One Step Forward: Periodic Computation in Dynamical Systems

Doug Rubino and Balaji Sriram

1 Abstract

The nonlinear dynamics of chaotic circuits generate a rich spectrum of signals. This observation suggests that these circuits could potentially provide an implementation of a novel framework of computation. In support of this hypothesis, computational complexity of the nervous system is achieved in large part through the nonlinear elements of electrically excitable membranes. In this study, we characterize the structure of the integral periodic components of signals generated by the chaotic inductor-diode (LD) circuit using a novel periodic decomposition. Specifically, we show that simple sinusoidal inputs, when passed through the LD circuit, can be used to store discrete, multiplexed periodic information. Further research could reveal principles of a periodic form of computation that can be implemented on simple dynamical systems with qualitatively complex behaviors. It is our hope that insights into the biological organizing principles of the nervous system will emerge from a precise, computational understanding of complex nonlinear systems such as the LD circuit.

2 Introduction

A dynamical system is abstractly defined as a series of coupled differential equations, which usually take some unspecified function as an input. The full characterization of a system can be thought of as the set of all possible trajectories in *phase space* over all possible inputs and initial conditions. Phase space is the vector function that relates the multiple variables of the system to their derivatives (*Izhikevich, 2007*). Dynamical systems that are not explicitly dependent on time are referred to as *autonomous systems*. Systems of interest are generally autonomous, and contain at least one fixed point that the system tends to, when given the null input (usually the zero function). In this case, the system is referred to as *resettable* (*Gold, 1972*).

Dynamical systems are often described with respect to the period (or lack thereof) of their trajectories in phase space, which result from a certain, usually periodic, inputs. In some cases, one particular variable of the system is causal, in a physical sense. This variable is then taken to be the output variable of the system. In the systems that describe the dynamics of the various neurons of the nervous system, voltage is taken to be the causal output variable of the system.

Given a periodic input, a system will usually tend to a *limit cycle*; a periodic trajectory through phase space that is of fixed length. Dynamical systems with dimension greater than two are, in general, capable of following aperiodic trajectories, even when given an input that is itself periodic. The systems that are capable of generating this kind of aperiodic behavior are referred to as *chaotic systems*. Understanding the various *routes to chaos* is one of the major goals of the relatively young field called *chaos theory*. Several routes to chaos have been identified, the most famous of which is the *period-doubling* route to chaos that is seen in dynamical systems with both discrete and continuous application of a non-invertible mapping function (*Feigenbaum, 1978*).

In this study, we approach the chaotic system in a novel way, using a computational perspective. There is a rich and well-established field of academic inquiry called *computability theory*, which is part of a more general field called *recursion theory*. Recursion theory is concerned with the dynamics of discrete, recursively defined functions, which include, but are not limited to, symbolic systems of logical inference and computational systems. The field takes its roots from the pioneering work of of Kurt Godel and David Hilbert, which concerned the completeness of *number theory* (arithmetic) as expressed in the inferential language of *first-order logic* (Smullyan, 1994). It is noteworthy that Godel's work is as relevant today as it was a century ago.

The earliest results in computability theory were first pioneered by Alan Turing, Alonzo Church and Stephen Kleene, among others. Their aims were to define precisely what computation is and to prove precisely what its limitations are. Computability theory is concerned with discrete systems, generally referred to as *machines*, following Turing's formalism. Through the lens of an interpreter, a machine can be capable of generating all the patterns that can be generated by some simplier machine, as well as having the capacity to generate other additional patterns. The powerful machine is said to have a higher *computational complexity*. There are machines that can generate the patterns of every other machine, and these special machines are referred to as *universal computers*, or more recently as just computers (Sipser, 2006). The term *computer* is colloquially used to refer to any physical device that, if given access to an infinite amount of memory, would then become a universal computer.

There are several significant similarities between chaotic systems and universal computers, which we use as the basis of our analogy. Computers are understood in terms of taking in a string of numbers as an input, changing their internal states in various ways, and then eventually halting (or not). Chaotic systems can be understood in terms of taking in a periodic input, displaying some transient behavior and then eventually converging a periodic output (or not). These phenomena are analogous to the computational notions of *computation time* (the transient period) and *halting* (converging to a periodic trajectory). The periodicities of the outputs are known to be integer values with respect to the period of a known periodic input, which is a fact that becomes intuitive after a bit of thought about trajectories and phase space. In this way, we can resolve a notion of discreteness with respect to the input. The discreteness of computational systems is of fundamental importance; it is the backbone underlying all results concerning isomorphisms between (equivalence of) computational systems. Discreteness also allows for the notions of computational complexity and universality to be well-defined.

We proceed in our analysis by characterizing the outputs of the LD circuit over some small but interesting input space, which consists only of sinusoids of various amplitudes and frequencies. The outputs from the system are taken to be the voltage fluctuations across the varactor. These output signals are decomposed into integral periodic components using a decomposition of our own design. In this way, an output is described in a manner that is analogous to a musical chord. We show that both the periodic components and the frequency ranges that evoke them contain fundamental aspects of discreteness. This provides a means of storing information in chaotic circuits and demonstrates implicit discretization in both the inputs and outputs of at least one chaotic circuit; lending credibility to our analogy between chaotic systems and universal computers.

3 Experimental Methods

3.1 Circuit Construction

Many types of non-linear circuits that show chaotic behavior have been described earlier: some of the most famous are the Linsay circuit (*Linsay, 1981*) and the Chua circuit (*Matsumoto, 1984*). We have implemented a modification of the Linsay circuit, chosen for its simplicity, ease of construction and dynamical complexity.

3.2 The Lindsay Circuit

The original Linsay circuit is a non-linear modification of the classical linear RLC circuit in which the capacitor is replaced by a non-linear varactor diode. A varactor diode is a variable capacitor, a circuit component in which the capacitance of the diode in reverse bias varies with the reverse bias voltage. A schematic of the Lindsay circuit is presented in **Figure 1**.

3.3 The LD Circuit

The kernel of our circuit consists of an inductor and a varactor, the same components included in the original Linsay circuit. The resistance, inductance and varactor used are specified below.

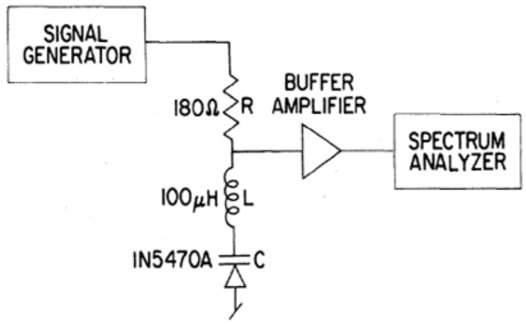

Figure 1: *The Lindsay Circuit. The circuit consists of an inductor and a diode called a varactor, a variable capacitor. The signal space consists of sine waves of fixed frequency and variable amplitude. The circuit is isolated with a single operational amplifier, labeled 'Buffer Amplifier' (Lindsay, 1983).*

3.4 Modifications

Any to query the state of any system will necessarily change the state of the system. Previous attempts to measure voltages along different parts of the circuit indicated that the voltages measured were dependent on our recording apparatus. Therefore several OpAmps were included to isolate the circuit and minimize the influence of experimental variability. The actual circuit is presented in **Figure 2**. The specifications of our LD circuit are:

Resistor:	100 Ohms
Inductor:	39 mH
Varactor:	IN4004
OpAmp:	AD825
Analog I\O:	NI PCI-6115

3.5 Choosing Operational Amplifiers

An OpAmp is a DC-coupled, high-gain, electronic voltage amplifier with differential inputs. **Figure 3** shows the schematic of a standard OpAmp. OpAmps are active components and require an external voltage source to function properly. The input-output relationship is given by the following formula:

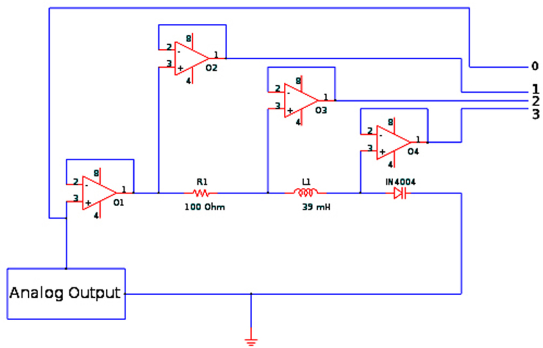

Figure 2: *The LD Circuit. This circuit is similar to the one studied by Paul Lindsay, though with several more OpAmps, digital signal generation and a different varactor.*

$$V_{out} = \begin{cases} G(V_- - V_+) & V_{s+}\ and\ V_{s-}\ are\ sufficient \\ V_{s+}\ or\ V_{s-} & Otherwise. \end{cases}$$

Then

$$\begin{array}{lcl} V_+ & : & non-inverting\, input \\ V_- & : & inverting\, input \\ V_{out} & : & output \\ V_{s_{+\backslash -}} & : & positive\backslash negative\, power\, supply \\ G & : & open-loop\, gain. \end{array}$$

Ideal OpAmps have the following properties:

1. Open loop gain G is infinite. Real OpAmps have finite but extremely high open loop gain (109). OpAmps are rarely used in their open-loop condition but usually one of many feedback modes the most frequent is the unity gain mode where the V_{out} is fed back into V_-.

2. Infinite Bandwidth. The input-output relationship holds for non-varying voltages and typically the gain falls rapidly with increasing frequency. Bandwidth is defined as that frequency range in which the gain is within 3dB of the maximum gain.

3. Infinite Slew rate. This is again a property of dynamic voltage inputs. This determines how fast the output voltage approaches the input voltage for a step change in the input voltage.

4. Infinite input impedance. Higher input impedances lead to lesser input current leading to lesser perturbation of the kernel.

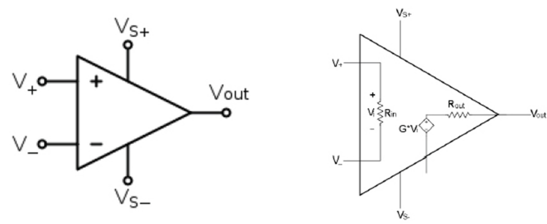

Figure 3: *The Operational Amplifier. Four identical OpAmps (depicted above) were used to isolate our inductor-diode (LD) circuit.*

3.6 The AD-825 Operational Amplifier

The AD-825 is general-purpose high-speed JFET amplifier having properties that make it ideal for isolating a circuit that is extremely sensitive to perturbations. Some facts about the AD-825 are available in **Figure 4**.

Parameter	Conditions
DYNAMIC PERFORMANCE	
Unity Gain Bandwidth	
Bandwidth for 0.1 dB Flatness	Gain = +1
−3 dB Bandwidth	Gain = +1
Slew Rate	R_{LOAD} = 1 kΩ, G = +1
OPEN-LOOP GAIN	V_{OUT} = ±10 V
	R_{LOAD} = 1 kΩ
	V_{OUT} = ±7.5 V
	R_{LOAD} = 1 kΩ
	V_{OUT} = ±7.5 V
	R_{LOAD} = 150 kΩ (50 mA Output)
INPUT RESISTANCE	
INPUT CAPACITANCE	
OUTPUT RESISTANCE	Open Loop

Figure 4: *Properties of the AD-825 Operational Amplifier.*

3.7 Choosing and Sampling the Input Space

We developed a measure of periodic complexity (**Analytical Methods**), and used our measure to choose an interesting input space. Once fixed, the input space was sampled in a variety of ways. For the purposes of this analysis it is fair to assume that the space was sampled randomly using the uniform distribution. The amplitude range of our inputs is 5-7 volts, and the frequency range of our inputs is 5-7 kHz.

4 Analytical Methods

4.1 Circuit Isolation and Stability

The goal of this section is to develop an analysis of stability that is straightforward. Though inputs are sampled over a two-dimensional space, for simplicity we can assume they are indexed by a single number j. Each input was repeated 10 times, and these trials are indexed by k. Let $std_k(f_{jk}(t))$ be the standard deviation across trials of the output signal f that was generated by input signal j at time t. We define the mean error as the average standard deviation across trials and time:

$$E(f_{jk}) = \langle std_k(f_{jk}(t)) \rangle_t$$

Two conditions were tested: the LD circuit without the four OpAmps (denoted C), and the LD circuit with the four OpAmps (denoted A). When $E_C(f_{jk}) - E_A(f_{jk}) > 0$, then we say that the OpAmps served to stabilize the circuit. Effectively, we are just testing whether the OpAmps serve to reduce variability in the LD circuit.

4.2 Periodic Complexity

We develop a novel measure of periodic complexity. Our goal in doing this is to choose an *interesting* region of the input space in a principled way. We imposed five intutive constraints on our complexity measure. They can be summarized as follows: *1*) the sine function is the simplest periodic function; *2*) periodic complexity is invariant to changes in phase; *3*) periodic complexity is invariant to changes in amplitude; *4*) periodic complexity is invariant to changes in height; *5*) if two signals have non-overlapping sinusoidal components (are *periodically independent*), the periodic complexity of their product is equal to the sum of their periodic complexities.

Formally, this requires a definition of *periodic independence*, the list of our five constraints, and a function C_p satisfying the constraints. We provide a function satisfying four of the five constraints, and outline how to veryify the fifth. We do not provide a proof of uniqueness, nor do we know if our function is unique. It does have the property of being intuitive and easy to calculate.

Periodic Independence Let ρ_1 and ρ_2 be two well-behaved functions (*bounded* and *differentiable everywhere*) and let $\mathcal{F}\{\rho\}$ be the Fourier transform of ρ. We say ρ_1 is periodically independent of ρ_2, and write $\rho_1 \perp \rho_2$ iff:

1. $|\mathcal{F}\{\rho_1\}| > 0 \rightarrow |\mathcal{F}\{\rho_2\}| = 0$.
2. $|\mathcal{F}\{\rho_2\}| > 0 \rightarrow |\mathcal{F}\{\rho_1\}| = 0$.

Periodic Complexity Let ρ, ρ_1 and ρ_2 be well-behaved functions. A function C_p is a measure of periodic complexity *iff* it satisfies the following conditions:

1. $\rho(t) = sin(t) \;\rightarrow\; C_p(\rho(t)) = 0.$
2. $\forall \theta \in \mathbb{R},\;\; C_p(\rho(t)) = C_p(\rho(t+\theta)).$
3. $\forall a \in \mathbb{R},\;\; C_p(\rho(t)) = C_p(a\rho(t)).$
4. $\forall b \in \mathbb{R},\;\; C_p(\rho(t)) = C_p(\rho(t) + b).$
5. $\rho_1 \perp \rho_2 \;\rightarrow\; C_p(\rho_1\rho_2) = C_p(\rho_1) + C_p(\rho_2).$

Theorem 4.1 (Periodic Complexity)**.** *Let ρ be a discretely measured timeseries, let $\mathcal{F}\{\rho\}$ be the discrete Fourier transform of ρ, and let $H(X)$ denote Shannon's entropy function. Then*

$$C_p(\rho) = H\left(\frac{|\mathcal{F}\{\rho\}|}{\sum\limits_{\omega>0}|\mathcal{F}\{\rho\}|}\right)$$

is a measure of periodic complexity. C_p is not defined for the constant function, which is, in general, only non-zero when $\omega = 0$. Following the standard derivation of the entropy function (Cover & Thomas, 1996), which an analogous case, we let $C_p(c) = 0$ for all $c \in \mathbb{R}$.

It is easy to see why C_p satisfies the first four constraints. 1*) If the function is the sine funcion, it's discrete Fourier transform has only one non-zero component. In this case, $\frac{|\mathcal{F}\{\rho\}|}{\sum|\mathcal{F}\{\rho\}|}$ is the delta distribution, which has an entropy of* 0*;* 2*) By considering only the amplitudes $|\mathcal{F}\{\rho\}|$, the measure is invariant to phase shifts;* 3*) Normalizing the amplitudes by dividing $|\mathcal{F}\{\rho\}|$ by $\sum|\mathcal{F}\{\rho\}|$, C_p is invariant to uniform changes in amplitude. The formal proof requires the fact that the Fourier transform is a linear transform; hence $\mathcal{F}\{a\rho\} = a\mathcal{F}\{\rho\}$;* 4*) The DC component is not considered in the measure, making it invariant to changes in height.*

5*) The last condition requires the relationship between multiplication in the time domain and convolution in the frequency domain. We need to prove that the covolution of periodically independent functions in the frequency domain will have a complexity that is the sum of the two individual complexities. This may not be true. Furthermore, the final constrain may not be required for a definition of periodic complexity. Therefore, our measure of periodic complexity, as well as our proposed solution, are both works in progress. Even so, our measure is sufficient for the purposes of this experiment.*

4.3 The Hilbert Factorization

The Hilbert factorization is a method of factorizing a well-behaved signal or time-series into phase and amplitude components, based on the following simple identity:

$$\begin{aligned} f &= Re\{f + ig\} \\ &= Re\{a\, e^{i\phi}\} \\ &= Re\{a\cos(\phi) + ia\sin(\phi)\} \\ &= a\cos(\phi). \end{aligned}$$

Choosing $g = H\{f\}$ has many analytically favorable properties. In this case, the time series a is called the analytic amplitude of f and ϕ is called the analytic phase of f. This will help us define the driving frequency of an output signal, which we show to be identical to the input frequency in all but a negligible number of cases in our input space (see **Results**). For more on the Hilbert transform see (Johansson, 1998).

4.4 Sinusoidal Normalization

We define a *sinusoidal normalization* as an invertible mapping of any well-behaved function to a function that is *more sinusoidal*. Specifically, the mapping projects f to the range [-1 1], and f can be reordered into the monotonic portion of the sine function. This will be made formal below.

The motivation behind the normalization procedure is to make our periodic decomposition general to signals that are all-positive or all-negative, as well as allowing our decomposition to be recursive. Recursivity is pivotal for interpreting signals in the context of computation, though is less significant to the results of this paper. In fact, insofar as this work is concerned, the sinusoidal normalization has approximately no impact on our results; we are analyzing signals which are already centered on the horizontal axis, and we do not include a characterization of the recursive components of our decomposition. Even so, it is worth mentioning for the sake of completeness.

The normalization function σ^{-1} is chosen to extend the analogy that links the relation between cosine and sine to the relation between a function and its hilbert transform. We start with the well know trigonometric identity:

$$\cos^2(t) + \sin^2(t) - 1 = 0.$$

We then ask for an invertible function σ^{-1} that minimizes the following equation for an arbitrary, well-behaved function f:

$$\sigma^{-1}(f)^2 + H\{\sigma^{-1}(f)\}^2 - 1.$$

The analogy between the two equations is straightforward if one thinks of $\sigma^{-1}(f)$ as analogous to the *cosine* function, and $H\{s^{-1}(f)\}$ as analogous to the *sine* function. Then

$$f = \sigma(\sigma^{-1}(f)).$$

We refer to $\sigma^{-1}(f)$ as the sinusoidal normalization of f. If f is a discrete time-series, we conjecture that 1) sorting the values of f; 2) mapping them one-to-one onto the values of the series between [sin(-pi/2) sin(pi/2)], and then 3) reording those new values back to their original configuration satisfies the minimization stated above. Solving this problem in the continuous case, as well as verifying our solution in the discrete case would most likely be difficult. Even without a formal solution, our inverible mapping does allow our decomposition to both be general and recursive, which serves our purposes. This is done at the expense of analytical rigor.

4.5 The Periodic Decomposition

The goal of our periodic decomposition is to take apart the output signal into a series of integral periodic components with respect to to driving frequency, which tends to be the frequency of the input (**Figure 8**).

The decomposition is multiplicative. While this was originally started for intuitive reasons, one should note that the period-doubling-like behavior observed in Paul Lindsay's original paper looks very similar to the successive power spectra that one gets when multiplying a series of coprime sinusoids (**Figure 5**). Here, one can see period doubling, as well as higher-order phenomena that are qualitatively similar to those observed in the powerspectra of signals from continuous chaotic systems. In light of this similarity, we derive a multiplicative decomposition of a signal into integral periodic components.

The first step of the decomposition makes use of the sinusoidal normalization. Let f denote an arbitrary, well-behaved, time-varing output function; a voltage signal across the varactor.

$f_t = \sigma(\sigma^{-1}(f_t)).$

This follows immediately from the fact that the sinusoidal normalization function was chosen to be invertible. Let $f'_t = \sigma^{-1}(f_t)$. Then

$f_t = \sigma(f'_t).$

The normalized function f' can then be decomposed using the hilbert factorization.

$f_t = \sigma(\alpha_t \cos(\phi'_t)).$

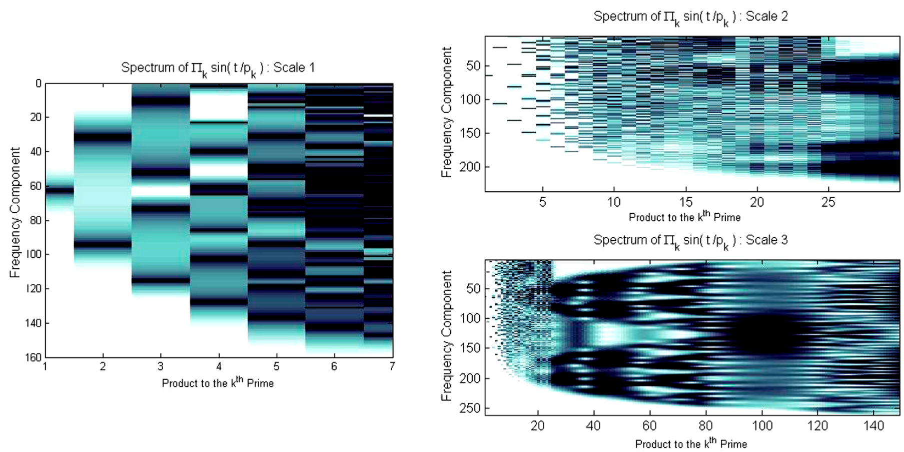

Figure 5: *Spectra of the successive multiplication of sinusoids of increasing periodicity. The three figures are different scales, but contain spectra from identical data. Product of the k^{th} prime period is on the horizontal axis, frequency is on the vertical axis, and power is on the color axis; black denotes high values and white denotes low values.*

A driving periodicity r_o can be extracted through polynomial fitting, so that ϕ_t can be represented in the following way:

$$f_t = \sigma(\alpha_t \cos(\frac{\phi_t}{r_o})),$$

where $\phi'_t = \frac{\phi_t}{r_o}$. ϕ_t is a function of phase, and can be unwrapped. In fact, the unwrapping procedure has already been done implicitly to extracting the driving periodicity. The unwrapped function will be *nearly* monotonic, and for the purposes of this derivation (and succictness), we will assume that it is monotonic. Let $\tau_t = \phi_t$ in order to emphasize that we are using an unwrapped, monotonic phase variable as a variable that represents relative time. α can then be stretched appropriately to be a function of τ. Therefore, we now have

$$f_t = \sigma(\alpha_\tau \cos(\frac{\tau}{r_o})).$$

With respect to τ, the cosine term is now a perfect sinusoid with period r_o. The rest of the decomposition concerns α, so we flip the equation around for convenience using the commutivity of multiplication:

$$f_t = \sigma(\cos(\frac{\tau}{r_o})\, \alpha_\tau).$$

α is an all-positive function, and due to the sinusoidal normalization, it happens to be in the range (0 2). Therefore, $log_2(\alpha)$ is a function that crosses zero.

Noting that $\alpha = 2^{log_2(\alpha)}$, we write

$$f_t = \sigma(\cos(\frac{\tau}{r_o})\, 2^{log_2(\alpha_\tau)}).$$

The next step is the trickiest and, perhaps, the most arbitrary. It is best described informally: i) first the power spectra of $log_2(\alpha)$ is computed; ii) the spectra is then parsed by local minima; iii) the center of mass of each amplitude is used to assign each partition to the closest period $n\, r_o$; and iv) the inverse transform of all partitions is taken. The signal corresponding to integral period $n\, r_o$ is denoted α_n. All unassigned, residual partitions are grouped into a single term α_r. We can then write

$$log_2(\alpha) = \sum_{n=1}^{N} \alpha_n \;+\; \alpha_r,$$

where N comes from the signal resolution. Let $R = 2^{\alpha_r}$. Using the fact that $2^{a+b} = 2^a\, 2^b$, we can write our signal as

$$f_t = \sigma((\cos(\frac{\tau}{r_o}) \prod_{n=1}^{N} 2^{\alpha_n})\, R).$$

The residual R is subject to recursive refactorization, but this is not included in our analysis. In general, R was found to be small, as almost all of the signal complexity is contained in periodic components with longer wavelengths than the driving periodicity. Each α_n is then factorized using the Hilbert factorization, to give $\alpha_n = \beta_n \cos(\phi_n)$. Therefore

$$f_t = \sigma((\cos(\frac{\tau}{r_o}) \prod_{n=1}^{N} 2^{\beta_n \cos(\phi_n)})\, R) \qquad =$$

$$f_t = \sigma((\cos(\frac{\tau}{r_o}) \prod_{n=1}^{N} (2^{\cos(\phi_n)})^{\beta_n})\, R).$$

$\cos(\phi_n)$ was then correlated with a sinusoid of the form $cos(\frac{\tau}{n\, r_o} + \theta)$ (using complex exponential multiplication). The Hilbert factorization of f, the values α_n and the correlation results can be seen in **Figure 6**.

We describe $\cos(\phi_n)$ in the following way:

$$\cos(\phi_n) = a_n \cos(\frac{\tau}{n\, r_o}) + R_n.$$

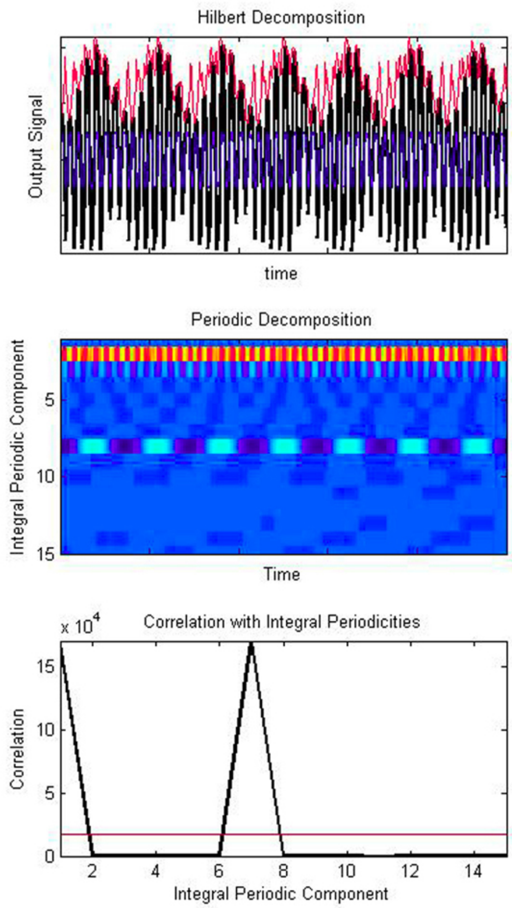

Figure 6: *The Periodic Decomposition. (Top) A single output signal, with the amplitude (red) and phase (blue) components of the Hilbert factorization. The red and blue signals multiply to the original output. (Middle) The components of one step of the periodic decomposition of the amplitude component. (Bottom) The correlation of each component with the appropriate integral periodicity. The red line represents ϵ, the cutoff parameter.*

Again, this decomposition can be done using multiplication with a complex exponential of the appropriate period. We let all terms $2^{\beta_n R_n}$ be absorbed by the multiplicative residual term R. Finally, we let

$$s_\tau = \cos(\frac{\tau}{r_o}) \ , \quad \text{and} \quad \rho_{n\tau} = 2^{a_n \cos(\frac{\tau}{n\, r_o})}$$

giving

$$f_t = \sigma((s_\tau \prod_{n=1}^{N} \rho_{n\tau}{}^{\beta_{n\tau}})\, R)$$

as the final form of our decomposition. Each $\rho_{n\tau}$ has an associated amplitude which, if below some threshold value ϵ, is excluded from the analysis.

It is noteworthy that this decomposition has a form that is similar to the prime factorizaton of an integer. If z is an integer, $s = sign(z)$, b_k are natural numbers, and p_k is the k^{th} prime, then there exist b_k such that

$$z = s \prod_k p_k^{b_k}.$$

This is one way of stating the *fundamental theorem of arithmetic*. This theorem has played a pivotal role in the development of the theory of computation; it is the substrate for *Godel numbering*, which was one of two pivotal insights required for Godel to prove his first and second incompleteness theorems. His methodology, rather than his results, are considered to be the first example of *programming* through arithmetic relations (Leary, 2000). This work has been followed up by others to solve important problems concerning the solutions of *diophantine equations* (Chaitin, 1987). Additionally, Godel's work provided Alan Turing with the inspiration to imagine the capabilities of his most highly-esteemed contraptions, which have since come to bear his name - Turing machines (Goldstein, 2006).

5 Results

5.1 Stability Analysis

The stability analysis confirmed that the OpAmps served to reduce the overall variability of the output voltages from the LD circuit (**Figure 7**). In the majority of cases, the outputs were more stable. There were also a significant number of cases where the variability remained unchanged. In a very small number of cases the outputs actually became more variable. Therefore, while the OpAmps do seem to help, the complexity of the changes they induce suggests that their inclusion may create a qualitatively different circuit.

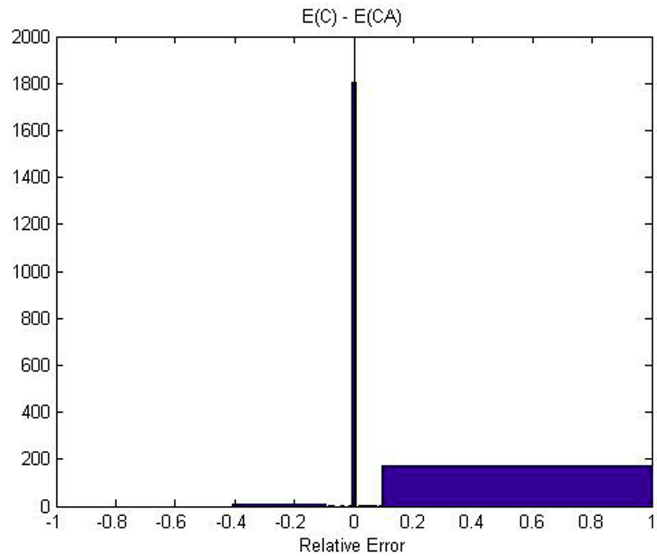

Figure 7: *Stability analysis for the LD circuit. The histogram shows the difference in output variability between the LD circuit with and without the four OpAmps.*

5.2 Properties of the Hilbert Factorization

One step in our periodic decomposition involves separating the signal into amplitude and phase components. We found that the phase component can be used to resolve the input frequency in the absence of prior knowledge (**Figure 7**). This holds true in all but a small minority of cases (> 100 of the $11,750$ signals analyzed).

5.3 Characterizing the Input Space

Given the qualitatively vast complexity of outputs, choosing an input space to study is somewhat arbitrary. We chose a space containing output signals which covered the full spectrum of periodic complexity. When viewed as a function of input amplitude, the periodic complexity of the signals is approximately randomly distributed. When viewed as a function of input frequency, there is well-defined structure, albeit intricate and unintuitive. Looking only at the periodic component with the maximal amplitude, one finds that there are four obvious scales of periodicity. They vary much more regularly with respect to input frequency than with respect to input amplitude (**Figure 8**).

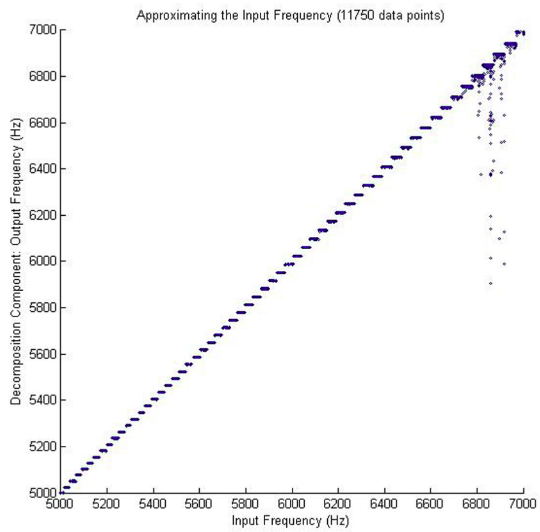

Figure 8: *Recovering the Input Frequency. The frequencies of the input sinusoids are plotted on the horizontal axis. The driving frequencies approximated using the Hilbert factorization are plotted on the vertical axis. 11,750 input/output pairs were used in this analysis.*

5.4 Periodic Discretization

The most interesting finding that came out of our exploratory characterization of the LD circuit is that non-overlapping (discrete), input frequency intervals evoke output signals with successive, discrete, periodic components (**Figure 10**). In other words, there is a natural input/output discretization in the LD circuit. This is certainly a desirable property for constructing a computational system, and supports our hypothesis that the principles of computation can be used to characterize complex systems including those like our LD circuit, which exhibit chaotic behavior.

Although discrete structure is clearly present in periodic components 23-28, there is still quite a bit of additional complexity across the input space. We are sampling a regime with a large number of periodic components, and for all intensive purposes, these outputs can be considered chaotic. While this may not actually be the case, the periodicity of the output signal is far far greater than the time over which we sampled the data; the period of a function is multiplicatively related to the periods of its individual components. Due to the observed complexity, regularity is only observed on average over an input range. Locally, quite a bit more complexity and discretization is apparent.

For instance, we observe striped patterns in each periodic component when

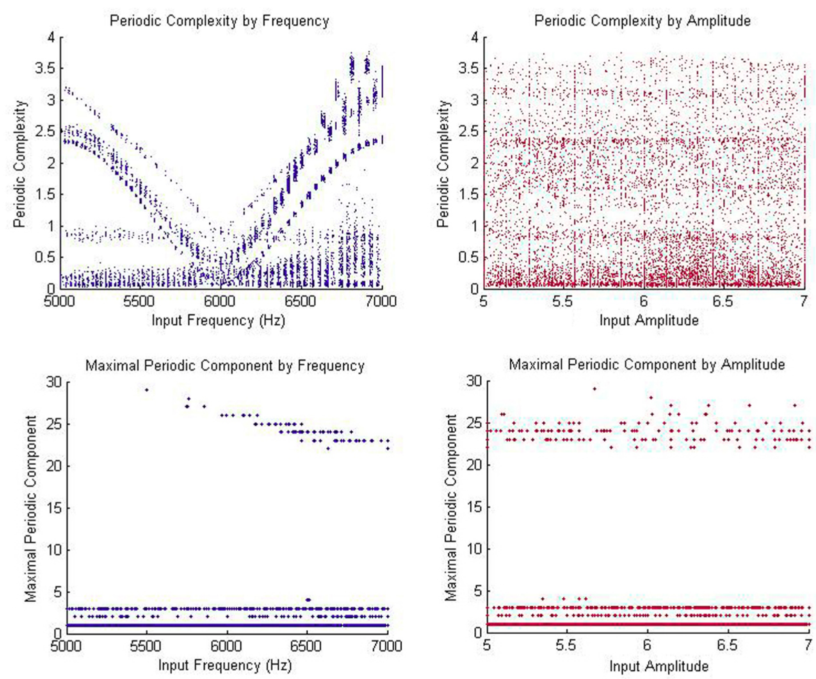

Figure 9: *Characterizing the input space. (Top) The periodic complexity of the outputs, plotted against input frequency and input amplitude. (Bottom) The largest integral periodic component, based on the periodic decomposition, plotted against input frequency and input amplitude.*

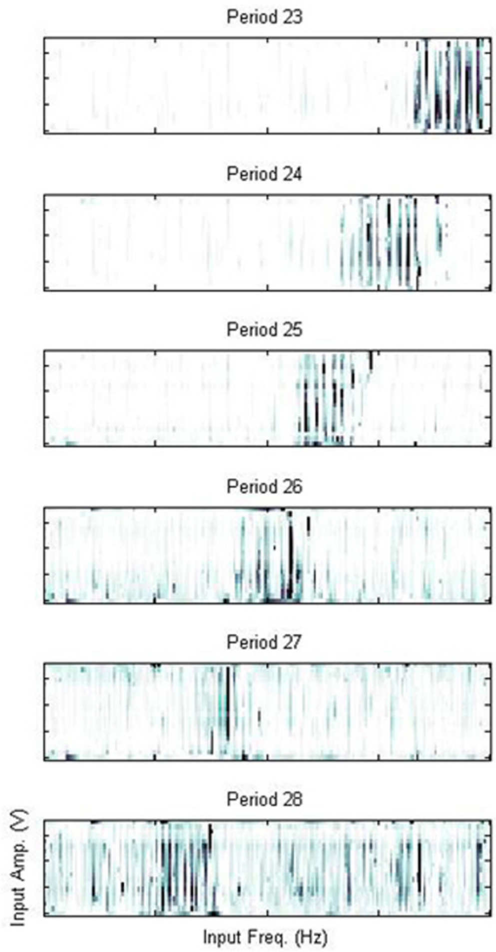

Figure 10: *Periodic Discretization. The series of panels represent the amplitudes of successive periodic components. Each panel shows the amplitude of the periodic component (black, high; while, low) across the entire input space. Input frequency is on the horizontal axis, input amplitude is on the vertical axis.*

viewed across the input space (**Figure 10**). While this may be attributed to either our interpollation method, or to the resolution of our data acquisition system, it is unlikely. In the first case, interpollation is a triagular procedure, and would be unlikely to produce stripes given that they are not explicitly supported by the distribution of the data. In the second case, input resolution would only serve to add redundancy to the inputs, and one would expect to see more regularity than might actually exist as a result of this redundancy. Therefore, the observed complexity may be a natural aspect of the system. If this is the case, our discrete regimes have further discretization embedded within them. This could serve to drastically increase the memory-storage capacity of the circuit.

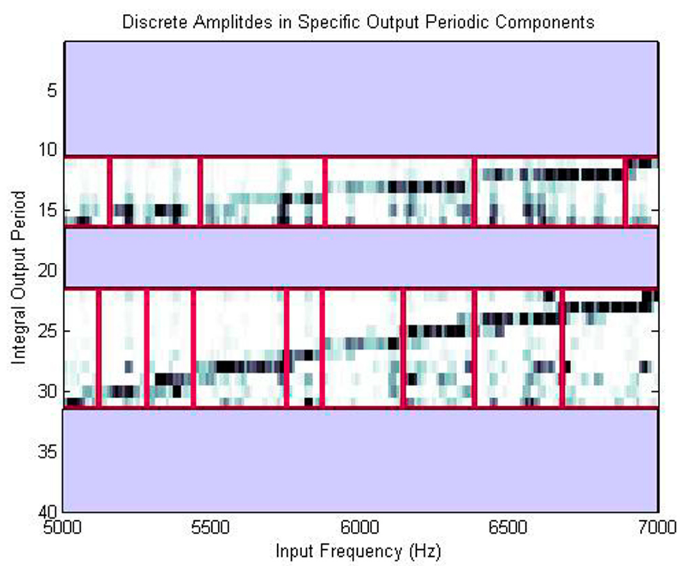

Figure 11: *Periodic Discretization across input frequency. Periodic amplitudes are averaged across input amplitudes for each input frequency. Input frequency is on the horizontal axis, and the integral periodic component is on the vertical axis. Red lines mark the approximate input boundaries of the discrete periodic components.*

Our second observation is that more than one discrete input/output band exists **Figure 11**. Our data show three bands, though the third is more probabilistic than the other two and is omitted for the sake of clarity. Regions shaded in grey represent integral periodicities with no apparent relationship to the input frequency. These too are omitted for the sake of clarity. The existence of multiple discrete output bands suggest that even in the case of simple sinusoidal inputs, the system is capable of generating discrete outputs with combinatorial complexity, as represented by the red bands. Each unique band represents a chord that can be distinguished from other output chords generated by different input intervals.

It is worth mentioning the fact that our input space consists only of sinusoids, oscillating at 3-5 kHz with amplitudes between 3-5 Volts. With respect to the space of all possible periodic functions, all reasonable input frequencies, and all reasonable input amplitudes, this space is extremely small; in fact virtually negligable. The number of discrete, combinatorial, periodic phenomena that can be generated by the LD circuit is potentially vast. We conjecure that through the appropriate combination of periodic inputs, one can uncover alegraic relationships between the periodic components which could be used to model such operations as set union and intersection. These operations would constitute the basis for constructing computational systems, in much the same way that boolean logic is used to construct computational systems using transistors.

6 Discussion

6.1 Frequency rather than Amplitude

Paul Lindsay's analysis of the inductor-varactor circuit revealed the period doubling route to chaos in a simple continuous-time system (Lindsay, 1981). Results concerning the period-doubling route to chaos in iterated, non-monotonic maps was pioneered and popularized by the seminal work of Mitchel Feigenbaum in the late seventies (Feigenbaum, 1978). In both of these systems, amplitude was the relevant variable that could be modulated to produce chaos. In Feigenbaum's work, the relevant variable was the height of a concave mapping function, whereas in Lindsay's work the relevant variable was the amplitude of the input sinusoid.

In contrast, we studied a low-frequency input space, chosen for its rich variation in periodic complexity. Surprisingly, the structured variation in periodic complexity was a function of the input frequency. With respect to amplitude, periodic complexity varied randomly, with a nearly uniform distribution (**Figure 9**). Because we characterize our output signals with respect to their driving frequency, which happens to be the input frequency, integral periodic components evoked by two different frequencies do not, themselves, correspond to the same periodicity. This is a fundamental difference between our analysis and those performed by Feigenbaum and Lindsay.

There is a good reason to normalize signals with respect to the input frequency. Should a signal converge to a periodic output, that output will have a period with an integral number with respect to the input period. Even so, were we to consider periodicity with respect to a fixed wavelength, it is possible that the various bands that we are observing actually represent the same output period. While this result would appear to challenge our conclusion, in fact it does not.

If upon varying the input frequency, one finds a periodic output component that remains fixed, then one has simply discovered another way in which the output of the system is discrete with respect to the continuous input. Either way, discretization plays a pivotal role in the interpretation of our results.

6.2 Exposing Discretization and Ignoring Chaos

Our analogy between computational systems and dynamical systems rests on the notion that a periodic output corresponds to a computer program that eventually halts. In the theory of computation, non-halting functions serve only as a means of characterizing the computational complexity of a system. This concept is epitomized in Turing's famous *halting problem*, whereby he proves that it is, in general, impossible to know whether a given system will halt without explicitly running it and waiting to see if it does. Past that, programs that do not halt are, in almost every respect, useless. A counter example to this would be an operating system, which halts only because the user forces it to. If our analogy is to have any merit, the field of nonlinear dynamical systems should not concern itself with the specific properties of chaotic trajectories and aperiodic output functions. Rather, their existence alone should denote a characterization of the system, and nothing more.

On the other hand, should a periodic input produce a periodic output, that output will have a discrete and therefore computational relationship to the input. This implicit discreteness, in the face of the vast mathematical oddity of uncountable infinities, should be the primary route to characterizing the behavior of dynamical systems. This path to understanding nonlinear systems allows for analogies to be made to mathematical structures like *the computer*, which are not only intuitive, but have changed the very nature of human interaction as we know it. An author by the name of Joseph Silverman published a book called '*The Arithmetic of Dynamical Systems*.' While the book is highly abstract, and is concerned with the iterative dynamics of mathematical structures that are far more intricate than periodic functions, he makes significant progress in decomposing dynamical systems within the context of arithmetic. It is exactly in this context that the formal and precise notion of *computation* is well-defined.

Further support for the fundamental relationship between dynamical systems and arithmetic can be shown explicitly by accepting only that the notion of periodicity (through the existence of limit cycles) is a fundamental requirement for understanding dynamical systems. This can be shown by simple example. If ρ_1 and ρ_2 are *well-behaved* periodic functions with rational (i.e. experimentally measurable) periodicities, any point-to-point combination of ρ_1 and ρ_2 will have a period that is the least-common multiple (LCM) of the individual periodicities of ρ_1 and ρ_2.

Here is the definition of the LCM function:

$$LCM(m,n) = \prod_k p_k^{min(m_k,n_k)},$$

where m and n are described with the respect to their prime factorizations:

$$m = \prod_k p_k^{m_k}.$$
$$n = \prod_k p_k^{n_k}.$$

Considering that the algebra of periodic functions is explicitly defined in the language of arithmetic, it is easy to see why the discretization present *number theory*, the language used to describe the relationships between the *natural numbers*, will have to play a fundamental role in the theory of nonlinear dynamical systems. In this light, we should should be exploiting the computational aspects of such systems, rather than exploring mathematical oddities such as chaos, which arise from mathematical attempts to describe sets that contain more numbers than one is even capable of symbolizing, even when given access to an infinite list of symbols (Leary, 2000)!

6.3 Implementation of the Periodic Decomposition

The periodic decomposition used for our analysis required three basical concepts: the sinusoidal normalization, two applications of the Hilbert transform, and band-pass filtering. Sinusoidal normalization aside, the other two components can be explicity realized in terms of circuit components. As stated above, the sinusoidal normalization was not even necessary for this analysis. Therefore, the computations performed on the computer can be similarly performed using a small number of well-precedented circuit components. This is a favorable property of our analysis, and lends credibility to a physical implementation of periodic computational devices. While we do not know if such devices would have practical use (in an economic sense), the authors of this paper feel that it is important for theory to be grounded in the substrate of reality, rather than in pure abstraction alone.

6.4 Potential Sources of Error

The bulk of work for this study primarily concerned the development of a novel analytical framework for approaching chaotic systems. Therefore, the potential sources of error rest mainly within our gaps in formality, which were pointed out in the various derivations above. The two major gaps; one in the derivation of *periodic complexity*, and the other in the conjectured solution to the *sinusoidal normalization* problem should have almost no impact on the results presented in this paper.

A very real source of potential error was in the methodology that we used to sample our input space. To avoid redundant and uninteresting data, we sampled areas that contained higher values of periodic complexity. So many sampling procedures were tested and appended that we feel it is fair to say that the data was approximately sampled randomly. Even so, this is not the case. Should there be any significant source of error in our results, it would rest on the fact that our data is biased towards sampling signals that are more periodically complex. The data was compiled using well over twenty different iterative sampling schemes, and the union of all such data is unlikely to fabricate the striking discretization that we observed between the inputs and the outputs. If anything, our sampling scheme would be more likely to contribute to the irregularities observed on the finer scale, as observed in **Figure 10** and **Figure 11**.

A second source of potential error is the limitations of our data acquisition system. The system outputs based on a discrete clock, and therefore certain frequencies will be misrepresented. Again, this is most likely to contribute to variations on a fine scale, and not at the level of discretization that we focused on in our analysis. Finally, OpAmps are known to be chaotic unto themselves, and there is no obvious way to describe the effect that they will have on the system, given that we lack a model of what the system *should* be doing. Even so, a complex system is a complex system, and the OpAmps merely served to provide us with a complex system that happened to be more stable, at least on average.

6.5 Conclusions

We describe a novel analysis that can be used to evaluate the computational properties of qualitatively complex circuits such as the LD circuit. It is our hope that a rigorous, computational understanding of nonlinear dynamics will provide a theoretical backbone for relating structure and function in the mammalian nervous system. Intuitively, we believe that the nervous system performs computations. Experimentally, we know that the nervous system is composed of neurons, which are known to be highly nonlinear and dynamic systems. Perhaps this work can serve as a first step in bridging the intuitive notion of computation in the nervous system with the experimentally verified existence of nonlinear dynamics in the nervous system. Independent of that, we also hope that the work serves as a refreshing departure from geometic analyses that rely on *routes to chaos* and *Hopf bifurcations* etc... It is only a first step, and a small one. But it is a promising one, nonetheless.

7 Works Cited

1. Chaitin, G.J. *Algorithmic Information Theory.* Cambridge University Press, New York NY. 1987.

2. Cover, T.M., Thomas, J.A. *Elements of Information Theory.* Second Edition. John Wiley & Sons Inc. Hoboken, NJ. 2006.

3. Feigenbaum, M.J. *Quantitative Universality for a Class of Nonlinear Transformations.* Journal of Statistical Physics, 19:1. 1978.

4. Gold, E.M. *System identification via state characterization.* Automatica. 8, 621-636. 1972.

5. Goldstein, R. *Incompleteness: The Proof and Paradox of Kurt Godel.* W.W. Norton & Company, Ltd. Castle House, London. 2006.

6. Izhikevich, E.M. *Dynamical Systems in Neuroscience.* The MIT Press. Cambridge, MA. 2007.

7. Johansson, M. *The Hilbert Transform.* Masters Thesis, Vaxjo University. 1998.

8. Leary, C.C. *A Friendly Introduction to Mathematical Logic.* Prentice-Hall Inc. Upper Saddle River, NJ. 2000.

9. Linsay, P.S. *Period-Doubling and Chaos in a Driven Anharmonic Oscillator.* Physical Review Letters, 47:19, 1349-52. 1981.

10. Matsumoto, T. *A Chaotic Attractor from Chua's Circuit.* IEEE Transactions, Circuit Systems, 31:12, 1055-58. 1984.

11. Silverman, J. *The Arithmetic of Dynamical Systems.* Springer Science. New York, NY. 2007.

12. Sipser, M. *Introduction to the Theory of Computation.* Second Edition. Thomson Course Technology. Boston MA. 2006.

13. Smullyan, R.M. *Diagonalization and Self-Reference.* Oxford University Press. Oxford, England. 1994.