UCLA

UCLA Electronic Theses and Dissertations

Title

Self-organized Criticality and Its CMOS Implementation

Permalink

https://escholarship.org/uc/item/49c9d4rh

Author

Chang, Hao-Yuan

Publication Date

2013

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Self-organized Criticality and Its CMOS Implementation

A thesis submitted in partial satisfaction
of the requirements for the degree Master of Science
in Electrical Engineering

by

Hao-Yuan Chang

2013

ABSTRACT OF THE THESIS

Self-organized Criticality and Its CMOS Implementation

by

Hao-Yuan Chang

Master of Science in Electrical Engineering
University of California, Los Angeles, 2013
Professor Kang Lung Wang, Chair

Self-organized criticality is a ubiquitous phenomenon that appears in many complex dynamical systems. During this special phase, the system experiences long-range correlation in space and in time. In order to understand its origin, we developed a method that can predict whether or not a stochastic dynamical system will be chaotic. We hypothesize that self-organized criticality and chaos originate from breaking the supersymmetries of the complex dynamical systems; therefore, the eigenvalue spectrum for the Fokker-Planck Hamiltonian should have pairs of complex eigenvalues on its imaginary axis. By applying the Fokker-Planck equation to the Chua oscillator, we show that the eigenvalues move closer to the imaginary axis as the system becomes chaotic. Also, we built a self-organized critical circuit using CMOS transistors. The circuit exhibits "avalanche" behaviors, in which groups of oscillators become out of synchronization together. The avalanche statistics show power laws in both avalanche size and inter-avalanche time.

The thesis of Hao-Yuan Chang is approved.

Robert N. Candler

Oscar M Stafsudd

Kang Lung Wang, Committee Chair

University of California, Los Angeles

2013

Table of Contents

Table of Figures

Acknowledgements

This work is partially supported by NSF IGERT: Clean Energy for Green Industry (CGI) – DGE-0903720 and the California NanoSystems Institute. The author would like to thank Dr. Igor V. Ovchinnikov, Professor Robert N. Schwartz, and Professor Kang Lung Wang for valuable discussions and comments.

1. Motivation—What is Self-organized Criticality (SOCr)?

Self-organized criticality is a ubiquitous phenomenon. Many complex dynamical systems such as the economy [1], the human brain [2], and traffic jams [3] all show SOCr behaviors. Complex dynamical systems usually comprise many coupled elements, and they often exhibit some collective behaviors that cannot be easily modeled using analytical equations. The U.S. power grid is an example of a complex dynamical system that can be driven to the SOCr phase: each generator is oscillating at 60 Hz, and they are connected through a network of transmission lines to various loads. Because of its enormous size, it is very difficult to accurately predict what will happen when additional loads and generators are connected dynamically to the grid. The power grid is often operating in its load limit due to the interaction between the electricity market and the demand [4]. A sudden surge of electricity demand in a hot summer afternoon can overload the transmission lines, causing them to force shout-down and resulting in regional blackouts. The statistics of the blackout sizes shows a power law relationship between probability and blackout size, which indicates that the power grid has been driven to the SOCr regime [5].

In the SOCr phase, dynamical systems have avalanches (e.g. blackouts in the case of the U.S. power grid) and show power law statistics both in avalanche size and in inter-avalanche time. When there is a power law in probability verses avalanche size, the system has long-range correlation between its elements. As a result, a small perturbation in one corner of the system can propagate to the opposite corner, and thus the system is sensitive to small changes. The surge in electricity demand can be treated as stochastic noise in a dynamical system. Because simulating a stochastic system at SOCr is computationally intensive, we need to develop an analytical tool that can predict whether or not the system will be driven to SOCr and chaos. In SOCr and chaos, oscillators become out of synchronization irregularly. This might cause surges in the voltage delivery and consequently system failure; as a result, blackouts occur. One potential application of such tool can be applied to stabilize the U.S. power grid: By modifying the bifurcation parameters of the system, we can move it further away from the SOCr and the chaos phase. In the case of the grid, one possible bifurcation parameter may be the connection strengths among different transmission lines. By stabilizing the grid (i.e. moving it away from chaos), we will be able to incorporate more renewable energy into our energy generation profile because many renewable energy sources such as wind and solar are intermittent. Their intermittency adds perturbations to the power system; these noises may drive the system into chaos and cause the system to crash.

Any dynamical system may be categorized into three different phases: Markovian, SOCr, and chaos [6]. In the Markovian phase, the system has predictable behaviors. Like a Markov process, the system has a finite number of states, in which it can traverse through. A depictive example would be a simple inductor-capacitor (LC) oscillator. If we track the voltage verses the current, we will see that the system circles around the origin with one frequency, and we can call this circle a state. On the other hand, a chaotic system does not have any well-defined state. As we will discuss in detail in Section 3, the Chua oscillator can be constructed by coupling a nonlinear negative resistor to the simple LC oscillator. The resulting circuit has three state variables: two voltages and a current. Hence, we can use Kirchhoff's laws to write down three differential equations (3.1)-(3.3) that describe the dynamics of the circuit. Due to the nonlinear negative resistor, the Chua oscillator has two strange attractors; the system traverses between them with an unpredictable trajectory. Consequently, there is no well-defined state. By modifying its bifurcation parameters, it is possible to bring a chaotic system such as the Chua oscillator into the Markovian regime, or vice versa, as we will discuss in Section 2. The bifurcation parameters are some coefficients in the three differential equations mentioned previously; they correspond to resistances and capacitances in the real circuit implementation.

We visualize the phase diagram of dynamical systems in Figure 1. In the deterministic limit in which noise is not present, there is a clear boundary between the chaotic phase and the Markovian phase (Figure 1a). However, when we add noise to the system, the SOCr regime will appear around the boundary between the two phases (Figure 1b). The width of the SOCr regime should increase as noise increases. SOCr can be understood as irregular transitions between the Markovian phase and the chaotic phase due to the present of noise; therefore, SOCr exhibits stop and go behaviors.

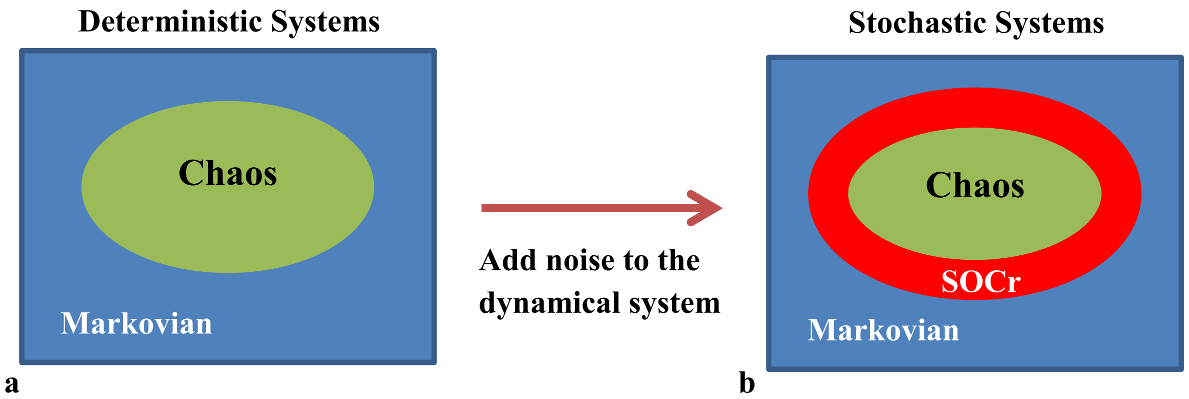

Figure 1 | General phase diagram for dynamical systems. In the deterministic case, a dynamical system can be in either the Markovian phase or the chaotic phase [6]. There is a clear boundary between the two phases. However, if we add noise to the system, a new intermittent phase appears. The SOCr phase exists at the boundary of the two phases; the width of the SOCr phase increases as noise amplitude increases. The SOCr phase can be understood as irregular transitions between chaos and Markovian due to noise; hence, SOCr exhibits intermittency.

In order to find the SOCr regime, we need to first find the boundary between chaos and Markovian when the noise is small. Then, we increase the noise to make the width of the SOCr regime wider. **This thesis addresses the following question: how do we predict chaos in stochastic dynamical systems?** In this work, we first use the eigenvalue pseudospectrum of the generalized Fokker-Planck equation to determine chaos. These eigenvalues also provide physical insights to the system dynamics. We will study the Fokker-Planck equation as described in Section 2.

Systems in SOCr show two distinct properties: intermittency and power law statistics. As mentioned previously, SOCr systems often have some quiet periods between each avalanche; hence, the intermittency. The size of the avalanches and the time between each avalanche both show a power law in their probability distributions, which represent long-range correlation in space and in time. We can utilize these two properties to build a solution generator for solving large optimization problems. Similar to conventional simulated annealing algorithms, our SOCr solution generator generates trial solutions. We then calculate the cost and provide feedback to the SOCr system in order to generate better solutions for faster convergence. In order to demonstrate this concept, we first engineer a dynamical system in SOCr using conventional circuit components. Clearly, **the second research question is: can we engineering a SOCr complex dynamical system using CMOS technology?** We will discuss the design of a 16x16 oscillator circuit and its avalanche statistics in Section 3.

2. Detection of Chaos Using the Fokker-Planck Equation

2.1. Introduction

Because SOCr lives between the Markovian phase and the chaotic phase, it is necessary for us to develop a tool that can predict when a dynamical system will approach chaos. Traditionally, whether or not a system is in chaos is determined by the system's trajectory in the phase space. The phase space contains all possible configurations of the system. A little perturbation in the initial condition makes a chaotic system follow a very different trajectory. Sometimes, it can cause the system to land on different strange attractors. A strange attractor is a point in the phase space where the system likes to circle around it. On the other hand, a Markovian system will follow a similar trajectory when its initial condition is perturbed slightly.

Mathematically, the Lyapunov exponent of a dynamical system measures the divergence of trajectories when the initial condition is slightly perturbed. The system is chaotic when the maximal Lyapunov exponent is positive; non-chaotic, negative. The Lyapunov exponent method only works for deterministic systems (i.e. systems without any noise or stochastic variable). Nevertheless, any real system experiences noise (such as perturbations in the power grid). **Thus, in this section, we aim to develop a tool for predicting if a stochastic dynamical system is chaotic.**

2.2. Theory

The Fokker-Planck equation is originally used to describe the Brownian motion of particles; particularly, it describes the time evolution of the probability density distribution [7]. We can generalize the Fokker-Planck equation to all dynamical systems, including chaotic systems with noise. Similar to Feynman's path integral method, the noise term can be described using a partition function and divided to a series of time slices; then, stochastic quantization techniques are used to rewrite the partition function of the noise in terms of the state variables of the dynamical system [8]. In the stochastic differential equations, the state variables and the noise interact with each other. The end result is the generalized Fokker-Planck Hamiltonian for all dynamical systems.

In the following section, we apply the Fokker–Planck equation to a general three dimensional system (i.e. any dynamical system with three state variables) via topological field theory (TFT). The time evolution of the Fokker-Plank probability wavefunction is governed by its Hamiltonian:

$$\frac{\partial}{\partial t}\Psi = -\hat{H}\Psi \tag{2.1}$$

, where

$$\hat{H} = \left[\hat{d}, \hat{\bar{d}}\right]_{+} = \hat{d}\hat{\bar{d}} + \hat{\bar{d}}\hat{d} \tag{2.2}$$

$$\hat{d} = \sum_{i} \chi^{i} \frac{\partial}{\partial \varphi^{i}} \tag{2.3}$$

$$\hat{\bar{d}} = \sum_{j} \frac{\partial}{\partial \chi^{j}} \left(-T \frac{\partial}{\partial \varphi^{j}} - A^{\,j}(\varphi)\right) \tag{2.4}$$

$\hat{H}$ is the Hamiltonian consists of the anti-Hermitian commutators for $\hat{d}$ and $\hat{\bar{d}}$ [9]. χ^{i} is the Faddeev-Popov ghost: It can be $\chi^{1}, \chi^{2}, or\ \chi^{3}$. φ^{i} is the state variable of the dynamical system: $\varphi^{1}, \varphi^{2}, or\ \varphi^{3}$. T is the temperature term that measures the amount of noise in the system. A^{j} is the flow term of the system. For Chua's circuit, A^{j} is constructed using the Kirchhoff's laws and is defined later in (2.47)-(2.49).

Next, we find the Hamiltonian of the Fokker-Planck equation for the wavefunction with full (three) ghosts. Faddeev-Popov ghosts arise during the quantization procedure of gauge theories [10, 11]. More specifically, they are mathematical tricks that alleviate the inconsistency when applying the path integral method. Because the dimension of our dynamical system is three, we will have three ghosts at most. The meaning of the three-ghost wavefunction is the probability of the system being at a specific location in the phase space. Although the system may have different trajectories, the probabilities for the system to follow different trajectories are averaged out and represented by the three-ghost wavefunction. Therefore, we expect the probability distribution to be stationary regardless of whether or not the system is in chaos. The wavefunction can be written as:

$$\mathit{\Psi} = \chi^1 {}^\wedge \chi^2 {}^\wedge \chi^3 \, f(\varphi) \tag{2.5}$$

, where ^ is the Wedge product operator for ghosts. In the following section, we will use the Einstein's notation, in which the summation symbol, $\sum$, on the left is omitted in the equation (i.e. χ^i in Einstein's notation means $\sum_i \chi^i \; = \chi^1 + \chi^2 + \chi^3$).

We first apply the $\hat{\bar{d}}$ operator onto the wavefunction and get:

$$\hat{\bar{d}}\mathit{\Psi} = \frac{\partial}{\partial \chi^j}\left(-T\frac{\partial}{\partial \varphi^j} - A^j\right)(\chi^1 {}^\wedge \chi^2 {}^\wedge \chi^3 \,)f(\varphi) \tag{2.6}$$

$$= \left(-T\frac{\partial}{\partial \varphi^j} - A^j\right)(\delta_j^1 \chi^2 {}^\wedge \chi^3 - \delta_j^2 \chi^1 {}^\wedge \chi^3 + \; \delta_j^3 \chi^1 {}^\wedge \chi^2)f(\varphi) \tag{2.7}$$

, where δ_j^i is one if and only if i = j; otherwise, it is zero. Then, we apply the $\hat{d}$ operator to the equation:

$$\hat{d}\hat{\bar{d}}\Psi = \chi^i \frac{\partial}{\partial\varphi^i}\left(-T\frac{\partial}{\partial\varphi^j} - A^j\right)\left(\delta_j^1\chi^2{}^\wedge\chi^3 - \delta_j^2\chi^1{}^\wedge\chi^3 + \delta_j^3\chi^1{}^\wedge\chi^2\right)f(\varphi) \tag{2.8}$$

$$= \left(-T\frac{\partial^2}{(\partial\varphi^i)^2} - \frac{\partial}{\partial\varphi^i}A^i\right)(\chi^1{}^\wedge\chi^2{}^\wedge\chi^3\,)f(\varphi) \tag{2.9}$$

The second term in the Hamiltonian (2.2) is zero because the product of two identical ghost sectors is zero. That is:

$$\hat{d}\Psi = \chi^i\frac{\partial}{\partial\varphi^i}(\chi^1{}^\wedge\chi^2{}^\wedge\chi^3\,)f(\varphi) \tag{2.10}$$

$$= \frac{\partial}{\partial\varphi^i}(\delta_i^1\chi^1{}^\wedge\chi^1{}^\wedge\chi^2{}^\wedge\chi^3 + \delta_i^2\chi^2{}^\wedge\chi^1{}^\wedge\chi^2{}^\wedge\chi^3 + \delta_i^3\chi^3{}^\wedge\chi^1{}^\wedge\chi^2{}^\wedge\chi^3)f(\varphi) \tag{2.11}$$

$$= \frac{\partial}{\partial\varphi^i}(0)f(\varphi) = 0 \tag{2.12}$$

Consequently, the final Hamiltonian for the three-ghost wavefunction is:

$$\hat{H} = \left[\hat{d},\hat{\bar{d}}\right]_+ = \sum_i\left(-T\frac{\partial^2}{(\partial\varphi^i)^2} - \frac{\partial}{\partial\varphi^i}A^i\right)(\chi^1{}^\wedge\chi^2{}^\wedge\chi^3\,)f(\varphi) \tag{2.13}$$

The Fokker-Planck Hamiltonian has two terms. The first term is the diffusion term, analogous to any diffusion process. The second term is the Lie derivative, which represents the flow of the dynamical system. The flow of the system is determined by the differential equations that govern its dynamics.

We also derive the Fokker-Planck equation for the two-ghost wavefunctions. The two-ghost wavefunctions have the meaning of "conditional probability density" because they are missing a ghost. The two-ghost wavefunction is measuring where the system is likely to be when the third dimension is determined. Because the three-ghost wavefunction is stationary in time, it is not useful for determining if a system is in chaos or not. Therefore, we expect the two-ghost

Hamiltonian to be a better indicator because the two-ghost wavefunction may represent the probability of transitioning between different attractors. Furthermore, because our Hamiltonians commute with the exterior derivative operator, $\hat{d}$, any non-zero eigenvalue will come in pairs in consecutive ghost sectors. That is some eigenstates for the two-ghost Hamiltonian are also eigenstates for the three-ghost Hamiltonian. Similarly, some eigenstates for the one-ghost Hamiltonian are also eigenstates for the two-ghost Hamiltonian. As a result, we have a higher chance to find purely imaginary eigenvalues in the two-ghost Hamiltonian rather than in the three-ghost case.

Because operating on the two-ghost wavefunction is significantly more complicated than the three ghost case, we would like to first expand the Hamiltonian operator in order to obtain a simpler form.

$$\hat{H} = \left[\hat{d}, \hat{\bar{d}}\right]_{+} = \hat{d}\hat{\bar{d}} + \hat{\bar{d}}\hat{d} \tag{2.14}$$

$$= \chi^{i} \frac{\partial}{\partial \varphi^{i}} \frac{\partial}{\partial \chi^{j}} \left(-T \frac{\partial}{\partial \varphi^{j}} - A^{j}(\varphi)\right) + \frac{\partial}{\partial \chi^{j}} \left(-T \frac{\partial}{\partial \varphi^{j}} - A^{j}(\varphi)\right) \chi^{i} \frac{\partial}{\partial \varphi^{i}} \tag{2.15}$$

By expanding this equation and rearranging the order, we obtain:

$$= -T \frac{\partial^{2}}{\partial \varphi^{i} \partial \varphi^{j}} \chi^{i} \frac{\partial}{\partial \chi^{j}} - \frac{\partial}{\partial \varphi^{i}} A^{j}(\varphi) \chi^{i} \frac{\partial}{\partial \chi^{j}} - T \frac{\partial^{2}}{\partial \varphi^{j} \partial \varphi^{i}} \frac{\partial}{\partial \chi^{j}} \chi^{i} - A^{j}(\varphi) \frac{\partial}{\partial \varphi^{i}} \frac{\partial}{\partial \chi^{j}} \chi^{i} \tag{2.16}$$

The two terms that involve the temperature, T, can be grouped together:

$$= -T \frac{\partial^{2}}{\partial \varphi^{i} \partial \varphi^{j}} \left[\chi^{i} \frac{\partial}{\partial \chi^{j}} + \frac{\partial}{\partial \chi^{j}} \chi^{i}\right] - \frac{\partial}{\partial \varphi^{i}} A^{j}(\varphi) \chi^{i} \frac{\partial}{\partial \chi^{j}} - A^{j}(\varphi) \frac{\partial}{\partial \varphi^{i}} \frac{\partial}{\partial \chi^{j}} \chi^{i} \tag{2.17}$$

We know that the term in the bracket is a Kronecker delta:

$$\left[\chi^i, \frac{\partial}{\partial\chi^j}\ \right]_+ = \left[\chi^i\frac{\partial}{\partial\chi^j} + \frac{\partial}{\partial\chi^j}\chi^i\right] = \delta^i_j \tag{2.18}$$

Because of this Kronecker delta, the first term in (2.17) will be one if and only if i = j. Thus, we can use index i to replace every index j in the first term:

$$\widehat{H} = -T\frac{\partial^2}{(\partial\varphi^i)^2} - \frac{\partial}{\partial\varphi^i}A^j(\varphi)\chi^i\frac{\partial}{\partial\chi^j} - A^j(\varphi)\frac{\partial}{\partial\varphi^i}\frac{\partial}{\partial\chi^j}\chi^i \tag{2.19}$$

The first term in (2.19) is the familiar diffusion (Laplacian) term. In order to further expand (2.19), we rewrite the Kronecker delta from (2.18) as:

$$\frac{\partial}{\partial\chi^j}\chi^i = \delta^i_j - \chi^i\frac{\partial}{\partial\chi^j} \tag{2.20}$$

Substitute (2.20) into (2.19), we obtain:

$$\widehat{H} = -T\frac{\partial^2}{(\partial\varphi^i)^2} - \frac{\partial}{\partial\varphi^i}A^j(\varphi)\chi^i\frac{\partial}{\partial\chi^j} - A^j(\varphi)\frac{\partial}{\partial\varphi^i}(\delta^i_j - \chi^i\frac{\partial}{\partial\chi^j}) \tag{2.21}$$

By expanding the third term and replace the index j with i (due to the Kronecker delta), we get:

$$= -T\frac{\partial^2}{(\partial\varphi^i)^2} - \frac{\partial}{\partial\varphi^i}A^j(\varphi)\chi^i\frac{\partial}{\partial\chi^j} - A^i(\varphi)\frac{\partial}{\partial\varphi^i} + A^j(\varphi)\frac{\partial}{\partial\varphi^i}\chi^i\frac{\partial}{\partial\chi^j} \tag{2.22}$$

And finally, we group similar terms:

$$= -T\frac{\partial^2}{(\partial\varphi^i)^2} - \left(\frac{\partial}{\partial\varphi^i}A^j(\varphi) - A^j(\varphi)\frac{\partial}{\partial\varphi^i}\right)\chi^i\frac{\partial}{\partial\chi^j} - A^i(\varphi)\frac{\partial}{\partial\varphi^i} \tag{2.23}$$

Next, we use this Hamiltonian to operate on the two-ghost wavefunction. We can write the two-ghost wavefunction as:

$$\Psi = \Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1 \tag{2.24}$$

The two ghosts Hamiltonian in (2.23) can be breaking down to three parts:

$$\hat{H}\Psi = \hat{H}_1\Psi + \hat{H}_2\Psi + \hat{H}_3\Psi \tag{2.25}$$

, where
$$\hat{H}_1 = -T\frac{\partial^2}{(\partial\varphi^i)^2} \tag{2.26}$$

$$\hat{H}_2 = -\left[\frac{\partial}{\partial\varphi^i}A^j(\varphi) - A^j(\varphi)\frac{\partial}{\partial\varphi^i}\right]\chi^i\frac{\partial}{\partial\chi^j} \tag{2.27}$$

$$\hat{H}_3 = -A^i(\varphi)\frac{\partial}{\partial\varphi^i} \tag{2.28}$$

We can then calculate $\hat{H}_1\Psi$, $\hat{H}_2\Psi$, and $\hat{H}_3\Psi$ separately and put the results together in the end. $\hat{H}_1\Psi$ and $\hat{H}_3\Psi$ are easy to calculate.

$$\hat{H}_1\Psi = -T\frac{\partial^2}{(\partial\varphi^i)^2}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \tag{2.29}$$

$$\hat{H}_3\Psi = -A^i(\varphi)\frac{\partial}{\partial\varphi^i}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \tag{2.30}$$

$\hat{H}_2\Psi$ is more difficult because we are taking derivatives against different ghost sectors:

$$\begin{aligned}
-\hat{H}_2\Psi = &\left[\frac{\partial}{\partial\varphi^1}A^1(\varphi) - A^1(\varphi)\frac{\partial}{\partial\varphi^1}\right]\chi^1\frac{\partial}{\partial\chi^1}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \\
&+\left[\frac{\partial}{\partial\varphi^1}A^2(\varphi) - A^2(\varphi)\frac{\partial}{\partial\varphi^1}\right]\chi^1\frac{\partial}{\partial\chi^2}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \\
&+\left[\frac{\partial}{\partial\varphi^1}A^3(\varphi) - A^3(\varphi)\frac{\partial}{\partial\varphi^1}\right]\chi^1\frac{\partial}{\partial\chi^3}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \\
&+\left[\frac{\partial}{\partial\varphi^2}A^1(\varphi) - A^1(\varphi)\frac{\partial}{\partial\varphi^2}\right]\chi^2\frac{\partial}{\partial\chi^1}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \\
&+\left[\frac{\partial}{\partial\varphi^2}A^2(\varphi) - A^2(\varphi)\frac{\partial}{\partial\varphi^2}\right]\chi^2\frac{\partial}{\partial\chi^2}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1] \\
&+\left[\frac{\partial}{\partial\varphi^2}A^3(\varphi) - A^3(\varphi)\frac{\partial}{\partial\varphi^2}\right]\chi^2\frac{\partial}{\partial\chi^3}[\Psi_{12}\chi^1{}^\wedge\chi^2 + \Psi_{23}\chi^2{}^\wedge\chi^3 + \Psi_{31}\chi^3{}^\wedge\chi^1]
\end{aligned} \tag{2.31}$$

$$+\left[\frac{\partial}{\partial\varphi^3}A^1(\varphi)-A^1(\varphi)\frac{\partial}{\partial\varphi^3}\right]\chi^3\frac{\partial}{\partial\chi^1}[\Psi_{12}\chi^1{}^\wedge\chi^2\ +\Psi_{23}\chi^2{}^\wedge\chi^3\ +\Psi_{31}\chi^3{}^\wedge\chi^1]$$

$$+\left[\frac{\partial}{\partial\varphi^3}A^2(\varphi)-A^2(\varphi)\frac{\partial}{\partial\varphi^3}\right]\chi^3\frac{\partial}{\partial\chi^2}[\Psi_{12}\chi^1{}^\wedge\chi^2\ +\Psi_{23}\chi^2{}^\wedge\chi^3\ +\Psi_{31}\chi^3{}^\wedge\chi^1]$$

$$+\left[\frac{\partial}{\partial\varphi^3}A^3(\varphi)-A^3(\varphi)\frac{\partial}{\partial\varphi^3}\right]\chi^3\frac{\partial}{\partial\chi^3}[\Psi_{12}\chi^1{}^\wedge\chi^2\ +\Psi_{23}\chi^2{}^\wedge\chi^3\ +\Psi_{31}\chi^3{}^\wedge\chi^1]$$

After $\chi^i\frac{\partial}{\partial\chi^j}$ operates on the wavefunction, we obtain:

$$\begin{aligned}
-\widehat{H}_2\Psi=&\left[\frac{\partial}{\partial\varphi^1}A^1(\varphi)-A^1(\varphi)\frac{\partial}{\partial\varphi^1}\right][\Psi_{12}\chi^1{}^\wedge\chi^2\ +0\ +\Psi_{31}\chi^3{}^\wedge\chi^1]\\
&+\left[\frac{\partial}{\partial\varphi^1}A^2(\varphi)-A^2(\varphi)\frac{\partial}{\partial\varphi^1}\right][0\ -\Psi_{23}\chi^3{}^\wedge\chi^1\ +0]\\
&+\left[\frac{\partial}{\partial\varphi^1}A^3(\varphi)-A^3(\varphi)\frac{\partial}{\partial\varphi^1}\right][0\ -\Psi_{23}\chi^1{}^\wedge\chi^2\ +0]\\
&+\left[\frac{\partial}{\partial\varphi^2}A^1(\varphi)-A^1(\varphi)\frac{\partial}{\partial\varphi^2}\right][0+0-\Psi_{31}\chi^2{}^\wedge\chi^3]\\
&+\left[\frac{\partial}{\partial\varphi^2}A^2(\varphi)-A^2(\varphi)\frac{\partial}{\partial\varphi^2}\right][\Psi_{12}\chi^1{}^\wedge\chi^2\ +\Psi_{23}\chi^2{}^\wedge\chi^3\ +0]\\
&+\left[\frac{\partial}{\partial\varphi^2}A^3(\varphi)-A^3(\varphi)\frac{\partial}{\partial\varphi^2}\right][0+0\ -\Psi_{31}\chi^1{}^\wedge\chi^2]\\
&+\left[\frac{\partial}{\partial\varphi^3}A^1(\varphi)-A^1(\varphi)\frac{\partial}{\partial\varphi^3}\right][-\Psi_{12}\chi^2{}^\wedge\chi^3\ +0\ +0]\\
&+\left[\frac{\partial}{\partial\varphi^3}A^2(\varphi)-A^2(\varphi)\frac{\partial}{\partial\varphi^3}\right][-\Psi_{12}\chi^3{}^\wedge\chi^1\ +0\ +0]\\
&+\left[\frac{\partial}{\partial\varphi^3}A^3(\varphi)-A^3(\varphi)\frac{\partial}{\partial\varphi^3}\right][0\ +\Psi_{23}\chi^2{}^\wedge\chi^3\ +\Psi_{31}\chi^3{}^\wedge\chi^1]
\end{aligned}\tag{2.32}$$

Now, we collect all the terms in (2.32) that will affect the time evolution of the first two-ghost wavefunction, Ψ_{12}. That is we collect all the terms that have $\chi^1{}^\wedge\chi^2$ in them.

$$
\begin{aligned}
-\widehat{H}_2\Psi_{12} = &\left[\frac{\partial}{\partial\varphi^1}A^1(\varphi) - A^1(\varphi)\frac{\partial}{\partial\varphi^1}\right]\Psi_{12} \\
&+\left[\frac{\partial}{\partial\varphi^1}A^3(\varphi) - A^3(\varphi)\frac{\partial}{\partial\varphi^1}\right](-\Psi_{23}) \\
&+\left[\frac{\partial}{\partial\varphi^2}A^2(\varphi) - A^2(\varphi)\frac{\partial}{\partial\varphi^2}\right]\Psi_{12} \\
&+\left[\frac{\partial}{\partial\varphi^2}A^3(\varphi) - A^3(\varphi)\frac{\partial}{\partial\varphi^2}\right](-\Psi_{31})
\end{aligned}
\tag{2.33}
$$

All four terms in (2.33) can be simplified further. As an example, we take the first term in (2.33) and expand it as follow:

$$
\left[\frac{\partial}{\partial\varphi^1}A^1(\varphi) - A^1(\varphi)\frac{\partial}{\partial\varphi^1}\right]\Psi_{12} = \frac{\partial}{\partial\varphi^1}(A^1(\varphi)\Psi_{12}) - A^1(\varphi)\left(\frac{\partial}{\partial\varphi^1}\Psi_{12}\right) \tag{2.34}
$$

We use the product rule to expand the first term in (2.34) because both A^1 and Ψ_{12} are functions of φ.

$$
= \left(\frac{\partial}{\partial\varphi^1}A^1(\varphi)\right)\Psi_{12} + A^1(\varphi)\left(\frac{\partial}{\partial\varphi^1}\Psi_{12}\right) - A^1(\varphi)\left(\frac{\partial}{\partial\varphi^1}\Psi_{12}\right) \tag{2.35}
$$

$$
= \left(\frac{\partial}{\partial\varphi^1}A^1(\varphi)\right)\Psi_{12} \tag{2.36}
$$

We perform the same procedure for the other three terms in (2.33) and obtain:

$$
\begin{aligned}
-\widehat{H}_2\Psi_{12} = &\left(\frac{\partial}{\partial\varphi^1}A^1(\varphi)\right)\Psi_{12} + \left(\frac{\partial}{\partial\varphi^2}A^2(\varphi)\right)\Psi_{12} - \left(\frac{\partial}{\partial\varphi^1}A^3(\varphi)\right)\Psi_{23} \\
&- \left(\frac{\partial}{\partial\varphi^2}A^3(\varphi)\right)\Psi_{31}
\end{aligned}
\tag{2.37}
$$

We again perform the same procedure for the other two-ghost wavefunctions (i.e. Ψ_{23} and Ψ_{31}) in (2.32).

$$-\widehat{H}_2\Psi_{23} = \left(\frac{\partial}{\partial\varphi^2}A^2(\varphi)\right)\Psi_{23} + \left(\frac{\partial}{\partial\varphi^3}A^3(\varphi)\right)\Psi_{23} - \left(\frac{\partial}{\partial\varphi^2}A^1(\varphi)\right)\Psi_{31} - \left(\frac{\partial}{\partial\varphi^3}A^1(\varphi)\right)\Psi_{12} \tag{2.38}$$

$$-\widehat{H}_2\Psi_{31} = \left(\frac{\partial}{\partial\varphi^3}A^3(\varphi)\right)\Psi_{31} + \left(\frac{\partial}{\partial\varphi^1}A^1(\varphi)\right)\Psi_{31} - \left(\frac{\partial}{\partial\varphi^3}A^2(\varphi)\right)\Psi_{12} - \left(\frac{\partial}{\partial\varphi^1}A^2(\varphi)\right)\Psi_{23} \tag{2.39}$$

By adding (2.29), (2.30), (2.37), (2.38), and (2.39) together, we can explicitly write out the final Hamiltonian in (2.25):

$$\widehat{H} = \begin{bmatrix} -\frac{\partial}{\partial\varphi^1}A^1 - \frac{\partial}{\partial\varphi^2}A^2 + \widehat{B} & \frac{\partial}{\partial\varphi^1}A^3 & \frac{\partial}{\partial\varphi^2}A^3 \\ \frac{\partial}{\partial\varphi^3}A^1 & -\frac{\partial}{\partial\varphi^2}A^2 - \frac{\partial}{\partial\varphi^3}A^3 + \widehat{B} & \frac{\partial}{\partial\varphi^2}A^1 \\ \frac{\partial}{\partial\varphi^3}A^2 & \frac{\partial}{\partial\varphi^1}A^2 & -\frac{\partial}{\partial\varphi^3}A^3 - \frac{\partial}{\partial\varphi^1}A^1 + \widehat{B} \end{bmatrix} \tag{2.40}$$

, where
$$\widehat{B} = \sum_i -A^i\frac{\partial}{\partial\varphi^i} - T\frac{\partial^2}{(\partial\varphi^i)^2} \tag{2.41}$$

$$\begin{pmatrix} \frac{\partial}{\partial t}\Psi_{12} \\ \frac{\partial}{\partial t}\Psi_{23} \\ \frac{\partial}{\partial t}\Psi_{31} \end{pmatrix} = -\widehat{H}\Psi = -\widehat{H}\begin{pmatrix}\Psi_{12} \\ \Psi_{23} \\ \Psi_{31}\end{pmatrix} \tag{2.42}$$

The three-ghost (2.13) and the two-ghost (2.40) Hamiltonians are true for any system with three state variables. The only change we need to make is to plug in different $A^i(\varphi)$ (i.e. the flow terms) for different dynamical systems.

Chaotic and SOCr systems are known to exhibit long-range correlations [12-14]. Furthermore, the Goldstone's theorem states that if the continuous symmetry is spontaneously broken, there exist low-energy excitations known as the Goldstone modes, which are massless particles that represent long-range correlations [15-17]. Under the construct of the Faddeev-Popov procedure, which we built our Fokker-Planck Hamiltonian upon, the system will have a BRST-symmetry when the expectation value of the correlator is zero [18-20]. As a result, in order to break the symmetry, the expectation values of the eigenstates must be nonzero when any d-exact operator acts on them [9].

$$< i\,|\hat{O}(\varphi)|i >\neq 0 \tag{2.43}$$

,where i is an eigenstates and $\hat{O}$ is the symmetry (correlation) operator that is d-exact. d-exact means that the operator is the result of commuting another operator with $\hat{d}$. In our case, because our Fokker-Planck Hamiltonian is d-exact, we can use it to replace the symmetry operator in (2.43):

$$\hat{O} = \hat{H} = \left[\hat{d}, \hat{\bar{d}}\right]_{+} \tag{2.44}$$

Consequently, in order to have long-range correlation in chaos and SOCr, our Hamiltonian should have nonzero eigenvalues, which represent the states that break the symmetry. These nonzero eigenvalues will come in complex conjugate pairs because all the elements in our Hamiltonian matrix are real numbers.

In addition, the real part of a complex eigenvalue represents exponential decay of the wavefunction; it damps its eigenvector to zero as time progresses. The only eigenstates that can survive time evolution are the ones that have no real component. Thus, **we hypothesize that the Fokker-Planck equation has purely imaginary eigenvalues for a chaotic system** because these eigenvalues not only break symmetries but also survive time evolution**.** On the other hand,

we should not see purely imaginary eigenvalues for non-chaotic systems. As a result, we may use the eigenvalue spectrum to determine if a stochastic dynamical system is in chaos or not.

2.3. Methods

In order to prove that chaotic systems have purely imaginary eigenvalues, we calculate the eigenvalue pseudospectrum for the Chua oscillator when it is in chaos. Then, we compare it with the pseudospectrum for the non-chaotic case. If our hypothesis is correct, complex eigenvalues should move closer to the imaginary axis when the Chua oscillator is in chaos. For the reasons mentioned before, we will most likely to see this phenomenon in the two-ghost wavefunction. Because the two-ghost wavefunctions are missing a ghost (i.e. similar to missing a dimension), the wavefunctions may represent the probability of transitioning between the attractors. The pair of imaginary eigenvalues creates oscillations in time for the wavefunctions. Therefore, the oscillations can be understood as transitions between different attractors.

The pseudospectrum is a measure of the numerical stability of the eigenvalues. Using the common notation in the Literature, we define the pseudospectrum as [21]:

$$\Lambda(H) = \{z \in C : \|(zI - H)^{-1}\| \geq \varepsilon^{-1}\} \tag{2.45}$$

, where z is a complex number that represents a location in the eigenvalue spectrum; z can take any value. Λ is the resulting pseudospectrum, which comprises contours that circle the areas where the right hand side of (2.45) is satisfied. I is the identity matrix which has the same dimension as H. H is our Fokker-Planck Hamiltonian; this definition of the pseudospectrum is true not only for our Hamiltonian but for any matrix in general. ε is the norm (e.g. sum of squares) of all the perturbations added to the Hamiltonian.

In practice, in order to find the pseudospectrum, we first define a grid on the eigenvalue spectrum and calculate $||(zI - H)^{-1}||$ for each point, z, on the grid. z is a complex number that corresponds to its location in the complex plane. If z happens to be an eigenvalue of H, $(zI - H)$ is zero. After taking the norm, we obtain a value for each point on the grid. Thus, using these values, we construct a contour plot in which each contour represent a specific level of noise, ε.

Equivalently, if we define the norm as the 2-norm, (2.45) can be rewritten into the following equation [22]:

$$\Lambda(H) = \{z \in C\colon \sigma_{\min}(zI - H) \geq \varepsilon^{-1}\} \tag{2.46}$$

, where $\sigma_{\min}$ is the smallest singular value of the matrix (zI – H). As a result, instead of calculating the norm for each point on the grid, we perform the singular value decomposition (SVD) of the matrix (zI – H) for every grid point. This is a very computationally intensive task: if the Hamiltonian has N elements, the computational effort for finding the pseudospectrum is $O(N^3)$ for each point in the pseudospectrum [23]. We will discuss a more efficient method later.

The resulting pseudospectrum will have many contours that represent different amounts of noise, ε. They represent where the eigenvalues could potentially shift to when the corresponding amount of noise, ε, is added to the Hamiltonian. Again, ε represents the sum of squares of all the perturbations on each element in the Hamiltonian; it is a concise way to measure the total amount of perturbations. In Figure 2c, for example, the pink line is for the case with a noise amplitude of $10^{-0.6}$ added to the Hamiltonian; in this case, the eigenvalues will at most move to the boundary of the pink line. We use the pseudospectrum instead of the regular eigenvalue spectrum because our grid for the Fokker-Planck Hamiltonian is coarse. In the ideal scenario, we will have infinite amount of grid points in each direction of the 3-D phase space. However, the amount of available computer memory limits the maximum size of the Hamiltonian that we can store. Furthermore, it is exponentially more difficult to diagonalize the Hamiltonian as its size increases, and the run time of the program becomes unreasonably long. Hence, our grid cannot be too dense. The Hamiltonian is perturbed a little due to the coarse grid. The effect of the grid on the Hamiltonian can be treated as a noise, ε. Because of this noise, the eigenvalues do not touch the imaginary axis directly. In order to cope with this shortcoming, we use the pseudospectrum to determine where the eigenvalues can move to with a specific level of noise.

We conduct our numerical experiment with the following steps:

1. Using the finite element method, we divide the three dimensional phase space into a grid with 40 points in each dimension. Hence, there are 64,000 total points representing the 3-D phase space. The boundaries of our phase space are -12 and 12 in each of the dimensions (i.e. x, y, and z). The wavefunctions will have a value on each of the grid point. The distance between two adjacent grid points (i.e. dx, dy, or dz) is 0.5854. The temperature coefficient for the Laplacian term, T, is defined as $5*(dx^2+ dy^2+ dz^2)$, which is 5.1398.
2. Construct the Hamiltonian for the two-ghost wavefunction according to (2.40). The Hamiltonian is a large two dimensional matrix (192,000x192,000) with 192,000 elements in each dimension (40 x 40 x 40 grid points * 3 wavefunctions). Therefore, the Hamiltonian has $3.6864*10^{10}$ total elements. Construct one Hamiltonian for the non-chaotic parameter set (α=1) and another one for the chaotic case (α=10). The Chua oscillator can be described using three differential equations [24], which are also the flow terms (A^1, A^2, and A^3) in our Fokker-Planck Hamiltonian (2.40). They are:

$$A^1 = -\dot{x} = -\alpha\big[y - x - \big(m_1 x + 0.5(m_0 - m_1)(|x+1| - |x-1|)\big)\big] \quad (2.47)$$

$$A^2 = -\dot{y} = -x + y - z \quad (2.48)$$

$$A^3 = -\dot{z} = \beta y \quad (2.49)$$

, with β=14.2886, m_0=-1/7, m_1=2/7, α = 1 for non-chaotic, and α = 10 for chaotic. Figure 2a and Figure 2b show the time evolutions of the Chua oscillator when α = 1 and α = 10, respectively. We clearly see that when α = 1, the system converges to one of attractor permanently. Hence, it is in the Markovian phase. When α =10, the system has an unpredictable trajectory that traverses between the two strange attractors; therefore, it is in chaos.

3. Use the Eigtool MATLAB program developed by Wright and Trefethen from Oxford University to calculate the pseudospectrum for our Hamiltonian [23]. It finds the pseudospectrum by using the following procedures: First, use the built-in eigs function in MATLAB to compute the eigenvalues for our Hamiltonian. The eigs function uses the Arnoldi process on a sparse matrix in order to find the n smallest eigenvalues. The Arnoldi algorithm projects the original Hamiltonian onto the Krylov subspace and produces a Hessenberg matrix, which is much smaller than the original Hamiltonian but contains the relevant information about the eigenvectors of interest. Because the pseudospectrum of the original Hamiltonian can be approximated using the pseudospectrum of the Hessenberg matrix, we calculate the singular values for the Hessenberg matrix on each grid point. We can further facilitate the calculations of the singular values by performing a QR factorization followed by an inverse Lanczos iteration, which makes the overall algorithm O(N^2) for each grid point in the

pseudospectrum [23]. O(N^2) is significantly faster than the speed of calculating the pseudospectrum directly from the original Hamiltonian, O(N^3).

4. Check if the eigenvalues move towards the imaginary axis as the system becomes chaotic.

2.4. Results

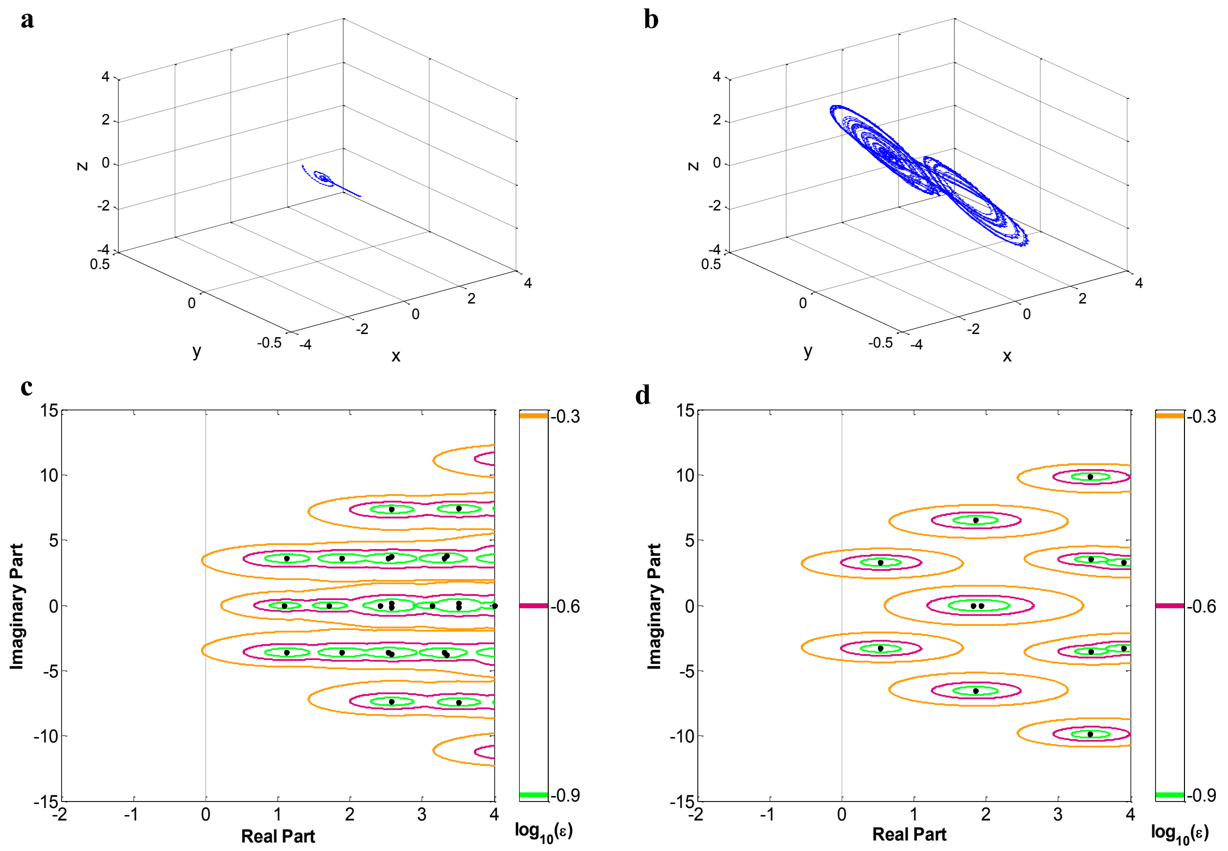

Figure 2 | Pseudospectrum of the two-ghost Fokker-Planck Hamiltonian for the Chua oscillator. We construct the Hamiltonian using (2.40), (2.47), (2.48), and (2.49) with these parameters: β=14.2886, m_0=-1/7, and m_1=2/7. α is the bifurcation parameter. **a**, For the non-chaotic case, we set α to one and plot the time evolution of the Chua oscillator in the phase space. The system starts at (0.1, 0.1, 0.1) and converges to one of the attractors. **b**, For the chaotic case, we set α to ten and plot the time evolution. The system moves between the two attractors and never stops. **c**, Pseudospectrum for the non-chaotic case (α=1). The black dots are the eigenvalues. The contours represent where the eigenvalues can shift to with a given amount of noise, ε, in the Hamiltonian. The color bar denotes different noise levels. **d**, Pseudospectrum for the chaotic case (α=10). The pair of complex eigenvalues moves closer to the imaginary axis, creating more oscillation in the time evolution. On the other hand, the eigenvalues on the real axis shift to the right.

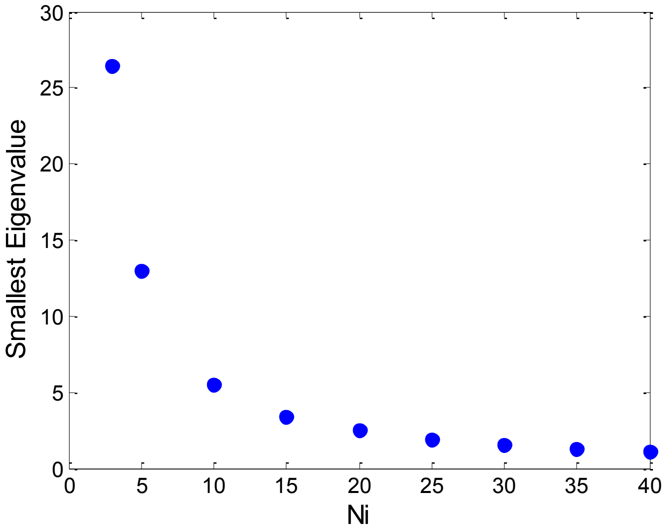

Figure 3 | Effect of grid size on the smallest eigenvalue. We track the smallest eigenvalue in **Figure 2c** (the non-chaotic case) while increase the number of points in each dimension of the 3-D grid in the phase space. The smallest eigenvalue always lies on the real axis regardless of the grid size. With a denser grid, the eigenvalues move closer to the imaginary axis.

2.5. Discussion

Comparing the pseudospectrum of the Chua oscillator when it is in Markovian (Figure 2c) and in chaos (Figure 2d), we observe that the complex eigenvalues move to the left when the Chua oscillator becomes chaotic. The color codes on the right of the figures represent different noise levels in log scale, and the black dots are the eigenvalues calculated by diagonalizing the Hamiltonian using the eigs routine from MATLAB. The contours represent where the eigenvalues can possibly shift to with noise, ε, added to the Hamiltonian. In particular, the pink contour is on the right of the imaginary axis when the system is in Markovian (Figure 2c) and touches the axis when it is in chaos (Figure 2d), showing that it is possible for the eigenvalues to touch the axis when the system is chaotic.

In Figure 2d for the chaotic case (α=10), the eigenvalues on the real axis move to the right as the system become chaotic. These purely real eigenvalues do not create oscillations in time; hence, they can be viewed as the “stable states” in the system. As they move further to the right, their corresponding eigenvectors damp out faster, leaving the pair of complex eigenvalues to their left. As a result, the pair of complex eigenvalues dominates the dynamics because they damp slower than any other state. In Figure 2b, the time evolution of the system shows irregular transitions between the two attractors when the system is in chaos.

On the contrary, in Figure 2c for the Markovian case (α=1), the real eigenvalue is slightly to the left of the complex pair. As a result, the pair of complex eigenvalues damp out faster. In Figure 2a, the time evolution of the system shows a non-chaotic behavior as expected; the system converges to one of the attractors and never escapes from it. However, with a significant amount of noise (e.g. the orange line: $\varepsilon=10^{-0.3}$), the contour for the complex pair touches the imaginary axis. This means that the system is able to become chaotic when noise is strong enough. Consequently, increasing the noise might expand the SOCr regime or the chaotic regime. This phenomenon agrees with the concept of “noise induced chaos” as reported in the Literature [25].

In order to show that the eigenvalue indeed resides on the imaginary axis, in Figure 3, we plot the amplitude of the smallest eigenvalue for the non-chaotic case as we increase the number of grid points (N_i) in each dimension of the 3-D phase space. The smallest eigenvalue is roughly inversely proportional to the grid size. As a result, it is significantly more difficult to get closer to the imaginary axis as the gird size increases beyond 40. In the case of infinite grid size, the eigenvalue may reside on the imaginary axis.

Although in deterministic systems, we can clearly define the boundary between chaos and Markovian, it is more difficult to find the exact boundary in stochastic systems because of the noise. The T (temperature) coefficient in the Hamiltonian (2.41) determines the amount of diffusion in the Hamiltonian. It acts like a noise because it smears out the wavefunction. In our numerical calculation of the eigenvalues, we experience difficulties in finding the eigenvalues when T is smaller than ($dx^2 + dy^2 + dz^2$), where dx, dy, and dz are the distances between two neighboring grid points in each dimension. When the T term is small, the eigenvectors for all eigenvalues tend to either exhibit high k (wavenumber) behavior or flow to the side of the grid. The high k behavior means that it has high-frequency oscillations in the spatial domain; in particular, the wavefunction change significantly between two adjacent grid points. In principle, all wavefunctions should be smooth so that its first and second derivatives are well defined. However, wavefunctions are not smooth when T is small, and thus we determine that these results are not meaningful. Furthermore, large k nodes correspond to large eigenvalues. Large eigenvalues should damp out quickly; hence, the high k behavior is just a transient response of the system but not its dynamics in the long term. We believe that this kind of behavior is caused by the effect of the coarse grid and does not resemble the physical dynamics of the system. T in the Laplacian term in (2.41) smoothens out the wavefunction, alleviating the effect of the grid. Consequently, we are not able to make T smaller than a certain threshold. As a result of our inability to set the temperature term to be near zero, we are unable to pinpoint the boundary of chaos and Markovian.

2.6. Conclusion

Our data suggest that our hypothesis is correct: the eigenvalues do shift towards the imaginary axis when the system becomes chaotic. In the limit of infinite grid size, the eigenvalues may reside on the imaginary axis, creating oscillations in time. Although the eigenvalues do move toward the imaginary axis, this method of determining chaos has some deficiencies: it takes a long time to diagonalize the Hamiltonian since its size grows with N^6, where N is the number of grid points in each direction of the phase space. Moreover, in chaos, the Hamiltonian becomes very difficult to diagonalize (similar to the case of diagonalizing the Hamiltonian for turbulence), resulting in a very unstable eigenvalue spectrum. The difficulty of simulating chaos and turbulence is due to the intrinsic nonlinearity of the systems [26, 27]. Because most algorithms solve matrixes using linearization techniques, they are inefficient when the system is highly nonlinear. Further efforts are needed to improve the numerical stability for calculating the eigenvalues.

One problem with our current way of constructing the Hamiltonian is that we approximate the derivatives of the wavefunctions by simply subtracting the two adjacent wavefunction values. This approximation is not accurate when the grid is coarse. Therefore, in order to cope with this problem, we can transform our phase space using the CW complex method [28, 29]. In this way, the Laplacian term has a well-defined representation based on the unit cell of the grid [30-32]; thus, we reduce the error in approximating the derivative terms. Furthermore, because we can only simulate a limited phase space, the flow terms are discontinuous on the boundaries of the grid. This can cause erroneous values for the derivatives of the flow terms at the boundaries, which may have a significant effect on the ground states of the system. Consequently, if we can replace the Chua oscillator with a dynamical system whose flow terms are zero at the boundaries, we can mitigate the problem of the discontinuity and can produce a more accurate eigenvalue spectrum.

3. Creation of a Self-organized Critical CMOS Circuit

3.1. Introduction

SOCr is observed in nature or in mathematical models; however, to date, we have not seen a physical complex system that is engineered deliberately to exhibit SOCr. Because SOCr has long-range correlations, it is possible for it to traverse through the entire phase space and will not be trapped into any specific local minimum. This unique characteristic is a very suitable for serving as a solution generator for simulated annealing algorithms.

In order for SOCr to occur, we must have nonlinear couplings between the elements. We also need to have a large number of elements in a system to show power law statistics. Moreover, we need to provide an energy drive to the system for sustained dynamics. CMOS is an ideal platform for implementing this complex dynamical system because of the existing infrastructure for fabricating VLSI circuits. With CMOS, we are able to make a large number of oscillators and coupled them together electrically. In addition, it is possible to read out the states electrically and pass them to the central processing unit (CPU) as trial solutions for the simulated annealing algorithms. Hence, this part of the thesis addresses the following **research question: how to build a CMOS circuit that can be driven to the SOCr phase?**

3.2. Theory

In order to confirm that our system is in SOCr, we need to record its dynamics. SOCr dynamics jumps between different attractors irregularly, producing intermittent (stops and goes) behaviors. Also, a SOCr system has long-range correlations. Therefore, its avalanche statistics must show a power law. There are two types of power law statistics. The first one is the size statistics, which plots probability verses avalanche size. The second one is the time statistics, which plots probability verses inter-avalanche time. The inter-avalanche time is defined as the amount of time between the start of two avalanches. Both statistics should show power laws. Our goal is to engineer a CMOS circuit that shows power law statistics and intermittency.

The Chua's circuit is a promising candidate for a building block of our SOCr circuit because it has two attractors, which can be used to represent digital one and zero. The original Chua circuit is depicted in Figure 4. In Figure 4a, there is a LC oscillator on the left. On the right, the circuit is connected to a nonlinear negative resistance, g(V_a), whose I-V characteristic is illustrated in Figure 4b. There are three state variables in the Chua's circuit: V_a, V_b, and I_c. Based on Kirchhoff laws, the dynamics of a Chua oscillator is governed by these three equations:

$$C_1 \frac{dV_a}{dt} = \frac{1}{R}(V_b - V_a) - g(V_a) \tag{3.1}$$

$$C_2 \frac{dV_b}{dt} = \frac{1}{R}(V_a - V_b) + I_c \tag{3.2}$$

$$L \frac{dI_c}{dt} = -V_b \tag{3.3}$$

When the circuit is in the chaotic phase, it jumps between two attractors aperiodically. More specifically, with certain choice of C_1, C_2, L, R, and g(V_a), the circuit will exhibit double scroll chaotic behavior as shown in Figure 5. The state variables V_a, V_b, and I_c are converted to their dimensionless forms, x, y, and z, respectively, in Figure 5. With simple algebra, (3.1-3.3) can be translated into their dimensionless forms (2.47-2.49) as we have seen in Section 2 [24].

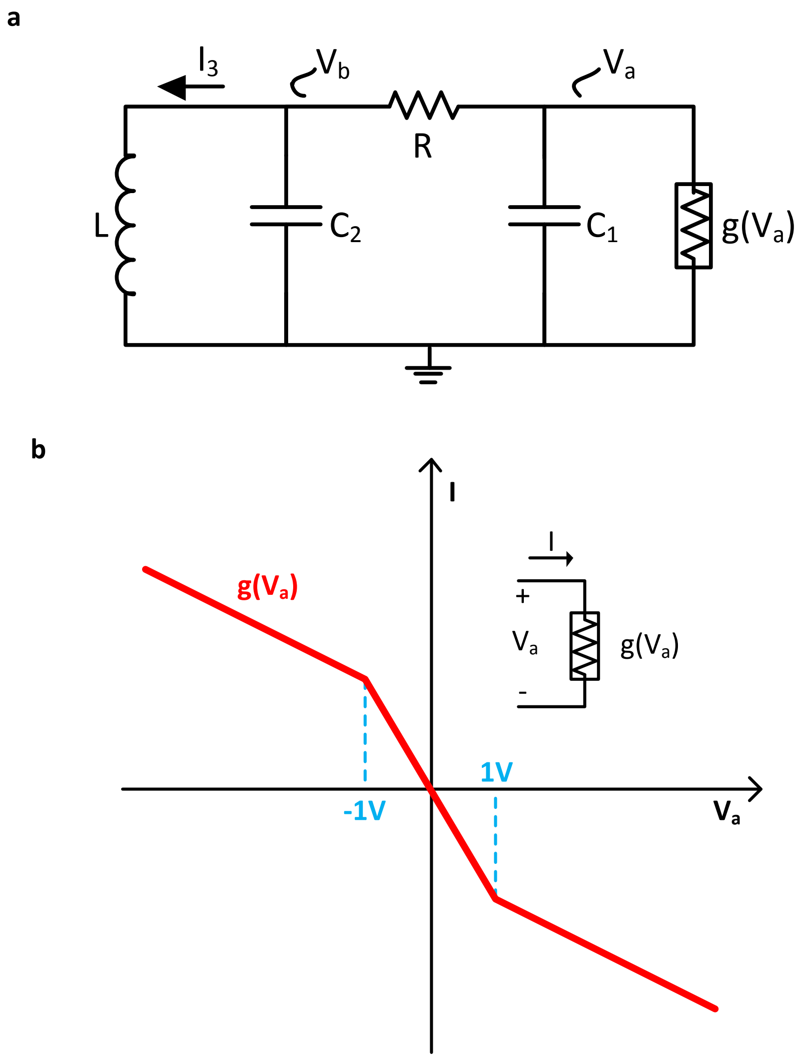

Figure 4 | The original Chua oscillator. **a**, The inductor, L, and the capacitor, C_2, form a simple LC oscillator. The three state variables are the voltage across C_1 (V_a), the voltage across C_2 (V_b), and the current flowing through the inductor, L, on the left (I_3). The circuit element $g(V_a)$ on the right is a nonlinear negative resistor that provides energy drive to the circuit for sustained dynamics. **b**, The I-V characteristic of the negative resistor, $g(V_a)$. The I-V curve has three segments; this nonlinearity is responsible for the two strange attractors shown in **Figure 5**.

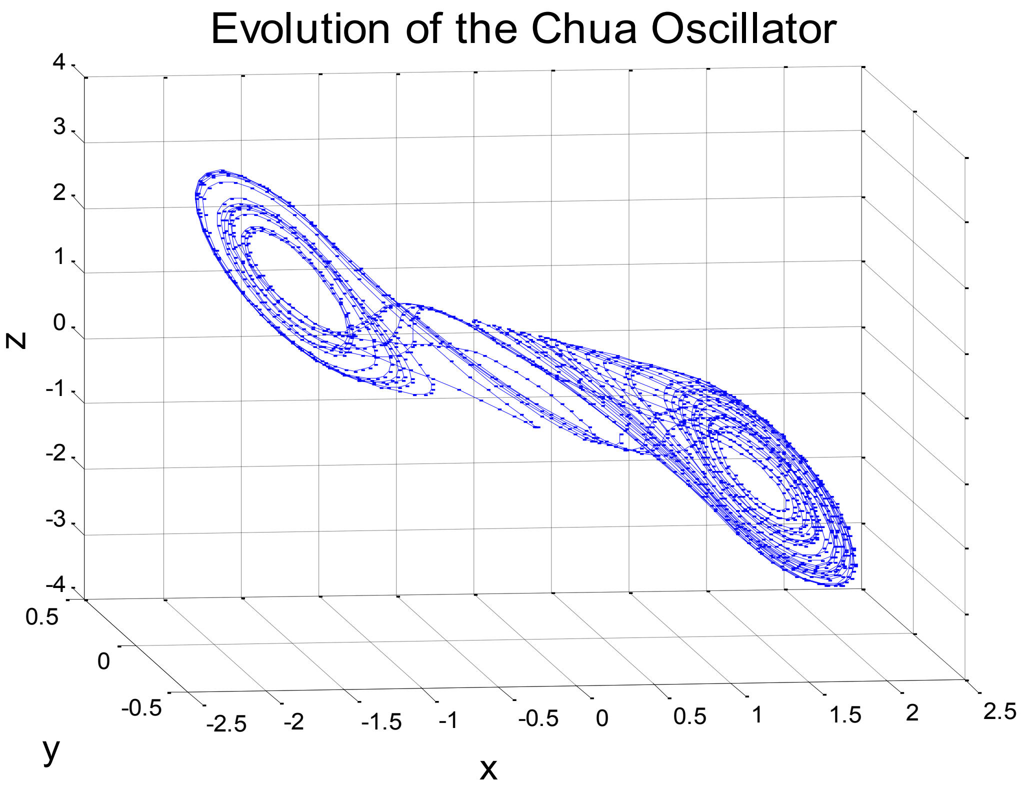

Figure 5 | Time evolution of the Chua oscillator. The three dimensionless state variables x, y, and z correspond to V_a, V_b, and I_c in the Chua's circuit. It has two attractors, and the system jumps between the two irregularly. The x=0 plane bisects the two attractors in the phase space.

The Chua circuit's has been implemented in many different ways. Considering the ease of fabricating the circuit onto a single integrated circuit, we selected a design (Figure 6) that only uses operational amplifiers (op amps), capacitors, and resistors, which can all be manufactured using standard lithography processes. A detailed design procedure of this implementation is reported by Kennedy [33]. In this design, the inductor in the original Chua's circuit is replaced by an active Gryo inductor, which is composed of op amps and resistors [34, 35]. The nonlinear negative resistor in Figure 4b is realized using two op amps; each op amp has different slopes and breakpoints. These slopes and breakpoints can be modified by changing the resistances connected to the op amps. As a result, different amounts of energy drive can be modified dynamically.

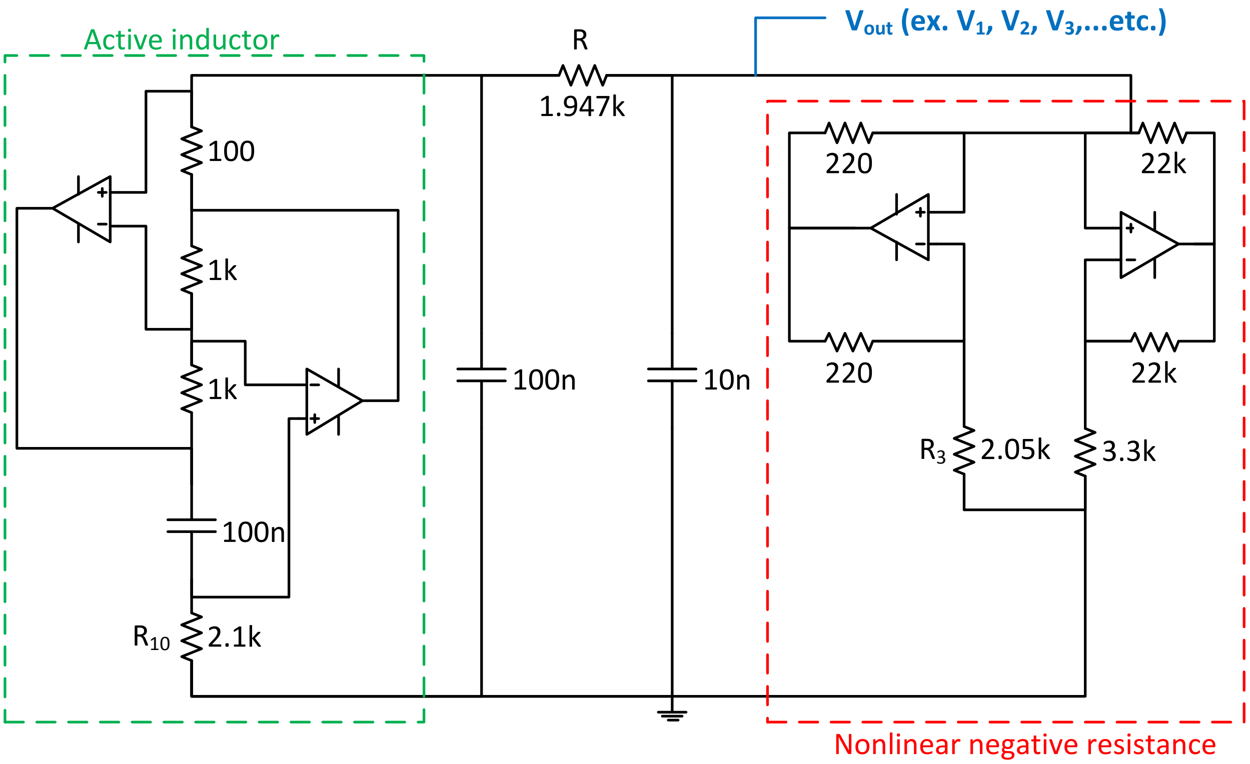

Figure 6 | The op amp implementation of a single Chua oscillator. This circuit topology was designed by Kennedy [33], in which the Chua's circuit is implemented using only op amps, resistors and capacitors. The two op amps on the left together with the capacitor form an active inductor [34, 35]. We avoid using an actual inductor so that we can manufacture the circuit into an integrated circuit using conventional fabrication processes in the future. The two op amps on the right form a nonlinear negative resistance which provides energy drive for the oscillator. The resistor R_3 controls the slope of this negative resistor and can be used as a bifurcation parameter that brings the circuit in and out of the chaotic regime. The resistor values for R, R_3, and R_{10} were determined experimentally. The output of the single Chua oscillator is V_{out}, which will be connected to its neighbors as illustrated in **Figure 7**.

When we put a matrix of Chua's circuits together with a nonlinear coupling element between neighboring oscillators, we expect the matrix of oscillators to show collective behaviors. Inspired by our previous work with ferromagnetic systems, we learned that a piece of magnetic film can show power laws during relaxation near the zero field. The matrix of Chua's circuits resembles a ferromagnetic system: each individual oscillator can be at one of the two states, and the couplings between the oscillators are nonlinear. Also, each Chua's circuit is capable of synchronizing with its neighbors. If the coupling strength is high, the entire circuit should be synchronized. On the other hand, when the coupling strength is low, each circuit is independent from one another. There exists a regime in which the coupling is not strong enough to have every cell in the circuit to be synchronized together, but strong enough to have part of the circuit synchronized locally. In this intermediate regime, SOCr may occur. **Our hypothesis is that we can build a SOCr CMOS circuit by coupling a matrix of Chua's oscillators with transmission gates.**

Consequently, we design our 16x16 SOCr circuit in the following way: We construct a matrix of 256 Chua oscillators in which each oscillator is coupled to its four neighbors through transmission gates in Figure 7a. The transmission gate is made of a pair of NMOS and PMOS transistors (Figure 7c). Their sources and drains are connected together. The body terminal of the NMOS is connected to the most negative voltage in the circuit, V_{ss} (-9V), while the one of the PMOS is connected to the most positive voltage, V_{dd} (9V). We can control the strength of coupling by changing the gate voltages W1 and W2. Figure 8 is a detailed schematic depicting the connections between neighboring oscillators.

In order to understand how adjacent oscillators couple, we measure the I-V characteristic of the transmission gates using the setup depicted in Figure 9a. The ground terminal is defined as the midpoint between V_{dd} and V_{ss}. We apply 0.2V to W1 and -0.2V to W2; the I-V characteristic of the transmission gate is plotted in Figure 9b. In Figure 9b, the x axis is the voltage difference between the two terminals of the transmission gate, V_x. The y axis is the current flowing through the transmission gate, I_x. The I-V characteristic shows a nonlinear relationship. Thus, the transmission gate can be viewed as a non-linear resistance that can be controlled by W1 and W2.

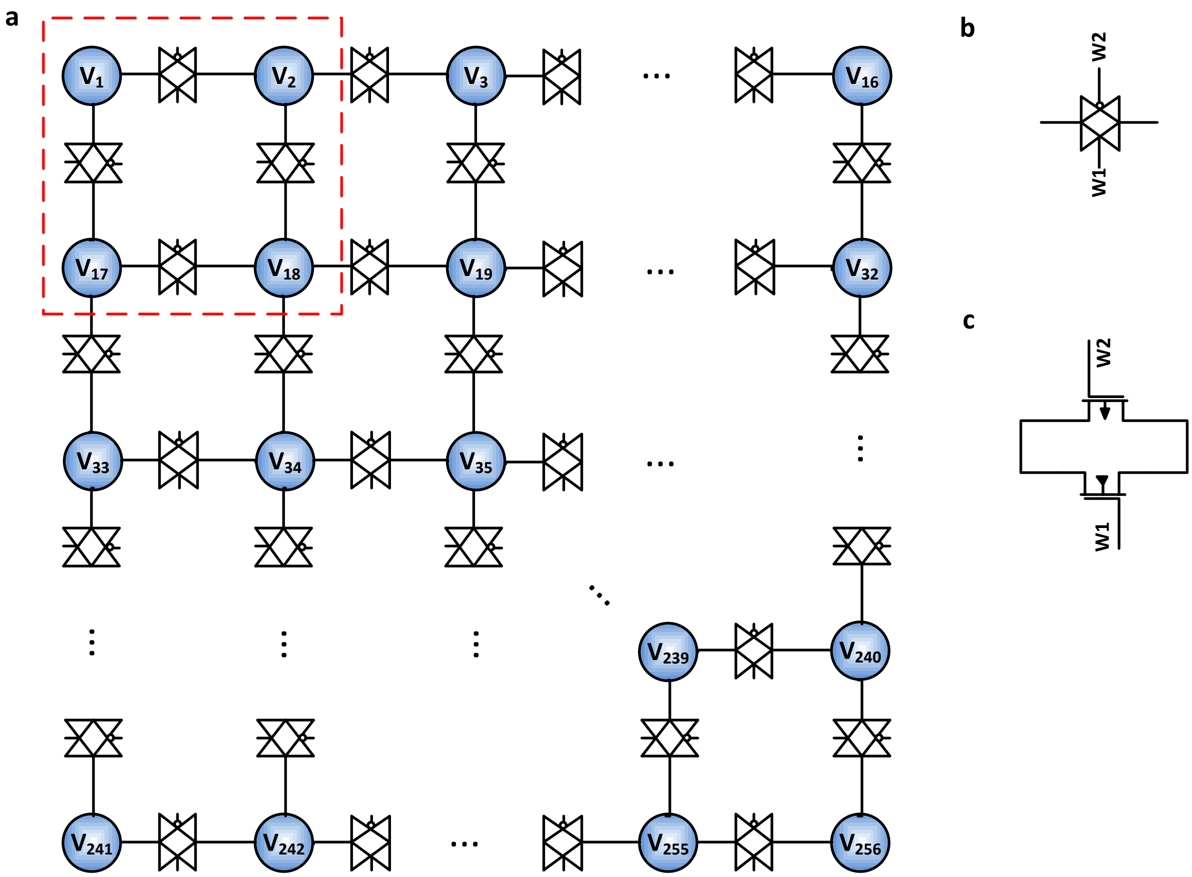

Figure 7 | The schematic of our 16x16 SOCr circuit. a, This circuit comprises 256 Chua's oscillators. Using transmission gates, we couple the V_{out} terminals of each oscillator with their neighbors. The outputs of this 16x16 circuit are the V_{out} terminals from each oscillator; therefore, it has 256 output signals. This circuit operates on an 18V DC power supply. **b**, The schematic symbol of the transmission gate. **c**, The implementation of the transmission gate using a pair of CMOS transistors. The sources and the drains of the NMOS and the PMOS are connected together to form a transmission gate. We can control the coupling strength between two neighboring oscillators by changing the gate voltages W1 and W2.

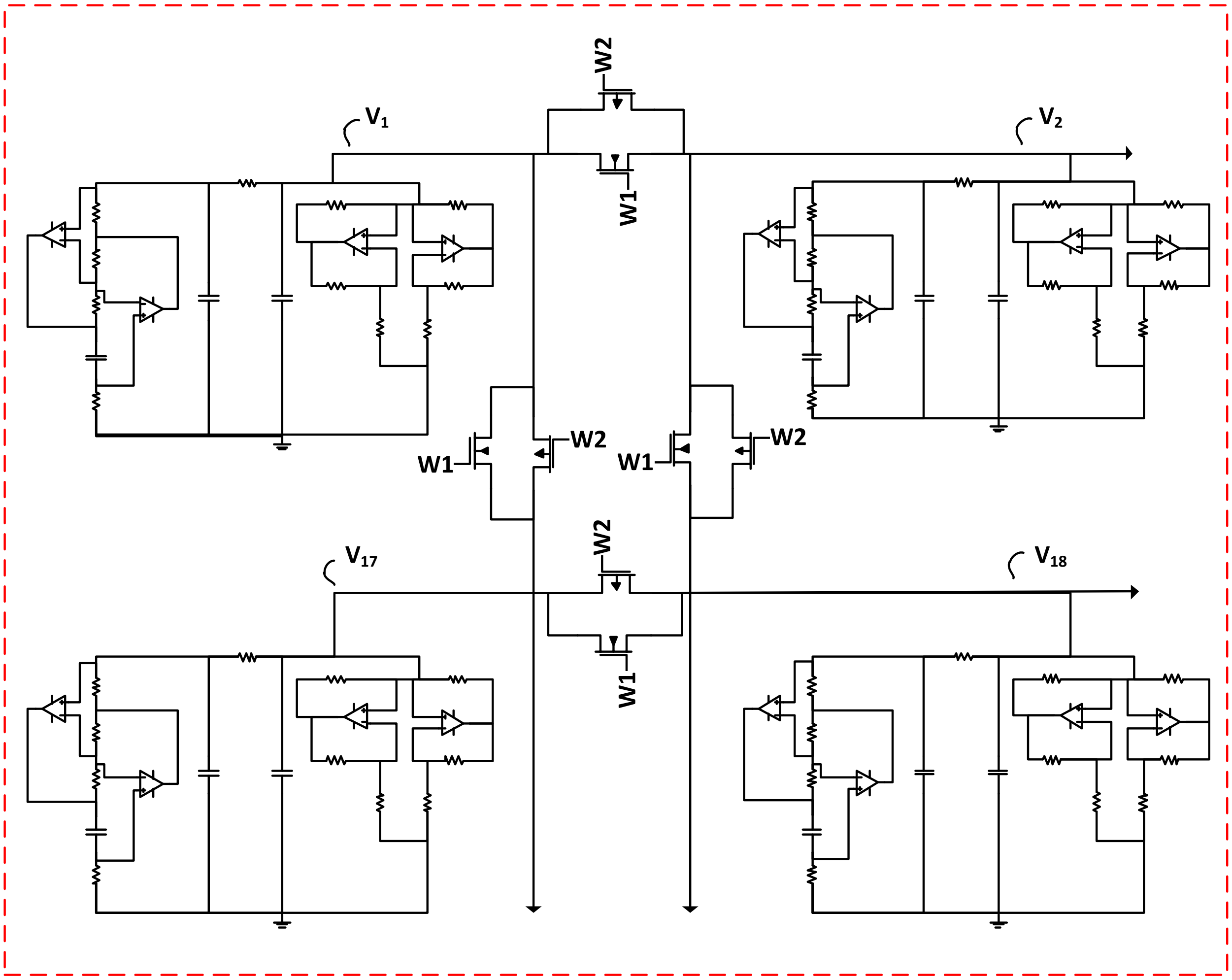

Figure 8 | Interconnections between adjacent oscillators. This is a detailed version of the red box in **Figure 7**, specifically showing the transmission gates that couple neighboring oscillators. W1 is a DC voltage that controls the amount of coupling between the adjacent oscillators for the NMOS. Similarly, W2 is for the PMOS. We set W1 to be 0.2V and W2 to be -0.2V.

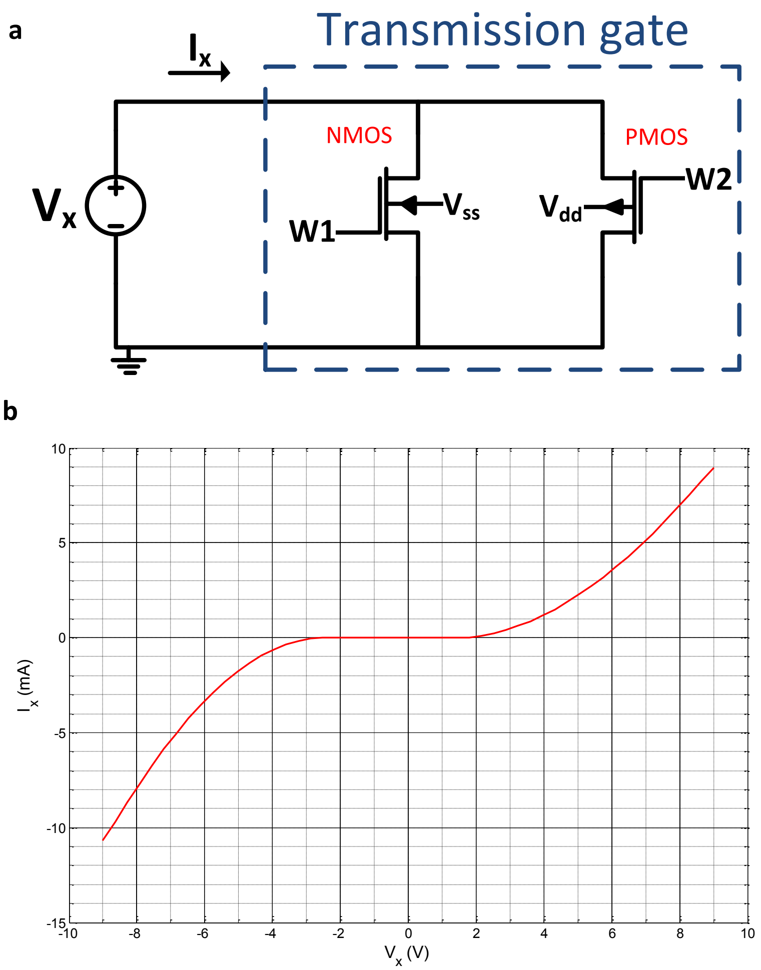

Figure 9 | Finding the I-V characteristic of the transmission gate. a, We setup a test bench for characterizing the transmission gate. The transmission gate can be understood as a nonlinear resistor that couples the two neighboring voltage terminals as shown in **Figure 8**. We characterize the transmission gate by applying a DC voltage (V_x) to the two terminals of the transmission gate and measure the current flowing through it. **b**, The I-V characteristic of the transmission gate. The I-V curve shows a nonlinear relationship. Hence, it can be viewed as a nonlinear resistor that couples the two terminals, which satisfies the requirement of the nonlinear couplings between elements in a SOCr system.

3.3. Methods

We use Cadence to simulate the dynamics of the coupled circuits. The op amps are being modeled by a voltage controlled voltage source (VCVS) with a gain of 1000, minimum voltage of -9V, and maximum voltage of 9V. We use a perfect voltage controlled voltage source instead of using an accurate op amp model in order to improve the simulation speed. Furthermore, because the frequency of our Chua oscillator is only around 3 kHz, the precise frequency response of the op amp is not required. As a result, we can model the op amps shown in Figure 6 using perfect voltage controlled voltage sources for the Chua oscillators.

On the other hand, we used an accurate model for the transmission gates because they play an important role in coupling the oscillators. We used the SPICE transistor models provided by the IC manufacturer: ALD1106 for PMOS and ALD1107 for NMOS. The models are available from the Web. First, we setup the 16x16 circuit (Figure 7) in Cadence with the resistor and capacitor values listed in Figure 6. We apply 0.2V to W1 and -0.2V to W2. Second, we simulate the 16x16 circuit for seven seconds and record the 256 output voltages with a strobe period of 10 µs. As a demonstration, the first three voltage outputs, V_1, V_2, and V_3 are plotted in Figure 10. Then, we use MATLAB to perform statistics on the voltage signals using the following procedures.

1. The intrinsic period of a single Chua oscillator is about 0.34 ms. Therefore, in order to filter out the intrinsic oscillations in the output signals, we use a moving average technique on each of the 256 voltage signals separately. As an example, we show the resulting signal in Figure 11. The green curve is signal V_3 before averaging, and the pink curve is after averaging. In order to draw the pink curve, we first define a window size of

0.34 ms. Then, we average the voltage values in the first data window (for example, from 130 ms to 130.34 ms). We record the mean as the first data point in the resulting signal (the pink curve). Then, we find the average during the second window (from 130.01 ms to 130.35 ms). Again, we record the mean as the second data point in the resulting signal. We continue to slide the data window further until we reach the end of the original V_3 signal (the green curve). This technique reduces the intrinsic oscillation in the signal.

Moreover, the purpose of the moving average is to detect when the oscillator is in synchronization with the neighboring oscillators, and when it is out of synchronization. When two neighboring oscillators are synchronized with each other, they both produce a large voltage swing between +9V and -9V as depicted during region A in Figure 10. When an oscillator becomes de-synchronized from others, it oscillates with a much smaller amplitude such as V_3 during region B in Figure 10, and it has a much higher average value. On the other hand, when it is in sync, its voltage signal has a much lower average value (ex. regions A and C in Figure 11) comparing to when it is out-of-sync (ex. region B in Figure 11). Consequently, we can use this moving average technique to identify when an oscillator becomes out-of-sync.

Furthermore, the oscillators become out-of-sync irregularly; when one oscillator becomes out-of-sync, it tends to trigger its neighbors to become out-of-sync as well. As a result, regions of the 16x16 oscillators will become out-of-sync at the same time. That is different “patterns” are being generated intermittently. Its intermittency indicates that the system is in SOCr.

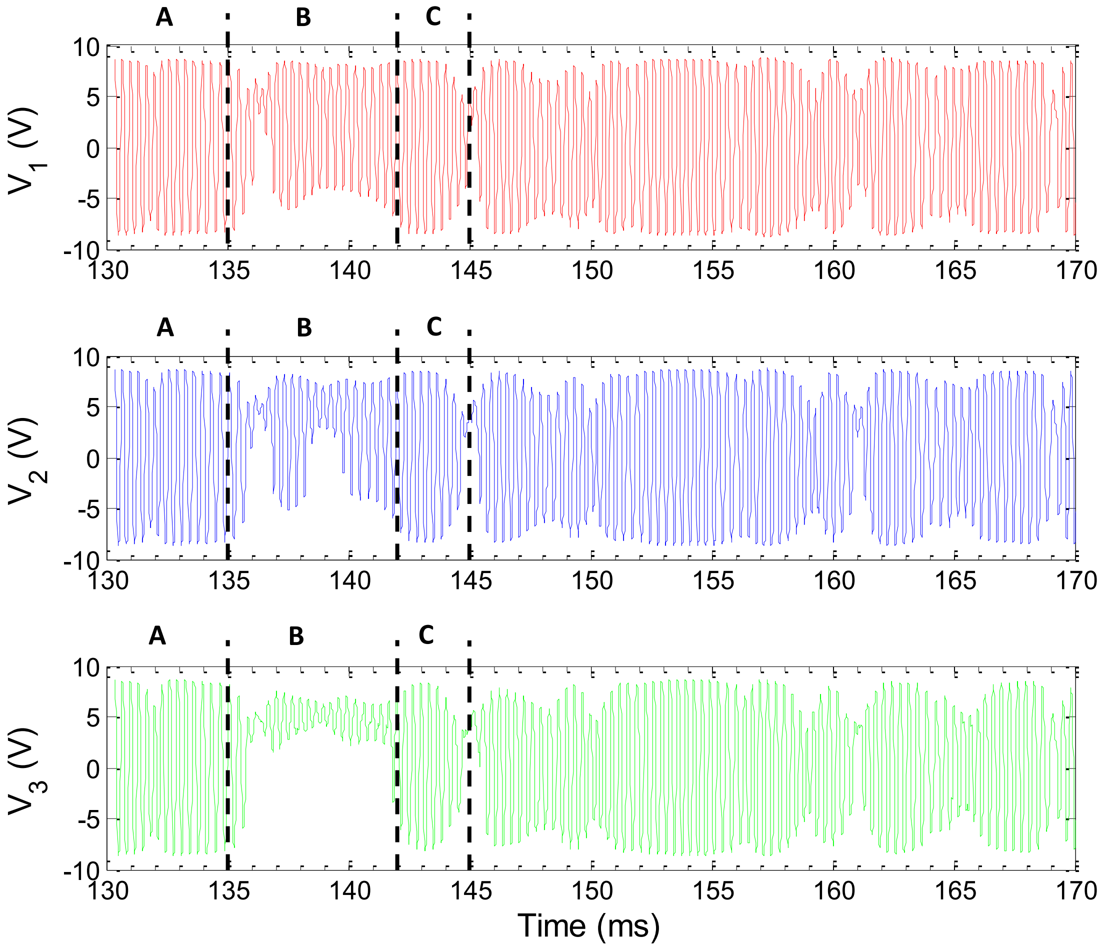

Figure 10 | Time domain output signal of the 16x16 circuit. As an example, we plot the voltage outputs (V_1, V_2, and V_3) from the first three oscillators in the 16x16 SOCr circuit illustrated in **Figure 7**. From 130 ms to 135 ms (region A), the three signals are in synchronization with each other; they all exhibit large oscillations between ±8V. During 135 ms to 142 ms (region B), V_3 goes out of synchronization from the other two voltages. From 142 ms to145 ms (region C), the three signals are in sync again. In order to detect when the signals are in sync and out-of sync, we perform a moving average on each of the 256 signals as shown in **Figure 11**.

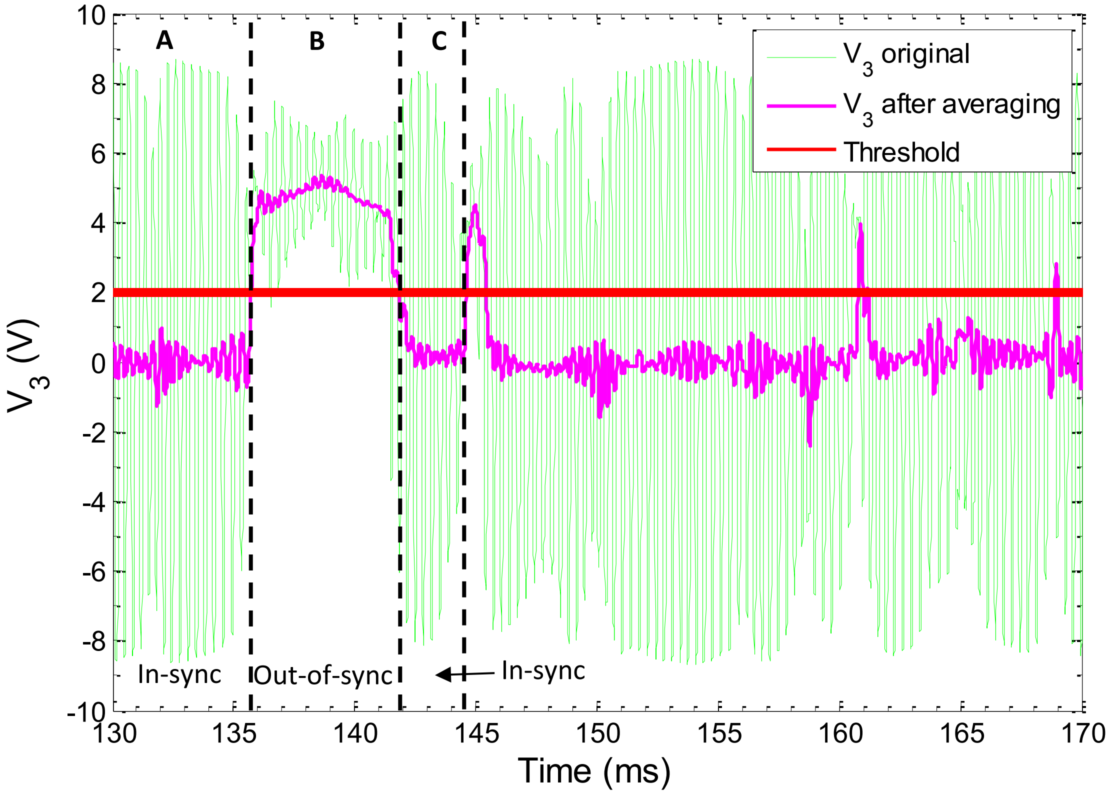

Figure 11 | Output signal V3 after moving averaging. We apply a moving average to the signal V3 from **Figure 10**. The sampling period of V3 is 10 µs. The moving average window size is 0.34 ms, which is roughly the intrinsic period of a single Chua oscillator. Our program finds the average value of V3 during the first 0.34 ms and replaces the first data point of V_3 with this value. Then, it slides the data window 10 µs further in time and replaces the second data point with the calculated average value. The program continues until the end of the signal; consequently, it smoothens out the intrinsic oscillation of the Chua's circuit. We define a threshold at 2V, which is the red line. If the resulting signal (the pink curve) is above the red line, the oscillator is out-of-sync. When it is below, the oscillator is in-sync. Therefore, we can precisely identify when V_3 is in-sync (region A and C) and when it is out-of-sync (region B).

2. We define a threshold to detect if a voltage node is in-sync or out-of-sync. Continuing from the previous step, we define a threshold voltage at 2V, which is the red line in Figure 11. We subtract this threshold from the averaged signal (the pink curve) calculated previously. Then, we digitize the signal such that any positive value becomes one and any negative value becomes zero. Consequently, we get a binary signal that is one when the node is out-of-sync and zero when it is in-sync. At any instance, the ones and zeros can be mapped onto a "picture frame" that represents the physical locations of the oscillators. In Figure 12, white pixels (ones) represent oscillators that are currently out-of-sync. Black pixels (zeros) denote oscillators that are currently in sync. There are 16x16=256 total pixels, representing the 16x16 Chua oscillators from Figure 7. As a result, by putting all the frames together, we construct a movie that delivers a visual representation of the 256 voltage recordings.
3. For a given time, we calculate the number of nodes that become out-of-sync by subtracting the current frame from the next frame; we do this for the entire duration of the movie. By subtracting the current frame from the frame at the next time instance, we obtain the number of nodes that become out-of-sync between the two frames. A negative difference means that more oscillators become in-sync. Hence, we obtain a single time domain signal that records the number of nodes that become out-of-sync as a function of time in Figure 13.

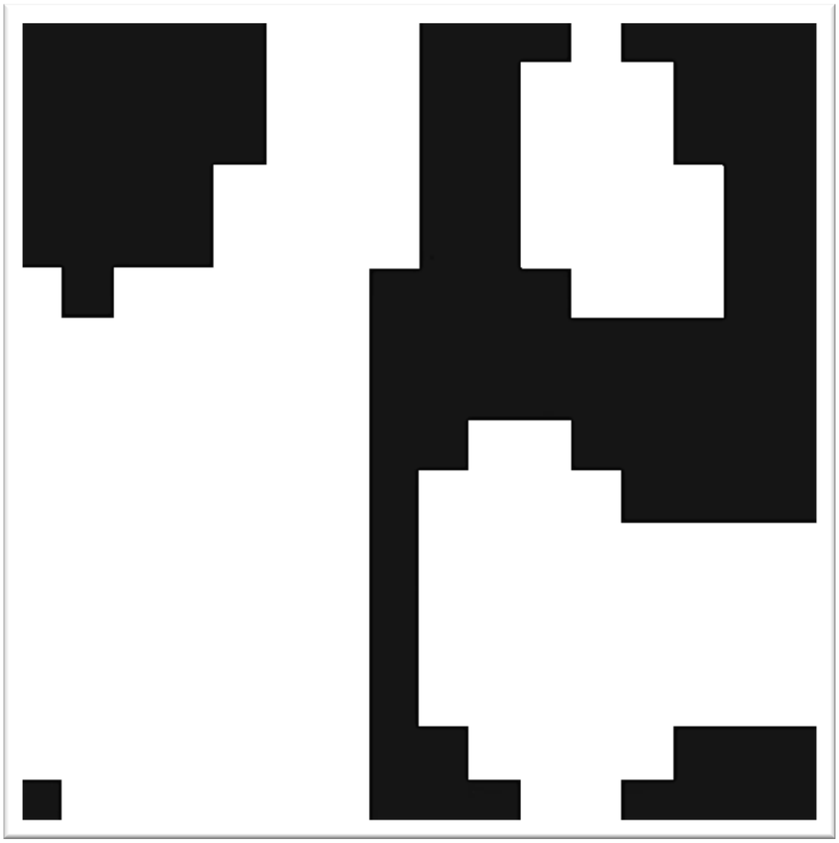

Figure 12 | A graphical representation of the 256 voltage signals. At any given moment, each oscillator can be in one of the two states: in-sync or out-of-sync. We represent the in-sync ones with black tiles; out-of-sync ones, white tiles. We can draw a 16x16 binary map according to where the signal is taken from. That is the top row represents voltage V_1 to V_{16} in **Figure 7**, and so on. This map (frame) is only for a specific moment in time. We can then construct a movie by cascading different frames.

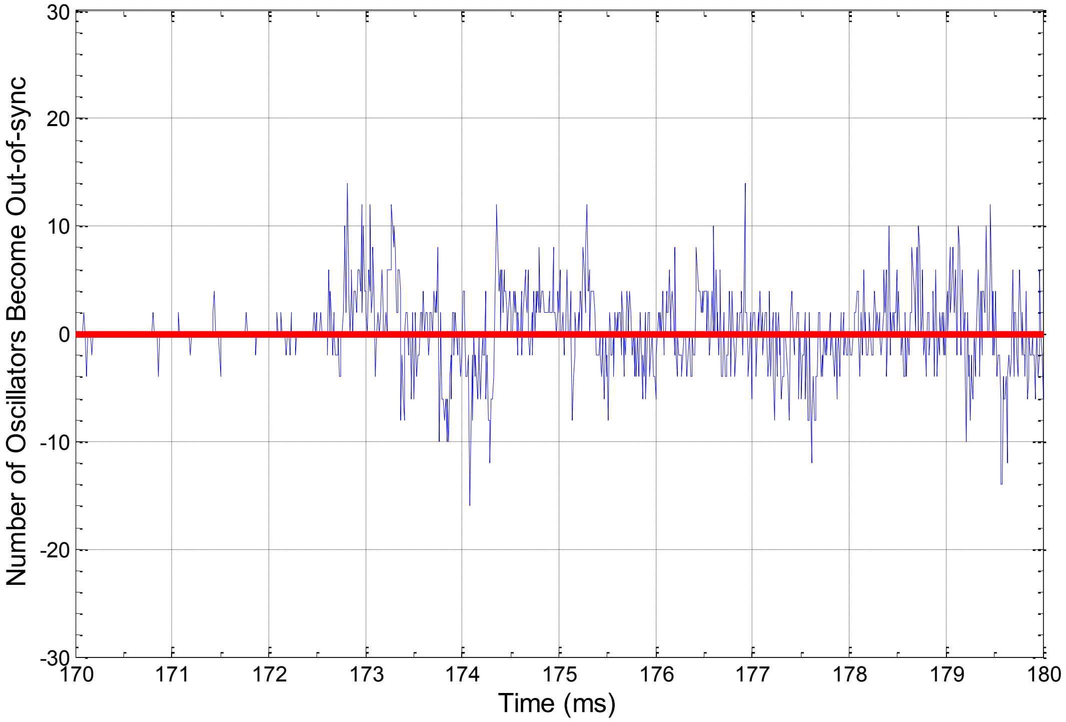

Figure 13 | Number of oscillators become out-of-sync versus time. After obtaining a movie in **Figure 12**, we find the number of oscillators that become out-of-sync at a specific moment by subtracting the two sequential movie frames. We perform this calculation for the entire movie. If the difference between the two adjacent frames is positive, more oscillators become out-of-sync at this moment. Likewise, if the difference is negative, more oscillators become in-sync. By calculating the positive area underneath the curve (below the blue line and above the red line), we obtain the size of each avalanche.

4. We define an “avalanche” as a group of nodes become out-of-sync together. Hence, the avalanche size is defined as the maximum number of nodes that are out-of-sync during an avalanche. When the signal in Figure 13 crosses zero from negative to positive, a new avalanche occurs. At the moment when it crosses zero from positive into negative, the

avalanche has reached its peak and start decreasing in size. For each avalanche, we calculate its size by integrating the positive area underneath the curve. We only count the positive area because it represents the duration between the start of an avalanche and its peak. Then, we collect all avalanche sizes and plot probability verses avalanche size in log-log scale in Figure 14.

5. Lastly, we compute the time statistics. We mark the starts of the avalanches by finding the instances when the signal in Figure 13 crosses zero from negative into positive. We find the differences between two consecutive start times and record them as inter-avalanche time. We plot probability verses inter-avalanche time in log-long scale in Figure 15.

3.4. Results

Statistics for the 16x16 circuit simulation:

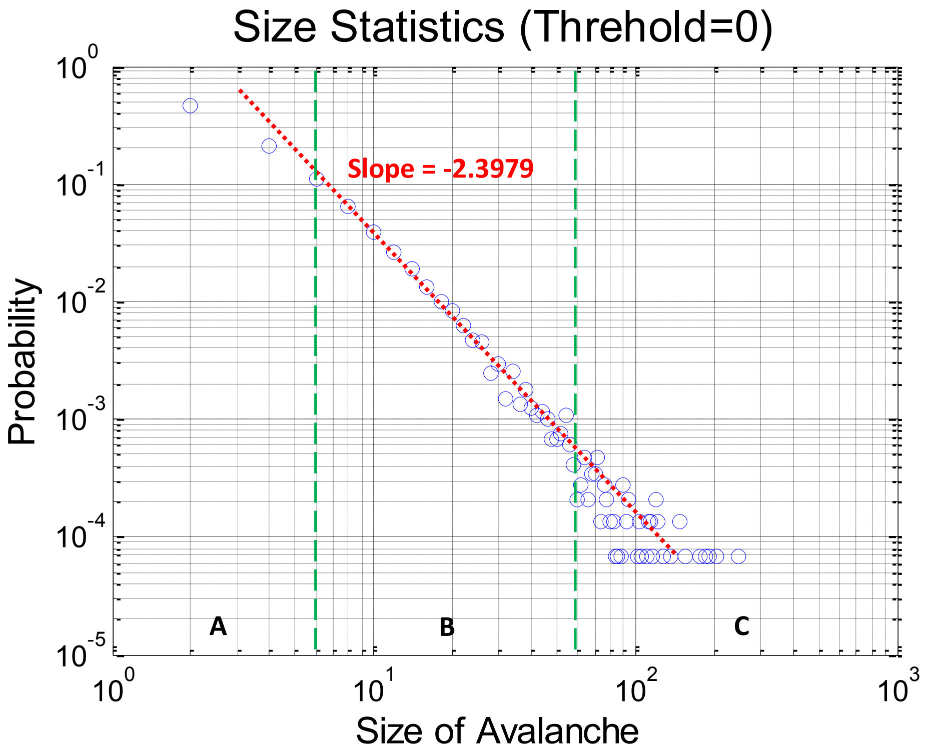

Figure 14 | Avalanche size statistics for the 16x16 circuit. Using the data processing method as described in Section 3.3, we record the size of each avalanche. Then, we plot the probabilities of different avalanche sizes in log-log scale. The data shows a power law relationship between probability and size, suggesting that the 16x16 circuit is in the SOCr phase. The curve spreads out near large avalanche sizes because we do not have sufficient number of avalanches near this regime. Hence, the data points become more discretized near large sizes in Region C. Furthermore, our statistics program does not distinguish two different avalanches occurring in the same frame. Two small avalanches might be recorded as one large avalanche. Consequently, there are less small avalanches being recorded; the data curve downward near small avalanche sizes in Region A.

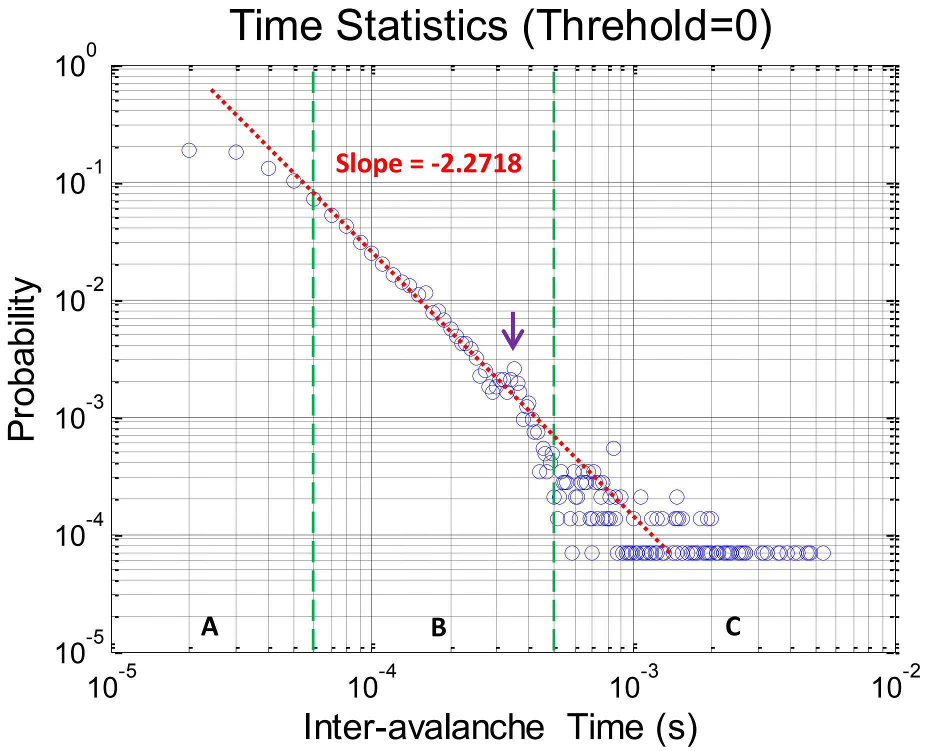

Figure 15 | Avalanche time statistics for the 16x16 circuit. We determine the start of a new avalanche by finding the moment that the curve in **Figure 13** crosses zero from the negative side to the positive side. We measure the time difference between two sequential avalanches and plot probability verses inter-avalanche time. The data has a power law relationship, suggesting long-range correlation in time.

3.5. Discussion

The probability verses avalanche size curve in Figure 14 is close to a straight line in Region B, showing a power law relationship between the two variables. The linear region (Region B) spans for two and a half decades in probability and around one decade in size with a critical exponent of -2.3979. Because the 16x16 circuit only contains 256 oscillators, the largest avalanche size is 256. As a result, the power law distribution is limited in size to about two decades. The power law in the avalanches size statistics is an evidence for long-range correlation. This means that a node from far away can still "sense" the flipping of the node at the current position. This long-range interaction is responsible for the collective SOCr phenomenon, in which patterns suddenly appear and go away.

When the avalanche size is more than 60 in Figure 14 (Region C), the curve is spread out. This might happen because of the finite recording time. It is more difficult for all 256 nodes to de-synchronize with each other; therefore, we must wait longer in order for this event to happen. However, we only simulated the circuit for seven seconds. Events like this only happens a handful of times during this seven second recording window. As a result, we see quantized levels in the probability distribution when avalanche size approach 256. The spreading of the curve is not a physical property; however, it is due to the measurement technique. Simulating our 16x16 circuit is very computationally intensive because the voltage at each node needs to be accurately calculated in order track the actual dynamics of the system; because all the oscillators are coupled through nonlinear elements, it is very difficult to solve a large matrix with nonlinear couplings. In fact, we simulated the transient response of the 16x16 circuit for seven seconds, and it took 18 days for the simulation to complete on our server.

Furthermore, our statistics program does not distinguish two different avalanches occurring in the same frame. Two small avalanches might be recorded as one large avalanche. Consequently, in Figure 14, there are less small avalanches being recorded than it should be; the data curve downward when avalanche size is less than six (Region A).

The inter-avalanche time statistics in Figure 15 also shows a power law relationship between probability and the wait time between two consecutive avalanches. The distribution is roughly linear in Region B with a critical exponent of -2.2718; it spans for about two decades in probability and one decade in inter-avalanche time. If the arrivals of avalanches are independent from one another, it is analogous to a Poisson process, in which the distribution of arrival time should follow an exponential relationship [36]. However, our data suggest that the distribution for the arrival time has a power law. This implies that the triggering of the avalanches is not independent; in particular, the power law suggests that the system has long-range correlation in the triggering process. On the contrary to power law, an exponential distribution implies exponential decay in correlation; thus, it is considered as short-range in comparison. Long-range correlation in time is often associated with memory of the system, because in order to have correlation in time, the system's future trajectory will depend on its past states. Ideally, a long-range correlation in time means that the system's dynamics will depend on all its history.

The curve in Figure 15 spreads out after $5*10^{-4}$ second (Region C) in a similar fashion as the previous graph. Again, we might have not waited long enough to obtain enough data to resolve the curve at this region. Hence, we see quantized levels in probability near the end of the curve. However, in addition to the spreading of the curve, there is also a bump at around $2.5*10^{-4}$ seconds (at the purple arrow). Because the internal period of the Chua oscillator is around $3*10^{-4}$

seconds, we believe that this bump is caused by this internal period of each individual circuit. Although we attempt to filter out this intrinsic frequency by a moving average technique, some effects may remain when the signals transition from out-of-sync to in-sync, or vice versa, causing the edge of the signals to be imperfect after the moving average.

In Figure 15 Region A, the data curve downwards towards smaller inter-avalanche time. This may happen because in the case of two avalanches simultaneously growing in the same frame, our statistical program registers them as a single avalanche. As an instance, let us assume that one avalanche starts at time equals 2 ms, and another avalanche starts in the opposite corner of the 16x16 oscillator matrix at time equals 2.04 ms. This is possible because the sampling period of the 256 voltage signals is 10 μs. If both avalanches continue to grow after they start, our program erroneously considers them as one avalanche. However, if the first avalanche stops growing before the second avalanche starts, our program will register them as two avalanches. In short, the decrease of probability near the region of small inter-avalanche time (Region A) is caused by the inability of our statistical program to distinguish the start of two different avalanches growing in the same frame.

3.6. Conclusion

We have successfully engineered a SOCr CMOS circuit using Chua oscillators and transmission gates. The 16x16 circuit shows a power law in both avalanche size statistics and inter-avalanche time statistics, which are the traits of a SOCr system. Although the circuit does not output digital signals directly, we can easily build low pass filters to find the moving average of each signal and then use a comparator to detect whether or not it is in-sync or out-of-sync. In other words, we can build a simple hardware to perform the data processing procedures described in Section 3.3. Thus, this circuit can be used to replace the solution generator for any simulated annealing algorithm.

4. Conclusion

SOCr is a mysterious phenomenon whose origin has not been fully understood. SOCr exist at the noisy boundary between chaos and the Markovian regime in the phase diagram (Figure 1). We obtain evidence in Section 2 that chaos may originate from symmetry breaking. Our data suggests that in chaos, eigenvalues move towards the imaginary axis as predicted in Section 2.2 and as shown in the pseudospectrum (Figure 2) in Section 2.4. Therefore, our hypothesis is true that the two-ghost Fokker-Planck Hamiltonian has purely imaginary eigenvalues in chaos.

We have designed a CMOS circuit in Section 3 that exhibits SOCr, using the necessary conditions for SOCr (nonlinear couplings, large number of elements, and energy drive) as guidelines for design specifications. The nonlinear couplings between neighbors are realized using transmission gates, which have a nonlinear I-V characteristic (Figure 9). We use the Chua oscillator (Figure 6) as a unit cell with the negative resistor as the energy drive. Using the signal processing techniques presented in Section 3.3, we obtain power laws in both avalanche size (Figure 14) and inter-avalanche time statistics (Figure 15), proving that our 16x16 circuit is indeed in SOCr. Consequently, our hypothesis is true that we can build a SOCr complex dynamical system using Chua oscillators and transmission gates.

5. References

[1] P. Bak, *How nature works : the science of self-organized criticality*. New York, NY, USA: Copernicus, 1996.

[2] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems (vol 10, pg 186, 2009)," *Nature Reviews Neuroscience,* vol. 10, Apr 2009.

[3] K. Nagel and M. Paczuski, "Emergent Traffic Jams," *Physical Review E,* vol. 51, pp. 2909-2918, Apr 1995.

[4] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization," *Chaos,* vol. 17, Jun 2007.

[5] B. A. Carreras, D. E. Newman, I. Dobson, and A. B. Poole, "Evidence for self-organized criticality in a time series of electric power system blackouts," *IEEE Transactions on Circuits and Systems I-Regular Papers,* vol. 51, pp. 1733-1740, Sep 2004.

[6] I. V. Ovchinnikov, "Topological field theory of dynamical systems," *Chaos,* vol. 22, Sep 2012.

[7] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry, and the natural sciences*, 2nd ed. Berlin ; New York: Springer-Verlag, 1994.

[8] I. V. Ovchinnikov, "Self-organized criticality as Witten-type topological field theory with spontaneously broken Becchi-Rouet-Stora-Tyutin symmetry (vol 83, 051129, 2011)," *Physical Review E,* vol. 84, Dec 27 2011.

[9] I. V. Ovchinnikov, "Topological field theory of dynamical systems. II," *Chaos,* vol. 23, Mar 2013.

[10] L.D. Faddeev and V. N. Popov, "Feynman diagrams for the Yang-Mills field," *Physics Letters B,* vol. 25, pp. 29-30, 24 July 1967 1967.

[11] M. Nakahara, *Geometry, topology, and physics*. Bristol, England ; New York: A. Hilger, 1990.

[12] L. Brunnet, H. Chate, and P. Manneville, "Long-Range Order with Local Chaos in Lattices of Diffusively Coupled Odes," *Physica D,* vol. 78, pp. 141-154, Nov 15 1994.

[13] K. Linkenkaer-Hansen, V. V. Nikouline, J. M. Palva, and R. J. Ilmoniemi, "Long-range temporal correlations and scaling behavior in human brain oscillations," *J Neurosci,* vol. 21, pp. 1370-7, Feb 15 2001.

[14] P. H. Jones, M. M. Stocklin, G. Hur, and T. S. Monteiro, "Atoms in double-delta-kicked periodic potentials: Chaos with long-range correlations," *Physical Review Letters,* vol. 93, Nov 26 2004.

[15] A. Auerbach, *Interacting electrons and quantum magnetism*. New York: Springer-Verlag, 1994.

[16] T. W. B. Kibble, "Symmetry Breaking in Non-Abelian Gauge Theories," *Physical Review,* vol. 155, pp. 1554-&, 1967.

[17] I. Low and A. V. Manohar, "Spontaneously broken spacetime symmetries and Goldstone's theorem," *Physical Review Letters,* vol. 88, Mar 11 2002.

[18] V. G. Ivancevic and T. T. Ivancevic, *Geometrical dynamics of complex systems : a unified modelling approach to physics, control, biomechanics, neurodynamics and psycho-socio-economical dynamics*. Dordrecht: Springer, 2006.

[19] J. Roca, "An Introduction to Topological Field-Theories," *Rivista Del Nuovo Cimento,* vol. 16, pp. 4-6, 1993.

[20] J. H. Schwarz, "Faddeev-Popov Ghosts and Brs Symmetry in String Theories," *Progress of Theoretical Physics Supplement,* pp. 70-80, 1986.

[21] L. N. Trefethen and M. Embree, *Spectra and pseudospectra : the behavior of nonnormal matrices and operators*. Princeton, N.J.: Princeton University Press, 2005.

[22] L. N. Trefethen, "Computation of Pseudospectra," *ACTA NUMERICA,* vol. 8, pp. 247-295, 1999.

[23] T. G. Wright and L. N. Trefethen, "Large-scale computation of pseudospectra using ARPACK and eigs," *Siam Journal on Scientific Computing,* vol. 23, pp. 591-605, Aug 15 2001.

[24] L. Fortuna, M. Frasca, and M. G. Xibilia, *Chua's circuit implementations : yesterday, today, and tomorrow*. Singapore ; Hackensack, N.J.: World Scientific Pub., 2009.

[25] V. S. Anishchenko, M. Safonova, and L. Chua, "Stochastic resonance in Chua’s circuit," *International Journal of Bifurcation and Chaos,* vol. 2, pp. 397-401, 1992.

[26] D. K. Lilly, "On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems," *Monthly Weather Review,* vol. 93, pp. 11-25, 1965.

[27] L.-S. Yao, "Non-linear Instabilities," in *Advanced Topics in Nonlinear Chaotic Systems*, ed: LAP LAMBERT Academic Publishing, 2012, pp. 127-140.

[28] J. H. C. Whitehead, "Combinatorial Homotopy .1.," *Bulletin of the American Mathematical Society,* vol. 55, pp. 213-245, 1949.

[29] J. H. C. Whitehead, "Combinatorial Homotopy .2.," *Bulletin of the American Mathematical Society,* vol. 55, pp. 453-496, 1949.

[30] P. B. Gilkey, T. P. Branson, and S. A. Fulling, "Heat-Equation Asymptotics of Nonminimal Operators on Differential Forms," *Journal of Mathematical Physics,* vol. 32, pp. 2089-2091, Aug 1991.

[31] S. Rosenberg, *The Laplacian on a Riemannian manifold : an introduction to analysis on manifolds*. Cambridge, U.K. ; New York, NY, USA: Cambridge University Press, 1997.

[32] C. Lange and A. I. Bobenko, "Combinatorial Curvatures, Group Actions, and Colourings: Aspects of Topological Combinatorics," 2004.

[33] M. P. Kennedy, "Robust OP AMP Realization Of Chua's Circuit," *Frequenz,* vol. 46, pp. 66-80, 1992.

[34] F. Yuan, *CMOS active inductors and transformers : principle, implementation, and applications*. New York: Springer, 2008.

[35] V. Siderskiy, "The Antoniou Inductance-Simulation Circuit Derivation," 2012.

[36] O. Castellanos, J. M. Lopez, J. M. Sentíes, and E. Anabitarte, "Intermittency, avalanche statistics, and long-term correlations in a turbulent plasma," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2013, p. P04022, 2013.