

A TRIBUTE AND TWO PAPERS PUBLISHED FOR THE FIRST TIME · OCTOBER 2026

Listening to Living Signals

The work of Dr. Charles C. Jorgensen

Chief Neuro Scientist, NASA Ames Research Center

Neural flight control, bioelectric interfaces, silent speech, and computing the way living systems do: a career, and the papers it left unpublished

Part I A life spent listening to living signals · edited by Bryan M. Ingram

Part II Analysis of Pilot Scheduling Using Recurrence Analysis · C. C. Jorgensen and S. Johan

Part III Oscillating Field Computation · Chuck Jorgensen

Parts II and III are published with the author's permission, from his own files, with his wording unchanged. Part I is compiled from the public record. bryaningram.com

Foreword

Some people change the world loudly. Dr. Charles Jorgensen, Chief Neuro Scientist at NASA Ames Research Center and Chuck to everyone who has worked beside him, has changed it the other way: in laboratories, in technical reports, in patents assigned to the people of the United States, and in ideas that arrived a decade or two before the rest of us were ready for them. Pilots have flown safer aircraft, people who cannot speak have been given a glimpse of a voice, and a whole field of research on how to talk to machines without a keyboard has grown, in part, on ground he cleared. Most people who benefit from that work will never know his name. This volume is an attempt to put it where it belongs.

It does two things. Part I tells the story of his career as the public record has it, and tries to say plainly what that work has given to our society and, in the long run, to the planet and to our species. Parts II and III publish, for the first time, two papers he wrote at NASA Ames Research Center and never released: a recurrence analysis of airline pilots' heart rhythms under different flight schedules, written with S. Johan, and a design for a computer built from oscillating fields the way living nervous systems compute. Both are working drafts. Both are, in my judgment, ahead of their time. Dr. Charles Jorgensen has authorized their publication here, and I am grateful to him for trusting me with them.

My own debt to these papers is specific. The methods in the first one gave our recent work its practical half: a way to read the hidden state of a system from a single measured rhythm, the heart included. The ideas in the second gave us a mechanism for why three coupled oscillators settle into a stable pattern.

There is a kind of work that is never finished, because it was never meant to be. It was meant to be handed on.

This book is a small act of listening in return: my attempt to say, in print and for as long as print lasts, that the signal was received.

Thank you for the flights that landed, and the voices that were heard. Thank you for the questions you asked decades before the world knew how to answer them. Thank you for trusting me with work you had not yet let go of. And thank you for being, in the truest sense, a friend to our species.

In the oldest stories the raven is the one who watches, remembers, and carries word between worlds. You have been that for more of us than you will ever know.

*With the highest honors and my deepest gratitude,
from one raven to another,*

Bryan

Bryan M. Ingram · San Juan Capistrano · October 2026

Part I. A life spent listening to living signals

1. The question underneath everything

Read across fifty years of Dr. Charles Jorgensen's work and one question keeps returning in different clothes: what is a living system telling us, and how can a machine learn to listen? He came to it from an unusual direction. His doctorate, from the University of Colorado Boulder in 1973, is in mathematical psychology, a discipline he has described as a strange combination of the soft and the hard sciences: the study of minds and behavior with the tools of mathematics. That combination shaped everything that followed. Where an engineer sees a signal and a psychologist sees a person, he has always seen both at once.

Before NASA he worked across academia, industry and government: at Carnegie Mellon University, the U.S. Army Research Institute, Martin Marietta, Oak Ridge National Laboratory, and Thomson-CSF. He joined NASA's Ames Research Center in Moffett Field, California, around 1990 (accounts give both 1989 and 1990), and there he started the laboratory now known as the Neuro Engineering Laboratory, which he went on to lead as NASA Ames' Chief Neuro Scientist (the Chief Scientist of its Neuro Engineering Laboratory). Over his career he has published more than 150 papers and holds seven patents.

Figure 1. Charles C. Jorgensen: a career in public sources (dates as reported; see Sources)

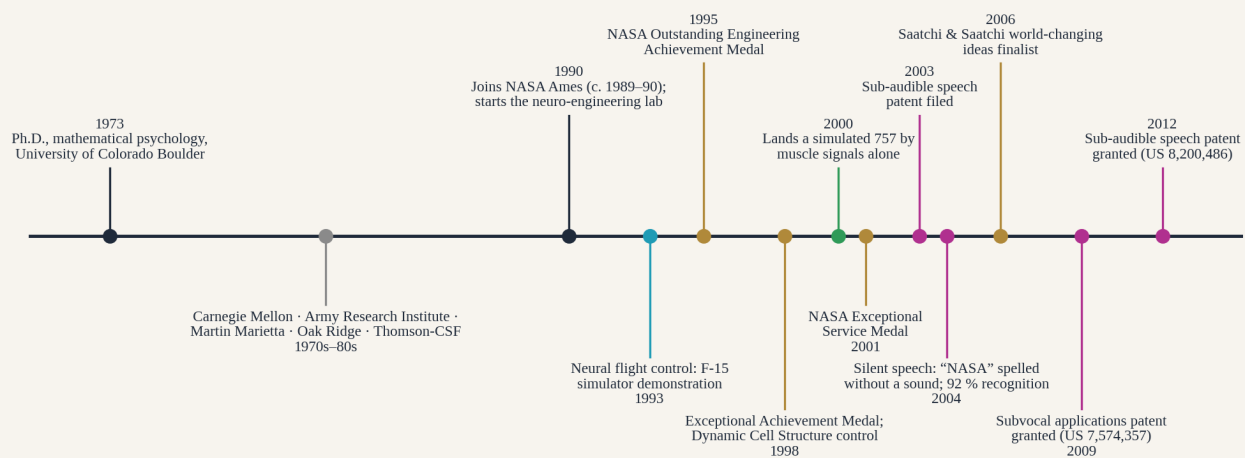


Figure 1. Milestones drawn from NASA, Stanford, patent and press records cited at the end of Part I. Where sources disagree, the range is given.

2. Teaching damaged aircraft to fly

His first great project at Ames was a problem with lives attached. When an aircraft is damaged in flight, by a jammed control surface, a failed hydraulic system or a lost piece of wing, the equations its flight computer was designed around stop being true. The question is whether a control system can notice that and learn the new aircraft fast enough to bring it home.

Dr. Charles Jorgensen saw that neural networks, then still a curiosity in aeronautics, were suited to exactly this: they can approximate nonlinear behavior, learn while running, and reconfigure themselves when the system they control

changes. In the early 1990s he joined forces with Dryden Flight Research Center and McDonnell Douglas on the Intelligent Flight Control program, which demonstrated rapid prototyping of a neural flight controller on a full-scale F-15 simulator in 1993 and pointed toward flight tests at Dryden. He pressed, early and persistently, for networks that learn in real time rather than ones trained in advance, and developed the Dynamic Cell Structure approach for it; NASA's own history of self-repairing flight control records that his ideas began to be implemented in 2001. Along the way his group taught a neural network to learn a full aircraft model from wind-tunnel data in under thirty seconds, with errors well under one percent, cutting the test data needed by about forty percent in the trials reported.

This line of work belongs to the family of research that made it reasonable to think a crippled aircraft could be flown by throttles alone, or by a controller that relearns the airplane in the air. Its benefits are the kind no one notices: the flights that land.

3. Flying a jet with a closed fist

In 2000 his team did something that sounded like science fiction and was reported to the World Automation Congress as engineering. A pilot wearing an armband of eight dry electrodes landed a simulated Boeing 757 at San Francisco International Airport using nothing but the electrical signals of the muscles in his forearm. No joystick; a closed fist in midair. The team combined it with the adaptive neural control and propulsion-only landing work, and the pilot brought the simulated aircraft down through cascading failures, a locked rudder and a total hydraulic loss among them.

“This is a fundamentally new way to communicate with machines — another way to talk with our mechanical world.”

He said at the time that such interfaces could replace keyboards, mice and joysticks for some uses, and help astronauts control tools through the stiff gloves of a spacesuit. Twenty-five years later, wrist-worn muscle-signal interfaces are a product category of their own.

4. Words without sound

Then he turned the same idea toward the most human signal of all. When we read silently or speak to ourselves, the nerves that drive the tongue and larynx still fire, faintly, even with no movement of the lips. In the early 2000s his group placed small, button-sized sensors under the chin and on either side of the Adam's apple, recorded those signals, and taught a neural network to recognize the words they carried. In the first experiments, recognition of six words and ten digits averaged ninety-two percent. In a demonstration in 2004, a user silently spelled “NASA” letter by letter, submitted it to a web search engine and browsed the results, all without a sound or a keyboard.

“This proved we could browse the Web without touching a keyboard.”

He saw its uses at once: astronauts in spacesuits, controllers in noisy airport towers, firefighters and first responders, private conversation on a crowded train, and above all people whose injury or illness has taken their voice. Later work extended the vocabulary toward a thousand words and more. The work was protected in two patents assigned to NASA: U.S. 8,200,486, *Sub-audible speech recognition based upon electromyographic signals*, with Diana D. Lee and Shane T. Agabon (filed 2003, granted 2012), and U.S. 7,574,357, *Applications of sub-audible speech recognition based upon electromyographic signals*, with Bradley J. Betts (filed 2005, granted 2009). In 2006 it made him a finalist for the Saatchi & Saatchi award for world-changing ideas.

Silent-speech interfaces are now an active research field in universities and companies around the world. It would be wrong to credit any one laboratory with all of it, and he would be the first to say so. It is fair to say that his group was among the first to show it working, to put numbers on it, and to say clearly who it was for.

5. Reading what a person feels

Later he turned to the hardest signals of all: emotion, stress and fatigue. Working with voice prosody and spectra, facial expression, and the behavior of the pupils and eyes, his laboratory studied how to infer a person's state, especially that of pilots and air traffic controllers, whose fatigue is a safety problem for everyone in the sky. He was candid about the limits of the field: he has said publicly that voice-stress analysis, as commonly sold, lacks scientific support, and that a credible approach has to look at many variables at once. That insistence on rigor runs through the first paper in this volume, which applies recurrence analysis to pilots' heart rhythms precisely because a single average cannot see the structure of a changing state.

6. Computing the way life does

Underneath the applications is a theoretical conviction he has held for decades: that living systems do not compute the way our computers do. Nerve cells are oscillators, bursters and resonators; they lock into patterns, entrain one another, and store memory in phase. If we want machines that adapt as living things adapt, he argues, we should build them from the same mathematics. The second paper in this volume is the fullest statement of that view he has written: an architecture in which bursters detect, resonators amplify, oscillators remember, and behavior emerges from fields rather than from instructions. Much of what it anticipates, from neuromorphic hardware to oscillator-based computing, has since become a research program of its own.

7. Honors

NASA recognized the work with its Outstanding Engineering Achievement Medal (1995), its Exceptional Achievement Medal (1998) and its Exceptional Service Medal (2001). He has received numerous awards from the American Nuclear Society and the Department of Defense, and was a 2006 finalist for the Saatchi & Saatchi prize. His technical reports list him as a Senior Member of the IEEE. He has presented his work at Stanford's Human-Computer Interaction seminar and in the technical press, and his name runs through NASA's history of self-repairing flight control.

8. The contributions most people never see

It is easy to list medals and patents. It is harder, and more important, to say what a life's work has done for people who will never read a technical report. Here is an honest attempt.

Safety in the air. Every passenger who has landed safely after a failure the crew was trained to handle owes something to the generation of researchers who asked how a damaged aircraft can be flown and relearned in flight. He was one of the people who asked it first and pressed hardest for real-time learning.

A voice for people who have lost theirs. The idea that the body's own electrical signals can carry speech when the voice cannot is now part of how researchers approach communication for people with paralysis, laryngeal injury and degenerative disease. His group showed, early and in public, that it could be done without surgery.

A gentler relationship with machines. Controlling a device with a gesture or a silent word, rather than bending the body to a keyboard, is a different idea of what technology should ask of us. He said in 2000 that this was a new way to talk with the mechanical world, and the world has since moved toward it.

Reading people with care. His work on fatigue and emotional state, and his insistence that it be done with many variables and real statistics rather than gimmicks, is the kind of rigor that protects people from bad science as much as it serves them with good.

A different idea of computing. His long argument that computation can be built from coupled oscillators and fields, as living systems do it, points toward machines that use far less energy than today's and adapt the way organisms do. For a planet whose computing already draws a measurable share of its electricity, that is not an academic preference.

Bridges between disciplines. Perhaps his deepest contribution is the habit of mind itself: treating the mathematics of dynamical systems, the biology of nerve cells and the engineering of aircraft as one subject. The two papers that follow are examples of it. So, in a small way, is the work they have already inspired.

9. A reader's guide to Parts II and III

Part II. Analysis of Pilot Scheduling Using Recurrence Analysis (C. C. Jorgensen and S. Johan)

A study of physiological data, heart rate first among them, recorded from commercial airline pilots across different flight schedules in a joint NASA and EasyJet monitoring experiment. It explains recurrence analysis from first principles, Takens' delay embedding, the reading of recurrence plots and the recurrence quantification measures, and applies them to individual pilots, finding periodic processes that differ between flight and non-flight periods and differ between pilots. It is a working draft: the analysis of the full data set was not finished, and the conclusions are left in outline. Its lasting value is the method and the clarity with which it is taught.

Part III. Oscillating Field Computation (Chuck Jorgensen)

Prepared as an Ames Research Center technical report draft under NASA's CICT/ITSR program, this paper proposes a computing architecture built from canonical nerve-cell models: bursters that respond to complex temporal patterns, resonators that amplify specific signals for a limited time, and oscillators that phase-lock and store patterns as an associative memory. Behavior emerges as "modes," locked regions organized in a hierarchy of time scales, represented as potential fields that drive micro-level controls. It compares the design point by point with conventional computer architecture and discusses implementation.

Editorial note

Both papers are reproduced from the author's own files with his wording unchanged. Tracked changes left in the files were accepted as the author made them. Where the drafts contain notes the author wrote to himself, or placeholders for material not yet supplied, these are shown in square brackets so the reader can see exactly where the draft stood; no missing content has been invented. The appendix of individual pilot results mentioned in Part II was not among the files and is not reproduced. Original page headers and footnotes, including the author's NASA address of the time, are kept as historical record.

Sources

- [1] Stanford University, Human-Computer Interaction Seminar, “Subvocal Speech and the Development of Biological Interfaces,” Charles Jorgensen, NASA Ames, 17 February 2006: speaker biography (Ph.D. 1973; prior positions; NASA Ames since 1990; Chief Scientist, Neuro Engineering Laboratory; 150+ publications, seven patents; NASA medals 1995, 1998, 2001; Saatchi & Saatchi finalist 2006). hci.stanford.edu/seminar/abstracts/05-06/060217-jorgensen.html
- [2] NASA Tech Briefs, interview with Chuck Jorgensen, Chief Scientist, Neuro Engineering Lab, NASA Ames Research Center (NTRS 20130001672): mathematical psychology background; “I started what is now referred to as the Neuroengineering Laboratory”; bioelectric control, subvocal vocabulary growth, emotion and fatigue detection, view of voice-stress analysis.
- [3] Tomayko, J. E., *The Story of Self-Repairing Flight Control Systems*, NASA Dryden Flight Research Center history publication: joined NASA 1989 from Thomson-CSF; started the Ames “neuro” laboratory; partnership on real-time learning neural flight control; implementation of his ideas from 2001; Dynamic Cell Structure nets.
- [4] Jorgensen, C. C., “Neural Networks for Flight Control,” NASA Ames Research Center (NTRS 19960047555): Intelligent Flight Control program with NASA Dryden and McDonnell Douglas; 1993 F-15 simulator demonstration; wind-tunnel model learning in under thirty seconds with about 40 % less test data.
- [5] Jorgensen, C. C., “Feedback Linearized Aircraft Control Using Dynamic Cell Structures,” ISSCI, 1998.
- [6] NASA Ames Research Center press release, 2001, reported in SpaceNews and elsewhere: bioelectric interface used to land a simulated 757 at San Francisco International Airport with an eight-electrode armband (World Automation Congress, October 2000); quotations on a “fundamentally new way to communicate with machines.”
- [7] NASA Ames Research Center press release, March 2004, “NASA Develops System to Computerize Silent, ‘Subvocal Speech’” (SpaceRef; also The Register, Machine Design): sensor placement; 92 % average recognition on six words and ten digits; silent web search for “NASA”; quotation “This proved we could browse the Web without touching a keyboard.”
- [8] Jorgensen, C. C., Lee, D. D., and Agabon, S. T., U.S. Patent 8,200,486, “Sub-audible speech recognition based upon electromyographic signals,” filed 5 June 2003, granted 12 June 2012; assigned to NASA.
- [9] Jorgensen, C. C., and Betts, B. J., U.S. Patent 7,574,357, “Applications of sub-audible speech recognition based upon electromyographic signals,” filed 24 June 2005, granted 11 August 2009; assigned to NASA.
- [10] Jorgensen, C., Lee, D. D., Agabon, S., and Lau, S., “Subauditory Speech Recognition based on EMG/EPG Signals,” International Joint Conference on Neural Networks, Portland, Oregon, 20–24 July 2003 (NTRS 20030018896).

Part I was compiled from the public record listed above; where sources differ, both are given. Corrections and additions are welcome and will be incorporated in later printings.

PART II

Analysis of Pilot Scheduling Using Recurrence Analysis

C. C. Jorgensen and S. Johan · NASA Ames Research Center

A working draft, previously unpublished. Physiological data from commercial airline pilots, recorded under different flight schedules in a joint NASA and EasyJet monitoring experiment, are analyzed with recurrence analysis: Takens delay embedding, recurrence plots, and recurrence quantification.

Reproduced from the authors' file with the wording unchanged. Square brackets mark the authors' own working notes and outline headings, and a detail the draft left unrecorded. The appendix of individual pilot results mentioned in the text was not among the files. See the editorial note in Part I, section 9.

Analysis of Pilot Scheduling Using Recurrence Analysis

C.C. Jorgensen, S. Johan

NASA Ames Research Center

Abstract:

Recurrence analysis (RA) is explored as a method for the detection and quantification of pilot physiological condition under varying flight schedules. Three sets of physiological measurements were explored taken during commercial flights in three blocks spanning 15 days. Subsets of these measurements were mapped into 2-D visual projections using functional Recurrence Analysis (RA). The plots reflected changes in time of underlying but not necessarily directly observable human physiological states changing with workloads and circadian conditions. A state space dimension embedding number for these hidden processes was estimated using global false nearest neighbor values in accordance with the Taken's embedding theorem. A required temporal delay was estimated using the correlation dimension. Three kinds of RA plots were examined to determine if a pilots physiological state changed systematically as a function of flight/non-flight activity, number of flights per day, or time of day. Recurrence Quantification Analysis (RQA) was used to numerically characterize plots along six numerical dimensions and quantify the images for comparisons. The viability of RA visualization and RQA quantification is considered as an approach to study pilot responses and select optimal workload schedules.

Keywords: pilot workload, data visualization, recurrence analysis, speech recognition

Introduction:

An important problem in the study of complex systems is early detection of change. As the number of interacting variables and system states grows, formal mathematical analysis scales poorly (e.g. NP complete or NP hard) and human visualizations in displays become cluttered and highly complex. In large systems where variables number into thousands of states, both visual and numerical approaches to change detection can be overwhelmed. The present paper has human data visualization as its primary goal but the origins of the methodology used derive from tools developed for analysis of non linear dynamical equations and represent in effect a hybridization of both visual and mathematical approaches. To determine whether they provide insight into the difficult task of human pilot workload monitoring in this paper, two well understood tools from nonlinear mathematics are explored (Weber & Zbilut (ref. 13))

To gain insight into human factors data, traditionally, global measures such as means and standard deviations or analysis of variance would be used along with time series plots and summary histograms. With strong effects, such methods are often viable but can become problematic if the data reflects distributed, non linear, temporal, or chaotic patterns. Such messy characteristics often define biological systems and are characteristics of mathematical operators in time domains. As a result there has been a search for many years for optimal ways to understand and describe such phenomena

across many fields. For a while it appeared Chaos theory would be a solution, but although providing powerful insights into the nature of the processes it did not directly address the issue of visualization of data. Many approaches were considered from a graphics standpoint but one approach in particular has stood out in recent years.

The focus of the present paper is on the use of one such technique and its usability for detection of subtle changes in pilot's physiological status. We approach this problem using two theoretical tools from nonlinear analysis, namely Recurrence Analysis (RA), and Quantitative Recurrence Analysis (QRA). We present our first visualization results for three variables collected from commercial EasyJet pilots using a Zephyr BioHarness with OmniSense software, specifically: heart rate, chest temperature, and a ratio (LF/HF) to estimate autonomic nervous system balance (- reference to Ames researchers doing report with Irv ---) .

The paper is organized as follows. We begin with an overview of RA giving an intuitive mathematical foundation for how it creates a 2-D visualization. We follow with a discussion of RQA showing how a visual RA plot can be characterized in terms of summary numerical values. We then apply both techniques to a series of eight EasyJet pilot recordings, generating visual plots of the above selected features along with a discussion of numeric RQA differences and their potential usability for visual pattern recognition of one pilot's physiological change under different flight conditions. An appendix is attached containing more extensive results including plots and summary information for all the pilots having sufficient data.

Recurrence Analysis

The nonlinear analysis technique called recurrence analysis (RA) has proved important for understanding the generating mechanisms underlying subsets of observable variables in a number of fields. Originally created to facilitate the visualization of genetics, the variables of interest are experimentally measured values that reflect but do not always directly measure properties of the entire system actually producing the behavior. An example often used is patterns in Pacific Ocean waves at a given time. A subset of possible measurements such as height or direction of movement can be obtained using a standard experimental design, but all the complex interactions of environment, solar energy, currents, wind patterns or other factors that ultimately create observed wave behaviors cannot. These interactions could be linear but all too often they are non linear or chaotic. With a greater understanding of chaotic system behavior has come the understanding that even seemingly random measurements still have periods of approximate repetition or recurrence. So, though a chaotic process may never repeat itself exactly, it may recur with enough periodic similarity that useful levels of state recognition or control can be achieved. This has led to practical applications such as chaotic control for electrical power demand grids and signal compression techniques.

Finding these patterns of near repetition and describing them in meaningful ways has motivated another related analysis technique, Recurrence Quantification Analysis (RQA). Tools have been developed that can quantify and periodic recurrence areas created using RA plotting such as stretches of long or short repeating states. Because

these patterns occur in time, capturing and describing the global but hidden underlying generating function clearly becomes very important. Fortunately a rather amazing result called the Taken's Embedding Theorem (2) emerged in 1981 that permits just such characterization of processes. The Taken's theorem was built upon an earlier theoretical result by Whitney in 1936. (4) The general idea is that although all dimensions that clearly define the hidden system may not be available, it is possible to look at a specially constructed time series of the systems behavior along a single dimension that can be measured. This is used to create multi-variable vectors that do have the correct system dimension for the hidden process. Sauer (ref.5 2006) gives an excellent discussion of state space reconstructions. Paraphrasing his discussion:

"Modeling a deterministic dynamical system is build on the concept of a phase space, which is a collection of all possible system states. A deterministic system's state at time t consists of all information needed to uniquely determine its future system states for times greater than t . Consequently for a system that can be modeled precisely, the phase space is known immediately from the equations of its motion.

In experimental and most naturally occurring chaotic or dynamical systems, the phase space and complete mathematical description are often unknown. Methods have been developed to reconstruct a phase space and hence develop new predictive models. To do this, one or more signals from the system must be observed as a function of time. The time series are then used to construct a multidimensional framework (or manifold) over which the observed states change (or move in time)." Whitney's theorem (Whitney 1936) stated that the generic mapping from some n dimensional manifold to another $2n+1$ dimensional Euclidean space produces an "embedding." This embedding can be described more precisely as follows:

Embedding is a multivariate transformation of an initial manifold that changes all trajectories present on the original manifold to produce a new one without any crossings. Stated another way, an embedding is a globally one to one relationship between manifolds so all places on the first manifold where trajectories would have crossed over each other no longer do. This crossing condition if not re-mapped means that future predicted paths (system behaviors) will not be uniquely determined. Such crossing points are formally called *singularities*.

If done perfectly, a visual image of an n degree manifold that would be awkwardly compressed in one number of dimensions is completely unfolded clearly in some other larger space. This occurs when no two points from the original n -dimensional manifold map to the same point in a $(2n+1)$ dimensional space. Whitney's theorem meant that the original states present in this larger new manifold could actually be identified uniquely by a vector of $2n+1$ measurements, thereby reconstructing the actual but not observable phase space.

Takens Embedding theorem (2) took this result one very important step further by showing that this same result could be attained using only a *single* measured variable! Takens proved that instead of needing $2n+1$ generic signals as Whitney found, a time-delayed version of only one state variable say $[y(t), y(t-\tau), y(t-2\tau), \dots, y(t-2n\tau)]$ sufficed

to properly embed the n -dimensional manifold. There are technical assumptions that must still be satisfied to work properly, but for the most part this process is surprisingly straightforward.

The gist of Takens theorem is elaborated summarized by Yap and Rozell (6) in their paper on embeddings for linear dynamical systems. They note that one of the most important tasks in science and engineering is to predict states of some dynamical system from only a stream of its measured output data. An example is stock market prediction using daily market data. Such state identification is usually very challenging because some internal (possibly high-dimensional) system state $x(t) \in \mathbb{R}^N$ (real number manifold N) is often observed only indirectly through a one-dimensional time series of measurements $s(t) = h(x(t))$ like ticker values, where $h : \mathbb{R}^N \rightarrow \mathbb{R}$. In our case we are interested in the physiological time series of the pilots. In most cases what a researcher ever sees is only $s(t)$ but the actual processes that creates our observation (h) is not observable. For pilots this might be underlying mechanisms setting muscle temperature, breath rates, heartbeat rates ect.

The Takens theorem builds its new manifold as follows: Takens defines a *delay coordinate map* $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$ (M being the new manifold N the old) as a mapping of some current state vector $x(t)$ to a point in the new *reconstruction space* ($\mathbb{R}^M$). It does this by taking uniformly spaced samples of the past time series (with sampling interval T_s) and concatenating them into a single vector for every point in the original series,

$$F(x(t)) = [s(t) \ s(t - T_s) \ s(t - 2T_s) \ \cdots \ s(t - (M - 1)T_s)]^T.$$

The effect of this operation is that each point in the first vector now is a set of points defined by the dimension of states chosen for the new manifold M .

The theorem states that under a few conditions on T_s , for almost every smooth observation function $h(\cdot)$, the resulting delayed coordinate map is now an *embedding* of the real state space attractor M . In other words, despite a state being hidden from direct observation, the topology of the attractor that characterizes the real dynamical system can be found in its time series data when it is arranged as a delay coordinate map.

This result has important implications to visualize and understand a complex dynamical system like human physiological response and is the rationale for the method used in this paper. If the underlying manifold of pilot reactions can indeed be ascertained using only a single time delayed measurement e.g. some measured variable, then perhaps a visual representation of the underlying manifold where the state trajectory is more clearly captured may yield a new tool to recognize pilot changes in fatigue or awareness. How we perform the attractor reconstruction on pilot data and use the mathematical tools of RA for visualization and RQA for its numerical analysis constitutes the balance of this paper.

Recurrence Quantification Analysis (RQA):

(Webber, et. al. 2009 refs. 14 & 8) describes RQA as a graphical, statistical tool for the study of nonlinear dynamical systems. Paraphrasing: "Stimulated by the work of Poincaré in 1896 (9) and first developed as a method to represent multidimensional systems in two-dimensional plots by Eckmann et.al. in 1987 (10), other researchers including Zbilut initially defined five RQA features extracted from RA plots of

physiological data (12, 13). Later, three additional variables were added to this set by Marwan et.al.in 2002 (11). The value of RQA in addition to RA came from its independence from assumptions and limitations frequently plaguing other analyses. For example, because plotting recurrence structures (repeating areas where an attractor behaves similarly) are simply tallied, there is no need to precondition time series data with filtering, linear detrending, or other transformations to conform to a particular statistical distribution. Recurrence analysis is not invalidated by signal non-stationaries, transients, outliers or noise, and can be used to detect state changes in highly complex systems. RQA has established a twenty year history with practical implementation in numerous real world problems across multiple disciplines (e.g. physiology, mathematics, physics, chemistry, biology, medicine, neurology, pathology, psychology, astronomy, geology, metallurgy, forestry, botany, genomics, proteomics, semiotics, linguistics).

RQA emphasizes quantification of long or short repeating patterns in living and non-living signals. To the extent patterns can be observed, the systems from which they are derived become potentially more or less predictable. Recurrence is thus fundamental to the scientific process because all experimental results must be repeatable to be verified by others or applied practically. Such patterns must also have mathematical underpinnings and for RQA these are particularly well developed. (An excellent summary of RQA research and assumptions can be found in (1)). To summarize the gist of several papers, seven derived variables have been the most useful in defining how the RA pattern is turned into a plot for later RQA numerical value calculations:

1. The *embedding dimension* of the underlying process (M) is often estimated by using a nearest neighbor methodology for stable or low noise systems and other approximation methods such as correlation dimension for noisier, non stationary systems.
2. The *Delay* (D) between measurements is used to open up or expand an attractors' plot by presenting its largest profile. This variable is frequently determined using the first minimum of a non linear mutual information function or linear autocorrelation function.
3. A *Range* R (i.e. start point and end point) of a time series is used to set window sizes that highlight small or large scale recurrences.
4. The *Norm* (minimum, maximum, or Euclidean) defines the geometric size and shape of a neighborhood that surrounds each reference point in a recurrence plot. Such area around a point is the largest for the maximum, smallest for the minimum, and somewhere in between for the Euclidean distance metric.
5. The fifth parameter for analysis is a *rescaling option*. Typically rescaling is used for multi scale comparisons between plots and includes a unit interval (0.0 to 1.0) or percent scaling interval (0% to 100%).
6. *Radius* is another variable referenced by Weber & Zbilut. The radius creates a cut off limit (or Heaviside function) turning a matrix of differences between path vectors (Distance Matrix or DM) into a recurrence matrix (RM). The wider the

radius, the more points are plotted in a recurrence matrix. Too wide a radius causes loss process resolution.

7. Weber and Zbilut also added a seventh parameter they call the *line parameter* used to aid the extraction of quantitative features from VR plots such as number of vertical or horizontal lines, their point counts, or the number of parallel diagonals showing up in the RM plots. This will be discussed in more detail during the result interpretation process used in RQA.

The most effective way to understand RA and RQA is to operationally see the process working on the visual plots of pilot data. Consequently we now demonstrate RA and RQA use and quantification with EasyJet data plots for a particular pilot. A similar application for the rest of the pilots with sufficient usable data is contained in the appendix. Following this walk through, we will endeavor to draw some conclusions about regularities observed in the pilot data.

EasyJet Analysis

A number of publicly available software programs have been developed to facilitate both generation of RA plots and RQA parameters. These include BASIC subroutines, Matlab, Java, and stand alone packages. So far, we have found the most useful to be GUI based VRA (Visual recurrence Analysis version 4.9) developed by Eugene Kononov. This was originally a commercial product but later publicly released with its last update in 2005. The other packs tend to be isolated line entry subroutine calls individually performing the multiple functions required to study and apply the methods. The present paper used VRA and custom NASA code developed to run under the Rapidminer data mining package (22) with occasional use of other routines to validate our calculations.

Procedure:

As a result of a joint NASA/Easyjet pilot monitoring experiment undertaken in [date and location not recorded in the draft] , physiological data was collected from commercial pilots under varying flight durations and schedules using a Zephyr Bioharness. This wearable harness was capable of measuring heart rate, chest temperature, and multi-axis movements. This data included time of day information with activity recording in three dimensions, as well as direct and derived physiological ratios such as heart rates. Of interest was whether there were statistical regularities in the data that could facilitate choices among alternative flight schedules to promote both reduced pilot fatigue and increase performance. Recognizing that circadian effects may be subtle and hard to detect with traditional summary statistics, NASA decided to explore other methods of analysis designed for time changing systems to see whether they could facilitate the detection of human factors responses to schedule changes. Data was collected from eight pilots over three 5 day blocks (A, B, and C) totaling 15 days of flights for each pilot. Most days had different numbers of flights and different durations including both early morning and late evening departures. Numerous variables were recorded but for this paper our focus was heart rates recorded every three minutes. We also considered chest temperature and autonomic ratio's where possible. Table 1

presents a summary of the pilots considered, the number of their flights, and the number of data points recorded successfully for both flight and non flight activity. Weekends were not recorded during this test.

---- Table one here ----.

To help increase the usability of the RQ and RQA approaches, routines were written by the second author for Rapidminer (ref. 22) that tabulate, sort, and interpret the data, place it into file formats readable by VRA, and permit application of other visual coding schemes including SAX (ref 23) and TESPAPAR (24). Pilot data was then imported into VRA and mapped onto an appropriate dimensional manifold using the Takens theorem.

The optimal dimensionality for this manifold was selected using heuristics suggested by other authors as well as empirical methods. Global False Nearest Neighbors and Correlation Dimension were used to determine a starting point for the embedding dimension selection and Mutual information analysis was used to determine the time lag. As experience grew in interpreting RA plots, these starting points were often adjusted to better accommodate the available data and its limitations. RQ plots were also varied in terms of the color mapping table used, distance metric (e.g. Hamming or Euclidean), mapping locality, and recurrence proximity cut off. Simple grey scales proved useful for seeing periodicities in diagonal lines while spectrum and rainbow color plots were best for observing oscillations between states, local recurrences, and slow transitions.

Figures 1, 2, and 3 below illustrate what typical RA plots look like for pilot 4's heart rate data using a rainbow color scheme for block A, B, and C of pilot 4's. We have added call outs to point out particularly useful features which help interpret what is happening with the underlying generating process during the data period. A full set of visual RA properties and their meanings according to available literature are listed following these examples for both large scale plotting features and small scales.

Figure 1: Some important RA features to look for:

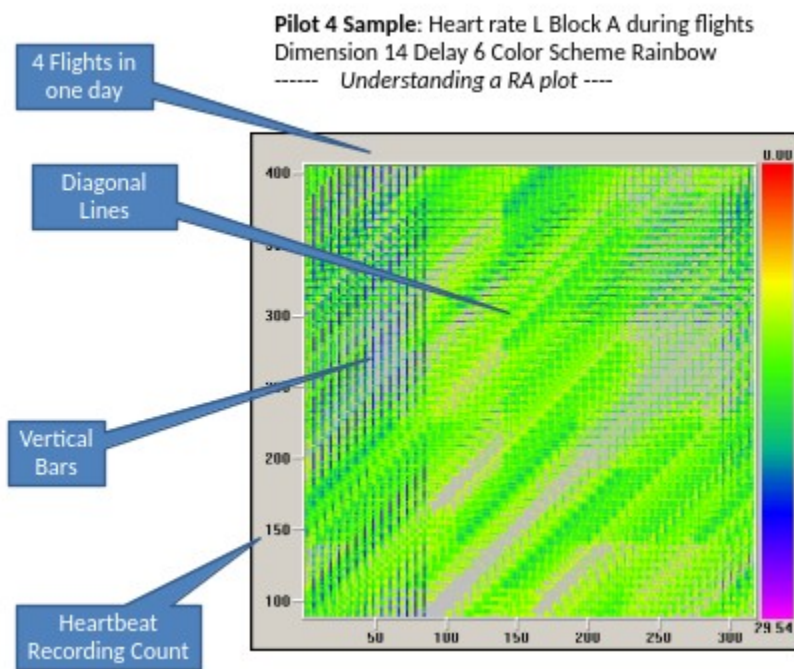


Figure 2: More RA features to look for in Block B :

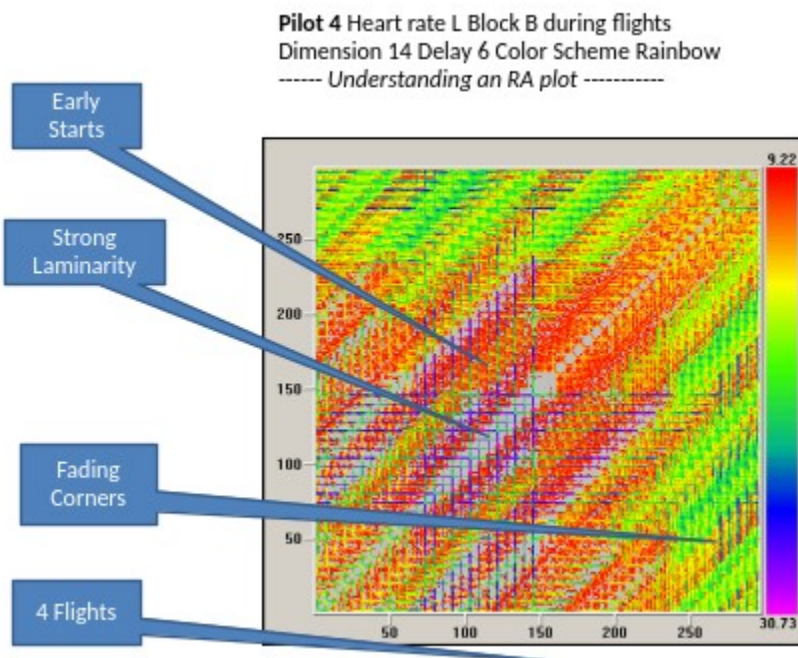
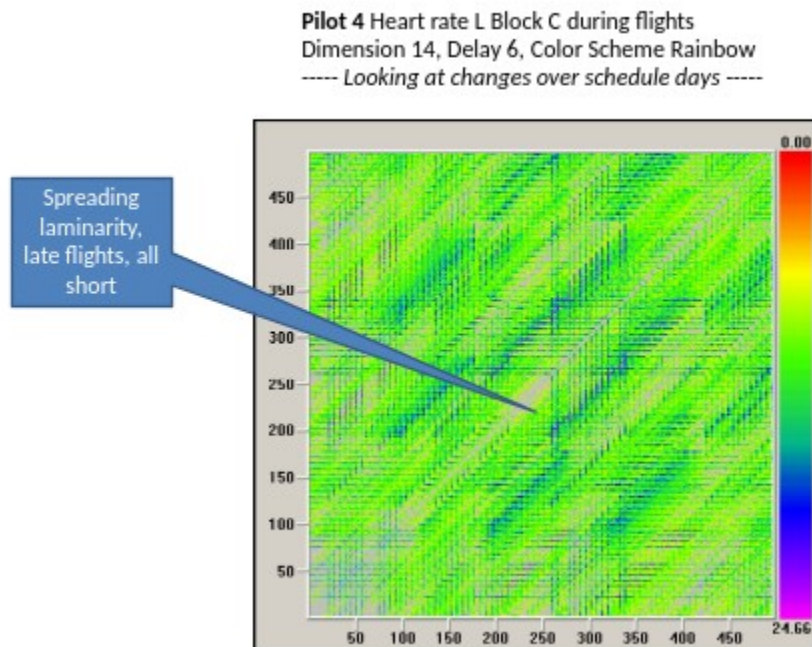


Figure 3: RA features to look for with block C of Pilot 4:



Once a plot is generated, to be most useful for experimental conclusion drawing, there should be theory based guidelines to interpret what is being seen. Because the underlying dynamics of dynamical systems can be quite complicated, this has proved challenging. As a result of many different scientists studying RA plots, some guidelines have emerged spread across multiple research papers. We summarize these below for large plot structures and for smaller areas of change. These guidelines were used to interpret Easyjet experimental results. For pilots with sufficiently large data samples, RQA was also applied using an option in VRA. Here is the list of guidelines used:

Large Scale Structures:

1. homogeneous texture: stationary or autonomous – relaxes quickly to base states – example: random data
2. Oscillating system: periodic recurring structures (diagonal lines, checkerboards)
3. Quasi periodic system: distances between diagonals change
4. Drifting systems: slow changing parameters. Change brightens the upper left and Lower right corners.
5. Extreme events or changes in dynamics: white areas or bands
6. Blocks – transition points, singularities, areas between regular or oscillating behaviors

Small Scale Structures:

1. Single points – rare states
2. Diagonal lines: segments running parallel mean the same region of phase space is visited at different times. The longer the line, the more time two time periods match such behavior implies a stable process
3. Diagonal lines perpendicular to the main diagonal – direction of flow reversed probably the wrong embedding dimension
4. Vertical line: times a state does not change – i.e. staying in one behavioral state
5. Bowed diagonal lines: a pattern repeats, but parameters change at different rates

To get started understanding such information rich plots, it's helpful to take a look at a single heartbeat signal using the same method. Figure 4 shows a typical one dimensional ECG signal coming from an electrocardiogram and its attractor plotted in three dimensions. Because we know a human heart is actually a three dimensional object (the true dimensionality of the underlying generating mechanism), we can plot its attractor function with considerable clarity as seen in the second figure. This figure has three loops corresponding to contraction characteristics of the heart valves. In figure 5, a recorded heartbeat signal of the first author is plotted but as an RA graph. Notice the clearly repeating diagonal lines representing the highly recurrent part of the heartbeat signal and the other blue globular structures representing periods when the heart is resetting nerve fibers for the next beat (corresponding to the heart staying in one particular state). It is these kinds of regularities we are most interested in finding in the EasyJet data because they would correspond to stable processes and changes in those processes as a result of pilot flight conditions.

Figure 4: a heartbeat signal and its attractor function

Human Heartbeat (Webber 2003)

- A. 1-D heartbeat signals plotted in time
 B. a 3-D reconstruction using Taken's theorem for the same series: note the three cycles P, QRS, & T in the plot
 ----- Understanding how Recurrence Plots reflect processes -----

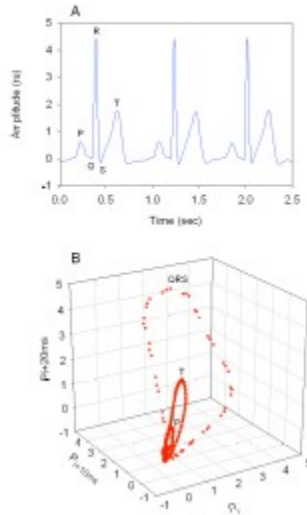
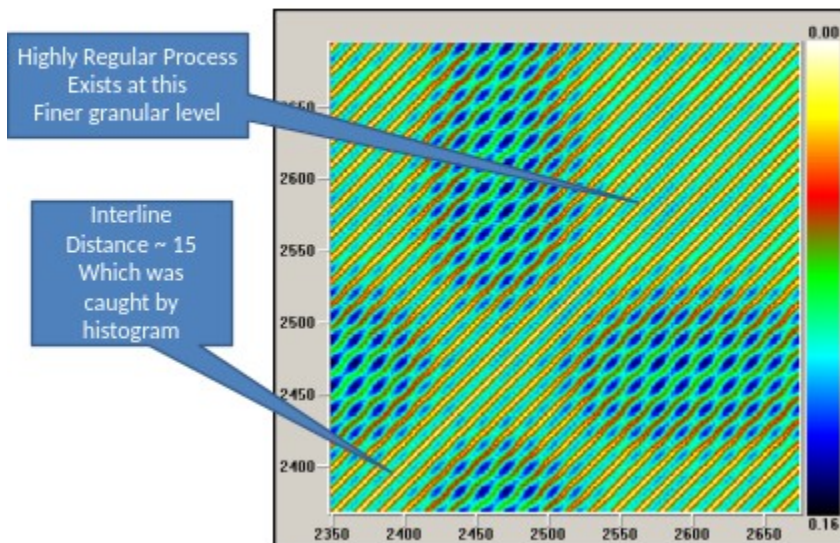


Figure 5: heartbeat signals as an RA graph

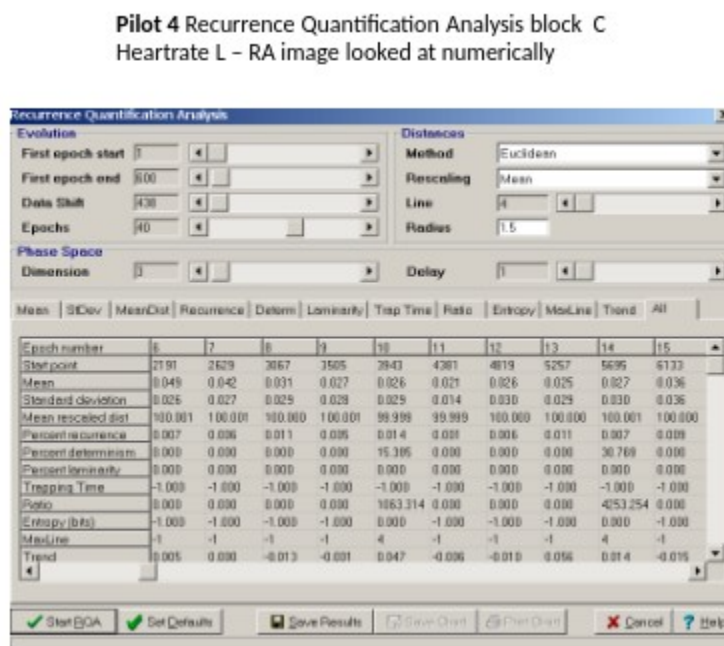
The first author's heartbeats (Dimension 8 delay 3) RA plot zoomed in



Results:

Our first analysis of visual RA pilot data showed that there are periodic processes occurring at different rates between flight and non flight times. [Author's working note: more conclusions to be drawn here from the full data set.]. This was most evident in the width, frequency and length of the diagonal lines and according to guideline 2 above for small scale structures, implied there were stable periods for heart rates but appearing at different frequencies depending on flight or non flight periods. If so, one would expect this observation to show up numerically when quantification analysis is applied to the same charts. An example of one such an analysis along with key RQA factors is presented in table 2 for pilot 4.

Table 2: RQA analysis of pilot 4 block C flight and non flight data combined



As for RA plots there are also a number of derived RQA numeric values which can be extracted from RA plots. Again looking at the literature and summarizing we list them below in Table 3. Not all are generated by the VRA program, but are available from separate routines.

Table 3: The RQA Variables:

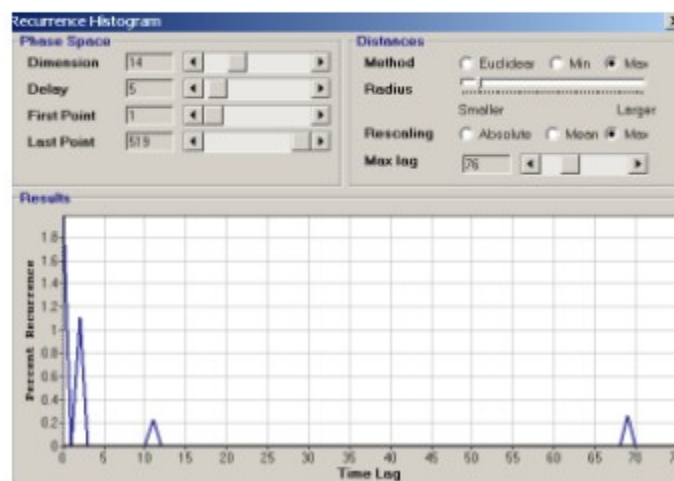
1. **%Det** (percentage of determinism)—the percentage of recurrence points forming diagonals from all recurrence points;
2. **Lmax**—maximal length of a diagonal;
3. **Trapping time (TT)**—mean length of the vertical lines;
4. **Laminarity (Lam)**—proportion of recurrence points forming verticals;

5. **Vmax**—maximal length of a vertical:
6. **Ratio *RATIO***: The ratio between %*DET* and *RR*:
7. **Averaged diagonal line length *L***: The average length of the diagonal lines:
8. **Divergence *DIV***: The inverse of L_{max} Related to the *KS* entropy of the system, i.e. with the sum of the positive Lyapunov exponents
9. **Entropy *ENTR***: The Shannon entropy of the probability distribution of the diagonal line lengths $p(l)$:
10. **Trend *TREND***: The paling of the RP as you move towards its edges: Low *TT*, *Lam* and *Vmax* values imply high complexity in the system's dynamics, because the state of the system stays only for a short time in a state similar to the previously occurring state.
11. **Recurrence Rate (*RR*)**: percent of recurring points overall

Because our first interesting results occurred for recurrent process differences between flight and non flight times, we focused on understanding how often these cycles occur and at with what frequency. To study such differences, we applied yet another analysis that tabulates a frequency histogram showing how often recurrences occurred at specific times. Figure 6 shows a histogram plot for Pilot 1 block C during only flight times.

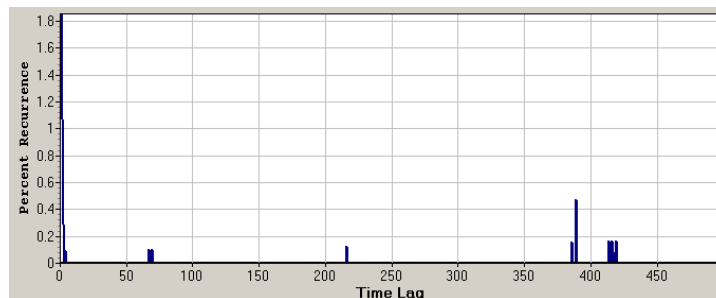
Figure 6: Recurrence Histogram of recurring diagonals for Pilot 1 at different time lags

Pilot 1 Block C flying times
Repeating recurrence patterns at lags of 6, 36
minutes and 141 minutes each lasting
about 7 minutes



As can be seen in the figure, there appear to be three distinct periods of high recurrence for pilot 1 when flying, one somewhere around 2 time lag steps, one at around 11 and one at around 68. Since each lag was a delay of 5 steps for this projection, and each step corresponded to a heart recording every three minutes, we infer there was a repeating effect at $2 \times 5 \times 3$ or 30 minutes, $11 \times 5 \times 3$ or 115 minutes, and $68 \times 5 \times 3$ or 17 hours. When non flying times for pilot one were plotted the behavior changed (Figure 7).

Figure 7: Pilot 1 RA histogram during non flight times



There were four periods of high recurrence, one at around 60, one at 220, one at 390, and one at 420. In other words the periods of recurrence increased and were at greater distances from each other. This behavior repeated for pilot 4 but not for pilots 10 and 15 who showed little if any recurrence behavior. This leads to a series of interesting questions. Were other systematic factors different for pilot 5 and 10 during the flight schedule, for example were they early morning flights versus evening. Were average numbers per day different between those pilots, or possibly other factors not measured effecting the responses? [Author's working note: this is what we need to examine more carefully in this section.]

Conclusions

[Outline left by the authors: usability of the approach; problems in application.]

Future Directions

[Outline left by the authors: other data sets; refinement of RA plots; selection of RQA measures; emotion detection.]

References:

1. Webber, C.L., Jr., Zbilut, J.P. (2005). Recurrence quantification analysis of nonlinear dynamical systems. In: *Tutorials in contemporary nonlinear methods for the behavioral sciences*, (Chapter 2, pp. 26-94), M.A. Riley, G. Van Orden, eds. Retrieved December 1, 2004 <http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.pdf>
2. Takens, F. (1981) Detecting Strange Attractors in Turbulence. D. Rand, & L.S Young (Eds.), *Lecture notes in Mathematics, Vol 898, Dynamical Systems and Turbulence*, Warwick 1980 (pp. 366-381). Berlin: Springer-Verlag.
3. Deyle ER, Sugihara G (2011) Generalized Theorems for Nonlinear State Space Reconstruction. PLoS ONE 6(3): e18295. doi:10.1371/journal.pone.0018295
4. Whitney, H. (1936) Differentiable manifolds. Ann. Math. 37:645-680
5. Timothy D. Sauer, "Attractor Reconstruction", (2006), Scholarpedia, 1(10):1727.
6. H.L. Yap and C.J. Rozell. "Stable Takens' embeddings for linear dynamical systems", IEEE Trans Sig. Proc., December 2010.
7. S. H. Strogatz, *Nonlinear Dynamics and Chaos*. Westview, 1994.
8. Introduction to Recurrence Quantification Analysis, v 14.1) Author: C.L. Webber, Jr. Date: February 3, 2012 RQA toolbox.
9. Poincaré, H. (1896). Sur les solutions periodiques et le principe de moindre action. *Comptes rendus de l'Académie des Sciences, Paris* 123: 915-918.
10. Eckmann, J.-P., Kamphorst, S.O., Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhys. Lett.* 4: 9773-977.
11. Marwan, N, Wessel, N., Meyerfeldt, U., Schirdewan, A., Kurths, J. (2002). Recurrence plot-based measures of complexity and their application to heart rate variability data. *Physical Review E*, 66, 026702.1-026702.8
12. Zbilut, J.P. and Webber, C.L., Jr. (1992). Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A* 171: 199-203.
13. Webber, C.L., Jr. and Zbilut, J.P. (1994). Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology* 76: 965-973.
14. Webber C., Marwan N., Facchini A., Giuliani A. Simpler methods do it better: Success of Recurrence Quantification Analysis as a general purpose data analysis tool. *Physics Letters A* 373 (2009) 3753-3756.

15. Zbilut, J., Thomasson, N., Webber, C. Recurrence quantification analysis a tool for nonlinear exploration of nonstationary cardiac signals. *Medical Engineering & Physics* 24 (2002) 53-60.
16. Marwan, N. and Kurths, J. Cross Recurrence Plots and Their Applications. In *Mathematical Physics Research at the Cutting Edge*. Nova Science Publishers (2004) pp. 101-139.
17. Bautista-Thompson E., Brito-Guevara, R. Garz-Domingues R. RecurrenceVs: A Software Tool for Analysis of Similarity in Recurrence Plots. *Electronic, Robotics, and Automotive Mechanics Conference* (2008). IEEE computer society.
18. Gao, J. and Cai, H. On the structures and quantification of recurrence plots. *Physics Letters A* 270 (2000) 75-87.
19. Zbilut, J., Giuliani, A., & Webber, C. Recurrence quantification analysis as an empirical test to distinguish relatively short deterministic versus random number series. *Physics Letters A* 267 (2000) 174-178.
20. Borell, E., Langbein, J., Despres, G. et. al. Heart rate variability as a measure of autonomic regulation of cardiac activity for assessing stress and welfare in farm animals – A review. *Physiology & Behavior* 92 (2007) 293-316.
21. Thiel, M. , Romano M. C., Kurths J., Meucci R. , Allaria, E., Arecchi F.T. Influence of observational noise on the recurrence quantification analysis. *Physica D* 171 (2002) 138-152.
22. RapidMiner <http://rapid-i.com/content/view/181/190/>
23. SAX coding <http://www.cs.ucr.edu/~eamonn/SAX.htm>
24. R. A. King and T. C. Phipps, "Shannon, TESPAP And Approximation Strategies" Domain Dynamics Limited Cranfield University, RMCS Swindon SN6 8LA United Kingdom

PART III

Oscillating Field Computation

Chuck Jorgensen · NASA Ames Research Center

A technical report draft prepared at Ames under NASA's CICT/ITSR program, previously unpublished. It proposes a computing architecture built from canonical nerve-cell models, in which bursters detect, resonators amplify and oscillators store patterns by phase locking, and behavior emerges from a hierarchy of field-like modes.

Reproduced from the author's file with the wording unchanged. Tracked changes were accepted. Square brackets mark references and figure numbers the draft left incomplete. See the editorial note in Part I, section 9.

Oscillating Field Computation

Chuck Jorgensen[†] *Senior Member IEEE*

1

Abstract—Biological research, quantum effects at small material scales, and better understanding of large non linear networks are leading to a reconsideration of the theoretical foundations of current computer designs. Insights gained from the mathematical dynamics of nerve cells imply the generating mechanism of algorithm like behavior in living systems is fundamentally different from that used in classical computation.

Inherently analog, living system “computation” appears grounded in the mathematics of large, coupled oscillating networks. Derivative behavioral computer architectures based upon dynamically entrained network elements may be better suited for several NASA mission problems including adaptive intelligent robotics, time-dependent systems, and application of materials fabricated for quantum or nano scales. This paper is an initial effort to consolidate recent biological and mathematical insights into a unified computing architecture using an IT perspective. The approach combines canonical nerve cell models and the mathematics small network theory with dual dynamic control concepts and behavior-oriented field computer models. Implications for new computational algorithms, analog logic, and programming are considered.

Index Terms—Neural Networks, coupled oscillators, analog logic, computing architecture, field computers, dynamical systems, small world networks, statistical mechanics

I. INTRODUCTION

Computation like processes occurring within living systems can appear to an outside observer as algorithmic or even procedural. Hence there has been a long-term interest in whether biological research can serve an inspirational role for computational algorithms and computer architecture designs. When the analogy with traditional computer designs is examined more closely, however, fundamental differences in the computation mechanism and system purposes become evident. In particular, philosophical and architectural differences arising from totally analog computation impact a broad range of computing functions ranging from memory to logic [2,20,21]. As a result, many simplifications have been made by researchers to distill the most useful engineering principles from the huge body of biological research findings and maintain focus on simple, useful functions. A current example is the field of neural networks which has been largely based on a highly simplified model of a Hebbian neuron. This simplification has not been without cost, however in that many important non linear, dynamic properties of the cell have been omitted.

The difficulty of accurately modeling even small collections of nerve cells for computing purposes illustrates

the magnitude of the problem. As the number of simulated neurons increases and the goal becomes biologically realistic calculation, problems of scale predominate. For example, the human brain alone has over ten to the 11th power cells, each has up to ten to the third power afferent inputs. Considering just one aspect of these biochemical cells, there are over twenty known types of intracellular current flows. This would lead to over two to the 20th theoretical model variations were models based on current flow alone.

However, if cell models are analyzed as interconnected dynamical systems and simplifications such as lumping of variables is used, one can take advantage of regularities in the state spaces of the equations as they evolve with time [13]. With all the complexity of reasonably accurate biological cell models, under general assumptions, there are still only a limited number of possible dynamical system behaviors possible [18]. Labels for these behaviors include attractors, basins, saddle nodes, bifurcations, cycles, and many variants such as blowouts, attractor bubbling or on-off intermittencies. Using canonical equations rather than specific cell models it is possible to build an inclusive mathematical taxonomy of cell behaviors. Some of these have been previously categorized in biological literature (e.g. Type I and II Hopf bifurcations) and some provide potentially new behaviors not currently observed or used.

Analysis of canonical dynamics has resulted in a number of key behaviors observed when exogenous and endogenous stimuli cause a cell to fire. These include:

- Point attraction - the summing mechanism of most simple Neural Nets
- Bursting - where a cell responds to complex time and phase patterns with sudden activity
- Resonating - damped oscillations triggered by burst frequencies for limited time periods
- Oscillating - self excited oscillations that frequency lock within limited frequency or phase ranges
- Pulse-coupling, phase locking, frequency locking, and entrainment where weakly connected cells begin to move in synchronization
- Periodic and chaotic bifurcation where sudden branches and changes in behaviors occur and
- Relaxing, current leakage, and uncoupling processes which can disconnect phase and frequency locked units.

Other conditions have been found which effect the way a cluster of cells communicate and whether or not they will

[†] This work was supported by NASA Ames Research Center under the CICT/ITSR program, Program manager Eugene Tu. [†]Dr Jorgensen. is with the Computational Sciences Division, NASA Ames Research Center, Moffett Field CA 94035. (e-mail: cjorgensen@mail.arc.nasa.gov)

entrain. These include:

- How slow and fast connectivity processes determine sensitivity to features
- How weak and strong energy exchanges within a network effects global locking.
- And whether dynamical systems defined as field equations can exceed computational capabilities of Turing machines e.g. Hyper Turing Machines [19].

In juxtaposition to dynamical systems theory which has been vital in understanding individual cell behavior, statistical mechanics has been shown capable of dealing with larger cell populations and provides a rigorous way to describe non-linear network behavior. The mathematical analysis of nonlinear networked systems remains a significant challenge for research in a number of fields [1, 22]. As a result, scientists applying cell inspired algorithms have tended to focus on simple, homogeneous oscillatory networks and using limited sets of highly structured cell models. Even such small nets exhibit coupling behavior that leads to useful associative memory and pattern recognition algorithms. Recent work demonstrates the potential of coupled oscillator approaches for application to real world problems. Some examples are:

- Novelty filters (Skarda & Freeman 1987)
- Chaotic Logic Gates [2]
- Efficient neuron communication via frequency locking (Izhikevich 1999)
- Nonlinear pattern classifiers (Freeman 1988)
- Pulse coupled oscillators for clustering [32]
- Long term memory storage (Tsuda 1991 and Wang 1999)
- Self referential logic generator s (Nicolis 1985)
- Dynamical system control (Banke, 1997)
- Database categorization (Frigui 2002) and
- Speech recognition enhancement (Volkmer 2001)

II. APPROACH

Formulating a biologically motivated computing architecture without abusive simplification of what is known about living systems is indeed a challenge. We have chosen to depend on canonical equation models capable of demonstrating the range of generic cell functions discussed above, but limit ourselves to network designs that are most likely to generate control like behaviors given specific input stimuli. Even after determining a viable subset of differential equations, measurable parameters need to be identified and their values set so as to specify initial conditions and temporal behavior. Merely having a set of computing

components without understanding their role in the broader context of computing functionality necessitated an examination of current computer designs and their original design rationales.

Hence, we next contrasted the selected equation models with existing computer components that were functionally equivalent to determine similarity and differences. This step provided important clues as to what a biological system must achieve to generate their observed behaviors. The most difficult step was to formulate an integrated architecture using these models was still amenable to formal with the potential to provide new insights and performance improvements over exiting computer structures.

1) Overview

Using canonical cell models as building blocks, we have focused on three nerve cell behavioral models: **bursters**, **resonators** and **oscillators**. In general, models referring to these nerve behaviors will be represented later in this paper by ψ . There will be other general models concerned with input fields and signal filtering, these will be referred to by Φ . Models associated emphasizing control and output behavior will be distinguished by Ω .

This paper proposes that the end results of ψ firing behavior are three important processes: feature detection of temporal and energy magnitude patterns, feature segmentation and amplification, and associative storage respectively. Each math model corresponding to a given nerve type has a specific function. **Burster** cells respond to complicated time and phase patterns due to the mathematical structure of their phase space. **Resonators** amplify small but specific signal patterns but only for limited periods due to their damping. **Oscillators** lock together and store patterns because they remain excited after switching on.

In contrast to a bottom-up pattern recognition scheme which begins with simple static features and generates classification trees, this paper uses a mixed structure in which a burster cell responds to an extended phase and amplitude pattern entering from outside the system. The bursters relatively complex firing sequence is later dissected by a spatially distributed set of resonator cells who broadcast small piecemeal responses to sets of self exciting oscillator cells. At the same time their larger numbers serve to amplify the relatively small response of the more externally signal specific burster cells.

Oscillator cells then phase lock to the resonator cell patterns producing a dynamic associative memory storage system. Because of temporal asynchrony caused by non clocked resonator behavior, networks formed must come from energy exchange occurring in irregular pulses. Until recently [29] it was not clear that such “pulse coupled networks” generated phase and frequency locking behaviors required to produce a stable associative memory. It has now been established that such networks do indeed entrain, and that pulse coupling can be used as a memory mechanism in an architecture composed only of discrete cell models.

The locking response creates spatial and temporal regions subdividing the full set of oscillators. We call these areas behavioral **modes**. Due to dynamics, modes form in a hierarchy of frequencies and time scales. The mode time scales implicitly delineate sub clusters depending on coupling frequencies. High level behaviors have slow time scales, lower levels have faster shorter times up to some minimum level where a **micro behavior** terminates the subdivision.

Micro **behavior** is the lowest level process and has two parts. An integrating point attractor equation, which serves as a switch and whose threshold crossing triggers a second differential equation that is a controller. The controller generates a time-based signal that influences some process affecting its environment. In a biological context this could correspond to the firing of a single muscle fiber. In the current architecture it corresponds to firing any type of control activity or switch.

Individual micro controls within the bounds of a single behavior mode compete to fire. Competition becomes a balancing act of excitatory and inhibitory field constraints imposed by the neighboring cells. Execution of each micro behavior continues until energy stability is reached for a mode. At that time activation moves upward to the next slower time scale or until no other external stimulation excites firing i.e., a top level “goal” or stable condition is reached.

This summary overview will now be expanded by following a signal through the system in a series of steps. We begin with the receipt of external sensory information, and continue processing steps through memory storage, micro task firing and behavior management. We then analyze the overall model in terms of activation fields and contrast it to a traditional computer design. We finish with a discussion of the feasibility of implementation and our current progress in this regard.

2) Initialize Canonical Models

Several parameters must be specified that determine oscillator, burster, and resonator phase space characteristics. These parameters change the oscillation rate, resonance frequency, and the signal damping. They are:

1. An excitability threshold for each neuron
2. A frequency of oscillation to which it responds
3. The signal amplitude produced when fired
4. The firing threshold type e.g. oscillating or static
5. Initial and boundary value conditions
6. and damping rates where appropriate for the cell type.

The above cell models are all variations of oscillators commonly simplified into terms of a second order system such as

$$\psi'' + \omega^2 \psi = 0 \quad \omega \text{ is frequency of oscillation}$$

This second order system is reduced further to two first order equations through substitution of variables. In the individual oscillator equations that follow, ψ will generally be the second order equation decomposed into two variables, x and y corresponding respectively to a fast term reflecting membrane potential and a slow term defining the evolution of firing thresholds and slow chemical processes moderating cell behaviors.

Getting parametric values into the network from outside requires a communication concept of how a user would access the architecture and reconfigure it. One concept might be the following. Burster parameters for example could be broadcast from a single-entry point into the network where the bursters are contained. The entire system is envisioned as being composed of weakly connected networks of oscillators so a method of selectively addressing equations and broadcasting parameter changes is needed. There are some unique communication properties of coupled oscillating networks which make a wide band broadcast method a desirable approach. These are associated with the analytic behavior of small random networks and will be discussed more later.

At present I envision the broadcast of a complex valued waveform carrying four information channels. Information would be carried on the phase and frequency dimensions of both the real and complex axes of this signal. The parameters sent over the channels consist of initial time conditions, burster thresholds, initial values, and external stimuli values.

Since learning is presently not defined for the system, these must be predetermined. As will be discussed later, treating the entire system as a set of interacting potential fields permits linkages to be established to neural network learning paradigms that might resolve this deficiency.

Targeted cells are uniquely addressed using a specific frequency on one of the waveform channels and frequency locking to set up communication. For Example, each cluster of bursters would be activated by one particular frequency. I presume such a waveform would be programmed using a software tool. This method requires development and if infeasible a hard wiring approach through simulation code could be used in the short term. Ideally, the ability to quickly change response characteristics of components by broadcast should be maintained since dynamic flexibility is one of the major computational advantages of this type of architecture [14].

Micro behavior control equations must also be connected to micro behavior activation functions. Micro behaviors must be associated with modes; modes must be associated with goals. Goal attainment is viewed as minimizing a gradient optimization of multiple mode field potentials which arise when an area of oscillators phase lock. Phase lock field

patterns ultimately determine which modes are accessed and which micro actions can fire.

Behavior mode hierarchies are formed dynamically from harmonics in oscillator couplings and from the distribution of oscillator frequencies used. This is a very nice feature, however, cross associations can also emerge automatically from the interaction dynamics. It is possible these may be undesirable much in the same way associative storage can produce incorrect associative responses in Hopfield networks.

Initially, hierarchies will be job specific and either must be newly defined or drawn from a universe of building block tasks previously specified off line. A micro behavior and the mode hierarchy must also implement blocking and unblocking conditions for activation. This will depend upon threshold values of oscillators associated with each task group. Ultimately, I believe a number of these functions may become a by product of adaptive learning behaviors not yet specified.

3) Stimuli Excite the System

Exogenous stimuli effect the larger system via sensor fields Φ . These fields are generated by the responses of multiple cell models filtering input signals. A sensor's field currents impact the firing of internal (endogenous) oscillator sets. At any given time, current firing patterns define the present state of the system. All signals propagate (in a yet loosely defined manner into the entire network) using small-scale network dynamics. After oscillations within a frequency locked portion of the network have stabilized (a process that initiates firing of multiple micro actions), outside stimuli related to the goal state are evaluated to determine whether a high level goal has been attained. This would be a continuous process in a closed loop for living systems but for the present architecture expanding the definition to include recursive calculations exceeds the scope of the present paper.

4) Bursters Generate Features in Response

When neuron activity alternates between a quiescent state and repetitive spiking, the neuron is said to be bursting. There are two important bifurcations in phase space associated with bursting behavior: those that lead to repetitive spiking and those that lead to quiescence. The former are characterized by: folds, saddle nodes on invariant circles, supercritical andronov hopf bifurcations, sub critical andronov hopf bifurcations, fold limit cycles, and saddle homoclinic orbits. The latter are characterized by saddle nodes on invariant circles, saddle homoclinic orbits, supercritical andronov hopf bifurcations, and fold limit cycle bifurcations. The resulting twenty four categories and general burster behavior are detailed in [18]. More specifically a burster can be described as a singularly perturbed dynamical system of the form:

$$\begin{aligned} \dot{x} &= f(x, y) \\ \dot{y} &= g(x, y) \quad x \in \mathbb{R}^n, y \in \mathbb{R}^m, \mu \ll 1 \end{aligned}$$

$\dot{x}$ is the fast subsystem modeling spike generation and the membrane potential. $\dot{x}$ depends on the bifurcation parameter y which is assumed fixed in this case rather than slow varying. $\dot{y}$ actually models a very slow subsystem which describes the evolution of bifurcation parameters. The fast system can exhibit the multiple types of bursting behaviors alluded to above. Among the more well known types are parabolic bursting which arises when a saddle node bifurcation is on a limit cycle, square wave bursting when a saddle node bifurcation is not on a limit cycle, and elliptic bursting. The latter typically occurs where a bifurcation is of the subcritical Andronov-Hopf type. Well known examples of this type of neuron are the Morris-Lecar, Fitzhugh-Nagumo, or Hodgkin-Huxley neurons. Ermentrout and Kopell's theorem [37] showed it was possible to reduce such a system to a cleaner canonical form:

$$\begin{aligned} \dot{\varphi} &= 1 - \cos \varphi + (1 + \cos \varphi) p(\theta) \\ \dot{\theta} &= \omega \end{aligned}$$

φ describes the fast activity of x , and θ describes the slow activity of y .

$$\begin{aligned} \omega &= O(\mu / \sqrt{\epsilon}) \\ p: S^1 \rightarrow \mathbb{R} &\text{ is a continuous function describing} \\ &\text{the influence of } \theta \text{ on } \varphi \end{aligned}$$

If p is negative for all θ the fast variable φ is quiescent and bursting does not occur. If $p(\theta) > 0$ for all θ , φ fires continually and there is no quiescent state. Bursting occurs only when p changes signs.

Thus complicated feature patterns of time and frequency from exogenous signals stimulate specific bursters to fire by exceeding their thresholds at the correct times and magnitudes to maintain continued burster firing. Because a phase transition for a burster can only occur when both correct energy and correct time signal impacts the cell, burster activity is signal pattern specific and will quickly die out if the stimulating signal is not correct. The correct external stimulation triggers a single calculation of one phase trajectory after which the burster enters a refractory period. This refractory period also serves an important stability function for the network

Bursters are anticipated to be distributed in large

stochastically parameterized sets to cover many potential input signal feature spaces. The values for the burster equations effect the phase space behavior (the dynamics such as the bifurcation points, attractors and cycles discussed above) and the manifold of the temporal and frequency responses of the bursters. Burster equations must be custom defined at present and the nature of temporal behavior and dynamic phase space landscape clearly delineated.

Some work defining phase space landscapes has been done for the learning of robotic trajectories. This research used non linear regressions on canonical sets of motion equations to define primitive movements [7]. As a starting point, bursters could also be selected from a set of observed behaviors using generate and test methods as well.

5) Resonators Dissect the Features

As noted by Izhikevich [30] an important feature of resonator models near Andronov-Hopf bifurcation points is they exhibit a damped subthreshold oscillation of their membrane potential. This oscillation permits a resonator to fire multiple times in response to a frequency that matches the oscillation timing. Because it is damped, firing only continues for a limited period for specific frequencies. A resonator neuron in a network can be described by a complex equation:

$$\dot{y} = (b_i + i\omega_i)z_i + \sum_{j=1}^n c_{ij} \delta(t - t_j^*)$$

where $z_i \in \mathbb{C}$ describes the state of the i -th neuron

$b_i + i\omega_i \in \mathbb{C}$ is its internal parameter

$c_{ij} \in \mathbb{C}$ is a synaptic coefficient

δ is the Dirac delta function

and t_j^* is the nearest moment of firing of j

A resonator firing produces a pulse that displaces the activity of the other neurons by a complex valued constant $c(ij)$. After $z(i)$ fires an action potential it is reset to a new value.

As bursters produce their output signals, they re-map the original input stimulus field into a convolution of internal system states and external stimulus influences. Different segments of these burster signals will have similar timing and frequency to the frequency response of members of larger sets of resonators distributed through the network system network. For their short period, excited resonators thus act as an amplifier of parts of a burster's signal and simultaneously segment a burster sequence into temporally

and spatially distributed firing maps

These spatially distributed signal patterns diffuse into the larger network stimulating the task modes and micro actions discussed above. Without continuous new stimulation, a signal from a damped resonator stops and eventually it resets only to re-fire given a new signal or part of the same signal later in time.

6) Oscillators Create Associations

In general oscillators are the workhorse components of this architecture. Their usefulness comes because until switched off they remain self exciting after exceeding a threshold. Continued excitation makes them a natural candidate for an associative memory. For example, a detailed treatment of how oscillators based on the Kuramoto model can be used in a computing architecture can be found in [31]. A simple oscillating cell network can be described by:

$$\dot{x} = 1 + \varepsilon(1 - \cos 2\pi x_i) \sum_{j=1}^n s_j \delta(x_j)$$

where δ is the dirac delta function and x_j and s_j refer to connections to other oscillators and their signals

This system can also be transformed into a phase based canonical form for easier implementation purposes.

By this point in the processing sequence it is assumed a large set of resonators are firing. In aggregate, they can produce sufficient energy to exceed the higher thresholds of the self-excited oscillators (assumed to require a much larger cumulative number of weak excitations to fire). Multiple sums of weak excitations have been proven to be critical for phase locking [17]. The stimulated oscillators switch on depending on their current state, thresholds, and refractory condition. In contrast to bursters and resonators, they will remain excited until switched off or suppressed by inhibitory currents. In this way oscillators can have a long lasting structure and produce "associations" with other areas of the memory through frequency locking.

In normal operation, a constant stream of firing energies would be transmitted into the network of oscillator cells. The network is assumed randomly connected, obeying the statistical mechanics of small world network theory [28]. The message passing dynamic of such nets will cause other oscillators of like characteristics and frequency to lock quickly in local regions. The small world nets (SWN's) are assumed to exhibit three basic properties [23] to insure this behavior:

1. "Small world" means there is a small number of intervening steps between any two nodes. This is

analogous to the sociological theory of “six degrees of freedom” between individuals.

2. Cliques occur in the form of clusters where:

$$C_i = \frac{2E}{k_i(k_i - 1)} \text{ where } E \text{ is the existing edges}$$

$$\frac{k_i(k_i - 1)}{2} \text{ equals the total number of possible edges}$$

3. Degree distribution is present, where few nodes (cell models) have the same number of edges (afferent paths)

As a point of historical interest, significant computing power may be available as a result of only network architecture. Turing's proposed B-Machine Neural Architecture of 1938 operated on such a principle. A neuron could do only two things: pass data through unfiltered or block the flow of data. He showed such a system implemented NAND gate logic and hence could compute. His architecture permitted unlimited feedback and recursion in the data flows. It is interesting to consider is that the present system may well have important computation capabilities simply by communicating in small nets and passing some data and not other during locking.

7) Locking builds clusters

After oscillators switch on, they also exchange weak energies via network connections. This causes locking with other non local sub clusters within the network. It also leads to the generation of harmonics at different time scales. Locking of the entire network can be prevented by inhibitory energies following the dynamics of the equations selected for the oscillator [25]. One model we have explored is a simple network of identical oscillators is that does not use inhibition:

$$\dot{x} = bx - \omega y$$

$$\dot{y} = \omega x + by$$

$$x(t) = e^{bt} [x_0 \cos(\omega t) + y_0 \sin(\omega t)]$$

$$y(t) = -e^{bt} [x_0 \sin(\omega t) + y_0 \cos(\omega t)]$$

where b = damping a coefficient

ω = frequency of oscillation

and x_0 and y_0 are the initial conditions

This simple model can exhibit locking and unlocking

behavior. This is because two types of excitation can cause locking or unlocking of cells within network areas. The first is due to locality of effects and occurs upon the reception of signals in the immediate vicinity of an oscillator. The second type is emergent and can result from sudden large changes in the noise levels in local signals.

A cell oscillation usually initiates for one of three reasons: first, paradoxically, it can be triggered by a large local inhibition - in effect providing a sudden jolt of negative energy to a sub threshold membrane oscillation at just the correct time to increase its magnitude above a firing threshold [15]. Second, an oscillator might move out of a refractory period after turning off and enter the pool of potential firing oscillators. Finally, an oscillator might fire simply because of mutual excitation of its neighbors.

Although oscillators are assumed maintain excitation, if there is still some small minimum of external energy required to maintain excitation, consistent changes in currents in an area could cause an oscillator to drop out of excitation. This would tend to be at a much lower probability that that of staying in an excited mode but would permit the breaking of locked connections.

8) Behavior Modes Form

By this point it is assumed an external stimulus has caused a number of internal oscillators to lock into temporal and topological clusters or modes. A mode is a functional association with “similar” stimulus patterns. Each locked region can be thought of as a consistent response by the architecture to somewhat different stimuli and define the range of excitatory conditions. Modes are the enabling path to micro actions. Micro actions in turn effect the environment external to the network or change its internal parameters. Micro actions are defined in terms of differential equations linked to individual oscillators. Without active inhibition from a mode field, micro actions execute.

Specifying the details of how modes and micro actions interact has not been completed at this time. One way the problem might be approached is to use principles analogous to a dual dynamic robotic control system [4, 5]. In such a system, at each behavior level, an excited mode is considered a potential field. A potential field approach for goal seeking is common in robotic obstacle avoidance and navigation literature. The potential field can be represented in a set of field equations. The analysis of field equations has proved a rich area of theoretical research for analog computing designs and concepts for the description of reasoning and task formation in time varying systems.

The global effects of an excitation field can be specified a number of ways but MacLennan [34] provides a particularly elegant treatment for field calculations whose form we follow here. Let

$\phi(z, t)$ represent input field excitation at time t at oscillator z

$\Phi(t)$ represent the global field of all oscillators z

$\omega(x, t)$ represent micro actions that effect processes external to the net

$\Omega(t)$ represent the field produced by these micro actions

$\psi(y, t)$ represent internal oscillators

$\Psi(t)$ represent the field generated by the internal oscillators.

For simplicity, assume linearity of field effects. Define

$$\Omega(t) = E\Phi(t) + G\Psi(t)$$

$$\Psi(t) = D\Phi(t) + F\Psi(t)$$

Let D, E, F , and G be slowly changing mode fields representing a linearized coupling strengths between modes. For example, E_{yz} would represent the coupling strength between elements of Φ with oscillators z . G_{zx} represents the coupling strength between the internal oscillators of Ψ i.e. z and output micro task x in Ω . Following [34] the system can be analyzed by taking a Laplace transform of the state evolution equation on ψ and solving for the forced response of the system $\omega(x, t)$. As is shown below the system response is a superposition of convolutions. That is, $\omega(x, t)$ at output micro task x is generated by the sum of each stimulus signal $\phi(z, t)$ convolved with a transfer function $\mathbf{h}_{xz}$ where $\mathbf{h}_{xz}$ is the micro task stimulation signal seen at x when an impulse is injected into oscillator z with all other input sites fixed at zero. The calculation proceeds as follows.

Taking the Laplace transform L of the evolution equation above

$$s\Psi(s) - \psi(0) = D\Phi(s) + F\Psi(s) \text{ where}$$

$$\Phi(s) = L\{\phi(t)\}$$

$$\text{and } \Psi(s) = L\{\psi(t)\}$$

$$\Psi(s) = (sI - F)^{-1}[D\Phi(s) + \psi(0)]$$

Setting $\psi(0) = 0$ then the Laplace transform of the output equals :

$$\Omega(s) = E\Phi(s) + G(sI - F)^{-1}D\Phi(s)$$

its transfer function H equals :

$$H(s) = E + G(sI - F)^{-1}D$$

$H(s)\Phi(s)$ when expanded and

transformed back to the space time domain yields :

$$\omega(x, t) = \sum_x \mathbf{h}_{xz}(t) \otimes \phi(z, t)$$

or the superposition of convolutions where

$\otimes$ represents the convolution operation

MacLennan [35] points out that for normalized signals, the inner product is maximized when they are equal . This fact has been used in the neural network community as a pattern matching technique. However, the inner product is also the final value of a reversed convolution. If the linear system above has an impulse response that is the time reverse of some stimulus pattern, then the system will also act as a pattern matcher for that input pattern i.e. a “matched filter”.

Another question is how a mode field of Ψ might actually select a specific micro task. If the input and oscillator fields were viewed as superimposed over a grid of micro tasks in Ω so that their potentials couple, then local maxima or minima in Ψ and Φ would excite the energy thresholds of the summer equations restraining the firing of microtasks. For example keeping an superpositioned field model in mind (see figure [number not assigned in the draft]), one could begin to see an analogy with dual dynamic robot path navigation. An example is illustrative.

9) Selecting Micro Actions

Suppose we define micro task grid relationships as follows. Assume Φ impacts Ψ and together they produce some point of maximum potential. Let it be a goal. Similarly let the gradient of largest values on Ψ define a sequence of micro task grid points receiving maximum stimulation from the task mode field going from largest to smallest values.

As discussed above, task mode locking creates a series of potential fields. Field models have been used in goal seeking, navigation, and robotic obstacle avoidance research.

[reference to Thorp left incomplete in the draft]. One way a mode's potential field might influence device control or navigation behavior to a goal would be to select desired behaviors by vectors defined in terms of field gradients. Although in theory a single oscillator $\varphi(t,z)$ value could be the link controlling a single micro action, greater robustness to noise would be achieved if a field from multiple oscillators determined which micro action fires. Which micro task gets selected would be determined a point of maximum average field energy. Similarly, if a field reflected spatially defined stimulus variables, goal areas could be specified in terms of there maximum positive field points. Points resulting from overlaying of fields would create both positive attractors and negative field points or repellers. Repellers would be the analog of obstacles to positive goals. Within a field, it would be possible to specify directional vectors correspond ing to field gradients. The vector direction would then serve as a pointer to eligible micro actions.

At higher levels, the system would have fields compete with each other based on the largest total activation energies. Because field dynamics change as oscillators lock and unlock, corresponding active modes would change with time and stimulation.

Slow time scales reflect the broadest impacting inclusive fields in the mode structure and presumably correspond to the most general behaviors. The fast time scales correspond to the most fine grained behaviors with terminal points ending in micro actions.

Ω is also continually impacted by external stimulus energies $\varphi(z,t)$ moving through the network into the system from sensors. If these do not excite Ω dynamics they are treated as noise. If they excite the oscillators they cause fluctuations in the fields with time and result in changes in the sequence of micro actions firing.

For example using the dual dynamic architecture format vectors defining micro task sequences caused by fields might be represented as follows. Let:

$$F_{\text{attract}} = -\alpha \cdot \sin(\psi d - \omega d_{\text{goal}})$$

F_{attract} is the angle to a goal in an oscillator field
 ψd is a vector giving magnitude and direction of a mode gradient, ωd_{goal} a vector to a stimulus goal from the modes maximum potential point

$$F_{\text{repel}} = D_{\text{repel}} * R_{\text{repel}} * S_{\text{repel}}$$

F_{repel} is a composite of repeller influences in 3 parts

$$D_{\text{repel}} = \frac{(\psi d - \omega d_i)}{\Delta \omega d_i} e^{-\frac{(\psi d - \omega d_i)}{\Delta \omega d_i}}$$

$$R_{\text{repel}} = \frac{1}{2} \left[\tanh(h_1 (\cos(\psi d - \omega d_i) - \cos(2\Delta \omega d_i + \sigma))) + 1 \right]$$

$$S_{\text{repel}} = e^{-\frac{r_1 - r_2 - r_3}{d_{\text{loc}}}}$$

$$F_{\text{repel}} = \sum_{i=1}^n F_{i \text{ repel}}$$

$$\omega d'_{\text{goal}} = F_{\text{attract}} + F_{\text{repel}} + \text{noise}$$

$$\dot{\Psi}_k(t) = F_k[\Psi_1(t), \dots, \Psi_n(t); \Phi_1(t), \dots, \Phi_n(t)] + N(t)$$

i.e. the change in a local oscillator field is some function F of other mode fields at that level plus the input stimulus fields and noise. If we separate F into two linear operators L and M and operate on the mode oscillator state and stimulus field respectively then :

$$\dot{\Psi}_k = L\Psi + M\Phi + N(t)$$

We can define a diffusion process $\delta = k^2 \nabla^2 \delta$

where ∇^2 is the Laplacian defined as $\nabla^2 \delta = \sum_k \frac{\partial^2}{\partial x_k^2}$

Then the stimulus fields influence on Ψ is a result of a diffusion of stimulus excitation points x on Φ

$$\dot{\Phi}_x = k^2 \left(-\Phi_x + \frac{1}{n} \sum_i \Phi_i \right)$$

Using a convolution of Φ with a Gaussian γ

to perform the averaging of $\frac{1}{n} \sum_i \Phi_i$

$$\dot{\Phi} = k^2 (\gamma \otimes \Phi - \Phi)$$

In the above equations, D repel corresponds to a generic potential well in the stimulus field. R limits the angular range of its influence, and S is a scaling factor representing its field influence as a function of distance from the most active potential in the oscillator mode field. The ρ and P terms in R refer to the radius of influence for the repeller radius, and the micro task potential radius. "d" refers to the effective maximum distance of influence of a field. The overall effect then is a sum of the field vectors for attraction, repulsion, and system noise. Task sequences become vector following as mode fields stimulate micro task grid elements.

B. Field Computation

There are some excellent papers in which a presentation of field based systems have been formally analyzed. One of the most extensive is contained in [22] where MacIennan proposes "field transformation computers". He also discusses a universal field computer and how it might exceed Turing machine capabilities. A key point is that fields can be considered functions or sets of functions describing a field distribution.

Although each potential point in an oscillator field has a

discrete value, if the number of points is large enough it can be treated as continuous. This permits a direct mapping of oscillators to fields and fields to functional descriptions. For example, the dynamics of $\Psi(t)$ can be expressed as a differential field equation. Using [35] let:

In this formulation, interactions between competing fields are treated as a diffusion process. Presumably, there are multiple mode levels above any micro task so there is a resultant hierarchy of mode extensiveness, time scales, and micro tasks. If sufficiently large number, the oscillators emulate continuous fields. Since modes are qualitatively stable regimes in a system of differential equations there is also a need to “sculpt” basins of attraction when designing a task specific architecture.

Modes become distinct because they represent a stimulus driven segmentation of an ongoing stream of many possible behaviors. Presumably they begin to have meaning to us because they can be labeled by an external observer. Modes could either be pre structured by design or evolve naturally as a result of learning behaviors that adjust equation parameters. Certainly the former would be the case in the present research.

Micro actions are most easily thought of as direct motor or process control equations capable of effecting an environment outside the computer. Alternatively at higher levels they also could be parameter setting functions within the computer. Regardless, micro actions should be restricted to interactions between behaviors occurring only within the same level.

Specifying when control laws governing micro behaviors could fire, requires that a micro behavior be implemented as a pair of equations. One equation manages threshold conditions for action firing, the other generates the control trajectory that an action takes in its phase space.

•

$$\alpha = \sum_i \alpha_i M_i (\alpha, I(t)) + \delta$$

where α_i are the activations of slower behaviors

M is a dynamical law governing the micro action control

for mode i over time i.e. the trajectory

I(i) is the input either from sensor channels or oscillators

M behaviors are self actuated once their threshold is reached and

they are not suppressed by other M' s

δ is a decay term corresponding to leaky integration

C. Architecture Flexibility

Building an initial mode hierarchy could be a significant programming task. At the level of behaviors approach and design tools have been generated for a differential equation robot control operating system i.e. the Dual Dynamic Architecture discussed above. It may be possible to continue to modify that approach or the field computing analysis for use in the current work. [4-9] discuss the dual dynamic method in detail for a highly modular, asynchronous robotic platforms with small well defined actions. I extrapolate from their experience at least four factors that would constrain the present computer's behavior control or “operating” system:

1. Computing must be defined as for a robotic system in terms of goal directed behavior rather than classical numerical results. Behaviors are must also be ordered in hierarchies with competition between modes of behavior at the same level though not defined in terms of fields. Top level behaviors need to have associated with them sensor channels fed by outside information from the environment and other cell oscillators. only With each of the lowest level of behavior are associated dynamical equations which when fired act as controllers (micro tasks) changing the environment.
2. Each controller effects only subparts of the overall architecture. They must use primarily local information. The control system could test and change parameters through learning leading to changes in controls. Sensor channels will filter signals, and control actions should be added flexibly.
3. The entire architecture will be based on competing sets of dynamical equations. Hence the architecture will become a single global dynamical system.
4. The firing of a micro behavior corresponds to the actuation of a differential equation defining a trajectory in phase space that the controlled object receives as input. It simply follows its specific time evolution. The summing neuron determines the switch on or off.

Other complex functions such as information exchange might occur between cell models once frequency locking takes place. One example could be results of micro task firing. Another might be the propagation of stimuli information from the outside environment. One way to implement this information exchange between oscillators is to use phase effects. The phase differences become the information carrier. We could also have chosen the reverse (phase lock on oscillators with frequency as a carrier) which also has been implemented [24, 25]. However, analysis by

Izhikevich of simultaneous locking and communication requirements indicates phase along will not lock some oscillator networks but frequency will. Hence, frequency is proposed for the present architecture as the vehicle to lock associative memory, and phase differences are used to carry timing and other information.

In our initial experiments with oscillators we have focused on phase based canonical equations. The need to design a model's phase space for specific behaviors means there need to be methods to learn differential equation parameters by example rather than code them directly. Further, a requirement for frequency locking and unlocking (learning and forgetting) requires an unlocking mechanism - either algorithmic or parametric.

III. COMPARISON

It is useful as a summary to consider the behavior of the above architecture from an IT perspective and define what kind of "computation" occurs and what does not. As mentioned above there is research stating that a change in architecture to a continuous analog process may improve computability, i.e. the so called hypercomputation architectures [20,21]. In contrast to traditional programming which follows a Von Neuman action sequence, one does not program a dynamic analog computer for step by step behavior (unless training a behavior is viewed as programming).

Without adaptive learning, the above system like any analog system must be "rewired" for each type of job. This means that for new behaviors, one would change the interconnection structure between computing elements and reparameterize micro tasks and modes. If we assume over time a large number of micro tasks are learned or hard coded and they could phase lock into new mode hierarchies, there might be the potential of using learning algorithms to develop modes and higher level goals using external stimulation only (reinforcement learning). There exist learning rules for spike based neural networks (analogous to a pulse coupled net) which are an area of active research. Numerous examples exist which suggest approaches [32].

In the present conceptualization, oscillator stimulation patterns would have to be chosen in the frequency domain to resonate to other units with fairly specific and complex time and amplitude patterns. Once frequency lock is established, in effect a rewiring of communication takes place and energy exchanges occur through the reconciliation of phase differences between the units.

Which units will lock and which fire depends upon the values that define the differential equations modeling the oscillator neuron models. I do not yet understand exactly how oscillators will connect to a micro control action other than perhaps using a direct read out of a phase difference

magnitude or a change in a cell's energy field over time to exceed some firing threshold switch. Clearly, feedback will also be needed for non-trivial problems so signals can change with internal and external conditions. Assuring how stability would be maintained given such feedback has not yet been worked out.

A. Comparison of Computation Functions

A comparison with traditional computer functions proves useful in understanding how the proposed architecture differs from a traditional one. A traditional architecture includes a memory, logic system, control software, branching logic, a way to readout values from registers after or during computation, a communication method between elements, a math processor, a system clock, and an overall communication configuration. Most of these elements have some analog in the new architecture but function in very different ways and for different ends.

For example, a traditional computers primary function is to calculate a numeric or symbolic value. The present design emphasizes the generation of time based adaptive behaviors and control strings. It is in effect a goal seeking, analog behavior computer. Below are some comparisons between implementation of standard computing activities and their corresponding components in the proposed architecture:

Memory

Discrete static addresses in a traditional architecture are contrasted to a continuous, distributed, associative memory. Regions of memory are located by addressing in a standard system, oscillator locking/unlocking defines regions of associations to stimuli in the present system.

Logic

Binary logic is used versus analog logic much like a fuzzy logic or Lukasiewicz machines [36]. Analog logic is a result of branching behaviors at the bifurcation points of dynamical equations. (e.g. logistic equations [26], or resistance fields [36])

Operating System

Software is used in a traditional computer. In the proposed system, flow control occurs through competition among competing equations. Behavior results from the superposition of field potentials. Global task metrics determine goal achievement and stopping points.

Branching

Explicit code is used in a standard architecture versus bifurcation points of functions in the analog system.

Status Readout

Register values versus observation of system control behaviors

Communication

Normally through fixed channels in a standard computer and networking can be near order N^2 connections.

The present system communicates through phase differences in frequency locked cells. Non relevant signals pass through non resonant nodes as noise. This method has a proven potential of order N connections [14].

Math

A floating point processor computes math values in a standard computer. The architecture generates analog values and outputs continuous signals. Equations may need to be approximated numerically e.g. Runge Kutta methods, if simulated on existing computers. Ideally the values are a byproduct of circuit design and the equations merely model the dynamics of the materials so not numerical approximation is required. Signal values and their timing are determined by the attractor landscape and phase transition points.

Clock

Fixed and centralized versus non clocked and distributed. The new architecture is characterized by asynchronous control and distributed task execution.

Architecture

Predetermined functional elements process data flowing through system components. In the proposed architecture, dynamic, multi purpose equations “rewire” the architecture to perform different calculations. Computation is reconfigured continuously as a byproduct of inter cell communication and competition.

Figures 1 through 3 present graphics of how the computation flow is envisioned to occur. For the system to perform in real time without a clock, time coded events must exert influences somewhere in the system architecture. This creates a number constraints on design. Table 1 presents some of the more critical time sensitive components:

TABLE 1: CRITICAL TIME BASED BEHAVIORS:

1. The arrival of an event
2. The magnitude of the event
3. The availability of communication channels
4. The grouping speed of memory units into clusters
5. Recognition of time and magnitude patterns
6. Recall of patterns from memory
7. Permanence of memories or change rate
8. Adaptation of computing units
9. Inhibition of excited states to maintain stability

IV. STATUS OF CURRENT RESEARCH

Our first efforts at constructing parts of this architecture have focused on modeling differential equations associated with canonical cell models. To do this we have experimented at present with three types of cell models: chaotic logistic equations to implement binary logic [26], Wang Neurons to

study inhibition architectures, [24,25] and Izhikevich Neurons [14-18,23] to study communication and locking. We have also developed equation analysis tools using Matlab and XPPAUT [37]. To minimize adjusting the number of parameters we derived simplified phase forms of oscillator models. We also demonstrated parametric changes letting logistic equations operate as oscillators, damped oscillators, and bifurcation switches.

We have working demonstrations of frequency locking, selective communication, and phase locking. We have not yet demonstrated that burster equations can function as complex feature recognizers nor have we developed learning algorithms to parameterize differential equations based on sensor signals. Work by Smelyanskiy [ref] shows considerable promise as one way in which equation parameterization might be accomplished. At present we rely on trial and error observation of equation output signals as behavioral building blocks.

Summarizing our current progress we have:

1. Generated and tested phase locked homogeneous neurons with frequency selectivity
2. Created attractor based AND/OR/NOT/NOR circuits and two bit multiplier circuits composed of logistic neurons, 8 bit adder/subtractor, and 3 bit multipliers
3. Derived canonical phase equations which decompose a convolved signal and illustrate selective communication to other neurons in a network.

V. FUTURE DIRECTIONS

Implementing an oscillator based architecture will require significant resources in time, multidiscipline integration, and materials. In the interim, it is important to produce useful spinoffs of the above ideas to support ongoing or near future NASA mission requirements. Looking at parts of the above architecture and considering concept verifying demonstrations in the literature, it should be possible to extend some of the above ideas to illustrate:

1. More human like pattern recognition for large complex data sets
2. Dynamic associative data retrieval given partial information searches
3. Temporal and phase pattern learning
4. Ultra small multi function computing components
5. Communication architectures which scales at order N
6. A dynamic equation based processor element which can have multiple hardware functions by reprogramming parameters
7. Very rapid searching using function bifurcations
8. Autonomous generation of features through frequency locking
9. New psychological models of meaning
10. A strategy for computing with new materials

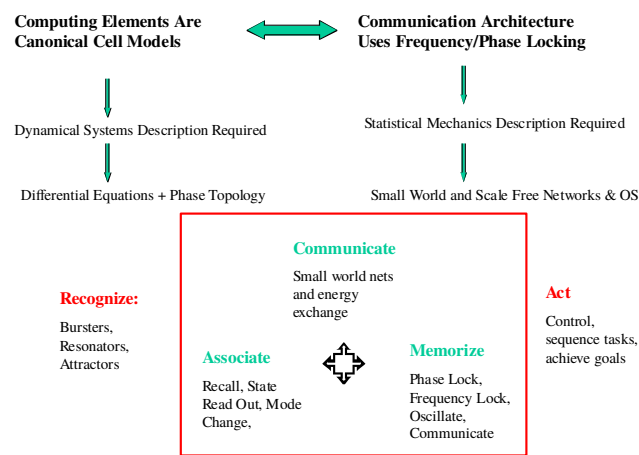
(quantum, nano, or biological)

What appears evident from the present analysis and the growing literature is that the use of coupled oscillator models as a alternative computational paradigm is likely to increase. How rapidly it can be moved into the mainstream of applied engineering will depend in large part on further development of basic research questions on non linear dynamics of large systems, network theory, and statistical mechanics. It is hoped that the present integrating architecture is a step in that direction.

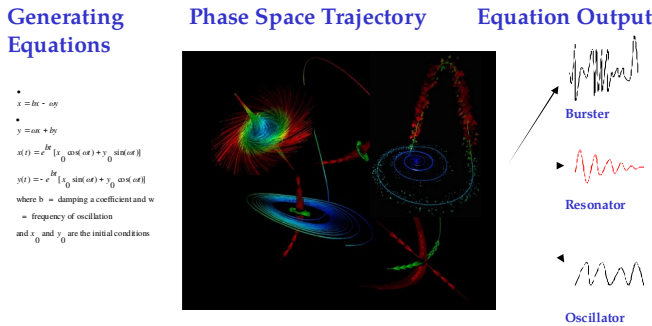
REFERENCES

- [1] Strogatz, S.H. "From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators", *Physica D* 143 (2000) 1-20.
- [2] Munakata, T., Sudeshna, S. and Ditto, W. "Chaos Computing: Implementation of Fundamental Logical Gates by Chaotic Elements", *IEEE Transactions on circuits and systems-I: Fundamental theory and applications*, vol 49, No. 11, 2002.
- [3] G. Ramsberger, "The human brain: Understanding the physical bases of intrapersonal communication," in *Intrapersonal communication: Different voices, different minds*, D.R. Vocate (Ed). (pp57-76) Erlbaum 1994.
- [4] Bredenkamp, A., Christall, Th., Gohring W., Gunther, H., Jaeger H., Kobialka, H., Ploger, B., Shooll, P., Sieberg, A., Streit, A., Verbeek Ch, and Wilberg, J. "Behavior Engineering with "Dual Dynamics" Models and Design tools", pp. 1-24. German National Research Center for Information Technology, Institute for Autonomous Intelligent Systems. Overview Presentation, 2002. <http://ais.gmd.de>
- [5] Large, E., Christensen, H.I., and Bajcsy, R. "Dynamic Robot Planning: Cooperation through Competition", GRASP Laboratory Report, University of Pennsylvania performed under NSF Grant IRI93-07126. 1996.
- [6] Iossifidis, I. and Steinhage, A. "Dynamical Systems: A Framework for Man Machine Interaction" Institut für Neuroinformatik, Ruhr-Universität, Bochum Germany.
- [7] Ijspeert A.J., Nakanishi J., Schaal S.: "Learning attractor landscapes for learning motor primitives," *Advances in Neural Information Processing Systems* 15 (NIPS2002), Becker S., Thrun S., Obermayer K. (Eds),
- [8] Ijspeert A.J., Nakanishi J., Schaal S.: "Trajectory formation for imitation with nonlinear dynamical systems," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, 2001, pp 752-757.
- [9] Ijspeert A.J., Nakanishi J., Schaal S.: Movement imitation with nonlinear dynamical systems in humanoid robots, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*, 2002, pp 1398-1403.
- [10] Hedger, L. "Analog Computation: Everything Old is New Again", *Research and Creative Activity*, Indiana University april 1998, Vol. XXI No.2. (Work by Prof Johnathan Mills)
- [11] E. Dubrova, Y. Jamal, and J. Mathew. "Non-silicon non-binary computing: Why not?" In *Proceedings of 1st Workshop on Non-Silicon Computation*, pages 23-29, Boston, USA, February 2002.
- [12] Maass, W. "Paradigms for Computing with Spiking Neurons" in Maass & Bishop eds. **Pulsed Neural Networks**, MIT Press Cambridge Mass. 1999. (shows spiking neurons can learn to approx functions and emulate logic circuits surveys temporal coding schemes)
- [13] Hoppensteadt, F. and Izhikevich, E. "Canonical Neural Models", Invited chapter Arbib, M.A. Ed. **Brain Theory and Neural Networks**, Second Edition MIT press, 2001. (derives canonical models from neuron families)
- [14] Izhikevich, E.M., "Resonance and Selective Communication Via Bursts in Neurons Having Subthreshold Oscillations", *BioSystems*, 2002. <http://www.nis.edu/users/Izhikevich> (shows neurons can selectively communicate through large numbers of loose network paths)
- [15] Izhikevich, E.M. "Weakly Pulse Coupled Oscillators, FM Interactions, Synchronization, and Oscillatory Associative Memory", *IEEE Transactions on Neural Networks*, Vol 10, No. 3, May 1999.
- [16] Hoppensteadt, F.C. and Izhikevich E.M. "Pattern Recognition Via Synchronization in Phase Locked Loop Neural Networks", *IEEE Transactions on Neural Networks*, Vol 11, No. 3, May 2000. (proposes a phase lock loop computer architecture)
- [17] Hoppensteadt F.C., and Izhikevich, E.M. "Oscillatory Neurocomputers with Dynamic Connectivity", *Physical Review Letters*, Vol. 82, No. 14, 5 April 1999. (demonstrates external forcing of weak network leads to associative properties based on Kuramotos model of networks)
- [18] Izhikevich, E. "Neural Excitability, Spiking, and Bursting", *International Journal of Bifurcation and Chaos*, Vol 10, No. 6, (2000), 1171-1266. (Detailed study of burster behaviors with math analysis)
- [19] Albert R. and Barabasi, A. "Statistical mechanics of complex networks", *Reviews of modern physics*, volume 74, Jan. 2002.
- [20] MacLennan, B.J. "Can Differential Equations Compute", Technical Report UT-CS-01-459, University of Tennessee, Knoxville, May 8, 2001.
- [21] MacLennan, B.J., "Transcending Turing Computability", Technical Report UT-CS-01-473, Dept. of Computer Science, University of Tennessee, Knoxville, Nov. 12, 2001.
- [22] MacLennan, B.J., "Field Computation: A Theoretical Framework for Massively parallel Analog Computation, Parts 1-IV, University of Tennessee, TR. CS -90-100, Dec 4, 1991.
- [23] Strogatz, S.H. "Synch" MIT press 2003.
- [24] Chen, K. and Wang, D. "A dynamically coupled neural oscillator network for image segmentation", *Neural Networks* 15 (2002) 423-439.
- [25] Wang, D. "Relaxation Oscillators and Networks", *Wiley Encyclopedia of Electrical and Electronics Engineering*, Wiley & Sons, vol. 18, pp. 396-405, 1999.
- [26] [Reference left incomplete in the draft. The author's note reads: "Ditto ref here, Izhikevich model used next, then Ermentrout ref."]
- [27] Izhikevich, E. "Class 1 Neural Excitability, Conventional Synapses, Weakly Connected Networks, and Mathematical foundations of Pulse-Coupled Models", *IEEE Transactions on Neural Networks*, Vol. 10, no.3, may 1999.
- [28] Strogatz, S., "Exploring complex networks", *Nature*, vol.410, pp. 268-276, 8 March 2001.
- [29] Mirollo, R. E., and Strogatz, S.H. "Synchronization of pulse coupled biological oscillators", *SIAM J. Appl. Math.* Vol. 50, No. 6, pp. 1645-1662, December 1990.
- [30] Izhikevich, E.M. "Resonate and Fire Neurons", *Neural Networks* 14, (2001) 883-894.
- [31] Hoppensteadt, F. and Izhikevich E.M. "Oscillatory Neurocomputers with Dynamic Connectivity", *Physical review letters* vol. 82, no. 14, 1999.
- [32] Rhouma, M., and Frigui, H., "Self Organization of Pulse Coupled Oscillators with Application to Clustering", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, No. 2, 2001.
- [33] Froemke, R.C. and Dan, Y. "Spike timing dependent synaptic modification induced by natural spike trains", *Nature*, 2002, 416, pp.433-438
- [34] MacLennan, B.J. "Continuous Computation and the Emergence of the Discrete" proceedings of the 2cd Appalachian Conference of Behavioral Neurodynamics: Processing in Biological Neural Networks, Radford University, Radford VA., Oct. 3, 1993.
- [35] MacLennan, B.J. "Field Computation in motor control", *Self-Organization, Computational Maps and Motor Control*, Pietro Morasso, ed. Elsevier-North Holland, 1996.
- [36] Mills J.W. "Programmable VLSI Extended Analog Computer for Cyclotron Beam Control", Indiana University Technical Report, 1995. <http://www.cs.indiana.edu/~jwmills/ANALOG.NOTEBOOK/kim/klm.html>
- [37] Ermentrout, B. "Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students", *SIAM* ISBN 0-89871-506-7, 2002.
- [38] Hoppensteadt F.C. , and Izhikevich, E.M., *Weakly Connected Neural Networks*, Springer, Applied Mathematical Sciences 126, 1997.

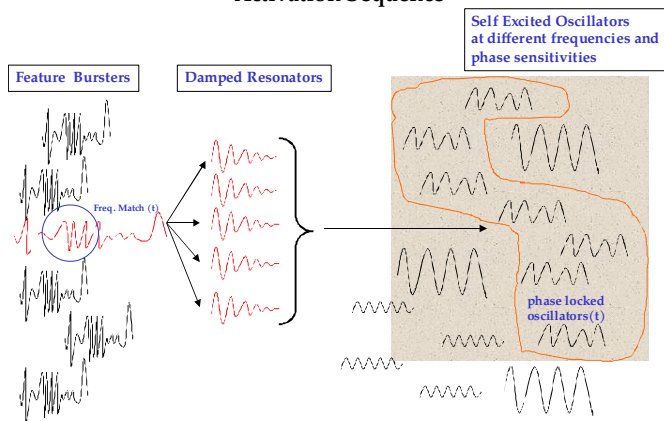
A BIC Architecture is a Jigsaw Puzzle



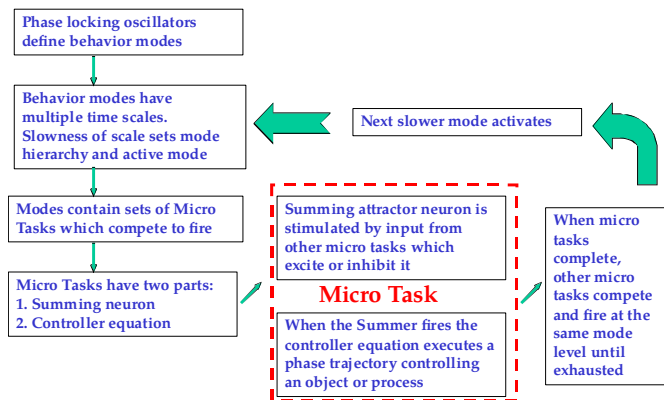
Generation of Bursters, Resonators, Oscillators



Activation Sequence



Oscillator Locking Defines Behavior Modes



Traditional Versus BIC Computation



	<i>Traditional Computer</i>	<i>Biological Inspired Computer (BIC)</i>
memory	static/dynamic cells	continuous, associative, locking/unlocking
logic	binary	analog (e.g. logistics or other chaotic functions)
control	software	parameter waves with task performance metrics
branch	explicit code lines	bifurcation points of functions
readout	register values	dynamic system states and observable actions
commo	via channels, order N squared	broadcast phase or frequency locking at order N
math	FP processor	determined by dynamic attractor landscape
clock	fixed and centralized	asynchronous and distributed task execution
architecture	set functional elements	dynamic, multi functional elements