

# ABSTRACT

Title of dissertation:      EXPERIMENTS ON NETWORKS  
   OF COUPLED OPTO-ELECTRONIC  
   OSCILLATORS AND PHYSICAL  
   RANDOM NUMBER GENERATORS

Joseph David Hart  
Doctor of Philosophy, 2018

Dissertation directed by:   Professor Rajarshi Roy  
   Department of Physics

In this thesis, we report work in two areas: synchronization in networks of coupled oscillators and the evaluation of physical random number generators.

A “chimera state” is a dynamical pattern that occurs in a network of coupled identical oscillators when the symmetry of the oscillator population is spontaneously broken into coherent and incoherent parts. We report a study of chimera states and cluster synchronization in two different opto-electronic experiments. The first experiment is a traditional network of four opto-electronic oscillators coupled by optical fibers. We show that the stability of the observed chimera state can be determined using the same group-theoretical techniques recently developed for the study of cluster synchrony. We present three novel results: (i) chimera states can be experimentally observed in small networks, (ii) chimera states can be stable, and (iii) at least some types of chimera states (those with identically synchronized coherent regions) are closely related to cluster synchronization.

The second experiment uses a single opto-electronic feedback loop to investigate the dynamics of oscillators coupled in large complex networks with arbitrary topology. Recent work has demonstrated that an opto-electronic feedback loop can be used to realize ring networks of coupled oscillators. We significantly extend these capabilities and implement networks with arbitrary topologies by using field programmable gate arrays (FPGAs) to design appropriate digital filters and time delays. With this system, we study (i) chimeras in a five-node globally-coupled network, (ii) synchronization of clusters that are not predicted by network symmetries, and (iii) optimal networks for cluster synchronization.

The field of random number generation is currently undergoing a fundamental shift from relying solely on pseudo-random algorithms to employing physical entropy sources. The standard evaluation practices, which were designed for pseudo-random number generators, are ill-suited to quantify the entropy that underlies physical random number generation. We review the state of the art in the evaluation of physical random number generation and recommend a new paradigm: quantifying entropy generation and understanding the physical limits for harvesting entropy from sources of randomness. As an illustration of our recommendations, we evaluate three common optical entropy sources: single photon time-of-arrival detection, chaotic lasers, and amplified spontaneous emission.

EXPERIMENTS ON NETWORKS OF COUPLED  
OPTO-ELECTRONIC OSCILLATORS AND PHYSICAL  
RANDOM NUMBER GENERATORS

by

Joseph David Hart

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2018

Advisory Committee:  
Professor Rajarshi Roy, Chair/Advisor  
Professor Thomas E. Murphy, Co-Advisor  
Professor Daniel Lathrop  
Professor Edward Ott  
Dr. Louis M. Pecora

© Copyright by  
Joseph D. Hart  
2018





## Dedication

To my parents.

*Forsan et haec olim meminisse iuvabit*

AENEID, book I, line 203

## Acknowledgments

The work in this thesis was supported in part by the Office of Naval Research (ONR) Grant No. N000141410443 and Grant No. N000141612481.

# Table of Contents

Dedication	ii
Acknowledgements	iii
List of Figures	vii
1 Introduction	1
1.1 Network patterns of synchronization . . . . .	3
1.2 Physical random number generation . . . . .	5
1.3 Organization of the thesis . . . . .	7
2 Opto-electronic oscillators	10
2.1 Overview . . . . .	10
2.2 Background . . . . .	11
2.2.1 Delay systems . . . . .	11
2.2.2 Brief history of opto-electronic oscillators . . . . .	14
2.3 Experimental implementation . . . . .	16
2.4 Modeling opto-electronic oscillators . . . . .	18
2.5 Coupled opto-electronic oscillators and synchronization . . . . .	23
2.5.1 Experimental implementation . . . . .	25
2.5.2 Modeling networks of coupled opto-electronic oscillators . . . . .	26
3 Patterns of synchrony in a 4 node globally coupled network of optoelectronic oscillators	31
3.1 Overview . . . . .	31
3.2 Background . . . . .	33
3.2.1 Chimera states . . . . .	33
3.2.2 Patterns of synchronization . . . . .	34
3.3 Theory of cluster synchronization . . . . .	37
3.3.1 Existence of synchronous clusters . . . . .	37
3.3.2 Stability analysis for patterns of synchronization . . . . .	41
3.4 Partial synchronization in an small globally coupled opto-electronic network . . . . .	47

3.4.1	Allowed patterns of synchrony in a globally coupled network	47
3.4.2	Experiments	48
3.4.3	Stability analysis	50
3.5	Results	54
3.5.1	Stability calculation results	54
3.5.2	Chimera states and multistability	56
3.6	Discussion	60
4	Using a single nonlinear node with delayed feedback to realize networks of coupled oscillators	64
4.1	Overview	64
4.2	Background	68
4.2.1	Filter impulse response description of delay dynamical systems	68
4.2.2	Space-time representation of delay systems	68
4.3	A single delay system as a ring network	72
4.3.1	Applications to reservoir computing	84
4.3.2	Observation of chimera states	93
4.4	Arbitrary networks in a single delay system	100
4.4.1	Theoretical description	100
4.4.2	Alternative interpretation	104
4.4.3	Experimental implementation	105
4.5	Discussion	110
5	Experimental observation of patterns of synchronization using a single nonlinear node as a network	112
5.1	Overview	112
5.2	Dynamics of a single nonlinear map	114
5.3	Synchronization of two coupled maps	118
5.4	Chimera states in a five-node globally-coupled network	123
5.5	Synchronization of input-clusters	126
5.5.1	Motivation	127
5.5.2	Stability analysis for input clusters	129
5.5.3	Stability calculation results	134
5.5.4	Experimental results	136
5.6	Asymmetry-induced synchronization of symmetry clusters	137
5.6.1	Motivation	138
5.6.2	Theoretical considerations	139
5.6.3	Experimental results	147
5.6.4	Extension to intertwined clusters	150
5.7	Discussion	155
6	Evaluating physical random number generators	157
6.1	Evaluation of Physical Random Number Generators	162
6.1.1	State of the art	162
6.1.2	2016 NIST Draft Recommendations	169

6.1.3	Recommendation: A dynamical systems approach to entropy estimation . . . . .	171
6.1.3.1	Noise, chaos, and $h(\epsilon, \tau)$ . . . . .	172
6.1.3.2	Cohen-Procaccia entropy . . . . .	174
6.1.3.3	A comment about the relationship between PRNGs and deterministic chaos . . . . .	180
6.2	Review of some optical entropy sources . . . . .	183
6.2.1	RNG with Single Photon Detection . . . . .	183
6.2.2	RNG from Chaotic Lasers . . . . .	187
6.2.3	RNG from Amplified Spontaneous Emission . . . . .	194
6.3	Conclusions . . . . .	200
6.4	Epilogue . . . . .	202
7	What's next? . . . . .	207
A	Details of the experimental implementation . . . . .	213
B	Sample verilog code for two-node delay network . . . . .	223
C	Adjacency matrices from Chapter 5 . . . . .	230
D	Derivation of the $h_0$ limit . . . . .	233
	Bibliography . . . . .	234
	Curriculum Vitae . . . . .	254

## List of Figures

2.1	Nonlinear delayed feedback system . . . . .	13
2.2	Opto-electronic oscillator . . . . .	19
2.3	Opto-electronic oscillator dynamics . . . . .	20
2.4	Photograph of opto-electronic oscillator network . . . . .	27
3.1	Network Symmetries . . . . .	39
3.2	Globally coupled optoelectronic oscillator network . . . . .	49
3.3	Experimentally observed patterns of synchronization . . . . .	51
3.4	Stability of patterns of synchrony as a function of coupling strength and coupling delay . . . . .	55
3.5	Stability of chimera state in a 10-node network of optoelectronic os- cillators . . . . .	57
3.6	Multistability of patterns of synchronization in optoelectronic oscil- lator networks . . . . .	58
4.1	Illustration of the space-time representation . . . . .	70
4.2	Illustration of impulse responses . . . . .	76
4.3	Illustration of the coupling term in the space-time representation of delay systems . . . . .	77
4.4	Illustration of the adjacency matrices implemented by various filters. . . . .	79
4.5	Classical reservoir computing vs reservoir computing in a delay system . . . . .	86
4.6	Speech recognition using a reservoir computer implemented by a delay system . . . . .	92
4.7	Experimental record of chimera solutions generated in delay dynamics . . . . .	96
4.8	Features of the nonlinear coupling function for obtaining chimera pat- terns in delay dynamics. . . . .	98
4.9	Experimental apparatus for realizing arbitrary networks in a delay system . . . . .	107
4.10	Experimental apparatus for realizing arbitrary networks through mul- tiple time-delays . . . . .	108
4.11	Intensity modulator nonlinearity . . . . .	109
5.1	Illustration of single nonlinear map . . . . .	116

5.2	Bifurcation diagram of the sine-squared map . . . . .	117
5.3	Illustration of a time-delayed implementation of two coupled oscillators	119
5.4	Synchronization of two sine-squared maps . . . . .	122
5.5	Chimera in a five-node globally-coupled network . . . . .	124
5.6	Equitable vs. orbital partitions . . . . .	128
5.7	Transformation matrices . . . . .	133
5.8	Equitable partition measurements and stability calculations . . . . .	135
5.9	Improvement of synchronizability by breaking the cluster symmetry .	146
5.10	Experimental demonstration of structural AISync in global synchro- nization . . . . .	148
5.11	Experimental demonstration of structural AISync in cluster synchro- nization . . . . .	151
5.12	Structural AISync in intertwined clusters . . . . .	154
6.1	Methods of evaluating Physical Random Number Generators . . . . .	163
6.2	Survey of optical random number generation rates . . . . .	164
6.3	Information theoretical limits for physical RNG . . . . .	168
6.4	Entropy rate of the logistic map . . . . .	175
6.5	Entropy rate of single photon counting . . . . .	184
6.6	Harvesting entropy from cascaded chaotic lasers . . . . .	189
6.7	Entropy rate of cascaded chaotic lasers . . . . .	192
6.8	Harvesting entropy from filtered amplified spontaneous emission . . .	196
6.9	Entropy rate of filtered amplified spontaneous emission . . . . .	199
A.1	Data Conversion Card Modifications . . . . .	214
A.2	Non-inverting amplifier circuit diagram . . . . .	216
A.3	FPGA timing diagram . . . . .	222



## Chapter 1: Introduction

They believed that prediction was just a function of keeping track of things. If you knew enough, you could predict anything. That's been cherished scientific belief since Newton.

*And?*

Chaos theory throws it right out the window.

---

*Jurassic Park*

MICHAEL CRICHTON

The idea that long-term prediction could be impossible even for truly deterministic systems was perhaps first understood by Henri Poincaré when he noticed that “small differences in the initial conditions [can] produce very great ones in the final phenomena” [1]. This sensitive dependence on initial conditions turned out not to be unique to the three-body problem Poincaré was studying in the early 1900s. It appeared again and again, in van der Pol’s electronic circuits [2, 3], Lorenz’s simplified convection model [4], May’s population models [5], Haken’s unstable lasers [6], and so on. Eventually, this behavior came to be called chaos [7, 8].

It turns out that in chaotic systems, trajectories that start with nearby initial

conditions diverge exponentially quickly. This rate of exponential convergence or divergence of nearby initial conditions can be quantified by Lyapunov exponents. In order to display the extreme sensitivity to initial conditions necessary for chaos, a system must have at least one positive Lyapunov exponent. For detailed descriptions of Lyapunov exponents in particular and chaotic systems in general, see the texts by Strogatz [9] and Ott [10].

One thing that may be surprising about chaotic systems is that two similar chaotic systems, started from different initial conditions, can synchronize when they are coupled together. While the Lyapunov exponents of the individual chaotic systems typically remain the same, the systems will synchronize if all of the Lyapunov exponents associated with the difference dynamics are negative. Physically, this means that the coupled chaotic systems as a set are still extremely sensitive to initial conditions, but a perturbation away from synchrony will damp out exponentially quickly. The synchronization of chaotic systems was discovered by at least three independent groups, in Japan [11], the Soviet Union [12], and the U.S. [13]. For a thorough presentation of synchronization, see the book by Pikovsky, Rosenblum, and Kurths [14]. A discussion of the synchronization of two Lorenz systems can be found in the book by Strogatz [9].

One thing that may be less surprising about chaotic systems is that they are difficult to predict. This has led to the idea of using chaotic systems to generate random numbers, which are useful for gambling, Monte Carlo simulations, and particularly data encryption. While this idea is not new – pseudo-random number generators have been based on chaotic maps since at least 1947 [15] and chaotic elec-

tronic circuits were shown to generate random bits at a rate of 1 Mbit/s in 2001 [16] – in 2008 it was demonstrated that chaotic lasers could produce random bits at rates that were previously unobtainable (greater than 1 Gbit/s) [17]. Chaotic random number generation is perhaps the most successful and potentially useful application of experimental chaos developed so far.

## 1.1 Network patterns of synchronization

Given that two chaotic systems can synchronize, it may not be surprising that networks containing more than two chaotic systems can synchronize. It may be surprising, however, that in a network of many coupled chaotic systems, some sets of nodes can synchronize while others do not. This is called cluster synchronization or a chimera state, depending on the details.

A “chimera state” is a dynamical pattern that occurs in a network of coupled identical oscillators when the symmetry of the oscillator population spontaneously breaks up into coherent and incoherent regions. Cluster synchronization occurs when one or more subsets of nodes in a network synchronize.

In this thesis, we observe cluster synchronization and chimera states in two different opto-electronic network experiments. The first experiment is a traditional network of four opto-electronic oscillators coupled by optical fibers. In fact, this is the smallest possible network in which a chimera state can be observed. Further, we show that the stability of the observed chimera state can be determined using the same group-theoretical techniques recently developed for the study of cluster

synchrony. There are three novel results: (i) chimera states can be experimentally observed in small networks, (ii) chimera states can be stable, and (iii) at least some types of chimera states (those in which the coherent region is identically synchronized) are closely related to cluster synchronization.

This method of creating experimental networks – namely, physically coupling nominally identical oscillators – has some limitations. First, the oscillators can never be identical. Second, changing the network topology is difficult because the physical connections typically have to be changed. Most important, however, is that the traditional approach of interconnecting a large number of independent oscillators is not scalable to larger networks. Not only is building many individual oscillators costly and time-consuming, but the number of physical connections between  $N$  oscillators scales as  $N(N - 1)$  so reconfiguring the network becomes challenging.

In order to overcome these limitations, we developed a new opto-electronic experiment that utilizes the space-time representation of time-delayed systems [18] to realize networks of truly identical oscillators. Our experiment consists of a single opto-electronic oscillator with a long time delay and digital filtering. The nodes of the network are distributed along the delay line, and the coupling topology is implemented by means of a digital filter. Because all oscillators that comprise the system sequentially traverse the same physical device, all the nodes are *truly identical*. Because the coupling is implemented by a digital filter, the network can have arbitrary topology and can be easily reconfigured. The price we pay is that the network update rate scales as  $N^{-1}$  because the network nodes update serially.

Similar set-ups that use time-invariant analog filtering have been used to ob-

serve chimera states [19, 20] and to perform reservoir computing [21, 22]. These experiments were limited to circularly symmetric (ring) networks because of the symmetry imposed by the time-invariant filter. Our use of a digital time-dependent filter gives us the flexibility of implementing arbitrary networks.

We use this flexibility to study chimera and cluster states in a variety of networks. First, we experimentally verify our 2016 prediction [23] that chimeras can also be observed in globally-coupled networks of more than four nodes. Then we show how to apply techniques from group theory to determine the stability of non-symmetric synchronous clusters in non-Laplacian networks and confirm these stability calculations with the first experimental observation of such cluster states. Finally, we present theoretical arguments for and experimental confirmation of a surprising phenomenon: In order to optimize the synchronizability of a given symmetry cluster of a Laplacian network, one must break the cluster’s internal symmetry.

## 1.2 Physical random number generation

Physical random number generation has been the basis of gambling since pre-historic times. Even today, physical random number generation forms the basis for just about all of the classic casino games: Roulette, craps, and card games all rely on unpredictable (although not chaotic) physical processes. Newer electronic gaming machines rely on pseudo-random number generators because they are cheap, fast, and reliable. The only problem is that they are deterministic, and, in theory, predictable if an attacker knows how they work and their internal state (or “seed”).

This theory was “experimentally verified” by Russian players in 2017, who obtained slot machines from defunct casinos, opened them up and learned how they worked, then used this knowledge to win what is thought to be millions of dollars [24]. An official statement of the slot machine’s manufacturer said only that their machines “are built to and approved against rigid regulatory technical standards” [24].

Similar pseudo-random number generators underlie many of the encryption techniques we use for secure communications, including financial transactions. In order to improve security, digital information systems have begun to utilize physical sources to generate high-speed unpredictable signals. This has also necessitated an evolution of the methods for evaluating random number generators. The U.S. National Institute for Standards and Technology (NIST) has recognized this and just this year released a new set of standards for evaluating physical random number generators [25].

In this thesis, we review the state of the art in the evaluation of physical random number generation and recommend a new paradigm: quantifying the rate of entropy generation and understanding the limits of harvesting entropy from physical sources of randomness. We advocate for the separation of the physical entropy source from deterministic post-processing in the evaluation of random number generators and for the explicit consideration of the impact of the measurement and digitization process on the rate of entropy production. We present the Cohen-Procaccia estimate of the entropy rate  $h(\epsilon, \tau)$  as one way to do this. As an illustration of our recommendations, we apply the Cohen-Procaccia estimate as well as the entropy estimates from the 2016 NIST draft standards [26] for physical random number

generators to evaluate and compare three common optical entropy sources: single photon time-of-arrival detection, chaotic lasers, and amplified spontaneous emission.

### 1.3 Organization of the thesis

Chapter 2 describes the traditional network of four opto-electronic oscillators. Some history of the development and use of opto-electronic oscillators is provided. Our experimental implementation is described, and both continuous- and discrete-time models are presented.

Chapter 3 describes the experimental observation of chimera states and cluster synchronization in a four-node globally-coupled network of opto-electronic oscillators. We use group theory-based arguments to explain what patterns of synchronization might be expected, and we show experimental time series that demonstrate the observation of all possible patterns. We perform stability calculations for all chimera and cluster states; the results are found to be in good agreement with our experimental measurements. This chapter demonstrates that chimeras can be observed in small networks, that they can be stable, and that they can be understood using the theory of cluster synchronization.

Chapter 4 describes our second opto-electronic experiment, which utilizes the space-time representation of time-delayed systems to realize networks of truly identical oscillators. We review previous works that use the space-time representation to realize ring networks in a single delay system with a time-invariant analog filter for applications to reservoir computing and for the observation of chimera states.

We show that networks with arbitrary topology can be realized in delay systems by replacing the time-invariant filter with a time-dependent digital filter, and we describe in detail our opto-electronic implementation of such a network.

Chapter 5 presents a series of experimental results using our opto-electronic delay network. We first study the dynamics of a single node and two bidirectionally-coupled nodes. We then report a chimera in a five-node globally-coupled network, confirming our predictions from Chapter 3 that chimeras should be present in larger globally-coupled networks as well. We show how to extend the stability analysis for symmetry clusters to synchronized clusters that are not symmetry clusters, and we use our experimental network to verify these stability calculations. We then report the first experimental confirmation of Asymmetry-Induced Synchronization (AISync) of symmetry clusters, a counter-intuitive phenomenon in which the synchronizability of a symmetry cluster can, in general, be improved by breaking the cluster's internal symmetry. Our experimental results, each interesting on their own, combine to show the versatility of our new approach to realizing networks in the lab.

Chapter 6 describes our work on the analysis of entropy sources for physical random number generation. We explain the difference between pseudo-random number generators and physical random number generators, and we review the state of the art in the evaluation of physical random number generators. We provide some recommendations for how to improve the most-commonly used evaluation procedures, including emphasizing the physical origins of the randomness and the role that the measurement process plays in the extraction of entropy from a physical



source. We present the Cohen-Procaccia estimate of the entropy rate  $h(\epsilon, \tau)$  as one way to do this. In order to provide an illustration of our recommendations, we apply the Cohen-Procaccia estimate as well as the entropy estimates from the 2016 NIST draft standards for physical random number generators to evaluate and compare three common optical entropy sources: single photon time-of-arrival detection, chaotic lasers, and amplified spontaneous emission.

Chapter 7 considers the impact of our work and poses some questions that came up during our research that might be interesting to pursue in the future.

## Chapter 2: Opto-electronic oscillators

### 2.1 Overview

Chapters 3-5 of this thesis concern experiments on opto-electronic oscillators. Our opto-electronic oscillators are fiber-based nonlinear feedback loops made of standard telecommunications components; the combination of the nonlinearity and time-delayed feedback allows the system to exhibit a wide variety of dynamical behaviors. This chapter presents relevant historical, experimental, and modeling background on opto-electronic oscillators. Much of the information in this chapter was developed and presented in greater detail in prior Ph.D. theses, in particular those of Yanne Chembo Kouomou [27] at Universitat de les Illes Balears, and Adam Cohen [28], Bhargava Ravoori [29], and Caitlin Williams [30] at the University of Maryland, College Park.

Section 2.2 presents some historical context surrounding time delayed systems in general and opto-electronic oscillators in particular. Our experimental implementation of a single opto-electronic oscillator is described in Section 2.3. In Section 2.4 both continuous- and discrete-time models are developed and the variety of dynamics that can be displayed by an opto-electronic oscillator are discussed. The idea of coupling two (or more) oscillators with the intent of studying their synchronization

properties is introduced in Section 2.5. The experimental details of our coupling scheme are described in Section 2.5.1. A brief general theoretical description of networks of coupled oscillators followed by both continuous- and discrete-time models of our network of four opto-electronic oscillators are presented in Section 2.5.2.

## 2.2 Background

### 2.2.1 Delay systems

Time delays often arise when the intrinsic dynamics of a system are fast enough that the finite propagation velocity of signals must be taken into account. For example, in a semiconductor laser with time delayed feedback through an external mirror, the photon lifetime is significantly shorter than the feedback time, which can cause the laser intensity to oscillate chaotically [31]. From an experimental point of view, delay systems are particularly attractive for two reasons: 1) a small number of variables (often only one) needs to be measured to obtain the full state of the system and 2) the dimensionality of the dynamics often increases linearly with the delay [32, 33], which is typically easy to control. This is in contrast to, for example, spatially extended systems, another type of high-dimensional system in which the variable of interest must be measured in many different locations simultaneously.

The basic form of a delayed feedback system is depicted by the block diagram in Fig. 2.1a. The output of a nonlinearity  $F(\cdot)$  is amplified, filtered, and delayed before being fed back as the input to the nonlinearity. Even if filtering is not intentionally implemented, the bandwidth limitations of the system itself will cause

some filtering. In many delay systems, the filtering can be modeled as a simple low-pass filter:

$$\tau_L \dot{x}(t) + x(t) = F(x(t - \tau_D)), \quad (2.1)$$

where  $x$  is the dynamical variable of interest,  $F(x)$  is a nonlinear function of  $x$ ,  $\tau_L$  is the intrinsic time scale of the system, and  $\tau_D$  is the time delay. Equation 2.1 has been used to model systems from many different areas of science [34], including physiology [35], population dynamics [36], and laser physics [37]. Systems described by Eq. 2.1 have been shown to display a wide variety of interesting behaviors, including square waves [38, 39], new types of chaos (in the case that  $\tau_D$  varies in time) [40], and spatiotemporal phenomena [41].

Indeed, research over the last 25 years has shown that a wide variety of spatio-temporal phenomena can be observed in temporal systems with a long delayed feedback. The interpretation of dynamics in delayed systems as spatio-temporal phenomena is enabled by the space-time representation, in which a spatial coordinate is mapped into a temporal time-slot [18]. Some of the theoretically predicted and experimentally observed spatio-temporal phenomena include defect-mediated turbulence [42, 43], coarsening [44, 45], domain nucleation [46], spatial coherence resonance [47], and phase transitions [48]. As mentioned previously, the ability to study these spatio-temporal phenomena in a delay system is great benefit to experimentalists because delay systems are often easier to measure and control. We pursue the link between delay systems and spatio-temporal systems further in Chapter 4.

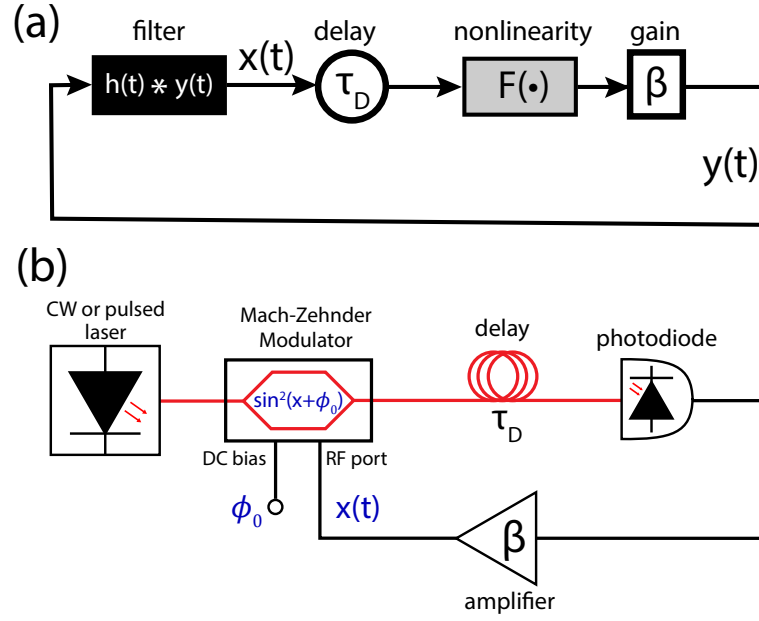


Figure 2.1: Nonlinear delayed feedback system. (a) Block diagram of a delay system.  $y(t) = \beta F(x(t - \tau_D))$  is the input to the linear filter described by the impulse response  $h(t)$ , and  $x(t)$  is the filter output. (b) Experimental setup of an opto-electronic oscillator delayed feedback system. The filtering is performed either by the component with the narrowest bandwidth (usually the photodiode) or by a stand-alone filter (not shown). The oscillator can be a discrete-time map (when powered by a pulsed laser) or a continuous-time system (when powered by a CW laser).

### 2.2.2 Brief history of opto-electronic oscillators

The delay system under study for most of this thesis is the opto-electronic oscillator. The opto-electronic oscillator was originally studied in bulk optics [49] and soon after implemented using standard telecommunications components [50]. These systems have been found to be extremely rich in their dynamics, in part because they can span an enormous range of time scales [27, 51]. The fact that such an interesting dynamical system can be created with cheap, commercially available components makes the opto-electronic oscillator a particularly attractive system for experimentalists. Opto-electronic oscillators have been used to study chaotic breathers [52], broadband chaos [53], network dynamics [23, 54–57], and the transition from noise to chaos [58]. Additionally, opto-electronic oscillators are useful for a variety of applications, including the generation of high spectral purity microwaves [59], chaos communications [60, 61], and reservoir computing [22, 62]. For a recent review of opto-electronic oscillators, see Ref. [63].

An illustration of the type of opto-electronic oscillator studied here is shown in Fig. 2.1b. Constant intensity light from a fiber-coupled CW laser passes through an integrated electro-optic Mach-Zehnder intensity modulator, which provides the nonlinearity  $F(x) = \sin^2(x + \phi_0)$ . The quantity  $x(t)$  represents the normalized voltage applied to the intensity modulator, the function  $F(\cdot)$  represents the normalized optical power transmitted by the modulator, and  $\phi_0$  is the normalized DC bias voltage. The time delay is implemented by an optical or electronic delay line. The filtering is performed either by the photodiode (the component with the narrowest

bandwidth) or a stand-alone analog [59] or digital [64] filter (not shown).

The original opto-electronic oscillators could be modeled by Eq. 2.1. However, in order to operate the oscillators at higher frequencies, high-speed microwave amplifiers, which rarely go down to DC, were used. Novel dynamics that cannot be explained by Eq. 2.1 were observed in these systems [20, 52]. In order to model the oscillator with band-pass filter (rather than a low pass filter), an integral term must be added to Eq. 2.1:

$$\tau_L \dot{x}(t) + \left(1 + \frac{\tau_L}{\tau_H}\right)x(t) + \frac{1}{\tau_H} \int_{-\infty}^t x(s)ds = \beta F(x(t - \tau_D)) \quad (2.2)$$

where  $\tau_D$  is the time delay,  $\tau_L = 1/2\pi f_L$  is the low pass filter response time, and  $\tau_H = 1/2\pi f_H$  is the high pass filter response time. By considering the limit  $\tau_H \rightarrow \infty$  (i.e., the case of a low pass instead of a band pass filter), the integral term vanishes and Eq. 2.2 reduces to Eq. 2.1.

Alternatively, the system can be turned into a discrete-time map by pulsing the laser at a repetition rate  $f_r = N/\tau_D$  [65]. In this case, the system can be modeled as

$$x[k] = \beta \sum_{m=-\infty}^k h[k-m]F(x[m-N]) \quad (2.3)$$

where  $x[k]$  is the height of the  $k^{th}$  electrical pulse applied to the modulator,  $h$  is the impulse response [66] of the filter sampled at the repetition rate  $f_r$ . An equivalent discrete-time system can be created by using a CW laser, digitally sampling the electronic signal using an analog-to-digital converter (ADC) clocked at rate  $f_r$ , and driving the modulator with a digital-to-analog converter (DAC) clocked at  $f_r$ , as

is often done to implement a variable time delay via a digital delay line [64]. We take this approach in Chapter 4. As the repetition rate  $f_r \rightarrow \infty$ , time becomes continuous, the sum becomes a convolution integral, and we obtain

$$x(t) = h(t) * \beta F(x(t - \tau_D)) = \beta \int_{-\infty}^{\infty} h(t - t') F(x(t' - \tau_D)) dt' \quad (2.4)$$

$$= \beta \int_{-\infty}^t h(t - t') F(x(t' - \tau_D)) dt' \quad (2.5)$$

where in the last step we use the property that  $h(t)$  is causal. Therefore, this system allows for the study of the transition from discrete to continuous time in chaotic systems [65, 67].

The convolution integral Eq. 2.4 is a general model for a continuous-time single variable delay system (including opto-electronic oscillators). Eqs. 2.1 and 2.2 are special cases of Eq. 2.4 for specific filter impulse responses. This convolution integral formalism is particularly useful when combined with the space-time interpretation of delay systems [18] to realize networks of coupled maps from a single continuous-time opto-electronic oscillator, as will be discussed in detail in Chapter 4.

## 2.3 Experimental implementation

In this section, we outline the details of the opto-electronic oscillators that were used to perform the experiments that will be discussed in Chapter 3. This same apparatus was used to perform the measurements in Refs. [54–57] and is described in full detail (circuit diagrams, etc.) in Refs. [29, 30]. The main advantage of this system over other opto-electronic oscillators is that the time delay and filtering are



performed digitally; here this is done on a digital signal processing (DSP) board, while in Chapters 4 and 5 we will use a field-programmable gate array (FPGA). Both methods allow the time delay to be controlled digitally rather than by carefully measuring and splicing delay lines made of optical fiber. This is especially useful for our experiments in Chapter 3, where we perform extensive parameter sweeps of the time delay. Similarly, having a digital filter allows us to experiment with different kinds of filtering, which we do extensively in Chapters 4 and 5. Here, we use the digital filtering to slow the system down to audio frequencies, which allows us to use low-cost electronic components.

An illustration of the apparatus of a single opto-electronic oscillator is shown in Fig. 2.2a. A 1550nm fiber-coupled distributed feedback laser diode (Bookham) serves as the optical power source. The light passes through a Lucent 2623NA electro-optic intensity modulator. In the modulator, the light is split into two arms, each of which passes through a  $\text{LiNO}_3$  electro-optic crystal. The voltage applied to the modulator creates an electric field, changing the index of refraction of the crystals. The electric fields are in opposite directions, so that the resulting changes in refractive index of the two arms are opposite in sign. When the two arms recombine at the output of the modulator, there is interference between the two optical waves. Therefore, the output of the modulator can be modeled as  $P_{out} = P_{max} \sin^2(x(t) + \phi_0)$  where  $P_{max}$  is the maximum optical power through the modulator and  $(x(t) + \phi_0)$  is the normalized voltage applied to the modulator. The normalized time-dependent voltage is given by  $x(t) = \pi v(t)/2V_\pi$ , where the half-wave voltage is given by  $V_\pi = 3.4$  V. The normalized DC bias  $\phi_0$  is added to  $v(t)$  with an adder circuit.

The light that passes through the modulator is detected by a photoreceiver circuit, the diagram of which can be found in Ref. [30]. The circuit is a photodiode (to convert the light to a photocurrent) followed by a transimpedance amplifier (to convert the current to an amplified voltage) with a gain of 1000 V/A. This voltage is read by the analog-to-digital converter (ADC) of the DSP board (Texas Instruments TMS320C6713 TSK), time delayed and digitally filtered on the board. The digital filter we use for the experiments in Chapter 3 is a first order Butterworth band-pass filter with cut-on frequency  $f_H = 100$  Hz and cut-off frequency  $f_L = 2.5$  kHz. The DSP sampling frequency is 24 kSamples per second. The filter output is turned into a voltage by the DAC of the DSP board then amplified by an inverting amplifier with a gain of  $V_{out}/V_{in}=-20$ . The output of this amplifier is the time-varying voltage  $v(t)$  applied to the modulator, closing the feedback loop.

## 2.4 Modeling opto-electronic oscillators

In this section, we describe two models for the opto-electronic oscillator described in Section 2.3: a continuous-time model and a discrete-time model. We lump all the gains and losses into a single round-trip gain  $\beta$  that can be measured experimentally with a network analyzer. Therefore, the input to the filter  $y(t)$  in terms of the filter output  $x(t)$  is

$$y(t) = -\beta \sin^2(x(t - \tau_D) + \phi_0), \quad (2.6)$$

where the minus sign comes from the fact that we use an inverting amplifier.

As mentioned in Section 2.2.2, Eq. 2.2 is a good model for opto-electronic



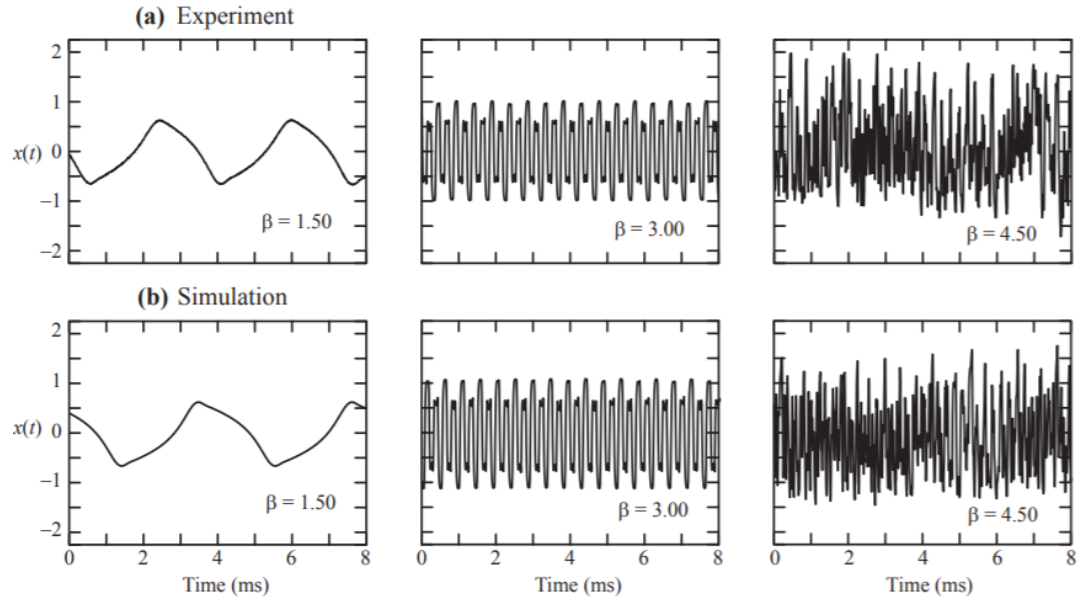


Figure 2.3: Dynamical behavior of the opto-electronic oscillator. The complexity of the oscillator dynamics generally increases with  $\beta$ . *a*) Experimentally measured time series. *b*) Time series from numerical simulations. Figure from Ref. [29].

oscillators when there is band-pass filtering. Therefore, using the modulator's non-linearity  $F(x) = \beta \sin^2(x + \phi_0)$ , we have

$$\tau_L \dot{x}(t) + \left(1 + \frac{\tau_L}{\tau_H}\right)x(t) + \frac{1}{\tau_H} \int_{-\infty}^t x(s) ds = -\beta \sin^2(x(t - \tau_D) + \phi_0). \quad (2.7)$$

Because the integral can be cumbersome for both analytical and numerical exploration of these systems, Eq. 2.7 is often written as

$$\begin{aligned} \dot{x}(t) &= -(\omega_L + \omega_H)x(t) - \omega_L z(t) - \omega_L \beta \sin^2(x(t - \tau_D) + \phi_0) \\ \dot{z}(t) &= \omega_H x(t), \end{aligned} \quad (2.8)$$

where  $\omega_H = 2\pi f_H$  and  $\omega_L = 2\pi f_L$ . Physically,  $x(t)$  is the normalized voltage applied to the modulator and  $z(t)$  represents an unobserved filter state variable. For a detailed analysis of Eq. 2.8, see Ref. [27]. Eq. 2.8 is sometimes written more succinctly in filter notation as [64]

$$\dot{\mathbf{u}}(t) = \mathbf{E}\mathbf{u}(t) - \mathbf{F}\beta \sin^2(x(t) + \phi_0), \quad (2.9)$$

$$x(t) = \mathbf{G}\left(\mathbf{u}(t - \tau_D)\right) \quad (2.10)$$

where

$$\mathbf{E} = \begin{bmatrix} -(\omega_L + \omega_H) & -\omega_L \\ \omega_H & 0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \omega_L \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Again,  $x$  is the normalized voltage applied to the modulator, and  $\mathbf{u}(t)$  describes the state of the filter.

These continuous-time models do provide an accurate model of our opto-electronic oscillators. However, because of the complex dynamics exhibited by these systems, Eq. 2.8 typically needs to be solved numerically. Furthermore, because we use a DSP board in our feedback loop, our opto-electronic oscillators are fundamentally discrete-time systems. Instead of discretizing Eq. 2.8 directly, we use a discrete-time model taken directly from the discrete-time filter equations implemented on the DSP board.

Our DSP board implements a first order (two-pole) Butterworth band-pass filter:

$$x[n] = -a_1x[n-1] - a_2x[n-2] + b_0y[n] + b_1y[n-1] + b_2y[n-2], \quad (2.11)$$

where  $x[n]$  is the filter output at discrete time  $n$  and  $y[n]$  is the filter input at discrete time  $n$ . The filter coefficients are given by  $a_n$  and  $b_n$  and depend on the filter type, the cut-on and cut-off frequencies, and the sampling rate. Simplifying Eq. 2.11 leads to the discrete-time model of our opto-electronic oscillator

$$x[n] = -a_1x[n-1] - a_2x[n-2] + b_0(y[n] - y[n-2]), \quad (2.12)$$

where  $y[n] = -\beta \sin^2(x[n - k_D] + \phi_0)$  (the discretized version of Eq. 2.6), where  $k_D$  is the number of digital filter time steps per time delay. This is the model we use for the numerics in all of our direct simulations and stability calculations.

Comparison of experimentally measured and numerically computed time series are shown in Fig. 2.3. The complexity of the dynamics generally increase with the round trip gain  $\beta$ , from fixed point (not shown) to periodic dynamics and finally

chaotic dynamics. The simulations agree well with the experiment, validating the model.

## 2.5 Coupled opto-electronic oscillators and synchronization

Synchronization is a phenomenon in which two (or more) oscillating dynamical systems adjust their rhythms due to an interaction (or “coupling”) between them [14]. When thinking of synchronization, people often think of isochronal synchronization, in which two (or more) oscillators exhibit exactly the same dynamical behavior, at exactly the same time. Of course, due to the heterogeneities, time delays, noise, and other imperfections inherent in experiments, we have to relax the “exactly” and replace it with “approximately.”

Synchronization between periodic oscillators has been studied since Huygens’ observation of the anti-phase synchronization of pendulum clocks coupled through the wall [68]. Synchronization of periodic oscillators is used every day in radio communications, in which the receiver synchronizes with the transmitter. Once it was understood how two periodic oscillators could synchronize, people began to wonder how many (not necessarily identical) limit cycle oscillators could synchronize. These questions were motivated by, for example, the synchronized blinking of fireflies in Thailand [69] and the synchronized beating of heart cells [70]. A magnificent theoretical advance in this area came with the development of Kuramoto’s model of weakly coupled limit cycle oscillators [71], through which Kuramoto was able to show analytically that oscillators whose natural frequencies were not too different

could all synchronize. The Kuramoto model has been so successful because it is an analytically tractable model that displays a wide variety of complex dynamical behaviors including complete [71, 72] and partial synchronization [73, 74] and phase transitions [72].

However, many real world networks are not composed of weakly coupled limit cycle oscillators. The synchronization of two chaotic oscillators was first demonstrated in the 1980s and early 1990s by several different groups independently [11–13, 75–77]. Soon after, work began on global synchronization in networks of coupled chaotic oscillators, culminating in the development of the master stability function approach [78], which theoretically solved the problem of the global synchronization of networks of coupled identical oscillators by showing that the synchronizability of a network of coupled oscillators depends on only the eigenvalues of the network adjacency matrix (described in the next section) rather than on the full details of the network topology.

Despite these theoretical advances, network experiments lagged behind because it is typically difficult and expensive to build many nonlinear oscillators that have tunable parameters and can be coupled together in a reconfigurable manner. Some of the first experimental networks were made with electronic circuits [79] and free space lasers with optical coupling [80]; these were constrained to networks of three oscillators.

Our system is an attractive solution for studying small networks of optoelectronic oscillators. Our oscillators are made of low-cost, highly reliable standard telecommunications components. All of the light is contained in fiber, which allows



us to avoid tedious optical alignments. And the DSP boards simplify the tuning of parameters such as delay time, round trip gain, and coupling strength.

One of the great benefits of our opto-electronic oscillators is that, because they are made of standard telecommunications components, they are low-cost and relatively easy to build. Two (or more) of these oscillators can be built and allowed to communicate with one another in order to experimentally study coupled dynamical systems.

In this section, we first describe how we couple together up to four of our opto-electronic oscillators in a network with arbitrary topology. Then we briefly define the network adjacency matrix and present a model of our network of coupled opto-electronic oscillators.

### 2.5.1 Experimental implementation

We have built four nominally identical opto-electronic oscillators as described in Section 2.3. We couple them together optically as shown in Fig. 2.2b. The light that passes through the intensity modulator is split using a  $1 \times 4$  optical splitter. One of these four signals is detected by the feedback photoreceiver and the other three are sent out as coupling signals, one to each of the other three nodes. There is a variable optical attenuator on the fiber containing each coupling signal, allowing for control of the coupling strengths and topology. The coupling inputs from the other three nodes are summed using an optical combiner and detected all at once by the coupling photoreceiver (Note: the optical signals add incoherently because the differences

in optical frequencies are much greater than the bandwidth of the photoreceiver). Each DSP board thus receives two input electrical signals: one feedback signal and one coupling signal. The feedback and coupling signals are delayed and filtered on the DSP board. They are then combined on the DSP board, which also controls the feedback and overall coupling strength. Finally, the combined signal is output through the DAC, amplified, and applied to the modulator, as before. A photograph of our experimental apparatus is shown in Fig. 2.4.

## 2.5.2 Modeling networks of coupled opto-electronic oscillators

In this section we will first formalize what we mean by networks of oscillators using the concept of an adjacency matrix. We will discuss two different types of coupling. Finally, we will present both continuous- and discrete-time models of our network of opto-electronic oscillators.

The topology of a network of  $N$  coupled oscillators can be described by an adjacency matrix. An adjacency matrix is an  $N \times N$  matrix  $A_{ij}$  such that  $A_{ij} = 1$  if a signal from node  $j$  is coupled into node  $i$ ; otherwise,  $A_{ij} = 0$ . A weighted adjacency matrix has elements  $W_{ij} \neq 0$  if a signal from node  $j$  is coupled into node  $i$ ; otherwise,  $W_{ij} = 0$ . Often, the equations of motion of a network of coupled oscillators can be written in the form

$$\dot{\mathbf{u}}_i(t) = \mathbf{F}(\mathbf{u}_i(t)) + \sigma \sum_j A_{ij} \mathbf{H}(\mathbf{u}_j(t)), \quad (2.13)$$

where  $\mathbf{u}_i$  is a vector describing the state of the  $i^{th}$  oscillator in the network, and  $\sigma$



Figure 2.4: Photograph of our 4-node network of opto-electronic oscillators.

is the coupling strength.

In general, Eq. 2.13 does not permit a globally synchronized solution. Laplacian coupling is often implemented to create a network that allows for the possibility of global synchronization. The coupling is called Laplacian because the adjacency matrix in Eq. 2.13 is replaced with a Laplacian matrix. In order to obtain the Laplacian matrix from the corresponding adjacency matrix, one simply adjusts the diagonal such that the sum along each row is zero (Note: this is actually a negative Laplacian matrix, but we use it here for notational consistency with the adjacency matrix). In other words,  $L_{ii} = -\sum_j A_{ij}$ , and  $L_{ij} = A_{ij}$  if  $i \neq j$ . For a Laplacian network, the equations of motion are

$$\dot{\mathbf{u}}_i(t) = \mathbf{F}(\mathbf{u}_i(t)) + \sigma \sum_j L_{ij} \mathbf{H}(\mathbf{u}_j(t)) \quad (2.14)$$

$$= \mathbf{F}(\mathbf{u}_i(t)) + \sigma \sum_j A_{ij} \mathbf{H}(\mathbf{u}_j(t)) - \mathbf{H}(\mathbf{u}_i(t)) \quad (2.15)$$

We say Laplacian coupling allows global synchronization because when all the oscillators in the network are initialized with the same initial condition ( $\mathbf{u}_i = \mathbf{u}_j$  for all  $i, j$ ), the coupling term vanishes and so all the oscillators stay synchronized for all time in the absence of any perturbations.

Our network of opto-electronic oscillators is capable of implementing either adjacency matrix or Laplacian coupling because the DSP board is able to distinguish between feedback and coupling signals. To implement adjacency matrix coupling, the DSP board simply scales the coupling signal by  $\sigma$  and adds the result to the feedback signal. Alternatively, to implement Laplacian coupling, the DSP board

corresponding to the  $i^{th}$  oscillator again scales the coupling signal by  $\sigma$  but also scales the feedback signal by  $1 - n_i^{in}\epsilon$ , where  $n_i^{in}$  is the number of input signals that are combined to form the coupling signal ( $n_i^{in} = \sum_j A_{ij}$ ).

In Chapter 3, we use the Laplacian-like coupling scheme depicted in Fig. 2.2b. Our network of opto-electronic oscillators can be modeled in continuous time as

$$\begin{aligned} \dot{x}_i(t) = & -(\omega_L + \omega_H)x_i(t) - \omega_L z_i(t) \\ & - \omega_L \beta \left\{ I(x_i(t - \tau_f)) + \sigma \sum_j A_{ij} \left( I(x_j(t - \tau_c)) - I(x_i(t - \tau_f)) \right) \right\} \end{aligned} \quad (2.16)$$

$$\dot{z}_i(t) = \omega_H x_i(t),$$

where the feedback delay  $\tau_f$  and the coupling delay  $\tau_c$  are not necessarily equal, and  $I(x) = \sin^2(x + \phi_0)$  is the normalized light intensity that passes through the modulator. We call this Laplacian-like because  $\tau_f$  and  $\tau_c$  are in general different, so the coupling term does not cancel and the system does not necessarily admit a globally synchronized solution (global synchronization is only possible in the case that all row sums of the coupling matrix are equal). Only in the case that  $\tau_f = \tau_c$  does Eq. 2.16 represent Laplacian coupling.

Because they are linear, we can commute the filtering, delay, amplification, and coupling. Therefore we can also write this in the filter notation form of Eqs. 2.9 and 2.10

$$\dot{\mathbf{u}}_i(t) = \mathbf{E}\mathbf{u}_i(t) - \mathbf{F}\beta \sin^2(x_i(t) + \phi_0) \quad (2.17)$$

$$x_i(t) = \mathbf{G}\left(\mathbf{u}_i(t - \tau_f) + \sigma \sum_j A_{ij}(\mathbf{u}_j(t - \tau_c) - \mathbf{u}_i(t - \tau_f))\right). \quad (2.18)$$

Both of these are mathematically equivalent models of our opto-electronic oscillator network; however, they represent different physical coupling schemes. Equation 2.16 represents optical coupling of the modulator output (which is how the coupling is actually done), while Eqs. 2.17 and 2.18 represent electronic coupling of the filter output [64]. We use the models interchangeably depending on which is most mathematically convenient for the calculation that is being performed.

As mentioned above, for numerical simulations, we often use a discrete model of our oscillators. The coupling is added to the discrete-time Eq. 2.12 as follows:

$$x_i[n] = -a_1 x_i[n-1] - a_2 x_i[n-2] - \beta b_0 (r_i[n] - r_i[n-2]) \quad (2.19)$$

$$r_i[n] = \sin^2 \left\{ x_i[n - k_f] + \sigma \sum_j A_{ij} (x_j[n - k_c] - x_i[n - k_f]) + \phi_0 \right\}, \quad (2.20)$$

where  $k_f$  and  $k_c$  are the feedback and coupling delays in units of time steps.

These models have been tested and verified for use in modeling coupled opto-electronic oscillators for applications in adaptive synchronization [81], prediction of chaos [28], and network dynamics [54–56]. In Chapter 3, we use these models to predict the stability of chimera states and patterns of cluster synchrony in our four node network of opto-electronic oscillators. There, too, we find that the models predictions are verified by our experimental measurements.

## Chapter 3: Patterns of synchrony in a 4 node globally coupled network of optoelectronic oscillators

*This section is based on work from the following publication:*

*Joseph D Hart, Kanika Bansal, Thomas E Murphy, & Rajarshi Roy. “Experimental observation of chimera and cluster states in a minimal globally coupled network.” Chaos, 26(9), 094801 (2015).*

### 3.1 Overview

A “chimera state” is a dynamical pattern that occurs in a network of identical oscillators coupled in a homogeneous way when the symmetry of the oscillator population is broken into synchronous and asynchronous parts. “Cluster synchrony” occurs when a network can be divided into different sets of oscillators such that oscillators in the same set are synchronized. In this chapter, we report the experimental observation of chimera and cluster synchronous states in a network of four globally coupled chaotic opto-electronic oscillators. This is the minimal network that can support chimera states, and our study provides new insight into the fundamental mechanisms underlying their formation. Cluster synchronization has thus far been

studied independently of chimera states; however, we use a unified approach to determine the stability of all the observed patterns of synchronization, highlighting the close relationship between chimera and cluster states.

We obtain two important results: A) we provide a first experimental demonstration that chimeras can appear in small networks, contrary to the conventional assumption that a large network with non-local coupling is necessary [82], and B) we show that both cluster states and chimera states can be regarded as special cases of the more general phenomenon of partial synchronization. Our methods apply to networks of different size and topology, opening up potential applications to chimeras and other patterns of synchronization in real world networks such as power grids.

First, we provide a brief description of previous work on chimera states and cluster synchronization in Sections 3.2.1 and 3.2.2, respectively. Section 3.3 describes the theory of cluster synchronization, including when clusters can form (Section 3.3.1) and how to determine their linear stability (Section 3.3.2).

In Section 3.2.2 we show how these ideas can be applied to chimera states in a network of four globally coupled opto-electronic oscillators with time delays in the feedback and coupling. In particular, we show that chimera states emerge from partial (or subgroup) symmetries in the network topology, and we calculate their linear stability using the methods presented in Section 3.3, highlighting that chimera and cluster states are closely related patterns of partial synchrony.

The results of our experiments and stability calculations are compared in Section 3.5. We conclude with a discussion of the importance of multistability of par-



tially synchronous states for the existence of chimera states in Section 3.6.

## 3.2 Background

### 3.2.1 Chimera states

Since their original discovery [73, 74], there has been a great deal of discussion about the definition of chimera states and the conditions for their existence. One definition is that a “chimera state” is a spatio-temporal pattern in which a system of identical oscillators is split into coexisting regions of coherent (phase and frequency locked) and incoherent (drifting) oscillation [82]. It was originally thought that chimeras could exist only in large networks of non-locally coupled oscillators and only from special initial conditions [82]. These assumptions were reflected in the decade-long gap between their theoretical discovery [73] and the first experimental realization of chimeras in a spatial light modulator feedback system [83] and chemical oscillator system [84]. However, recent studies have shown that chimeras can actually appear in a much wider variety of networks: chimeras have now been observed experimentally in a mechanical system of metronomes [85], optical frequency combs [86], electrochemical systems [87], Lorenz oscillators [88], and electronic and opto-electronic delay systems [19, 20, 89]. This suggests that chimeras may exist more widely than at first expected. Indeed, recent numerical and experimental work has found chimeras in small networks [88, 90], from random initial conditions [91, 92], and with global coupling [91, 93–95]. Additionally, chimera-like states with one large cluster and many small ones have long been observed in numerical simulations from

random initial conditions [96] and for global coupling [97]. These are well beyond the conditions initially assumed necessary for the existence of chimera states.

While chimeras can exist in many different systems, one common characteristic seems to be that chimeras often appear in regions of multistability with other synchronous patterns [20, 84, 85, 97, 98]. Recently, Böhm et al. proposed a network of four globally coupled lasers in which chimera states can emerge from random initial conditions and have linked the emergence of chimeras to a multistable region of parameter space [99].

We find it useful to classify chimera states into two different classes. We call one class “coherent chimeras” because the oscillators in the coherent region are phase-locked but not perfectly synchronized. The chimeras originally discovered by Kuramoto [73] are of this type. We call the second class “synchronized chimeras” because the coherent region consists of identically synchronized oscillators. While this distinction is not often made in the literature, it is important here: The analytical techniques we use work for only synchronized chimeras.

### 3.2.2 Patterns of synchronization

When we talk about “patterns of synchronization,” we mean a dynamical state of a network of coupled oscillators in which some fraction of the oscillators are synchronized. As used here, “patterns of synchronization” include global synchronization of networks, cluster synchronization on networks, and chimera states. “Cluster” is the term used to describe a set of nodes that synchronize. Global syn-

chronization occurs when all nodes in a network synchronize identically (i.e., when there is only one cluster). Cluster synchronization occurs when nodes within a cluster synchronize with each other but not with nodes in other clusters. In this sense, synchronized chimeras can be considered to be cluster synchronous states.

We are especially concerned with three questions related to patterns of synchronization:

1. What sets of nodes can synchronize with each other? In other words, what patterns of synchronization are permitted to exist by the equations of motion?
2. Given a pattern of synchronization that is permitted, *does* it synchronize? And does it remain synchronized in the presence of small perturbations? This is a question of the stability of the pattern of synchronization.
3. Given a stable cluster synchronous state, what do the dynamics look like? How well do the clusters synchronize?

### **Global synchronization**

Global synchronization is permitted by the equations of motion of a network (Eq. [2.13](#)) when all nodes of the network have the same number of inputs (i.e., when the sum of the entries in a row of the adjacency matrix is the same for each row). The most common such network is a Laplacian network, in which the row sums are all zero.

The problem of the stability of global synchronization in Laplacian networks was solved by the development of the Master Stability Function [\[78\]](#). The Master

Stability Function determines the stability of global synchronization and depends only on the eigenvalues of the Laplacian coupling matrix and the dynamics of a single node. It is straightforward to extend the Master Stability Function to the case when all row sums are equal but not zero [100]. The phenomenon of global synchronization has been well studied in the literature (e.g., the review in Ref. [101]) and will not be discussed further here.

### Cluster synchronization

The general phenomenon of cluster synchrony encompasses several partial synchrony phenomena including group synchronization [55, 102] and remote synchronization [103–105]. Cluster synchrony has been found in networks of chaotic oscillators [55, 102, 106], of Kuramoto oscillators [104], and with time delay in the coupling [107]. All of these studies of cluster synchronization focused on specific networks in which there is obvious symmetry in the network structure or for particular oscillator dynamics that invite cluster synchronization.

The appearance of cluster synchronization in such a variety of systems motivates the need for a general theory of cluster synchronization and its stability. Section 3.3.1 will be dedicated to the theory of the formation of cluster synchronization. The stability of cluster synchronization will be discussed in Section 3.3.2.

In the following, we consider only *model independent* cluster synchronization; that is, patterns of synchronization that are allowed by the network architecture independent of the individual node dynamics. While our discussions are equally valid for networks of discrete-time oscillators, for simplicity we will consider continuous-

time networks that can be modeled by Eq. 2.13.

### 3.3 Theory of cluster synchronization

#### 3.3.1 Existence of synchronous clusters

The question of the existence of synchronous clusters was thoroughly investigated by the group of Ian Stewart and Martin Golubitsky using abstract math techniques from the theory of symmetry groupoids [108, 109]. They showed that the cluster states that are permitted by the network equation Eq. 2.13 can be revealed by finding “equivalence relations that are balanced” (in graph theory, these are sometimes called balanced colorings or equitable partitions). In other words, a particular pattern of synchronization is allowed when all the nodes of each cluster receive the same number of inputs from all other clusters.

Pecora et al. showed that many of the allowed patterns of cluster synchronization can be revealed by an analysis of the symmetries of the network topology. This method of identifying the allowed patterns of cluster synchronization is particularly useful because it allows one to take advantage of extremely efficient algorithms from computational group theory [110, 111]; however, there are some equitable partitions that are not revealed by a symmetry analysis [112]. Pecora et al. also showed that the symmetry analysis was helpful in determining the stability of the symmetric cluster synchronous states [113], as we will discuss in Section 3.3.2.

We now present a few definitions that will be useful for discussing cluster synchronization. These definitions will be related to methods of partitioning a network

of coupled oscillators into disjoint sets of nodes. Two methods of partitioning networks are especially useful for the study of cluster synchronization: orbital partitions and equitable partitions.

### Orbital partitions

An orbital partition is a partition of a network according to the orbits of its symmetry group. The permutation symmetries of a network are the set of relabelings of network nodes that do not change the network topology and leave the equations of motion invariant. These symmetry operations form a mathematical group. The orbits of the symmetry group are the disjoint sets of nodes that are permuted among one another by the symmetry operations. For example, consider the network in Fig. 3.1. Nodes 1, 2, and 3 can be permuted by rotations about node 0 (or by reflections); but no symmetry operation can permute node 0 with any of the other nodes. Therefore, the orbits of this star network are  $\{0\}$  and  $\{1,2,3\}$ . This example was taken from Ref. [114]. Through the rest of this thesis, we use the term “symmetry cluster” to refer to a set of nodes in the orbital partition of a network. When we refer to “the orbital partition,” we mean the coarsest of all possible orbital partitions.

### Equitable partitions

An equitable partition is a partitioning of a network into disjoint sets of nodes such that each node in a set has the same total input from each of the other sets. Stated more precisely, in an equitable partition, any node  $x_k$  in set  $S_i$  has exactly  $N_{ij}$  inputs from nodes in set  $S_j$  for all  $j$  regardless of the choice of  $k$ . We use the term

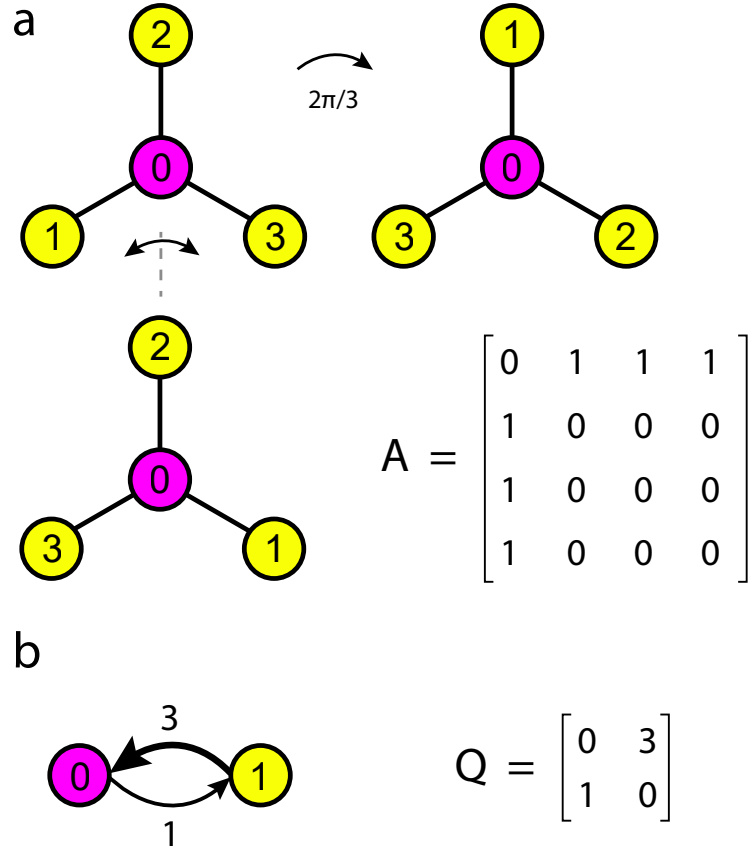


Figure 3.1: *a*) Simple example of a network with symmetries. A rotation by  $2\pi/3$  or a reflection about a vertical line through node 0 leaves the network invariant. These nodes that are permuted by symmetry operations determine the orbits, or symmetry clusters. The orbits of this network are  $\{0\}$  and  $\{1,2,3\}$ . The adjacency matrix  $\mathbf{A}$  is shown. *b*) The quotient network for the star network shown in *a*. The adjacency matrix  $\mathbf{Q}$  for the quotient network is shown.

“input cluster” to refer to a set of nodes that forms a set of the equitable partition of a network. When we refer to “the equitable partition,” we are talking about the coarsest of all possible equitable partitions. A scalable algorithm for finding the equitable partition of a network can be found in Ref. [115].

In order to illustrate the idea of an equitable partition, consider again the network in Fig. 3.1a. All nodes are identical in this network; the color is used only to identify which equitable partition set each node is in. Each of the yellow nodes receives one input from the magenta node and no inputs from other yellow nodes. Similarly, the magenta node receives inputs from all three of the yellow nodes and no inputs from other magenta nodes.

All orbital partitions are equitable partitions, but not all equitable partitions are orbital partitions [112]. When an equitable partition is not also an orbital partition, more care must be taken when deriving the variational equations for stability calculations. This case will be considered in detail in Section 5.5.

### **Relation to cluster synchronization**

As stated previously, the clusters that are allowed to synchronize are given by the equitable partitions [112]. It is easy to see why: Nodes in the same input cluster have the exact same equations of motion when the entire network is in the cluster synchronous state represented by the equitable partition. Therefore, if the network is started in that cluster synchronous state, it will remain there in the absence of any external perturbations.

This says nothing about the cluster state’s stability, which determines if the



dynamical state is robust to the small perturbations that will occur in any real-world experiments. The symmetries – in particular, the irreducible representations (IRRs) of the symmetry group – are essential for determining the stability of the clusters, even if the clusters are not symmetry clusters. This will be discussed further in Section [3.3.2](#).

### 3.3.2 Stability analysis for patterns of synchronization

In this section, we describe how to perform the stability analysis for orbital clusters. It closely follows the procedure developed in Ref. [\[113\]](#). The extension of the stability analysis to input clusters is presented in Section [5.5.2](#).

The outline of the stability analysis is as follows:

1. Find the dynamical solution of the desired pattern of synchronization using the equations of motion of the quotient network. This can be done by forcing the equations of motion into the desired pattern of synchronization using a “quotient network.”
2. Linearize about the cluster synchronous solution to obtain the variational equations.
3. Change coordinates of the variational equations from the “node coordinate space” to the “IRR coordinate space” associated with the cluster synchronous state. This transformation separates out the cluster synchronization manifold and the transverse directions, similar to what is done in the Master Stability Function approach [\[78\]](#).

4. Use standard numerical algorithms to calculate the Lyapunov exponents of the transformed variational equations. The Lyapunov exponents corresponding to the direction(s) transverse to a given synchronized cluster determine the stability of that cluster.

### Step 1: Quotient networks

The cluster synchronous solution is determined by forcing a desired pattern of synchronization upon the equations of motion and then solving them, typically numerically. One way of forcing a desired pattern of synchronization upon the equations of motion is to use the concept of a quotient network.

Consider a pattern of cluster synchronization with  $M$  clusters. A quotient network is a reduction of the full network into an  $M$ -node network, where the connectivity between the  $M$  nodes in the quotient network corresponds to the inter-cluster connectivity in the true network. The adjacency matrix  $\mathbf{Q}$  that describes the quotient network can be determined from the true network adjacency matrix  $\mathbf{A}$  according to

$$Q_{\mu\nu} = \sum_{j \in S_\nu} A_{\mu j}, \quad (3.1)$$

where  $\mu$  labels the cluster and  $S_\nu$  is the set of nodes corresponding to the  $\nu^{th}$  cluster. The quotient network for the full network in the triplet-singlet cluster state shown in Fig. 3.1a is shown in Fig. 3.1b. Once the quotient network is obtained, the cluster synchronous solution can be determined by solving

$$\frac{d}{dt}\mathbf{s}_\mu(t) = \mathbf{F}(\mathbf{s}_\mu(t)) + \sigma \sum_{\nu} Q_{\mu\nu} \mathbf{H}(\mathbf{s}_\nu(t)), \quad (3.2)$$

where  $\mu$  labels the cluster and  $x_\mu^{(s)}$  is the cluster synchronous solution of the  $\mu^{th}$  cluster. For the same full network, different patterns of synchronization will have different quotient networks.

### Step 2: Linearize about the cluster synchronous solution

In order to determine the stability of the cluster synchronous state calculated in the previous step, we need to study the behavior of small perturbations away from the synchronized trajectory. Consider an infinitesimally small perturbation  $\delta\mathbf{u}_i$  to each node in Eq. 2.13. In order to obtain the equation describing the behavior of this perturbation, we make the change  $\mathbf{u}_i \rightarrow \mathbf{u}_i + \delta\mathbf{u}_i$  and linearize to obtain the variational equation

$$\frac{d}{dt}\delta\mathbf{u}_i(t) = D\mathbf{F}(\mathbf{s}_{\mu_i}(t))\delta\mathbf{u}_i(t) + \sigma \sum_j A_{ij} D\mathbf{H}(\mathbf{s}_{\nu_j}(t))\delta\mathbf{u}_j(t), \quad (3.3)$$

where  $\mathbf{s}_{\mu_i}(t)$  is the cluster synchronous solution corresponding to the cluster  $\mu$  to which node  $i$  belongs.  $D\mathbf{F}$  and  $D\mathbf{H}$  are Jacobian functions.

### Step 3: Transform to IRR coordinate space

Equation 3.3 describes how a small perturbation to a single node evolves in time. However, we are not concerned with all perturbations; we are concerned with only those perturbations that disturb the cluster synchrony. In other words, we need to separate out the perturbations along the synchronization manifold (which do not

affect the stability of cluster synchrony) from the perturbations transverse to the synchronization manifold (the evolution of which determines the stability of cluster synchronization). This is facilitated by changing coordinates from “node coordinate space” to “IRR coordinate space,” where this separation is easily done.

The transformation matrix  $\mathbf{T}$  for symmetry clusters can be calculated using the irreducible representations (IRRs) of the symmetry group. A software implementation of the technique developed in Ref. [113] can be found in Ref. [116].

When we transform to the IRR coordinate space by performing the change of variables  $\delta\mathbf{y}_a = \sum_b T_{ab}\delta\mathbf{u}_b$  and then left matrix-multiplying both sides by  $\mathbf{T}$ , we obtain

$$\frac{d}{dt}\delta\mathbf{y}_a(t) = D\mathbf{F}(\mathbf{s}_{\mu_a}(t))\delta\mathbf{y}_a + \sigma \sum_b B_{ab}D\mathbf{H}(\mathbf{s}_{\mu_b}(t))\delta\mathbf{y}_b(t), \quad (3.4)$$

where  $\mathbf{s}_{\mu_a}(t)$  is the cluster synchronous solution corresponding to the cluster  $\mu$  to which node  $a$  belongs and  $\mathbf{B} = \mathbf{TAT}^{-1}$ .

The rows of  $\mathbf{T}$  can be arranged such that  $\mathbf{B}$  is block diagonal. The block diagonal form of  $\mathbf{B}$  partially decouples the variational equations. For a pattern of cluster synchronization with  $M$  clusters, if the first  $M$  rows of  $\mathbf{T}$  are chosen so that they correspond to perturbations along the synchronization manifold,  $\mathbf{B}$  will have an  $M \times M$  block in the upper-left corner that couples together the perturbations along the synchronization manifold. This block is not used for stability calculations. The remaining blocks are called transverse blocks, and the Lyapunov exponents of Eq. 3.4 that correspond to those blocks determine the stability of the cluster

synchronization: if all these transverse Lyapunov exponents are negative, the cluster synchronous state is stable.

As an example, we return to the network in Fig. 3.1. The transformation matrix  $\mathbf{T}$  for this network is

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & -2 & 1 \end{bmatrix}. \quad (3.5)$$

The first two rows of  $\mathbf{T}$  correspond to perturbations along the synchronization manifold: the first row represents a perturbation of the magenta cluster, and the second row represents a perturbation that affects all nodes of the yellow cluster in exactly the same way. These perturbations cannot disturb the triplet-singlet cluster synchronization. The last two rows correspond to perturbations transverse to the synchronization manifold; they represent different ways that the yellow cluster can desynchronize.

The corresponding  $\mathbf{B}$  is

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.6)$$

$\mathbf{B}$  is block diagonal. The upper  $2 \times 2$  block corresponds to the dynamics of the variational equation (Eq. 3.4) in the synchronization manifold, which does not

impact the stability of the cluster synchronization. The lower two rows have “ $1 \times 1$  blocks” (with value 0) on the diagonal that represent perturbations transverse to the synchronization manifold; the Lyapunov exponents of Eq. 3.15 corresponding to these two rows determine the stability of cluster synchronization because they determine the growth or decay rates of small perturbations that disturb the synchronization of the yellow cluster.

The transformation to “IRR space” serves two purposes. Its main purpose is to separate out the transverse perturbations (which matter for the stability analysis) from the perturbations along the synchronization manifold (which do not matter for the stability analysis). It also partially (in the four-node example given, fully) decouples the the transverse perturbations from each other, which make the Lyapunov exponent calculations simpler.

#### **Step 4: Calculate the maximal Lyapunov exponents**

The Lyapunov exponents of the decoupled variational equations 3.4 determine the stability of the cluster synchronous state for which they are calculated. We calculate Lyapunov exponents using the standard *QR*-based algorithm [117].

### 3.4 Partial synchronization in an small globally coupled opto-electronic network

#### 3.4.1 Allowed patterns of synchrony in a globally coupled network

As mentioned above, the orbits of the full permutation symmetry group of the adjacency matrix make up the full symmetry clusters. The orbits of the subgroups of the symmetry group determine the subgroup symmetry clusters that can emerge via symmetry-breaking.

We now show that these ideas can be applied to chimeras in globally coupled networks of identical oscillators. For a globally coupled network of  $N$  nodes, the nodes are indistinguishable, so the group of permutation symmetries of the adjacency matrix is the symmetric group  $S_N$  (the group of all the permutations that can be performed on  $N$  nodes). Since any node can be permuted with any other node, the orbit of the symmetric group is all of the nodes, and the maximal symmetry case is global synchrony. To understand the allowed partial symmetry cases, the subgroups of the symmetry group must be considered. The orbits of the subgroups of  $S_N$  are such that *any* partition of the  $N$  oscillators is allowed to synchronize. In particular, a chimera state (that is, a state of one large synchronized cluster of  $N_s$  oscillators and  $N - N_s$  singlet “clusters”) is permitted by the equations of motion. Whether these chimera states should be observable in practice is determined by the linear stability analysis.

### 3.4.2 Experiments

The apparatus used in this experiment is described in detail in Section 2.3. The DSP board implements the filter described by Eq. 2.12 with a clock rate of 24 kHz. The digital filter coefficients used in Eq. 2.12 are given by  $a_1 = -1.49617$ ,  $a_2 = 0.50953$ ,  $b_0 = -b_2 = 0.24524$ , and  $b_1 = 0$ . Thus, the DSP implements a two-pole Butterworth bandpass filter with a cut-on frequency of 100 Hz and a cut-off frequency of 2.5 kHz. The opto-electronic oscillators are configured in a globally coupled network, depicted in Fig. 3.2.

For each trial of the experiment, the nodes are initialized from noise by recording the random electrical signal into the digital signal processing (DSP) board for 50 ms. Then feedback is turned on without coupling for 500 ms in order for transients to die out. At the end of this period, the coupling is turned on for 1450 ms. We use the last 400 ms of recording to determine which synchronous state is exhibited by the network.

For a network of four globally coupled nodes, the five possible patterns of synchrony are (Fig. 2): (a) the globally synchronized state, (b) the doublet-doublet state, (c) the triplet-singlet state, (d) the doublet-singlet-singlet (DSS) state, and the desynchronized state (not shown). We refer to doublet-doublet and triplet-singlet as “cluster states” and DSS as a “chimera state.”

We observe all possible patterns of synchrony in the experiment, as shown in Fig. 2(a-d), including chimera states that persist for many delay times and appear to be stable. We find similar time series in simulations without noise. For realizations



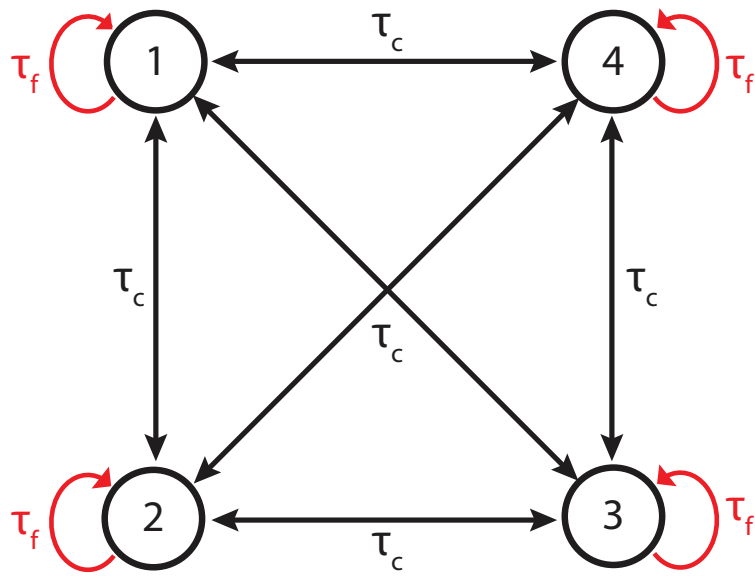


Figure 3.2: Sketch of our globally coupled network. Each node has a self-feedback with feedback time delay  $\tau_f$  (red) and is coupled to every other node with coupling time delay  $\tau_c$ .

from different initial conditions, nodes appear in different clusters, confirming that the partially synchronous patterns are not a result of parameter mismatch between the oscillators. As far as we are aware, this is the first time a chimera state has been experimentally observed in such a small network. In fact, this is the minimal network of globally coupled oscillators that can support a chimera state [90, 99]. We call this the minimal chimera because we believe there should be at least two oscillators in each region (the coherent region and the incoherent region). Others have suggested that a three-node network in which two of the nodes are synchronized should also be considered a chimera [118].

### 3.4.3 Stability analysis

While we perform the stability analysis for all possible patterns of synchronization for the four node network described in Section 2.5.1 with global coupling, we explicitly derive the variational equation for only the case of the chimera state (doublet-singlet-singlet cluster state). The derivation of the other patterns of synchrony were performed similarly.

The equations of motion of the chimera state are determined by enforcing the quotient network upon Eq. 2.17:

$$\frac{d}{dt}\mathbf{s}_\mu(t) = \mathbf{E}\mathbf{s}_\mu(t) - \mathbf{F}\beta \sin^2(x_\mu^{(s)}(t) + \phi_0) \quad (3.7)$$

$$x_{\mu_i}^{(s)}(t) = \mathbf{G} \left[ (1 - \sigma \sum_{\nu} Q_{\mu\nu}) \mathbf{s}_\mu(t - \tau_f) + \sigma \sum_{\nu} Q_{\mu\nu} \mathbf{s}_\nu(t - \tau_c) \right], \quad (3.8)$$

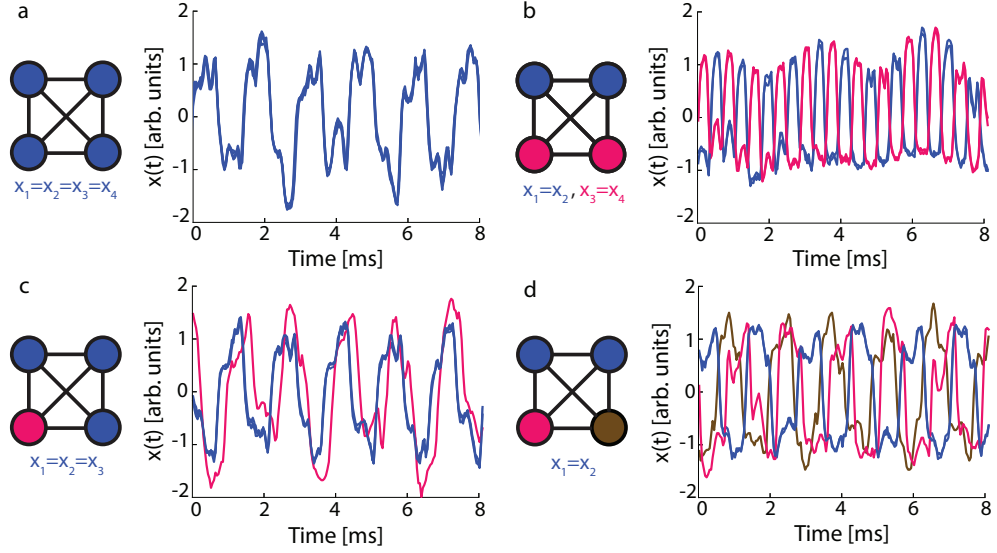


Figure 3.3: Experimentally observed patterns of synchrony. Illustration and typical experimental time series of all synchronous states for a globally coupled network of four nodes: (a) global synchrony, (b) doublet-doublet synchrony, (c) triplet-singlet synchrony, (d) doublet-singlet-singlet synchrony (chimera). Nodes of the same color are part of the same cluster. All measurements were performed with  $\beta = 3.8$  and  $\phi_0 = -\pi/4$ . This global state was observed with  $\sigma = 0.13$  and  $\tau_c = 1.8$  ms, the doublet-doublet and triplet-singlet states with  $\sigma = 0.15$  and  $\tau_c = 1.8$  ms, and the chimera state with  $\sigma = 0.13$  and  $\tau_c = 2.3$  ms. Numerical simulation of equations (2.9) and (2.10) gives similar time series.

where

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}. \quad (3.9)$$

The first row of  $\mathbf{Q}$  corresponds to the doublet cluster, and the other two rows correspond to the two singlet clusters.

The variational equations are determined by considering the time evolution of a small perturbation  $\delta \mathbf{u}$  to the synchronous state and are given by

$$\frac{d}{dt} \delta \mathbf{u}_i(t) = \mathbf{E} \delta \mathbf{u}_i(t) + \mathbf{F} \beta \sin(2x_{\mu_i}^{(s)}(t) + 2\phi_0) \delta x_i(t) \quad (3.10)$$

$$\delta x_i(t) = \mathbf{G} \left[ (1 - 3\sigma) \delta \mathbf{u}_i(t - \tau_f) + \sigma \sum_j A_{ij} \delta \mathbf{u}_j(t - \tau_c) \right], \quad (3.11)$$

where  $x_i(t)$  is the behavior of node  $i$  in the desired partially synchronous state and we have used the fact that the network contains four globally coupled nodes.

In order to decouple the variational equations corresponding to perturbations transverse to the synchronization manifold from those corresponding to perturbations along the synchronization manifold, we now transform to the IRR coordinate system. The transformation matrix for the (DSS) chimera state is

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix}, \quad (3.12)$$

where the three upper rows correspond to the synchronization manifold and the

bottom row corresponds to the direction transverse to the synchronization manifold. In order to determine the stability of the chimera state, we need to consider only perturbations along directions transverse to the synchronization manifold. If we define IRR basis vectors  $\delta \mathbf{v}_i(t) \equiv T_{ij} \delta \mathbf{u}_i(t)$ , then  $\delta \mathbf{v}_4(t)$  is the only IRR basis vector corresponding to perturbations transverse to the synchronization manifold. Thus in the following we consider only  $\delta \mathbf{v}_\perp(t) \equiv \delta \mathbf{v}_4(t)$ . Left-multiplying by  $\mathbf{T}$  to transform the variational equations to the IRR coordinate system, we obtain

$$\frac{d}{dt} \delta \mathbf{v}_\perp(t) = \mathbf{E} \delta \mathbf{v}_\perp(t) + \mathbf{F} \beta \sin(2x_d^{(s)}(t) + 2\phi_0) \delta x_\perp(t) \quad (3.13)$$

$$\delta x_\perp(t) = \mathbf{G}[(1 - 3\sigma) \delta \mathbf{v}_\perp(t - \tau_f) + \sigma \sum_j B_{ij} \Delta \mathbf{v}_j(t - \tau_c)] \quad (3.14)$$

where  $x_d^{(s)}(t)$  is the behavior of one node in the doublet cluster (computed from Eqs. 3.7 and 3.8) and

$$\mathbf{B} = \mathbf{T} \mathbf{A} \mathbf{T}^{-1} = \begin{bmatrix} 1 & \sqrt{2} & \sqrt{2} & 0 \\ \sqrt{2} & 0 & 1 & 0 \\ \sqrt{2} & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (3.15)$$

is the adjacency matrix transformed to the IRR coordinates. Explicitly performing the sum in equation (3.14), we obtain

$$\Delta x_\perp(t) = \mathbf{G}[(1 - 3\sigma) \Delta \mathbf{v}_\perp(t - \tau_f) - \sigma \Delta \mathbf{v}_\perp(t - \tau_c)]. \quad (3.16)$$

To determine the stability, we calculated the largest Lyapunov exponent (LLE) of equations (3.13) and (3.16), which indicates how infinitesimal perturbations trans-

verse to the synchronization manifold grow or decay in time. If the LLE is negative, perturbations decay exponentially to zero, indicating that the state is stable. For our calculations we used discrete-time versions of the equations presented above, which are more suitable for the experimental conditions. Calculations were averaged over a period of at least 500,000 times larger than the coupling delay. A similar procedure was followed to obtain the stability for the other patterns of synchrony. The results of the stability calculations are presented in Section 3.5.1.

## 3.5 Results

### 3.5.1 Stability calculation results

In Fig. 3.4, we compare the results of experiments and stability calculations in the parameter space of coupling strength ( $\sigma$ ) and coupling delay ( $\tau_c$ ) for all the patterns of synchrony that the system displays. Experiments were performed by selecting regularly spaced points in the parameter space. A minimum of 20 experimental trials from different random initial conditions were performed for each point in parameter space. In principle, one can experimentally observe any state in the parameter space that theoretically shows stable solutions; however, in practice it can be difficult to observe states with small basins of attraction. As discussed in Ref. [119], the size of the basin of attraction is of great practical interest, and in future work it may be interesting to investigate whether symmetries in the network topology can help to shed light on so-called basin stability. For all four patterns of synchrony, agreement between experimentally observed states and their calculated

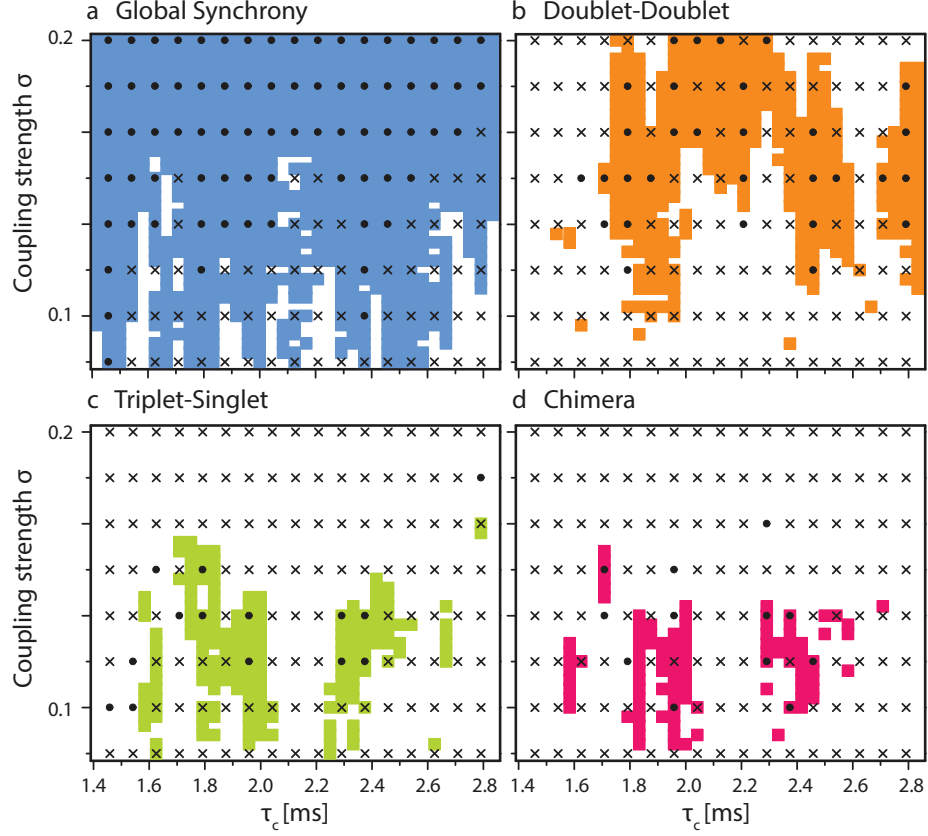


Figure 3.4: Comparison of experimental results and stability calculations. (a)-(d) Region of stability for different synchronous states. The shaded regions are stable; that is, the LLE of the variational equations is negative. The markers represent experimental results. Dots indicate that the state has been observed in experiments; crosses indicate that the state was not observed in experiments. At least 20 trials from different random initial conditions were performed for each experimental data point. For both experiments and simulations, the round-trip gain  $\beta = 3.8$ , the feedback time delay  $\tau_f = 1.4$  ms, and  $\phi_0 = -\pi/4$ .

stability is quite good. However, it is not surprising that there is slight disagreement near the boundary where the stability behavior is quite fragmented. This slight disagreement can be attributed to the finite number of random initial conditions that were used for the calculations and experiments, as well as slight mismatch between the actual experimental parameters and the parameter values used in simulation and stability calculations.

The procedure for stability calculations described in Section 3.4.3 can in principle be used to determine the stability of partially synchronous states (clusters and chimeras) in networks of any size. While our experiment is restricted to four nodes, we have performed the same type of stability analysis for a chimera state in a 10-node network consisting of one cluster of 5 and 5 singlet clusters and found that it agrees with direct simulations of equations (2.9) and (2.10), as shown in Fig. 3.5.

### 3.5.2 Chimera states and multistability

Recently, the existence of chimeras has been theoretically associated with multistability in the system [99]. Our observations support this idea.

In Fig. 3.6 we show a direct connection between multistability and chimeras in our network of four oscillators. From the stability calculations we identified the regions where at least two of the globally synchronized, doublet-doublet, and triplet-singlet states are stable. In Fig. 3.6a such regions are marked as multistable. Calculated stable chimera solutions coincide well with these multistable regions. In experiments, we also observe this multistability for the parameter values that exhibit



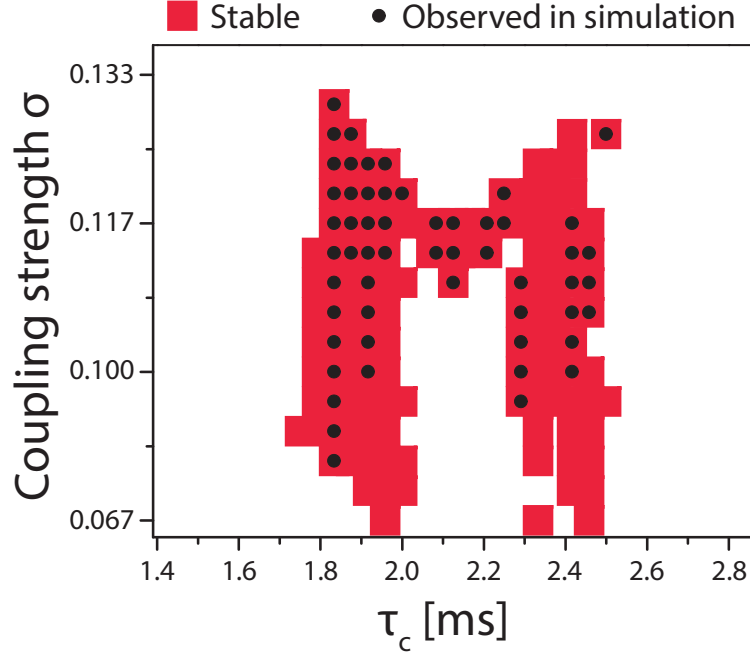


Figure 3.5: Comparison of stability calculations of a chimera state (one cluster of 5 and 5 singlet clusters) with direct simulation of equations (2.9) and (2.10) for a 10- node, globally-coupled network. Direct simulations consisted of 100 trials with different initial conditions for each point in  $\sigma - \tau_c$  parameter space. For all simulations, the round-trip gain  $\beta = 3.8$ , the feedback time delay  $\tau_f = 1.4$  ms and  $\phi_0 = -\pi/4$ .

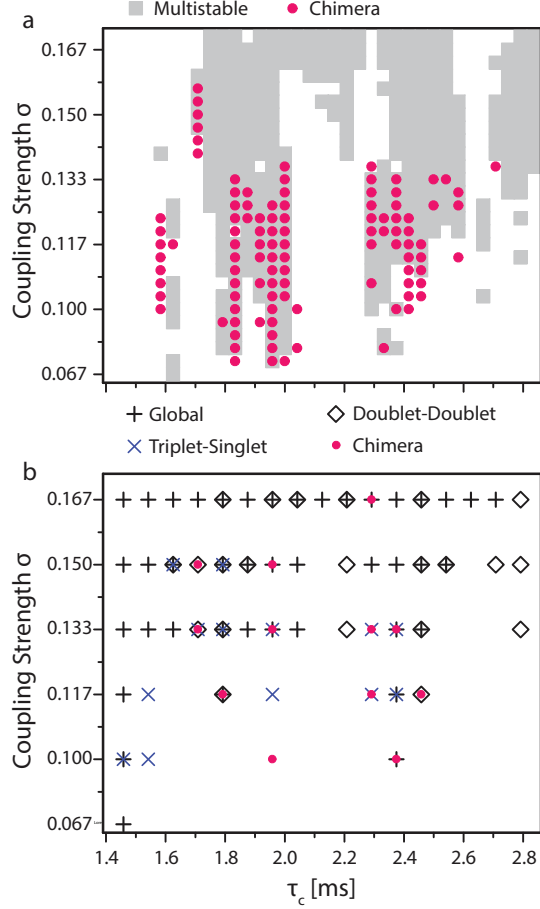


Figure 3.6: Relationship between chimeras and multistability in the four node, globally coupled network. (a) Stability calculations show that (DSS) chimeras tend to be stable in the regions of parameter space where other synchronization patterns coexist. Here multistable means that at least two of the globally synchronized, doublet-doublet, and triplet-singlet states are stable. (b) Experiments also show that chimeras tend to show up in regions of multistability. For both experiments and simulations, the round-trip gain  $\beta = 3.8$  and the feedback time delay  $\tau_f = 1.4$  ms.

chimera states, as shown in Fig. 3.6b.

In addition to the multistability between different partially synchronous patterns, we also observe multistability within a single pattern. For example, while the dynamics of the globally synchronized state in Fig. 3.3a appear chaotic, there are other globally synchronized states which appear to be quasiperiodic. We do not distinguish between different dynamical behaviors of the same partially synchronous state; for example, in Fig. 3.4a if any of the possible dynamical realizations of global synchrony is stable, we consider the globally synchronized pattern to be stable.

As discussed previously, partial synchronization patterns (doublet-doublet, triplet-singlet, and (DSS) chimera states in this case) emerge from the partial (subgroup) symmetries in the network [113, 114]. In systems like ours this can be predicted by a detailed inspection of all the subgroup symmetries of the network by analyzing the adjacency matrix. Still, what mechanism breaks the maximal symmetry is an interesting question. For our particular system, it is the presence of two different time delays in the system introduced by the mismatch between the coupling delay and the feedback delay. When these two time delays exactly match (the Laplacian coupling case), we observe only global synchrony. This observation is consistent with previous work on these opto-electronic oscillator networks [55].

We understand that the dependence of the resulting synchronous state on initial conditions leads to complicated fragmented regions of stability. As the initial conditions change, the high degree of multistability (both within a single synchronous pattern as well as between different synchronous patterns) of parameter

space allows different stable states (depending upon their basins of attraction) including global synchrony and complete desynchrony. Thus, rather than observing a smooth boundary between synchronized and desynchronized regions we observe fragmented regions of stability, as can be seen in Figs. 3.4 and 3.6.

Hence, the multistability or the possibility of various partially synchronous solutions in the system can be seen as a requirement for chimeras in any system, but the physical mechanism that generates such multistability can be different for different systems. It is well-known that time-delay in the coupling can induce multistability between synchronous states (e.g., the review in ref. [120]). This is the case in our system, while in the laser system described in ref. [99], amplitude-phase coupling induces the multistability necessary for the appearance of chimera states.

### 3.6 Discussion

Our results are fundamentally important in the context of chimeras. This is a small system without any breaking of symmetry in the network topology, yet we experimentally observe chimera states starting from random initial conditions. Our system violates most of the conditions previously believed to be necessary for the formation of chimera states: it is a small network, it is initialized from random initial conditions, and it is globally coupled. Importantly, our stability calculations show that the observed chimeras are not long transients but stable physical states that persist in experiments.

The mechanism that allows the patterns of synchrony to form in our system

is a general phenomenon called isolated desynchronization in which some clusters separate out from the synchronized state without destroying the synchrony completely [113]. This is possible due to the partial (subgroup) symmetries of the network. The subgroup structure guarantees that all the nodes in one cluster receive the same effective coupling signal from nodes in other clusters. Hence even if one cluster is desynchronized, the others can remain in identical synchrony. For example, in the case of our DSS chimera, each of the oscillators in the doublet cluster receives the same total signal from the two desynchronized singlets, allowing them to remain synchronized even though the two singlet oscillators behave incoherently. The idea that the chimeras and clusters in our system arise from the same mechanism of isolated desynchronization and that their stability can be calculated in the same manner highlights the close relationship between chimera and cluster states as partial synchronization patterns. We emphasize that the analysis we have presented here is not restricted to globally coupled networks of oscillators. The group theoretical analysis and mechanism of isolated desynchronization extend to any network with cluster states or synchronized chimera states, such as those found in the non-locally coupled systems in refs. [84, 85, 121] and the star network in ref. [88].

While in the simulations and stability analysis we consider identical oscillators with identical coupling, some heterogeneity and noise are inevitable in experiments. Despite the small heterogeneities in our experiment, we still observe persistent chimera and cluster states, in agreement with the simulations and stability calculations. Determining the amount of heterogeneity for which the group theoretical analysis and stability calculations remain valid is an important question

that has recently been studied [122].

We have observed all possible patterns of synchrony, including a chimera state, in our experimental network of four globally coupled chaotic opto-electronic oscillators. We used group theoretical methods recently developed for cluster synchrony to calculate the linear stability of these states and found excellent agreement with our experiments. These methods are quite general in that they extend to large networks and can be used to analyze the stability of any synchronized chimera state, suggesting that such chimeras and cluster states are closely related. Our results indicate that multistability of different synchronous patterns seems to be important for the existence of stable chimera states and can be determined by analyzing the symmetries of a given network topology; however, the mechanism that generates the multistability can be different in different systems. For our case we identify it to be the breaking of the symmetry present in Laplacian coupling by having two different time delays in the network.

We mentioned in Section 3.4.1 that these synchronized chimeras might be observable in globally coupled networks of any size, and we supported this by showing in Section 3.5.1 that we calculate them to be stable in a 10-node model of our opto-electronic network. However, our network of opto-electronic oscillators is limited to small networks of four nodes or fewer. In principle, we could build additional opto-electronic oscillators and couple them together, but this would be expensive and difficult to manage (the 4-node network is already complex as shown in Fig. 2.4) as the number of fiber feedback and coupling links grows as  $N^2$  where  $N$  is the number of nodes in the network. In order to study larger networks, we had to find a

more suitable experimental system. This system is described in Chapter 4, and the different network experiments we have performed using this apparatus (including chimeras in a globally-coupled network of five nodes) are reported in Chapter 5.

## Chapter 4: Using a single nonlinear node with delayed feedback to realize networks of coupled oscillators

We are not going in circles, we are  
going upwards. The path is a spiral; we  
have already climbed many steps.

---

*Siddhartha*

HERMANN HESSE

*This chapter is based on work from the following publications:*

*Joseph D Hart, Don C Schmadel, Thomas E Murphy, and Rajarshi Roy. “Experiments with arbitrary networks in time-multiplexed delay systems.” *Chaos*: 27(12), 121103 (2017).*

*Joseph D Hart, Laurent Larger, Thomas E Murphy, and Rajarshi Roy. “Delayed Dynamical Systems: Networks, Chimeras and Reservoir Computing.” *arXiv preprint arXiv:1808.04596* (2018).*

### 4.1 Overview

Our focus in this chapter is on the implementation of networks of truly identical coupled oscillators through the use of a single nonlinear delayed feedback system. These networks can consist of a large number of nodes and have arbitrary



topology. This is made possible through the same space-time representation that enabled the observation of other spatio-temporal phenomena in delay systems. Originally invented for the implementation of neural networks for reservoir computing in hardware [21, 22], this technique for implementing networks has subsequently been adapted for basic research, such as the study of chimera states in ring networks [19, 20] and cluster synchronization in arbitrary networks [123, 124]. This framework for implementing networks is particularly attractive because it allows for experiments on large networks without building a large number of separate physical oscillators and it allows for experiments on truly identical oscillators. We focus on opto-electronic implementations, which are popular due to their speed, cost, and ease of implementation; however, the techniques described are applicable to other delay systems as well.

Section 4.2 describes some of the research that made possible the idea of viewing a single delay system as a network of connected nodes. In Section 4.2.1 we recall the mathematical description from filter theory that employs a convolution integral of the feedback signal with the impulse response that describes the bandwidth limitations of the system. This description is less commonly used than, but equivalent, to the delay differential equation formalism used in Chapters 2 and 3. This second formalism, when viewed in the space-time representation, gives insight into how networks of oscillators can be realized with a single nonlinear system with delayed feedback. The space-time representation of delay systems is presented in Section 4.2.2. The space-time representation relies on the separation of time scales – fast dynamics and a long delay – to parameterize time as a time-like integer number that

counts the number of round-trip times and a continuous, space-like variable that denotes the position within each delay. This analogy between feedback systems with a long time delay and spatio-temporal systems has allowed for a deeper understanding of many complex phenomena observed in delay systems, including defect-mediated turbulence [42, 43], coarsening [44, 45], domain nucleation [46], spatial coherence resonance [47], phase transitions [48] and now, network dynamics.

Section 4.3 describes in detail how the space-time representation allows for the implementation of networks of truly identical coupled oscillators using only a single delayed feedback system. Traditional networks are spatially multiplexed: all nodes are updated simultaneously in parallel depending on their previous states. Delay feedback networks replace the spatial multiplexing of traditional networks with temporal multiplexing, in which the single nonlinear element serially updates the nodes, which are distributed across the delay line (interleaved in time). The nodes are coupled together by the “inertia,” or non-instantaneous response time, of the system, which can arise from the bandwidth limitations of the components. When this filtering is time-invariant, the resulting network has cyclic symmetry.

The use of delay networks for hardware implementations of reservoir computers is discussed in Section 4.3.1. Reservoir computing – alternatively echo state networks [125] or nonlinear transient computing [62] – is a type of neural network in which only the output connections are trained (the input and internal connections are fixed). Reservoir computers are particularly attractive because they can be trained by simple linear regression and because they are well-suited for implementation in specialized hardware. Delay networks have proven to be particularly well-suited for

reservoir computing.

Chimera states are an unexpected coexistence of spatial domains of coherence and incoherence in a system of identical oscillators with symmetric coupling [73, 74]. Chimera states were particularly difficult to observe in experiments because they typically (but not always [23, 99, 126]) occur in large networks, which are difficult to experimentally implement. Initially observed in 2012 [83, 84] a decade after their prediction, they were soon after observed in electronic [19] and opto-electronic [20] delay systems, as presented in Section 4.3.2.

A recently developed technique that allows a network with any topology to be implemented in a delay system is described in Section 4.4. This technique replaces the time-invariant filters used in the original delay network implementations with a time-dependent filter. The time-dependent filter, implemented digitally with a field-programmable gate array (FPGA), extends the range of networks that can be realized from only networks with rotational symmetry to networks with completely arbitrary topology, including networks with time-varying topologies and adaptive coupling schemes.

## 4.2 Background

### 4.2.1 Filter impulse response description of delay dynamical systems

In Section 2.2.2, we stated that a general model of delayed feedback systems is given by the convolution integral:

$$x(t) = h(t) * \beta F(x(t - \tau_D)) = \beta \int_{-\infty}^{\infty} h(t - t') F(x(t' - \tau_D)) dt' = \beta \int_{-\infty}^t h(t - t') F(x(t' - \tau_D)) dt' \quad (4.1)$$

where in the last step we use the property that  $h(t)$  is causal. In Eq. 4.1,  $x(t)$  is the filter output,  $\beta$  is the round trip gain, and  $\tau_D$  is the time delay. While this formulation of the model is mathematically equivalent to the more commonly used integro-differential equation given by Eq. 2.2, the convolution integral provides insight that is essential to the interpretation of a single nonlinear delayed feedback system as a network of truly identical coupled oscillators.

### 4.2.2 Space-time representation of delay systems

The space-time representation of delay systems was originally motivated by the numerical treatment of delay differential equations [32]. The time variable is split up into a continuous variable  $\sigma$  bounded between 0 and  $\tau_D$ , and an independent discrete variable  $n$  that counts the number of delays since the origin. Ikeda and Matsumoto [127] were the first to consider  $\sigma$  to be a “spatial” variable in their modeling of optical turbulence. The space-time representation was formalized and first used on experimental data by Arecchi et al. in 1992 [18] in order to study long-

time correlations on the order of one delay in a CO<sub>2</sub> laser with delayed feedback. Since then, the relationship between delay systems and spatio-temporal systems has been investigated thoroughly [19, 20, 42, 48], and in many cases, equivalence has been rigorously established [43, 128–130]. For a recent review, see Ref. [41].

The space-time representation of delay systems is particularly meaningful when the delay  $\tau_D$  is long compared to the time scale  $t_c$  of the temporal dynamics of the system, as measured by the width of the zeroth peak in the autocorrelation [41]. In this case, there is a separation of time scales, and so it is natural to parameterize time as

$$t = n\tau_D + \sigma, \quad (4.2)$$

where  $n$  is an integer that counts the number of delay times since the origin, and  $\sigma$  is a continuous variable between 0 and  $\tau_D$  that gives the position along the delay. As a result,  $n$  is often considered to be a discrete time and  $\sigma$  a continuous pseudo-spatial variable. We note that  $t_c$  is a property of the dynamics and therefore depends on  $\beta$  and  $F(x)$  in addition to the time scales  $\tau_L$  and  $\tau_H$  in Eq. 2.2; in practice, however, it is often the case that  $t_c \approx \tau_L$  [41].

When working with delay systems, one often obtains a long time series  $x(t)$  such as the one shown in Fig. 4.1(a). It seems that there are (and indeed one expects there to be) correlations on the order of one time delay  $\tau_D$ . Plotting the time series in the space-time representation in Fig. 4.1(b) shows long time correlations (on the order of several  $\tau_D$ ) as spatial structures that evolve in discrete time.

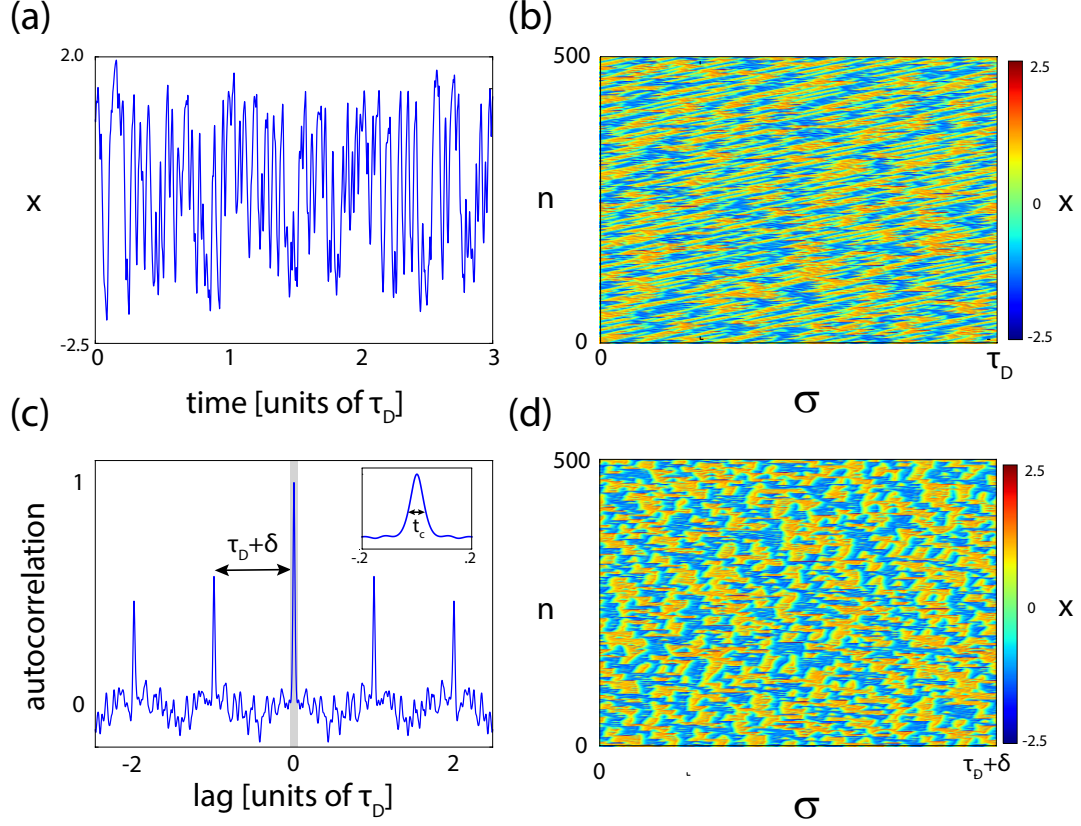


Figure 4.1: Illustration of the space-time representation. (a) Time series of the delayed system in Eq. 2.2. (b) Space-time representation of the time series shown in (a), where  $\sigma \in [0, \tau_D]$ . (c) Autocorrelation of the time series shown in (a). The distance to the first autocorrelation peak is  $\tau_D + \delta$ . Here  $\tau_D = 4$  ms and  $\delta = 250 \mu\text{s}$ . *Inset.* Zoom in on central autocorrelation peak. The width of this peak is  $t_c$ . (d) Space-time representation with drift correction ( $\sigma \in [0, \tau_D + \delta]$ ). These figures were made from a numerical simulation of Eq. 2.2 with  $\beta = -5$ ,  $\tau_L = 400 \mu\text{s}$ ,  $\tau_H = 10$  ms,  $\tau_D = 4$  ms, and  $F(x) = \sin^2(x(t) - \pi/4)$ , which describes the opto-electronic oscillator shown in Fig. 2.1b.

While Fig. 4.1(b) does reveal long-time correlations as spatio-temporal structures, it is clear that as  $n$  increases the structures are drifting to the right in  $\sigma$ -space. In other words, the long-time correlations occur over a time slightly larger than  $\tau_D$ . This can be seen by looking at the autocorrelation of the time series, shown in Fig. 4.1(c). The autocorrelation begins to increase near a lag of  $\tau_D$ , but only reaches its peak at  $\tau_D + \delta$  due to the non-instantaneous response time of the system [41]. Therefore  $\delta$  is related to the widths of the zeroth autocorrelation peak  $t_c$  as well as the width of the first autocorrelation peak. Previous works have extensively studied this drift and its relation to co-moving Lyapunov exponents [44, 128].

The drift is a reflection of the fact that the system is causal. The delayed term  $x(t - \tau_D)$  cannot affect the dynamics before, *or even at*, the time  $t$ . Therefore, in Fig. 4.1(d), we use

$$t = nT + \sigma \tag{4.3}$$

to create space-time representations, where  $T = \tau_D + \delta$  is the recurrence time and now  $\sigma \in [0, T]$ . When the space-time representation is done in this way, the structures are stabilized in space (i.e., they have a nearly stationary average spatial position). Indeed, it has been shown that this is often the correct moving frame in which to view the spatio-temporal behavior of time-delayed systems [41].

### 4.3 A single delay system as a ring network

Recently, the space-time representation has been used to interpret a single nonlinear node with delayed feedback as a network of coupled oscillators. These experiments replace the spatial multiplexing of a traditional network (in which all nodes are updated simultaneously in parallel) with time multiplexing, in which the single nonlinear element serially updates each of the nodes, which are distributed across the delay line. There are two major benefits to this network implementation: this is the *only* way to create a network of truly identical nodes, and it allows one to implement a large network without building a large number of separate physical nodes. While originally used for a hardware implementation of reservoir computing [21, 22, 62, 131–133], these types of delay systems have since been used to study chimera states in cyclic networks [19, 20] and cluster synchronization in arbitrary networks [123, 124].

Because delay systems require a continuous function to describe their initial conditions, they are considered infinite dimensional systems. However, it was noticed early on that chaotic attractors of delay systems have finite dimension in practice [32]. In trying to explain this finite dimensionality, Le Berre et al. conjectured that the dimension of the attractor is equal to  $\tau_D/t_c$ , where  $t_c$  is the width of the zeroth peak of the autocorrelation of the chaotic time series [33]. In other words, in practice, only  $\tau_D/t_c$  values are needed to specify a point on the attractor [134]. Even more, it was suggested that a delay can be thought of as a set of  $\tau_D/t_c$  roughly independent time slots, such that the  $k^{th}$  time slot in one delay is correlated with



only the  $k^{th}$  time slot in the following delay, as confirmed by the secondary peaks in the autocorrelation function (e.g. Fig. 4.1c). If each of these independent time slots is considered to be a “node,” one can think of the delay system as consisting of a set of  $\tau_D/t_c$  independent, discrete time nonlinear systems. Clearly, this reasoning is similar to the reasoning that led to the development of the space-time representation and is particularly useful in the same types of situations, i.e., when  $\tau_D \gg t_c$ .

### Networks in a discrete-time delay system

Temporal discretization arises naturally in many experimental implementations of delay systems. The electro-optic feedback system with a pulsed laser described in Section 2.2.2 is one such example [65,67]. Further, many experimental delay systems implement the delay line with a digital shift register because of the ability to easily vary the delay [20,21,23,56,62,64,123]. In these implementations, the digital shift register discretizes time into steps of size  $\Delta t = \tau_D/N$ , where  $N$  is an integer. These digital shift registers apply a constant feedback for one time step  $\Delta t$ , then sample the system at the end of the time step. Because of the discretization, the use of the co-moving frame  $T = \tau_D + \delta$  discussed in Section 4.2.2 is not always necessary, and we can simply use a discretized version of original space-time representation Eq. 4.2.

In order to reveal the link between these systems and networks, we explicitly discretize time into time steps of length  $\Delta t$ , and we call each time slot a network node. If  $\Delta t$  is chosen to be slightly less than  $t_c$ , the nodes (which span an interval  $\Delta t$ ) are no longer roughly independent, but are now coupled through the “inertia”

due to the non-instantaneous response time of the system to which delayed feedback is applied. This non-instantaneous response time can be described by a filter impulse response. In this way we have a network of coupled nodes, where the strength and topology of the coupling are determined by the shape of the filter impulse response. The temporal discretization  $\Delta t$  is chosen depending on the application, and can have an important impact on the dynamics and coupling, as we discuss at the end of this section.

In order to show explicitly how the network structure arises in these cases, we consider the discretized space-time representation

$$k = nN + i, \quad (4.4)$$

where  $k$  is the original discrete time,  $n$  is an integer that counts the number of delays that have passed,  $N = \tau_D/\Delta t$  is the number of time steps in a delay, and  $i$  is the discrete spatial variable. In our network interpretation,  $n$  will be the network time and  $i$  will be the node index. We impose this discrete space-time representation (Eq. 4.4) upon the discrete time delayed Eq. 2.3:

$$x_i[n] = \beta \sum_{m=-\infty}^{nN+i} h[nN + i - m] F(x[m - N]), \quad (4.5)$$

where  $N = \tau_D/\Delta t$  is the number of nodes in the network,  $n$  is the network time, and  $i$  is the node index. We can then split up this summation as follows:

$$x_i[n] = S_i[n] + C_i[n] \quad (4.6)$$

$$S_i[n] = \beta \sum_{m=-\infty}^{(n-1)N+i} h[nN+i-m]F(x[m-N]) \quad (4.7)$$

$$C_i[n] = \beta \sum_{m=(n-1)N+i+1}^{nN+i} h[nN+i-m]F(x[m-N]). \quad (4.8)$$

Further insight into the meaning of  $S_i[n]$  can be provided by a concrete example. For concreteness we consider a single-pole low pass filter described by  $h(t) = \tau_L^{-1}e^{-t/\tau_L}u(t)$ , where  $u(t)$  is the Heaviside step function, as depicted in Fig. 4.2a. In this case Eq. 4.7 becomes

$$S_i[n] = \beta e^{-\tau_D/\tau_L} x_i[n-1]. \quad (4.9)$$

Eq. 4.9 shows that  $S_i[n]$  is a self-feedback term with a weight  $w_h$  that depends on the form of  $h(t)$ . In general when the delay is long relative to the filter time scales,  $w_h \rightarrow 0$ , as is clear from Eq. 4.9 for the particular case of a low pass filter where  $w_h = e^{-\tau_D/\tau_L}$ .

In order to interpret  $C_i[n]$ , we perform a simple change of variables  $p = m - nN$  in Eq. 4.8 to obtain

$$C_i[n] = \beta \sum_{p=i+1-N}^i h[i-p]F(x_p[n-1]). \quad (4.10)$$

Therefore  $C_i[n]$  is a coupling term: the summation “couples” the values of  $x_p[n-1]$  (weighted by  $h$ ) to the value of  $x_i[n-1]$  to determine  $x_i[n]$ .

Equation 4.6 along with Eqs. 4.9 and 4.10 now resembles a network equation:

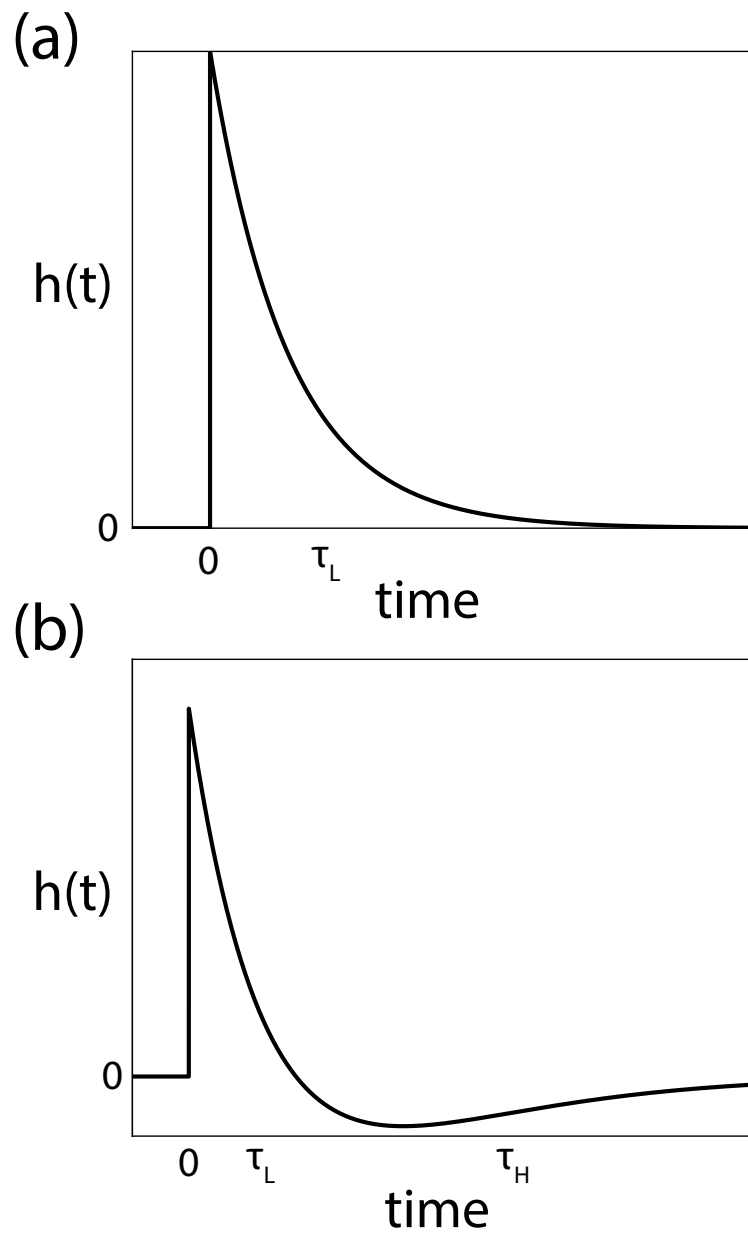


Figure 4.2: Impulse response for (a) single-pole low pass filter and (b) two-pole band pass filter. The poles are real in both cases.

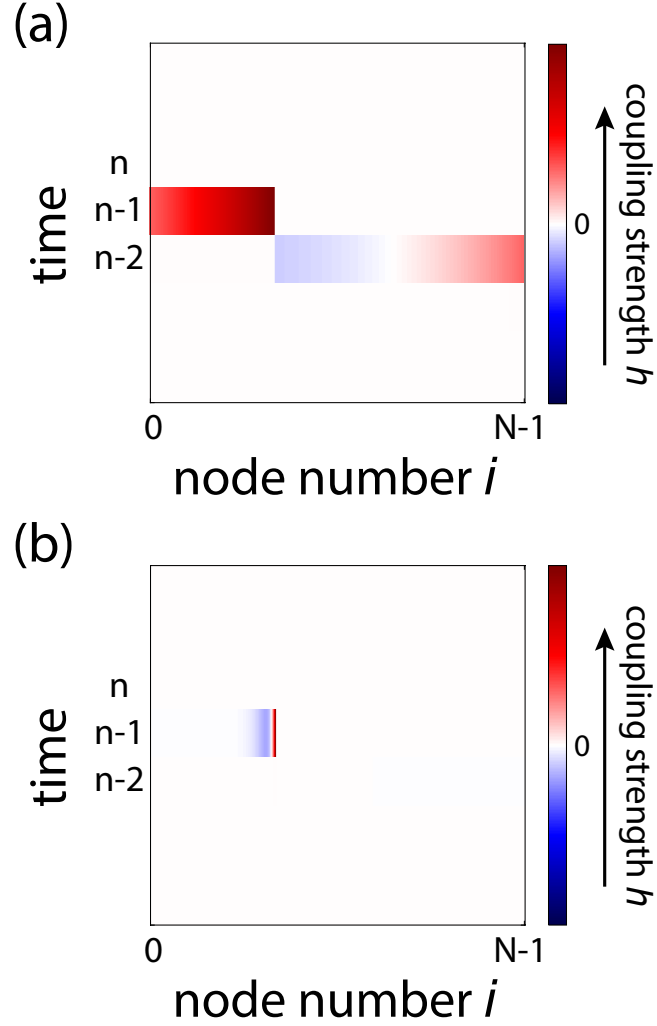


Figure 4.3: Illustration of the coupling term in the space-time representation of delay systems (second term in Eq. 4.6) (a) when  $\tau_D \approx \tau_L$  and (b) when  $\tau_D \gg \tau_L$  when the coupling is implemented by a band pass filter. The coloring indicates the strength of the coupling  $h[k]$  from the shaded nodes ( $x_p[n-1]$ ) to the node represented by the black rectangle ( $x_i[n]$ ). Red shading represents positive coupling, blue negative coupling, and white no coupling. In (a), the coupling spans two full time steps ( $n-1$  and  $n-2$ ), and so this should not be considered a network. In (b), however, the coupling is significant over only a small range (from  $p - k_\Delta$  to  $p$ ) and so for almost all nodes  $i$  the coupling comes from nodes only at time step  $n-1$ . Therefore, this can be considered to be a network.

each node  $i$  is coupled to all the other nodes through the coupling weights  $h$ . However, this should not yet be considered a network. We recall that the superscript on  $x$  denotes a node index and must be in the range  $[0, N-1]$ ; however, in Eq. 4.10  $p$  runs from  $i + 1 - N$  to  $i$ , which can include negative values. Physically, this means that the coupling summation runs over some  $x$  values at time  $n - 2$  in addition to those from time  $n - 1$ . This is illustrated in Fig. 4.3a, where the black rectangle denotes  $x_i[n]$  and the shaded region denotes the  $x$  values that are coupled to  $x_i[n - 1]$  by  $C_i[n]$  to determine  $x_i[n]$ .

In cases where the delay  $\tau_D = N\Delta t$  is long (relative to the filter time scales), the filter impulse response is significant for only a small range, from  $i - k_\Delta$  to  $i$ , where  $k_\Delta \ll N$  is a small number of time steps (determined by the form of  $h[k]$ ) above which  $h[k_\Delta]$  is negligible. For long delays, we can approximate Eq. 4.6 as

$$x_i[n] = w_h x_i[n - 1] + \beta \sum_{p=i-k_\Delta}^i h[i - p] F(x_p[n - 1]), \quad (4.11)$$

where the superscript denotes the node number and the number in square brackets denotes the discrete network time.

Equation 4.11 is now an exact correspondence with the standard network equation

$$x_i[n] = G(x_i[n - 1]) + \sum_{j=1}^N A_{ij} F(x_j[n - 1]), \quad (4.12)$$

where  $G(x)$  is a function that describes the self-feedback and  $A_{ij}$  is the weighted network adjacency matrix. By comparing Eqs. 4.11 and 4.12,  $G(x) = w_h x$ . The filter impulse response  $h(t)$  is the equivalent of the adjacency matrix; it determines the

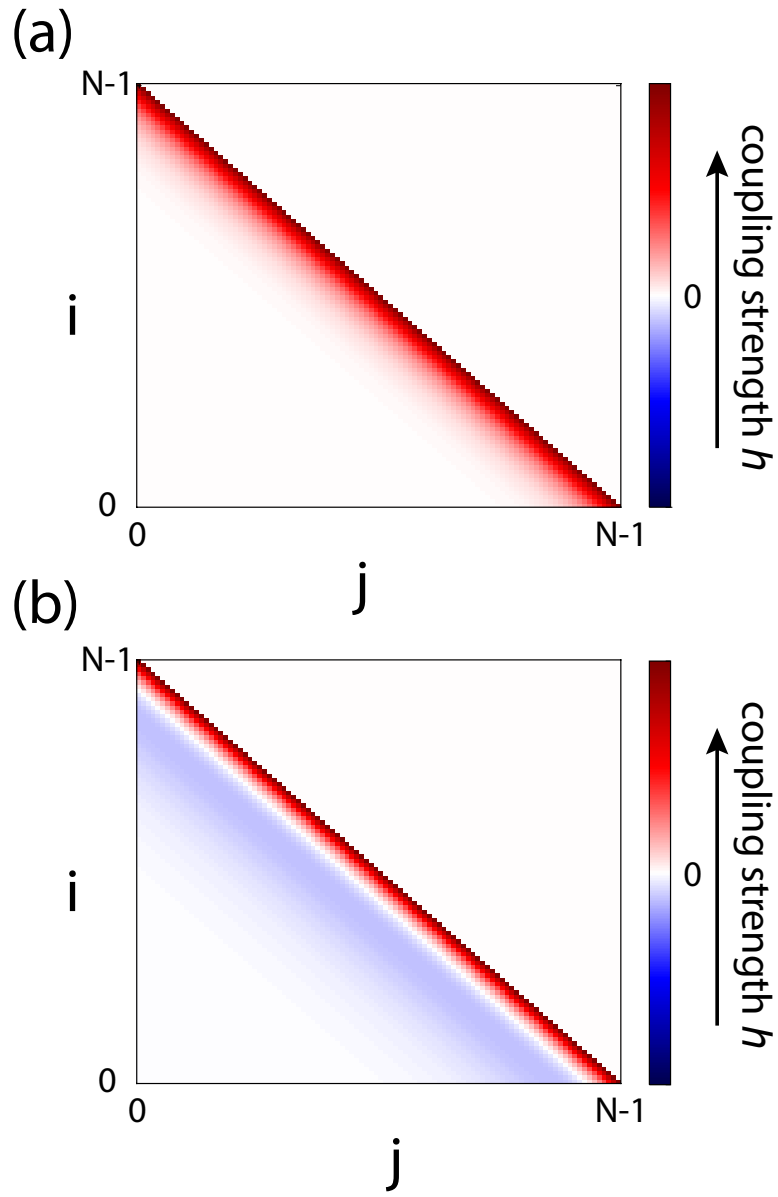


Figure 4.4: Illustration of the adjacency matrices for (a) low pass filter and (b) band pass filter. The adjacency matrix is cyclically symmetric due to the time invariance of the filter.

strength and topology of the coupling.

For concreteness in demonstration, we now present the adjacency matrices induced by two simple but common impulse responses: the low pass filter and the band pass filter. The single pole low pass filter response is given by [66]

$$h_{LP}(t) = \tau_L^{-1} e^{-t/\tau_L} u(t), \quad (4.13)$$

where  $\tau_L$  is the filter time constant, and  $u(t)$  is the Heaviside step function. This is the impulse response that one would use, for example, when solving the Ikeda equation, Eq. 2.1. The adjacency matrix that corresponds with this low pass filter is given by

$$A_{ij}^{LP} = \beta \frac{\Delta t}{\tau_L} \begin{cases} e^{-(i-j)\Delta t/\tau_L} & \text{if } 0 \leq i - j \leq k_R \\ 0 & \text{otherwise} \end{cases}. \quad (4.14)$$

A depiction of this adjacency matrix is shown in Fig. 4.4a. We note that all couplings are positive and that the network is a directed ring. Another common type of filtering is the two-pole band pass filter, which has impulse response [66]

$$h_{BP}(t) = \frac{\frac{1}{\tau_L} e^{-t/\tau_L} - \frac{1}{\tau_H} e^{-t/\tau_H}}{1 - \tau_L/\tau_H} u(t), \quad (4.15)$$

where  $\tau_H$  is the high pass filter time constant and  $\tau_L$  is again the low pass filter time constant, depicted in Fig. 4.2a. This impulse response corresponds to the filtering in Eq. 2.2. The corresponding adjacency matrix is



$$A_{ij}^{BP} = \beta \frac{\Delta t}{1 - \tau_L/\tau_H} \begin{cases} \tau_L^{-1} e^{-(i-j)\Delta t/\tau_L} - \tau_H^{-1} e^{-(i-j)\Delta t/\tau_H} & \text{if } 0 \leq i - j \leq k_R \\ 0 & \text{otherwise} \end{cases}. \quad (4.16)$$

A depiction of this adjacency matrix is shown in Fig. 4.4b. We note that the network is again a directed ring; however some of the couplings are now negative. Time-invariant filters, such as the two discussed above, will lead to ring networks, and the ring is directed due to causality. However, networks with arbitrary topologies can be created by the introduction of a time dependent filter, as we discuss in Section 4.4.

Here we make a note about the design of these network experiments and the choice of  $\Delta t$  relative to the time scales  $\tau_L$  and  $\tau_D$ . If  $\Delta t < \tau_L$ , the (time invariant) filter impulse response will couple the nodes in a cyclically symmetric adjacency matrix, with the coupling radius and coupling strength determined by the form of the impulse response. If  $\Delta t \gg \tau_L$ , no coupling will be induced by the filtering, and the system will consist of completely independent but identical nodes. Further, the quantity  $\tau_D/\tau_L$  should be large for the network interpretation to hold in general. If  $\tau_D/\tau_L$  is not large, then for a significant fraction of nodes the  $C_i[n]$  includes terms from both time  $n - 1$  and time  $n - 2$  as shown in Fig. 4.3a. The fraction of nodes for which this is the case tends to zero as  $\tau_D/\tau_L \rightarrow \infty$ .

### Extension to continuous-time delay systems

Networks can also be realized using the space-time representation in the case of fully analog delay lines, such as those that rely on the finite propagation speed of light.

Such a system can also be well-approximated by the discrete time systems discussed in Section 4.3 by taking the limit that  $\Delta t/\tau_D \rightarrow 0$  [19, 20]. In these situations, time is continuous, so we return to the space-time representation given by Eq. 4.3. This allows us to think of a continuum of nodes that are labeled by their position  $\sigma$  and evolve in discrete time  $n$ .

The realization of a network follows very much along the lines of Section 4.3, but in continuous time rather than discrete. Therefore, the summations will be replaced by integrals, and we will have to account for the drift  $\delta$  in the space-time representation. What follows is an elaboration of the presentation contained in Ref. [20]. First, we note that the solution to Eq. 2.2 is given by the convolution of the filter input  $v(t) = F(x(t - \tau_D))$  with the filter impulse response  $h(t)$ :

$$x(t) = h(t) * F(x(t - \tau_D)) = \int_{-\infty}^t h(t - t') F(x(t' - \tau_D)) dt'. \quad (4.17)$$

For systems with linear filters such as the low pass filter described by Eq. 2.1 or the band pass filter described by Eq. 2.2, the form of  $h(t)$  can be found analytically.

We now analyze this solution from the perspective of the space-time representation by setting  $t = nT + \sigma$  where  $n$  is an integer that counts the number of drift-corrected delays  $T = \tau_D + \delta$  that have passed since the origin, and  $\sigma \in [0, T]$  is the node's position in pseudo-space. Re-writing Eq. 4.1 with this change of variables results in

$$x_\sigma[n] = \int_{-\infty}^{nT+\sigma} h(nT + \sigma - t') F(x(t' - \tau_D)) dt'. \quad (4.18)$$

We can then separate the integral into two domains as follows:

$$x_\sigma[n] = S_\sigma[n] + C_\sigma[n] \quad (4.19)$$

$$S_\sigma[n] = \int_{-\infty}^{(n-1)T+\sigma} h(nT + \sigma - t') F\left(x(t' - \tau_D)\right) dt' \quad (4.20)$$

$$C_\sigma[n] = \int_{(n-1)T+\sigma}^{nT+\sigma} h(nT + \sigma - t') F\left(x(t' - \tau_D)\right) dt'. \quad (4.21)$$

Further insight into the meaning of  $S_\sigma[n]$  can be provided by a concrete example, so that we can evaluate the integral. Here we consider the simplest filter, a single pole low pass filter described by  $h(t) = \tau_L^{-1} e^{-t/\tau_L} u(t)$  (Eq. 4.13). In this case Eq. 4.20 becomes

$$S_\sigma[n] = e^{-T/\tau_L} x_\sigma[n-1]. \quad (4.22)$$

The meaning of  $S_\sigma[n]$  is now clear: it is a self-feedback term (from the state  $x$  at the spatial position  $\sigma$  at discrete time  $n-1$  to the state at the spatial position  $\sigma$  at discrete time  $n$ ) with a strength determined by the form of  $h(t)$ .

In order to interpret  $C_\sigma[n]$ , we make a change of variables  $t'' = t' + \delta - nT$ :

$$C_\sigma[n] = \int_{\sigma-\tau_D}^{\sigma+\delta} h(\sigma + \delta - t'') F\left(x_{t''}[n-1]\right) dt''. \quad (4.23)$$

Therefore  $C_\sigma[n]$  is a coupling term: the integral “couples” the values of  $x_{t''}[n-1]$  to the value of  $x_\sigma[n-1]$  to determine  $x_\sigma[n]$ .

When the delay  $\tau_D$  is long (relative to the filter time scale), the filter impulse response is significant for only a small range, from  $\sigma - \Delta$  to  $\sigma + \delta$ , where  $\Delta \ll \tau_D$  is a short time (determined by the form of  $h(t)$ ) above which  $h(t)$  is negligible. For long delays, we can approximate Eq. 4.23 as

$$C_\sigma[n] \approx \int_{\sigma-\Delta}^{\sigma+\delta} h(\sigma + \delta - t'') F(x_{t''}[n-1]) dt''. \quad (4.24)$$

Eqs. 4.23 and 4.24 reveal the network structure that results from viewing the system with long delay through the space-time representation. The system can be interpreted as a continuum of discrete-time nodes whose position (node index) is given by  $\sigma$ . Each node is coupled to its neighbors within a distance  $\Delta$  on the left and  $\delta$  on the right through the system's impulse response  $h(t)$ , as shown in Fig. 4.3. Importantly, the coupling term in Eq. 4.24 includes only nodes from time step  $n-1$  for almost all nodes  $\sigma$  since  $\Delta \ll \tau_D$ . Indeed, in the limit  $\tau_L/\tau_D \rightarrow 0$ , the fraction of nodes whose input coupling spans two time steps vanishes. It is clear from Eq. 4.24 that  $h(t)$  determines both the coupling strength and the coupling width. The particular form of  $h(t)$  plays a crucial role in the types of dynamics that the system can exhibit.

### 4.3.1 Applications to reservoir computing

**Acknowledgment:** Laurent Larger performed the experiments and contributed the figures and much of the text for Section 4.3.1.

Reservoir computing is a recently proposed brain-inspired processing technique, corresponding to a simplified version of conventional recurrent neural network (RNN) concepts. It was independently proposed in the machine learning community under the naming Echo State Network (ESN) [135] and in the brain cognitive research community as Liquid State Machine [136]. It was later unified with the

now adopted name, Reservoir Computing (RC) [137, 138]. The generic architecture of a RC system is thus rather conventional (see Fig. 4.5), consisting of:

- An input layer aimed at expanding the input information to be RC-processed onto each node of the RNN;
- An internal network having a recurrent connectivity thus potentially possessing a complex internal dynamics depending on the spectral radius of its connectivity matrix;
- And an output layer intended to extract the computed result from the global observation of the network response, typically performing a linear combination of the different internal state variables of the network.

The most important difference of RC compared to conventional RNN consists in the restriction of the learning process (i.e. finding the optimal synaptic weights for the nodes and layer connectivity) to the output layer only. The input layer and the internal network connecting weights are usually set at random and remain fixed during training and operation. This is a linear regression problem, making the learning phase of RC computationally efficient compared to traditional multi-layer neural networks, which typically require gradient descent algorithms for training [135]. In many situations, the effective computational power of RC has been found comparable, or in some cases even better than, their standard RNN counterpart.

One major technological challenge of neuromorphic computing is designing a physical hardware that implements its specific concepts, instead of translating them

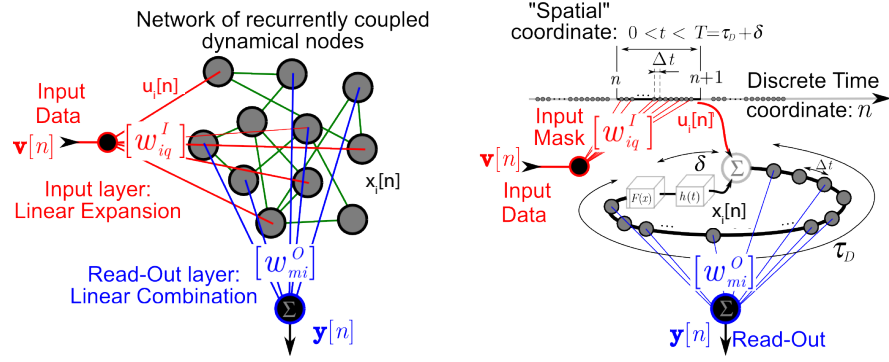


Figure 4.5: Graphical comparison between two reservoir computing implementations: A classical RNN architecture (left), and a delay dynamics based reservoir.

into algorithms to be programmed in standard, however structurally unmatched, digital processors. The generally recognized poor energy efficiency of artificial intelligence (AI) is related to the fact that neuromorphic computing concepts have to be adapted into Turing von Neumann machines, whose architecture and principles of operation are very far from what we have learned from the brain. Up to now, there is essentially no other easily available and dedicated computing platform capable of efficiently running artificial intelligence techniques. Turing von Neumann machines are practically the only effectively working solution today for investigating AI.

An essential problem when one wants to design a dedicated hardware implementation of neural network processing concepts is the difficulty to physically fabricate a well controlled three dimensional dynamical network, as nature easily does with any brain. Based on the assumption that what matter are the dynamical complexity and the high phase space dimension, but not the internal structure itself of the reservoir network, the EU project PHOCUS (PHOtonic liquid state machine based on delay CoUpIed Systems) started in 2010 with the objective to demonstrate the RC implementation suitability of nonlinear delay dynamics. Delay dynamics have thus been proposed as a way to replace a neural network architecture in the implementation of the RC concepts, with a first successful demonstration through an electronic delay system mimicking the Mackey-Glass dynamics [21]. To do so, extensive use of the space-time analogy of delay dynamics was made in order to properly adapt the RC processing rules previously used in networks of dynamical nodes (and effectively always programmed or simulated with digital processors).

Figure 4.5 shows on the left a standard network-based RC processing (ESN),

whereas the right figure displays its analogue based on nonlinear delayed feedback dynamics for the reservoir. The experimental setup first proposed for photonic RC is precisely the one depicted in Fig.2.1b, in which an external signal is superimposed at the rf input port of the Mach-Zehnder.

### Input layer

The input information in standard RNN is expanded into the network according to spatial multiplexing: The coordinates of the original input vector  $\mathbf{v}[n] \in \mathbb{R}^Q$  is expanded through the multiplication with the input connectivity matrix  $W^I \in \mathbb{R}^N \times \mathbb{R}^Q$ . Each node  $i = 0 \dots N - 1$  of the network is thus receiving an input signal  $u_i[n]$ :

$$u_i[n] = \sum_{q=1}^Q w_{iq}^I v_q[n] \quad (4.25)$$

When one is making use of a delay dynamics instead of network of nodes, time division multiplexing is naturally adopted to address the virtual nodes  $i$  distributed in time all along the recurrence time  $T$ . The required temporal waveform which will need to be injected into the delay dynamics, reads as follows:

$$u(t) = \sum_{i=0}^{N-1} \left[ \sum_{q=1}^Q w_{iq}^I v_q[n] \right] p_{\Delta t}(t - nT - i\Delta t), \quad (4.26)$$

where  $p_{\Delta t}(t)$  is a staircase function with  $p_{\Delta t}(t) = 1$  from time  $t = 0$  to  $t = \Delta t$  and zero otherwise. The duration  $\Delta t$  is the sampling period, or differently speaking, also the temporal spacing between two virtual nodes in the recurrence time interval  $T$ .



The scalar signal  $u(t)$  is practically programmed in an arbitrary waveform generator, it has the shape of a piecewise constant signal for each sample  $i = 0 \dots N - 1$  of each time slot of duration  $\Delta t$ . When dividing  $u(t)$  into sequences of  $N$  samples, and stacking horizontally these vectors of length  $N$  for each consecutive discrete time  $n$ , one obtains the space-time representation of the input signal, as depicted in Fig. 4.1c.

### Reservoir layer

A transient dynamic is then triggered in the reservoir due to the injection of the information signal  $u^{(i)}[n]$  or  $u(t)$ . For the ESN, this transient is ruled by the following discrete time update rule, from time  $(n - 1)$  to time  $n$ :

$$x_i[n] = F \left[ \sum_{j=1}^N w_{ij}^R x_j[n-1] + \rho \cdot u_i[n] \right], \quad (4.27)$$

where  $W^R \in \mathbb{R}^N \times \mathbb{R}^N$  is the internal connectivity matrix of the reservoir.  $F[\cdot]$  is a nonlinear function (usually a sigmod, e.g. a hyperbolic tangent, in classical ESN), and  $\rho$  is a scaling factor weighting the input signal defined in Eq. 4.25.

In the case of a delay reservoir, the update rule is similar to Eq. 4.1, except the delay dynamics is now non-autonomous. The input waveform defined in Eq. 4.26 is superposed on to the delayed feedback. It is thus contributing directly to a nonlinear transient in the delay dynamics phase space, with a contributing weight  $\rho$ :

$$x(t) = h(t) * F(x(t - \tau_D) + \rho \cdot u(t)) = \int_{-\infty}^t h(t - t') F[x(t' - \tau_D) + \rho \cdot u(t)] dt'. \quad (4.28)$$

The delay reservoir, unlike the discrete time ESN, is continuous in time. The definition of virtual spatial nodes, and their discretization, is experimentally introduced

through the sampling period  $\Delta t$  from Eq. 4.26. The adjacency matrices represented in Figs. 4.4a and 4.4b then correspond to the internal connectivity matrix  $W^R$  used for the ESN.

The time scale  $\Delta t$  is very important, as it has to be properly tuned with respect to the internal short time  $\tau_L$  of the delay dynamics. Optimal processing efficiency of the delay reservoir is empirically found for  $\Delta t \simeq \tau_L/5$  [21]. This highlights a compromise between:

- $\Delta t$  should not be too short, otherwise adjacent nodes will display nearly identical dynamics because they are too strongly coupled through the delay dynamics inertia (the reservoir response to the input data would also be too small in amplitude, since it would be strongly filtered; this has detrimental signal-to-noise ratio impacts in the RC processing);
- the adjacent nodes could be too decoupled when  $\Delta t$  is too large; If they would be too far one from each other, they would allow each stepwise transition of the input information to reach an asymptotic state independently of the farther past.

### Output layer

The last processing operation in RC concerns the read-out layer, consisting of a linear combination of the reservoir internal states  $x_i[n]$ . This step aims to provide the expected computational result. The read-out operation generates an output vector  $\mathbf{y}[n] \in \mathbb{R}^M$ , which is a linear superposition of the internal states:

$$y_m[n] = \sum_{i=0}^{N-1} w_{mi}^O x_i[n]. \quad (4.29)$$

The same equation holds in the case of a delay reservoir, where however the node state  $x_i[n]$  corresponds to the extraction of a virtual node state in the delay reservoir, through the sampling of  $x(t)$ . The signal defined by Eq. 4.28, is sampled to provide  $x(t_k)$ , with  $t_k = k \cdot \Delta t$ ,  $k$  being defined as in Eq. 4.4.

The coefficients of the linear combination (i.e. the elements  $w_{mi}^O$  of the read-out matrix  $W^O \in \mathbb{R}^M \times \mathbb{R}^N$ ) are determined by a learning task. In the case of supervised learning, one simply applies a ridge regression to an ill-posed problem for a set of known data couples,  $\{(reservoir\ response_l = A_l, target\ read-out_l = \tilde{B}_l), l = 1 \dots L\}$ . This corresponds to a training set of  $L$  couples of temporal data (evolution of the discrete time  $n$ ), each having a duration  $N_l$ .  $A_l \in \mathbb{R}^N \times \mathbb{R}^{N_l}$  is thus the concatenation of the reservoir state vector  $\{x_i[n] | i = 0 \dots N-1, n = 1 \dots N_l\}$ , and  $\tilde{B}_l \in \mathbb{R}^M \times \mathbb{R}^{N_l}$  is the same concatenation for the corresponding target vectors  $\tilde{\mathbf{y}}[n]$ . The learning requires one to consider all reservoir responses  $A_l$  for the different elements of the training set, which are gathered into a matrix  $A$  (of dimension  $N \times (\sum N_l)$ ). The RC outputs  $W^O A$  are expected to provide the correct corresponding answers  $\tilde{B}$  (of dimension  $M \times (\sum N_l)$ , where  $\tilde{B}$  is the concatenation of the target matrices  $\tilde{B}_l$ ):  $\tilde{B} = W^O A$ . Ridge regression can be applied to solve this ill-posed problem through the following formula giving the optimal read-out matrix:  $W_{opt}^O = \tilde{B} A^T (A A^T - \lambda I)^{-1}$ , where  $\lambda$  is the small regression parameter,  $I$  is the  $N \times N$  identity matrix, and the matrix inversion can be calculated through a

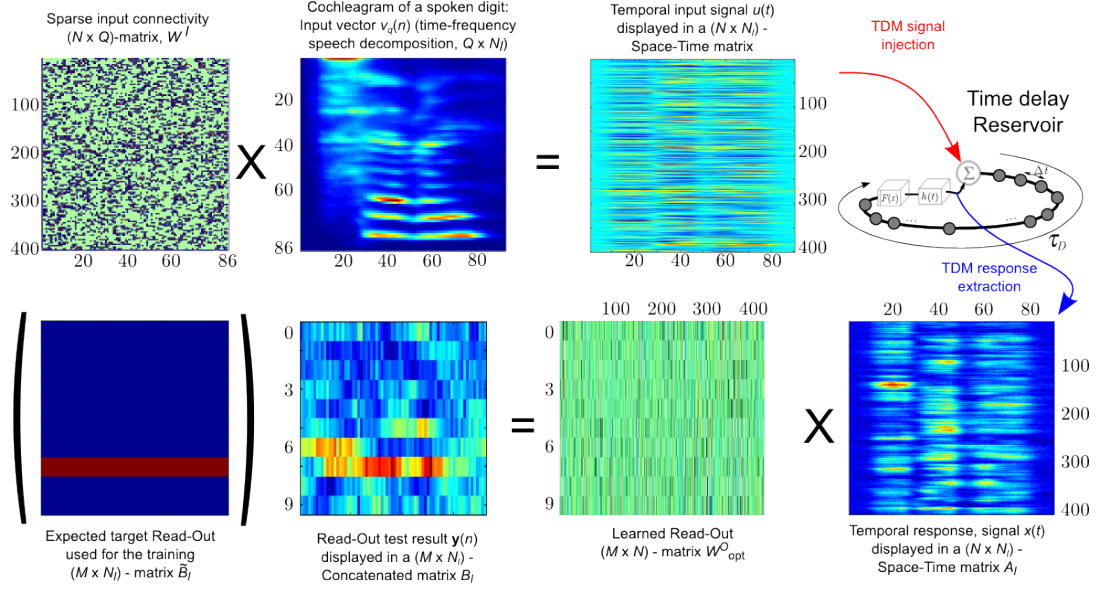


Figure 4.6: Graphical illustration of the RC processing steps in the case of a speech recognition task, performed with an optoelectronic delay oscillator used as a Reservoir with 400 virtual nodes. Each input cochleagram consists of 86 frequency components which energy content (color encoded) are evolving over the duration of the spoken digit (this duration  $N_i$  amounts here to 88 steps in  $n$ ).

Moore-Penrose algorithm.

Reservoir Computing has already achieved much success, revealing its computational potential both in ESN numerical simulations [139, 140], and also in physical hardware implementation. Successful physical hardware implementations have been based on delayed dynamical systems [21, 22, 131, 141–143], but also more recently they have employed real spatially extended photonic systems [144, 145].

Figure 4.6 illustrates the previously described RC processing steps, in the case the processing of a classification problem (speech recognition), as performed with an optoelectronic delay dynamics [22]. It makes an extensive use of the space-time representation for delay dynamical systems.

### 4.3.2 Observation of chimera states

**Acknowledgment:** Laurent Larger performed the experiments and contributed the figures and much of the text for Section 4.3.2.

Chimeras and reservoir computing surprisingly share a temporal and a spatial coincidence. They were “temporally” discovered and invented respectively in the early 2000s [73, 135, 136], and they were “geographically” connected to delay dynamics during the Delay Complex System conference DCS’12, a decade later. Since delay dynamics had demonstrated the ability to emulate a virtual network of neurons in RC applications, a straightforward next step was to confirm the relevance of this network emulation for the experimental observation of chimera patterns. Moreover, chimeras had just been experimentally discovered in 2012, in setups modeled by

spatio-temporal equations [83, 84]. A chimera state in a delayed dynamical system was first observed in 2013 [63], illustrating that networks of dynamical nodes can indeed be emulated by a delayed dynamical system.

One of the models used to numerically explore chimera solutions is the network of continuously distributed coupled Kuramoto oscillators, defined as follows:

$$\frac{\partial \phi}{\partial t} = \omega_0 + \int G(x - \xi) \cdot \sin[\alpha + \phi(t, x) - \phi(t, x - \xi)] d\xi. \quad (4.30)$$

This governs the dynamics of the phases  $\phi(t, x)$  of the oscillators that are continuously distributed in space,  $\omega_0$  being their natural angular frequency. Oscillators have coupled phases according to a sine nonlinear dependency of the coupling (with an important coupling offset  $\alpha$ ), depending on the relative phase difference between the two coupled oscillators at position  $x$  and  $x - \xi$ . Each phase coupling is weighted by a distance-dependent factor  $G(x - \xi)$ , which is typically vanishing beyond a certain coupling distance (sometimes referred as to the coupling radius) defined by the shape of  $G(\cdot)$ . The phase dynamics is thus ruled by the contribution of the coupling with all the other oscillators, as the integral term in Eq. 4.30 covers the entire space of the network. Chimera solutions of such an equation typically consist of clusters, in which oscillators are synchronized with the same phase in a cluster, and in other clusters, oscillators are completely desynchronized with chaotically fluctuating phases.

It is then interesting to compare qualitatively the integral term in Eq. 4.30, with the one derived in Eq. 4.24. As previously discussed and as it can be also inferred from

the comparison with the network of Kuramoto oscillators, one can clearly identify the specific role of  $h(t)$ , when it is considered in the space-time representation of the delay dynamical variable  $x_\sigma[n]$  as derived in Eqs. 4.19 to 4.23. The impulse response  $h(t)$  is clearly ruling the coupling strength and the coupling distance within the virtual network of dynamical nodes. The nonlinear function  $F(x)$  plays the role of the nonlinear coupling between the amplitudes of the virtual nodes.

Figure 4.7 reports typical chimera patterns obtained experimentally with nonlinear delay dynamics. It shows both the temporal waveform during growth and stabilization of the pattern, as well as the space-time representation in the  $(\sigma, n)$ -plane, with color encoding of the waveform amplitude. The space-time picture clearly shows the sustained chimera pattern along the horizontal virtual space domain. It consists of a flat plateau (blue color) surrounded by a chaotic sea (red and orange colors), with which it coexists, filling in a balanced and stable way the shared spatial domain. The figure also shows two possible solutions (single-headed and two-headed chimera), obtained with the same parameter conditions, but produced from different noisy initial conditions. Depending on the temporal parameters (hence the properties of the coupling function  $h(t)$  as depicted in Fig. 4.2b, e.g. the actual values of  $\tau_L$  and  $\tau_H$  relatively to  $\tau_D$ ), one can obtain a highly multistable dynamics of chimera patterns [20].

To comment more on the details under which conditions chimera solutions can be obtained in delay dynamics, it is worth mentioning that  $h(t)$  requires a bandpass profile. There are many different arguments to explain this requirement. The first is related to the carrier waveform of a chimera pattern over the virtual spatial domain

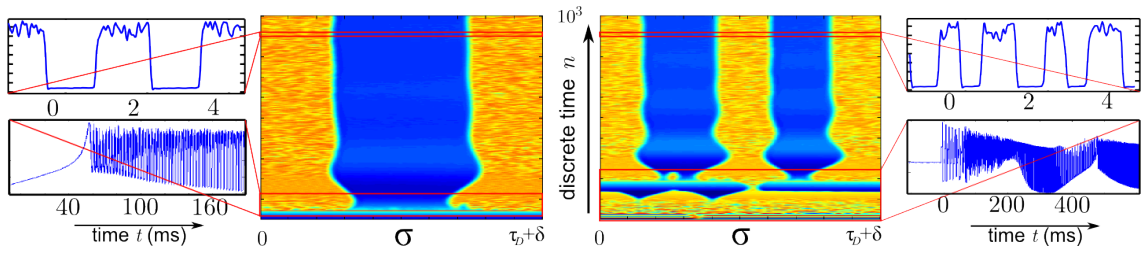


Figure 4.7: Experimental record of single- and two-headed chimera solutions generated in delay dynamics. The two central plots show the space-time representation of the chimeras, as they grow and then stabilize. The side plots, left and right, are temporal waveforms showing parts of the chimera, during the initial transient (the lower time-traces, which cover a few hundreds of time recurrences in the delayed feedback loop), and during the stabilized part at the end of the full record (upper time-traces, which cover approximately two recurrent times  $T = \tau_D + \delta$ ).



$[n(\tau_D + \delta); (n + 1)(\tau_D + \delta)]$ , which is necessarily a stable period-1(delay) carrier waveform, and not a period-2 carrier waveform as usually concerned in the period-doubling bifurcation cascade typically known for delay dynamics. To allow for such a stable period-1 carrier waveform, the bandpass character for  $h(t)$  is necessary (stable period-1 pattern have been analyzed e.g. in [146]), since the low-pass one is known to lead to unstable period-1 pattern, as was reported in [44] about the “coarsening” of any forced initial pattern in the virtual spatial domain. Last but not least, one could also mention that with a fixed  $\tau_L$ , the impulse response with  $\tau_H$  (bandpass) necessarily exhibits a broader width than without the presence of  $\tau_H$  (low-pass). This remark is in line with the known fact that chimera states are favored when the coupling range is extended (i.e., beyond the classical case of nearest neighbor coupling only, which does not allow for chimera states).

From the point of view of the nonlinear coupling function between virtual nodes (as the function is involved in Eq. 4.24), there are also specific requirements on  $F(x)$  for obtaining chimera solutions. This is illustrated in Fig. 4.8, where both the nonlinear function profile is represented, and next to it, with the same vertical scaling, the temporal chimera waveform. From the standard fixed point analysis for a nonlinear map defined by the same function  $F(x)$ , one can notice the following:

- The nonlinear function operates around an average value centered along a positive slope of  $F(x)$ , between two extrema, where an unstable fixed point for the map is located (middle black circle);
- The high amplitude chaotic part of the chimera waveform corresponds to the

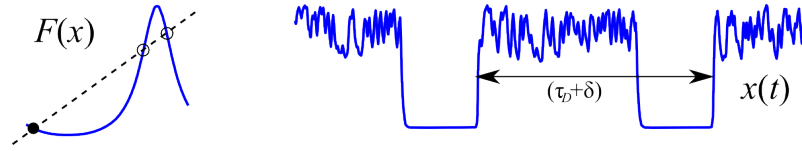


Figure 4.8: Features of the nonlinear coupling function for obtaining chimera patterns in delay dynamics. Left: nonlinear function profile  $F(x)$ , with a dotted first bisector line highlighting the fixed points for a map  $x_{n+1} = F(x_n)$ . Right: amplitude correspondence in the temporal chimera waveform  $x(t)$ .

sharp maximum of  $F(x)$ , and it develops a chaotic motion along this maximum, essentially on the negative slope side and centered around an unstable fixed point (upper-right black circle);

- The low amplitude plateau of the chimera waveform corresponds to a stable fixed point (lower-left black disk) of the map, along a weak negative slope, thanks to the presence of a broad minimum.

This remark points out the important requirement on  $F(x)$  about its necessary asymmetric shape resulting in a sharp maximum and a broad minimum. This was experimentally obtained in [20] with the Airy function provided by a low finesse Fabry-Pérot resonator, which is providing a non linear transformation of the wavelength of a dynamically tunable laser diode, into the output optical intensity of the Fabry-Pérot.

Space-time representation was recently found not to be restricted to a single virtual space dimension. Adding a second delay much larger than the first one, enabled 2D chimeras to be obtained in delay systems. Among various solutions observed in this two-delay system, one could observe chaotic islands surrounded by a calm sea, or its contrary, a flat plateau island in the middle of a chaotic sea [147].

## 4.4 Arbitrary networks in a single delay system

### 4.4.1 Theoretical description

Section 4.3 described the realization of circularly symmetric networks in a single nonlinear system with delayed feedback. In these experiments, the network nodes were time slots of length  $\Delta t$ , where  $\Delta t \ll \tau_D$ , and the coupling between nodes was due to the inherent bandwidth of the electronics. This inherent bandwidth was described using a time-invariant infinite impulse response filter; the time invariance results in a circularly symmetric network. However, Eq. 4.5 does not require the impulse response to be time-invariant. In this section, we describe recent work that uses a digital filter with a time-varying impulse response to realize arbitrary networks in an experimental delay system [123].

There are two modifications of previous systems necessary in order to obtain a network with arbitrary topology. *(a)* the inherent circularly symmetric coupling due to the (time-invariant) bandwidth limitations of the system must be removed. *(b)* the desired coupling must be implemented by an appropriately designed filter with a time-dependent impulse.

#### **Removing the inherent coupling**

There are two convenient options for removing the inherent circularly symmetric coupling due to the time-invariant bandwidth limitations of the system.

(I) Perhaps the most straightforward way to remove the coupling due to the bandwidth limitations of the system is to extend the  $\Delta t$  described in Section 4.3.

This can be done in the pulsed laser system described by Eq. 2.3 by choosing the pulse repetition rate  $f_r = N/\tau_D \ll 1/\tau_L$ . In this case, the filter response decays before the next pulse arrives, and so the system reduces to the  $N$ -dimensional map:

$$x[k] = \beta F(x[k - N]), \quad (4.31)$$

where  $k$  is the discrete time. This map equation requires the specification of  $N$  different initial conditions, but the trajectory of each initial condition is completely independent of the trajectories of the others. Therefore, 4.31 can be thought of as a set of  $N$  completely independent but *truly identical* oscillators using the space-time representation:

$$x_i[n] = \beta F(x_i[n - 1]), \quad (4.32)$$

where  $i = k \bmod N$  is the oscillator number and  $n$  is the network time.

(II) An easier-to-implement experiment that displays the same map dynamics is obtained by using a CW laser and sample-and-hold electronics that are clocked at a rate  $f_r$ . Synchronously clocked shift registers have long been used to implement digital delays in experimental set-ups because of the ease of varying the delay [20, 21, 23, 49, 56, 62, 64, 123]. Such a system can also be described by Eq. 2.3. However, in previous experiments, the clock rates have typically been chosen so that the discrete-time nature of the digital delay line minimally impact the dynamics; that is, the sampling time  $\Delta t = 1/f_r$  has typically been much smaller than any other dynamical time scale, and so the digital delay line is a good approximation of an analog delay. In these cases, the experiment is well-described by Eq. 4.1. Here, we

intentionally choose a sampling time that is much longer than the other dynamical time scales in the system, but still shorter than the time delay  $\tau_D = N\Delta t$ . With this choice of clock rate, the dynamics of the system is well-described by Eq. 4.32.

### Implementing an arbitrary coupling topology

The systems described in the last few paragraphs create  $N$  identical, uncoupled nodes using a single delayed dynamical system. In order to couple the nodes together in a network, we must implement a filter that can be described by a time-varying impulse response. This is easiest to do with a digital filter, since in this case we are not restricted by what can be easily implemented by analog components.

It is convenient to implement both the delay and the digital filter on a single device such as a field-programmable gate array (FPGA). In this case, the filter can be *acausal* in the sense that we can implement the following

$$x[k] = \sum_{m=-\infty}^{(k+N-i-1)} h[k-m; k] F(x[m-N]), \quad (4.33)$$

where the impulse response  $h$  is explicitly written as a function of the discrete time  $k$  to denote that it is varying in time. The acausality of the filter is necessary in order to permit couplings to node  $i$  from nodes  $j > i$ .

The impulse response of the digital filter necessary to implement a given network is determined by the adjacency matrix  $A_{ij}$  that describes the network as follows:

$$h[m; k] = \begin{cases} \beta & \text{if } m = k \\ \sigma A_{ij} & \text{if } m \neq k \text{ and } m = k - i + j, \\ 0 & \text{otherwise} \end{cases} \quad (4.34)$$

where  $i = k \bmod N$  and  $j$  is an integer between 0 and  $N - 1$ .

When the digital filter described by the impulse response in Eq. 4.34 is implemented and Eq. 4.33 is written in the space-time representation, we obtain

$$x_i[n] = \beta F(x_i[n-1]) + \sigma \sum_j A_{ij} F(x_j[n-1]) = \beta I_i[n-1] + \sigma \sum_j A_{ij} I_j[n-1], \quad (4.35)$$

where  $I_i[n]$  is the normalized light intensity associated with the  $i^{th}$  node at network time  $n$ . Equation 4.35 describes a network of discrete-time oscillators that are coupled by the arbitrary adjacency matrix  $A_{ij}$ .

There are two adjustments, then, that need to be made to the systems described in Section 4.3 in order to realize an arbitrary network of coupled oscillators in a single delay system:

1. Time must be discretized in such a way as to break the nearest-neighbor coupling that would otherwise be induced by the bandwidth limitations of the system.
2. A filter with a time-dependent impulse response must be used in order to obtain a network topology that is not cyclically symmetric. This filter must also be acausal to allow for the construction of all possible networks (e.g. to couple node  $N - 1$  to node 0).

**Note about terminology:** In this section, we use the terms “time-dependent” or “time-varying” and “acausal” to describe filters in ways that differ from their usual meaning in signal processing contexts. This is done in order to contextualize our work implementing arbitrary networks in a single time-delayed system within the formalism developed to describe the implementation of ring networks in a single time-delayed system. In this section, we say that our FPGA implements a filter with a time-varying impulse response; those in the signal processing community might classify this as an  $N$ -dimensional linear time-invariant filter with temporally interleaved outputs. Similarly, we say that our filter is “acausal” in the sense that node  $N - 1$  can be coupled to node 0. This “acausality” is only possible because both the time delay and the digital filter are implemented on the same FPGA. In Section 4.4.2, we present an equivalent formalism using multiple time delays and temporal multiplexing that does not use any “time-varying” or “acausal” filters.

#### 4.4.2 Alternative interpretation

There is an alternative (but equivalent) way to view the technique used to create arbitrary networks that does not involve acausal or time-varying filtering. This perspective is described in detail in Ref. [123]. Here, the acausal, time-varying filter is replaced by multiple delays that are switched on and off as a function of time in order to implement the desired network. The idea of using multiple time delays to create a more interesting network was pioneered for the purpose of reservoir computing [62]; however, in this case each delay was always switched on, resulting



again in a circularly symmetric network (albeit with longer range connections than with a single delay). Switching the additional delays on and off in time breaks the time-invariance (and therefore circular symmetry of the network) and allows an arbitrary network topology. The time-dependent switching is determined according to the following recipe:

- The time delay of length  $N/f_s$  is always switched ON. This is the feedback time delay and is multiplied by  $\beta$ . This delay is modeled by the first term in Eq. 4.35.
- Time delays of length  $(N + i - j)/f_s$  are switched ON if  $A_{ij} = 1$ , where  $i = k \bmod N$  is the active node. These time delays determine the coupling and are summed then multiplied by  $\sigma$ . This is modeled by the second term in Eq. 4.35.
- All other time delays are switched OFF.

Time delays and switches are easily implemented in FPGA, making this a particularly powerful implementation because the networks are easy to reconfigure.

### 4.4.3 Experimental implementation

The techniques described in Section 4.4.1 and 4.4.2 have been used to implement arbitrary networks in an optoelectronic feedback loop [123]. Illustrations of these (equivalent) experiments are shown in Figs. 4.9 and 4.10, respectively. Light (850 nm) of constant intensity is emitted from a fiber-coupled CW laser (Thorlabs

DBR852P). The light passes through an integrated electro-optic intensity modulator (EOSpace AZ-0K5-10-PFA-SFA-850-UL), which provides a nonlinearity that is well-modeled by  $F(x) = \sin^2(x + \phi_0)$ , where  $x = \pi v_{RF}/2V_{\pi,RF}$  is the normalized RF voltage applied to the modulator and  $\phi_0 = \pi v_{DC}/2V_{\pi,DC}$  is a constant DC bias. A measurement of the modulator's nonlinearity is shown in Fig. 4.11. The half-wave voltages  $V_{\pi,DC} = 3.0$  V and  $V_{\pi,RF} = 2.40$  V for our modulator. A variable DC power supply (Sorensen XDL 35-5T) is used to set the DC bias voltage  $v_{DC}$ . The light is converted to an electrical signal by a photoreceiver (New Focus 1811) and sampled at a frequency  $f_r$  by an analog to digital converter (ADC). The FPGA (Altera Cyclone V GT) implements the delay and the time-dependent digital filtering described in Section 4.4.1 (or, equivalently, the multiple time delays and temporal multiplexing described in Section 4.4.2), and outputs the feedback electrical signal through a digital to analog converter (DAC). This electrical signal is amplified and fed back to the RF input of the intensity modulator, completing the feedback loop. Both the normalized intensity  $I = \mathcal{I}/\mathcal{I}_{max}$  measured by the photoreceiver and ADC and the normalized voltage  $x$  output by the DAC are streamed from the FPGA to a personal computer (PC) via PCIe using the Xillybus IP core for storage. Xillybus is freely available online for use in academic research. Further details regarding the experimental implementation can be found in Appendix A.

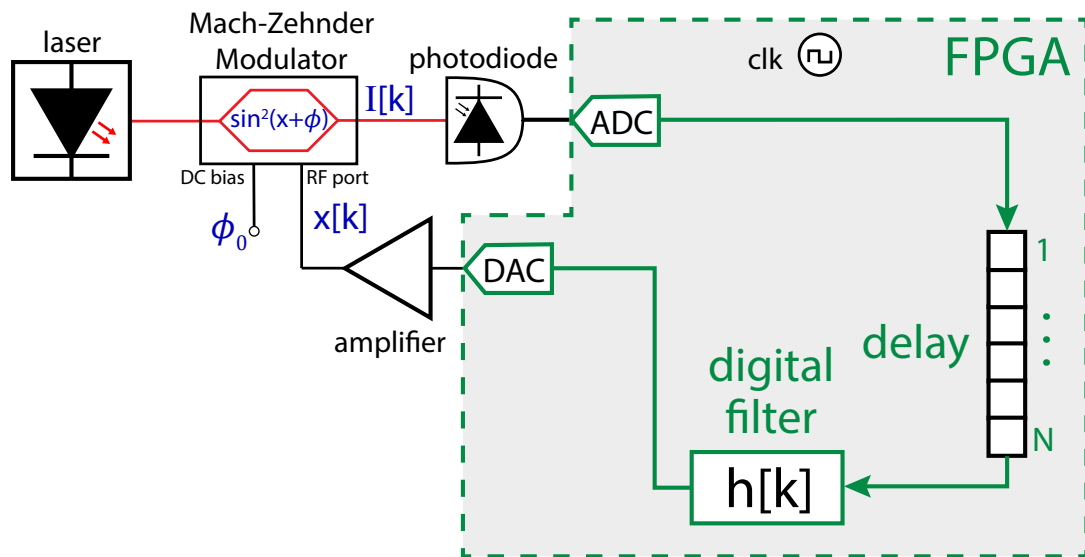


Figure 4.9: Experimental apparatus for realizing arbitrary networks using a single nonlinearity with time multiplexing through a single delay and time-dependent filtering.

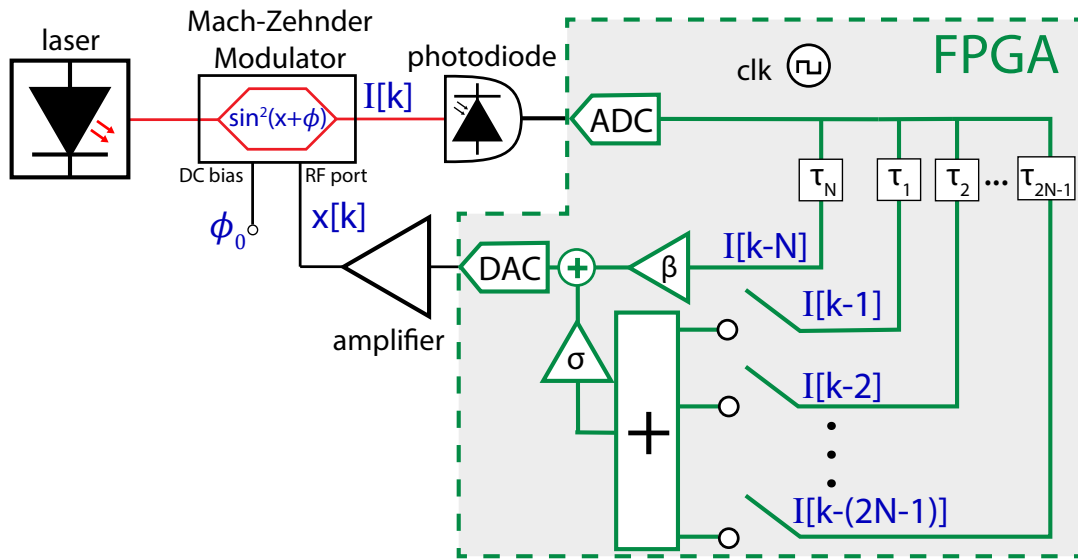


Figure 4.10: Experimental apparatus for realizing arbitrary networks using a single nonlinearity with time multiplexing through multiple time-delays that are switched on and off in time.

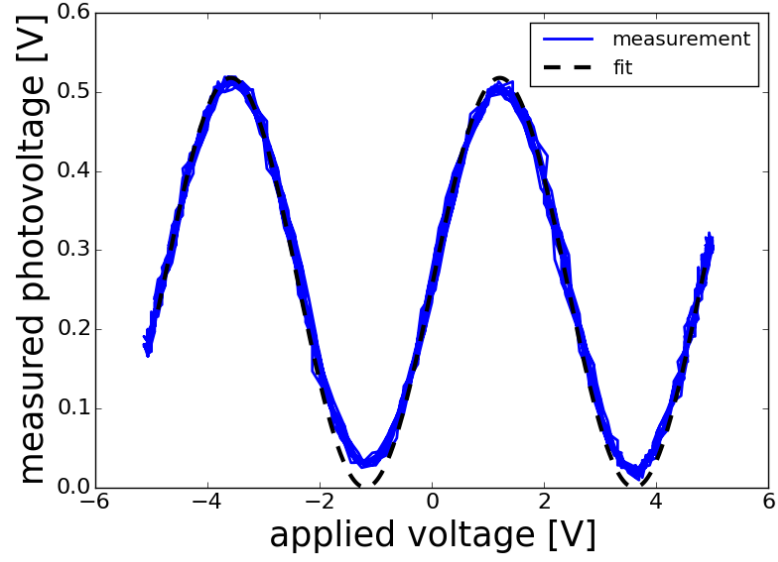


Figure 4.11: Nonlinearity of the integrated lithium niobate Mach-Zehnder intensity modulator. The solid blue curve shows the experimental measurement, and the black dotted curve shows the best fit to  $A \sin^2(\pi v/2V_\pi + \phi_0)$ . As determined by the fit,  $A = 0.52$  V,  $V_\pi = 2.40$  V, and  $\phi_0 = 0.78$ . The RMS error of the fit is 0.018.

## 4.5 Discussion

The realization of networks of coupled oscillators is a challenging experimental task because of the difficulty and expense of obtaining, coupling, and measuring a large number of identical oscillators. In this section, we have reviewed recently developed techniques that overcome these obstacles by implementing the network in a single nonlinear delay system through temporal multiplexing. They offer the additional benefit, impossible in other network implementations, that the oscillators are *truly identical* since they are all implemented in the same physical hardware. These delay networks were first developed for their vast potential as a physical implementation of reservoir computing. In addition to these important information processing applications, delay networks are also opening up entirely new avenues of research in basic experimental science, as exemplified by the observation of novel 1 and 2 dimensional chimera states and cluster synchronization. These techniques, first conceived only in 2011, are still in their infancy and continue to stimulate basic and applied research.

Future work might explore the use of experimental arbitrary networks for hardware-based reservoir computing, where a time-dependent filter impulse response might allow for faster or more accurate information processing. This technique can also be used for the experimental study of a variety of fundamental questions of network dynamics, including the impact of targeted perturbations on network dynamics [119,148], the effect of heterogeneities on network dynamics [122,149], the control of network dynamics [150], and the impact of noise on network dynamics.

While the delay systems themselves are often continuous time systems, the space-time representation causes delay networks to be discrete in time. Research is currently under way to allow the realization of continuous-time networks in a single delay by adopting the multiple time delay implementation of arbitrary networks. Importantly, this technique is not reliant on opto-electronics: one could replace the optics with any system of interest. This might be useful for building prototypes for large networks of coupled oscillators when the oscillators are expensive, such as in the case of power grids. It may also allow for the experimental study of large networks of truly identical oscillators in situations where the oscillators are rarely identical in practice (e.g. biological systems such as neurons). This permits the study of the impact of heterogeneity on the network dynamics.

## Chapter 5: Experimental observation of patterns of synchronization using a single nonlinear node as a network

If it disagrees with experiment, it's wrong.

Thats all there is to it.

---

RICHARD FEYNMAN

### 5.1 Overview

This chapter presents the results of a series of experiments to study the dynamics of networks of coupled oscillators that were performed using the single nonlinear node with time-delayed feedback described in Chapter 4. We begin by exploring the dynamics of a single, discrete-time nonlinear oscillator in Section 5.2. We experimentally measure the bifurcation diagram of a single node, and find that our system can display fixed point, periodic, or chaotic behavior, depending on the system parameters. We compare the experimental bifurcation diagram with one determined by numerical simulation of a mathematical model and find good agreement.

In Section 5.3 we investigate the synchronization of two identical nodes that are bidirectionally coupled. We present an illustration of the experimental imple-



mentation using the multiple delays and time-multiplexing interpretation described in the general case in Section 4.4.2. We then experimentally measure synchronization error as a function of the coupling strength, and we compare it with numerical simulations both with and without noise. We find excellent agreement when the appropriate noise level is used.

We then revisit the patterns of synchronization in globally-coupled networks from Chapter 3. The experimental apparatus used in Chapter 3 was limited to networks of up to four nodes; however, it was predicted that chimera and cluster states should be allowed to form in globally-coupled networks of any size. In Section 5.4 we describe the experimental observation of chimera states in a globally-coupled network of five nodes. We further show that these triplet-singlet-singlet chimera states arise from a symmetry breaking bifurcation called “isolated desynchronization” in which the cluster of two in a triplet-doublet cluster state becomes unstable.

We provide the first experimental observation of the cluster synchronization of the equitable partition of a network in Section 5.5. In 2014 it was shown how to determine the stability of a pattern of synchronization corresponding to the orbital partition of a network [113]; however, it was recently pointed out that there may be some additional partitions that can also synchronize, even though they are not associated with a symmetry [112]. We observe the synchronization of these so-called “input-clusters,” and discuss how the machinery developed for determining the stability of symmetry-clusters can be used to calculate the stability of input-clusters as well.

In Section 5.6 we shift from studying what patterns of synchronization exist

and whether they are stable to the question of how to optimize the stability of a pattern of synchronization. We find that, in almost all Laplacian networks, the region of parameter space in which a given pattern of synchronization is stable can be extended by removing links to break or reduce the topological symmetry of the network. We term this phenomenon structural Asymmetry-Induced Synchronization, or AISync. We describe the theory underlying this general phenomenon and provide experimental verification by using our set-up to implement three different networks, two of 16 nodes and one of 17 nodes.

## 5.2 Dynamics of a single nonlinear map

*This section is based on work from the following publication:*

*Joseph D Hart, Don C Schmadel, Thomas E Murphy, and Rajarshi Roy. “Experiments with arbitrary networks in time-multiplexed delay systems.” *Chaos*: 27(12), 121103 (2017).*

We first discuss the simplest version of the experiment: operation as a single nonlinear map. Figure 5.1 shows an illustration of the experiment. The details of the experimental apparatus are the same as those described in Section 4.4.3. Light from an 850 nm continuous-wave, fiber-coupled distributed feedback laser passes through an integrated LiNO<sub>3</sub> electro-optic intensity modulator and is converted into an electrical signal by a photoreceiver. A sample-and-hold operation is performed on the electrical signal at a rate of  $F_s = 20$  kHz. This is implemented on an FPGA (which controls a DAC and ADC), and the delay caused by the clocked sample-and-hold is

much longer than the optical delay. This makes time discrete and decouples consecutive time steps. The electronic output of the DAC is amplified then applied to the RF port of the modulator. The modulator provides the nonlinearity. An independent power supply controls the DC bias of the electro-optic modulator. There is no digital filtering in the implementation of a single node.

This experiment can be described by the following nonlinear map:

$$x[k+1] = \beta I(x[k]), \quad (5.1)$$

where  $x = \pi v / 2V_{\pi, RF}$  is the normalized voltage applied to the modulator at discrete time  $k$ , and  $\beta$  is the normalized round-trip gain. The normalized intensity of the light passing through the electro-optic modulator can be modeled as  $I(x) = \sin^2(x + \phi_0)$ , where  $\phi_0 \equiv \pi V_{DC} / 2V_{\pi, DC}$  is the DC bias point of the electro-optic modulator. The sine-squared nonlinearity is intrinsic to all wave-interference devices, including our intensity modulator. With this experimental system, one could obtain any desired nonlinearity  $I(x)$  by performing a nonlinear operation on  $x$  in the FPGA. Alternatively, one could introduce a different nonlinearity by using different, sufficiently fast nonlinear optical or electronic components.

Figure 5.2a shows an experimentally measured bifurcation diagram of the single nonlinear map with  $\phi_0 = \pi/4$ . Our experimental map can exhibit fixed point, periodic, and chaotic behaviors. A bifurcation diagram numerically simulated from Eq. 5.1 and shown in Fig. 5.2b, agrees well with the experiment and suggests that Eq. 5.1 is an accurate model.

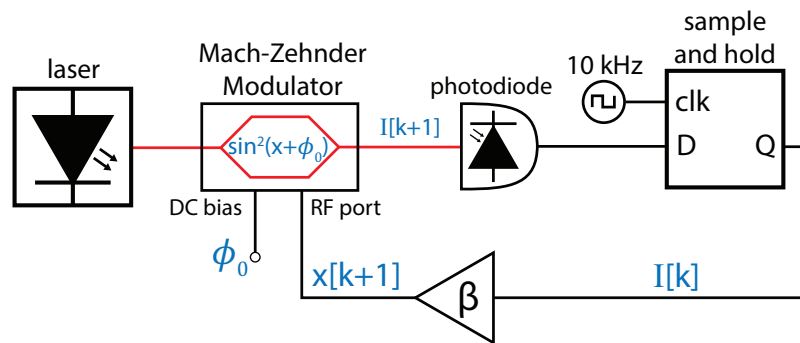


Figure 5.1: Illustration of the setup for a single nonlinear map. In our experiment, the flip flop and clock are implemented on a FPGA.

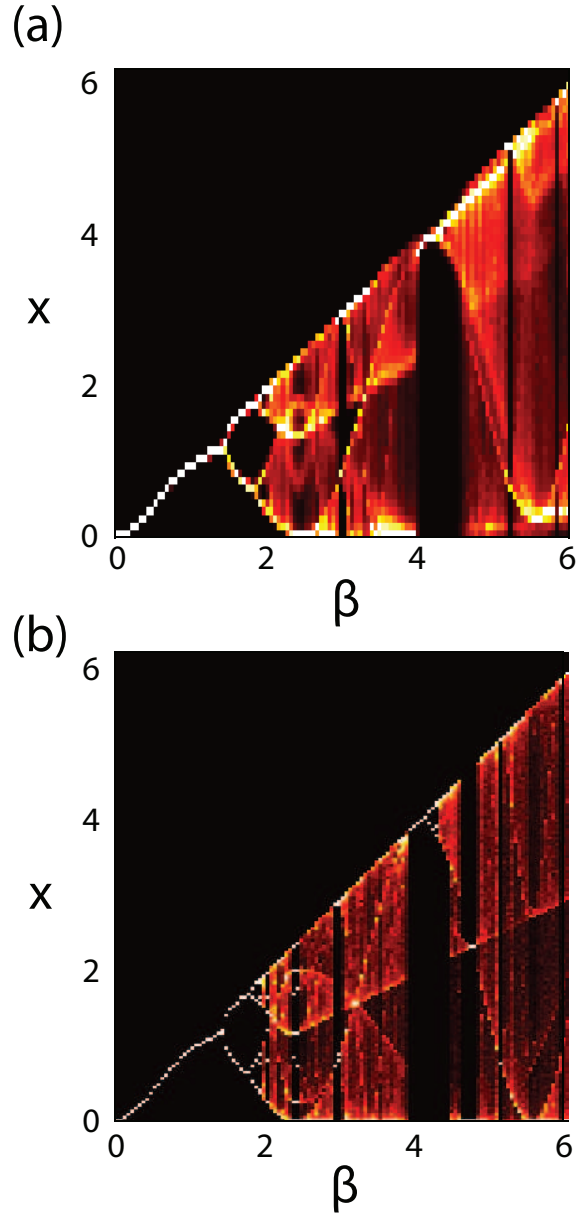


Figure 5.2: *a)* Experimental bifurcation diagram of the single nonlinear oscillator depicted in Fig. 1. The DC bias  $\phi_0$  was set to  $\pi/4$ . *b)* Numerically simulated bifurcation diagram of the single nonlinear map described by Eq. 1.  $\phi_0$  was set to  $\pi/4$ . 15000 samples were used for each value of  $\beta$ .

### 5.3 Synchronization of two coupled maps

*This section is based on work from the following publication:*

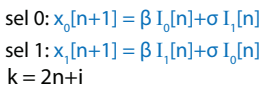
*Joseph D Hart, Don C Schmadel, Thomas E Murphy, and Rajarshi Roy. “Experiments with arbitrary networks in time-multiplexed delay systems.” *Chaos*: 27(12), 121103 (2017).*

The simplest example of a network of coupled oscillators is perhaps a bi-directionally coupled pair of maps. Such a network can be modeled as

$$\begin{aligned}x_0[n+1] &= \beta I(x_0[n]) + \sigma I(x_1[n]) \\x_1[n+1] &= \beta I(x_1[n]) + \sigma I(x_0[n]),\end{aligned}\tag{5.2}$$

where  $\sigma$  is the coupling strength. We implement this simple network as a first example of the implementation of a network of coupled maps using a single nonlinear element with time-delayed feedback and time multiplexing, using the interpretation presented in Section [4.4.2](#).

The experimental implementation for a 2-node, bidirectionally coupled network is shown explicitly in Fig. [5.3](#). In this setup, the single oscillator with time-delays and a multiplexer (MUX) functions as two individual nonlinear maps coupled together. Every even time step  $k$  of the full system is interpreted as an iteration of node 0, and every odd time step  $k$  of the full system is interpreted as an iteration of node 1. This system can also be modeled by Eq. [5.2](#), but requires the assembly of only one experimental apparatus. Further, each of the two nodes are identical, since they utilize the same electronic and optical components. The individual maps



119

now update at half the rate since the measurement of  $I$  is performed serially.

In practice, the sample-and-hold, time delays, time multiplexing, multiplication by  $\beta$  and  $\sigma$ , and addition of the self-feedback and coupling components are all implemented on an FPGA. This allows for a compact and flexible implementation of the experiment: The experiment can be switched between the single nonlinear map described in section 5.2 and the coupled map system simply by reconfiguring the FPGA.

Figure 5.4 shows the measured and simulated root-mean-square (RMS) synchronization error of the coupled map system depicted in Fig. 5.3 and described by Eq. 5.2. We define the RMS synchronization error as

$$\theta \equiv \left( \frac{\langle (x_0[n] - x_1[n])^2 \rangle}{\langle x_0[n]^2 + x_1[n]^2 \rangle} \right)^{1/2}, \quad (5.3)$$

where  $\langle \cdot \rangle$  denotes an average over time.  $\theta$  is zero in the case of a completely synchronized solution and approaches 1 in the limit that the  $x_0$  and  $x_1$  are uncorrelated.

One can determine the stability of the synchronized state by linearizing about the synchronized solution  $x_s[n] \equiv x_0 = x_1$  and calculating the Lyapunov exponent of the variational equation for the perturbations transverse to the synchronization manifold. The variational equation is

$$\Delta x_{\perp}[n+1] = (\beta - \sigma) \sin(2(x_s[n] + \phi_0)) \Delta x_{\perp}[n], \quad (5.4)$$

where  $\Delta x_{\perp}$  is a perturbation transverse to the synchronization manifold. We have calculated the Lyapunov exponent of Eq. 5.4 as a function of  $\sigma$  for fixed  $\beta = 3.5$ ;



the results are shown in Fig. 5.4. The values of  $\sigma$  with negative Lyapunov exponent correspond exactly to the values where we observe synchronization in the simulations without noise. Further, all of the  $\sigma$  values where we observe synchronization in the experiment correspond to  $\sigma$  with negative Lyapunov exponent; however there are some narrow regions of  $\sigma$  with negative Lyapunov exponent where we do not observe synchronization in the experiment. This is due to noise in the experiment. Noise in the experiment comes from a variety of sources, including discretization noise in the ADC and DAC, electronic noise in the DAC amplifier, and Johnson noise in the photoreceiver.

We model all of these sources of noise by applying additive white Gaussian noise with standard deviation 0.001 to each normalized intensity  $I_i$  at each time step:

$$\begin{aligned} x_0[n+1] &= \beta(I(x_0[n]) + aR_0[n]) + \sigma(I(x_1[n]) + aR_0[n]) \\ x_1[n+1] &= \beta(I(x_1[n]) + aR_1[n]) + \sigma(I(x_0[n]) + aR_0[n]) \end{aligned} \tag{5.5}$$

where  $R_i[n]$  are independent, identically distributed random variables taken from a Gaussian distribution with zero mean and unit variance. The synchronization error from the simulation with  $a = 0.001$ , shown in red in Fig. 5.4, shows good agreement with the experimentally measured result.

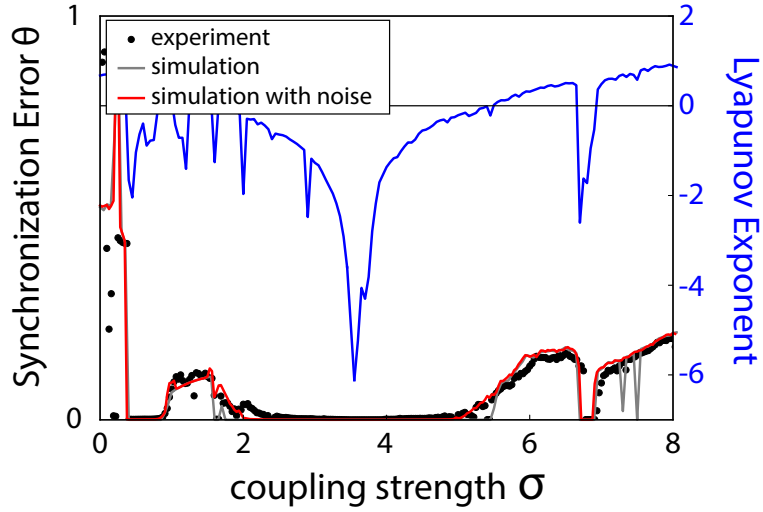


Figure 5.4: Synchronization of two nodes as a function of  $\sigma$  for  $\beta = 3.5$  and  $\phi_0 = \pi/4$ . The coupling is symmetric and bidirectional, as in Eq. 5.2. The gray line indicates the synchronization error in a noiseless simulation, while the red line indicates the synchronization error in a simulation in which Gaussian noise with standard deviation 0.001 was added to the intensities  $I$  at each time step. The blue line indicates the numerically computed Lyapunov exponent corresponding to perturbations transverse to the synchronization manifold.

## 5.4 Chimera states in a five-node globally-coupled network

*This section is based on work from the following publication:*

*Joseph D Hart, Don C Schmadel, Thomas E Murphy, and Rajarshi Roy. “Experiments with arbitrary networks in time-multiplexed delay systems.” *Chaos*: 27(12), 121103 (2017).*

Experimental observations and stability analysis of chimera states in small networks of four globally-coupled continuous time oscillators have been studied in Chapter 3. It was predicted that similar chimera states should exist for larger globally coupled networks, and that it should be possible to determine their stability using the analysis techniques described in Ref. [23, 113, 114] and in Chapter 3. However, due to experimental limitations on the number of nodes and links, chimeras in larger networks of globally coupled oscillators have not been experimentally observed in opto-electronic networks.

We now investigate chimera states in a network of five globally-coupled nodes in our experimental system. The chimera state we consider here is a dynamical state in which three of the nodes are synchronized, and the other two nodes are desynchronized both from the cluster of three and from each other. The group theoretical arguments from Section 3.4.1 suggest that such chimera states can exist in globally coupled networks [23].

By implementing the digital filtering procedures described in Sec. 4.4.3, we can modify our set-up to realize a globally coupled network of 5 nodes. As in Ref. [23]

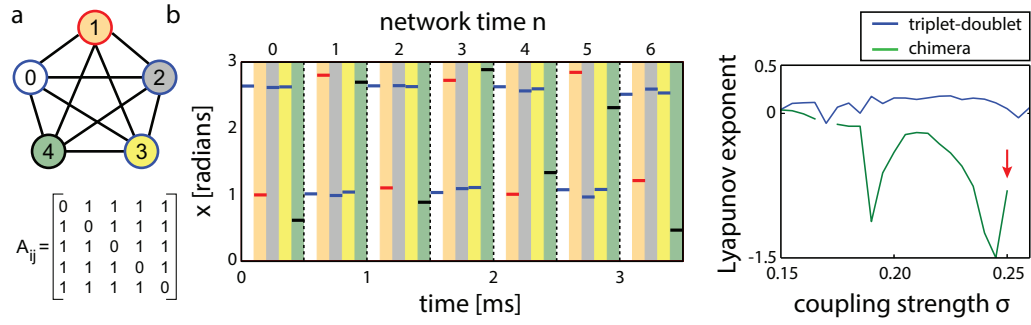


Figure 5.5: *a)* Depiction of a 5 node globally coupled network. The color of the outline on each node indicates which synchronous cluster it belongs to. *b)* Time series of the 5 node network in the chimera state depicted in *a*. The background color of each time step indicates the node from *a* that is represented by that time step. The coloring of the horizontal bar indicates the synchronous cluster to which that node belongs. The vertical black dotted lines indicate a full network time step  $n$ . *c)* Stability of chimera state and triplet-doublet cluster state. The red arrow indicates the value of  $\sigma$  used to obtain the experimental time series in *b*.

and Chapter 3, we added a coupling time delay to observe chimera states in our globally-coupled network. This is straightforward to do with the FPGA hardware via shift registers. The network can be modeled as

$$x_i[n+1] = \beta I(x_i[n]) + \sigma \sum_j A_{ij} I(x_j[n - \tau_c]), \quad (5.6)$$

where  $I(x) = \sin^2(x + \phi_0)$  as before and  $\tau_c$  is the coupling delay. For this experiment, we take  $\tau_c = 1$  iteration.

Figure 5.5b shows a typical time series from a chimera in the experiment. The parameter values are  $\beta = 2.3$ ,  $\sigma = 0.25$ , and  $\phi_0 = \pi/4$ . As one can see from Fig. 5.2, an uncoupled oscillator is chaotic with these parameters. In this chimera, nodes 0, 2, and 3 are in the synchronous (coherent) region, and nodes 1 and 4 are in the desynchronized (incoherent) region. When we start from different initial conditions, different nodes end up in the coherent and incoherent regions. This is expected because all of our nodes are identical.

In order to investigate the stability of these chimera states, we performed a stability analysis according to the methods described in Chapter 3. We linearize about the chimera state to obtain the variational equations for the network, then use the group theory-based techniques presented in Chapter 3 to pick out the perturbation directions transverse to the synchronization manifold. The stability of the chimera state is determined by the largest Lyapunov exponent (LLE) of these transverse variational equations: if the LLE is negative, the chimera state is stable. The result of this calculation for  $\beta = 2.3$  and  $\phi_0 = \pi/4$  is shown in Fig. 5.5c. We

see that the chimera state is stable in the region from  $\sigma = 0.17$  to  $\sigma = 0.25$ .

We were not able to calculate the Lyapunov exponent in the transverse direction for some values of  $\sigma$ . In regions where the chimera state is unstable and a more symmetric state (such as the globally synchronized state or the triplet-doublet state) is stable, the trajectory of the chimera state cannot be determined. Therefore, one cannot linearize about the chimera state and calculate the transverse Lyapunov exponents in the usual way.

We also calculate the stability of the triplet-doublet cluster synchronous state in order to show how the chimera state forms. The results of this calculation are also shown in Fig. 5.5c. For lower values of  $\sigma$ , the triplet-doublet cluster state is stable. As  $\sigma$  increases, the doublet cluster undergoes isolated desynchronization [113] and becomes unstable; however, the triplet cluster remains stable. This results in a triplet-singlet-singlet state, which we call a chimera state.

## 5.5 Synchronization of input-clusters

*This section is based on work from the following publication:*

*Abu Bakar Siddique, Louis Pecora, Joseph D Hart, and Francesco Sorrentino. “Symmetry- and input-cluster synchronization in networks.” *Phys. Rev. E*: 97(4), 042217 (2018).*

### 5.5.1 Motivation

In Chapter 3 we showed how to determine the stability of symmetry clusters by using the IRRs of the permutation symmetry group of the adjacency matrix to create a matrix  $\mathbf{T}$ , and using  $\mathbf{T}$  to transform coordinates into the “IRR space,” in which the stability calculations can be performed. However, this procedure (originally described in Ref. [113]) works only when the clusters are symmetry clusters. There can be input clusters that are not directly related to network symmetries. An example of an input cluster that is not a symmetry cluster is shown in Fig. 5.6a.

Shortly after the original paper on the stability of symmetry clusters, it was recognized that in the case in which there is Laplacian coupling, symmetry clusters can sometimes combine to form synchronous clusters whose nodes are not related by a network symmetry. Sorrentino et al. extended the stability analysis based on symmetries to include so-called Laplacian clusters [114].

Subsequently, it was pointed out that it may be possible for a synchronous cluster that is not related to any network symmetry to form even in non-Laplacian network topologies [112]. In this section, we show how to determine the stability of input-cluster states by adapting the symmetry-based techniques from Ref. [113], and we confirm our stability calculations with the first experimental observation of a non-symmetric synchronous cluster in a non-Laplacian network.

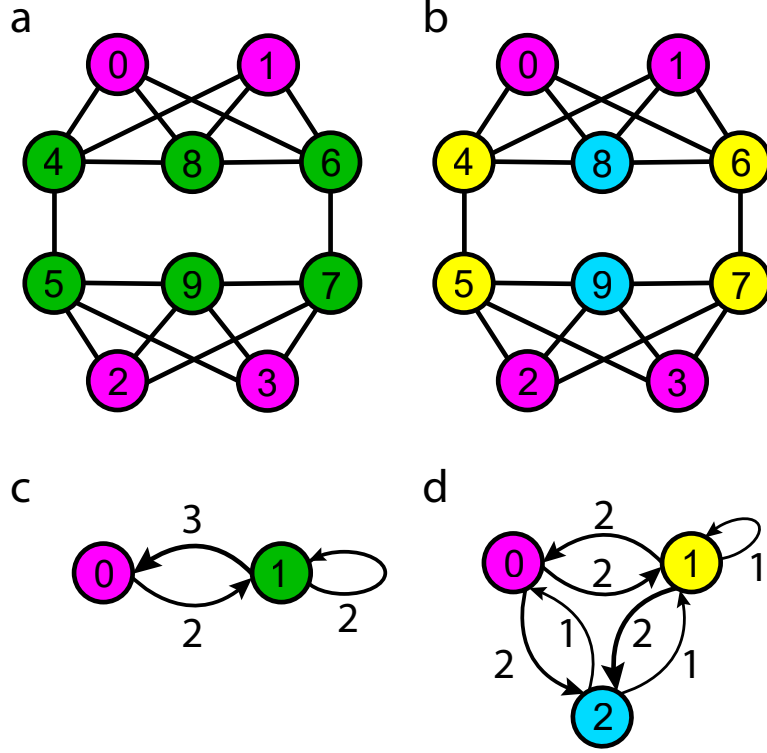


Figure 5.6: *a)* The coarsest equitable partition of a 10-node network. The network is partitioned into two sets: magenta and green. *b)* The coarsest orbital partition of the same 10-node network. The orbits partition the network into three sets: magenta, blue, and yellow. Note that there is no network symmetry that permutes any blue nodes with any yellow nodes. *c)* Quotient network for the equitable partition. *d)* Quotient network for the orbital partition. All link strengths for  $a$ - $b$  are 1; link strengths for  $c$ - $d$  are as indicated.



### 5.5.2 Stability analysis for input clusters

In this section, we describe how to perform the stability analysis for input clusters that are not symmetry clusters. As in Chapter 3, we use the term “input cluster” to refer to a set of nodes that forms a set of the equitable partition of a network. In particular, we will consider the network and input cluster state depicted in Fig. 5.6a, and we will use the discrete-time oscillators described in Section 4.4. The stability analysis closely follows the procedures developed in Refs. [113, 114].

The outline of the stability analysis is as follows:

1. Choose the pattern of input cluster synchronization for which we want to determine the stability. Force the equations of motion into this pattern of cluster synchronization using a “quotient network.” Find the cluster synchronous solution using the equation of motion of the quotient network.
2. Linearize about the cluster synchronous solution.
3. Change coordinates of the equations of motion from the “node coordinate space” to the “modified-IRR coordinate space” associated with the cluster synchronous state. This transformation separates out the cluster synchronization manifold and the transverse directions, similar to what is done in the Master Stability Function approach [78]. Implicit in this step is the determination of the transformation matrix (T-matrix). The determination of the T-matrix for input clusters that are not symmetry clusters is slightly more complicated and is the main difference from the symmetry cluster case.

4. Use standard numerical algorithms to calculate the Lyapunov exponents of the transformed equations of motion. The Lyapunov exponents corresponding to the direction(s) transverse to a given synchronized cluster give the stability of that cluster.

### Step 1: Quotient networks

The pattern of cluster synchronization determines the cluster synchronous solution. The cluster synchronous solution is determined by forcing that cluster pattern upon the equations of motion and then solving them, analytically if possible, but most often numerically. As discussed in Chapter 3, this can be done using a quotient network.

The quotient networks for the full networks shown in Fig. 5.6a-b are shown in Fig. 5.6c-d. Once the quotient network is obtained, the cluster synchronous solution can be determined by solving

$$x_{\mu}^{(s)}[n] = F(x_{\mu}^{(s)}[n-1]) + \sigma \sum_{\nu} Q_{\mu\nu} H(x_{\nu}^{(s)}[n-1]), \quad (5.7)$$

where  $\mu$  labels the cluster and  $x_{\mu}^{(s)}$  is the cluster synchronous solution of the  $\mu^{th}$  cluster. Eq. 5.7 is typically solved numerically.

### Step 2: Linearize about the cluster synchronous solution

In order to determine the stability of the cluster synchronous state calculated in the previous step, we need to study the behavior of small perturbations away from the synchronized trajectory. Consider an infinitesimally small perturbation  $\delta x_i$  to

each node in Eq. 4.35. In order to obtain the equation describing the behavior of this perturbation, we make the change  $x_i \rightarrow x_i + \delta x_i$  and linearize to obtain the variational equation

$$\delta x_i[n] = \frac{\partial F}{\partial x}(x_{\mu_i}^{(s)}[n-1])\delta x_i[n-1] + \sigma \sum_j A_{ij} \frac{\partial H}{\partial x}(x_{\mu_j}^{(s)}[n-1])\delta x_j[n-1], \quad (5.8)$$

where  $x_{\mu_i}^{(s)}[n-1]$  is the cluster synchronous solution corresponding to the cluster  $\mu$  to which node  $i$  belongs.

### Step 3: Transform to IRR coordinate space

Equation 5.8 describes how a small perturbation to a single node evolves in time. However, we are not concerned with all perturbations; we are concerned with only those perturbations that disturb the cluster synchrony. In other words, we need to separate out the perturbations along the synchronization manifold (which do not affect the stability of cluster synchrony) from the perturbations transverse to the synchronization manifold (which determine the stability of cluster synchronization). This is facilitated by changing coordinates from “node coordinate space” to “IRR coordinate space,” where this separation is easily done.

The transformation matrix  $\mathbf{T}$  for symmetry clusters can be calculated using a software implementation [116] of the technique developed in Ref. [113]. The  $\mathbf{T}$  matrix for an input cluster requires an extra step. First, one computes  $\mathbf{T}_{sym}$ , the transformation matrix for the orbital partition one level “finer” (i.e., with more clusters) than the desired equitable partition. Then, one merges two (or more) symmetry clusters to form the desired input cluster according to the method originally

developed for Laplacian clusters in Ref. [114]. This results in one synchronous vector (that corresponds to a perturbation in the plane of the synchronization manifold of the input cluster) and one (or more) vectors that are transverse to the synchronous vector. All these vectors lie in the plane defined by the merged symmetry clusters. An example of this procedure will be provided in the following section.

When we transform to the IRR coordinate space by performing the change of variables  $\delta y_a = \sum_b T_{ab} \delta x_b$  and then multiplying through by  $\mathbf{T}$ , we obtain

$$\delta y_a[n] = \frac{\partial F}{\partial x}(x_{\mu_a}^{(s)}[n-1])\delta y_a[n-1] + \sigma \sum_b B_{ab} \frac{\partial H}{\partial x}(x_{\mu_b}^{(s)}[n-1])\delta y_b[n-1], \quad (5.9)$$

where  $x_{\mu_a}^{(s)}[n-1]$  is the cluster synchronous solution corresponding to the cluster  $\mu$  to which node  $a$  belongs, as before.

The rows of  $\mathbf{T}$  can be arranged such that  $\mathbf{B}$  is block diagonal. The block diagonal form of  $\mathbf{B}$  decouples the variational equations. For a pattern of cluster synchronization with  $M$  clusters, if the  $M$  rows of  $\mathbf{T}$  corresponding to perturbations along the synchronization manifold are on top,  $\mathbf{B}$  will have an  $M \times M$  block in the upper-left corner that corresponds to the synchronization manifold. This block is not used for stability calculations. The remaining blocks are called transverse blocks, and the Lyapunov exponents of Eq. 5.9 that correspond to those blocks determine the stability of the cluster synchronization: if all these transverse Lyapunov exponents are negative, the cluster synchronous state is stable.

#### **Step 4: Calculate the maximal Lyapunov exponents**

**a** Equitable Partition

$$T = \begin{array}{c} \begin{array}{ccc|ccc|cc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -2 & -2 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \left. \begin{array}{l} \lambda_0 \\ \lambda_1' \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \right\} \end{array}$$
  

$$B = \begin{array}{c} \begin{array}{ccc|ccc|cc} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \left. \begin{array}{l} \lambda_0 \\ \lambda_1' \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \right\} \end{array}$$

**b** Orbital Partition

$$T = \begin{array}{c} \begin{array}{ccc|ccc|cc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \left. \begin{array}{l} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \right\} \end{array}$$
  

$$B = \begin{array}{c} \begin{array}{ccc|ccc|cc} 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & -2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \left. \begin{array}{l} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \right\} \end{array}$$

Figure 5.7: *a*)  $\mathbf{T}$  and  $\mathbf{B}$  matrices for the equitable partition of the network shown in Fig. 5.6a. *b*)  $\mathbf{T}$  and  $\mathbf{B}$  matrices for the orbital partition of the network shown in Fig. 5.6b. The shadings correspond with the colors in Fig. 5.6 and indicate which cluster the entries correspond to. Grey shading indicates the synchronization manifold, the Lyapunov exponents of which do not help determine the stability of the clusters.

The Lyapunov exponents of the decoupled variational equations 5.9 determine the stability of the cluster synchronous state for which they are calculated.

### 5.5.3 Stability calculation results

We now explicitly perform the stability calculation for the input cluster synchronous state depicted in Fig. 5.6a. In order to proceed, we must use the explicit forms of the feedback function  $F(x) = \beta \sin^2(x + \phi_0)$  and the coupling function  $H(x) = \sin^2(x + \phi_0)$  with  $\beta = 3.5$  and  $\phi_0 = \pi/4$ .

The cluster synchronous dynamics about which we will linearize are determined by the equation of motion of the quotient network:

$$x_\mu^{(s)}[n] = \beta \sin^2(x_\mu^{(s)}[n-1] + \phi_0) + \sigma \sum_\nu Q_{\mu\nu} \sin^2(x_\nu^{(s)}[n-1] + \phi_0), \quad (5.10)$$

where  $\mu$  labels the cluster and  $x_\mu^{(s)}$  is the cluster synchronous solution of the  $\mu^{th}$  cluster. Eq. 5.10 is typically solved numerically.

The variational equation in IRR coordinates is given by

$$\delta y_i[n] = \beta \sin(2(x_{\mu_i}^{(s)}[n-1] + \phi_0)) \delta y_i[n-1] + \sigma \sum_j B_{ij} \sin(2(x_{\mu_j}^{(s)}[n-1] + \phi_0)) \delta y_j[n-1], \quad (5.11)$$

where  $x_{\mu_i}^{(s)}[n-1]$  is the cluster synchronous solution corresponding to the cluster  $\mu$  to which node  $i$  belongs.

### Equitable partition

We now consider the stability problem applied to the equitable partition. The quotient network for the equitable partition is shown in Fig. 5.6c. Equation 5.10 along

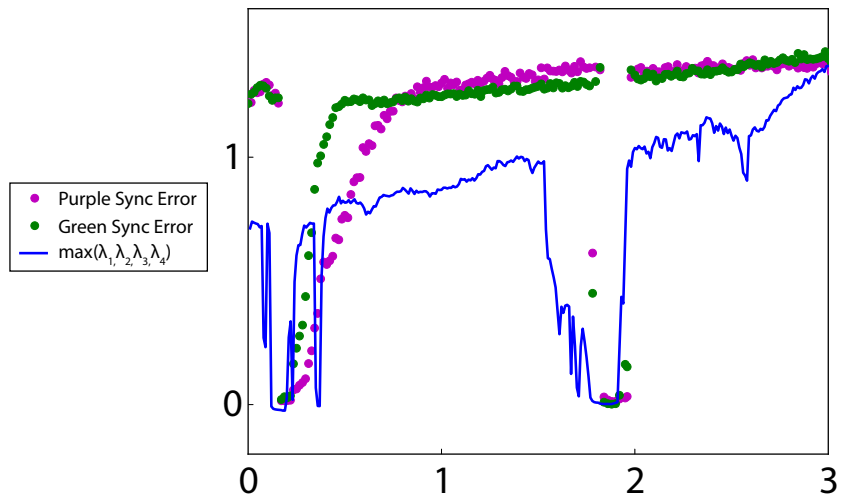


Figure 5.8: Comparison of experimental measurements with stability calculations for the equitable partition of the network shown in Fig. 5.6a. The experiment exhibits input-cluster synchronization in the regions where the largest transverse Lyapunov exponent is negative.

with this quotient network can be used to numerically determine the synchronous dynamics.

By following the method in [114], we can now construct a new transformation matrix  $T'$  that corresponds to the network equitable partition. Namely, we merge the yellow and blue clusters (Fig. 5.6b) to generate a new synchronous vector (describing perturbations along the synchronization manifold)  $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$  and an orthogonal vector  $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -2 & -2 \end{bmatrix}$  in the plane determined by the first and third vectors/rows of the matrix  $T$ . In Fig. 5.7a we present the new transformation matrix  $T$  and the corresponding matrix  $\mathbf{B} = \mathbf{TAT}^{-1}$ . As can be seen, when the clusters are merged, the dimension of the parallel block decreases by 1. A new transverse block is generated with associated MLE  $\lambda'_1$ .

Now that  $\mathbf{B}$  is known, the transverse MLEs (i.e.,  $\lambda_1$ ,  $\lambda'_1$ ,  $\lambda_2, \lambda_3$ , and  $\lambda_4$ ) of the variational equation 5.11 are calculated and used to determine the stability. The calculated transverse MLE is shown in Fig. 5.8 for  $\beta = 3.5$  and  $\phi_0 = \pi/4$  as  $\sigma$  is varied.

#### 5.5.4 Experimental results

We use the experiment described in Chapter 4 to implement the network shown in Fig. 5.6. We fix  $\beta = 3.5$  and  $\phi_0 = \pi/4$ , and we sweep  $\sigma$ . The comparison of the experimentally measured synchronization error with the stability calculations is shown in Fig. 5.8. The synchronization error is defined as  $\Delta = \sqrt{\sum_{1 \leq i \leq N} \|x_i - \bar{x}\|^2 / N}$ ,



where  $\| \cdot \|$  is the shortest distance between two points on a circle, and the mean  $\bar{x}$  on a circle can be found as the phase of  $\sum_{1 \leq j \leq N} e^{ix_j}$ . This definition of distance is necessary because  $x_i$  is defined modulo  $2\pi$  (i.e., on a circle), as described in Chapter 4. Our experimental observation of input-cluster synchronization in the places where the largest transverse Lyapunov exponent is negative confirm the accuracy of our calculations. Therefore our approach of using symmetry-based techniques to perform the stability calculations for non-symmetric input-clusters is validated.

We find an interesting correspondence between the stability of input cluster synchronization and the complexity of the dynamics of the oscillators in the network. For all the parameter sets for which input cluster synchronization is observed, the dynamics of the oscillators are periodic, even though in the absence of any coupling, the oscillators behave chaotically for these parameters. Whether this is simply anecdotal or a more general phenomenon is an interesting question for future research.

## 5.6 Asymmetry-induced synchronization of symmetry clusters

*This section is based on work from the following publication:*

*Joseph D Hart, Yuanzhao Zhang, Rajarshi Roy, and Adilson E Motter. “Topological control of synchronization patterns: Trading symmetry for stability.” Submitted (2018).*

**Acknowledgment:** The work described in Section 5.6 was done in conjunction with Yuanzhao Zhang and Adilson E Motter from Northwestern University. YZ and AEM contributed the theoretical work and numerical calculations. JDH

performed the experiments. All authors contributed to the figures and text. JDH and YZ contributed equally to this work.

### 5.6.1 Motivation

In the previous section, we considered the problem of whether or not a cluster synchronous state is stable for a given network topology. In this section, we study how to make a desired pattern of cluster synchrony more stable by altering the network topology.

We investigate the relation between symmetry and synchronization in the general contexts of global and cluster synchronization. In both cases we show that, in order to improve the synchronizability of a cluster, one often has to break the underlying structural symmetry. This counterintuitive result holds for the general class of networks of Laplacian coupled identical oscillators with bounded stability regions, and follows rigorously from our demonstration that almost all networks (or subnetworks) exhibiting optimal synchronizability are necessarily asymmetric. In particular, the synchronizability of almost any non-intertwined symmetry cluster can be enhanced precisely by breaking the internal structural symmetry of the cluster. This is demonstrated for arbitrary networks and also for the special case in which the entire network consists of a single symmetry cluster. These findings add an important new dimension to the recent discovery of *parametric* asymmetry-induced synchronization [151–153], a scenario in which the synchronization of identically coupled oscillators is enhanced by setting making the oscillators themselves non-identical.

Here, we show that synchronization of identically coupled identical oscillators is enhanced by changing the connection patterns of the oscillators to be non-identical. We refer to this effect as *structural* asymmetry-induced synchronization (AISync). We confirm that this behavior is robust against noise and can be found in real systems by providing the first experimental demonstration of structural AISync using networks of coupled optoelectronic oscillators. In excellent agreement with theory, the experiments show unequivocally that networks can be optimized for both global and cluster synchronization by reducing structural symmetry.

### 5.6.2 Theoretical considerations

We consider a network of  $N$  Laplacian-coupled identical oscillators,

$$\dot{\mathbf{x}}_i = \mathbf{F}(\mathbf{x}_i) + \sigma \sum_{j=1}^N L_{ij} \mathbf{H}(\mathbf{x}_j), \quad (5.12)$$

where  $\mathbf{x}_i$  is the state of the  $i$ -th oscillator,  $\mathbf{F}$  is the vector field governing the uncoupled dynamics of each oscillator,  $\mathbf{L} = \{L_{ij}\}$  is the (negative) Laplacian matrix describing the structure of an arbitrary unweighed network,  $\mathbf{H}$  is the interaction function, and  $\sigma > 0$  is the coupling strength. We are interested in the dynamics inside a symmetry cluster.

For simplicity, we assume the cluster to be optimized is non-intertwined [113, 154]; that is, it can synchronize independent of whether other clusters synchronize or not. The control of the synchronization of intertwined clusters is also important, and we discuss the extension of our results to this case in Section 5.6.4.

Numbering the oscillators in the cluster to be optimized from 1 to  $m$ , we obtain

the dynamical equation for the cluster:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{F}(\mathbf{x}_i) + \sigma \sum_{j=1}^m L_{ij} \mathbf{H}(\mathbf{x}_j) + \sigma \sum_{j=m+1}^N A_{ij} \mathbf{H}(\mathbf{x}_j) \\ &= \mathbf{F}(\mathbf{x}_i) + \sigma \sum_{j=1}^m L_{ij} \mathbf{H}(\mathbf{x}_j) + \sigma \mathbf{I}(\{\mathbf{x}_j\}_{j>m}),\end{aligned}\tag{5.13}$$

where  $L_{ij} = A_{ij} - \delta_{ij}\mu_i$ ,  $\mathbf{A} = \{A_{ij}\}$  is the adjacency matrix of the network,  $\mu_i = \sum_j A_{ij}$ , and the equation holds for  $1 \leq i \leq m$ . Here, we denote the input term from the rest of the network  $\sum_{j=m+1}^N A_{ij} \mathbf{H}(\mathbf{x}_j)$  by  $\mathbf{I}(\{\mathbf{x}_j\}_{j>m})$  to emphasize that this term is independent of  $i$  and hence equal for all oscillators  $1, \dots, m$ . This term is zero, and  $m = N$ , only in the special case in which the entire network consists of a single symmetry cluster.

For  $m < N$ , if we regard the cluster subnetwork consisting of oscillators  $1, \dots, m$  as a separate network (by ignoring its connections with other clusters), then the corresponding  $m \times m$  Laplacian matrix  $\tilde{\mathbf{L}}$  is closely related to the corresponding block of the  $N \times N$  Laplacian matrix  $\mathbf{L}$  of the full network:

$$L_{ij} = \begin{cases} \tilde{L}_{ij}, & 1 \leq i \neq j \leq m, \\ \tilde{L}_{ij} + \tilde{\mu}, & 1 \leq i = j \leq m, \end{cases}\tag{5.14}$$

where  $\tilde{\mu} > 0$  is the number of connections each oscillator in the cluster receives from the rest of the network. It is then clear that there are two differences in the dynamical equation when the cluster subnetwork is part of a larger network (i.e., as a symmetry cluster, described by Eq. 5.13) rather than as an isolated network. First, the Laplacian matrix  $\tilde{\mathbf{L}}$  in the dynamical equation is replaced by  $\hat{\mathbf{L}} = \{L_{ij}\}_{1 \leq i, j \leq m} = \tilde{\mathbf{L}} + \tilde{\mu} \mathbf{1}_m$ ; that is, the diagonal entries are uniformly increased by  $\tilde{\mu}$ . Second, each oscillator now receives a common input  $\sigma \mathbf{I}(\{x_j\}_{j>m})$  produced by

its coupling with other clusters, which generally alters the synchronization trajectory  $\mathbf{s}_I \equiv \mathbf{x}_1 = \dots = \mathbf{x}_m$ , causing it to be different in general from the ones generated by the uncoupled dynamics  $\dot{\mathbf{s}} = \mathbf{F}(\mathbf{s})$ . This has to be accounted for in calculating the maximum Lyapunov exponent transverse to the cluster synchronization manifold to determine the stability of the cluster synchronous state.

Despite these differences, a diagonalization procedure similar to the one used in the master stability function approach [78] can still be applied to the variational equation in order to assess the cluster's synchronization stability. The variational equation describing the evolution of the deviation away from  $\mathbf{s}_I$  inside the cluster can be written as

$$\delta\dot{\mathbf{X}} = \left( \mathbf{1}_m \otimes J\mathbf{F}(\mathbf{s}_I) + \sigma\hat{\mathbf{L}} \otimes J\mathbf{H}(\mathbf{s}_I) \right) \delta\mathbf{X}, \quad (5.15)$$

where  $\delta\mathbf{X} = (\delta\mathbf{x}_1^\top, \dots, \delta\mathbf{x}_m^\top)^\top = (\mathbf{x}_1^\top - \mathbf{s}_I^\top, \dots, \mathbf{x}_m^\top - \mathbf{s}_I^\top)^\top$  and  $\otimes$  denotes the Kronecker product. The rest of the network does not enter the equation explicitly, other than through its influence on the coupling matrix  $\hat{\mathbf{L}}$  and the synchronization trajectory  $\mathbf{s}_I$ . If  $\hat{\mathbf{L}}$  is diagonalizable (as for any undirected network), the decoupling of Eq. 5.15 results in  $m$  independent  $d$ -dimensional equations corresponding to individual perturbation modes:

$$\dot{\eta}_i = [J\mathbf{F}(\mathbf{s}_I) + \sigma\hat{v}_i J\mathbf{H}(\mathbf{s}_I)]\eta_i, \quad (5.16)$$

where  $d$  is the dimension of node dynamics,  $J$  is the Jacobian operator,  $\eta = (\eta_1^\top, \dots, \eta_m^\top)^\top$  is  $\delta\mathbf{X}$  expressed in the new coordinates that diagonalize  $\hat{\mathbf{L}}$ , and  $\hat{v}_i = \tilde{v}_i + \tilde{\mu}$  are the eigenvalues of  $\hat{\mathbf{L}}$  in ascending order of their real parts [with  $\{\tilde{v}_i\} = \text{eig}(\tilde{\mathbf{L}})$ ]. If  $\hat{\mathbf{L}}$  is not diagonalizable [155], the analysis can be carried out by us-

ing the Jordan canonical form of this matrix to replace diagonalization by block-diagonalization.

In both cases the cluster synchronous state is stable if  $\Lambda(\sigma\hat{v}_i) < 0$  for  $i = 2, \dots, m$ , where  $\Lambda$  is the largest Lyapunov exponent of Eq. 5.16 and  $\hat{v}_2, \dots, \hat{v}_m$  represent the transverse modes; the maximum transverse Lyapunov exponent (MTLE) determining the stability of the synchronous state is  $\max_i \Lambda(\sigma\hat{v}_i)$ . Moreover, for the large class of oscillator networks for which the stability region is bounded [156, 157], as assumed here, the synchronizability of the symmetry cluster can be quantified in terms of the eigenratio  $R = \text{Re}(\tilde{v}_m)/\text{Re}(\tilde{v}_2)$ : the smaller this ratio, in general the larger the range of  $\sigma$  over which the cluster synchronous state can be stable<sup>1</sup>. The cluster subnetwork is most synchronizable when  $\hat{v}_2 = \dots = \hat{v}_m$ , which also implies that all eigenvalues are real and in fact integers if the network is unweighted as considered here [158]. It is important to notice that the optimality of the cluster subnetwork and associated properties are conserved in the sense that if  $\tilde{v}_2 = \dots = \tilde{v}_m$  for the isolated cluster, then  $\hat{v}_2 = \dots = \hat{v}_m$  will hold for the cluster as part of a larger network. This analysis also holds for discrete-time systems, such as the ones we consider below.

Now we can compare symmetry clusters with optimal clusters and show rigorously that almost all optimally synchronizable clusters are asymmetric. Without loss of generality, we consider an unweighted cluster in isolation and assume it has

---

<sup>1</sup>Synchronizability may depend on the imaginary part of the eigenvalues in some specific cases, but this does not influence our results since we focus on networks that are initially undirected and the optimization of symmetry clusters necessarily results in real eigenvalues.

$m$  nodes and  $\ell$  directed links internal to the cluster. In a symmetry cluster, because the nodes are structurally identical, the in- and out-degrees of all nodes must be equal. Thus,  $\ell$  must be divisible by  $m$  if the cluster is symmetric. In an optimal cluster, because  $\tilde{v}_2 = \dots = \tilde{v}_m \equiv \tilde{v}$  and thus  $\text{tr}(\tilde{\mathbf{L}}) = (m-1)\tilde{v}$ , it follows that  $\tilde{v} = \ell/(m-1)$ . The fact that  $\tilde{v}$  is an integer implies that  $\ell$  must be divisible by  $m-1$  if the cluster is optimal. Since  $\ell \leq m(m-1)$ , the two divisibility conditions can be satisfied simultaneously if and only if  $\ell = m(m-1)$  (i.e., when the network is a complete graph). But there are numerous optimal networks (and hence clusters) for  $\ell < m(m-1)$  [155, 158]. Therefore, for any given number  $m$  of nodes, all optimal clusters other than the complete graph are necessarily asymmetric, meaning that (with the exception of the complete graph) the synchronization stability of any symmetry cluster can be improved by breaking its structural symmetry. This general conclusion forms the basis of structural AISync and holds, in particular, when an entire network consists of a single symmetry cluster (as illustrated below).

When viewed as isolated subnetworks, symmetry clusters are equivalent to the vertex-transitive digraphs in algebraic graph theory, defined as directed graphs in which every pair of nodes is equivalent under some node permutation [159, 160]. Thus, in order to improve the stability of any non-intertwined symmetry cluster from an arbitrary network, we only need to optimize the corresponding vertex-transitive digraph by manipulating its (internal) links. In particular, this can always be done by removing links inside the symmetry cluster [158, 161], despite the fact that sparser networks are usually harder to synchronize. For concreteness, we focus on clusters that are initially undirected and consider the selective removal of individual











symmetry					
clusters	$C_6$	$Y_6 = K_2 \times C_6$	$K_{3,3}$	$K_{2,2,2}$	$K_6$
nontrivial eigenvalues	1,1,3,3,4	2,3,3,5,5	3,3,3,3,6	4,4,4,6,6	6,6,6,6,6
eigenratio	4	2.5	2	1.5	1
optimal clusters					
nontrivial eigenvalues	1,1,1,1,1	2,2,2,2,2	1,1,1,1,1	3,3,3,3,3	6,6,6,6,6

Table 5.1: Connected symmetry clusters of 6 nodes and optimal clusters embedded within them. Some symmetry clusters have more than one embedded optimal network, in which case we show one that can be obtained through a minimal number of link deletions.



directional links. As an example, we show in Table 5.1 all connected (undirected) symmetry clusters of 6 nodes and their embedded optimal networks. Apart from the complete graph, which is already optimal to begin with, the synchronizability of the other symmetry clusters as measured by the eigenratio  $R$  is significantly improved in all cases.

Because it can be costly or unnecessary to fully optimize a symmetry cluster, it is natural to ask whether the synchronizability can be significantly improved by modifying just a few links. We systematically investigate how many directional links need to be rewired to reduce the eigenratio gap  $R - 1$  by half. Fig. 5.9 summarizes results for all undirected symmetry clusters of sizes between  $m = 8$  and 17, where the rewiring percentage  $p = h/\ell$  is the ratio between the minimal number of link rewirings  $h$  that halves  $R - 1$  and the total number  $\ell$  of internal directed links of the cluster. Fig 5.9(a) shows that on average only about 15% of the links need to be rewired to significantly improve synchronizability of symmetry clusters, and it is largely size independent. This illustrates the potential of structural AISync as a mechanism for the topological control of synchronization stability. Simulated annealing code to improve a symmetry cluster's synchronizability is available at [162].

Fig. 5.9(b) shows the rewiring percentage  $p$  as function of the eigenratio  $R$  and link density  $D = \frac{\ell}{m(m-1)}$ , where each data point represents one symmetry cluster. It is clear that clusters that are small in both  $D$  and  $R$  require the highest percentage of links to be rewired in order to significantly reduce the eigenratio gap. This confirms the intuition that if a network achieves a small eigenratio with a relatively small number of links, then its organization is efficient and its synchronizability is

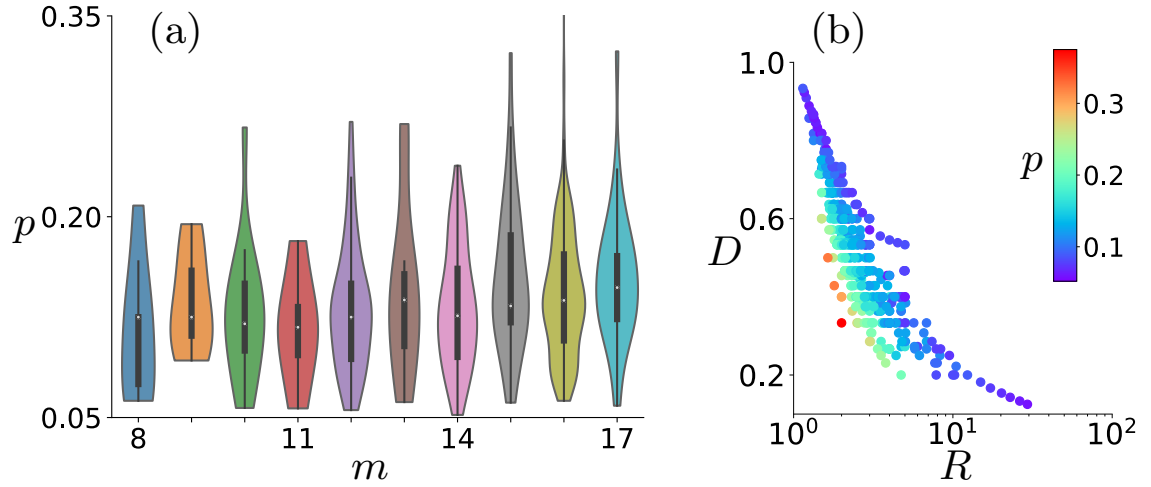


Figure 5.9: Improvement of synchronizability by breaking the cluster symmetry through link rewiring. (a) Percentage of rewirings,  $p$ , needed to reduce the eigenratio gap  $R - 1$  by half. The violin plots show the distribution of  $p$  over all connected undirected symmetry clusters for each cluster size  $m$ . (b) Color-coded  $p$  in the diagram of link density  $D$  versus eigenratio  $R$  for all symmetry clusters considered in panel (a), where each cluster corresponds to one data point.

relatively hard to improve.

### 5.6.3 Experimental results

Having established a theoretical foundation, we now turn to our experimental results. The experiments are performed using the network setup described in Chapter 4. The system can be modeled as

$$x_i[n] = \beta I(x_i[n]) + \sigma \sum_{j=1}^n L_{ij} I(x_j[n]) \mod 2\pi, \quad (5.17)$$

where  $n$  is discrete time,  $\beta$  is the feedback strength,  $I(x_i) = \sin^2(x_i + \phi_0)$  is the normalized intensity output of the modulator,  $x_i$  is the normalized voltage applied to the modulator, and  $\phi_0$  is the operating point (set to  $\pi/4$  in our experiments).

We first consider a case in which the full network is symmetric and we seek to optimize global synchronization. In Fig. 5.10 we study a 16-node symmetric network and show explicitly through our experiments that it becomes more synchronizable with less symmetry. In the original network [Fig. 5.10(a)], all nodes play exactly the same structural role. After seven directional link rewirings [marked in Fig. 5.10(b)], the symmetry of the network is largely broken and almost all nodes are now structurally different: the original 16-node symmetry cluster is reduced to 14 single-node clusters and only 2 nodes occupying symmetric positions. The eigenratio, however, reduces from  $R = 4.62$  to  $R = 2.80$  and thus improves significantly.

The experimental results are presented in Fig. 5.10(c), where we show the average synchronization error as a function of the coupling strength for both networks. For each network, we performed 8 runs of the experiment starting from

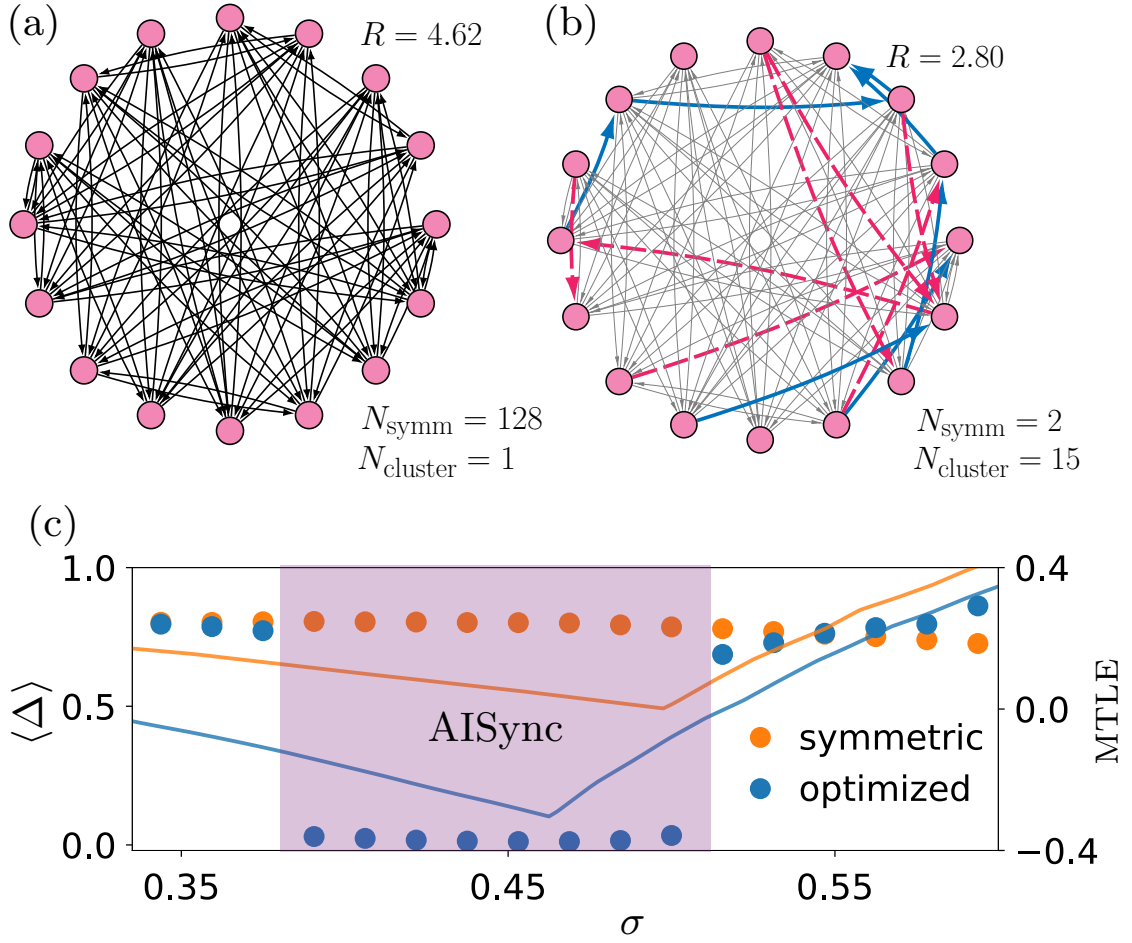


Figure 5.10: Experimental demonstration of structural AISync in global synchronization. (a) Symmetric 16-node network of coupled optoelectronic oscillators used in the experiment, which has 128 symmetries, consists of a single (global) symmetry cluster, and has an eigenratio of  $R = 4.62$ . (b) Optimized network found through simulated annealing, where 7 links are removed (red) and 7 links are added (blue) to the network in (a), resulting in a network with only 2 symmetries and 15 symmetry clusters but an eigenratio of  $R = 2.80$ . The adjacency matrices can be found in Appendix C. (c) Experimentally measured average synchronization error  $\langle \Delta \rangle$  for the symmetric and symmetry-broken network in (a) and (b), respectively, where only the latter can synchronize for a range of  $\sigma$  (purple region). The feedback strength was set to  $\beta = 2.8$ . This experimental result is consistent with the theoretically computed MTLE (color-coded curves), which is more negative for the symmetry-broken network.

different random initial conditions, and measured the normalized voltages  $x_i$  for 8196 iterations at each fixed coupling strength before increasing  $\sigma$  by 0.015. The synchronization error is defined as  $\Delta = \sqrt{\sum_{1 \leq i \leq N} \|x_i - \bar{x}\|^2 / N}$ , where  $\| \cdot \|$  is the shortest distance between two points on a circle, and the mean  $\bar{x}$  on a circle can be found as the phase of  $\sum_{1 \leq j \leq N} e^{ix_j}$ . The data points in Fig. 5.10(c) correspond to the average synchronization error  $\langle \Delta \rangle$ , defined as  $\Delta$  averaged over the last 5000 iterations for each  $\sigma$  and then further averaged over the 8 runs. The error bars corresponding to the standard deviation across different runs are smaller than the size of the symbols, and are not shown. The experimental data clearly demonstrates that synchronization is only achieved for the network with reduced symmetry.

The experimental result is consistent with the MTLE determined from numerical calculations of the variational equation of the model in Eq. 5.17 [color-coded curves in Fig. 5.10(c)]. Indeed, for values of  $\sigma$  close to the boundary of linear stability, synchronization is not observed in experiments due to noise [123], but synchronization is consistently observed once the MTLE becomes sufficiently negative.

Now we consider the case in which the symmetry cluster is a proper subnetwork (i.e.,  $m < N$ ) and the goal is to optimize synchronization in that cluster. Again using optoelectronic oscillators, we perform experiments in the network configuration shown in Fig. 5.11(a), which is a complex network with five symmetry clusters. This time we optimize a symmetry cluster that, when isolated, corresponds to a ring network. The MTLE calculation in Fig. 5.11(b) predicts AISync to be common in the parameter space. Fixing  $\beta = 6$ , the same experimental procedure as for Fig. 5.10 is performed to find the average synchronization error  $\langle \Delta \rangle$  in the

cluster. As shown in Fig. 5.11(c), AISync is consistently observed over a wide range of the coupling strength  $\sigma$ , matching the theoretical prediction.

#### 5.6.4 Extension to intertwined clusters

Now we turn to the case of intertwined clusters. When two clusters are intertwined, desynchronization in one cluster will in general lead to the loss of synchrony in the other cluster (an example would be the Red and Green clusters in Fig. 5.6a). This is because the symmetry group acting on the two clusters does not admit a geometric decomposition; that is, symmetry permutations cannot be applied to each cluster independently. As a consequence, a desynchronized cluster sends incoherent signals to nodes in the other cluster, causing its intertwined counterpart to desynchronize as well. The irreducible representation transform introduced in Ref. [113] enables stability analysis on many cluster synchronization patterns. In that framework, the presence of intertwined clusters is reflected by nontrivial transverse blocks (i.e., blocks with dimension greater than 1) in the transformed coupling matrix, whereas non-intertwined clusters only give rise to  $1 \times 1$  transverse blocks. Unfortunately, the high-dimensional transverse blocks make the effect of topological perturbation on cluster synchronizability opaque and offer little insight on how to optimize the clusters to support desired synchronization patterns.

Here, we present a new perspective that gives a simple necessary condition for the synchronization in intertwined clusters. This in turn points to an easy extension of the previous optimization strategy for the topological control of synchronization

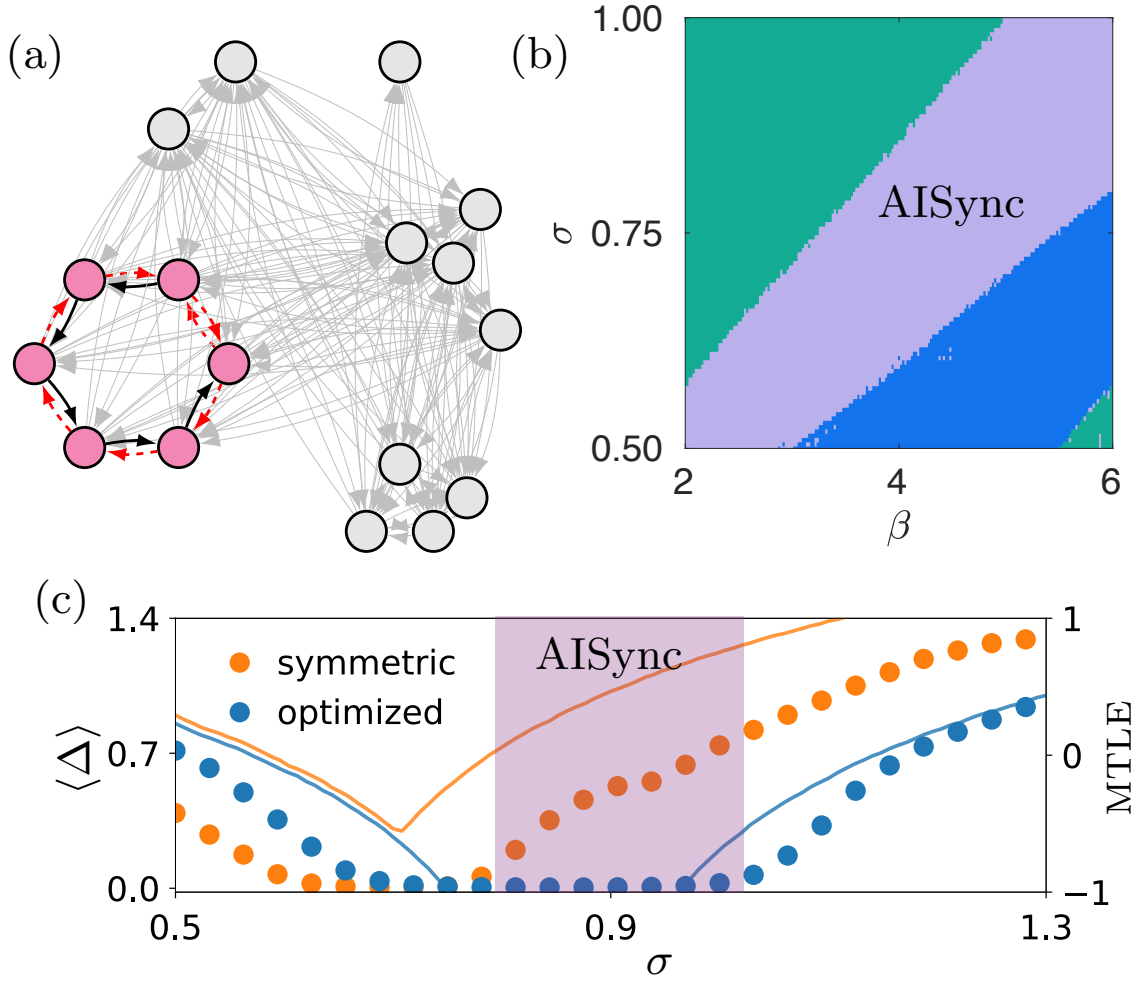


Figure 5.11: Experimental demonstration of structural AISync in cluster synchronization. (a) Example network in which a symmetry cluster (magenta) is optimized for synchronization by removing links (red). The adjacency matrices can be found in Appendix C. (b) Predictions based on the theoretical computation of the MTLE, showing that in the  $\sigma \times \beta$  parameter space there is an AISync region (purple); the other colors indicate the regions where both clusters synchronize (blue) and where neither cluster can synchronize (green). (c) Experimentally measured average synchronization error  $\langle \Delta \rangle$  in the original (orange) and optimized (blue) cluster for  $\beta = 6$ . The experimental results are in good agreement with the MTLE calculations (color-coded curves).

in intertwined clusters.

Consider two intertwined clusters subject to transverse perturbations  $\delta\mathbf{X}$  and  $\delta\mathbf{Y}$ , respectively. Their variational equation reads

$$\begin{cases} \delta\dot{\mathbf{X}} = \left( \mathbf{1}_m \otimes J\mathbf{F}(\mathbf{s}_{I_X}) - \sigma\hat{\mathbf{L}}_X \otimes J\mathbf{H}(\mathbf{s}_{I_X}) \right) \delta\mathbf{X} + \sigma\mathbf{C} \otimes J\mathbf{H}(\mathbf{s}_{I_Y})\delta\mathbf{Y}, \\ \delta\dot{\mathbf{Y}} = \left( \mathbf{1}_m \otimes J\mathbf{F}(\mathbf{s}_{I_Y}) - \sigma\hat{\mathbf{L}}_Y \otimes J\mathbf{H}(\mathbf{s}_{I_Y}) \right) \delta\mathbf{Y} + \sigma\mathbf{D} \otimes J\mathbf{H}(\mathbf{s}_{I_X})\delta\mathbf{X}. \end{cases} \quad (5.18)$$

Here,  $\mathbf{C}_{ij} = 1$  if the  $i$ -th oscillator in cluster  $\mathbf{X}$  receives an input from the  $j$ -th oscillator in cluster  $\mathbf{Y}$  and  $\mathbf{C}_{ij} = 0$  otherwise. The inter-cluster coupling matrix  $\mathbf{D}$  is similarly defined with the role of two clusters exchanged ( $\mathbf{D} = \mathbf{C}^\top$  if the inter-cluster coupling is undirected). Without the cross-coupling term, Eq. 5.18 reduces to the non-intertwined case discussed earlier

$$\begin{cases} \delta\dot{\mathbf{X}} = \left( \mathbf{1}_m \otimes J\mathbf{F}(\mathbf{s}_{I_X}) - \sigma\hat{\mathbf{L}}_X \otimes J\mathbf{H}(\mathbf{s}_{I_X}) \right) \delta\mathbf{X}, \\ \delta\dot{\mathbf{Y}} = \left( \mathbf{1}_m \otimes J\mathbf{F}(\mathbf{s}_{I_Y}) - \sigma\hat{\mathbf{L}}_Y \otimes J\mathbf{H}(\mathbf{s}_{I_Y}) \right) \delta\mathbf{Y}. \end{cases} \quad (5.19)$$

Because of the intertwined nature of the two clusters, they must be considered concurrently when synchronization is desired in either of them. That is,  $\hat{\mathbf{L}}_X$  and  $\hat{\mathbf{L}}_Y$  should be optimized to ensure  $\delta\mathbf{X}$  and  $\delta\mathbf{Y}$  are both damped to zero in Eq. 5.18. It is difficult to establish a synchronizability measure on two clusters based on Eq. 5.18, but we can see the following connection between Eqs. 5.18 and 5.19<sup>2</sup>:

If  $|\delta\mathbf{X}| \rightarrow 0$  and  $|\delta\mathbf{Y}| \rightarrow 0$  in Eq. 5.18, then

$$|\delta\mathbf{X}| \rightarrow 0 \text{ and } |\delta\mathbf{Y}| \rightarrow 0 \text{ in Eq. 5.19.} \quad (5.20)$$

---

<sup>2</sup>This is easy to see. For example, if  $\delta\mathbf{X}$  does not shrink according to Eq. 5.15, then in order for  $|\delta\mathbf{X}| \rightarrow 0$  in Eq. 5.18  $\delta\mathbf{Y}$  must be significantly away from zero.



That is,  $\delta\mathbf{X}$  and  $\delta\mathbf{Y}$  must go to zero for the synchronization in intertwined clusters. This implies that we can optimize the synchronizability of two clusters independently, using the same method introduced earlier for non-intertwined clusters. Such optimization is guaranteed to expand the region in parameter space satisfying condition (5.20). Inside this expanded region, one is likely to observe structural AISync.

We demonstrate this by considering a random network of 17 nodes that contains two intertwined clusters (highlighted in Fig. 5.12(a)). Each cluster is optimized by removing the red dashed links, which breaks its structural symmetry but improves the eigenratio to 1. The pale orange shade in Fig. 5.12(b) indicates the region where condition (5.20) is satisfied by the original clusters. This region is expanded to include the purple area when the clusters are optimized. Not all points in the expanded region show structural AISync, since the condition is necessary but not sufficient. Nonetheless, a significant portion of it does, which is verified by direct simulations and highlighted in dark purple.

We test these predictions in our experiment. We vary  $\beta$  and  $\sigma$  along the dashed line in Fig. 5.12(b). The experimentally measured synchronization error in each cluster is shown in Fig. 5.12(c). Our results confirm that AISync can also be observed in networks with intertwined clusters.

In summary, we established the role of structural asymmetry (or structural heterogeneity) in promoting spontaneous synchronization through both theory and experiments. Our theory confirmed the generality of the phenomenon, while our experiments demonstrated its robustness. Because symmetry clusters arise naturally in complex networks, our findings are applicable to a wide range of coupled

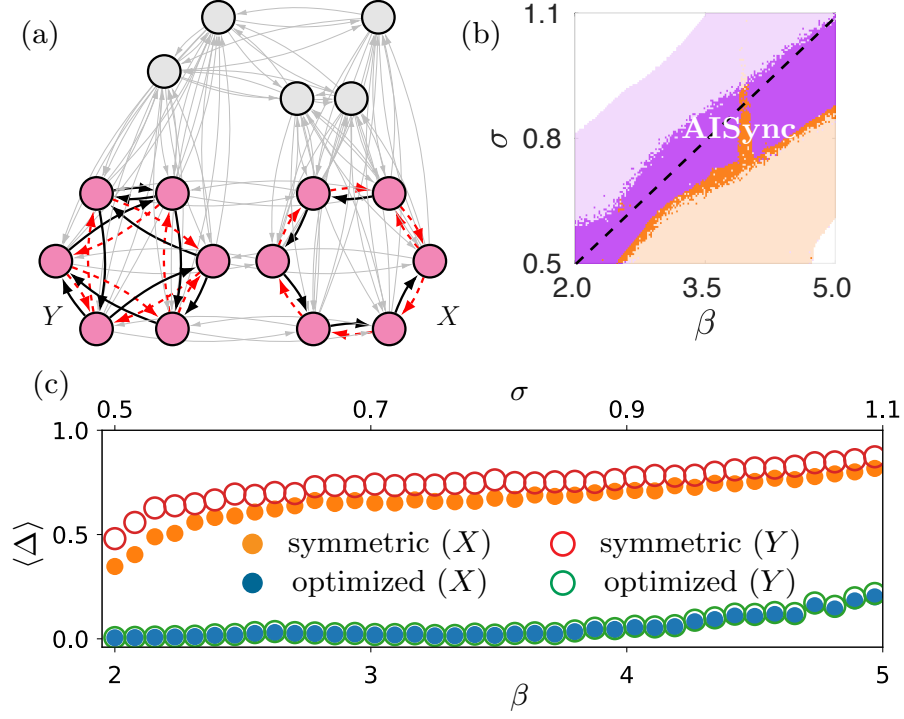


Figure 5.12: Structural AISync in intertwined clusters. (a) Example network in which two intertwined clusters (magenta) are optimized to induce synchronization by removing red links. The adjacency matrices are included in Appendix C. (b) The region in the  $\sigma \times \beta$  parameter space satisfying condition (5.20) is expanded from orange shaded area to include purple shaded area when the clusters are optimized. The dark purple area corresponds to the AISync region determined through direct simulations. (c) Experimentally measured synchronization error  $\Delta$  in the original and optimized clusters when moving through the parameter space along the dashed line in (b).

dynamical systems. In particular, since identical synchronization in a symmetry cluster is the basic building block of more complex synchronization patterns, our results can be used for the *targeted* topological control of cluster synchronization in complex networks, which echoes the positive effect of structural asymmetry on input control [163].

## 5.7 Discussion

In this chapter, we presented a series of experiments performed on the time-multiplexed opto-electronic network system described in Chapter 4. We began by studying the dynamics of a single uncoupled oscillator and comparing it with the model of a sine-squared map model. We then investigated synchronization in a the simplest possible “network” of two bidirectionally coupled nodes and found excellent agreement with our model.

Having validated our simple models, we turned to the implementation of more interesting networks. First, we observed synchronized chimeras in a network of five globally-coupled nodes, confirming our predictions from Chapter 3 that these synchronized chimeras can be observed in larger globally-coupled networks and with different oscillator dynamics. Interestingly, we again found, as we did in Chapter 3, that a coupling delay was necessary to find stable synchronized chimeras.

Next, the implementation of a 10-node network allowed us to perform the first experimental measurements of input-cluster synchronization (as opposed to symmetry-cluster synchronization) in a non-Laplacian network. Our measurements

confirmed that the symmetry-based stability analysis for cluster synchronization can be extended to work for input-clusters.

The stability analysis for cluster synchronization depends on network symmetries. However, we used our experiment to study a counter-intuitive phenomenon termed structural Asymmetry-Induced Synchronization (AISync), in which breaking the symmetry within a cluster generally improves the cluster's synchronizability in Laplacian networks.

In addition to being interesting studies in their own right, the experiments presented in this chapter also demonstrate the versatility and utility of our novel approach for creating arbitrary networks in the lab.

## Chapter 6: Evaluating physical random number generators

Entropy isn't what it used to be.

---

BUMPER STICKER

Anyone who considers arithmetical  
methods of producing random  
digits is, of course, in a state of sin.

---

JOHN VON NEUMANN [[164](#)]

*This chapter is based on work from the following publication:*

*Joseph D Hart, Yuta Terashima, Atsushi Uchida, Gerald B Baumgartner, Thomas E Murphy, & Rajarshi Roy. “Recommendations and illustrations for the evaluation of photonic random number generators”. APL Photonics, 2(9), 090901 (2017).*

Random number generation underlies modern cryptographic techniques used to ensure the privacy of digital communication and storage. In order to improve security, digital information systems have begun to utilize optical or other phys-

ical sources to generate high-speed unpredictable signals. However, the methods most commonly used to evaluate random number generators (RNGs) have not yet evolved to reflect the increasing importance of physical entropy sources for modern cryptography.

Historically, random number generation has been dominated by algorithms that, given a seed, produce a sequence of pseudo-random numbers. However, since pseudo-random number generators (PRNGs) are deterministic algorithms, if an attacker is able to determine the seed, all security is lost. In order to defend against such problems, RNG designers are increasingly turning to physical means to either frequently re-seed or completely replace PRNGs, as evidenced by the commercialization of optical RNGs by companies such as PicoQuant [165], IDQuantique [166], and Whitewood Encryption [167].

Unlike PRNGs, physical processes can generate true randomness. Classical stochastic processes such as thermal or electrical noise can be used for entropy generation [168]. Additionally, boolean chaos [169, 170] and timing jitter in ring oscillators [171, 172] have been used to create electronic entropy sources. However, optical systems are especially well-suited for random number generation due to resistance to external interference, speed, and access to quantum mechanical processes. Therefore, even though our analysis and recommendations are relevant for all physical entropy sources, we focus specifically on optical systems.

The fundamental randomness of quantum mechanics present in many optical systems can be employed to generate true random numbers. In some optical entropy sources such as single photon measurements [166, 173–176], optical para-

metric oscillators [177], and spontaneous Raman scattering [178], the measurements themselves are quantized. In others, such as those based on amplified spontaneous emission [167, 179–185], laser phase noise [186–191], quantum vacuum fluctuations [192–196], and stimulated Raman scattering [197, 198], an unpredictable analog waveform with quantum mechanical origins is sampled and digitized. In this chapter, we will provide an in-depth analysis of one of each type: single photon time-of-arrival measurements and amplified spontaneous emission. Optical RNGs based on photon detection [165, 166] and spontaneous emission [167] are now commercially available. See Ref. [199] for a review of stochastic RNGs based on these and other optical systems.

Chaotic systems amplify uncertainties in initial conditions and sources of intrinsic noise [200, 201]; only in the last decade has this inherent unpredictability been harnessed for random number generation in the form of chaotic lasers [17, 183, 202–209]. For a review of chaotic lasers including their applications to RNGs, see Ref. [210] and [211]. While we know of no commercially available physical RNGs based on chaotic lasers, new developments in photonic integrated circuits [209] and real-time, high-speed bit streaming [212] for chaotic laser RNGs lay the groundwork for commercialization.

Physical sources of randomness and PRNGs are best used in complementary roles. Physical sources can provide true randomness, but the raw output of a physical source is typically biased and not uniformly distributed. PRNGs, on the other hand, can provide a binary sequence that is unbiased and uniformly distributed but completely deterministic. The most secure RNGs combine the benefits of both meth-

ods by using physical sources to seed PRNGs or other post-processing algorithms; such implementations are used by the Intel Secure Key (the RDRAND command), available in Ivy Bridge processors [168], and in the commercially available optical RNGs provided by PicoQuant [165], IDQuantique [166], and Whitewood Encryption [167]. Official guidelines for how to combine a physical entropy source and a PRNG are currently under development by the U.S. National Institute of Standards and Technology (NIST) [213].

Due to the increasing importance of the security of digital information and the wide variety of physical methods used to generate random numbers, a standard set of evaluation metrics for random number generation is essential. Previous works have used a variety of methods to estimate the entropy of physical sources of randomness [173–176, 184, 188, 190–192, 204–206, 208, 209, 214]; however, many of these techniques assume that there are no inter-sample correlations. As of this writing, there is no widely accepted technique to estimate the entropy of physical entropy sources. It is important that evaluation metrics and standards reflect the fundamental differences between PRNGs and physical entropy sources; however, common testing practices do not currently distinguish between the two. We are not the first to recognize these problems; indeed, NIST is currently developing a new set of standards and evaluation techniques specifically for physical entropy sources [26]. These new standards recommend entropy rate as the figure of merit for physical entropy sources.

In this chapter, we review the current practices in the evaluation of physical RNGs and call for a renewed emphasis on understanding the origin of and physical



and information theoretical limitations on the randomness in the design, testing, and validation of optical entropy sources. We advocate for the separation of the physical entropy source from deterministic post-processing in the evaluation process and for the use of the  $h(\epsilon, \tau)$  entropy rate analysis [58, 215]. The  $h(\epsilon, \tau)$  entropy rate analysis emphasizes that the entropy rate is a function of measurement resolution  $\epsilon$  and sampling period  $\tau$ . While the existing statistical tests used for physical RNG evaluation offer a simple “pass or fail” evaluation, the  $h(\epsilon, \tau)$  analysis provides insight that is more relevant for the design of optical RNGs, including information about the physical origins of randomness and the impact of the digitization process on entropy extraction. Finally, we use the  $h(\epsilon, \tau)$  analysis to compare three state-of-the-art optical entropy sources: single photon counting, chaotic lasers, and amplified spontaneous emission noise.

In Section 6.1, we review current physical RNG evaluation practices and present our recommendations. In particular, Section 6.1.1 describes the current standards for physical RNG evaluation and some of its shortcomings. In Section 6.1.2 we briefly describe the new NIST draft recommendations for the evaluation of physical random number generators, and we provide our own recommendations in Section 6.1.3. In Section 6.2, we review three different methods of optical entropy generation and present the results of our own measurement and evaluation of these entropy sources. We provide some concluding thoughts in Section 6.3.

## 6.1 Evaluation of Physical Random Number Generators

Evaluating a PRNG is relatively straightforward: NIST has published specific guidelines for the design and testing of PRNGs [216]. In contrast, physical RNGs are much more difficult to evaluate due in part to the wide variety of physical processes that can be used [26].

### 6.1.1 State of the art

The most common procedure used to evaluate physical RNGs, depicted in Fig. 6.1a, is the following: collect data from the physical system; perform deterministic post-processing (such as von Neumann’s method) and/or conditioning (such as least significant bit extraction [203], exclusive or (XOR) [17], and hashing [173]) on the data in order to remove bias and whiten the output; and use a suite of statistical tests (such as the NIST suite [217] or DIEHARD [218]) on the output bit sequence to determine whether the resulting bitstream has statistical properties that are compatible with a truly random sequence. The distinction between post-processing and conditioning is defined somewhat arbitrarily by NIST [26], as discussed in the next section. The rate of random bit generation claimed is typically the highest possible rate such that the output bit sequence can pass the suite of statistical tests. As we will discuss, this method of evaluating physical RNGs has significant shortcomings.

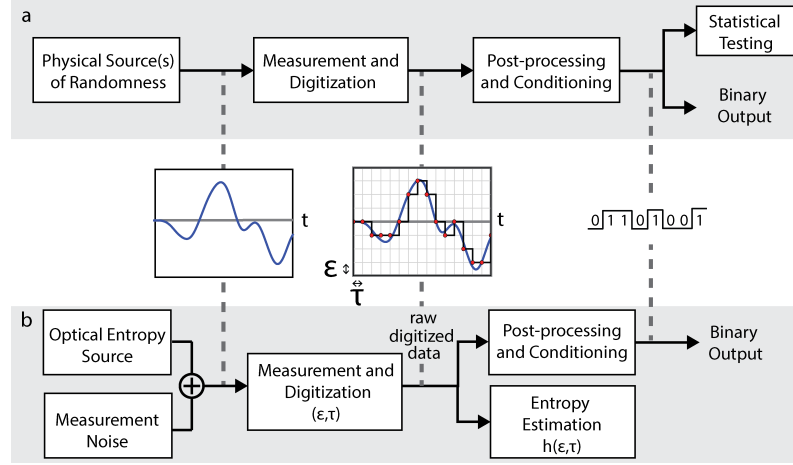


Figure 6.1: Methods of optical RNG testing and evaluation. a) Current practice involves performing statistical tests on the final, post-processed output bit sequence. b) Our recommendation is to use the raw digitized data to make an estimate of the rate at which one is entitled to harvest entropy, then use appropriate post-processing to extract that entropy from the digitized data. The measurement parameters  $\epsilon$  and  $\tau$  and the post-processing method should be carefully chosen such that entropy is extracted from the desired physical entropy source rather than from measurement noise. The sample time series show an analog signal, a digitized signal with measurement parameters  $\epsilon$  and  $\tau$ , and a post-processed output bit sequence.

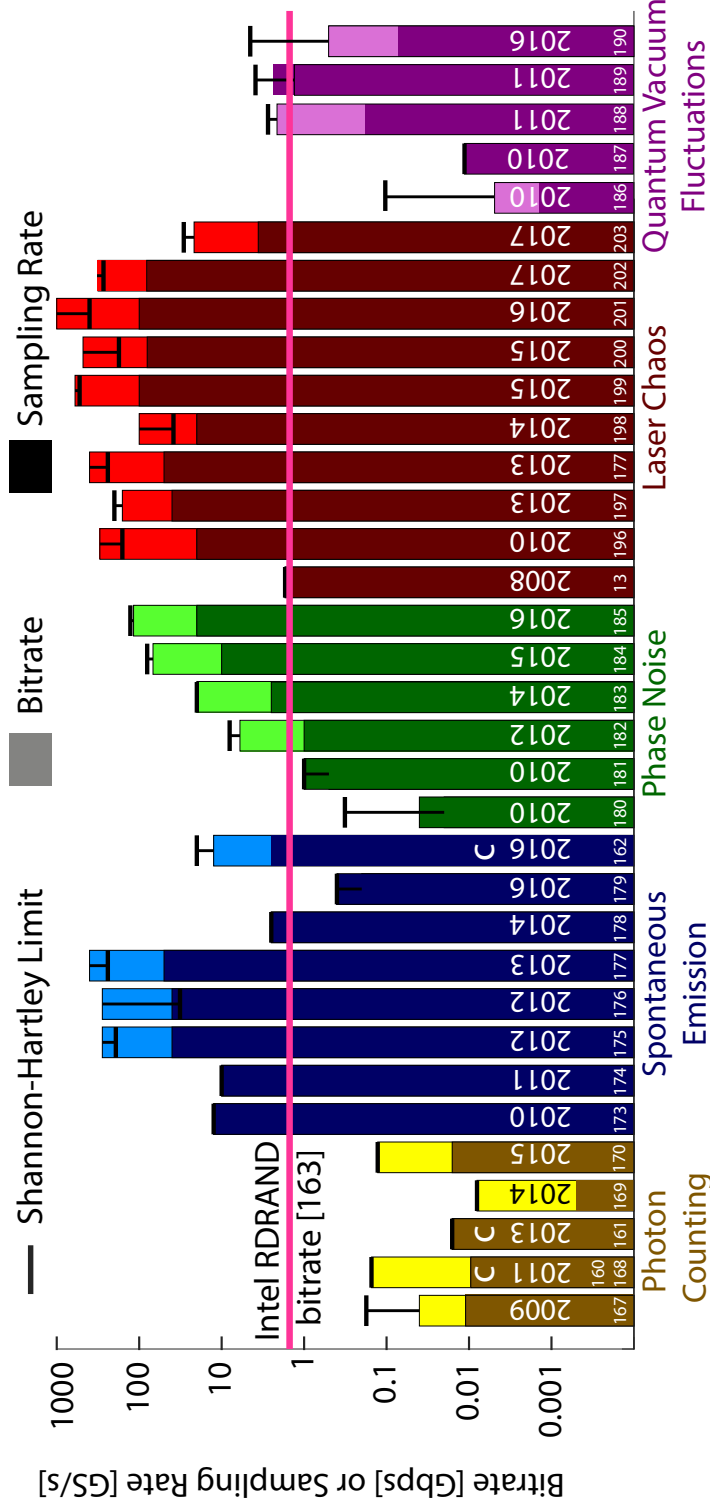


Figure 6.2: Survey of recent optical random number generation rates. The darker vertical bars indicate the sampling rate used and the lighter bars indicate the claimed random bit rate. The number of the reference is written on each bar. The black bars indicate the Shannon-Hartley limit given by Eq. 6.1. Commercial products are denoted by a C.

Figure 6.2 provides a survey of some recently published results using five common optical techniques for random number generation. The bit rates given are those claimed by the authors, and the sampling rate is the number of samples per second measured by the digitizer. In many cases the claimed bit rate is higher than the sampling rate; this is a result of the digitizer obtaining more than 1 bit per sample (e.g., by using an 8-bit analog to digital converter).

An information theoretical upper bound on the entropy rate is given by the Shannon-Hartley limit [219]

$$h_{SH} = 2BW \cdot N_{\epsilon}, \quad (6.1)$$

where  $N_{\epsilon}$  is the number of bits per sample that the digitizer measures at a given measurement resolution  $\epsilon$ , and  $BW$  is the bandwidth of the signal measured by the digitizer.  $BW$  is limited by the analog bandwidth of the physical entropy source as well as the detectors and digitizer. Eq. 6.1 gives the maximum rate at which information can be obtained from the signal by the digitizer [219]. Of course, Eq. 6.1 overestimates the upper bound because the effective bandwidth is often less than the standard signal bandwidth [220] and the effective number of bits of a digitizer is often less than the stated number of bits [221].

Figure 6.2 reveals a significant shortcoming in the current practice of quantifying optical random number generation: several of the recently reported RNG systems, while producing data that passes all of the existing statistical tests, achieve a rate that exceeds even our overestimate of the Shannon-Hartley limit (horizontal black bars). In most of the violating cases, the sampling rate is higher than the

Nyquist rate ( $2BW$ ) of the signal, resulting in strong inter-sample correlations. Post-processing may obscure these correlations from statistical tests; however, because post-processing is deterministic it cannot increase the rate of entropy production.

When one considers that many PRNGs pass the statistical tests, it is not surprising that those statistical tests can be passed by post-processing the output of a physical system, even if that output is not random. While statistical tests can provide some assurance of statistical uniformity, they provide no guarantee that there is no underlying pattern that could later be discovered. Therefore, statistical tests are perhaps best viewed as a sanity check against blatant errors, rather than a proof of randomness.

Statistical test suites are also limited to a simple “pass or fail” evaluation that provides little insight into the physical processes generating the random numbers. Choosing a physical process that can be theoretically justified as random and then showing that the measured entropy is actually coming from that random process provides much greater assurance of unpredictability than can simply putting a sequence of bits through statistical tests that any good PRNG will pass.

While there is nothing wrong with having the sole aim of passing statistical tests, physical RNGs that do this while violating information theory limits are perhaps better called physical-based PRNGs, rather than true physical RNGs, as suggested in refs. [\[214\]](#) and [\[222\]](#).

The upper bound on entropy harvesting provided by Eq. [6.1](#) assumes that the probability density function (PDF) is uniform, the maximal entropy distribution. For small  $\epsilon$  (high resolution), one can find a more stringent upper bound on the

entropy rate by accounting for the fact that PDFs found in physical systems are generally not uniform [215]:

$$h_0 = \min(\tau^{-1}, 2BW)(N_\epsilon - D_{KL}(p(x)||u(x))), \quad (6.2)$$

where  $p(x)$  is the PDF of the physical entropy source,  $u(x)$  is the uniform distribution over the interval of  $x$ -values for which  $p(x)$  is non-zero, and  $D_{KL}(p(x)||u(x))$  denotes the relative entropy or Kullback-Leibler divergence [223] of  $u(x)$  from  $p(x)$ , and  $\tau^{-1}$  is the sampling rate; according to the Nyquist theorem, one cannot obtain more entropy by sampling faster than  $2BW$  [219]. Eq. 6.2 simply says that the maximum obtainable entropy rate is the maximum sampling rate times the average entropy per sample. For further discussion on the  $h_0$  limit, see Appendix D.

We emphasize that Eq. 6.2 is valid only for fine measurement resolution; in the case of thresholding ( $N_\epsilon=1$  bit), one can always obtain  $h(N_\epsilon) = \tau^{-1}$  bits/s for any independent, identically distributed (IID) random process by setting the threshold at the median. The sampling rate for thresholding is also limited by the Nyquist rate  $2BW$ .

Figure 6.3 illustrates the relationships between these information theoretical limitations on obtainable entropy rates. The limits in Eq. 6.2 are information theoretical limits that depend on the specifications of the measurement apparatus and on the bandwidth and PDF of the physical system; an additional physical limit, the Kolmogorov-Sinai (metric) entropy rate, exists for dynamical (chaotic) systems. We discuss the Kolmogorov-Sinai entropy rate in sections 6.1.3.1 and 6.2.2.

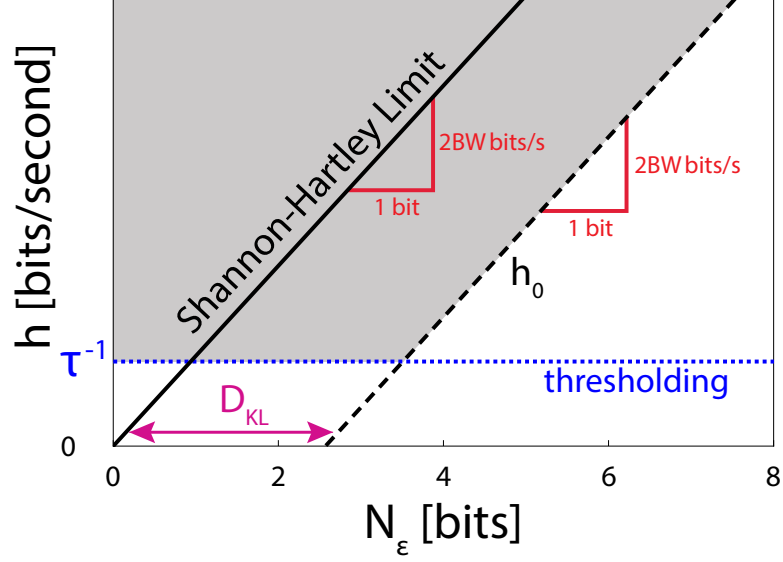


Figure 6.3: Illustration of information theoretical limits for physical RNG. The Shannon-Hartley limit (solid black line, Eq. 6.1) is the theoretical upper limit for rate of information transfer for a system with a given bandwidth.  $h_0$  (dashed black line, Eq. 6.2) is a correction to the Shannon-Hartley limit obtained by accounting for non-uniformity of the PDF of the physical process generating the entropy. The Shannon-Hartley limit has a slope of  $\tau_{max}^{-1} = 2BW$  bits per second per bit of resolution, while the slope of  $h_0$  is  $\min(\tau^{-1}, 2BW)$  bits per second per bit of resolution. Most RNG designers want to push the limit of random bit generation, so here we show the slope of  $h_0$  as the maximum of  $2BW$  bits per second per bit of resolution. The x-intercept of  $h_0$  is given by the Kullback-Leibler divergence of the uniform distribution from the experimental probability distribution. The  $h_0$  limit is only valid for fine resolution; for IID systems an entropy rate of  $\tau^{-1}$  bits per second can always be obtained by thresholding ( $N_\epsilon = 1$ , blue dotted line). The sampling rate for thresholding is also limited by the Nyquist rate  $2BW$ . The region of entropy rates that is unobtainable according to the limits provided by Eqs. 6.1 and 6.2 is indicated by the gray shading.



### 6.1.2 2016 NIST Draft Recommendations

The 2016 NIST Draft Recommendation for the Entropy Sources Used for Random Bit Generation [26] tries to resolve some of these problems by separating out the algorithmic, pseudo-random parts of random number generation from the physical entropy source. It also gives recommendations on how to combine the pseudo-random algorithm and the entropy source once they have been separately validated [213]. The NIST draft recommendation additionally requires a justification of how the entropy source works and why it produces acceptable entropy. While the draft recommendation has been replaced by the final 2018 NIST Recommendation for the Entropy Sources Used for Random Bit Generation [25], we discuss it here in order to place our work (which was published in 2017) in context. For a discussion of the final 2018 standards, see Section 6.4.

The NIST draft standards are based on an entropy source model similar to the one shown in Fig. 6.1b; The only difference is that it does allow for some simple post-processing techniques to be applied to the raw digitized data before estimation of the entropy rate. NIST distinguishes post-processing (only von Neumann’s method, linear filtering method, or length-of-runs method) from conditioning (such as some hash functions), which has fewer restrictions and is not allowed to increase the entropy estimate.

For entropy sources that are potentially not IID, the entropy estimation procedure is quite simple. Run two suites of tests on the (post-processed but not conditioned) data. The first suite of tests estimates the min-entropy [224] per sam-

ple in the data set; the min-entropy is designed to provide a conservative estimate of the entropy. The second suite of tests is a pass/fail set of so-called “Restart Tests”: These ensure that the entropy source does not behave the same way each time it is restarted. We will discuss only the entropy estimation suite. If a conditioning procedure is used, one should adjust down the min-entropy estimate if appropriate, as described in the NIST recommendation [26]. The resulting estimate of min-entropy per sample gives the upper limit on the rate at which entropy can be extracted from the source.

The min-entropy estimation suite includes 10 different tests, and the final min-entropy per sample estimate is the minimum of all the estimates. We have run all 10 tests on our data, but since most of them give similar results we discuss only two of the estimates here: the most common value (MCV) estimate and the Markov estimate. The simplest entropy estimate is the MCV estimate. It assumes that samples of the signal are independent and identically distributed and estimates the entropy as  $-\log_2(p_{max})$ , where  $p_{max}$  is the fraction of samples appearing in the most common bin. This is the exact min-entropy for an IID entropy source. The second entropy estimate is the Markov estimate, which takes into account first-order correlations. For a complete description of all the tests, see Ref. [26].

The NIST draft recommendation allows some post-processing before the entropy analysis, but deterministic algorithms cannot increase the entropy rate and serve only to make the entropy estimation process more difficult. Indeed, this was recently shown for the commonly used post-processing technique of least significant bit extraction [214]. It is known that some of the tests in the NIST suite severely

underestimate the min-entropy of entropy sources with normal distributions [225]; this is a real problem because many of the best physical entropy sources have normal or approximately normal PDFs. Thus, the NIST test suite unintentionally encourages designers of these systems to include post-processing before testing, since this is the only way their source can receive a high entropy estimate from the NIST suite. The NIST draft recommendation does not address the details of the digitization process, which has a bandwidth due to the detection apparatus, measurement resolution  $\epsilon$ , and sampling frequency  $\tau^{-1}$  that all impact the rate at which entropy can be harvested from a physical system. Finally, the NIST draft standards do not mention chaotic entropy sources, even though it is well-known that entropy can be harvested from chaotic systems due to their sensitive dependence on initial conditions [17, 210, 226]. In light of these concerns, we provide some additional recommendations to designers and evaluators of physical entropy sources in the next section.

### 6.1.3 Recommendation: A dynamical systems approach to entropy estimation

Both stochastic and chaotic physical systems have been used to generate entropy at high rates. It is therefore important to have techniques that can accurately estimate the entropy from both stochastic and chaotic sources. For this, we recommend a dynamical systems approach to entropy estimation.

Our first recommendation regards the role of post-processing in the evaluation

of RNGs. As we have previously mentioned, deterministic post-processing algorithms are useful for extracting entropy but cannot increase the entropy production rate of the physical source. In order to obtain a more accurate and insightful measure of the suitability of a physical system for random number generation, we recommend estimating the rate of entropy production *directly from the raw digitized data*. Post-processing and conditioning techniques can then be chosen to extract random bits from the physical source at a rate up to the entropy rate. This procedure is depicted in Fig. 6.1b.

As described above, many different physical processes can generate entropy. It is even possible for a single system to have multiple sources of entropy; for example, a chaotic laser entropy source might have entropy from the chaotic dynamics (which amplifies intrinsic quantum mechanical noise from spontaneous emission) and from electronic noise in the detector. We therefore recommend that designers take care in choosing measurement parameters – measurement resolution  $\epsilon$  and sampling period  $\tau$  – and post-processing techniques that extract entropy from the desired physical source. In the rest of this section, we present  $h(\epsilon, \tau)$  as a technique to guide these choices.

#### 6.1.3.1 Noise, chaos, and $h(\epsilon, \tau)$

Gaspard and Wang [215] have shown that  $h(\epsilon, \tau)$  estimated by the Cohen-Procaccia algorithm [227] can be used to directly compare the entropy production of stochastic and chaotic processes.  $h(\epsilon, \tau)$  treats the entropy rate as a function of the

measurement resolution  $\epsilon$  and the sampling rate  $\tau^{-1}$ . Such an analysis can provide insight into the type of physical process (stochastic or chaotic) that is generating the entropy at a given measurement resolution or time scale. For example, in a deterministic chaotic system as the measurement resolution  $N_\epsilon$  increases,  $h(\epsilon, \tau)$  approaches a constant given by the Kolmogorov-Sinai entropy rate; however, in a purely stochastic system,  $h(\epsilon, \tau)$  scales as  $N_\epsilon$  [215]. These predictions were recently verified experimentally [58]. In the context of physical RNGs, the  $h(\epsilon, \tau)$  analysis can guide the choice of the best  $\epsilon$ ,  $\tau$ , and post-processing method to extract entropy from the desired physical source.

The  $h(\epsilon, \tau)$  analysis is not limited to continuous time series; it can also be applied to physical entropy sources based on discrete events such as single photon detection. For example, entropy sources using single photon time-of-arrival measurements [173, 174, 176] can be analyzed by considering the entropy rate as a function of the temporal precision ( $\epsilon$ ) of the measurement of the arrival times and of the maximum count rate ( $\tau^{-1}$ ), as we show in the section 6.2.1

Despite these important advantages, the  $h(\epsilon, \tau)$  analysis has only recently been applied in the context of physical random number generation [58]. The RNG system in Ref. [58] was designed to study the  $(\epsilon, \tau)$  entropy rate estimate on low-dimensional chaotic experimental data and generated only a few hundreds of bits of entropy per second. In the following sections, we use the  $h(\epsilon, \tau)$  analysis to compare three state-of-the-art optical RNG techniques: single photon time-of-arrival measurements, digitization of chaotic laser data, and digitization of amplified spontaneous emission noise data. An  $h(\epsilon, \tau)$  analysis can be performed with any entropy

estimation method; we use the Cohen-Procaccia estimate since it has been shown to work for both chaotic and stochastic sources of entropy [215]. While the NIST draft [26] does not discuss  $h(\epsilon, \tau)$ , we also compute an  $h(\epsilon, \tau)$  using the MCV and Markov estimates from the NIST suite [26] in order to compare with the Cohen-Procaccia estimate.

### 6.1.3.2 Cohen-Procaccia entropy

In this section, we present the Cohen-Procaccia algorithm as a method to estimate  $h(\epsilon, \tau)$  [227]. The Cohen-Procaccia estimate is especially useful because it can be used to directly compare stochastic and chaotic sources [215]. Further, it does not unnecessarily penalize entropy sources with PDFs that are approximately normal like some of the NIST draft tests do [225].

The Shannon entropy, or average amount of information contained per sample, of a random variable  $X$  is given by

$$H(X) = - \sum p(x) \log_2(p(x)), \quad (6.3)$$

where the summation is taken over all possible values of  $x$  [223]. For a joint probability distribution, this definition of entropy extends naturally to

$$H(X_1, X_2, \dots, X_d) = - \sum p(x_1, x_2, \dots, x_d) \log_2(p(x_1, x_2, \dots, x_d)). \quad (6.4)$$

One definition of entropy rate is the average amount of new information obtained by measuring the current sample given the history of previous samples:

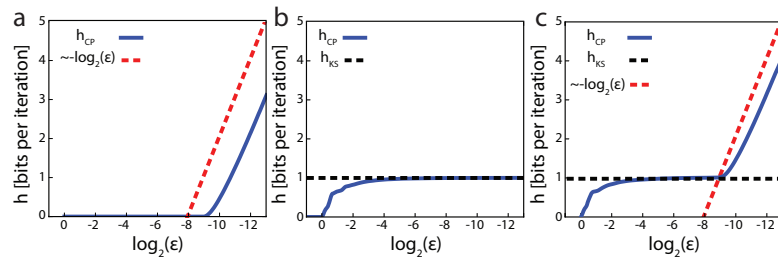


Figure 6.4: Demonstration of the Cohen-Procaccia entropy rate estimate (blue line) on a simple time series. *a* Gaussian noise with strength  $a=0.001$ . *b* Logistic map with parameter  $r=4$ . *c* Logistic map with additive Gaussian noise with standard deviation 0.001. The red dashed line shows a line with slope  $-\log_2(\epsilon)$ . The black dashed line gives the Kolmogorov-Sinai entropy rate of the logistic map with  $r=4$  ( $h_{KS}=1$  bit per sample).

$$\begin{aligned}
h &= \lim_{d \rightarrow \infty} \frac{1}{d\tau} H(X_d | X_{d-1}, X_{d-2}, \dots, X_1) \\
&= \lim_{d \rightarrow \infty} \frac{1}{\tau} (H(X_d, X_{d-1}, \dots, X_1) - H(X_{d-1}, X_{d-2}, \dots, X_1)),
\end{aligned} \tag{6.5}$$

where  $\tau^{-1}$  is the rate at which the distribution is sampled. In Eq. 6.5, one considers the rate at which the entropy of the set of patterns of length  $d$  symbols changes with  $d$ .

To calculate the entropy of a dynamical system, the patterns of length  $d$  that are used are obtained by a  $d$ -dimensional time-delay embedding [10] of the data with delay  $\tau$ . The time-delay vectors can be considered samples from a  $d$ -dimensional probability distribution in phase space.  $h(\epsilon, \tau)$  can then be calculated according to Eq. 6.5.

For an IID random process, each sample will be completely independent of all previous samples, so Eq. 6.5 becomes  $h = \tau^{-1} H(X_d)$ . However, when there are temporal correlations of length  $d\tau$  or less, the  $d$ -dimensional pattern entropy rate will be reduced.

In principle, one can estimate the Shannon entropy directly. First estimate the  $d$ -dimensional joint probability distribution by making a histogram with  $d$ -dimensional boxes of width  $\epsilon$  and use this in Eq. 6.5 to estimate the entropy rate. This approach requires a large amount of data and computing resources for systems with large embedding dimension.

Cohen and Procaccia [227] developed a more efficient way to estimate the entropy rate in order to estimate the Kolmogorov-Sinai entropy from experimental



data of chaotic systems. For additional information about the close relationship between the Kolmogorov-Sinai entropy and the Shannon entropy, see Ref. [226]. Gaspard and Wang [215] later showed that the Cohen-Procaccia algorithm can also be accurately estimate the entropy rate of stochastic systems. We now briefly review the Cohen-Procaccia algorithm.

First, one makes the previously described  $d$ -dimensional time-delay embedding. Then one randomly selects  $M$  of these points as reference points. Typically  $M$  is much smaller than the length of the time series. For each reference point  $n$ , one computes  $f_n(\epsilon)$ , the fraction of other points within a  $d$ -dimensional box of width  $\epsilon$  (that is, within a distance  $\epsilon/2$  of the reference point). Here distance is given by the square metric  $\text{dist}[\mathbf{x}, \mathbf{y}] = \max\{|x_1 - y_1|, |x_2 - y_2|, \dots, |x_d - y_d|\}$ , where  $\mathbf{x}$  and  $\mathbf{y}$  are two  $d$ -dimensional vectors. The  $d$ -dimensional pattern entropy estimate is then given by

$$H_d = -\frac{1}{M} \sum_{n=1}^M \log_2(f_n(\epsilon)). \quad (6.6)$$

The Cohen-Procaccia entropy rate estimate is then obtained by using Eq. 6.6 in Eq. 6.5

$$h_{CP}(\epsilon, \tau, d) = \tau^{-1}(H_d(\epsilon, \tau) - H_{d-1}(\epsilon, \tau)), \quad (6.7)$$

where we have explicitly added in the dependence of  $h$  and  $H$  on the box width  $\epsilon$  and the embedding time-delay  $\tau$ . The only differences between the Cohen-Procaccia estimate and a direct calculation of the Shannon entropy are that the Cohen-Procaccia

estimate uses  $M$  reference points, and that the histogram bins are centered on the reference points instead of being a fixed rectangular array. The Cohen-Procaccia calculation still requires a large amount of data, but is much more computationally efficient than a direct calculation of the Shannon entropy. We note that placing the centers of the bins on the reference points results in poor entropy rate estimates for large  $\epsilon$  (small  $N_\epsilon$ ), but accurate estimates for small  $\epsilon$ .

Gaspard and Wang [215] used the Cohen-Procaccia estimate to compare the entropy generation rates of Gaussian noise and the logistic map  $x_{n+1} = rx_n(1 - x_n)$  with  $r=4$ . We have replicated these results and present them in Fig. 6.4a and b. The Cohen-Procaccia estimate can distinguish a stochastic process from a chaotic one by revealing the dependence of the entropy rate on  $\epsilon$ . For a stochastic process,  $h(\epsilon)$  should scale with  $\log(\frac{1}{\epsilon})$ , while for a chaotic process  $h(\epsilon)$  should converge to  $h_{KS}$  as  $\epsilon \rightarrow 0$ . This behavior is captured by the Cohen-Procaccia algorithm for the logistic map in Figs. 6.4a and 6.4b. In Fig. 6.4b, for low resolution (large  $\epsilon$ ), no entropy can be harvested from the system because the resolution is too coarse. As the resolution increases, the amount of entropy that can be extracted from the system also increases, until the full attractor is resolved. At this point,  $h_{CP}(\epsilon)$  plateaus at  $h_{KS}$ , even as  $\epsilon \rightarrow 0$ .

Gaspard and Wang also considered a noisy logistic map, in which the time series was obtained by iterating the logistic map with  $r=4$  then adding Gaussian noise of standard deviation 0.001 to the output. This result is shown in Fig. 6.4c. For low resolution (large  $\epsilon$ ), the entropy increases as the resolution increases. At intermediate resolution, the chaotic attractor can be fully resolved, but the noise

cannot, so  $h(\epsilon)$  plateaus at  $h_{KS}$ . At high resolution (small  $\epsilon$ ), the noise is resolved and  $h(\epsilon)$  scales as  $\log(\frac{1}{\epsilon})$ . It is clear from the figure not only that the Cohen-Procaccia estimate can accurately predict the rate of entropy production of both stochastic and chaotic systems, but also that the scaling of the entropy rate with  $\epsilon$  can provide some information about where the entropy is coming from at a given measurement resolution.

As we mentioned previously, an  $(\epsilon, \tau)$  entropy rate analysis can be performed with any entropy estimation method. We use the Cohen-Procaccia estimate because it is known to treat both chaotic and stochastic sources fairly [58, 215] and it can recognize higher order correlations better than metrics such as the MCV and Markov estimates. While in principle, the Cohen-Procaccia algorithm can identify correlations of any time scale, this requires an impractically large amount of data if the time scale of the slowest correlations is much slower than the fastest time scale. We do not consider this a significant problem, however, because good physical RNG design involves stabilization techniques to remove slow fluctuations due to external factors, such as power supply or temperature fluctuations.

However, there are some disadvantages to the Cohen-Procaccia entropy rate estimate. In general, the Cohen-Procaccia algorithm does require significantly more data than the NIST entropy estimation suite does, and the amount of data needed increases with the dimension of the entropy estimate. It does, however, pick up higher order correlations better than the NIST entropy estimation techniques. As a result, for entropy analyses of systems that require a high dimension for the Cohen-Procaccia algorithm to converge, it may be difficult to obtain an accurate estimate

of the entropy rate. Still, the Cohen-Procaccia algorithm has value: for example, if the entropy rate estimate varies with dimension, then it is clear that the system is not behaving completely randomly and that the entropy source should probably be sampled less frequently. Once the Cohen-Procaccia entropy rate estimate does converge, then one has both an entropy rate estimate and some assurance that they are sampling at an appropriate rate.

### 6.1.3.3 A comment about the relationship between PRNGs and deterministic chaos

We have stated that PRNGs cannot increase the entropy rate of their input because they are deterministic algorithms. Given the set of equations and the initial conditions, one can calculate the full future output of the system. However, it is well-established that deterministic chaotic systems do have an associated entropy rate,  $h_{KS}$  [10, 215, 226]. One might wonder why a chaotic system can generate entropy while a PRNG cannot.

PRNGs require a seed (initial condition) as input, which contains all the entropy [216]. The PRNG then performs deterministic computations on the seed in order to generate pseudo-random numbers as output. These numbers appear random: they are uniformly distributed and pass all statistical tests of randomness. The numbers are called pseudo-random because for a given seed, the PRNG will always produce the exact same output sequence. Therefore, if an attacker obtains the seed, the future output of the PRNG is completely predictable.

Due to the finite precision of computers, the seed must be finite length and the PRNG will repeat after a finite (often very large) number of iterations [228]; that is, once the entropy from the seed is used up, no new entropy can be obtained from the PRNG. To resolve this, PRNGs are often re-seeded somewhat frequently with additional entropy. Essentially, then, the PRNG serves to reveal the entropy from the seed at a given rate and with some desired properties (e.g. uniformly distributed output); however, the entropy itself must be provided from some other source (often physical entropy sources). Due to finite precision constraints, computer simulation of a deterministic chaotic system is essentially a PRNG and will encounter these same restrictions [228].

We now consider mathematical chaotic systems that are described by deterministic equations with infinite precision. As in the case of PRNGs on a classical computer with finite precision, the entropy in an infinite-precision chaotic system is stored in the initial condition. Unlike in a PRNG, in an infinite-precision chaotic system, an infinite amount of information (or entropy) is stored in the infinitely precise initial condition. Imagine that an observer of this system cannot measure the initial condition to more than a handful of most significant bits. The chaotic system, due to its inherent sensitivity to initial conditions, amplifies the bits of lower significance so that they become measurable [229, 230]. The average rate at which this information about the precise value of the initial condition is revealed by the chaotic system can be quantified by the Kolmogorov-Sinai entropy rate, which is typically equal to the sum of the positive Lyapunov exponents [215, 226].

Of course, infinite-precision chaotic systems do not exist in the real world; all

physical systems are at some level granular, quantized, and susceptible to sources of noise and uncertainty, which together prevent the physical chaotic system from having infinitely precise initial conditions. This uncertainty continuously scrambles the least significant bits of the state of the system, continuously re-seeding the chaotic system. The noise is amplified by the chaos [200, 201] and contributes to the entropy production. In the case of laser chaos, the intrinsic noise has been considered to be quantum mechanical in origin and due to spontaneous emission in the laser [231].

There may also be classical sources of noise. These may be intrinsic, in which case they will be amplified by the chaos, or external (e.g. measurement noise), in which case they may or may not contribute to the measured entropy rate, depending on the  $\epsilon$  and  $\tau$  and post-processing used in the physical RNG, as discussed above.

In summary, for both PRNGs and ideal chaotic systems, the entropy comes from the initial conditions; PRNGs and chaotic systems are similar in that they amplify the bits of low significance of their initial conditions. The fundamental difference is that the amount of entropy in the initial conditions of a PRNG is limited by the finite precision of a computer, while the initial conditions of an ideal chaotic system has infinite precision. Because of the finite precision, PRNGs must eventually repeat and therefore are periodic systems with long periods; however, chaotic systems never repeat. Once a PRNG repeats, it has used up all the entropy in its seed and the output entropy rate is 0; a chaotic system reveals the infinitely many low-significance digits of the initial conditions at an average rate of  $h_{KS}$  forever. Further, because of finite measurement precision and intrinsic noise, chaotic systems

are truly random; Gaspard and Wang show that a chaotic system and a stochastic Markov chain with the same  $h_{KS}$  have the same degree of dynamical randomness [215].

## 6.2 Review of some optical entropy sources

### 6.2.1 RNG with Single Photon Detection

The detection of single photons is perhaps the most established optical RNG technique. There are many different techniques for generating randomness from single photon detection; for a recent review, see Ref. [199]. Perhaps the most straightforward way is to send a single photon through a 50:50 beam splitter and assign a “0” if it is detected at one output port and a “1” if it is detected at the other; this is the method used by the commercial RNG from *ID Quantique* [166], and provides 1 bit per photon of entropy. Another method is to count the number of photons  $n$  detected in a given time window from a low intensity light source. If the average photon interarrival time is much less than the detector dead time,  $n$  will be a random variable that follows the Poisson distribution [199]. It turns out that this technique has the same rate of entropy production as the one we focus on here: single photon time-of-arrival measurements [173, 174, 176].

Fig. 6.5a depicts our experimental realization of high-precision time-tagged photon counting. The optical output of a 1550 nm CW laser is attenuated to an average photon rate of several million photons per second. Photons are detected using superconducting nanowire detectors, and single photon arrivals are time-tagged by

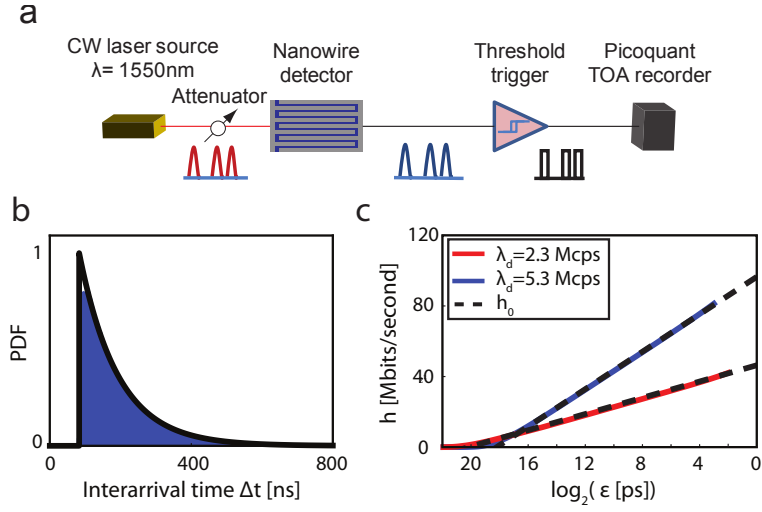


Figure 6.5: a) Apparatus used to make single photon time-of-arrival measurements with a precision of 1 ps. b) Experimental histogram (shaded) and theoretical PDF (black line). The PDF is given by  $p(\Delta t) = \lambda_d \exp[-\lambda(\Delta t - \tau_d)]$  where  $\lambda_d = 5.37\text{ Mcps}$  and  $\tau_d = 88\text{ ns}$ . c) Entropy rate  $h$  as a function of timing precision for single photon time-of-arrival measurements. Two different detected photon rates  $\lambda_d$  were used: 2.3 Mcps and 5.37 Mcps. Eq. 6.10 is used for  $h_0$ . In both cases, the Cohen-Procaccia entropy rate estimate agrees excellently with the prediction from Eq. 6.10.



the HydraHarp 400 (PicoQuant) with a precision of 1 ps. The digitized interarrival times between consecutive photons serve as our random signal.

The photon arrivals can be described as a Poisson process with constant rate  $\lambda$ : the probability per unit time for a photon to arrive is constant and independent of previous photon arrivals. It is well-known that the interarrival times of a constant rate Poisson process follow an IID exponential distribution of the form  $p(t) = \lambda \exp[-\lambda t]$  [173, 174, 199]. We can calculate the entropy per photon generated from these interarrival times as a function of the time-tagging resolution  $\tau$  using Shannon's definition of entropy:

$$H(\tau) = - \sum_{k=0}^{\infty} p_k(\tau) \log_2(p_k(\tau)), \quad (6.8)$$

where  $p_k = \exp[-k\lambda\tau](1 - \exp[-\lambda\tau])$  is the probability of the photon interarrival time occurring between time  $k\tau$  and  $(k+1)\tau$ . This can be evaluated in closed form as

$$H(\tau) = \frac{(1 - p_0) \log_2(1 - p_0)}{p_0} + \log_2(p_0), \quad (6.9)$$

where  $p_0 \equiv 1 - \exp[-\lambda\tau]$ . The entropy generation rate  $h = \lambda H$ .

Of course, real single photon detectors have a dead time. For non-paralyzable detectors, the dead time  $\tau_d$  does not affect the shape of the PDF; it only shifts it by  $\tau_d$ , as shown in Fig. 6.5b. This does not affect  $H$ , the entropy per photon. The dead time does, however, affect the average rate of photons that are detected:  $\lambda_d = \lambda/(1 + \lambda\tau_d)$  [174]. Thus the entropy rate becomes  $h = \lambda_d H$ . Our detector *is* paralyzable; however, if the photon rate is not too high, we can approximate it as non-paralyzable. Further, if the probability of more than one photon arriving in a

single time bin is small ( $\lambda\tau \ll 1$ ), we can approximate the entropy rate for photon time-of-arrival measurements as

$$h(\tau) = -\lambda_d \log_2\left(\frac{\lambda\tau}{e}\right). \quad (6.10)$$

This is exactly the same entropy rate one would obtain by using Eq. 6.2, with  $\min(\tau^{-1}, 2BW) = \lambda_d$ .

We estimate the entropy rate of experimentally measured photon interarrival time measurements using the Cohen-Procaccia algorithm as a function of the time-tagging resolution  $\tau$ . We have performed time-of-arrival measurements for two different detected rates:  $\lambda_d=2.3$  Mcps and  $\lambda_d=5.37$  Mcps. In the first case the dead time is not important ( $\lambda_d \approx \lambda$ ), while in the second case the dead time causes a loss of about 40% of the photons. In both cases these results give excellent agreement with Eq. 6.10, as shown in Fig. 6.5c. Furthermore, only 1 dimension is needed for the Cohen-Procaccia algorithm to converge. This suggests that the photons were indeed generated by a Poisson process with no intersample correlations.

The entropy rates obtainable from modern photon counting experiments are on the order of a few hundred Mbits/s, and physical limitations present a significant challenge for improving performance. As shown in Eq. 6.10,  $h$  scales *logarithmically* with the time-tagging resolution, suggesting that increasing the time-tagging resolution beyond the current state-of-the-art of 1ps is not an economical way to improve performance. The most efficient way to increase the entropy rate is to increase the detected photon rate  $\lambda_d$ , which is limited by the detector dead time and is typically on the order of a few tens of millions of counts per second with current technology.

It is of course possible to increase the entropy rate by combining techniques. For example, Stipcevic and Bowers were able to obtain one additional bit per photon by combining the 50:50 beam splitter method with time-of-arrival measurements [232]. However, it is clear that this does not improve the scaling.

Single photon detection techniques are attractive because of their quantum mechanical nature and conceptual simplicity. However, in light of the physical limitations described above, we find it unlikely that RNG using single photon detection techniques can become competitive with the high-speed digitization of unpredictable analog waveforms, which can produce entropy rates of hundreds of Gbits/s.

### 6.2.2 RNG from Chaotic Lasers

It has long been known that a semiconductor laser can be made chaotic by creating a time-delayed optical feedback via a reflector [31, 233]. In 2008 it was demonstrated for the first time that these chaotic lasers can be used to generate random numbers, and could do so at rates an order of magnitude faster than any previous physical RNGs [17]. Since then, there has been much progress in building faster chaotic RNGs [183, 202–206].

Chaotic systems produce entropy by magnifying the small uncertainties in the initial conditions. The maximum rate at which entropy can be harvested from a chaotic system is called the Kolmogorov-Sinai entropy rate  $h_{KS}$ , which is equal to the sum of the positive Lyapunov exponents [215, 226]. Ref. [226] provides an in-depth review of the relationship between the Kolmogorov-Sinai entropy rate and the

Shannon entropy rate. Of course, real dynamical systems also have intrinsic noise sources, which fundamentally limit the precision with which initial conditions can be measured. For the chaotic laser systems used for optical random number generation, spontaneous emission noise (which is of quantum mechanical origin) and potentially other noise sources are continuously amplified by the chaotic dynamics and contributes to the entropy production. The rate at which entropy can be harvested from a chaotic laser is limited both by the bandwidth of the detectors (Eq. 6.2) and  $h_{KS}$  of the chaos.

The chaotic laser system we consider here obtains an enhanced bandwidth by cascading three semiconductor lasers (NTT Electronics, KELD1C5GAAA), as described in detail in Ref. [205] and shown in Fig. 6.6a. The first laser has time-delayed optical feedback from the reflector. The chaotic output intensity of the first laser is injected into the second laser, and the chaotic output of the second laser is then injected into the third laser. This cascading increases the standard bandwidth from 12.5 GHz at the first laser to 34 GHz at the final output. The final output intensity is detected by a 38 GHz photodetector (New Focus, 1474-A). The electrical signal from the photodetector is sampled by a high speed 8-bit oscilloscope (Tektronix DPO73304D, 33 GHz bandwidth, 100 GigaSamples/s). The RF power spectrum is shown in Fig. 6.6b, and a typical time series is shown in Fig. 6.6c. One can immediately tell from the time series in Fig. 6.6c that a few of the least significant bits of the signal are due to electronic noise rather than the optical signal.

Figures 6.7a and 6.7b compare the results of different entropy rate estimates on the chaotic laser signal. For the Cohen-Procaccia estimate, we use  $d = 6$  embedding

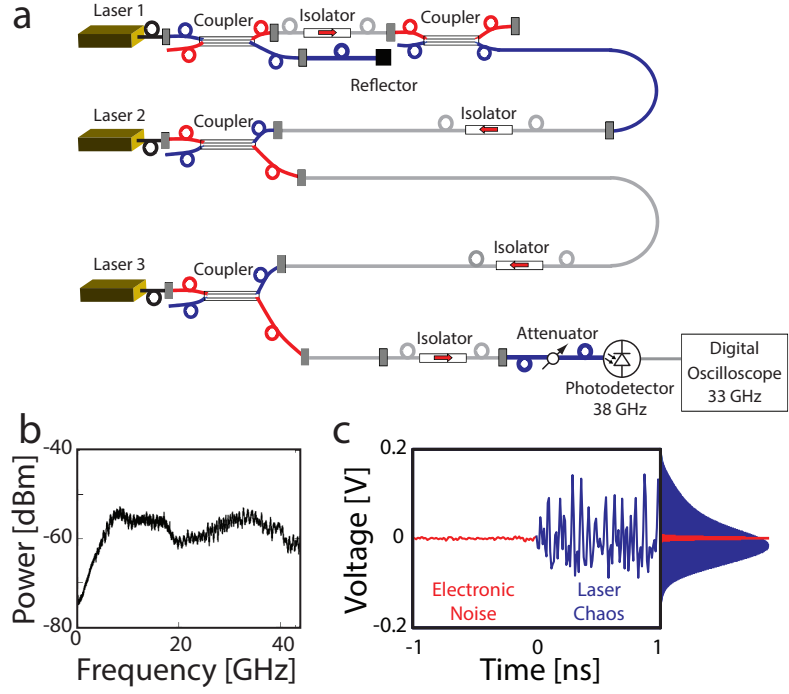


Figure 6.6: a) Schematic of a three cascaded semiconductor laser entropy source. b) RF power spectrum of chaotic laser system. The injection currents of lasers 1, 2, and 3 are 58.5 mA, 59.0 mA, and 59.0 mA, respectively. The standard bandwidth is 34 GHz. c) Time series and PDF of the chaotic laser system (blue) and the electronic noise (red). The electronic noise is measured with all optics turned off.

dimensions. While it is unlikely that the attractor can be fully embedded in 6 dimensions, we choose  $d = 6$  because the entropy rate estimate did not change much for  $d > 6$ . The bandwidth used in determining  $h_0$  is the detector bandwidth  $BW = 33$  GHz because the bandwidth of the chaos (34 GHz) is greater than the detector bandwidth. Because we do not have a theoretical prediction for a PDF for the chaotic laser system, we use the experimentally measured PDF for computing  $D_{KL}$  in Eq. 6.2.

In Fig. 6.7a, we consider  $h(\epsilon)$  for a fixed  $\tau$ .  $N_\epsilon$  is the number of bits per sample measured by the detector. For the NIST tests  $N_\epsilon$  is obtained by taking the appropriate number of most significant bits from an 8-bit oscilloscope measurement; for the Cohen-Procaccia estimate  $N_\epsilon$  is obtained by referencing the bin width  $\epsilon$  (described in Section 6.1.3.2) to the full 8-bit resolution of the oscilloscope (1.6mV). We also show the thresholding ( $N_\epsilon = 1$  bit) limit  $h = 2BW$  bits/s as a blue dotted line. The shading above these limits denotes a region of unobtainable entropy rates for a system with the experimental PDF shown in Fig. 6.6c. For comparison, we show the  $d = 1$  Cohen-Procaccia entropy rate estimate, which estimates the entropy of the experimentally measured histogram and considers no temporal correlations. This shows what the entropy rate would be if the system were actually IID when sampled at  $\tau^{-1} = 100$  GSamples/s. Of course, the system *cannot* be IID when sampled at faster than 66 GSamples per second, since the detector bandwidth is 33 GHz. Unsurprisingly, both the MCV and the  $d = 1$  estimates significantly overestimate the entropy rate. The Markov estimate does little better, still providing an entropy rate estimate that is significantly higher than the  $h_0$  limit. The  $d = 6$

Cohen-Procaccia estimate agrees well with the Eq. 6.2 limit for  $N_\epsilon \geq 2$ . However, the Cohen-Procaccia estimate is unable to provide a  $d = 6$  estimate for  $N_\epsilon > 4$  due to the data requirements (we used 1GB of data).

Fig. 6.7a also shows the entropy rate obtained by simply turning off all optics and measuring the electronic noise at the detector. It is clear that the electronic noise in the detector and ADC contributes a significant fraction (about 20%) of the entropy at full resolution. This noise is in part due to the inherent sampling noise of high-speed digitizers, which is typically quantified by the effective number of bits (ENOB). For the oscilloscope used here, the ENOB is 5.5 bits.

Fig. 6.7b shows the entropy rate as a function of the sampling rate  $\tau^{-1}$  for a fixed  $N_\epsilon = 3$  bits. As the sampling rate is increased, the maximum entropy rate increases, then starts to plateau at a sampling rate of about 50 GSamples per second. As expected, the MCV estimate detects no correlations and continues to increase for  $\tau^{-1} > 2\text{BW}$ ; the Markov estimate does only a little better, showing a slight roll off. The  $d = 6$  Cohen-Procaccia estimate has the most noticeable roll off, indicating that it detects temporal correlations better than the other two methods. By detecting these correlations, the Cohen-Procaccia algorithm informs us that the experiment is not purely random at high sampling rates; this can inform the choice of  $\tau^{-1}$  in RNG design. For example, it suggests that this system should be sampled at  $\tau^{-1} < 50 \text{ GSamples/s}$  if the designer wants there to be no temporal correlations.

As noted above, there are two important limits to consider in the design of a RNG based on chaotic lasers:  $h_{KS}$  and  $h_0$ . One might wonder about the interplay between  $h_{KS}$ , which describes the dynamics of the chaotic system and  $h_{SH}$ , which

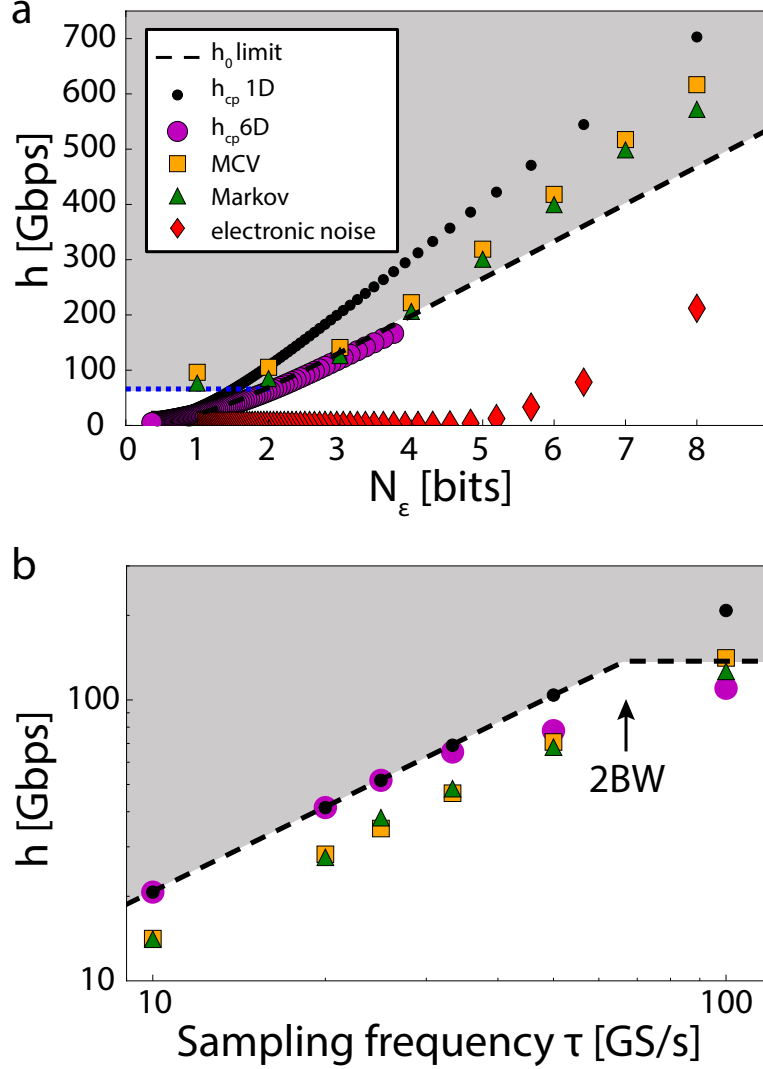


Figure 6.7: a) Entropy rate  $h(\epsilon)$  of the chaotic laser system for a fixed  $\tau^{-1} = 100$  GSamples/s. The blue dotted line is the entropy rate obtained by thresholding at the median,  $\tau^{-1}$ . b) Entropy rate  $h(\tau)$  of the chaotic laser system for a fixed  $N_\epsilon = 3$  bits. The  $h_0$  limit (Eq. 6.2) is the information theoretical limit for the entropy rate given the PDF and bandwidth of the signal. Since we do not have a theoretical prediction for the PDF of the laser chaos, we estimate the PDF using the experimental histogram shown in Fig. 6.6 and determine  $D_{KL} = -0.95$  bits.  $h_{CP}$  is the Cohen-Procaccia entropy rate estimate performed on the data. Here we show  $h_{CP}$  using embedding dimensions (pattern lengths) of  $d = 1$  and  $d = 6$ .  $MCV = -\tau^{-1} \log_2(p_{max})$  is the Most Common Value entropy rate estimate from the NIST draft recommendations [26]. The Markov estimate, also from the NIST draft recommendations, takes into account first-order correlations in the data.  $h_{CP}$  ( $d = 3$ ) for the electronic noise in the detectors is also shown.



limits the amount of information that can be transmitted through a finite bandwidth channel. It has long been known that filtering a chaotic system does not change the  $h_{KS}$  [234, 235]. Passing a chaotic signal through a linear filter simply makes the current output of the filter some linear combination of all the previous inputs to the filter. In principle, if one knows the linear combination that describes the filter, one can determine the unfiltered output of the chaotic system from the filtered output and thus obtain an entropy rate of  $h_{KS}$ .

It might seem, then, that a RNG based on a chaotic system can violate the Shannon-Hartley limit described above by low-pass filtering a chaotic signal so that  $h_{SH} < h_{KS}$ . This is not the case. There is a minimum resolution necessary to obtain  $h_{KS}$ , as described in Section 6.1.3.2 and Ref. [215]. One must use a higher resolution to obtain  $h_{KS}$  from the filtered chaotic system than is necessary to obtain  $h_{KS}$  from the unfiltered system [236]. This increase in resolution increases  $N_\epsilon$  in such a way as to cancel the decrease in bandwidth  $BW$  and ensure that the Shannon-Hartley limit (Eq. 6.1) is not violated.

Essentially,  $h_{KS}$  describes the rate at which the chaotic system generates entropy. This is a property of the physical entropy source and is independent of filtering or any other part of the digitization process.  $h_{SH}$  and  $h_0$  describe the rate at which entropy can be harvested by the measurement apparatus given the resolution and bandwidth limitations of the measurement apparatus. In short, just because a physical system is generating entropy at a rate  $h_{KS}$  does not mean that a given measurement apparatus is able to harvest that much entropy from the system.

$h_{KS}$  can in principle be determined from a deterministic model of the chaotic

system by calculating the Lyapunov spectrum; however, a reliable estimate of  $h_{KS}$  from experimental data is problematic due to the high dimensionality of three time-delayed chaotic lasers. We note that the largest Lyapunov exponent has been calculated numerically to be on the order of several  $\text{ns}^{-1}$  for two cascaded chaotic lasers [237]. We expect that  $h_{KS}$  should be several times greater than this for our three cascaded laser system, since the additional chaotic laser should increase the complexity and  $h_{KS}$  depends on all the Lyapunov exponents, not just the largest one. Our entropy estimates shown in Fig. 6.7 are consistent with this expectation.

### 6.2.3 RNG from Amplified Spontaneous Emission

The final optical RNG technique we analyze is the detection and digitization of optically filtered amplified spontaneous emission (ASE) noise from a light source such as superluminescent diode (SLD). ASE sources provide an easily measurable, high bandwidth noise signal and have been used for RNG since 2010 [179]. Because it is inherently quantum mechanical in origin, ASE cannot be described deterministically; thus, entropy can be harvested by detecting and sampling the ASE signal.

There have been several different but closely related schemes to generate random numbers from ASE sources [167, 179–185]. Here, we discuss the system depicted in Fig. 6.8a. The ASE output of a SLD (DenseLight Semiconductors DL-CS5254A-FP) passes through an optical isolator, a tunable optical filter (Santec OTF-970), and an erbium-doped fiber amplifier (EDFA, PriTel FA-18-IO). This optical intensity is then attenuated before being detected by a 38 GHz photodetector (New Focus

1474-A). The electrical signal from the photodetector is amplified by a 35 GHz electronic amplifier (Picosecond pulse labs, 5882-219) then sampled by a high speed 8-bit oscilloscope (Tektronix DPO73304D, 33 GHz bandwidth, 100 GigaSamples/s).

The optical filter is used to control the bandwidth of the ASE signal. It has been shown that the optimal optical filter bandwidth for RNG is approximately equal to the photodetector bandwidth [179]. In this case, we used an nominal optical bandwidth of 0.6nm at 1550 nm center wavelength. The experimentally measured power spectrum of the ASE signal with this filter is given in Fig. 6.8b; the 90% signal bandwidth is 31 GHz. The PDFs and representative time series of both the full ASE signal and the electronic noise (with all optics turned off) are given in Fig. 6.8c. As in the case of the chaotic laser, one can tell from the time series in Fig. 6.8c that a few of the least significant bits of the signal are due to electronic noise rather than the optical signal. This is confirmed by the entropy analysis in Fig. 6.9a.

While the probability distribution of the photocurrent output by the photodetector depends on the properties of the optical filter and photodetector used, it is known that the photocurrent distribution can be reasonably approximated by the gamma distribution

$$p_i(x) = \frac{x^{a-1} \exp[-x/b]}{b^a \Gamma(a)}, \quad x > 0 \quad (6.11)$$

where  $a$  is the shape parameter (signal-to-noise ratio) and  $b$  is the scale parameter [179,238]. Fig. 6.8c shows that this is a reasonably good approximation in this case, with  $a=2.77$  and  $b=17.7$  mV.

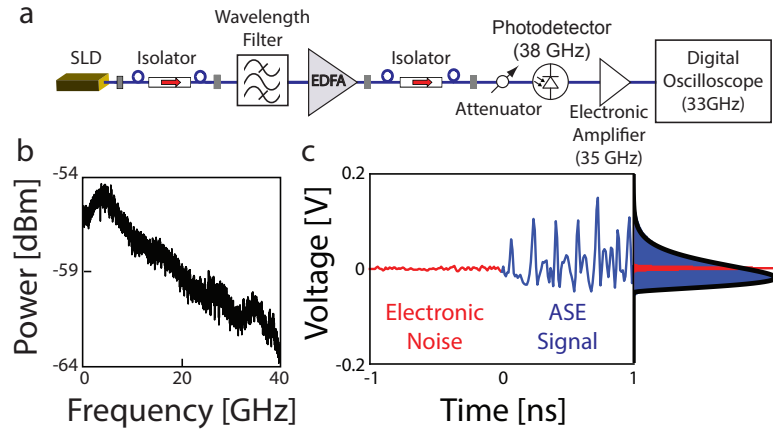


Figure 6.8: a) Experimental setup for digitizing filtered ASE signal from a SLD. The injection current of the SLD is 300.0 mA. b) RF spectrum for detected ASE signal when the optical filter has a width of 0.6nm and center frequency of 1550nm. The 90% bandwidth of the resulting signal is 31 GHz. c) Time series and PDFs of electronic noise only (red) and detected ASE signal (blue) when optical filter has a width of 0.6nm. The solid black line is the best-fit of Eq. 6.11 to the experimental data.

Figures 6.9a and 6.9b compare the results of different entropy rate estimates on the ASE signal. As with the chaotic laser,  $N_\epsilon \leq 8$  is obtained by taking the appropriate number of most significant bits from an 8-bit oscilloscope measurement for the NIST tests; for the Cohen-Procaccia estimate  $N_\epsilon$  is obtained by referencing the bin width  $\epsilon$  to the full 8-bit resolution of the oscilloscope (1.6mV). The upper limit  $h_0$  is given by Eq. 6.2 as a black dashed line and the thresholding ( $N_\epsilon = 1$  bit) limit  $h = 2BW$  bits/s as a blue dotted line. Here,  $BW = 31$  GHz since the 90% signal bandwidth is the smallest relevant bandwidth. We use Eq. 6.11 with best-fit parameters  $a=2.77$  and  $b=17.7$  mV to determine  $D_{KL}$  and  $h_0$ . The shading above these limits denotes a region of unobtainable entropy rates for a system with the PDF given by Eq. 6.11. In Fig. 6.9a, the  $d = 1$  Cohen-Procaccia estimate shows what the entropy rate would be if the system were IID when sampled at  $\tau^{-1} = 100$  GSamples/s. Of course, the system *cannot* be IID when sampled at faster than 62 GSamples per second, since the signal bandwidth is 31 GHz. Thus, it is expected that the  $d = 1$  Cohen-Procaccia estimate would lie in the shaded region of unobtainable entropy rates.

For this data, we use the Cohen-Procaccia entropy rate estimate with  $d = 6$  dimensions because the entropy rate estimate did not change much for  $d > 6$ . The  $d = 6$  Cohen-Procaccia estimate closely follows the  $h_0$  limit, while the other entropy estimates show a significantly larger slope, resulting in large overestimates of the entropy rate. The  $d = 6$  Cohen-Procaccia lies slightly above the  $h_0$  limit; this is likely due to the mismatch between the theoretical PDF in Eq. 6.11 and the actual experimental PDF. The  $d = 6$  Cohen-Procaccia estimate is the only one of the 4

entropy estimates shown that seems to follow the  $h_0$  limit; however, the Cohen-Procaccia estimate is unable to provide a  $d = 6$  estimate for  $N_\epsilon > 4$  due to the data requirements.

Fig. 6.9a also shows the entropy rate obtained by simply turning off all optics and measuring the electronic noise at the detector. As for the case of the chaotic laser, it is clear that the electronic measurement noise contributes a significant fraction (about 20%) of the entropy at full resolution. Again, the typical ENOB for the oscilloscope used is 5.5 bits.

Fig. 6.9b shows the entropy rate as a function of the sampling rate  $\tau^{-1}$  for a fixed  $N_\epsilon = 3$  bits. Again, the dashed black line denotes the upper limit provided by Eq. 6.2. For lower sampling rates, the Cohen-Procaccia estimate is slightly above the Eq. 6.2 limit; this is due to the mismatch between the theoretical and experimental PDFs shown in Fig. 6.8c. As the sampling rate is increased, the maximum entropy rate increases, then plateaus as the sampling rate approaches twice the signal bandwidth (62 GSamples/s). As expected, the MCV estimate detects no correlations and continues to increase for  $\tau^{-1} > 2BW$ ; The Markov and Cohen-Procaccia estimates do quite a bit better, showing a roll off with increasing sampling rate. This indicates that there are temporal correlations in the data. By detecting these correlations, the entropy estimates inform us that the experiment is not behaving purely stochastically for sampling rates that are too high. While the Markov and Cohen-Procaccia estimates perform similarly for  $N_\epsilon = 3$ , it is clear from Fig. 6.9a that the Markov estimate does not give valid results for  $N_\epsilon \geq 7$ .

Since ASE sources can have large bandwidth, the main factor that limits the

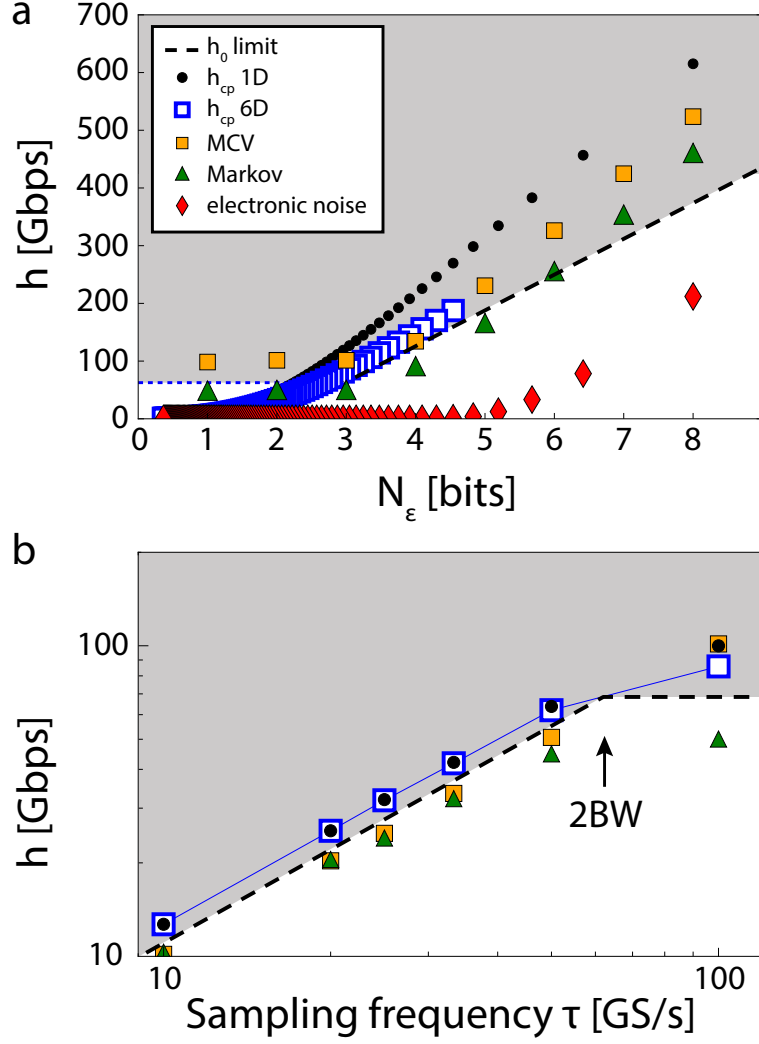


Figure 6.9: a) Entropy rate  $h(\epsilon)$  of the ASE signal from an SLD and a 0.6nm filter for a fixed  $\tau^{-1} = 100$  GSamples/s. The blue dotted line is the entropy rate obtained by thresholding at the median,  $\tau^{-1}$  b) Entropy rate  $h(\tau)$  from an SLD and a 0.6nm filter for a fixed  $N_\epsilon = 3$  bits. The  $h_0$  limit (Eq. 6.2) is the information theoretical limit for the entropy rate given the PDF and bandwidth of the signal ( $D_{KL} = -1.94$  bits for the best-fit PDF from Eq. 6.11 and shown in Fig. 6.8).  $h_{CP}$  is the Cohen-Procaccia entropy rate estimate performed on the data. Here we show  $h_{CP}$  using embedding dimensions (pattern lengths) of  $d = 1$  and  $d = 6$ .  $MCV = -\tau^{-1} \log_2(p_{max})$  is the Most Common Value entropy rate estimate from the NIST draft recommendations [26]. The Markov estimate, also from the NIST draft recommendations, takes into account first-order correlations in the data.  $h_{CP}$  ( $d = 6$ ) for the electronic noise in the detectors is also shown.

entropy rate for ASE sources is the bandwidth of the measurement apparatus. Thus, as the bandwidths of photodetectors and digitizers improve, we expect the entropy rate of ASE sources to similarly increase. Additionally, a single ASE source can be used to generate multiple independent bitstreams by taking different slices of the optical spectrum, as done in Ref. [180].

### 6.3 Conclusions

Physical RNGs are becoming increasingly important in digital communications and security, as evidenced by their widespread commercial availability, both embedded in CPUs [168] and as external devices [165–167]. Optical and photonic systems are leading the way as physical sources of randomness due to their high speed, access to the inherent randomness in quantum mechanical phenomena, and resistance to external interference from electric and magnetic fields. In the last decade, optical RNGs have elevated the state-of-the-art from a few hundred Mbits/s to one Tbit/s.

Motivated by this race for the highest random bit rates, researchers have often been insufficiently concerned about where the entropy is coming from. Instead, the standard practice is to sufficiently post-process some unpredictable signal so that the final output bit sequence can pass statistical tests designed for PRNGs. As we discussed above, the fastest physical entropy sources involve the digitization of high-bandwidth, unpredictable analog waveforms. The digitization process naturally forces one to think about what the measurement resolution  $\epsilon$  and sampling rate  $\tau^{-1}$  should be. As we have shown, the choices of  $\epsilon$ ,  $\tau$ , and post-processing technique can



determine which physical process or processes contribute to the extracted entropy rate.

The new NIST draft recommendations for the evaluation of physical RNGs [26] are a significant improvement upon the old standards [217]; they suggest that one estimate the entropy using minimally post-processed data and require some physical justification of where the entropy is coming from. However, the new standards do not recognize dynamical entropy sources or the importance of the digitization process ( $\epsilon$  and  $\tau$ ).

We recommend that physical RNG evaluation techniques evolve away from statistical tests designed for PRNGs toward entropy estimates that provide insight into the physical origins and limitations of the optical entropy source. In order to achieve this, we recommend that RNG designers perform an  $(\epsilon, \tau)$  entropy analysis on the raw digitized data (as in Fig. 6.1b) using a variety of entropy estimates, including the the Cohen-Procaccia estimate and tests from the NIST entropy estimation suite. The  $h(\epsilon, \tau)$  analysis, in conjunction with considering simple physical and information theoretical limits of entropy generation, provides more than a simple pass/fail validation of a RNG; it provides relevant information about an entropy source such as how finely and frequently to sample the source and what types of post-processing and conditioning should be used to extract entropy from the desired source.

As an example, we have performed this analysis for three state-of-the-art optical entropy sources. We found that the digitization of unpredictable, high-bandwidth analog signals generates significantly higher entropy rates than does sin-

gle photon detection. Chaotic lasers and ASE signals can produce similar entropy rates (on the order of hundreds of Gbits per second); however, the simplicity of the ASE setup is attractive as is the ability for one SLD to generate multiple independent bitstreams, as done in Ref. [180]. We also found that post-processing methods that use least-significant bit extraction might be taking their entropy from electronic noise in the detector or digitizer rather than from the desired optical entropy source.

## 6.4 Epilogue

In May 2016, we presented much of the research on which this chapter was based at the NIST Random Bit Generation Workshop (video of our presentation is available at Ref. [239]) in Gaithersburg, MD. The purpose of this workshop was for “the public” (non-NIST researchers as well as private companies) to comment on the Second Draft of NIST’s SP800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation. The manuscript on which this chapter was based was published as a “Perspective” in APL Photonics in September 2017.

In January 2018, the final version of NIST’s SP800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation was released [25]. This document is very similar to the SP800-90B Second Draft, which was the most up-to-date version at the time of writing of our manuscript. However, the final version did make some important changes. Some of these changes are summarized below. A complete list of the changes can be found in Ref. [240].

## Post-processing

Limited post-processing of the digitized data before entropy estimation was allowed in the Second Draft of SP800-90B [26] (italics as in the original document):

If the non-deterministic activity being sampled produces something other than binary data, the sampling process includes a *digitization* process that converts the output samples to bits. The noise source [sic] may also include some simple *post-processing* operations that can reduce the statistical biases of the samples and increase the entropy rate of the resulting output. The output of the digitized and optionally post-processed noise source is called the *raw data*.

The document goes on to list Von Neumann’s method, the linear filtering method, and the length-of-runs method as the only approved post-processing methods. We (and others [214]) criticized the allowance of any post-processing before entropy estimation because deterministic post-processing cannot increase the entropy rate of the entropy source, but instead just makes the entropy estimation more difficult.

In the final version of SP800-90B, the “concept of post-processing of the noise source is removed” [240]. The final version reads [25]:

If the non-deterministic activity being sampled produces something other than binary data, the sampling process includes a *digitization* process that converts the output samples to bits. The output of the digitized noise source is called the *raw data*.

There is no mention of “post-processing” in the final document. “Conditioning” to extract entropy from the entropy source is still allowed, but must occur after entropy estimation.

### **A comment about the scope and intent of the entropy estimation tests**

A comment clarifying the limitations of the entropy estimation was added to the final version [25]. The document now states that the

entropy estimation methods described in this section rely on some statistical assumptions that may not hold for all types of noise sources.

The methods should not replace in-depth analysis of noise sources, but should be used to support the initial entropy estimate of the submitter.

This comment seems to emphasize the importance of the initial entropy estimate of the submitter, which should be based on the physical principles of the entropy source. We view this as an important improvement because, as we stated in Section 6.1.1 “statistical tests are perhaps best viewed as a sanity check against blatant errors, rather than a proof of randomness.” We are hopeful that the fact that NIST is now also emphasizing this view will encourage designers of physical RNGs to focus on the physical origins and limitations of the entropy source.

### **Entropy estimation tests**

Researchers at NIST [225] and elsewhere [241] have found that some of the tests in the NIST suite severely underestimate the min-entropy of entropy sources with normal distributions. As we pointed out in Section 6.1.2 this is a practical problem

because many of the best physical entropy sources have normal or approximately normal PDFs. The final version of SP800-90B solves this problem by requiring the collision, Markov and compression entropy estimates for only sources that produce binary outputs.

This is an important concession, as the combination of allowed post-processing and the requirement of these tests by the previous Recommendation unintentionally encouraged designers to include post-processing before testing, since this is the only way a normally-distributed source could have received a high entropy estimate from the NIST suite.

### **The measurement process**

Documentation explaining why the data collection method does not interfere with the noise source is now required. The draft Recommendation did not require this. While this does not go as far as our own recommendation to encourage some sort of  $(\epsilon, \tau)$  analysis, this is a significant improvement in the final version, as an understanding of the impact of the measurement process on the entropy source being measured and on the measured result itself is essential to determining whether any physical device is working properly.

We do not claim credit for these or any of the changes present in the NIST's final Recommendation. Hundreds of researchers and industry representatives from all over the world attended the Random Bit Generation Workshop (either in person or via teleconference) and/or sent in comments regarding the Second Draft. All of these comments as well as journal publications and other factors were surely considered

by NIST when deciding their final Recommendation. However, we do think these changes significantly improve the final Recommendation for the Entropy Sources Used for Random Bit Generation. Still, we continue to encourage researchers to use an  $(\epsilon, \tau)$  entropy analysis in addition to following the NIST Recommendation to help them in the design and testing of physical entropy sources.

## Chapter 7: What’s next?

Today is gone. Today was fun.

Tomorrow is another one.

Every day, from here to there,

funny things are everywhere.

---

*One fish two fish red fish blue fish*

DR. SEUSS

In Chapters 2 and 3, we presented our experimental observation of “synchronized chimeras” and patterns of cluster synchronization in a network of four optoelectronic oscillators. This was the first observation of chimeras in such a small network. We showed that “synchronized chimeras” are a special case of cluster synchronization and that the stability of both “synchronized chimeras” and cluster synchronization can be calculated using the same group theoretical techniques. This additionally showed that some chimeras *can* be stable, something that was not known previously. As far as I am aware, it is still not clear whether there are cases in which “coherent chimeras” can be stable.

In our experiments, chimeras were always observed in regions of multistability with other patterns of synchronization. It is not clear whether this is a general

requirement. Further, we do not have any real understanding of the sizes of the basins of attraction of different patterns of synchronization in these regions of multistability. Because our oscillators had feedback and coupling delays, we had an extremely large initial condition space, and so a study of the basins of attraction was not feasible. Perhaps the new experimental network described in Ch. 4 would permit the study of basins of attraction, since the initial condition space is “only”  $N$  dimensional, where  $N$  is the number of nodes in the network.

Chapters 4 and 5 concern a new experimental technique that we developed for the implementation of coupled oscillator networks with arbitrary topology. We used the space-time interpretation of systems with a long delay along with non-traditional digital filtering to realize a complex network in a single, table-top opto-electronic oscillator. This technique is the only way to implement a network of truly identical nodes in an experiment.

The question of the usefulness of our arbitrary network experiment for reservoir computing tasks is one obvious and important path for future research. Of course, our system would have to be sped up significantly from the kHz rates used here. Along these lines, it has been demonstrated using faster electronics that similar opto-electronic networks with ring topology can characterize up to 1 million words per second [143]. However, it is not clear that this is the optimal network structure for reservoir computing (or even if the topology of the reservoir network matters at all). Whether using the longer range connections available in an arbitrary network like our system would help is an interesting question from the point of view of both basic and applied science. At the very least, we suspect the ease of network



re-configuration in our set-up should make it useful for research and development, before implementation on dedicated hardware once optimal networks are better understood.

While delay systems themselves are often continuous time systems, the space-time representation causes delay networks to be discrete in time. We believe it is possible, at least in some cases, to realize continuous-time networks in a single delay system by adopting the multiple time delay implementation of arbitrary networks discussed in Section 4.4.2. Importantly, this technique is not reliant on opto-electronics: one could replace the optics with any system of interest. This might be useful for building prototypes for large networks of coupled oscillators when the oscillators are expensive, such as in the case of power grids. It may also allow for the experimental study of large networks of truly identical oscillators in situations where the oscillators are rarely identical in practice (e.g. biological systems such as neurons). This could permit the study of the impact of heterogeneity on the network dynamics.

In Chapter 6 we presented some recommendations for how to evaluate the rate of entropy production of the entropy source in physical random number generators. Some of the ideas we suggested, such as requiring the estimation of entropy rates before any post-processing of the raw data and emphasizing the in-depth analysis of the physical processes generating the entropy, have been incorporated into the NIST Recommendation for the Entropy Sources Used for Random Bit Generation [25], published January 2018.

The aforementioned release of the official NIST recommendations for physical

entropy sources has provided some new entropy estimation techniques. These now need to be tested on all types of optical entropy sources. The statistical analysis of data from physical entropy sources is a highly challenging but important problem in the field of RNG. We hope that our work as well as the recent release of the NIST recommendations will lead the optical RNG community to embrace entropy estimation from the raw physical data, and to continue to develop new and better entropy estimates.

In this thesis we advocated the Cohen-Procaccia method for estimating the entropy production of a physical system from raw data. The Cohen-Procaccia method is one of a series of “phase space” or “time-delay embedding techniques” that involve estimating the entropy production by counting the frequency of appearance of data patterns of a given length.

An alternative approach is to try to predict the next output of the system using the best possible deterministic model. An attacker could then try to synchronize their deterministic model with the system and use their model to predict the next output. However, it is not always clear how to come up with “the best deterministic model.” It has recently been demonstrated that a reservoir computer can successfully predict chaotic systems so well that the Lyapunov exponents can be accurately calculated from the known reservoir equations of motion, even in the presence of some observation noise [242]. In this sense, the reservoir “learns” the best deterministic model of the noisy chaotic system. The attacker can then try to synchronize the reservoir to the system and use this deterministic model to predict the next output of the entropy source.

This suggests a new entropy estimation technique. A trained reservoir computer (on data from the entropy source) gives the best guess (from a deterministic model) you can have for the next output. The deterministic equations of the reservoir can then be used to obtain an estimate of the entropy rate of the best deterministic model of the entropy source. The use of model-agnostic AI methods, in general, to predict otherwise random variations and quantify entropy is an interesting topic that has not been widely explored.

Another crucial issue facing the optical RNG community is to bridge the gap between the ultrafast RNG rates possible in the lab ( $\sim 1$  Tb/s) and the significantly slower speed of commercially available optical RNGs ( $\sim 1$  Gb/s). The ultrafast rates in laboratory experiments have been obtained by taking one-time measurements with an oscilloscope; they cannot be sustained for more than tiny fractions of a second. Further, the postprocessing necessary to extract the entropy is often done offline. One critical path of future research is developing real-time implementations of the post-processing necessary for optical RNGs. There has been some work in this area [209,212,243,244]; however, as of now real-time implementations of optical RNGs have a long way to go to reach the Tb/s rates of the one-time oscilloscope measurements. Once post-processed, there is the problem of transferring the data to the memory of the user in real time. High-speed entropy extraction and data transfer are two major practical problems that the field of optical random number generation will have to address in the coming years.

Additionally, in order to be practically useful, the size and cost of the laboratory RNGs must be reduced. One promising way to do this is by implementing

the optical RNG on photonic integrated circuits [209]. Low-cost photonic integrated circuits are currently being developed for optical computing and information transfer within traditional computing; we anticipate that optical RNGs will begin to be developed on chip as well. These circuits provide the additional benefit that they are robust against temperature fluctuations and air turbulence.

We also anticipate the development of optical RNGs with special properties suited for application-specific purposes. One recent example of this is the laser phase noise-based RNG used in the recent loophole-free Bell tests [245–247]. Each of these tests relied on the real-time optical RNG described in Ref. [244] to randomly and independently choose the measurement bases such that the choice of basis is space-like separated. This allowed the researchers to close the locality loophole.

Over the last decade, much of the optical RNG research has focused on breaking bit rate records that pass the statistical test suites. Once the optical RNG community shifts focus from record breaking to entropy analysis and physical origins of randomness, we will also see an increasing focus on decreasing the size and power constraints and increasing the robustness to external (potentially unsafe) noise sources. When all of these considerations are taken into account, optics and photonics will emerge as the most promising technology for physical random number generation.

## Appendix A: Details of the experimental implementation

We now provide more details about the experimental implementation of the setup described in Section 4.4.3, including a discussion of some of the important design choices. As mentioned previously, FPGAs are particularly well-suited for this experiment because they are easily reconfigurable. Indeed, any arbitrary network can be implemented by simply reconfiguring the FPGA without modifying any other part of the experimental apparatus. We do the FPGA programming in verilog, a common hardware description language. An example of the verilog code used to implement a simple two node network with bidirectional coupling is given in Appendix B. However, for large networks, writing out the verilog code can be tedious. Therefore, we use a python script that accepts an arbitrary adjacency matrix as its input and writes the verilog code that implements the network described by that adjacency matrix on the FPGA.

We chose the Altera Cyclone V GT as our FPGA because the Cyclone V GT Development Board was the least expensive board that provides both a PCIe slot (for streaming data at high speeds from the FPGA to a PC) and the capability for interfacing with both an ADC and DAC. The Cyclone V has a High-Speed Mezzanine Card (HSMC) interface, which allows us to attach the Terasic Data

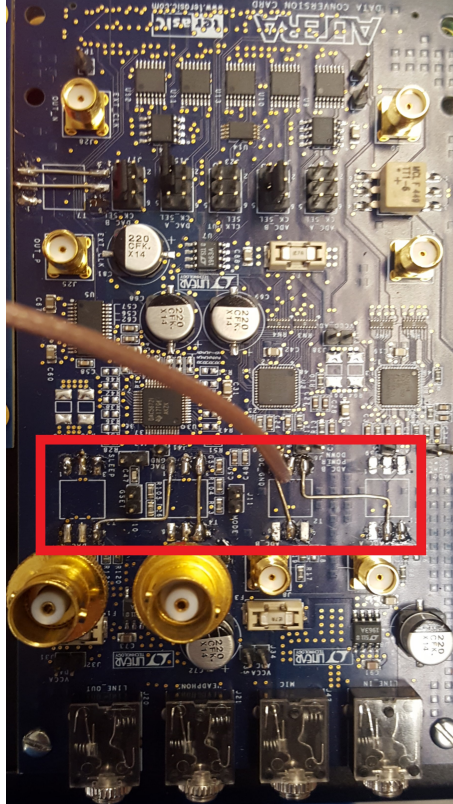


Figure A.1: Data Conversion Card modifications. This is a photograph of the Terasic Data Conversion Card used in our experiment. The transformers that come on the card are removed, and they are replaced with wiring that directly connects the ADC and DAC pins to the SMA outputs. The wiring that we added is denoted in the figure by the red box.

Conversion Card to our FPGA.

The Data Conversion Card (Fig. [A.1](#)) is an extension card with two 14-bit ADCs (Analog Devices AD9254) and two 14-bit DACs (Texas Instruments DAC5672), along other components that are not relevant for this experiment. The Data Conversion Card is designed such that the input of the ADCs and the output of the DACs are transformer coupled. These transformers were a problem because we needed our converters to be DC-coupled so that there would be no unintended time-invariant (in this case, high-pass) filtering. Therefore, we removed the transformers from the Data Conversion Card. This created a new problem: The ADCs require differential inputs and the DACs require differential outputs. We wired the two differential inputs to one of the ADCs directly to two on-card SMA ports. Similarly, we wired two differential outputs of one of the DACs directly to two on-card SMA ports. These modifications are highlighted by the red box in the photograph of the Data Conversion Card in Fig. [A.1](#).

As mentioned above, our ADC requires a differential input; however, the photoreceiver output (which serves as the input to the ADC) is single-ended. Therefore, we use a unity gain single-ended to differential amplifier (Analog Devices AD8138) to convert the single-ended output of the photoreceiver to a differential signal suitable for input to the ADC.

Similarly, our DAC provides a differential output; however the intensity modulator requires a single-ended input. Therefore we use a preamplifier (Stanford Research Systems Model SR560) to convert the differential output of the DAC to a single-ended voltage referenced to ground. The preamplifier also provides a voltage

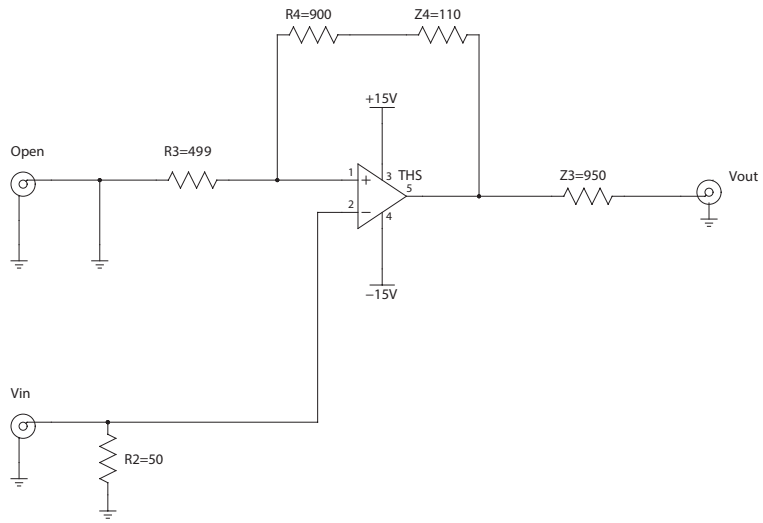


Figure A.2: Non-inverting amplifier circuit diagram. This amplifier is placed after the SR560 differential amplifier as described in Section 4.4.3 and provides an additional gain of 3. The operational amplifier we use is the Texas Instruments THS4631DDA. This circuit is implemented on the THS4631DDA EVM board from Texas Instruments. The resistor numbering is the same as that on the board's reference manual.



gain of 10dB. We then use a single-ended amplifier (Texas Instruments THS4631 configured as a non-inverting amplifier as in Fig. A.2) to provide a further voltage gain of 3.

We also had to make a few experimental modifications related to the electro-optic intensity modulator. As mentioned previously, the modulator has two electrical inputs: a DC input and a RF input. The DC input is high-impedance, but the RF input is typically terminated with a  $50\ \Omega$  resistor to allow for impedance matching with high-frequency RF drivers. As mentioned previously, our set-up needs to allow for DC signals to be applied to the RF port. The voltages being applied to the RF port can be up to  $2V_{\pi,RF} = 4.80\text{ V}$ . Therefore, the power dissipated in the  $50\ \Omega$  resistor can be quite high, on the order of  $0.25\text{ W}$ . This can heat the resistor. In order to avoid this, we opened up the modulator and removed the resistor, so that now the RF input port is also a high-impedance port. Once this was done, we found that the modulator behaved much more stably and repeatably.

We had to make two additional modifications due to the specified voltage limits of the RF port of the modulator. The RF port is rated for  $\pm 5\text{ V}$ ; however, depending on the system parameters and resulting dynamics,  $x$  (and therefore  $v_{RF}$ ) can be any real number. In order to keep  $v_{RF}$  within the manufacturers specifications and to circumvent the limited output voltage range of the DAC, we take advantage of the fact that the modulator's nonlinearity is invariant when taken modulo  $4V_{\pi}$ . We implement this modulo in the FPGA by taking  $x$  modulo  $2\pi$ . Even with this modulo operation, the voltage range does not fit into the manufacturers specifications: for  $x \in [0, 2\pi)$ ,  $v_{RF} \in [0, 4V_{\pi,RF}) = [0, 9.60\text{V})$ . In order to resolve this, we shift  $v_{RF} \rightarrow$

$v_{RF} - 2V_{\pi,RF}$ , taking advantage of the fact that the modulator's nonlinearity is invariant under a voltage shift of  $2V_{\pi}$ . This results in  $v_{RF} \in [-2V_{\pi,RF}, 2V_{\pi,RF}) = [-4.80 \text{ V}, 4.80 \text{ V})$ , a voltage range within the manufacturers specifications. This shift is also implemented in the FPGA.

### **Fixed point notation in the FPGA**

All arithmetic in the FPGA is integer arithmetic. In order to represent real numbers like the normalized intensity  $I$  or the normalized voltage  $x$ , we use signed fixed point arithmetic. In signed fixed point arithmetic, real numbers are represented as  $M$ -bit signed integers, where the most significant bit (MSB) determines the sign and the remaining  $M-1$  bits determine the value. In order to represent non-integer numbers, a fixed point can be chosen such that all bits (except the MSB) to the left of that point represent the integer part of the number, and all bits to the right of that point represent the fractional part of the number. This can be denoted as  $P.Q$  signed fixed point notation, where  $Q$  is an integer giving the number of decimal bits and  $P = M - Q$  gives the number of remaining bits. In our implementation, we use 5.10 signed fixed point notation, meaning that the MSB determines the sign, the next four MSBs determine the integer part, and the 10 least significant bits (LSBs) determine the decimal part of the number being represented. For example, in 5.10 fixed point notation, the decimal number  $5.375_{10}$  is represented as  $00101_{(.)}0110000000_2$ , where numbers to the right of the decimal place give the fractional part and numbers to the left of the decimal place give the sign and integer part.

The particular type of signed arithmetic we use is called twos-complement. In

twos complement notation, positive integers have zero as the MSB, and the rest of the integer is represented in standard binary notation. For example, the positive integer  $10_{10}$  in 15-bit twos complement notation is represented as  $000000000001010_2$ . A negative number is represented by writing  $2^M - K$  in binary notation, where  $K$  is the absolute value of the negative number to be represented. For example, to represent the negative integer  $-10_{10}$  in 15-bit twos complement notation, one first computes  $32768_{10} - 10_{10} = 32758_{10}$ , then converts the decimal number  $32758_{10}$  into its binary equivalent  $11111111110110_2$ . The idea behind this formalism is that the sum of  $K$  and  $-K$  is zero *when the sum is written as an  $M$ -bit integer* (i.e.  $K + (-K) = 0 \text{ (modulo } 2^M)$ ). For example,  $10_{10} + (-10_{10}) = 000000000001010_2 + 11111111110110_2 = 100000000000000_2$ . Notice that the “1” in the sum is the  $16^{th}$  bit, which means that it is dropped when written as a 15-bit integer. Twos complement notation can represent integers in the range  $[-2^{M-1}, 2^{M-1} - 1]$ . Twos complement notation can be combined with the fixed point notation described above to represent real numbers with a precision of  $2^{-Q}$ , where  $Q$  is the number of decimal bits.

We chose to represent all numbers used in computations on the FPGA as 5.10 signed fixed point notation for practical reasons. As mentioned above, the ADC reads in the normalized intensity  $I$  with 10-bits of precision, so 10 bits are needed to represent  $I$ . As shown in Eq. 4.35, in order to compute  $x$ ,  $I$  is multiplied by  $\beta$ , which can, in principle, be any real number. We allocate an additional 5 bits (1 sign bit, 4 integer bits), allowing reliable computations to be performed with  $\beta$  in the range  $[-16, 16)$  for the case of only a single node. To implement networks with

large  $\beta$  or  $\sigma$  or with nodes with large in-degree, it may be necessary to use a larger number of integer bits in the computation of  $x$ , and then to take  $x$  modulo  $2\pi$  when determining the voltage to output from the DAC, as discussed previously.

The 5.10 signed fixed point notation was chosen because it provides a reasonable compromise between sufficient resolution to prevent overflows and fill up the 14 bit DAC output, while not requiring too many FPGA resources. However, as mentioned above, the normalized voltage  $x$  to be output by the DAC is taken modulo  $2\pi$  in order to keep  $v_{RF}$  within the manufacturer's specifications. Therefore, if we simply output  $x$  from the DAC, the full 14-bit dynamic range of the DAC will not be used. In order to use the full dynamic range of the DAC, we scale  $x$  by  $(2^{14} - 1)/2\pi$  immediately before outputting it to the DAC.

The ADC has 14 bits of resolution, but the MSB is dedicated to detecting the sign of the measured voltage. Since the voltage that we are measuring is proportional to the (non-negative) intensity of the light that passes through the modulator, we can use only the 13 LSBs of the ADC. Equation 4.35, which is implemented in the FPGA, uses the normalized intensity  $I = \mathcal{I}/\mathcal{I}_{max} \in [0, 1]$ . This is implemented by setting the laser power so that the electrical output of the photoreceiver when the intensity modulator is maximally transmitting fills up the 13 LSBs of the ADC. This occurs when the maximum electrical output of the photoreceiver is 520 mV. The electrical output of the photoreceiver is read into the FPGA in 5.10 signed fixed point notation (where the sign bit is always 0). This means only 10 of the ADC bits are actually utilized; the 3 LSBs are discarded. This quantization error (1 part in 1024) is sufficiently small that it does not contribute significantly to the noise, as

discussed below.

### FPGA timing

The timing of operations is crucial to obtain reliable performance from an FPGA. Often, the timing of FPGA operations are controlled by one or more binary clocks that oscillate periodically at a given frequency. This is the case in our set-up. The Altera Cyclone V GT Development Board has an on-board 50 MHz crystal oscillator that is hard-wired to an FPGA input pin. This crystal oscillator can then be used as the input to a phase-locked loop (PLL), which can multiply and/or divide the frequency to obtain a desired clock frequency. We use two different PLLs to generate two clocks, one with frequency  $f_r$  and the other with frequency  $f_r/N$ , where  $N$  is the number of nodes we want in our network. Typically, we choose  $f_r = 20$  kHz; however, our set-up has the potential to go up to about  $f_r = 1$  MHz. With state-of-the-art amplifiers, ADCs, DACs, and FPGA, such a system could obtain  $f_r$  on the order of 1 GHz or higher.

A timing diagram for our FPGA network set-up is shown in Fig. [A.3](#). The “node clock” oscillates at  $f_r$  and keeps the “node time”  $k$  from Eq. [4.4](#). On the positive edge of the “node clock,”  $x_i[n]$  is transmitted to the PC for storage and output by the DAC. On the negative edge of the “node clock,”  $I_i[n]$  is measured by the ADC and read in to the FPGA. The “network clock” keeps the “network time”  $n$  and triggers the updating of the network state  $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$  according to Eq. [4.35](#).

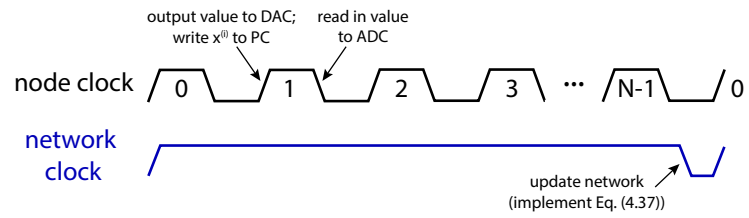


Figure A.3: FPGA timing diagram. There are two clocks, the “node clock,” which corresponds to  $k$ , and the “network clock,” which corresponds to  $n$ .

## Appendix B: Sample verilog code for two-node delay network

```
module two_node_network(
input CLK_50M, //this is assigned to a 50 MHz on-board clk
input [7:0] USER_SW, //on-board DIP Switches
input USER_PB_0, //on-board push-buttons
input USER_PB_1,
input USER_PB_2,
output [13:0] DAC, //DAC pins
output DAC_CLK,
input [13:0] ADC, //ADC pins
output ADC_CLK,
output ADC_DCS, //set to high
output ADC_DFS, //does formatting for adc
output ADC_CS, //chip select...make it high
output ADC_OE, //output enable
input ADC_DCO //adc data clock output
);

/*
SUMMARY OF INTERFACING WITH THE FPGA I/Os:
SW_1: ON allows self-feedback; OFF means beta=0
SW_2: ON allows coupling; OFF means sigma=0
SW_5: ON means that push buttons change beta;
      OFF means that push buttons change sigma
SW_6 and SW_7: 00: increment=1 01:increment=10 10:increment =500
*/

parameter bitwidth = 14;
parameter totalbits = 28; //Nnodes * bitwidth

//this does the ADC settings
assign ADC_CS = 1'b1;
assign ADC_DCS = 1'b1;
assign ADC_OE = 1'b0;
assign ADC_DFS = 1'b0;
```

```

wire [13:0] betasf;
wire [13:0] sigmasf;

wire [bitwidth-1:0] I_ADC_2_storage;
wire MUX_select;
wire store_select;
wire [totalbits-1:0] I_storage_2_iterate;
wire [totalbits-1:0] phi_iterate_2_MUX;
wire [bitwidth-1:0] phi_mux_2_DAC;

wire fast_clk;
wire slow_clk;

wire [7:0] USER_sw;
assign USER_sw=~USER_SW; //DIP switches for our board are inverted

//generate clocks
//The PLL outputs two clocks such that
//the frequency of the slow_clk = fast_clk / Nnodes.
//This must be done using the Altera PLL IP Core Megafunction wizard.
pll1 pll_1(
    .refclk(clk_50M),
    .rst(),
    .outclk_1(fast_clk),
    .outclk_4(slow_clk),
    .locked()
);

//read I from ADC
assign ADC_CLK = CLK_50M;
assign DAC_CLK = CLK_50M;

reg [13:0] ADCreg;

always @(posedge ADC_DCO)
    ADCreg<=ADC;
//chop off MSBs so that I_ADC_2_storage is normalized to 1
//in 4.10 fixed point notation
assign I_ADC_2_storage[13:10] = 4'b0;
assign I_ADC_2_storage[9:0] = ADCreg[12:3];

//read ADC at negedge of fast clock
always @ (negedge fast_clk)
    case (MUX_select)

```



```

1'b0 : I_storage_2_iterate[13:0] = I_ADC_2_storage;
1'b1 : I_storage_2_iterate[27:14] = I_ADC_2_storage;
default : I_storage_2_iterate[13:0] = 14'b0;
endcase
endmodule

//triggers at negedge slow_clk
iterate_network(
.clk(slow_clk),
.betasf(betasf),
.sigmasf(sigmasf),
.I(I_storage_2_iterate),
.phi(phi_iterate_2_MUX),
.phi_out(phi_out_storage)
);

//determine which node we're on
always @(posedge fast_clk)
begin
if (MUX_select>=1)
MUX_select<=0;
else
MUX_select<=select+1;
end
endmodule

//create MUX to select correct output
always @(MUX_select)
case(MUX_select)
0 : phi_mux_2_DAC = phi_iterate_2_MUX[13:0];
1 : phi_mux_2_DAC = phi_iterate_2_MUX[27:14];
default : phi_mux_2_DAC = 14'b0;
endcase

//For Vpi=2.42 modulator, the DAC amplifier is calibrated so that
//DAC=0 --> -4.84V and DAC=2^14-1 --> 4.84V.
assign DAC = 14'b00010100010111*phi_mux_2_DAC;

//allows me to change beta using pushbuttons
set_value set_beta(
.onoff(USER_sw[1]),
.enable(USER_sw[5]),
.reset(USER_PB_0),
.increase_clk(USER_PB_2),
.decrease_clk(USER_PB_1),

```

```

.sf(betasf),
.incrementSW({USER_sw[6],USER_sw[7]})
);

//allows me to change sigma using pushbuttons
set_value set_sigma(
.onoff(USER_sw[2]),
.enable(~USER_sw[5]),
.reset(USER_PB_0),
.increase_clk(USER_PB_2),
.decrease_clk(USER_PB_1),
.sf(sigmasf),
.incrementSW({USER_sw[6],USER_sw[7]})
);

endmodule

```

```

module iterate_network(
input clk,
input [13:0] betasf,
input [13:0] sigmasf,
input [27:0] I,
output [27:0] phi,
output [35:0] phi_out
);

reg [27:0] temp_feedback0;
reg [27:0] temp_feedback1;

reg [27:0] temp_sum0;
reg [27:0] temp_sum1;

reg [27:0] temp_coupling0;
reg [27:0] temp_coupling1;

reg [17:0] temp_phi0;
reg [17:0] temp_phi1;

assign phi_out = {temp_phi1, temp_phi0};

// node number 0
always @(negedge clk)
begin
temp_feedback0 = betasf * I[13:0];
temp_sum0 = I[27:14];
temp_coupling0 = sigmasf * temp_sum0;
temp_phi0 = temp_feedback0[27:10]+temp_coupling0[27:10];
end

// node number 1
always @(negedge clk)
begin
temp_feedback1 = betasf * I[27:14];
temp_sum1 = I[13:0];
temp_coupling1 = sigmasf * temp_sum1;
temp_phi1 = temp_feedback1[27:10]+temp_coupling1[27:10];
end

mod2pi mod2pi0(
.value(temp_phi0),
.out(phi[13:0]));

```

```
mod2pi mod2pi1(  
  .value(temp_phi1),  
  .out(phi[27:14]));  
  
endmodule
```

```

module set_value(
input onoff,
input enable,
input reset,
input increase_clk,
input decrease_clk,
input [1:0] incrementSW,
output [13:0] sf //scalefactor
);

wire [13:0] increment;
assign increment = ~incrementSW[1] ? (~incrementSW[0] ? 1 : 10) : 500;

reg [31:0] plus;
reg [31:0] minus;
wire [13:0] val;
assign val = plus - minus;
assign sf = (onoff) ? val : 14'b0;

always @(negedge increase_clk)
begin
if(enable)
begin
if (reset==0)
begin
plus <= minus;
end
else
begin
plus <= plus + increment;
end
end
end

always @(negedge decrease_clk)
begin
if(enable)
minus <= minus + increment;
end
endmodule

```

## Appendix C: Adjacency matrices from Chapter 5

The adjacency matrix for the symmetric network shown in Fig. 5.10a is

[illegible]

The adjacency matrix for the optimized network shown in Fig. 5.10b is

$$\mathbf{A}_{\text{opt}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{C.2})$$







## Appendix D: Derivation of the $h_0$ limit

In Eq. 6.2 we provide the information theoretical limit for the maximum entropy that can be harvested with a given probability distribution  $p(x)$  with a signal and measurement bandwidth of  $BW$ . Here we derive this from a previous result and explain how to calculate it in practice.

Gaspard and Wang [215] give the upper limit of  $\epsilon$ -entropy per sample as

$$H_0(\epsilon) = -\log_2(\epsilon) - \int dx p(x) \log_2 p(x) + O(\epsilon), \quad (\text{D.1})$$

where  $p(x)$  is the PDF of the signal from which entropy is being harvested. One can either use a theoretical PDF or estimate the PDF from an experimentally measured histogram for  $p(x)$ . A theoretical PDF is preferable, since one can calculate the integral exactly. We also know that the sampling rate is limited by information theory to the Nyquist rate  $f_{max} = 2BW$ . Here,  $BW$  is the limiting bandwidth, which is the minimum of all relevant bandwidths (signal bandwidth, detector bandwidth, digitizer bandwidth, etc.). Thus, as  $\epsilon \rightarrow 0$

$$h_0(\epsilon) = \min(\tau^{-1}, 2BW) \left( -\log_2(\epsilon) - \int dx p(x) \log_2 p(x) \right). \quad (\text{D.2})$$

In practice, it is often easiest to use Eq. D.2 to determine the  $h_0$  limit. However, the limit can be understood intuitively by writing it in terms of  $N_\epsilon$  and the Kullback-Leibler divergence, as we now show.

The number of bits per sample  $N_\epsilon = \log_2(\frac{b-a}{\epsilon})$ , where  $a$  and  $b$  are the end points of the measurement range of the digitizer. If we define  $U = \frac{1}{b-a}$ , we can write  $N_\epsilon = -\log_2(U\epsilon)$  and

$$h_0(\epsilon) = \min(\tau^{-1}, 2BW) \left( N_\epsilon + \log_2 U - \int dx p(x) \log_2 p(x) \right). \quad (\text{D.3})$$

Now define  $u(x)=U$  for  $a \leq x \leq b$  and  $u(x) = 0$  for all other  $x$ .  $p(x)$  is also only non-zero for  $a \leq x \leq b$ . Since  $\int p(x)dx=1$ , we can write  $\log_2 U = \int p(x) \log_2 U dx$ . We can then combine the integrals:

$$h_0(\epsilon) = \min(\tau^{-1}, 2BW) \left( N_\epsilon - \int dx p(x) \log_2 \left( \frac{p(x)}{u(x)} \right) \right). \quad (\text{D.4})$$

The Kullback-Leibler divergence is defined as

$D_{KL}(p(x)||u(x)) \equiv \int dx p(x) \log_2 \left( \frac{p(x)}{u(x)} \right)$ , giving us Eq. 6.2:

$$h_0 = \min(\tau^{-1}, 2BW) \left( N_\epsilon - D_{KL}(p(x)||u(x)) \right). \quad (\text{D.5})$$

## Bibliography

- [1] Henri Poincaré and Francis Maitland. *Science and method*. Courier Corporation, 2003.
- [2] Balthasar van der Pol. Forced oscillations in a circuit with non-linear resistance (reception with reactive triode). *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 3(13):65–80, 1927.
- [3] Mary L Cartwright and John E Littlewood. On non-linear differential equations of the second order: I. The equation  $\ddot{y} + \kappa(1 - y^2)\dot{y} + y = b\lambda\kappa \cos(\lambda t + a)$ ,  $\kappa$  large. *Journal of the London Mathematical Society*, 1(3):180–189, 1945.
- [4] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.
- [5] Robert M May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459, 1976.
- [6] Hermann Haken. Analogy between higher instabilities in fluids and lasers. *Physics Letters A*, 53(1):77–78, 1975.
- [7] Tien-Yien Li and James A Yorke. Period three implies chaos. *The American Mathematical Monthly*, 82(10):985–992, 1975.
- [8] Brian R Hunt and Edward Ott. Defining chaos. *Chaos*, 25(9):097618, 2015.
- [9] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018.
- [10] Edward Ott. *Chaos in dynamical systems*. Cambridge University Press, 2002.
- [11] Hirokazu Fujisaka and Tomoji Yamada. Stability theory of synchronized motion in coupled-oscillator systems. *Progress of Theoretical Physics*, 69(1):32–47, 1983.

- [12] Arkady S Pikovskii. Synchronization and stochastization of array of self-excited oscillators by external noise. *Radiophysics and Quantum Electronics*, 27(5):390–395, 1984.
- [13] Louis M Pecora and Thomas L Carroll. Synchronization in chaotic systems. *Physical Review Letters*, 64(8):821–824, 1990.
- [14] Arkady Pikovsky, Michael Rosenblum, Jurgen Kurths, and Jürgen Kurths. *Synchronization: a universal concept in nonlinear sciences*, volume 12. Cambridge University Press, 2003.
- [15] Stanislaw M Ulam. On combination of stochastic and deterministic processes. *Bulletin of the American Mathematical Society*, 53:1120, 1947.
- [16] Toni Stojanovski, Johnny Pihl, and Ljupco Kocarev. Chaos-based random number generators. part ii: practical realization. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(3):382–385, 2001.
- [17] Atsushi Uchida, Kazuya Amano, Masaki Inoue, Kunihiro Hirano, Sunao Naito, Hiroyuki Someya, Isao Oowada, Takayuki Kurashige, Masaru Shiki, Shigeru Yoshimori, et al. Fast physical random bit generation with chaotic semiconductor lasers. *Nature Photonics*, 2(12):728, 2008.
- [18] FT Arecchi, G Giacomelli, A Lapucci, and R Meucci. Two-dimensional representation of a delayed dynamical system. *Physical Review A*, 45(7):R4225, 1992.
- [19] Laurent Larger, Bogdan Penkovsky, and Yuri Maistrenko. Virtual chimera states for delayed-feedback systems. *Physical Review Letters*, 111(5):054103, 2013.
- [20] Laurent Larger, Bogdan Penkovsky, and Yuri Maistrenko. Laser chimeras as a paradigm for multistable patterns in complex systems. *Nature Communications*, 6:7752, 2015.
- [21] Lennert Appeltant, Miguel Cornelles Soriano, Guy Van der Sande, Jan Danckaert, Serge Massar, Joni Dambre, Benjamin Schrauwen, Claudio R Mirasso, and Ingo Fischer. Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468, 2011.
- [22] Laurent Larger, Miguel C Soriano, Daniel Brunner, Lennert Appeltant, Jose M Gutiérrez, Luis Pesquera, Claudio R Mirasso, and Ingo Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics Express*, 20(3):3241–3249, 2012.
- [23] Joseph D Hart, Kanika Bansal, Thomas E Murphy, and Rajarshi Roy. Experimental observation of chimera and cluster states in a minimal globally coupled network. *Chaos*, 26(9):094801, 2016.

- [24] B Koerner. Russians engineer a brilliant slot machine cheat and casinos have no fix. *Wired*, 2017.
- [25] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. NIST special publication 800-90B. <https://csrc.nist.gov/publications/detail/sp/800-90b/final>, 2018.
- [26] Elaine Barker and John Kelsey. NIST second draft special publication 800-90B. [https://csrc.nist.gov/CSRC/media//Publications/sp/800-90b/draft/documents/sp800-90b\\_second\\_draft.pdf](https://csrc.nist.gov/CSRC/media//Publications/sp/800-90b/draft/documents/sp800-90b_second_draft.pdf), (2016).
- [27] Yanne Chembo Kouomou. *Nonlinear Dynamics of Semiconductor Laser Systems with Feedback: Applications to Optical Chaos Cryptography, Radar Frequency Generation, and Transverse Mode Control*. PhD thesis, Universitat de les Illes Balears, 2006.
- [28] Adam B Cohen. *Synchronization and prediction of chaotic dynamics on networks of optoelectronic oscillators*. PhD thesis, University of Maryland, College Park, 2011.
- [29] Bhargava Ravoori. *Synchronization of Chaotic Optoelectronic Oscillators: Adaptive Techniques and the Design of Optimal Networks*. PhD thesis, University of Maryland, College Park, 2011.
- [30] Caitlin Rose Sanford Williams. *Optoelectronic experiments on random bit generators and coupled dynamical systems*. PhD thesis, University of Maryland, College Park, 2013.
- [31] Roy Lang and Kohroh Kobayashi. External optical feedback effects on semiconductor injection laser properties. *IEEE Journal of Quantum Electronics*, 16(3):347–355, 1980.
- [32] J Doyne Farmer. Chaotic attractors of an infinite-dimensional dynamical system. *Physica D*, 4(3):366–393, 1982.
- [33] M Le Berre, E Ressayre, A Tallet, HM Gibbs, DL Kaplan, and MH Rose. Conjecture on the dimensions of chaotic attractors of delayed-feedback dynamical systems. *Physical Review A*, 35(9):4020, 1987.
- [34] Thomas Erneux, Julien Javaloyes, Matthias Wolfrum, and Serhiy Yanchuk. Introduction to focus issue: Time-delay dynamics, 2017.
- [35] Michael C Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.
- [36] Yang Kuang. *Delay differential equations: with applications in population dynamics*, volume 191. Academic Press, 1993.

- [37] Kensuke Ikeda. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, 30(2):257–261, 1979.
- [38] S-N Chow, JK Hale, and W Huang. From sine waves to square waves in delay equations. *Proceedings of the Royal Society of Edinburgh Section A*, 120(3-4):223–229, 1992.
- [39] Lionel Weicker, Thomas Erneux, Otti DHuys, Jan Danckaert, Maxime Jacquot, Yanne Chembo, and Laurent Larger. Strongly asymmetric square waves in a time-delayed system. *Physical Review E*, 86(5):055201, 2012.
- [40] David Müller, Andreas Otto, and Günter Radons. Laminar chaos. *Physical Review Letters*, 120(8):084102, 2018.
- [41] Serhiy Yanchuk and Giovanni Giacomelli. Spatio-temporal phenomena in complex systems with time delays. *Journal of Physics A*, 50(10):103001, 2017.
- [42] G Giacomelli, R Meucci, A Politi, and FT Arecchi. Defects and spacelike properties of delayed dynamical systems. *Physical Review Letters*, 73(8):1099, 1994.
- [43] Serhiy Yanchuk and Giovanni Giacomelli. Pattern formation in systems with multiple delayed feedbacks. *Physical Review Letters*, 112(17):174103, 2014.
- [44] Giovanni Giacomelli, Francesco Marino, Michael A Zaks, and Serhiy Yanchuk. Coarsening in a bistable system with long-delayed feedback. *Europhysics Letters*, 99(5):58005, 2012.
- [45] J Javaloyes, T Ackemann, and A Hurtado. Arrest of domain coarsening via antiperiodic regimes in delay systems. *Physical Review Letters*, 115(20):203901, 2015.
- [46] Giovanni Giacomelli, Francesco Marino, Michael A Zaks, and Serhiy Yanchuk. Nucleation in bistable dynamical systems with long delay. *Physical Review E*, 88(6):062920, 2013.
- [47] Francesco Marino and Giovanni Giacomelli. Pseudo-spatial coherence resonance in an excitable laser with long delayed feedback. *Chaos*, 27(11):114302, 2017.
- [48] Marco Faggian, Francesco Ginelli, Francesco Marino, and Giovanni Giacomelli. Evidence of a critical phase transition in purely temporal dynamics with long-delayed feedback. *Physical Review Letters*, 120(17):173901, 2018.
- [49] Hyatt M Gibbs, Frederic A Hopf, DL Kaplan, and Richard L Shoemaker. Observation of chaos in optical bistability. *Physical Review Letters*, 46(7):474, 1981.

- [50] ANDREAS Neyer and Edgar Voges. Dynamics of electrooptic bistable devices with delayed feedback. *IEEE Journal of Quantum Electronics*, 18(12):2009–2015, 1982.
- [51] Michael Peil, Maxime Jacquot, Yanne Kouomou Chembo, Laurent Larger, and Thomas Erneux. Routes to chaos and multiple time scale dynamics in broadband bandpass nonlinear delay electro-optic oscillators. *Physical Review E*, 79(2):026208, 2009.
- [52] Y Chembo Kouomou, Pere Colet, Laurent Larger, and Nicolas Gastaud. Chaotic breathers in delayed electro-optical systems. *Physical Review Letters*, 95(20):203903, 2005.
- [53] Kristine E Callan, Lucas Illing, Zheng Gao, Daniel J Gauthier, and Eckehard Schöll. Broadband chaos generated by an optoelectronic oscillator. *Physical Review Letters*, 104(11):113901, 2010.
- [54] Bhargava Ravoori, Adam B Cohen, Jie Sun, Adilson E Motter, Thomas E Murphy, and Rajarshi Roy. Robustness of optimal synchronization in real networks. *Physical Review Letters*, 107(3):034102, 2011.
- [55] Caitlin RS Williams, Thomas E Murphy, Rajarshi Roy, Francesco Sorrentino, Thomas Dahms, and Eckehard Schöll. Experimental observations of group synchrony in a system of chaotic optoelectronic oscillators. *Physical Review Letters*, 110(6):064104, 2013.
- [56] Caitlin RS Williams, Francesco Sorrentino, Thomas E Murphy, and Rajarshi Roy. Synchronization states and multistability in a ring of periodic oscillators: Experimentally variable coupling delays. *Chaos*, 23(4):043117, 2013.
- [57] Joseph D Hart, Jan Philipp Pade, Tiago Pereira, Thomas E Murphy, and Rajarshi Roy. Adding connections can hinder network synchronization of time-delayed oscillators. *Physical Review E*, 92(2):022804, 2015.
- [58] Aaron Morgan Hagerstrom, Thomas Edward Murphy, and Rajarshi Roy. Harvesting entropy and quantifying the transition from noise to chaos in a photon-counting feedback loop. *Proceedings of the National Academy of Sciences*, 112(30):9258–9263, 2015.
- [59] X Steve Yao and Lute Maleki. Optoelectronic microwave oscillator. *JOSA B*, 13(8):1725–1735, 1996.
- [60] J-P Goedgebuer, Pascal Levy, Laurent Larger, C-C Chen, and William T Rhodes. Optical communication with synchronized hyperchaos generated electrooptically. *IEEE Journal of Quantum Electronics*, 38(9):1178–1183, 2002.

- [61] Apostolos Argyris, Dimitris Syvridis, Laurent Larger, Valerio Annovazzi-Lodi, Pere Colet, Ingo Fischer, Jordi Garcia-Ojalvo, Claudio R Mirasso, Luis Pesquera, and K Alan Shore. Chaos-based communications at high bit rates using commercial fibre-optic links. *Nature*, 438(7066):343, 2005.
- [62] Romain Martinenghi, Sergei Rybalko, Maxime Jacquot, Yanne K Chembo, and Laurent Larger. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Physical Review Letters*, 108(24):244101, 2012.
- [63] Laurent Larger. Complexity in electro-optic delay dynamics: modelling, design and applications. *Philosophical Transactions of the Royal Society A*, 371(1999):20120464, 2013.
- [64] Thomas E Murphy, Adam B Cohen, Bhargava Ravoori, Karl RB Schmitt, Anurag V Setty, Francesco Sorrentino, Caitlin RS Williams, Edward Ott, and Rajarshi Roy. Complex dynamics and synchronization of delayed-feedback nonlinear oscillators. *Philosophical Transactions of the Royal Society of London A*, 368(1911):343–366, 2010.
- [65] Laurent Larger, Pierre-Ambroise Lacourt, Stéphane Poinot, and Marc Hanna. From flow to map in an experimental high-dimensional electro-optic nonlinear delay oscillator. *Physical Review Letters*, 95(4):043903, 2005.
- [66] Alan V Oppenheim, Alan S Willsky, and S Hamid Nawab. Signals & systems. *Prentice-Hall, Inc., Upper Saddle River, NJ, USA*, 18:19–21, 1996.
- [67] Mélanie Grapinet, Vladimir Udaltsov, Maxime Jacquot, Pierre-Ambroise Lacourt, John M Dudley, and Laurent Larger. Experimental chaotic map generated by picosecond laser pulse-seeded electro-optic nonlinear delay dynamics. *Chaos*, 18(1):013110, 2008.
- [68] Christiaan Huygens. *Huygens Christiaan: oeuvres complètes*, volume 1. M. Nijhoff, 1897.
- [69] John Buck. Synchronous rhythmic flashing of fireflies. ii. *The Quarterly Review of Biology*, 63(3):265–289, 1988.
- [70] Charles S Peskin. Mathematical aspects of heart physiology. *Courant Institute of Mathematics*, 1975.
- [71] Yoshiki Kuramoto. *Chemical oscillations, waves, and turbulence*, volume 19. Springer Science & Business Media, 2012.
- [72] Steven H Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143(1):1–20, 2000.

- [73] Yoshiki Kuramoto and Dorjsuren Battogtokh. Coexistence of coherence and incoherence in nonlocally coupled phase oscillators. *Nonlinear Phenomena in Complex Systems*, 5(380), 2002.
- [74] Daniel M Abrams and Steven H Strogatz. Chimera states for coupled oscillators. *Physical Review Letters*, 93(17):174102, 2004.
- [75] VS Afraimovich, NN Verichev, and MI Rabinovich. Stochastic synchronization of oscillations in dissipative systems. *Radiofizika*, 29:1050–1060, 1986.
- [76] AR Volkovskii and NF Rulkov. Experimental study of bifurcations at the threshold for stochastic locking. *Soviet Technical Physics Lett*, 15:249–251, 1989.
- [77] Louis M Pecora and Thomas L Carroll. Synchronization of chaotic systems. *Chaos*, 25(9):097611, 2015.
- [78] Louis M Pecora and Thomas L Carroll. Master stability functions for synchronized coupled systems. *Physical Review Letters*, 80(10):2109, 1998.
- [79] Kenneth S Fink, Gregg Johnson, Tom Carroll, Doug Mar, and Lou Pecora. Three coupled oscillators as a universal probe of synchronization stability in coupled oscillator arrays. *Physical Review E*, 61(5):5080, 2000.
- [80] CM González, C Masoller, MC Torrent, and Jordi García-Ojalvo. Synchronization via clustering in a small delay-coupled laser network. *Europhysics Letters*, 79(6):64003, 2007.
- [81] Bhargava Ravoori, Adam B Cohen, Anurag V Setty, Francesco Sorrentino, Thomas E Murphy, Edward Ott, and Rajarshi Roy. Adaptive synchronization of coupled chaotic oscillators. *Physical Review E*, 80(5):056205, 2009.
- [82] Mark J Panaggio and Daniel M Abrams. Chimera states: coexistence of coherence and incoherence in networks of coupled oscillators. *Nonlinearity*, 28(3):R67, 2015.
- [83] Aaron M Hagerstrom, Thomas E Murphy, Rajarshi Roy, Philipp Hövel, Iryna Omelchenko, and Eckehard Schöll. Experimental observation of chimeras in coupled-map lattices. *Nature Physics*, 8(9):658, 2012.
- [84] Mark R Tinsley, Simbarashe Nkomo, and Kenneth Showalter. Chimera and phase-cluster states in populations of coupled chemical oscillators. *Nature Physics*, 8(9):662, 2012.
- [85] Erik Andreas Martens, Shashi Thutupalli, Antoine Fourrière, and Oskar Hallatschek. Chimera states in mechanical oscillator networks. *Proceedings of the National Academy of Sciences*, 110(26):10563–10567, 2013.



- [86] Evgeny A Viktorov, Tatiana Habruseva, Stephen P Hegarty, Guillaume Huyet, and Bryan Kelleher. Coherence and incoherence in an optical comb. *Physical Review Letters*, 112(22):224101, 2014.
- [87] Lennart Schmidt, Konrad Schönleber, Katharina Krischer, and Vladimir García-Morales. Coexistence of synchrony and incoherence in oscillatory media under nonlinear global coupling. *Chaos*, 24(1):013102, 2014.
- [88] Chandrakala Meena, Krishnamurthy Murali, and Sudeshna Sinha. Chimera states in star networks. *International Journal of Bifurcation and Chaos*, 26(09):1630023, 2016.
- [89] David P Rosin, Damien Rontani, Nicholas D Haynes, Eckehard Schöll, and Daniel J Gauthier. Transient scaling and resurgence of chimera states in networks of boolean phase oscillators. *Physical Review E*, 90(3):030902, 2014.
- [90] Peter Ashwin and Oleksandr Burylko. Weak chimeras in minimal networks of coupled phase oscillators. *Chaos*, 25(1):013106, 2015.
- [91] Gautam C Sethia and Abhijit Sen. Chimera states: the existence criteria revisited. *Physical Review Letters*, 112(14):144101, 2014.
- [92] Simbarashe Nkomo, Mark R Tinsley, and Kenneth Showalter. Chimera states in populations of nonlocally coupled chemical oscillators. *Physical Review Letters*, 110(24):244102, 2013.
- [93] Azamat Yeldesbay, Arkady Pikovsky, and Michael Rosenblum. Chimeralike states in an ensemble of globally coupled oscillators. *Physical Review Letters*, 112(14):144103, 2014.
- [94] Lennart Schmidt and Katharina Krischer. Clustering as a prerequisite for chimera states in globally coupled systems. *Physical Review Letters*, 114(3):034101, 2015.
- [95] Lennart Schmidt and Katharina Krischer. Chimeras in globally coupled oscillatory systems: From ensembles of oscillators to spatially continuous media. *Chaos*, 25(6):064401, 2015.
- [96] Hiroaki Daido and Kenji Nakanishi. Diffusion-induced inhomogeneity in globally coupled oscillators: Swing-by mechanism. *Physical Review Letters*, 96(5):054101, 2006.
- [97] Kunihiko Kaneko. Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements. *Physica D*, 41(2):137–172, 1990.
- [98] VK Chandrasekar, R Gopal, A Venkatesan, and M Lakshmanan. Mechanism for intensity-induced chimera states in globally coupled oscillators. *Physical Review E*, 90(6):062913, 2014.

- [99] Fabian Böhm, Anna Zakharova, Eckehard Schöll, and Kathy Lüdge. Amplitude-phase coupling drives chimera states in globally coupled laser networks. *Physical Review E*, 91(4):040901, 2015.
- [100] Lou Pecora, Tom Carroll, Gregg Johnson, Doug Mar, and Kenneth S Fink. Synchronization stability in coupled oscillator arrays: Solution for arbitrary configurations. *International Journal of Bifurcation and Chaos*, 10(02):273–290, 2000.
- [101] Alex Arenas, Albert Díaz-Guilera, Jurgen Kurths, Yamir Moreno, and Changsong Zhou. Synchronization in complex networks. *Physics Reports*, 469(3):93–153, 2008.
- [102] Francesco Sorrentino and Edward Ott. Network synchronization of groups. *Physical Review E*, 76(5):056114, 2007.
- [103] Andre Bergner, M Frasca, G Sciuto, A Buscarino, Eulalie Joelle Ngamga, L Fortuna, and Jürgen Kurths. Remote synchronization in star networks. *Physical Review E*, 85(2):026208, 2012.
- [104] Vincenzo Nicosia, Miguel Valencia, Mario Chavez, Albert Díaz-Guilera, and Vito Latora. Remote synchronization reveals network symmetries and functional modules. *Physical Review Letters*, 110(17):174102, 2013.
- [105] Lucia Valentina Gambuzza, Alessio Cardillo, Alessandro Fiasconaro, Luigi Fortuna, Jesus Gómez-Gardenes, and Mattia Frasca. Analysis of remote synchronization in complex networks. *Chaos*, 23(4):043103, 2013.
- [106] Vladimir N Belykh, Igor V Belykh, and Erik Mosekilde. Cluster synchronization modes in an ensemble of coupled chaotic oscillators. *Physical Review E*, 63(3):036216, 2001.
- [107] Thomas Dahms, Judith Lehnert, and Eckehard Schöll. Cluster and group synchronization in delay-coupled networks. *Physical Review E*, 86(1):016202, 2012.
- [108] Ian Stewart, Martin Golubitsky, and Marcus Pivato. Symmetry groupoids and patterns of synchrony in coupled cell networks. *SIAM Journal on Applied Dynamical Systems*, 2(4):609–646, 2003.
- [109] Martin Golubitsky, Ian Stewart, and Andrei Török. Patterns of synchrony in coupled cell networks with multiple arrows. *SIAM Journal on Applied Dynamical Systems*, 4(1):78–100, 2005.
- [110] William Stein. Sage: Software for algebra and geometry experimentation. <http://www.sagemath.org/sage/>; <http://sage.scipy.org/>, 2013.
- [111] The GAP Group. Gap: Groups, algorithms, and programming. <http://www.gap-system.org>, 2005.

- [112] Michael T Schaub, Neave O’Clery, Yazan N Billeh, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. Graph partitions and cluster synchronization in networks of oscillators. *Chaos*, 26(9):094821, 2016.
- [113] Louis M Pecora, Francesco Sorrentino, Aaron M Hagerstrom, Thomas E Murphy, and Rajarshi Roy. Cluster synchronization and isolated desynchronization in complex networks with symmetries. *Nature Communications*, 5(5079):5079, 2014.
- [114] Francesco Sorrentino, Louis M Pecora, Aaron M Hagerstrom, Thomas E Murphy, and Rajarshi Roy. Complete characterization of the stability of cluster synchronization in complex dynamical networks. *Science Advances*, 2(4):e1501737, 2016.
- [115] Igor Belykh and Martin Hasler. Mesoscale and clusters of synchrony in networks of bursting neurons. *Chaos*, 21(1):016106, 2011.
- [116] Aaron Hagerstrom. Network symmetries and synchronization. <https://sourceforge.net/projects/networksym/>, 2014.
- [117] F Hubertus, Firdaus E Udawadia, and Wlodek Proskurowski. An efficient qr based method for the computation of lyapunov exponents. *Physica D*, 101(1-2):1–16, 1997.
- [118] Jerzy Wojewoda, Krzysztof Czołczynski, Yuri Maistrenko, and Tomasz Kapitaniak. The smallest chimera state for coupled pendula. *Scientific reports*, 6:34329, 2016.
- [119] Peter J Menck, Jobst Heitzig, Norbert Marwan, and Jürgen Kurths. How basin stability complements the linear-stability paradigm. *Nature Physics*, 9(2):89–92, 2013.
- [120] Francisco A Rodrigues, Thomas K DM Peron, Peng Ji, and Jürgen Kurths. The Kuramoto model in complex networks. *Physics Reports*, 610:1–98, 2016.
- [121] Daniel M Abrams, Rennie Mirollo, Steven H Strogatz, and Daniel A Wiley. Solvable model for chimera states of coupled oscillators. *Physical Review Letters*, 101(8):084103, 2008.
- [122] Francesco Sorrentino and Louis Pecora. Approximate cluster synchronization in networks with symmetries and parameter mismatches. *Chaos*, 26(9):094823, 2016.
- [123] Joseph D Hart, Don C Schmadel, Thomas E Murphy, and Rajarshi Roy. Experiments with arbitrary networks in time-multiplexed delay systems. *Chaos*, 27(12):121103, 2017.

- [124] Abu Bakar Siddique, Louis Pecora, Joseph D Hart, and Francesco Sorrentino. Symmetry-and input-cluster synchronization in networks. *Physical Review E*, 97(4):042217, 2018.
- [125] Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, volume 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.
- [126] Mark J Panaggio, Daniel M Abrams, Peter Ashwin, and Carlo R Laing. Chimera states in networks of phase oscillators: the case of two small populations. *Physical Review E*, 93(1):012218, 2016.
- [127] K Ikeda and K Matsumoto. Information theoretical characterization of turbulence. *Physical Review Letters*, 62(19):2265, 1989.
- [128] Giovanni Giacomelli and Antonio Politi. Relationship between delayed and spatially extended dynamical systems. *Physical Review Letters*, 76(15):2686, 1996.
- [129] Matthias Wolfrum and Serhiy Yanchuk. Eckhaus instability in systems with large delay. *Physical Review Letters*, 96(22):220201, 2006.
- [130] Serhiy Yanchuk and Giovanni Giacomelli. Dynamical systems with multiple long-delayed feedbacks: Multiscale analysis and spatiotemporal equivalence. *Physical Review E*, 92(4):042903, 2015.
- [131] Yvan Paquot, Francois Duport, Antoneo Smerieri, Joni Dambre, Benjamin Schrauwen, Marc Haelterman, and Serge Massar. Optoelectronic reservoir computing. *Scientific Reports*, 2:287, 2012.
- [132] Daniel Brunner, Miguel C Soriano, Claudio R Mirasso, and Ingo Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4:1364, 2013.
- [133] Piotr Antonik, Marc Haelterman, and Serge Massar. Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems. *Physical Review Applied*, 7(5):054014, 2017.
- [134] S Lepri, G Giacomelli, A Politi, and FT Arecchi. High-dimensional chaos in delayed dynamical systems. *Physica D*, 70(3):235–249, 1994.
- [135] Herbert Jaeger. The echo state approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.
- [136] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

- [137] David Verstraeten, Benjamin Schrauwen, Dirk Stroobandt, and Jan Van Campenhout. Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, 95(6):521–528, 2005.
- [138] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [139] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.
- [140] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018.
- [141] Daniel Brunner, Miguel C Soriano, Claudio R Mirasso, and Ingo Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4:1364, 2013.
- [142] Nicholas D Haynes, Miguel C Soriano, David P Rosin, Ingo Fischer, and Daniel J Gauthier. Reservoir computing with a single time-delay autonomous boolean node. *Physical Review E*, 91(2):020801, 2015.
- [143] Laurent Larger, Antonio Baylón-Fuentes, Romain Martinenghi, Vladimir S Udaltsov, Yanne K Chembo, and Maxime Jacquot. High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Physical Review X*, 7(1):011015, 2017.
- [144] Daniel Brunner and Ingo Fischer. Reconfigurable semiconductor laser networks based on diffractive coupling. *Optics Letters*, 40(16):3854–3857, 2015.
- [145] Julian Bueno, Sheler Maktoobi, Luc Froehly, Ingo Fischer, Maxime Jacquot, Laurent Larger, and Daniel Brunner. Reinforcement learning in a large-scale photonic recurrent neural network. *Optica*, 5(6):756–760, 2018.
- [146] Lionel Weicker, Thomas Erneux, Otti D’Huys, Jan Danckaert, Maxime Jacquot, Yanne Chembo, and Laurent Larger. Slow-fast dynamics of a time-delayed electro-optic oscillator. *Philosophical Transactions of the Royal Society A*, 371(1999):20120459, 2013.
- [147] D Brunner, B Penkovsky, R Levchenko, E Schöll, L Larger, and Y Maistrenko. Two-dimensional spatiotemporal complexity in dual-delayed nonlinear feedback systems: Chimeras and dissipative solitons. *Chaos*, 28(10):103106, 2018.
- [148] Peter J Menck, Jobst Heitzig, Jürgen Kurths, and Hans Joachim Schellnhuber. How dead ends undermine power grid stability. *Nature Communications*, 5:3969, 2014.

- [149] Takashi Nishikawa and Adilson E Motter. Symmetric states requiring system asymmetry. *Physical Review Letters*, 117(11):114101, 2016.
- [150] Francesco Sorrentino, Mario Di Bernardo, Franco Garofalo, and Guanrong Chen. Controllability of complex networks via pinning. *Physical Review E*, 75(4):046103, 2007.
- [151] Takashi Nishikawa and Adilson E. Motter. Symmetric states requiring system asymmetry. *Physical Review Letters*, 117:114101, 2016.
- [152] Yuanzhao Zhang, Takashi Nishikawa, and Adilson E. Motter. Asymmetry-induced synchronization in oscillator networks. *Physical Review E*, 95(6):062215, 2017.
- [153] Yuanzhao Zhang and Adilson E. Motter. Identical synchronization of nonidentical oscillators: When only birds of different feathers flock together. *Nonlinearity*, 30, 2017.
- [154] Young Sul Cho, Takashi Nishikawa, and Adilson E Motter. Stable chimeras and independently synchronizable clusters. *Physical Review Letters*, 119(8):084101, 2017.
- [155] Takashi Nishikawa and Adilson E Motter. Maximum performance at minimum cost in network synchronization. *Physica D*, 224(1):77–89, 2006.
- [156] Mauricio Barahona and Louis M Pecora. Synchronization in small-world systems. *Physical Review Letters*, 89(5):054101, 2002.
- [157] Zhongkui Li, Zhisheng Duan, Guanrong Chen, and Lin Huang. Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Transactions Circuits and Systems I*, 57(1):213–224, 2010.
- [158] Takashi Nishikawa and Adilson E Motter. Network synchronization landscape reveals compensatory structures, quantization, and the positive effect of negative interactions. *Proceedings of the National Academy of Sciences U.S.A.*, 107(23):10342–10347, 2010.
- [159] Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.
- [160] Brendan D McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.
- [161] Takashi Nishikawa and Adilson E Motter. Synchronization is optimal in non-diagonalizable networks. *Physical Review E*, 73(6):065106, 2006.
- [162] Simulated annealing code to improve synchronizability through minimal link rewiring or removal: [https://github.com/y-z-zhang/optimize\\_sym\\_cluster/](https://github.com/y-z-zhang/optimize_sym_cluster/).

- [163] Andrew J Whalen, Sean N Brennan, Timothy D Sauer, and Steven J Schiff. Observability and controllability of nonlinear networks: The role of symmetry. *Physical Review X*, 5(1):011005, 2015.
- [164] John Von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12(36-38):1, 1951.
- [165] PicoQuant. *PQRNG 150 (Quantum Random Number Generator)*, 2015.
- [166] ID Quantique. *Quantis Datasheet*, 10 2013.
- [167] Richard Hughes and Jane Nordholt. Strengthening the security foundation of cryptography with Whitewood’s quantum-powered entropy engine. [www.whitewoodencryption.com](http://www.whitewoodencryption.com), (2016).
- [168] Mike Hamburg, Paul Kocher, and Mark E Marson. Analysis of Intels Ivy Bridge digital random number generator. *Online: [http://www.cryptography.com/public/pdf/Intel\\_TRN\\_G\\_Report\\_20120312.pdf](http://www.cryptography.com/public/pdf/Intel_TRN_G_Report_20120312.pdf)*, 2012.
- [169] David P. Rosin, Damien Rontani, and Daniel J. Gauthier. Ultrafast physical generation of random numbers using hybrid boolean networks. *Physical Review E*, 87:040902, Apr 2013.
- [170] Myunghwan Park. *Chaotic oscillations in CMOS integrated circuits*. PhD thesis, University of Maryland, College Park, 2013.
- [171] Markus Dichtl and Jovan Dj Golić. High-speed true random number generation with logic gates only. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 45–62. Springer, 2007.
- [172] Knut Wold and Chik How Tan. Analysis and enhancement of random number generator in FPGA based on oscillator rings. *International Journal of Reconfigurable Computing*, 2009:4, 2009.
- [173] Michael A Wayne, Evan R Jeffrey, Gleb M Akselrod, and Paul G Kwiat. Photon arrival time quantum random number generation. *Journal of Modern Optics*, 56(4):516–522, 2009.
- [174] Michael Wahl, Matthias Leifgen, Michael Berlin, Tino Röhlicke, Hans-Jürgen Rahn, and Oliver Benson. An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements. *Applied Physics Letters*, 98(17):171105, 2011.
- [175] Qiurong Yan, Baosheng Zhao, Qinghong Liao, and Nanrun Zhou. Multi-bit quantum random number generation by measuring positions of arrival photons. *Review of Scientific Instrumentation*, 85(10):103116, 2014.

- [176] Qiurong Yan, Baosheng Zhao, Zhang Hua, Qinghong Liao, and Hao Yang. High-speed quantum-random number generation by continuous measurement of arrival time of photons. *Review of Scientific Instrumentation*, 86(7):073113, 2015.
- [177] Alireza Marandi, Nick C Leindecker, Konstantin L Vodopyanov, and Robert L Byer. All-optical quantum random bit generation from intrinsically binary phase of parametric oscillators. *Optics Express*, 20(17):19322–19330, 2012.
- [178] MJ Collins, AS Clark, C Xiong, E Mägi, MJ Steel, and BJ Eggleton. Random number generation from spontaneous Raman scattering. *Applied Physics Letters*, 107(14):141112, 2015.
- [179] Caitlin RS Williams, Julia C Salevan, Xiaowen Li, Rajarshi Roy, and Thomas E Murphy. Fast physical random number generator using amplified spontaneous emission. *Optics Express*, 18(23):23584–23597, 2010.
- [180] Xiaowen Li, Adam B Cohen, Thomas E Murphy, and Rajarshi Roy. Scalable parallel physical random number generator based on a superluminescent led. *Optics Letters*, 36(6):1020–1022, 2011.
- [181] Apostolos Argyris, Evangelos Pikasis, Stavros Deligiannidis, and Dimitris Syvridis. Sub-Tb/s physical random bit generators based on direct detection of amplified spontaneous emission signals. *Journal of Lightwave Technology*, 30(9):1329–1334, 2012.
- [182] Wei Wei, Guodong Xie, Anhong Dang, and Hong Guo. High-speed and bias-free optical random number generator. *IEEE Photonics Technology Letters*, 24(6):437–439, 2012.
- [183] Taiki Yamazaki and Atsushi Uchida. Performance of random number generators using noise-based superluminescent diode and chaos-based semiconductor lasers. *Selected Topics in IEEE Journal of Quantum Electronics*, 19(4):0600309–0600309, 2013.
- [184] Lei Li, Anbang Wang, Pu Li, Hang Xu, Longsheng Wang, and Yuncai Wang. Random bit generator using delayed self-difference of filtered amplified spontaneous emission. *IEEE Photonics Journal*, 6(1):1–9, 2014.
- [185] Jianzhong Zhang, Mingjiang Zhang, Yi Liu, Pu Li, Xiaogang Yi, Mingtao Zhang, and Yuncai Wang. Fast random number generation with spontaneous emission noise of a single-mode semiconductor laser. *Laser Physics*, 26(11):115002, 2016.
- [186] Hong Guo, Wenzhuo Tang, Yu Liu, and Wei Wei. Truly random number generation based on measurement of phase noise of a laser. *Physical Review E*, 81(5):051137, 2010.



- [187] Bing Qi, Yue-Meng Chi, Hoi-Kwong Lo, and Li Qian. High-speed quantum random number generation by measuring phase noise of a single-mode laser. *Optics Letters*, 35(3):312–314, 2010.
- [188] Feihu Xu, Bing Qi, Xiongfeng Ma, He Xu, Haoxuan Zheng, and Hoi-Kwong Lo. Ultrafast quantum random number generation based on quantum phase fluctuations. *Optics Express*, 20(11):12366–12377, 2012.
- [189] ZL Yuan, M Lucamarini, JF Dynes, B Fröhlich, A Plews, and AJ Shields. Robust random number generation using steady-state emission of gain-switched laser diodes. *Applied Physics Letters*, 104(26):261112, 2014.
- [190] You-Qi Nie, Leilei Huang, Yang Liu, Frank Payne, Jun Zhang, and Jian-Wei Pan. The generation of 68 Gbps quantum random number by measuring laser phase fluctuations. *Review of Scientific Instrumentation*, 86(6):063105, 2015.
- [191] Jinlu Liu, Jie Yang, Zhengyu Li, Qi Su, Wei Huang, Bingjie Xu, and Hong Guo. 117 Gbits/s quantum random number generation with simple structure. *IEEE Photonics Technology Letters*, 2016.
- [192] Christian Gabriel, Christoffer Wittmann, Denis Sych, Ruifang Dong, Wolfgang Mauerner, Ulrik L Andersen, Christoph Marquardt, and Gerd Leuchs. A generator for unique quantum random numbers based on vacuum states. *Nature Photonics*, 4(10):711–715, 2010.
- [193] Yong Shen, Liang Tian, and Hongxin Zou. Practical quantum random number generator based on measuring the shot noise of vacuum states. *Physical Review A*, 81(6):063814, 2010.
- [194] Thomas Symul, SM Assad, and Ping K Lam. Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Applied Physics Letters*, 98(23):231103, 2011.
- [195] M Jofre, M Curty, F Steinlechner, G Anzolin, JP Torres, MW Mitchell, and V Pruneri. True random numbers from amplified quantum vacuum. *Optics Express*, 19(21):20665–20672, 2011.
- [196] Yicheng Shi, Brenda Chng, and Christian Kurtsiefer. Random numbers from vacuum fluctuations. *Applied Physics Letters*, 109(4):041101, 2016.
- [197] Philip J Bustard, Duncan G England, Josh Nunn, Doug Moffatt, Michael Spanner, Rune Lausten, and Benjamin J Sussman. Quantum random bit generation using energy fluctuations in stimulated Raman scattering. *Optics Express*, 21(24):29350–29357, 2013.
- [198] DG England, PJ Bustard, DJ Moffatt, J Nunn, R Lausten, and BJ Sussman. Efficient Raman generation in a waveguide: A route to ultrafast quantum random number generation. *Applied Physics Letters*, 104(5):051117, 2014.

- [199] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. Quantum random number generators. *Reviews of Modern Physics*, 89:015004, Feb 2017.
- [200] Ronald F Fox and Joel Keizer. Amplification of intrinsic fluctuations by chaotic dynamics in physical systems. *Physical Review A*, 43(4):1709, 1991.
- [201] C Bracikowski, RF Fox, and Rajarshi Roy. Amplification of intrinsic noise in a chaotic multimode laser system. *Physical Review A*, 45(1):403, 1992.
- [202] Ido Kanter, Yaara Aviad, Igor Reidler, Elad Cohen, and Michael Rosenbluh. An optical ultrafast random bit generator. *Nature Photonics*, 4(1):58–61, 2010.
- [203] Neus Oliver, Miguel Cornelles Soriano, David W Sukow, and Ingo Fischer. Fast random bit generation using a chaotic laser: approaching the information theoretic limit. *IEEE Journal of Quantum Electronics*, 49(11):910–918, 2013.
- [204] Martin Virte, Emeric Mercier, Hugo Thienpont, Krassimir Panajotov, and Marc Sciamanna. Physical random bit generation from chaotic solitary laser diode. *Optics Express*, 22(14):17271–17280, 2014.
- [205] Ryohsuke Sakuraba, Kento Iwakawa, Kazutaka Kanno, and Atsushi Uchida. Tb/s physical random bit generation with bandwidth-enhanced chaos in three-cascaded semiconductor lasers. *Optics Express*, 23(2):1470–1490, 2015.
- [206] Xi Tang, Zheng-Mao Wu, Jia-Gui Wu, Tao Deng, Jian-Jun Chen, Li Fan, Zhu-Qiang Zhong, and Guang-Qiong Xia. Tbits/s physical random bit generation based on mutually coupled semiconductor laser chaotic entropy source. *Optics Express*, 23(26):33130–33141, 2015.
- [207] T Butler, C Durkan, D Goulding, S Slepneva, B Kelleher, SP Hegarty, and G Huyet. Optical ultrafast random number generation at 1 Tb/s using a turbulent semiconductor ring cavity laser. *Optics Letters*, 41(2):388–391, 2016.
- [208] Anbang Wang, Longsheng Wang, Pu Li, and Yuncai Wang. Minimal-post-processing 320-Gbps true random bit generation using physical white chaos. *Optics Express*, 25(4):3153–3164, 2017.
- [209] Kazusa Ugajin, Yuta Terashima, Kento Iwakawa, Atsushi Uchida, Takahisa Harayama, Kazuyuki Yoshimura, and Masanobu Inubushi. Real-time fast physical random number generator with a photonic integrated circuit. *Optics Express*, 25(6):6511–6523, 2017.
- [210] Marc Sciamanna and K Alan Shore. Physics and applications of laser diode chaos. *Nature Photonics*, 9(3):151–162, 2015.
- [211] Atsushi Uchida. *Optical communication with chaotic lasers: applications of nonlinear dynamics and synchronization*. John Wiley & Sons, 2012.

- [212] Susumu Shinohara, Kenichi Arai, Peter Davis, Satoshi Sunada, and Takahisa Harayama. Chaotic laser based physical random bit streaming system with a computer application interface. *Optics Express*, 25(6):6461–6474, 2017.
- [213] Elaine Barker and John Kelsey. NIST draft special publication 800-90c, (2016).
- [214] Ned J Corron, Roy M Cooper, and Jonathan N Blakely. Entropy rates of low-significance bits sampled from chaotic physical systems. *Physica D*, 332:34–40, 2016.
- [215] Pierre Gaspard and Xiao-Jing Wang. Noise, chaos, and  $(\varepsilon, \tau)$ -entropy per unit time. *Physics Reports*, 235(6):291–343, 1993.
- [216] Elaine Barker, John Kelsey, et al. NIST special publication 800-90A, 2012.
- [217] Andrew Rukhin, Juan Soto, James Nechvatal, Elaine Barker, Stefan Leigh, Mark Levenson, David Banks, Alan Heckert, James Dray, San Vo, et al. NIST special publication 800-22, (2010).
- [218] George Marsaglia. Diehard: a battery of tests of randomness. See <http://stat.fsu.edu/geo/diehard.html>, 1996.
- [219] Emmanuel Desurvire. *Classical and quantum information theory: an introduction for the telecom scientist*. Cambridge University Press, 2009.
- [220] Fan-Yi Lin, Yuh-Kwei Chao, and Tsung-Chieh Wu. Effective bandwidths of broadband chaotic signals. *IEEE Journal of Quantum Electronics*, 48(8):1010–1014, 2012.
- [221] Robert H Walden. Analog-to-digital converter survey and analysis. *IEEE Journal on Selected Areas in Communications*, 17(4):539–550, 1999.
- [222] Nianqiang Li, Byungchil Kim, VN Chizhevsky, A Locquet, M Bloch, DS Citrin, and Wei Pan. Two approaches for ultrafast random bit generation based on the chaotic dynamics of a semiconductor laser. *Optics Express*, 22(6):6634–6646, 2014.
- [223] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
- [224] Robert Konig, Renato Renner, and Christian Schaffner. The operational meaning of min-and max-entropy. *IEEE Transactions on Information Theory*, 55(9):4337–4347, 2009.
- [225] John Kelsey, Kerry A McKay, and Meltem Sönmez Turan. Predictive models for min-entropy estimation. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 373–392. Springer, 2015.

- [226] Guido Boffetta, Massimo Cencini, Massimo Falcioni, and Angelo Vulpiani. Predictability: a way to characterize complexity. *Physics Reports*, 356(6):367–474, 2002.
- [227] Aviad Cohen and Itamar Procaccia. Computing the Kolmogorov entropy from time signals of dissipative and conservative dynamical systems. *Physical Review A*, 31(3):1872, 1985.
- [228] Massimo Falcioni, Luigi Palatella, Simone Pigolotti, and Angelo Vulpiani. Properties making a chaotic system a good pseudo random number generator. *Physical Review E*, 72(1):016220, 2005.
- [229] Peter Grassberger. Information flow and maximum entropy measures for 1-d maps. *Physica D*, 14(3):365–373, 1985.
- [230] Robert Shaw. Strange attractors, chaotic behavior, and information flow. *Zeitschrift für Naturforschung A*, 36(1):80–112, 1981.
- [231] Satoshi Sunada, Takahisa Harayama, Peter Davis, Ken Tsuzuki, Ken-ichi Arai, Kazuyuki Yoshimura, and Atsushi Uchida. Noise amplification by chaotic dynamics in a delayed feedback laser system and its application to nondeterministic random bit generation. *Chaos*, 22(4):047513, 2012.
- [232] Mario Stipčević and JE Bowers. Spatio-temporal optical random number generator. *Optics Express*, 23(9):11619–11631, 2015.
- [233] Jesper Mork, Bjarne Tromborg, and Jannik Mark. Chaos in semiconductor lasers with optical feedback: Theory and experiment. *IEEE Journal of Quantum Electronics*, 28(1):93–108, 1992.
- [234] R Badii, G Broggi, B Derighetti, Ms Ravani, S Ciliberto, A Politi, and MA Rubio. Dimension increase in filtered chaotic signals. *Physical Review Letters*, 60(11):979, 1988.
- [235] F Mitschke, M Möller, and W Lange. Measuring filtered chaotic signals. *Physical Review A*, 37(11):4518, 1988.
- [236] Holger Kantz, Wolfram Just, Nilüfer Baba, Katrin Gelfert, and Anja Riegert. Fast chaos versus white noise: entropy analysis and a Fokker–Planck model for the slow dynamics. *Physica D*, 187(1):200–213, 2004.
- [237] Kazutaka Kanno, Atsushi Uchida, and Masatoshi Bunsen. Complexity and bandwidth enhancement in unidirectionally coupled semiconductor lasers with time-delayed optical feedback. *Physical Review E*, 93(3):032206, 2016.
- [238] Joseph W Goodman. *Statistical optics*. John Wiley & Sons, 1985.
- [239] NIST random bit generation workshop 2016. <https://www.nist.gov/news-events/events/2016/05/random-bit-generation-workshop-2016>, 2016.

- [240] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. Summary of changes NIST SP 800 90B. <https://csrc.nist.gov/CSRC/media/Publications/sp/800-90b/final/documents/sp800-90B-changes-2nd-draft-to-final-summary.pdf>, 2018.
- [241] Shuangyi Zhu, Yuan Ma, Tianyu Chen, Jingqiang Lin, and Jiwu Jing. Analysis and improvement of entropy estimators in nist sp 800-90b for non-iid entropy sources. *IACR Transactions on Symmetric Cryptology*, 2017(3):151–168, 2017.
- [242] Jaideep Pathak, Zhixin Lu, Brian R Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos*, 27(12):121102, 2017.
- [243] Anbang Wang, Pu Li, Jianguo Zhang, Jianzhong Zhang, Lei Li, and Yuncai Wang. 4.5 Gbps high-speed real-time physical random bit generator. *Optics Express*, 21(17):20452–20462, 2013.
- [244] Carlos Abellán, Waldimar Amaya, Daniel Mitrani, Valerio Pruneri, and Morgan W Mitchell. Generation of fresh and pure random numbers for loophole-free Bell tests. *Physical Review Letters*, 115(25):250403, 2015.
- [245] Lynden K Shalm, Evan Meyer-Scott, Bradley G Christensen, Peter Bierhorst, Michael A Wayne, Martin J Stevens, Thomas Gerrits, Scott Glancy, Deny R Hamel, Michael S Allman, et al. Strong loophole-free test of local realism. *Physical Review Letters*, 115(25):250402, 2015.
- [246] Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenbergh, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, et al. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575):682–686, 2015.
- [247] Marissa Giustina, Marijn AM Versteegh, Sören Wengerowsky, Johannes Handsteiner, Armin Hochrainer, Kevin Phelan, Fabian Steinlechner, Johannes Kofler, Jan-Åke Larsson, Carlos Abellán, et al. Significant-loophole-free test of Bells theorem with entangled photons. *Physical Review Letters*, 115(25):250401, 2015.

## Curriculum Vitae

### Peer-Reviewed Journal Articles

Ryan J Suess, Joseph D Hart, Edward Leong, Martin Mittendorff, Thomas E Murphy. Black phosphorus frequency mixer for infrared optoelectronic signal processing. *APL Photonics*. Accepted. 2018.

Abu Bakar Siddique, Louis Pecora, Joseph D Hart, Francesco Sorrentino. Symmetry- and input-cluster synchronization in networks. *Phys. Rev. E* 79(2): 026208, 2018.

Joseph D Hart, Don C Schmadel, Thomas E Murphy, Rajarshi Roy. Experiments with arbitrary networks in time-multiplexed delay systems. *Chaos* 27(12): 121103, 2017.

Joseph D Hart, Yuta Terashima, Atsushi Uchida, Gerald B Baumgartner, Thomas E Murphy, and Rajarshi Roy. Recommendations and illustrations for the evaluation of photonic random number generators. *APL Photonics* 2(9): 090901, 2017.

Joseph D Hart, Kanika Bansal, Thomas E Murphy, and Rajarshi Roy. Experimental observation of chimera and cluster states in a minimal globally coupled network. *Chaos* 26(9) (2016): 094801.

Joseph D Hart, Jan Philipp Pade, Tiago Pereira, Thomas E Murphy, and Rajarshi Roy. Adding connections can hinder network synchronization of time-delayed oscillators. *Phys. Rev. E* 92(2): 022804, 2015.

### Submitted Journal Articles

Joseph D Hart, Laurent Larger, Thomas E Murphy, Rajarshi Roy. Delayed Dynamical Systems: Networks, Chimeras and Reservoir Computing. Submitted. arXiv preprint arXiv:1808.04596 (2018).

Joseph D Hart, Yuanzhao Zhang, Rajarshi Roy, Adilson E Motter. Topological control of synchronization patterns: Trading symmetry for stability. 2018.

### Journal Articles in Preparation

Yuanzhao Zhang, Zachary G Nicolaou, Joseph D Hart, Rajarshi Roy, Adilson E Motter. Globally attracting, spontaneously switching chimeras: theory and experiments.

### Book Chapters

Rajarshi Roy and Joseph D. Hart. A leading light: J.C. Bose and table-top experimental science. *Light and Its Many Wonders*. Editors: Ajoy Ghatak, Anirban Pathak, and V.P. Sharma. Publisher: Viva Books (2016).