the heater current that in fact *prevent* the attainment of a stable fixed point of temperature. Rather, the temperature wanders erratically around the fixed point. High resolution temperature measurements at points in the fluid would show that in many cases the values do not even vary periodically. The Fourier transform of the temperature "time series" is typified not by a single frequency spike (a fixed point would have a spike at zero frequency), but by a broad band, showing the instability. The desired setpoint (the goal state) in this case may not even be an attractor in the dynamical sense.

Restricting the goal to practical stability has important advantages and ramifications. Even though the system may be unstable or *chaotic*, unstable behavior can often be ignored when the oscillation amplitude is constrained. By contrast, insistence on equilibrium may cause larger fluctuations and possible loss of control.

The diminishing value of stability

Chaotic and fractal behavior occurs routinely in complex nonlinear dynamical systems, including chemical reactions, fluid flow, and mechanical systems. Understanding such behavior can be a key to the ability to control man made and biological systems and, through use in medical intervention, perhaps improve survival from disease or organ malfunction.

Biological systems appear to rely heavily on instability and chaos for survival and development. Chaotic patterns appear in the timing of pacemaker neurons, in the feeding times in birds, in heart beats, and in electroencephalographic signals. In fact, chaos and fractal patterns are ubiquitous in living organisms. This seems not to arise from lack of reliability or computational power, but rather because of the improved flexibility, control, and adaptation provided by such irregularity. Perfectly periodic oscillations may be undesirable for *all* oscillatory processes in physiology.

Recent data [GOLD 89] show that heartbeat in healthy subjects is chaotic, with a broadband frequency spectrum. On the other hand, the heart rate of some patients near death becomes essentially periodic. Periodic behavior that was traditionally thought to characterize health appears to lack needed flexibility. Chaotic sampling of states may be necessary for the heart to maintain its ability to adjust (adapt) to the widely varying needs of the body. The source of the loss of chaos appears not to be muscular, but rather a failure of brain mechanisms. A control task which might be necessary in this case would be to restore and maintain chaotic behavior.

It has been suggested that epileptic seizures are examples of chaotic behavior. The opposite seems to be true. The EEG trace during epileptic seizures, is periodic. The *normal* EEG is irregular and apparently chaotic [RAPP 89]. In a chaotic system, only a few parameters need be altered to move the system into and out of chaos. One may assume that the nervous system has developed

mechanisms to produce such chaotic behavior since *randomness seems too important to be left to chance.*

Inputs of energy, even steady ones, to a physical system may result in spectacularly complex behavior. Fluids heated from below can form cells (Rayleigh-Benard cells) of vortex motion, which can form long tubes which oscillate "unpredictably", but within narrow bounds, for suitable values of its power inputs. Similar behavior can also seen in fluids entrained between cylinders in relative rotational motion (Couette flow).

Chemical systems under continuous stirring (which prevents formation of concentration gradients and at the same time supplies energy) can also show complex oscillations (Belousov-Zhabotinsky reaction) with a broad frequency spectrum. Such behavior patterns are often referred to as *self-organizing.*

Most connectionist models (see e.g. [HOPF 82]) are designed to converge to stable fixed point attractors. Freeman [Free 87], [SKAR 87] points out that convergence to a single point attractor, without a mechanism to escape could be regarded in biological systems as"death" for the system.

The ubiquity of unstability points to the general *futility of seeking stability,* and the importance of control using chaos. Transition from chaotic behavior to traditional set point control can be precursor to death in a living organism.

Revisiting the "Noise" Concept

The classical assumption is that systems need control because of *noise inputs* from the environment. Noise can affect the controlled system, creating random variations in the system state, or can affect the controller, corrupting *measurements* of the system state. In the traditional view noise is regarded as *the* problem to be handled.

Given *known and stationary dynamics,* control can be accomplished by the usual methods of proportional, differential, and integral feedback. The values of control parameters are derived from prior knowledge of the system dynamics. The control systems have characteristic time constants, deemed important *only* because of the presence of noise (and the occasional need to change set points). Without noise, and given sufficient time, the controller would in theory bring the controlled variables asymptotically to the chosen values.

Again traditionally, noise is assumed to be *statistical* and to produce effects based on the "filtering" characteristic of the controlled system. When the external input is *not* stochastic, i.e. when it is due to interaction with other systems (external drive), the prevailing practice *analyzes* the external input, give it a mathematical form, and incorporates it in the state equations, in order to determine the control parameters.

Since their effects would be difficult to calculate analytically, traditional control theory often deals with *small* variations in the environment as stochastic noise. However, *even small* variations can cause large difficulties if their frequency components match system or controller time constants. *Nonlinear* systems can produce values which *appear* the same as noise when

evaluated only by statistical means, but are actually totally deterministic, but chaotic. Nonstatistical methods would be required to recognize and deal adequately with this situation.

In summary, the role of "noise" in control applications must be reassessed. Rather than being a problem, *noise may be a much needed asset* in probing the dynamics and detecting changes, allowing adaptation. This requires that the system not be too tightly controlled to provide a basis to allow an adaptive system to generalize. Such systems are better equipped to deal with partially unfamiliar situations. Indeed, injecting noise (or chaotic values) has been shown to be beneficial to train artificial neural nets.

Control Exploiting Instability

Chaotic systems.

There has been a recent surge in interest in the dynamics of nonlinear systems, due in large part to the increased power of contemporary computers. The occurrence of instabilities in the form of limit cycles and chaotic or fractal behavior in nonlinear dynamics is now well known. (See e.g. [DEVA 88] for the mathematical background.) The dynamics of continuous differential systems and those of iteration are closely related. Here we primarily consider discrete iterations since they are more closely related to computations.

The simplest nonlinear map is the so-called *logistic map* given by

$$x \leftarrow b\,x\,(1-x)$$

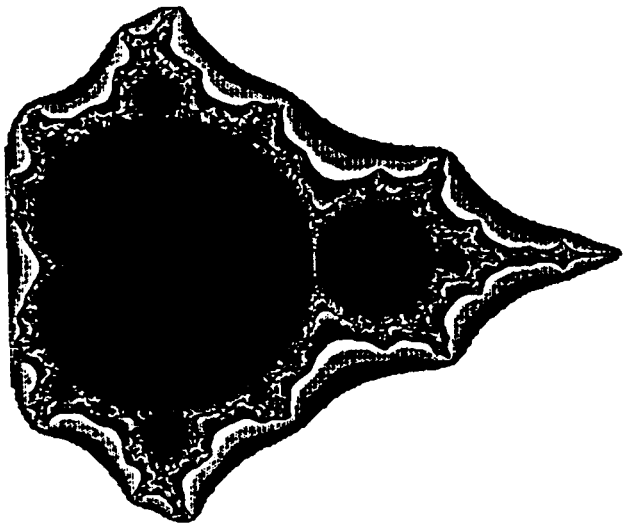
In this equation b is called a drive parameter (or sometimes an order parameter or control parameter). As b is increased, starting from 0, the values of x resulting after repeated iterations show drastic qualitative changes, called bifurcations. For $0 \pounds b \pounds 3$, iterations lead to single fixed points which depend only on b. For $b > 3$, limit cycles are produced. Chaos results for larger values of b. The one dimensional nature of this system makes it easy to study, and leads to substantial insights to the qualitative behavior of nonlinear systems.

A simple extension of this map is to allow the variables to be represented by complex numbers. This leads to the iterated map studied by Mandelbrot [MAND 82]:

$$z \leftarrow (z * z) + c$$

where $z = x + i\,y$ and $c = x_0 + i\,y_0$

What has become known as the Mandelbrot set contains those complex numbers c such that z goes to a bounded limit after repeated iterations of this equation starting with $z = 0$. Values of c "outside" the Mandelbrot set produce arbitrarily large absolute values of z ("escape to infinity"). The boundary between the two *attractor basins* is very complicated and is described as *fractal,* a word coined by Mandelbrot. The regions in the Mandelbrot map depict the transitions from order to chaos, and the fractal boundary.



The Mandelbrot map: Black areas indicate fixed attractor states. States in other areas diverge. Similarly shaded regions require similar number of iterations for divergence

A second way of looking at the iterations is of considerable interest for control theory. For a given value of *c*, iteration beginning at some values of *z* diverges to infinity (*escape*) while iteration from other *z* values leads to finite limiting values (*converge*). The *z* values forming the boundary between these regions, form a set called the *Julia Set* of the transformation, characteristic of the value of *c* used. The boundary formed by the Julia set is not generally continuous but is fractal for all but the simplest cases; it is a generalization of the Cantor set. Points "outside" the Julia set (outside is not really a well defined concept here) escape to infinity. Points near the Julia set take many iterations to escape, while points far from it escape after only a few iterations. Points which are members of the Julia set itself neither diverge nor escape after an arbitrarily large number of iterations.

All of the points just described, except the fixed point limiting values themselves, are unstable. However, points *near* the Julia set (we call them *precatastrophic*) are less unstable than those far away(*catastrophic*), in the sense that they require many iterations before the disaster of escape occurs. On the other hand, very near the Julia set it is easy to find points, e.g. on filaments extending outward from the Mandelbrot set, which lead to some limiting value. These also may be regarded as undesirable; a vehicle governed by the equations would *crash* to the limiting *z* if started from such points.

Cruising the Julia set for failure avoidance

A possible control goal is for the system to avoid either *crash* or *escape* for as long a time as possible, the problem of failure avoidance. In this case a solution would be to "cruise the Julia set". Unfortunately, membership in the Julia set is an undecidable problem [SMAL 89], so it is not possible to perfectly locate or to produce an analytic strategy to remain in the Julia set of

the system's dynamic boundary. Furthermore, the dynamics may change, thus changing the Julia set. The system must then *learn* how to *recognize* the suitable (precatastrophic) regions with sufficient time to modify its own actions so as to remain near the Julia set.

A possible control approach is to use a gradient ascent scheme in *iteration space*. Computation of the iterations, especially using massive parallelism, can be faster than typical vehicle response times. The Julia set, characterized by "infinite" iterability, can be *approximately* recognized from data obtained by exploration of the nearby region. This method can be used to compute a response which can carry the vehicle nearer to the set. As the boundary between the two modes of instability is approached, the fractal nature of the space becomes more apparent.

Converging and diverging states become arbitrarily close together as the Julia set is approached, so the gradient ascent scheme may then yield very erratic actions. It would have to be abandoned if the Julia set were approached too closely, to avoid exceeding maximal acceptable instability. The system response to control actions may become similarly ragged making control decisions more difficult. This suggests that they be addressed by table lookup methods. Using these, identical control decisions would be made for finite ranges of (the chaotic) input data, as long as the behavior stayed within acceptable bounds.

Of course, the Mandelbrot map is only illustrative of the dynamics to be encountered, but the approach would still be applicable if relevant parts of the map could be deduced from the data. As mentioned above, this is a task to which neural nets are applicable.

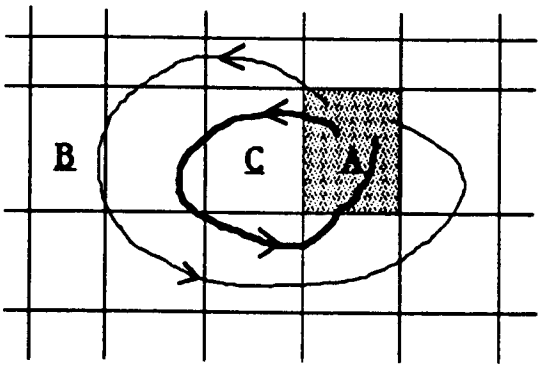
Reinforcement Learning -- Interpretation via chaos

In earlier work we have investigated problems, particularly of failure avoidance, where the nature of the environment or the dynamics of the controlled system is unknown. We have demonstrated a control strategy which need not cause the system to be made stable. Rather, a form of unstable controlled behavior in which cyclicreturn to previously visited points in the system state space are reinforced. Although the system is continually diverging, the loss of balance is reversed in time by the controller. The approach, referred to as "State Recurrence Learning" (SRL), can produce long term failure avoidance. .

The recurrence learning algorithm [ROSE 88] is a nonlinear reward-penalty method in which a state is reinforced when it is *revisited* after the system has wandered through the other states Reinforcement is mediated by the difference between the current time and its last activation time. Small differences are reinforced more than large differences. Furthermore, to insure exploration, choices of action in a given state are not entirely deterministic.

Negative reinforcement can be applied to discourage precatastrophic-to-catastrophic transitions, as is done on failure, while catastrophic-to-precatastrophic transitions are positively reinforced. Thus, a control map

develops that associates learned control actions with states that produce trajectories which avoid catastrophic areas of the system state space.
When an an effective control strategy has been achieved, the system continually makes transitions from one precatastrophic area of the state space to another, never becoming catastrophic (escaping) nor converging to a fixed point (crashing).



The State Recurrence algorithm: Decisions made in state A are reinforced when it is revisited. Reinforcement is larger when the path traverses the states shown by the dark heavy line through state C rather than by the lighter line through state B.

We illustrate by considering, again as an example, the dynamics of the Mandelbrot map. The approach taken by recurrence learning can be thought of as a procedure for probing points outside of Mandelbrot set (diverging) to determine whether a region is catastrophic (rapidly diverging) or pre-catastrophic (not rapidly diverging yet). The action selected by the controller then creates new initial conditions, making transition to other precatastrophic points more likely, and transitions toward catastrophic points less likely. Reinforcement learning may thus be regarded as a method of perturbing the drive parameter(s) (in *this* case the *c* value) to modify the *z* trajectory, pushing it towards the boundary of the Mandelbrot set, i. e. towards the Julia set.

Learning to control a dynamical system in these conditions may be described as adjusting parameters to move the system from one point outside its "Mandelbrot Set" to another, while avoiding catastrophic trajectories. Each control decision can be seen as an iteration of the dynamics. This topological view regards all possible failure modes as a single failure state with an associated attractor basin. Points to which iterations of the dynamics could converge are also attractors with their own basins. There may be other attractors in the system by virtue of the control process and the system dynamics. The separatrix between the basins forms the Julia set of the dynamical transformation, usually fractal. The study of iterated maps in the context of control problems appears to be a fruitful and as yet hardly tapped source of insight.

More specifically, learning the control dynamics of a system is in this case equivalent to mapping the exterior of its Mandelbrot set and the Julia set. This is done by first partitioning the entire state space into individual "boxes" and sampling points within each box. Each iteration causes a transition from one point to the next, with or without transition to another box. The decision made at each iteration chooses an action that moves the system to a new point nearby. Transitions are determined by the algorithm's control decision and the dynamics, including the environmental feedback. This perturbing action allows the stochastic probing of points. Learning in failure avoidance algorithms seek to maximize the number of iterations of the control dynamics. After sufficient points are sampled, a learning system effectively maps its own Mandelbrot set.

The task of control can be seen in two different ways: *First* as the attempt to avoid the attractor basin of the failure state(s), using tests which allow the controller to recognize and enter the basins of other attractors, such as limit cycles. *Alternatively,* it is possible to *dynamically modify* the control strategy *even while in the basin of the failure state* to avoid falling too deeply into the basin. The rapid determination of the onset of divergence, and the utilization of nonstable control strategies require further study.

Implications of finite resolution or discreteness

Since measuring equipment is discrete, the values of the state variables are never known precisely; infinite precision cannot be achieved. Empirical methods generate systematic lack of precision by operating with *ranges* of variables. There are two interpretations to this phenomenon. The *first* interpretation regards the uncertainty of the state variables as a source of noise which affects the iterated control map. There seems to be little work in this area at present.

The *second* interpretation uses the range size to find fractal dimensionality "on the fly" without the need to sample the entire space. Fractal dimension can be determined from the increase in size of the phase space region occupied by the system. The size of the range is regarded as the beginning size, and the size of the phase space occupied after control actions determined, by sampling or simulation. If good estimators of the fractal dimension, even in the midst of chaos, can be found, the dynamic reduction of this dimension can be used as part of a control strategy, like a high order version of gradient descent.

Generality of the methods

It should be noted also that there is nothing in these approaches which is restricted to state recurrence learning or to failure avoidance problems. The mathematical or numerical study of nonlinear and nonlocal dynamical systems can give profound new insights to a wide class of problems.

References

[GOLD 89] Goldberger, A., "Fractals, Chaos, and Sudden Cardiac Death, AAAS annual meeting , 1989.

[HOLL 75] Holland, J., "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, 1975

[HOLL 85] Holland, J., "Properties of the Bucket Brigade Algorithm," *Proceedings of an International Conference on Genetic Algorithms and their Applications*, July, 1985, p. 1.

[KLOP 82] Klopf, H., *The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence*, New York: Harper & Row/ Hemisphere, 1982.

[KLOP 86] Klopf, H., *A Drive Reinforcement Model of Single Neuron Functions: An Alternative to the Hebbian neural Model*, in J. S. Denker (ed.) *Neural Networks for Computing*, AIP Conference Proceedings 151, New York: American Institute of Physics, 277-281, 1986.

[KLOP 87] Klopf, H., *Drive Reinforcement Learning: A Real-time Learning Mechanism for Unsupervised Learning*, in Proceedings of the First Annual International Conference on Neural Networks, 1987.

[LAPE 87] Lapedes, A., et al. in "Neural Information Processing Systems" , James Anderson (ed.), AIP,(Nov. 1987).

[MAND 82] Mandelbrot, B., The Fractal Geometry of Nature, Freeman (1982) San Francisco.

[MICH 68] D. Michie and R. A. Chambers, "BOXES: An experiment in Adaptive Control," in *Machine Intelligence*, E. Dale and D. Michie, Ed.: Oliver & Boyd, Edinburgh 1968, p. 137.

[NARE 74] Narendara- IEEE SMC 4,4 July, 1974

[RAPP 89] Rapp, P., "Dynamical characterization of human electroencephalographic signals", AAAS annual meeting , 1989.

[ROSE 88] B. E. Rosen, J. M. Goodwin, J. J. Vidal, Learning by State Recurrence Detection", in "Neural Information Processing Systems" , James Anderson (ed.), AIP, p. 642, (Nov. 1987).

[RITT 88] Ritter, H., and Schulten, K. "Extending Kohonen's Self-Organizing Mapping Algorithm to Learn Ballistic Movements" in R. Eckmiller (ed.), Neural Computers, Springer-Verlag, 1988.

[RUME 86] Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "Learning Internal Representations by Error Propagation", p. 318-362 in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA (1986).

[SATO 84] M. Sato, K. Abe, and H. Takeda, "A Learning Algorithm for the Finite-Time Two-Armed Bandit Problem," in *IEEE Transactions on Systems, Man, and Cybernetics*, **14** (3), p.528, May 1984

[SHEP 87] J. F. Shepanski and S. AS. Macy, "Manual Training Techniques of Autonomous Systems Based on Artificial Neural Networks," First International. Conference. on Neural Networks, Vol. **4**, p. 697 (1987).

[SKAR 87] Skarda, C. A. and Freeman, W. "How the brain makes chaos in order to make sense of the world," Behavioral and Brain Sciences **10**, p. 161 1987.

[ZIPS83] Zipser, D. "The Representation of Location", Technical Report 8301, Institute for Cognitive Science, University of California, San Diego 1983.

PUBLICATIONS DURING GRANT PERIOD

The following section lists all the publications by the Distributed Mchine Intelligence research team during the Grant period (1986-88). Support by NASA under the NAG 2-302 Grant has been aknowledged for most of the publications. **Four key papers** (listed in bold face) **directly relevant to the work under this grant**. .have been **attached as appendices** to this report.

BERK87a Berke, P., and Vidal J.J., "Adaptive Menus: An Adaptive Interface With Minimal Overhead ", Proc. Second International Conference on Human-Computer Interaction, Honolulu, Hawaii, August, 1987.

BERK87b Berke, P., That Does Not Compute, e.g., Neural Networks " Proceedings, IEEE First International Conference on Neural Networks, San Diego, California, June 1987.

CHAN87a Chang, J. and Vidal J. J. , "Inferencing in Hardware", Proc. MCC-University Symposium, Austin, Texas, July 87.

CHAN87b Chang, J. , "Adaptive Boolean Networks", M.S. Thesis, University of California, Los Angeles, CA (1987).

GOOD87 Goodwin, J., B. Rosen, and J.J. Vidal, "A Design for an Associative Spin Glass Processor" , Proc. First Annual International Conf. on Neural Networks, San Diego, (June 87).

GOOD88a Goodwin, J., B. Rosen, and J.J. Vidal, "Exploration of Learning in an AssociativerMagnetic Processor", Proc. IEEE ICNN 88, San Diego, July 88 .

GOOD88b Goodwin, J., B. Rosen, and J.J. Vidal, "Progress on the Spin Chip Associative Processor", Proc. Int'l Neural Network Society 1988 Annual Meeting, Boston, MA., Sept. 1988, .

GOOD89 Goodwin, J., B. Rosen, and J.J. Vidal, "An Associative Spin Glass Processor for Image Reconstruction", IEEE International Workshop on Industrial Appliations of Machine Intelligence and Vision (MIV-89), Roppongi, Tokyo, Japan, April 10-12, 1989. (accepted for presentation)

LANG89 Lange, Trent E.,Vidal J. J.and M.G. Dyer, , "Phase-Locking of Artificial Neural Oscillators Can Perform Dynamic Role-Binding and Inferencing", IEEE/INNS International Joint Conf. on Neural Networks (IJCNN-89) , Washington, D.C., June 18-22, 1989. (submitted)

LE88 Le, Nhan T and Vidal J. J., "Weight-Free Binary Relaxation Networks", UCLA CS Dept. Report,, 1988.

LE89 Le, Nhan T and Vidal J. J., "Weight-Free Relaxation", IEEE/INNS International Joint Conf. on Neural Networks (IJCNN-89), Washington, D.C., June 18-22, 1989. (submitted)

MART86 Martinez, T., "Adaptive Self-Organizing Logic Networks," Ph.D. Dissertation, University of California, Los Angeles, CA (1986).

MART87 Martinez, T. and Vidal, J.J., "Adaptive Parallel Logic Networks," Journal of Parallel and Distributed Computing, Vol. 5, no. 1, (February 1988).

PEMB88a Pemberton, J. C., and Vidal, J.J. "When is the Generalized Delta Rule a Learning Rule? A Physical Analogy", Proc. IEEE ICNN 88, San Diego, July 88.

PEMB88b Pemberton, J. C. and Vidal, J.J. "Noise Immunity of Generalized Delta Rule Learning", Proc. Int'l Neural Network Society 1988 Annual Meeting, Boston, MA., Sept. 1988.

PEMB88c Pemberton, J. C. "Geometric Representation of Learning in Threshold Units", UCLA CS Dept. Rept. No. 890006, 1988.

PEMB89 Pemberton, J. C. , M. Ercegovac and Vidal J. J., "Analysis of a Dataflow Architecture for Implementing NETtalk", IEEE International Conf. on Computer Design: VLSI in Computers & Processors (ICCD), Cambridge, MA., October 2-4, 1989. (submitted)

ROSE87 Rosen, B., Goodwin J.M. and Vidal J.J. "Learning by State Recurrence Detection" Proc. IEEE Conf. on Neural Information Processing Systems Natural and Synthetic, Denver, Co. AIP Press, 1988.

ROSE88a Rosen, B.E., Goodwin, J.M., Vidal, J.J. "State Recurrence Learning". Proc. Int'l. Neural Network Society 1988 Annual Meeting, Boston, MA., Sept. 1988,

ROSE88b Rosen, B.E., Goodwin, J. M., and Vidal, J.J. "Machine Operant Conditioning". Proc. IEEE Engineering in Medicine and Biology Society 10th Annual Int'l. Conf., New Orleans, Nov. 1988.

VIDA85b Vidal, Jacques J. and Philip Kahn, "Evaluation of the Triad Method for Dynamic Edge Detection," Proc. 2nd Int. Tech. Symposium Optical and Electro-Optical Applied Science and Engineering, Cannes, France, (November 1985).

VIDA87a Vidal, J.J. , Pemberton J.C. and J.M. Goodwin, "Implementing Neural Nets with Programmable Logic , "First Annual International Conf. on Neural Networks , San Diego (June 87).

VIDA87b Vidal, J.J. and P.F. Salas, "Processing of Moving Edges with Pyramid Networks of Programmable Logic Elements," Technical Digest Series Vol. 12, Optical Soc. Topical Mtg. on Machine Vision , Lake Tahoe Nevada, (1987).

VIDA87c Vidal, J.J. and J. Haggerty, "Synchronization in Neural Nets", Proc. IEEE Conf. on Neural Information Processing Systems, Natural and Synthetic, Denver, Co. (Nov. 87) AIP Press, 1988.

VIDA87d Vidal, Jacques J. and Salas, Paul F., "Real-Time Parallel Processing of Pixel Arrays Using Adaptive Logic", Proc.1987 Int'l. Tech. Symposium Optical and Electro-Optical Applied Science and Engineering , Cannes, France, (November 1987).

VIDA87e Vidal, J.J. , "Emulating Conditioned Reflexes: Neural Networks as Combinational Processors", Proc. IEEE Conf. Engineering in Medicine and Biology, Boston Ma., (Nov. 87).

VIDA88 Vidal, J.J. "Implementing Neural Nets with Programmable Logic" , IEEE Trans. ASSP, Vol. ASSP-36, no. 7, July 1988 .