Doctoral theses at NTNU, 2021:373

Johannes Høydahl Jensen

Reservoir computing *in-materio*: Emergence and control in unstructured and structured materials

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Johannes Høydahl Jensen

Reservoir computing *in-materio*: Emergence and control in unstructured and structured materials

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-6942-4 (printed ver.)
ISBN 978-82-326-5163-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2021:373

Printed by NTNU Grafisk senter

Abstract

This thesis is an exploration of novel material substrates for computation. Guided by the principles of material computation, we investigate what computations a material supports *naturally*. The goal is to perform computation close to physics, exploiting physical properties directly for more efficient computing devices. At the same time, we seek to understand the emergent computational mechanisms that are present within a material. We stress the importance of suitable computational models for understanding and exploiting novel physical systems. The model used in this work, reservoir computing, provides a powerful lens in which the computational properties of a material can be analyzed.

Three material substrates with distinct qualities are explored in this work. First, we investigate the computational properties of carbon nanotube nanocomposites. We find that these unstructured materials can exhibit complex dynamics in response to electrical stimuli, but that the limited control offered makes exploitation difficult. Next, we consider a simple chaotic circuit as a substrate for reservoir computing. We demonstrate how such a simple physical system can be exploited for computation. Its simplicity, controllability and structure enables detailed analysis, yielding insight into what physical properties are important for computation. Finally, we dive into the world of artificial spin ice (ASI): a family of structured magnetic metamaterials composed of coupled nanomagnets. ASI systems display a rich variety of complex emergent behavior, while at the same time offering considerable control. We explore the complex dynamics available in "square" ASI, and demonstrate how its computational properties can be controlled and exploited. A new ASI simulator was developed to support further investigations, which can capture realistic dynamics of large-scale ASI systems within practical time frames. We introduce reservoir computing in artificial spin ice, and quantify the reservoir quality of "pinwheel" ASI. A key result is that the emergent large-scale patterns of pinwheel ASI can be exploited at different scales of observation, ranging from individual magnets to the collective state of the resulting metamaterial. These findings, combined with highly developed fabrication techniques, make ASI very promising material substrates for future computing devices.

Our results highlight important properties of material computing systems, the degree of control offered by different materials, and what insight can be gained from reservoir computing *in-materio*.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (PhD) in Computer Science at the Norwegian University of Science and Technology (NTNU). The work was conducted at and funded by the Department of Computer Science (IDI), with additional funding from the Norwegian Research Council under the SOCRATES project, grant number 270961. Professor Gunnar Tufte was the main supervisor for the work, and Professor Helge Langseth and Associate Professor Stefano Nichele were co-supervisors.

The thesis takes the form of a paper collection, where the papers have already been published or submitted to conferences or journals.

Acknowledgments

First and foremost, I am deeply grateful to my supervisor Professor Gunnar Tufte for his unwavering support, invaluable advice and enthusiasm. I have enjoyed our countless discussions, from technical to philosophical, and truly appreciate the considerable academic freedom he has given to pursue my own ideas.

Next, I would like to thank my colleagues at the Department of Computer Science, for many excellent collaborations and for providing a friendly and inspiring working environment. Odd Rune Lykkebø, Arthur Penty, Asbjørn Djupdal, Magnus Själander, Stefano Nichele and Dragana Laketić have all been hugely important for my social well-being and growth as researcher. Thanks also to the magnetic duo at the Department of Electronic systems, Erik Folven and Anders Strømberg, for introducing me to the wonderful world of nanomagnetic systems. Several people kindly provided invaluable feedback on the final drafts of this thesis, for which I am grateful: Magnus Själander, Stefano Nichele, Eirik Jensen and Anders Strømberg.

Finally, I wish to thank the hugely important people behind the scenes, that is my family. I am grateful to my parents, Kari Høydahl and Eirik Jensen, for their endless love, support and encouragement. Thanks also to my brother, Bendik Høydahl Jensen, for all the support in our lifelong friendship. Last but not least, my most sincere and heartfelt thanks to my loving wife, Mari Lie Arntsen. This thesis would not have been possible without her tremendous support, patience and understanding.

Structure of the thesis

This thesis is a paper collection and consists of two parts.

Part I (Overview) introduces the research area, provides an overview of the work, covers relevant background, and concludes with a discussion of the research.

In part II (Publications), the papers are included in full. For readability and consistency, the papers have been reformatted for inclusion in the thesis, and therefore deviate visually from the published versions.

The five papers are listed below:

A. **Dynamics in Carbon Nanotubes for In-Materio Computation**
Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketic, Odd Rune Strømmen Lykkebø, and Gunnar Tufte (2016)
International Journal On Advances in Systems and Measurements

B. **Reservoir Computing with a Chaotic Circuit**
Johannes H. Jensen and Gunnar Tufte (2017)
ECAL 2017: The Fourteenth European Conference on Artificial Life

C. **Computation in artificial spin ice**
Johannes H. Jensen, Erik Folven, and Gunnar Tufte (2018)
ALIFE 2018: The 2018 Conference on Artificial Life

D. **flatspin: A Large-Scale Artificial Spin Ice Simulator**
Johannes H. Jensen, Anders Strømberg, Odd Rune Lykkebø, Arthur Penty, Magnus Själander, Erik Folven, and Gunnar Tufte (2020)
In review, preprint in arXiv:2002.11401 [cond-mat, physics:physics]

E. **Reservoir Computing in Artificial Spin Ice**
Johannes H. Jensen and Gunnar Tufte (2020)
ALIFE 2020: The 2020 Conference on Artificial Life

A summary of the papers and author contributions is given in Section 1.3.

Contents

Part I

Overview

Chapter 1

Introduction

No modern invention has had such a profound impact on mankind as the digital computer. The stored program computer and semiconductor technology have proven an unbeatable combination for nearly 70 years. Easily programmable, general purpose, high-precision, blazingly fast computing has become a ubiquitous resource. One would be hard pressed to find areas of society unaffected by the digital revolution.

Progress has been remarkable over the years, with a seemingly endless exponential growth in computing capabilities, as predicted by Moore (1965). Primarily, the growth has been facilitated by shrinking the size of the transistor, meaning more logic can be packed in the same area, while maintaining roughly the same power budget (Dennard et al., 1974). Reductions in size have been crucial to keep power consumption within manageable bounds, i.e., within physical limits where the heat generated can be effectively dissipated. Although a range of innovations have allowed progress to continue seemingly undisturbed for decades, it has become clear that we are near the end of transistor scaling (Mack, 2011). Scaling further will soon involve features approaching the atomic scale, which is simply not feasible.

Power is *the* limiting factor of modern processors today. Even though we have managed to shrink the transistor beyond our wildest dreams, parasitic effects result in leakage currents and have become a constant headache for designers. A consequence is that modern processors can only be *partially* powered at any given time, since the heat generated from the entire chip cannot be dissipated quickly enough (Esmaeilzadeh et al., 2011). Reducing power is not only a technological requirement at the chip level. Power is a limiting factor at all levels, from simple devices to data centers. The operational life time of battery-powered devices is limited by energy usage. Data centers consume power in the megawatt (MW) range, enough to power a small city, and getting rid of all that heat means a large part of the budget is dedicated to cooling (Ebrahimi et al., 2014).

When it comes to power, why is the digital computer so inefficient? The answer, it turns out, is fundamentally rooted in digital logic. Digital computing uses a lot of resources to do very little computation: even the most basic operations involve an intricate interplay between a large number of components. In silicon, the transistor is used to implement a simple switch, on top of which more complex operations are built, layer by layer. By design, digital computers do only the bare minimum of the computation in the physical domain. By involving the physical domain as little as possible, physical challenges such as component imperfections and signal noise can be engineered around to have no effect on the resulting computation. Digital machines perform perfect abstract computation in an imperfect physical world. However, we pay a high price for perfect computation, in terms of manufacturing costs, complexity, and power consumption.

Computing could be much more efficient if we dare let go of the digital paradigm altogether, and explore more unconventional ways of performing computations. The fundamental nature of the issues mentioned above makes such an exploration pressing. It is within the diverse field of unconventional computation we find the primary research topic of this thesis: *material computation*.

Material computation

Material computation aims to exploit the intrinsic properties of materials *directly* for computation. Computing close to physics can be extremely efficient, since operations happen "for free" in the rich physical domain. Analog computers prevailed well into the 1980s exactly because they used electrical phenomena as computational building blocks, and therefore were orders of magnitude more efficient than the digital computers at the time. Although material computing shares a similar goal (computing close to physics), it is based on fundamentally different principles.

Classical computer engineering is concerned with the question: how can we implement this kind of computation in this physical substrate? The classical approach involves building abstraction layers to effectively hide the physical processes of the substrate. Material computation, on the other hand, asks instead: what kind of computation does this physical system do *naturally*? By focusing on what the material does naturally, we strive to do as much of the computation in the physical domain as possible. There is very little or no design involved, but rather an investigation into the available physical mechanisms and how they relate to computation. In doing so, we may also discover computational models based on entirely different principles.

A key principle of material computation is *emergent computation*, i.e., computational processes that emerge from the underlying physics of the material substrate. We may contrast emergent computation with designed computation, as exemplified by classical computer engineering. Identifying and quantifying such emergent properties is an important part of assessing the computing potential of a material. Through a bottom-up exploration of material properties, one seeks to find emergent computational mechanisms that can be readily exploited.

In the search for computational properties, we take inspiration from nature, which is full of systems performing some form of intrinsic computation. Neural systems are a good example, in which computation is a result of the collective activity of large, self-organized networks of relatively simple elements (neurons). It is the ultimate example of efficiency and scalability: our brain consists of billions of neurons and trillions of synapses, yet it consumes only 20W of power. This incredible efficiency can be attributed to the use of physical phenomena as computational primitives (Mead, 1990). Natural computing systems display properties such as vast parallelism, self-organization, local communication, simple elements, emergent behavior, robustness and adaptation. Materials that exhibit similar properties are therefore particularly interesting.

However, transferring natural computing principles to artificial substrates must be done with care, since the physical properties of the artificial substrates are likely quite different from those of nature. For example, the time scales of silicon dynamics are much shorter than those involved in biological systems. Subtle differences in physical properties may have significant consequences for the computational properties, and consequently affect what kind of computations can take place.

Reservoir computing

A crucial step is finding a suitable *computational model* for the material substrate. A distinctive feature of biological computing systems is a close coupling between the architecture and the physics of the computing substrate. After all, biological computation is the result of natural evolution, where computation has developed hand in hand with the physical biochemical substrate. To exploit the intrinsic computation within a material, there must be a good correspondence between the properties of the model and the material.

A suitable model not only enables a material to be exploited to solve our computational tasks, it also enables its properties to be analyzed and reasoned about. Models

with programmability features greatly extend the versatility of a material, i.e., allowing the material to be exploited for different tasks. In this work, we employ a reservoir computing model, which is particularly promising for material computation.

Reservoir computing (RC) is a computational model that enables dynamical systems to be exploited for computation. The dynamical system, referred to as the *reservoir*, provides a rich repertoire of behaviors that can be tapped into to perform useful computations. RC employs a readout layer which is trained to produce some desired function based on reservoir state. While classical RC employs reservoirs in software, the model can be readily applied to exploit *physical* systems as reservoirs. The primary topic of this thesis, reservoir computing *in-materio*, is the study of novel material substrates as reservoirs.

Material computation is basic research, with a long-term goal of more efficient computers, as well as enhancing our understanding of the phenomena of computation itself. It is an inherently cross-disciplinary effort, involving fields such as physics, chemistry, biology, mathematics and computer science. In doing so, we must proceed with an open mind, dare let go of established notions of what computing *is*, and start exploring computing as it *could be*.

1.1 Research questions

The overarching theme of this research project is the exploration of novel physical substrates for computation. We are guided by the principles of material computation: computation close to physics, emergent rather than designed, bottom-up exploration of computational properties. A key principle is the use of computational models that provide a good fit with the material substrate. Only then can we hope to exploit the full computing potential of the substrate, where physics is exploited directly for computation. Towards that end, we depart from the classical paradigm of digital computing, and consider unconventional computational models.

The goal of such an endeavor is twofold.

First, there is a *technological* goal: investigating novel computing machinery may lead to the development of *more efficient* computers. In this context, efficiency may include properties such as power consumption, area usage, design and fabrication costs, fault-tolerance, robustness, etc. The study of unconventional computation close to physics may lead to improvements to the classical computer, or entirely new computing devices altogether.

The second goal is to advance our *theoretical* understanding: by exploring computing as it *could be*, we hope to gain insight into the nature of computation itself. What physical properties and mechanisms are essential for computation? How does the physics of the substrate affect its computational properties? For a given physical system, what are suitable computational models? These questions are relevant not only for insight into computation within man-made machines, but also for understanding computational processes occurring in nature.

Of course, these are lofty goals whose scope extends well beyond what can possibly be covered in a single research project. Nevertheless, they illustrate the big picture in which this research project is situated.

With that backdrop in mind, we present the three central research questions which are investigated in this thesis.

Research question 1

What properties of a material are important for computation, and how do we quantify these properties?

The aim of this research question is to find methods to assess the *potential* computing capabilities of a material. We begin by identifying relevant material properties, and how they relate to computation. In this context, the focus is on *intrinsic* computation, i.e., the basic mechanisms needed for computation to take place. An understanding of these intrinsic properties sheds light on the kind of computation a material does *naturally*, and highlights strengths and weaknesses. Some properties are structural, given by the composition of the material, e.g., the number of computational elements in a material could be related to the number of constituent parts. Other properties must be measured, and methods to quantify them are needed, e.g., the dynamical characteristics of a system has direct consequences for computation. It should be stressed that intrinsic computation is task-independent, and hence provide somewhat of a birds-eye view of a material's computational capabilities.

Research question 2

How can the computational properties of a material be controlled to tune functionality?

The *computational* properties of a material emerge from its underlying *physical* properties. Hence, the computational properties may be affected by changes in

the physical properties. If we can find ways to *reliably control* the computational properties, the functionality of the material may be tuned, resulting in a kind of programmability. Some properties may only be changed structurally by modifying the material itself, e.g., the coupling between the particles of a material may depend on the distance between them. Other properties may be changed through the environment, leaving the material unchanged, e.g., the properties of a driven dynamical system is typically affected by the characteristics of the external driving signal. The ability to tune computational properties from the environment offers a kind of material field-programmability. Metamaterials, i.e., physical systems whose constituent parts give rise to emergent material-like properties at some higher level, are particularly interesting in this regard, as they offer direct control over the constituent parts at the lower level.

Research question 3

How can reservoir computing help elucidate and exploit the intrinsic computational properties of a material?

Reservoir computing (RC) is a promising computational model for material computation, due to its generic requirements and basis in dynamical systems theory. RC *in-materio* allows the intrinsic properties of a material to be exploited for *useful* tasks. In addition, RC provides a framework in which useful computational properties can be measured and analyzed. A key question is which material properties are most important from a RC perspective. How flexible is the RC model at accommodating different materials? What does RC theory have to say about the computational capabilities and limitations of a material? In addition to theory, established reservoir quality metrics can be used to quantify computational properties, at a spectrum from intrinsic to task-specific. In this way, RQ1 and RQ2 may be illuminated from a RC perspective.

1.2 Research overview

This thesis is a paper collection, where the research has culminated in five papers. The papers have been published in relevant conferences and journals, and are included in part II of the thesis. In this section, an overview of the work is provided, along with a summary of the papers, organized according to the three research questions presented earlier.

The five papers, labeled A-E, are as follows:

A. Dynamics in Carbon Nanotubes for In-Materio Computation

B. Reservoir Computing with a Chaotic Circuit

C. Computation in artificial spin ice

D. flatspin: A Large-Scale Artificial Spin Ice Simulator

E. Reservoir Computing in Artificial Spin Ice

Materials

To address the research questions, the work has been focused on three different materials with a distinct set of properties. Using a reservoir computing model, we investigate the computational capabilities of these materials, and highlight important strengths and limitations.

The three material substrates investigated in this work are:

1. A *complex*, *unstructured* carbon nanotube material (CNT).
2. A *simple*, *structured* "material" consisting of a driven chaotic circuit (CC).
3. A family of *complex*, *structured* magnetic *metamaterials* called artificial spin ice (ASI).

These materials are particularly interesting for our study because they exhibit certain key properties for material computation: nonlinearity, rich dynamics, parallelism, local interactions and self-organization. The motivation for these properties is discussed in Section 2.3.

In the above list, the materials have been characterized as *unstructured* or *structured*, *simple* or *complex*. A structured material has a well-defined and ordered internal structure, i.e., the internal organization of the material is known. Unstructured materials, on the other hand, have no well-defined internal structure, i.e., with a disordered and unknown arrangement of the constituent parts. Somewhat orthogonally, complex materials have many constituent parts, whereas a simple material consists of relatively few elements.

The three materials are described in more detail in chapter 2 (Background): Section 2.5 introduces carbon nanotube materials, Section 2.6 describes the chaotic circuit, and Section 2.7 details artificial spin ice (ASI). Of the three materials, ASI

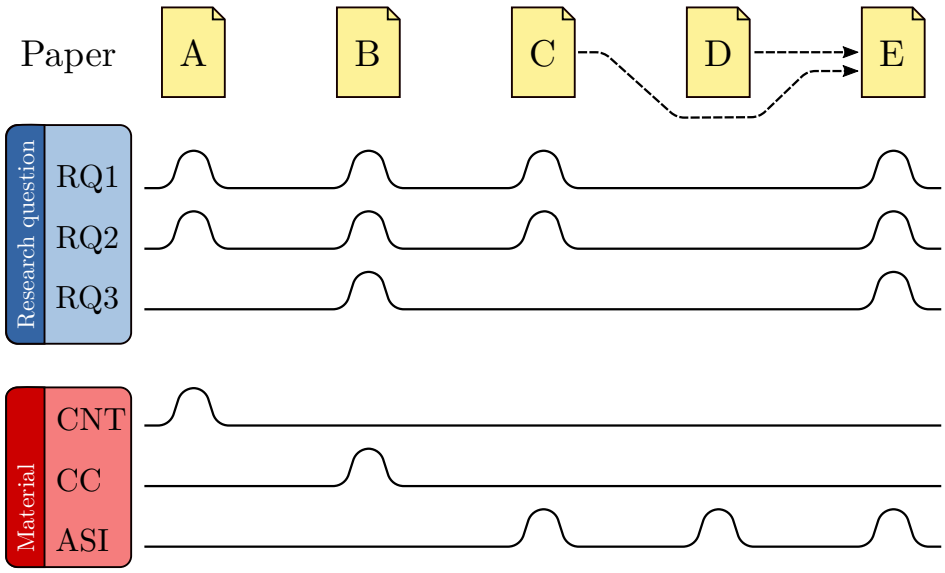


Figure 1.1: Overview of the papers in this thesis (top row) and how they relate to the research questions and material substrate (left column). The presence of a bump in the graphs indicate a relationship between a paper and research question / material. The dashed lines between papers indicate a dependency, i.e., paper E depends on the work in papers C and D.

is seen as particularly promising, and has been the primary focus of the work (three of the five papers are on ASI). Consequently, in the background chapter, ASI is covered in greater detail.

The papers at a glance

Fig. 1.1 illustrates how each paper (labeled A-E) relates to the three research questions (RQ1-3) and which material substrate is studied (CNT, CC or ASI). Relationships are depicted as six plots, where bumps indicate a relationship between paper and research question or material. As can be seen, RQ1 and RQ2 are addressed by all the papers (except paper D), whereas papers B and E involve reservoir computing (RQ3). Paper D is special in that it does not address any of the research questions directly, but presents a new ASI simulator which forms the foundation for further

research on ASI. Papers A and B are somewhat independent, while papers C-E are very much related in the common focus on ASI. In fact, paper E directly depends on the results from paper C and D, as indicated by the dashed arrows.

RC has been a key topic throughout all of the research, but not all papers employ RC directly. Papers A and C investigate the *intrinsic* computing properties of the material, and hence provide a foundation for choosing suitable computational models.

1.3 Research summary

In the subsections that follow, a summary of each paper is provided, along with relevant contextual information. Other relevant contributions are discussed at the end of this section.

The summary is given early to quickly give the reader an overview of the work. It makes use of terminology which is not introduced until later in chapter 2 (Background). As such, the reader may wish to skip ahead to chapter 2, before returning to this summary.

Paper A: Dynamics in Carbon Nanotubes for In-Materio Computation

By *Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketic, Odd Rune Strømmen Lykkebø, and Gunnar Tufte (2016)*

Published in *International Journal On Advances in Systems and Measurements*

In this paper, the intrinsic computational properties of a complex, unstructured bulk material are investigated. Specifically, nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and PolyButyl MethAcrylate (PBMA) are examined as material computing substrates. A basic requirement for computation is the availability of rich and complex dynamics, and the aim of this work was to gain a better understanding of the range of dynamics available in the material. The investigation was done at the experimental level of electrical signals and observations, i.e., using a model-free approach where the material is treated as a black box. First, a qualitative characterization is conducted by examining the material response to various input stimuli. Next, a quantitative measure of the dynamics based on computational

complexity was performed. We find that interesting, complex dynamics are available in the material. The dynamics depend on the material properties (such as the ratio of SWCNTs to PBMA), as well as properties of the input signals.

Role of authors:

Nichele and Jensen did the main experimental work in this paper.
Laketic made text contributions and the theoretical work.
Lykkebø and Tufte contributed with ideas, discussions and paper corrections.

Note:

This paper is an extended version of a conference paper (Nichele et al., 2015), which was invited to the *International Journal On Advances in Systems and Measurements*. This work was done as part of the NASCENCE project (see Section 1.4).

Paper B: Reservoir Computing with a Chaotic Circuit

By *Johannes H. Jensen and Gunnar Tufte (2017)*

Published in *ECAL 2017: The Fourteenth European Conference on Artificial Life*

This work was motivated by the question: what is the simplest physical reservoir? In this paper, we show how a very simple system can be exploited for computation through reservoir computing. The circuit in question is the driven Chua's circuit, which is one of the simplest chaotic circuits. Its rich repertoire of available dynamics suggests considerable intrinsic computing capacity. Dynamics range from stable to chaotic, readily tunable by the choice of circuit components and input stimuli. At the same time, the simplicity of the circuit enables detailed analysis to be made, offering insight into what physical properties are important for computation. In the paper we show, both through simulation and in-circuit experiments, how such a simple physical system can be readily exploited for reservoir computing. We obtain excellent performance on two non-temporal tasks, and demonstrate how RC allows the same physical system to be exploited for different computations. The fact that such a simple physical system can be utilized, suggests that a wide variety of physical systems should be viewed as potential reservoirs.

Role of authors:

Jensen did the main experimental work and writing of the paper.
Tufte contributed with text and corrections to the paper, as well as ideas and discussions.

Paper C: Computation in artificial spin ice

By *Johannes H. Jensen, Erik Folven, and Gunnar Tufte (2018)*

Published in *ALIFE 2018: The 2018 Conference on Artificial Life*

Artificial spin ice (ASI) is a class of magnetic metamaterials consisting of nanomagnets arranged on a 2D lattice. Local interactions between the magnets give rise to a range of complex emergent behavior. In this paper, we investigate intrinsic computing properties of ASI systems. We find several analogues between ASI and neural networks. Through micromagnetic simulations, we assess the range of dynamics available in "square" ASI, and quantify the number of reachable states. We find that the system exhibits a large number of reachable states and wide range of dynamics when perturbed by an external magnetic field. The strength and frequency of the external field has a direct influence on the dynamical regime and reachable states. We demonstrate how these properties may be exploited for temporal pattern recognition. Furthermore, we show that square ASI can be tuned to obtain effectively different modes of computation, ranging from classification to memory.

This paper laid the foundation for the remaining papers in this thesis, whose focus is also on ASI.

Role of authors:

Jensen did the main experimental work and writing of the paper.
Folven and Tufte contributed with text and corrections to the paper, as well as ideas and discussions.

Note:

This work was funded in part by the SOCRATES project (see Section 1.4).

Paper D: flatspin: A Large-Scale Artificial Spin Ice Simulator

By *Johannes H. Jensen, Anders Strømberg, Odd Rune Lykkebø, Arthur Penty, Magnus Själander, Erik Folven, and Gunnar Tufte (2020)*

In review, preprint in *arXiv:2002.11401 [cond-mat, physics:physics]*

Micromagnetic simulations (MM) provide a high-fidelity view of ASI systems. However, MM are computationally costly, which places hard limits on the size, time scales and number of ASI systems which can be investigated within reasonable time frames. For material computation, we are interested in emergent large-scale

behavior, at long time scales. To make matters worse, computational models such as reservoir computing require large number of simulations for training.

These issues prompted the development of a new fast simulator for ASI called *flatspin*, which is presented in Paper D. Flatspin employs a dipole model, where each magnet is approximated by a point dipole with a binary state (spin), and spins are coupled through magnetic dipole-dipole interactions. A novel switching criterion enables realistic dynamics to be captured with the dipole model. Through GPU acceleration, large ASI systems can be simulated within practical time frames. In the paper, the simulator is verified against a range of experimental results from the ASI literature. A highlight is that flatspin is able to reproduce the magnetization reversal of "pinwheel" ASI, which is the first time this has been demonstrated with a dipole model. The simulator enables quick exploration of new ASI geometries and properties, and has proven crucial for much of the ongoing ASI research within the SOCRATES project and beyond.

Role of authors:

Jensen and Lykkebø did the principal engineering work on the simulator, with important contributions from Strømberg and Penty.
The main authors of the paper were Jensen, Strømberg, Lykkebø and Penty.
Själander, Folven and Tufte contributed with text and corrections.
All authors contributed with ideas and discussions, both regarding the design of the simulator itself and its verification.

Note:

This work funded in part by the SOCRATES project (see Section 1.4).

Paper E: Reservoir Computing in Artificial Spin Ice

By *Johannes H. Jensen and Gunnar Tufte (2020)*

Published in *ALIFE 2020: The 2020 Conference on Artificial Life*

This paper presents the first study of reservoir computing in ASI. Specifically, we assess the reservoir quality of "pinwheel" ASI using established reservoir metrics (kernel quality and generalization capability). From a large number of flatspin simulations, we estimate the reservoir properties of pinwheel ASI under different field strengths and lattice spacings. As reservoir output, we consider the state of the ASI at different scales of observation, ranging from individual spins to the collective behavior of the resulting metamaterial. The results show excellent reservoir quality

for a range of parameters, as well as clear evidence of the fading memory property. Good reservoir quality is obtained at a range of different scales, yielding insight into the scalability and robustness of ASI reservoirs. The fact that good performance is found also at the larger scales suggests that ASI reservoirs are not only possible, but also practical.

Role of authors:

Jensen did the main experimental work and writing of the paper.
Tufte contributed with text and corrections to the paper, as well as ideas and discussions.

Note:

This work was funded in part by the SOCRATES project (see Section 1.4).

Other contributions

In addition to the papers above, other research contributions have been made which are worth mentioning.

The flatspin ASI simulator described in Paper D has been released as open source software under a GNU GPL license. flatspin is written in Python and uses OpenCL to accelerate calculations on the GPU. It has already received considerable interest from the ASI community, and has been instrumental for much of the work within the our research group. For more information, see the flatspin website[1], user manual[2] and source code repository[3].

Some of the work has been presented as posters and abstracts at various scientific venues. Worth mention is a poster (Strømberg et al., 2019) about our work on computation in ASI at the workshop *Frontiers in Artificial Spin Ice* in 2019. An extended abstract (Lykkebø et al., 2019) was submitted and presented at the *First International Workshop on Theoretical and Experimental Material Computing* (TEMC 2019), which was held in conjunction with the *Unconventional Computation and Natural Computation 2019* (UCNC 2019) conference.

[1] flatspin website: `https://flatspin.gitlab.io`
[2] flatspin user manual: `https://flatspin.readthedocs.io/en/latest/`
[3] flatspin source code repository: `https://gitlab.com/flatspin/flatspin`

1.4 Associated projects

The research is associated with three larger research projects.

The first project is NASCENCE (NAnoSCale Engineering for Novel Computation using Evolution), which was a EU-funded project (grant number 317662) exploring evolutionary computation in nanosystems (Broersma et al., 2012). A key principle of the project was a model-free approach, where the nanomaterial was treated as a black box, under direct control of an evolutionary algorithm. The project ran from 2012 until 2015. Although NASCENCE was being wrapped up in 2015 as this PhD project began, many of its ideas, principles and artifacts have been a hugely important foundation.

A few years later, in 2017, the SOCRATES[4] project (Self-Organising Computational substRATES) was started, funded by the Norwegian Research Council (NFR), grant number 270961. The goal of the project is the development of new computing paradigms to exploit self-organizing substrates. Biological neural networks and ensembles of nanomagnets are the primary substrates considered, where one seeks to replicate neuronal properties in nanomagnetic systems. In many ways, SOCRATES continues the work from NASCENCE, although moving away from the model-free evolutionary approach in favor of established physical and computational models. SOCRATES is a five-year project which will complete in 2022. The research in this PhD has been funded, in part, by the SOCRATES project.

Finally, in 2020, the SpinENGINE[5] project was funded under the FET Open EU programme, grant number 861618. The focus of SpinENGINE is on emergent computation in nanomagnetic ensembles. It is a continuation of the work set out in SOCRATES, but with a sole focus on nanomagnetic systems. SpinENGINE is a four year project which will last until 2024.

[4] SOCRATES website: `https://www.ntnu.edu/socrates`
[5] SpinENGINE website: `https://spinengine.eu/`

Chapter 2

Background

2.1 What is computation?

Before we go any further, we ought to define more precisely what is meant by computation. What, exactly, is a computer? Intuitively, we understand that a laptop running a spreadsheet program is performing computations, or that a calculator computes the sum 2+2. However, when we consider unconventional computing devices, such as material substrates, the issue becomes less clear. When is a material performing a computation, and when is it not?

Abstract theories of computation, such as Turing Machines (Turing, 1937), provide a mathematical formalism for the study of computational objects and their properties. They are pure mathematical theories, and describe computation entirely in the abstract domain.

However, in this thesis we are approaching computation from the physical point of view, i.e., *physical computation*. The bottom-up approach of material computation starts with the physical system, and asks what kind of computation it naturally supports. Hence, abstract theories of computation provides only half the picture. What is missing is the physical entity that is performing the computation: the computer.

A computer is a physical system that implements an abstract computation. For example, the digital computer is a physical device that implements an abstract Turing Machine. As such, a key question is how a physical system can be said to compute, when computation is defined entirely in the abstract domain. This is known as the problem of computational implementation, and there are many different theories. For an in-depth review, see Piccinini and Maley (2021).

At one end of the scale, pancomputationalism argues that everything is computation, i.e., that the universe can be considered a giant computer (Toffoli, 1982;

Wolfram, 2002). Although an interesting theory, many would argue that pancomputational reasoning is simply a rebranding of physics as computation, and gets us nowhere (Horsman et al., 2014). Clearly computers can be used to model most physical systems, but that does not mean, *ipso facto*, that all physical systems are computers (Hopfield, 1994). Based on these arguments, pancomputationalism must be rejected.

Other theories of computation impose much more stringent requirements on when a physical system can be said to compute. For instance, mechanistic theories require that some computational function can be tied to the constituent parts of the physical system (Piccinini, 2007). For material computation, such requirements seem much too strict. Importantly, they seem to exclude the possibility of emergent computation, i.e., computational mechanisms that cannot be tied to any constituent parts of the physical system.

For the purpose of our discussion, we define computation based on abstraction / representation (AR) theory (Horsman et al., 2014; Stepney and Kendon, 2019). AR theory is broad enough to include both classical and unconventional computing, but sufficiently narrow as to not include the entire universe.

The theory provides a useful definition of physical computation for our purpose:

> *Physical computing is the use of a physical system* (the computer) *to predict the outcome of an abstract evolution* (the computation).
> (Horsman et al., 2014)

Physical computing thus involves the abstract and physical domains, and is defined by the use of a physical system to implement a computational function.

As illustrated in Fig. 2.1, a computation is an abstract operation on some input, which produces some output as a result. To perform a computation on a physical system (the computer) the abstract input needs to be *encoded* into the physical domain. The state of the physical computer will then evolve according to its intrinsic dynamics. Finally, the physical state of the computer is *decoded* back to the abstract domain as the result of the computation.

A key element of AR theory is the *representation relation*, which relates physical variables to their abstract representations. In digital electronic computers, for example, (physical) voltages are used to represent (abstract) binary values. Put another way, the representation relation describes how abstract input is encoded into the physical system, and how physical state is decoded back into the abstract domain.

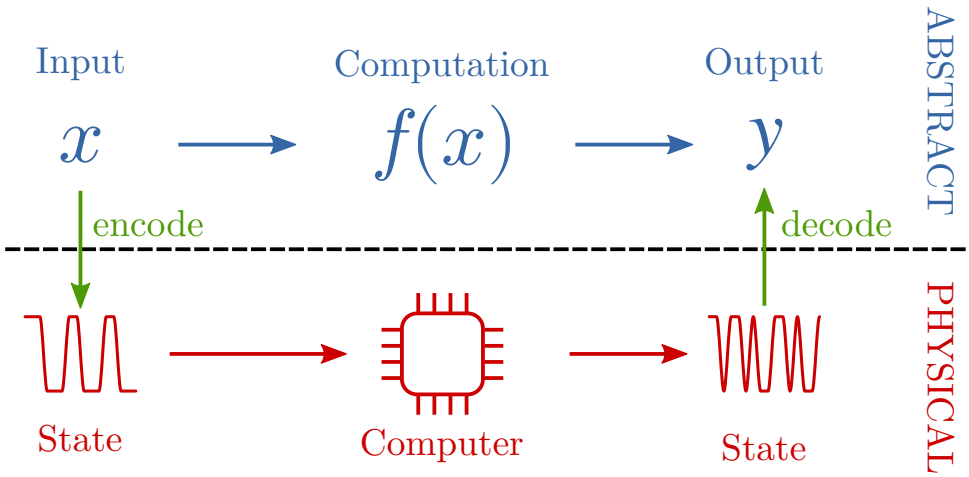


Figure 2.1: Physical computing is the use of a physical system (*the computer*) to predict the outcome of an abstract evolution (*the computation*). The figure is a simplified version of the "commuting diagrams" from AR theory.

We require that the (decoded) output from the physical computer matches the expected output of the abstract computation, within some margin of error. Thus a crucial step in the design of any physical computer, is to verify that the correct output is produced for a sufficiently large number of different inputs. Once we are confident that the physical computer works as intended, we can use the computer to perform the computation directly. At this point, we have established that the physical computer in Fig. 2.1 computes the abstract computation $f(x)$.

Note that we place no restrictions on the *form* of input and output: they may be discrete or continuous, static or time-dependent, etc. Furthermore, a particular form of abstract input/output does not necessarily dictate the form of physical input/output. For instance, a single discrete input value may be encoded as a continuous time-dependent signal in the physical domain.

Programmability has not been included in the above definition of computation. Granted, programmability is an essential feature of the modern computer: the ability to change functionality without re-wiring circuitry was key to the success of the stored-program computer. However, programmability is not a strict requirement for computation per se, e.g., fixed-function hardware accelerators can greatly speed up computations, at the cost of generality (non-programmability). In any case, a computer program can readily be included in the above definition as part of the input, or as an additional component.

Within our definition of computation, *any* physical system can be used as a computer as long as:

1. Input can be encoded into the physical system (encodability)
2. The physical system has the property of being influenced by the input (perturbability)
3. The influence can be observed as changes in the state of the physical system (observability)
4. Physical state can be decoded into the abstract domain as the output of the computation (decodability)
5. The resulting physical computation matches some desired abstract computation (correctness)

To summarize, computation involves an encoding and decoding process, as well as a physical dynamical process which implements an abstract computation. From this definition we immediately appreciate the close connection between computation and dynamical systems. Physical computation is the result of the time-evolution of a physical system, i.e., without dynamics there cannot be computation. We return to the topic of dynamics later in Section 2.2.

2.1.1 Natural computing

How does natural computing fit into the AR theory of computation? Do animals, bacteria or even plants "compute"?

According to AR theory, the answer depends on a key component: the representation relation. To identify computation, we must show that encoding and decoding takes place. In other words, "there is no computation without representation" (Fodor, 1981).

For example, animals make decisions based on sensory input such as smell, and an abstract computational problem could be "is this the smell of food?". In this case, encoding takes place in the nose, where aerosol particles are detected and transformed to electrical signals. The electrical signals are processed by neural circuits in the animal brain, which determine whether the scent stems from food, e.g., based on past experience. The result of the computation is an electrical signal, which can be decoded as "food" or "not food" by other parts of the brain, and eventually result in actions taken by the animal. Natural computation can thus be identified by the existence of a physical encoding and decoding process.

Computation can be distinguished from other physical processes by the arbitrariness of the physical signals involved (Horsman et al., 2017). For example, different organisms use different physical signals (electrical or chemical) to represent the same information ("food"). It is not the physical properties of the signals themselves that are being exploited, but their *representational* properties. This arbitrariness is key in determining whether an organism computes or not.

While AR theory may be used to understand computational processes occurring in nature, it has been challenged on more philosophical grounds. Such a discussion is outside the scope of this research project, but the interested reader is referred to Szangolies (2020).

2.1.2 Computational models

A computational *model* describes *how* an abstract computation is performed. In other words, it contains an abstract description of the internals of a machine which, when given an input, computes the desired output. It relates the computation to the computer in Fig. 2.1: it is an abstract description of a machine, which can be analyzed and reasoned about. A computational model should not be confused with a *physical model*: A physical model captures *physical* properties, while a computational model captures *computational* properties. The most widely studied computational model is the Turing Machine (Turing, 1937), but a number of of alternate models exist, e.g., neuromorphic computing (Mead, 1990), quantum computing (Benioff, 1980), cellular automata (von Neumann, 1951) and reservoir computing (Jaeger, 2001; Maass et al., 2002).

All modern digital computers are instantiations of the Turing Machine. It is a sequential, symbolic, mathematical construct, which is abstracted far away from the physical realm of silicon and transistors. Granted, the digital computer deviates from the Turing Machine in certain aspects for increased efficiency, e.g., the use of random-access memory instead of a sequential tape, shared memory for both instructions and data, etc. However, the core principle of the Turing Machine remain integral parts of the digital computer: the sequential execution of instructions, lossless information storage, and perfect mathematical computations.

With a computational model comes a set of properties which must be realized physically. Some properties can easily be transferred to the physical domain, e.g., information storage can be realized as electric charge, magnetic domains, the position of gears, etc. Other properties may require substantial engineering efforts, e.g., *lossless* information storage as realized in the digital computer requires extensive

support circuitry. The computational model also puts certain restrictions on computer architecture, e.g., sequential or parallel operation, global or local control, how information flows within the system, etc.

Clearly the choice of computational model has a huge impact on the performance of physical implementations. When there is a mismatch between the model and relevant physics, the speed and efficiency of computations will suffer. If the mismatch is too great, the model may not be physically realizable at all.

Nevertheless, a computational model is an invaluable tool for analyzing, comparing and reasoning about novel computing machines. Placing a physical system within a computational model highlights key properties of the system, its principles of operation, its strengths and weaknesses.

2.2 Computation in dynamical systems

Dynamical systems theory is the study of systems whose states change over time. Historically, the theory was developed for the study of physical systems (mechanical systems, electrical circuits, heat transfer, etc). It has since grown to encompass a wide range of more abstract models with loose connections to physics (cellular automata, neural networks, L-systems, etc). For an excellent introduction to dynamical systems theory, see Strogatz (2015).

Dynamical systems theory provides a solid foundation for the analysis of physical systems. The language of dynamical systems involves attractors, stability, chaos and bifurcations. If computation can be usefully described in the language of dynamical systems, the dynamical properties of a physical system may be directly related to computational mechanisms (Stepney, 2012). Dynamical systems theory thus enables a powerful link to be established between physical systems and computation.

Viewing computers as dynamical systems raises several important questions. What properties of a dynamical system are necessary for computation? How are basic computational mechanisms realized in a dynamical system? What is the relationship between dynamics and computation?

From our definition of computation in the previous section, almost any dynamical system can be used for computation, at least in principle. However, some dynamical systems will be more suited than others, as discussed next.

2.2.1 Styles of computation

The type of dynamical system will influence the *style* of computation.

Closed (autonomous) dynamical systems, for instance, evolve without any external influence. The only way to encode input into closed systems is through the initial condition. Hence, closed systems support a "ballistic" style of computation: once the input has been encoded, the computation is "fired off" after which the system evolves without any further input from the environment (Stepney, 2012). Once some equilibrium condition is reached, the particular attractor can be considered the output of the computation. If the system never reaches equilibrium, the computation has no result.

As an example of a closed dynamical system, consider a billiards table. The initial condition includes the placement of all the billiard balls, as well as the angle and force used to strike the cue ball. After the cue ball has been hit, the state of the system evolves on its own until all the balls have stopped moving, and an equilibrium has been reached. While the merits of "billiards computing" is questionable, it serves as a useful analogy for the ballistic style of computation.

The Turing Machine in its pure form can be considered a ballistic style of computation, in that input is provided as the initial state on the tape, after which the machine is left to run on its own. Another ballistic model is simulated annealing, which can realized in a physical system whose low energy states represents solutions. Computing based on attractors has the potential to exhibit robustness: a small perturbation might cause a small shift in system state, but over time the system will return to the same attractor.

Open (non-autonomous) dynamical systems, on the other hand, allow for external perturbations also during evolution. In this case, the dynamics will not only depend on the initial condition, but also on a time-dependent signal from the environment.

Returning to the billiards table example, imagine we modify the rules so that one is allowed to continuously tilt the table after the cue ball has been hit. In this modified system, the dynamics of the system will also depend on how and when the table is tilted, in addition to the original initial conditions. As such, the ability to tilt the table introduces a time-dependent signal from the environment.

For open dynamical systems, the computation may have a result even though the dynamics may never settle into an equilibrium. In other words, the output of the

computation can be a continuous function of time, or be based on the entire trajectory of the system. Such "trajectory-based" computation is more similar to the continuous processing of sensory information seen in biological systems, and real-time processing in digital systems. Reservoir computing (Section 2.4) can be considered a trajectory-based computational model.

Of course, computation is not limited to the two styles outlined above. Hybrid styles are also possible. If the state of a closed system can be observed continuously, for example, the entire trajectory may be considered as output. Open systems can also support a ballistic style of computation: input may be encoded as a time-dependent signal of finite length, after which the system is allowed to settle into an equilibrium.

2.2.2 Intrinsic computation and dynamics

Computation in dynamical systems can be viewed from two perspectives: *useful* computation and *intrinsic* computation.

Useful computation is related to the computations that help us solve concrete problems. Hence what is considered useful computation is somewhat subjective, in that it is related to human problems. Nevertheless, much can be learned by attempting to solve concrete problems with a dynamical system.

Intrinsic computation, on the other hand, concerns the fundamental *mechanisms* a dynamical system must possess in order to support computation (Crutchfield, 1994). Important mechanisms include the transmission, storage and modification of information. Intrinsic computation views computation as an emergent feature of a dynamical system. Hence intrinsic computation can guide us towards promising computational systems, but does not tell us how they can be used in practice.

The study of intrinsic computation has revealed some remarkable connections between dynamics and computation.

Wolfram (1984) categorized the dynamics of 1D cellular automata (CA) into four distinct classes, based on their long-term behavior. Class 1 systems relax to a homogeneous state, independent of the initial conditions (a point attractor). Class 2 systems exhibit periodic patterns of states (cyclic attractors). Class 3 systems are characterized by chaotic patterns (chaotic attractors). Class 4 systems display complex localized patterns, neither cyclic nor chaotic, with very long transients. He speculated that only class 4 systems are capable of universal computation, which was later confirmed (Cook, 2004).

Langton (1990) asks what dynamical regime supports the necessary ingredients for computation, i.e., information transmission, storage, and modification. Computation in dynamical systems involves a trade-off between information storage and transmission. From an information theoretic perspective, storage involves lowering entropy, while transmission involves raising entropy. Langton's work with CAs revealed that this trade-off is maximized near a phase transition, when the dynamical regime is between the ordered and chaotic regime. He hypothesized that adaptation in living systems involves adjusting their internal dynamics towards a critical regime near a phase transition, where computational capabilities are maximized.

That computational capabilities peak at the "edge of chaos" is a recurring observation and appears to be a very general feature of dynamical systems. Crutchfield and Young (1990) discovered similar features in continuous dynamical systems, using a measure of statistical complexity based on reconstructing the dynamics with a finite state machine.

Computation at the "edge of chaos" has also been a key topic in reservoir computing. The computational performance of random neural networks appear to peak when dynamics are at the edge of chaos (Bertschinger and Natschläger, 2004; Legenstein and Maass, 2007). Indeed, the peak in performance can be correlated with a maximization of information storage and transfer capabilities (Boedecker et al., 2012). Note that in the case of reservoir computing, computational performance can be directly related to *useful* computation. Reservoir computing is discussed later, in Section 2.4.

To summarize, dynamical systems theory provides a mathematical framework for the study of physical systems, and several connections have been established between dynamical systems and computing. Building on this foundation, *material computation* aims to exploit physical systems directly for computing, which is discussed next.

2.3 Material computation

The overarching goal of material computation is to perform computation close to physics, by exploiting the intrinsic properties of materials *directly*. Computing close to physics has the potential to be extremely efficient, since computations happen "for free" in the rich physical domain. The idea is to let physics do the job, by making use of what the material does *naturally*, rather than coercing the material to implement some abstract machine.

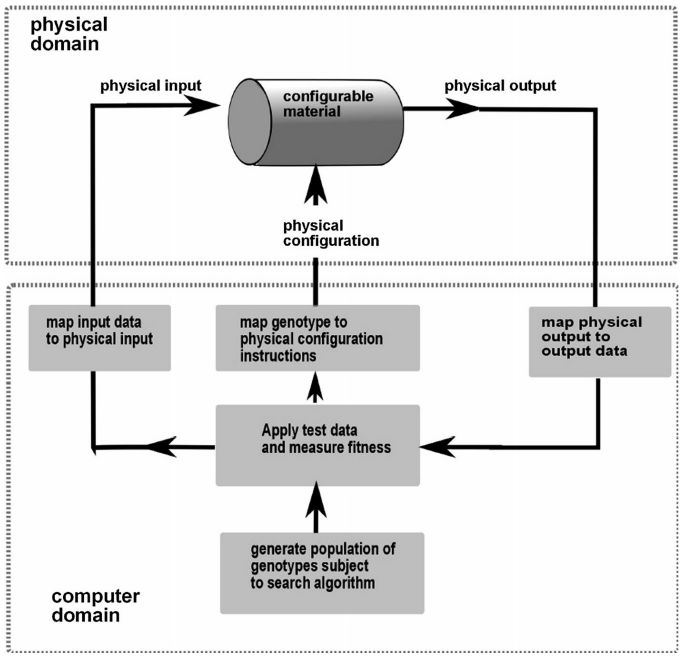


Figure 2.2: The hybrid architecture of EIM consists of a configurable material in the physical domain, and a conventional digital computer in the computer domain. See text for more details. Figure from Paper A (Nichele et al., 2016).

2.3.1 Evolution *in materio*

Material computation has its origins in evolution *in materio* (EIM), in which evolutionary algorithms (EAs) are used to "program" materials to solve computational problems (Miller and Downing, 2002). EIM treats the material substrate as a black box, and solutions to problems are evolved by searching for suitable input and configuration signals.

Fig. 2.2 illustrates the hybrid architecture of an EIM system, consisting of a configurable material (the physical domain), and a conventional digital computer (the computer domain). The digital computer controls the physical input and configuration signals that go into the material. In response to these signals, the material produces some physical output, which is measured by the digital computer. An EA runs on the digital computer, which searches for appropriate input and configuration signals, such that the output of the material matches some desired computation.

EIM deliberately avoids the use of computational models as well as physical models of the material. The motivation for such a model-free approach is to enable the EA

to discover and exploit physical properties which are either unknown or too complex to understand. Indeed, EAs have been known to discover highly unconventional solutions (Hornby et al., 2006), sometimes exploiting unknown physical properties (Thompson et al., 1999) or even physical mechanisms external to the experiment (Bird and Layzell, 2002).

A variety of computational problems have been solved successfully using EIM (see Miller et al. (2014) for a review). However, a critique of the black-box model-free approach is the limited insight it provides into the resulting computational process. It is difficult to discern what computations happen within the material, the measurement equipment, environment, or the EA itself (Lykkebø et al., 2015).

2.3.2 Early cybernetics

Although the field of material computation is still young, the underlying ideas and principles can be traced back to the early days of cybernetics. Ashby's work on adaptive mechanisms in the brain showed how adaptation is an emergent property of sparsely connected systems with random structure (Ashby, 1960).

Exploiting physical mechanisms for computation was a central theme in the early cybernetics movement. Pask experimented with growing conductive networks by varying the voltages of electrodes placed in a solution of ferrous sulfate. In this way he was able to grow an artificial "ear" in the solution: a self-organized conductive structure which could distinguish between different sounds (Pask, 1959).

A principal focus of early cybernetics was to understand the mechanisms that give rise to intelligence, in animals and machines. Central topics in the field were computation, communication, feedback, self-organization, adaptation, self-replication and reproduction as they occur naturally in animals and in artificial systems (Wiener, 1961). Although the modern field of cybernetics has evolved to become primarily concerned with control theory, many of the ideas from early cybernetics live on within fields such as Artificial Life, Complex Systems and Artificial Intelligence (Pickering, 2011).

2.3.3 Computation close to physics

The efficiency advantages of computing close to physics has been well-known since the days of analog computers (Sarpeshkar, 1998). Analog addition, for instance, is realized directly from Kirchoff's current law: the current flowing out of a wire junction is the sum of the currents flowing into that junction. As illustrated in Fig. 2.3,

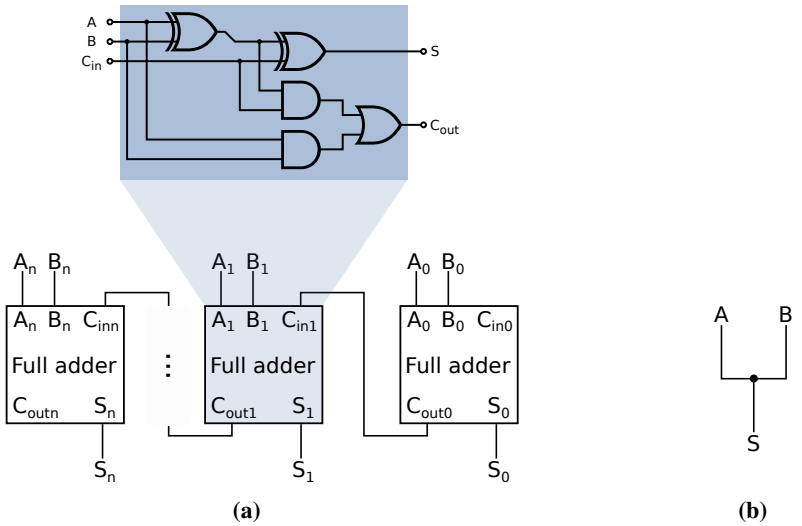


Figure 2.3: The use of physical phenomena as computational primitives can be very efficient. (a) Digital addition of two binary numbers A and B is realized as a cascade of full adders. Each adder circuit consists of a handful of logic gates. At the level below, each logic gate consists of a number of transistors. (b) Analog addition exploits Kirchoff's current law (KCL) to add two numbers A and B. From KCL, the output current S is the sum of the input currents A and B.

analog addition can be orders of magnitude more efficient than digital addition, both in terms of material resources and energy consumption during operation. Physics is full of interesting phenomena which may be exploitable for computation, ranging from simple operations such as addition to more complex computations.

Carver Mead, the father of neuromorphic computing, made similar efficiency observations when comparing biological and digital information processing systems. He writes: *"This advantage [of biological systems] can be attributed principally to the use of elementary physical phenomena as computational primitives"*. In other words, digital methods are inefficient exactly because they are abstracted far away from physics. He further argues that the deficiency is not due to any fundamental differences between the computational substrates (neural tissue vs silicon), but rather in how the substrates are being used in the system (Mead, 1990). In other words, suitable substrates (such as silicon) are readily available, but we are using them wrong.

How then, do we get it right? How do we effectively exploit the intrinsic computation available in a physical substrate? Material computation may have the answer.

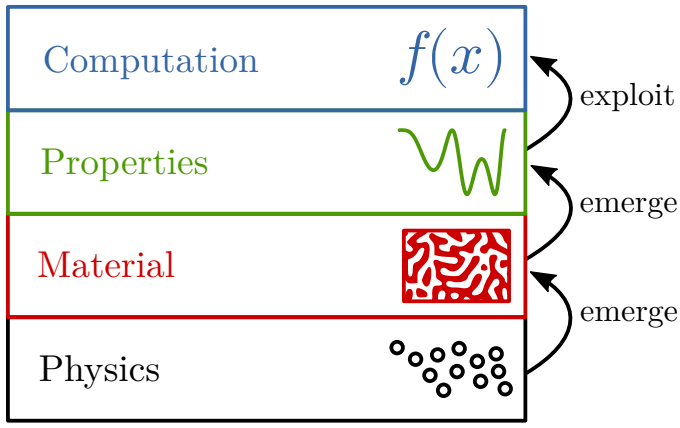


Figure 2.4: Abstraction levels in material computation. See text for details.

2.3.4 Emergent rather than designed

A key principle of material computation is *emergent computation*, i.e., the computational mechanisms that are naturally present in a physical system. Material computation calls for a bottom-up, exploratory approach, where the basic computational properties of a material are investigated. Once we have a sufficient understanding of these properties, then we can find good ways to exploit them for computation. Indeed, the bottom-up process is reminiscent of the natural processes which have ultimately resulted in biological information processing systems. Through exploration of physical building blocks, natural evolution has discovered extremely efficient ways of doing computation close to physics.

Both material computation and analog computation perform computations close to physics, but their approach is fundamentally different. Analog (and digital) computing is based on the top-down engineering of analog (or digital) circuits, which are carefully designed to implement some desired function. Material computation, on the other hand, is concerned with exploiting the intrinsic computation that emerges naturally from the physical properties of a substrate. The bottom-up emergent approach of material computation involves very little human design.

In some sense, classical computer engineering (be it analog or digital) is concerned with physical building blocks that have some desired functionality, which can be used to create larger structures. Material computation, in contrast, is the search for entire computational buildings within a physical medium.

Fig. 2.4 illustrates the different abstraction levels of material computation. At the

bottom level is the physics of the elementary components which constitute the material substrate (atoms, particles, nanomagnets). The hidden processes at the physics level gives rise to emergent phenomena at the material level above. It is at the material level we wish to perform computations, and consequently the level where input is applied and output observed. The phenomena at the material level give rise to computational properties at the level above. It is the computational properties we exploit when doing a computation, at the topmost level.

What kind of physical properties are relevant for computation? One answer is that it depends on the kind of problems we wish to solve: different problems exhibit different computational requirements. Nevertheless, it is possible to identify some generic properties which are desirable regardless of the application. Again, inspiration from nature guides us in the search for desirable properties.

2.3.5 Parallelism, local interactions and self-organization

Parallel computing is much more scalable than sequential computation. Intuitively, adding more computational elements is straightforward, whereas increasing the speed of every element is far from trivial. Luckily, most physical systems are inherently parallel, hence parallel computations are natural properties of most materials.

Biological computing systems are vastly parallel, consisting of large numbers of computing elements. While the computational capability of each element is fairly modest, it is from the collective operation of the ensemble as a whole that complex functionality emerges. The prototypical example is the complex function of the brain which emerges from the collective behavior of simple neurons. Hence a key question is how interactions between simple elements gives rise to some desired function.

Physical information transfer costs energy, a cost which increases with distance. Hence, we want to avoid sending data all over the place. Architectures where the communication between nodes is primarily local can have significant efficiency gains. Connectivity in the brain is mostly local, which has been attributed to the scaling of "wiring resources" (Mead, 1990). A range of physical mechanisms support *local interactions*, where the strength of physical signals decays naturally as a function of distance.

Maintaining global control of many parts is costly, complex and introduces a bottleneck in the controller. The centralized architecture of digital computers is the source of a range of performance problems in modern computing. Biological systems, in

contrast, employ distributed architectures without any central control mechanism. Instead, the individual parts *self-organize* to maintain some desired structure and/or dynamic behavior. Materials with elements that self-organize in some way are thus particularly relevant.

Self-organization can also be associated with robustness, when computation is the result of the collective behavior of many simple elements. For instance, elements may collectively perform error correction through some form of majority voting scheme. In the self-organized system, imperfections or even component failures can potentially have little impact on the computation.

2.3.6 Nonlinearity and rich dynamics

Most, if not all, computational models rely on some form of nonlinearity. Fundamental operations such as conditionals and branching requires the presence of a threshold function (a nonlinear function) at some lower level. Nonlinearity can be directly related to the computational capabilities of a system. Multilayer neural networks have been proven to be universal approximators, i.e., any function f can be approximated arbitrarily well by a network, but only when a nonlinear activation function is present (Pinkus, 1999). Luckily, most physical systems are nonlinear (Strogatz, 2015), although this might not be apparent at the material level.

Importantly, nonlinearity is fundamental in any type of complex dynamics. As we have discussed in Section 2.2, complex dynamics are associated with intrinsic computation. For a material to support computation, it must exhibit complex dynamics and consequently nonlinearity.

Rich dynamics refers to the presence of a variety of dynamics in a material, ranging from simple to complex. A rich variety of dynamics suggests that the material is capable of a range of different computations. Although chaotic dynamics may not be directly computationally useful, the presence of chaos indicates that a dynamical regime at the edge of chaos may be obtained by tuning parameters. Rich dynamics and chaos are thus key properties of promising materials.

2.3.7 Physical constraints

In addition to the abovementioned desirable properties, physical constraints place limits on the usability of a substrate. As discussed in Section 2.1, a strict requirement for computation is the ability to perturb the system with input, and observe the response. The input signals must be realized physically, which places limits on

the strength, speed and number of signals. Similarly, the output must be recorded within the capabilities of available measurement technology. For example, computation at the atomic scale may look promising on paper, but wildly impractical if output cannot be observed at larger scales.

Given the desirable properties discussed above, where should we start looking for candidate materials? Biological systems clearly exhibit most, if not all, of these properties. However, Stepney (2008) argues that biological material substrates should be avoided, since they have already been highly specialized through evolution. Put another way, biological systems are less malleable, and difficult to adapt for our computational benefits. Instead, we should focus our attention on non-living systems which have not yet been adapted towards some purpose. Relevant fields are thus physics and chemistry, rather than biology.

2.3.8 Quantifying computation

Some material properties will be intrinsically tied to the system, e.g., the presence of local interactions can be deduced directly from physical structure. Other properties, such as those related to dynamics, must be measured through experiments.

The *intrinsic* computational properties of a material can be quantified using metrics which are independent of any particular computational model. For example, complexity metrics give an overall indication of computational capacity, by considering the difficulty of reproducing material behavior with a machine. Although such metrics are suggestive of computational capabilities, they say little about the *usefulness* of the computations.

To investigate the amount of *useful* computational capabilities available in a system, one approach is to apply it to solve a variety of problems. Although such an application-focused approach certainly has value, it is often difficult to generalize the results to different problem domains. An alternative, and arguably more informative approach, is to measure and characterize the useful computational capabilities of a system.

If the material can be placed within a suitable computational model, more powerful analysis and reasoning about the computational properties can be performed. Within some computational models, there are metrics developed to assess specific computational capabilities which are directly translatable to useful computation. Hence, if a substrate can be placed within the framework of existing computational models, much more can be deduced about its useful computational capabilities.

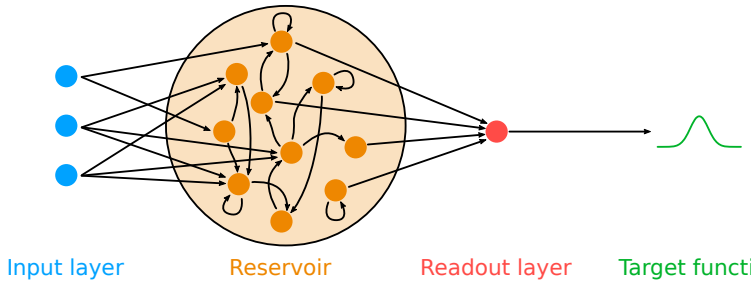


Figure 2.5: The reservoir computing architecture consists of the input layer, the reservoir, and the readout layer. Figure from Paper B (Jensen and Tufte, 2017).

As discussed earlier, it is crucial that the chosen computational model matches the natural properties of the material, lest the efficiency suffers (Section 2.1.2). Clearly different computational models place distinct requirements on the properties of a material. A model designed specifically for a given material has the potential to provide a perfect fit. However, the use of more generic models is beneficial at an early exploratory stage. Furthermore, generic computational models allow for comparison of the computational capabilities of different materials. One such generic computational model, which places rather loose requirements on the substrate, is reservoir computing, which is discussed next.

2.4 Reservoir computing

Reservoir computing (RC) is a computational model in which a dynamical system is exploited for computation. The dynamical system is referred to as the *reservoir*, which provides a rich repertoire of computations which can be tapped into to perform useful tasks.

Fig. 2.5 illustrates the typical RC architecture, which consists of three distinct parts: the input layer, the reservoir, and the readout (output) layer. A time-dependent input signal perturbs the reservoir, which in turn produces some complex dynamical response. Subsequently, the reservoir state is passed to a linear *readout* which is trained to produce some desired output as a weighted sum of reservoir states. The readout is memoryless, meaning it only has access to the *current* state of the reservoir at any given time. Crucially, the readout layer is the only trained part of the system. Both the input layer and the reservoir remains fixed throughout, and are typically randomly generated according to some hyperparameter.

RC was discovered twice, independently and simultaneously, under the names Echo State Networks (ESNs, Jaeger (2001)) and Liquid State Machines (LSMs, Maass et al. (2002)). ESNs were proposed as a way to exploit the computational power of recurrent neural networks (RNNs), which are difficult to train. Using randomly generated RNNs, Jaeger (2001) discovered that good performance could be obtained by training only a linear readout layer on the activations of the network. Training the readout amounts to linear regression, which is both easy and fast.

At the same time, LSMs were proposed as a computational model of the real-time processing in biological neural microcircuits (Maass et al., 2002). Although the focus was on neural modeling rather than purely computation, the ideas and overall architecture are identical to ESNs. ESNs and LSMs were later unified as reservoir computing.

RC has outperformed state of the art methods on a range of challenging tasks. The primary focus has been on temporal tasks, i.e., where the input and/or output are functions of time. Examples include time series prediction and classification, speech recognition, signal generation and attractor reconstruction. Although not as frequent, non-temporal tasks have also been treated successfully with RC methods. For an excellent introduction to RC methods, see Lukoševičius and Jaeger (2009). Gallicchio et al. (2020) reviews recent developments in the highly active field of RC.

2.4.1 Reservoirs are dynamical systems

It was early realized that reservoirs do not need to be neural networks at all. *Any* dynamical system can potentially be used as a reservoir, as long as it can be perturbed with input and (some of) its state observed. Reservoirs are open (nonautonomous) dynamical systems, whose states are continuously affected by external stimuli. What, then, is the computational function of a reservoir?

Reservoirs are high-dimensional dynamical systems, whose transient states represent nonlinear projections of the input. They can thus be viewed as nonlinear kernels with memory, projecting a low-dimensional temporal input into a high-dimensional space where salient features of the input are expressed. The high-dimensional reservoir space allows input to be separated using linear methods, as illustrated in Fig. 2.6. Reservoirs have memory in the sense that their state depends both on present and past input, i.e., a nonlinear memory of input history is maintained. This nonlinear memory is exploited by the readout layer to perform useful computations.

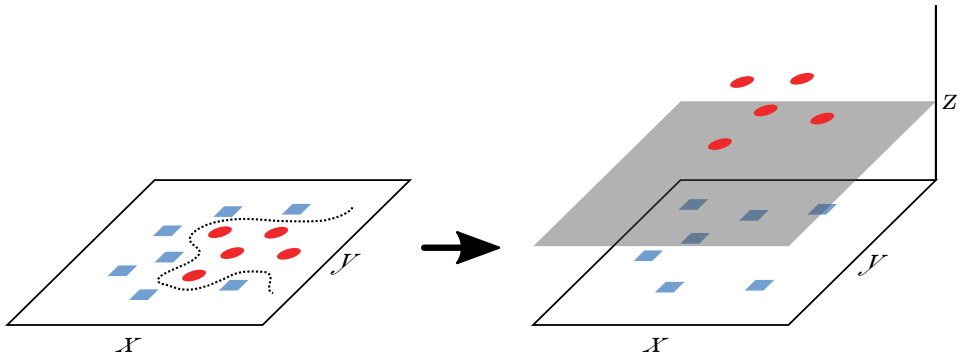


Figure 2.6: Reservoirs can be viewed as nonlinear kernels that map low-dimensional input into a high-dimensional space. Illustrated here is a set of points which represent two classes: square and circles. Left: In two dimensions, the two classes are not linearly separable, i.e., they cannot be separated by a line. Right: A nonlinear mapping into a third dimension allows the classes to be separated by a plane, i.e., the mapping has made the points linearly separable.

The "liquid" analogy of Maass et al. (2002) is fitting: dropping pebbles into a pond of water produces long lasting perturbations, visible as wave patterns traveling across the surface. Hence, the state of the reservoir (liquid) represents both current and past input (pebbles), transformed in a complex way which potentially highlights salient features of the input.

2.4.2 Reservoir properties

To be useful, a reservoir needs to possess a set of fairly general properties: high dimensionality, nonlinearity and fading memory. Each of these are discussed below, as well as some other related properties.

A reservoir needs to be *high-dimensional* to facilitate a rich repertoire of functions within its states. Theoretically, the inherent computational capacity of reservoirs is bounded by the number of output nodes, i.e., the number of state variables available to the readout layer (Dambre et al., 2012). Computational capacity is maximized when the output nodes are linearly independent, i.e., a maximum of diversity in the available functions. This make intuitive sense: the richer the repertoire of functions, the more freedom available to the readout layer to construct useful computations. However, low-dimensional reservoirs may still be employed, e.g., by time-multiplexing the output to produce a virtual high-dimensional state (Appeltant et al., 2011).

Nonlinearity is an important property of reservoirs. As discussed in Section 2.2, all non-trivial computations rely on some kind of nonlinearity, and nonlinearity is a requirement for rich dynamics. In RC, any memoryless linear computation may be implemented by the readout layer alone. Purely linear reservoirs thus have limited computing power, serving only as a memory of past input. Interestingly, there is an apparent universal trade-off between nonlinearity and memory (Inubushi and Yoshimura, 2017).

A key property of reservoirs is *fading memory* (Boyd and Chua, 1985), also known as the "echo state property". Informally, fading memory means that the reservoir forgets over time. In other words, reservoir state depends on a *finite window* of past states. The name ESN is based on exactly this notion of reservoir state as an "echo" of input history. Fading memory can be directly related the dynamical regime of the reservoir. Ordered dynamics have fading memory, but limited computational power as all inputs quickly converge to the same attractor. In contrast, chaotic dynamics never forget, since the effect of perturbations will be unbounded in time. Fading memory thus calls for a dynamical regime lying somewhere in between stability and chaos.

As discussed earlier (Section 2.2), there is ample evidence that reservoir performance peaks when the dynamical regime is at the edge of chaos (Bertschinger and Natschläger, 2004; Legenstein and Maass, 2005; Boedecker et al., 2012). It should be noted that an inherently chaotic system may still be used as a reservoir, as long as input is sufficiently strong to drive dynamics out of the chaotic regime (Ozturk and Principe, 2005).

Closely related to the dynamical regime are the *separation* and *approximation* properties (Maass et al., 2002, 2004). A reservoir exhibits the separation property if *different* inputs give rise to significantly different reservoir states, effectively separating the inputs. Likewise, the approximation property is fulfilled if *similar* inputs result in reservoir states that coincide. Both properties are required for useful computations, and both properties are related to the dynamical regime of the reservoir. Chaotic dynamics will only exhibit the separation property, since *all* inputs will be separated, regardless of their similarity. Stable dynamics, on the other hand, will only result in the approximation property, by mapping *all* inputs to a small subset of reservoir states. Thus a good reservoir will be in a dynamical regime somewhere between stability and chaos, i.e., where both the separation and approximation properties are fulfilled.

2.4.3 Physical reservoirs

The close relationship between RC and dynamical systems has spurred the use of physical systems as reservoirs. As discussed, RC is a very general computational model, with fairly permissive requirements of the reservoir. Hence there is a great deal of flexibility in the physical realization of reservoir properties. Given that the physical system is *nonlinear*, *high-dimensional* and exhibits *fading memory*, it can potentially be exploited for computations in a RC framework.

Due to the general and permissive nature of RC, it has become a popular model for exploring unconventional hardware. A diverse range of physical reservoirs have been proposed, including electronic (Schurmann et al., 2004), optoelectronic (Appeltant et al., 2011), photonic (Vandoorne et al., 2014), magnetic (Nomura et al., 2019) and mechanical (Coulombe et al., 2017) reservoirs. Some of the more unusual physical reservoirs include a bucket of water (Fernando and Sojakka, 2003), a soft robotic arm (Nakajima, 2017), the E.Coli bacteria (Jones et al., 2007) and the cat's primary visual cortex (Nikolić et al., 2007). For a recent review of physical RC, see Tanaka et al. (2019).

More recently, RC has attracted attention as a promising model for material computation (Dale et al., 2017). One of the earlier examples of a material reservoir is Atomic Switch Networks (Sillin et al., 2013): ensembles of silver nanowires that self-assemble into random memristive networks with inherent plasticity. Carbon nanotubes have a similar random network structure, but without the plasticity, and have been explored as reservoirs within an evolutionary framework (Dale et al., 2016b,a). A newly proposed class of reservoir materials are skyrmion fabrics: topological magnetic structures pinned in the plane of a ferromagnetic material (Pinna et al., 2020). It is clear that RC provides a very capable framework for exploring the inherent computational capabilities of such novel material substrates.

The versatility of physical reservoirs depends on the ability to control their behavior to obtain desirable properties. Clearly, different tasks have distinct computational requirements; some tasks may be dominated by memory, while others require more nonlinearity. If the properties of a physical reservoir can be tuned reliably, they may be exploited in a variety of tasks. Towards that end, Dale et al. (2019) have proposed a framework for the systematic characterization of material reservoirs based on established RC metrics. The potential of a reservoir can then be quantified from the number of different computational behaviors.

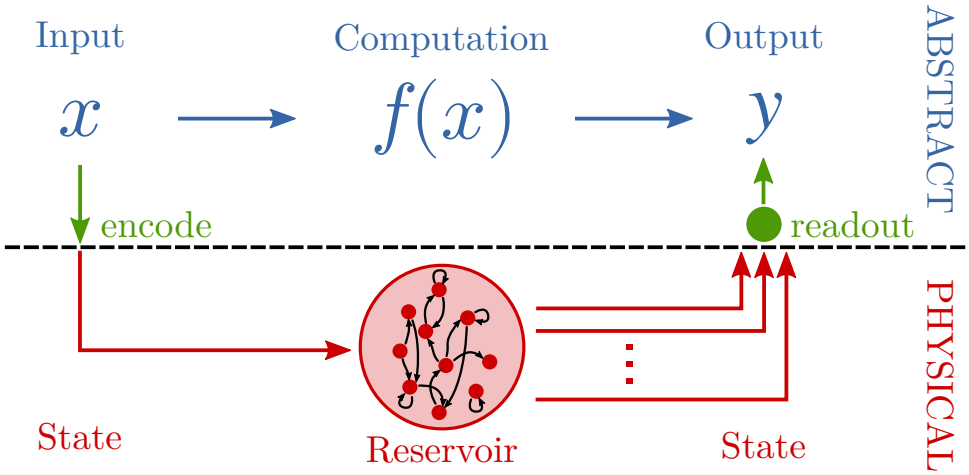


Figure 2.7: Physical reservoir computing exploits a physical dynamical system (the reservoir) for computation, by training a linear readout on the high-dimensional state of the reservoir.

2.4.4 The computational model of physical RC

As a computational model, how does physical RC fit into our definition of computation from Section 2.1? Comparing the RC architecture (Fig. 2.5) with Fig. 2.1, it becomes clear that RC is a fairly minimal computational model. All that is strictly required is the ability to encode an input as perturbations to the reservoir, and a way to decode the reservoir state.

Fig. 2.7 illustrates the RC model placed in our previous definition of computation. There are several options on what should be considered physical, i.e., is the input layer and/or readout part of the physical or abstract domain? For the purpose of our discussion, it is most natural to consider only the reservoir as part of the physical domain. This also reflects the typical approach of physical RC. The input layer is then part of the encoding stage, whereas the output layer takes part in the decoding.

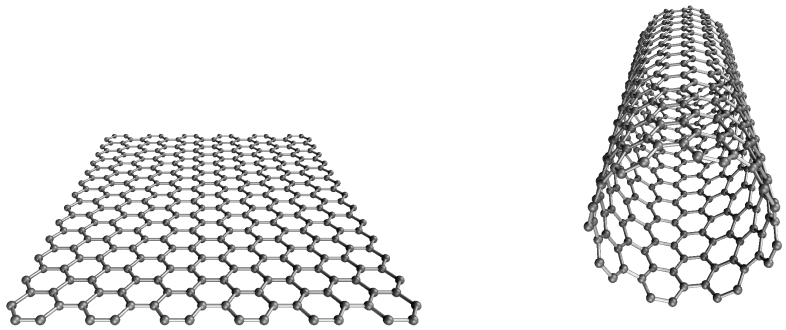
To be precise, the input and readout layers are the *abstract* part of the encoding and decoding stage. When the reservoir is a physical system, there will also be a *physical* part of the encoding and decoding. The physical encoding dictates what physical states to perturb, i.e., how the abstract input should be encoded as a physical signal. Likewise, the physical decoding specifies what physical state to record, and how the physical variables should be mapped to abstract values before being passed to the readout layer.

Within this definition, what is the *programming model* of RC? If we consider the reservoir to be a fixed resource which cannot be changed, then the programming model involves only the readout layer. In this case, programming is an exercise in supervised machine learning, subjecting the reservoir to a range of training examples, where the corresponding reservoir states are used as training data for the readout layer. It is an activity in the abstract domain, adapting the use of reservoir states for a particular task. This is a very advantageous model for material computation, since it allows the same material reservoir to be re-used for a range of different tasks. In fact, Jaeger (2001) used the very same ESN for all the benchmark tasks in his seminal paper. Another attractive possibility is the use of multiple readout layers to implement different computations on the same stream of input data (Maass et al., 2002).

Alternatively, programming may also involve modifications to the physical input encoder to perturb the reservoir in different ways, effectively tuning the computational properties of the reservoir. For example, many dynamical systems are sensitive to the strength of the input signal, allowing different dynamical regimes to be selected. Although it can be argued that this effectively changes the dynamical system itself, the underlying *structure* of the reservoir remains unaltered. Some physical systems may even allow additional modifications through other tuning signals, e.g., tuning of the nonlinearity.

Finally, programming may include making structural changes to the reservoir itself. Such changes may involve moving the internal components of the reservoir, modifying the coupling between the components, altering components, adding or removing elements, etc. In classical RC, this would be analogous to generating a new RNN with different hyperparameters. If such modifications can be done in a controlled manner, it could allow tuning the emergent computational properties of a physical reservoir towards some desired functionality.

Physical RC is a highly active research area. Novel physical systems are routinely proposed as promising reservoir candidates, ranging from highly engineered systems to more exotic substrates. Within this research project, we investigate three different material substrates (CNT, CC and ASI), which are described in the following sections. Note that, because a majority of the research has focused on ASI, this class of materials is discussed in greater detail.



(a) (b)

Figure 2.8: (a) A sheet of graphene consists of a single layer of carbon atoms arranged in a 2D honeycomb lattice. (b) A single-walled carbon nanotube (SWCNT) is obtained when a sheet of graphene is rolled up to form a cylinder. Figures derived from Eric Wieser, CC BY-SA 3.0, via Wikimedia Commons.

2.5 Single-walled carbon nanotubes

Carbon nanotubes (CNTs) are tubular carbon nanostructures, obtained when sheets of graphene are rolled up to form cylinders, as shown in Fig. 2.8b. Graphene consists of a single layer of carbon atoms arranged in a 2D honeycomb lattice (Fig. 2.8a). A single sheet of graphene results in the single-walled carbon nanotube (SWCNT), whereas multiple sheets of graphene creates the multi-walled carbon nanotube (MWCNT). While the diameter of a CNT is only a few nanometers, they can be several centimeters in length.

Carbon nanotubes have received enormous interest due to their unique properties (Volder et al., 2013). The electrical properties of CNTs are of particular relevance for novel computing devices. Depending on the chirality (a characteristic of how the sheet of graphene is rolled up), CNTs can be either metallic or semiconductive. Roughly one third of the possible CNTs are metallic, while the remaining two thirds are semiconductive.

There are two major directions for CNT applications in electronics: nanoelectronics and macroelectronics (Che et al., 2014). Nanoelectronics is concerned with the controlled alignment of nanotubes to form electrical circuits. Macroelectronics, on the other hand, employs thin films of random CNT networks as components in

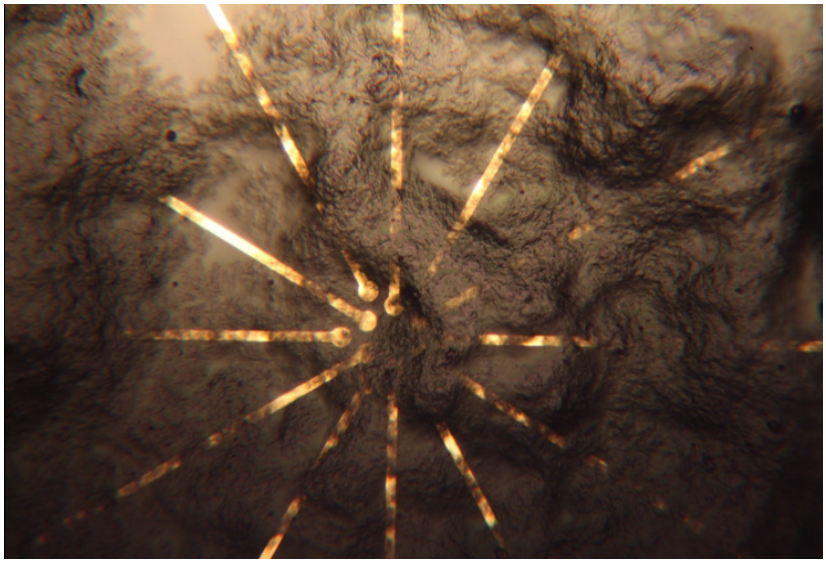


Figure 2.9: SEM image of gold electrode array with different coverage of nanotubes. Figure from Paper A (Nichele et al., 2016).

electronic devices. It is such thin-film CNT materials that are investigated in this thesis.

Specifically, the CNT materials investigated here are nanocomposites of SWCNTs and an insulating polymer (PolyButyl MethAcrylate, or PBMA). Mixing a solution of SWCNTs in PBMA results in random CNT networks, where the polymer creates insulation barriers within the material. A key parameter is the concentration of SWCNTs, i.e., the weight ratio of SWCNTs to PBMA. The resulting nanomaterial thus consists of random networks of metallic and semiconductive nanotubes.

The SWCNT nanomaterial may be interfaced electrically by the use of a microelectrode array (MEA), as illustrated in Fig. 2.9. Dispersing the material onto a MEA exposes parts of the material to electrical measurements and stimuli. Note that, with this method, there is no control over which parts of the CNT networks are being accessed. Furthermore, depending on the deposition of the material onto the MEA, some electrodes may be exposed to large amounts of material while others may receive little. This can be seen in Fig. 2.9 where some electrodes are covered by more material than others. As such, the MEA allows access to a random section of the material, both in terms of location and the amount of material.

2.5.1 Computation in carbon nanotubes

While carbon nanotubes have received considerable interest as building blocks in conventional digital electronics, their use in unconventional computing paradigms is much less explored.

From a computational point of view, the SWCNT material is, conceptually, a random network of linear and nonlinear elements. Such random networks may be exploited for computation, if the material can be configured to elicit a useful response to input stimuli.

One approach is to use an evolutionary algorithm (EA) to search for suitable input and configuration signals, i.e., evolution *in materio* (Section 2.3.1). CNT materials have been successfully exploited for computation in this manner to solve a variety of computational problems, e.g., traveling salesman problems (Clegg et al., 2014), graph coloring (Lykkebø and Tufte, 2014) and logic gates (Massey et al., 2015). While this demonstrates that useful computations can take place with a CNT material, the black-box EIM approach makes it difficult to pinpoint exactly what computational mechanisms are being exploited in the material itself (Lykkebø et al., 2015).

Another approach is to employ a reservoir computing (RC) model, which allows more insight into the computational properties of the CNT material. After all, classical RC exploits the inherent properties of random neural networks. Perhaps random CNT networks are exploitable in a similar manner? Dale et al. (2016a) show that the answer is yes: by using an EA to configure the CNT material, it can subsequently be used to solve RC tasks. In some cases, the CNT material is shown to outperform other hardware reservoirs (Dale et al., 2016b). RC metrics can be used to capture various computational properties of material reservoirs. Different physical reservoirs can thus be compared, and contrasted with classical software-based reservoirs. In this manner, CNT material reservoirs have been shown to be less versatile: they exhibit a limited set of computational capabilities compared to software models of similar size, as well as other physical reservoirs (Dale et al., 2019).

2.6 Chaotic circuit

Analog electronic circuits allows for the physical realization of ordinary differential equations (ODEs). By combining Kirchoff's circuit laws with the equations for various electrical components (resistors, capacitors, inductors, diodes, operation amplifiers), it is possible to create a circuit whose behavior matches some desired

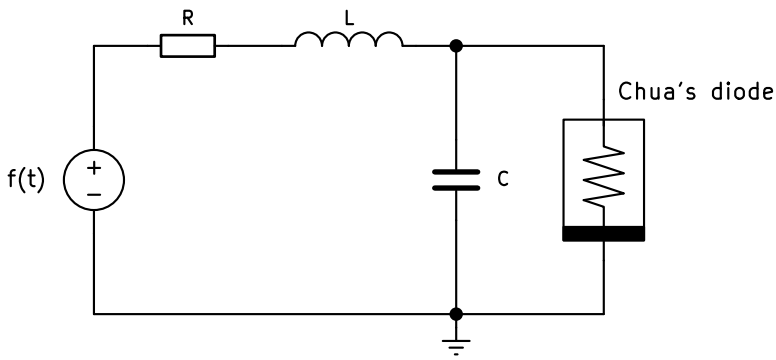


Figure 2.10: Driven Chua's circuit. Figure from Paper B (Jensen and Tufte, 2017).

ODE. In this way, electrical phenomena are exploited to effectively integrate the differential equations in the analog circuit. In an analog implementation, the variables from the original ODE take the form of electrical quantities such as voltage and current. The time-evolution of the (integrated) ODE can then be monitored in real-time using electrical measurement equipment such as an oscilloscope.

For the purpose of material computation, an analog circuit is a physical computing substrate in which we have a great deal of control. It offers a direct route to a physical system with specific characteristics, e.g., with a desired dynamic regime. As discussed in Section 2.2, computational capabilities peak when the dynamic regime lie at the edge of chaos. We are therefore interested in a circuit which exhibits rich dynamics, with the ability to tune the dynamic regime from stable to chaotic.

The chaotic circuit (CC) used in this work is the driven Chua's circuit, which is one of (if not the) simplest known circuit that exhibits chaos (Murali et al., 1994a,b). The circuit (Fig. 2.10) consists of only a handful of components: three linear elements (a resistor, an inductor and a capacitor) and a nonlinear resistor (a Chua's diode). An external periodic forcing signal $f(t)$ drives the dynamics of the circuit. It is a two-dimensional dynamical system: the system variables correspond to the voltage across the capacitor C, and the current through the inductor L.

Circuit dynamics depend on the choice of components, the nonlinear characteristics of the Chua's diode, and the properties of the external forcing signal. Fig. 2.11 shows the circuit implemented on a printed circuit board. Varying the amplitude and frequency of the external forcing signal allows different dynamical regimes to be obtained. Fig. 2.12a shows a phase plot of a stable regime, as measured on an oscilloscope in XY mode where the horizontal axis corresponds the voltage across

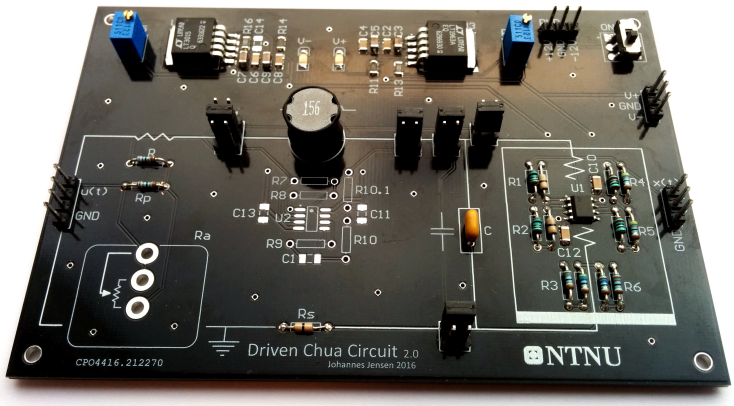
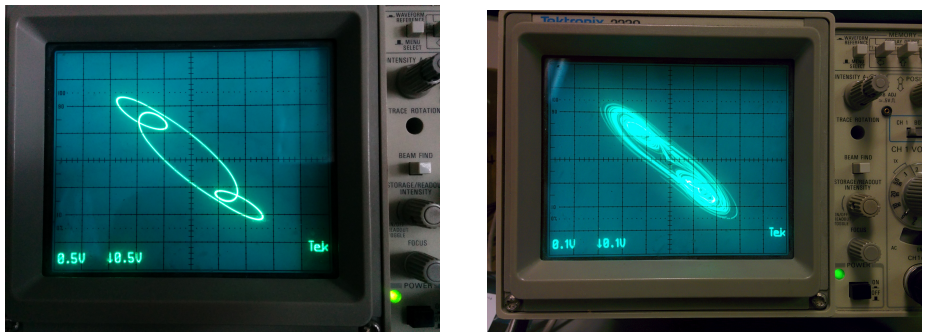


Figure 2.11: The driven Chua's circuit implemented on a printed circuit board.



(a) **(b)**

Figure 2.12: Phase plots of the driven Chua's circuit, measured on an oscilloscope in XY mode. The horizontal axis corresponds to the voltage across the capacitor, and the vertical axis represents the current through the inductor. Shown are plots of the (a) stable dynamic regime, and (b) chaotic dynamic regime.

the capacitor, and the vertical axis represents the current through the inductor. As can be seen, the trajectory is cyclic and stable. Fig. 2.12b depicts a chaotic regime, where the trajectory orbits a strange attractor.

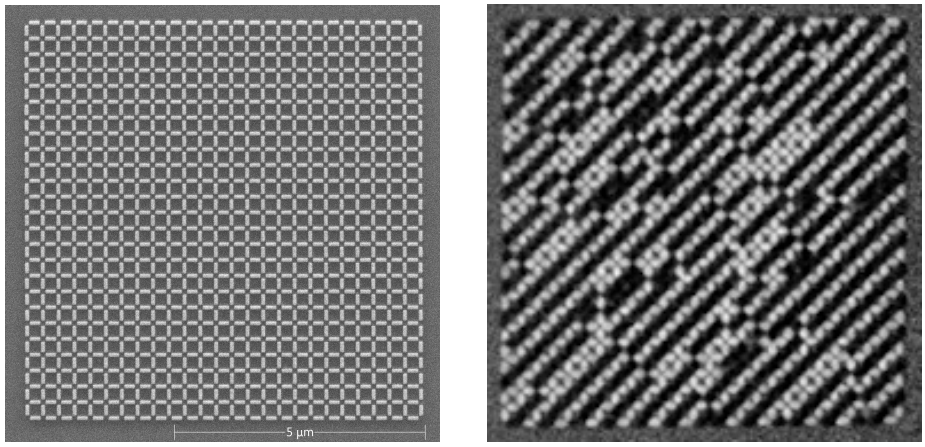
Further details about the circuit, including circuit equations, are given in *Paper B: Reservoir Computing with a Chaotic Circuit*.

2.6.1 Computation in chaotic circuits

Using chaotic systems as computational building blocks, aptly named "chaos computing", taps into the rich repertoire of behaviors available in chaotic systems (Ditto et al., 2008; Munakata et al., 2010). It has been hypothesized that chaos plays an important role in natural systems by providing a rich repertoire of behaviors that may be utilized for various tasks (Sinha and Ditto, 1998). The idea is that a single chaotic element can be used to implement different logic gates and arithmetic tasks (Munakata et al., 2002; Sinha et al., 2002). Furthermore, the ability to easily switch between different operations could enable more dynamic computer architectures. Several physical realizations of chaos computing using analog circuits have been demonstrated (Murali et al., 2005b,a; Ditto et al., 2008).

As discussed (Section 2.4), chaos plays a central role in reservoir computing. Indeed, classical RNN reservoirs are known to be chaotic for certain parameters. Chaotic reservoirs may still be usable as long as the input is sufficiently strong to drive dynamics out of the chaotic regime (Ozturk and Principe, 2005).

However, the low-dimensionality of (most) chaotic circuits poses a problem, as reservoir computing requires a high-dimensional dynamical system. One approach is to use a time-multiplexing technique, where the low-dimensional output is sampled over time to produce a virtual high-dimensional system (Appeltant et al., 2011). Time-multiplexing comes at a cost, however, as each input is presented to the reservoir for an extended period of time, and consequently memory of past inputs is lost in the process. This can be remedied by adding a delay line which provides delayed feedback to the reservoir, e.g., using digital electronics (Appeltant et al., 2011) or fiber optics (Paquot et al., 2012). Alternatively, if non-temporal tasks are considered, time-multiplexing can be used without a delay line.



(a) (b)

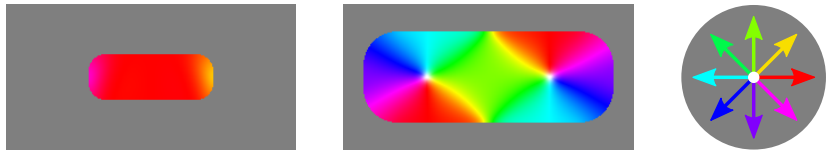
Figure 2.13: Images of 25×25 square ASI, consisting of 1300 nanomagnets. (a) SEM image shows the structure of the array, where each elongated rectangle is a nanomagnet with dimensions $220\,\text{nm} \times 80\,\text{nm} \times 20\,\text{nm}$. (b) PEEM image reveals magnetization direction of the magnets, where white represents right or up, and black represents left or down. Images courtesy of Anders Strømberg.

2.7 Artificial Spin Ice

Artificial spin ice (ASI) are systems of nanoscale ferromagnetic islands (nanomagnets) arranged on a 2D lattice. Collectively, the nanomagnets form a magnetic metamaterial, whose emergent properties can be controlled directly by the placement, orientation and shape of the magnets. Established nanofabrication techniques coupled with the ability to directly observe magnetic state, has enabled the study of a wide range of physical phenomena in ASI.

Fig. 2.13a shows a fabricated "square" ASI, imaged using a scanning electron microscope (SEM). Each elongated rectangle in the image is a nanomagnet with dimensions $220\,\text{nm} \times 80\,\text{nm} \times 20\,\text{nm}$. The entire 25×25 array consists of 1300 such nanomagnets, arranged on the square lattice. Fig. 2.13b shows the magnetic state of a square array, imaged using photoemission electron microscopy (PEEM). In the PEEM image, white represents magnetization pointing right or up, while black represents left or down. The emergent stripe pattern is the result of magnetic interactions between the nanomagnets.

ASI has received considerable interest over the last decade, primarily as a model system for the study of fundamental physics. The name "artificial spin ice" stems



(a) (b)

Figure 2.14: Internal magnetization of a (a) single-domain magnet and a (b) multi-domain magnet. The volume and shape of a nanomagnet determines its stable internal magnetization. Magnetization direction as indicated by the color wheel (white represents zero magnetization).

from the use of engineered systems to mimic the arrangement of molecules in water ice. Square ASI was the first ASI system to be proposed, and was shown to obey the so-called "ice rules" of molecular spin ice materials (Wang et al., 2006). Since then, a rich variety of ASI systems have been investigated, ranging from repeatable structures to stochastic disordered arrays. A wide range of emergent phenomena has been discovered in ASIs, e.g., collective ferromagnetic/antiferromagnetic ordering (Sklenar et al., 2019), domain wall propagation (Li et al., 2019), avalanche dynamics (Mengotti et al., 2011), and phase transitions (Levis et al., 2013). For a recent review of the ASI field, see Skjærvø et al. (2020).

2.7.1 Artificial spins

In ASI, each nanomagnet behaves as a *mesoscopic spin* with an effectively binary state. The small size ensures a uniform internal magnetization (a single-domain state), while an elongated shape constrains the direction of magnetization to lie along the long axis (a binary state). Fig. 2.14a shows the stable internal magnetization of a 220 nm × 80 nm × 15 nm magnet, as obtained from micromagnetic simulations, where the internal magnetization is randomly initialized and subsequently relaxed (at zero field) to obtain a low energy state. As can be seen, the magnetization is uniform and oriented along the long axis (pointing towards the right). Fig. 2.14b shows the same picture for a magnet of size 440 nm × 160 nm × 15 nm. Due to its larger size, its internal magnetization is no longer uniform, and two vortex structures have formed (centered at the white spots). A single-domain magnet will have a larger stray magnetic field than a multi-domain magnet, since all the atomic spins are aligned in the same direction and sum to a net positive magnetic moment. In a multi-domain magnet, on the other hand, the atomic spins are not aligned and their moments tend to cancel out.

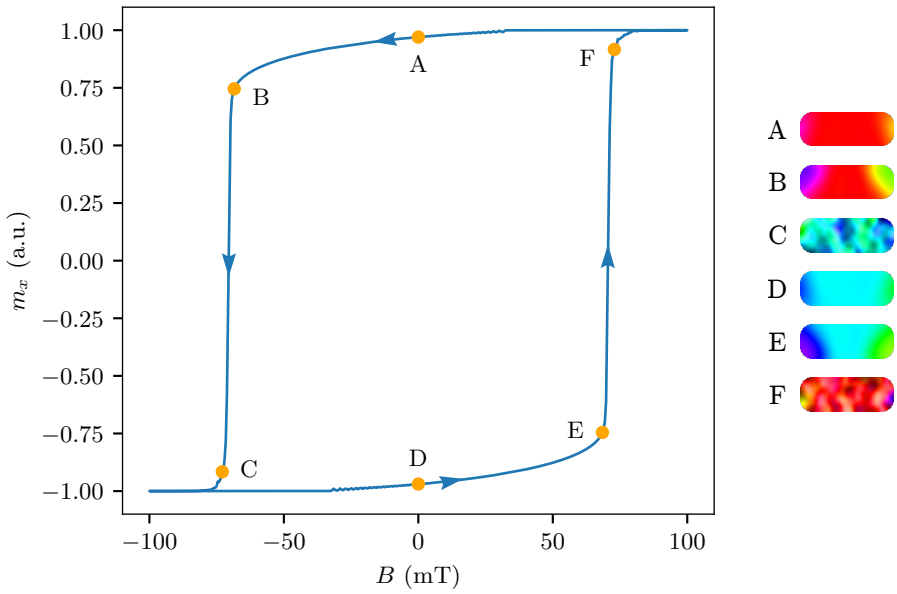


Figure 2.15: Hysteresis curve of a 220 nm × 80 nm × 15 nm magnet, when subject to an external magnetic field B applied along its long axis. m_x is the normalized magnetization along the long axis of the magnet. The images on the right show the internal magnetization of the nanomagnet at the different stages of reversal as marked in the curve. Colors indicate magnetization direction according to the color wheel in Fig. 2.14.

Magnetization reversal, or switching, is an inherently nonlinear process. If a magnet is subjected to an external magnetic field, which is sufficiently strong and directed in the opposite direction of its magnetization, the magnetization will switch direction. Fig. 2.15 plots long-axis magnetization m_x versus the strength of an external magnetic field B, applied along the long axis of the magnet. As can be seen, the magnetization follows a hysteresis curve, where reversal happens at $B \approx \pm 70\,\text{mT}$. The image insets show the internal magnetization of the nanomagnet at different stages of reversal. The critical field strength where switching occurs is referred to as the *coercive field*, which is an intrinsic property of the magnet, determined by the ferromagnetic material, the shape and size of the magnet, as well as the angle of the external field.

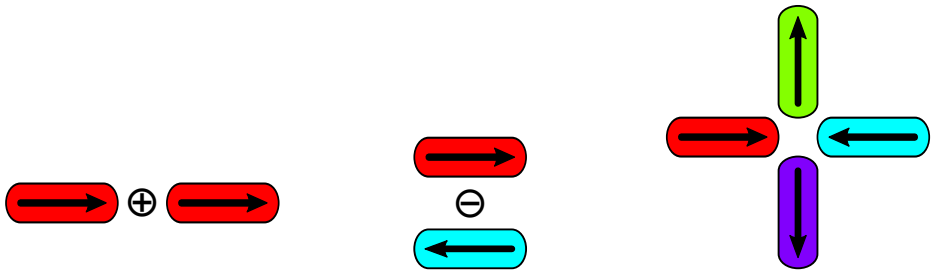


Figure 2.16: The relative arrangement of magnets determine the dipolar coupling between them. Shown here are two arrangements which results in a coupling where (a) aligned and (b) anti-aligned states are energetically favorable. (c) A geometrically frustrated system where all competing interactions cannot be satisfied at the same time. The state shown is one of the two configurations with the lowest energy. Colors indicate magnetization direction according to the color wheel in Fig. 2.14.

2.7.2 Magnetic interactions and frustration

When placed close together, the nanomagnets are coupled via the magnetic dipole-dipole interaction: each magnet is subject to the stray magnetic field of nearby magnets. The strength and nature of the coupling depends on the distance and relative orientation of the magnets. Fig. 2.16 shows different arrangements of nanomagnets which results in a coupling where aligned (Fig. 2.16a) and anti-aligned (Fig. 2.16b) states are energetically favorable.

Based on the placement and orientation of the magnets, it is possible to create a *geometrically frustrated* system, i.e., a system in which all competing interactions cannot be satisfied at the same time. Fig. 2.16c shows a system of four magnets which exhibits geometrical frustration. No configuration of states exist in which the energy of all four magnets is minimized. The state shown is one of two possible configurations with the lowest energy, obeying the so-called *ice rule* with two spins pointing in and out of the center. The relative strength of the coupling also depends on the geometrical arrangement, i.e., some arrangements will be dominated by nearest-neighbor interactions, while others will have significant long-range interactions.

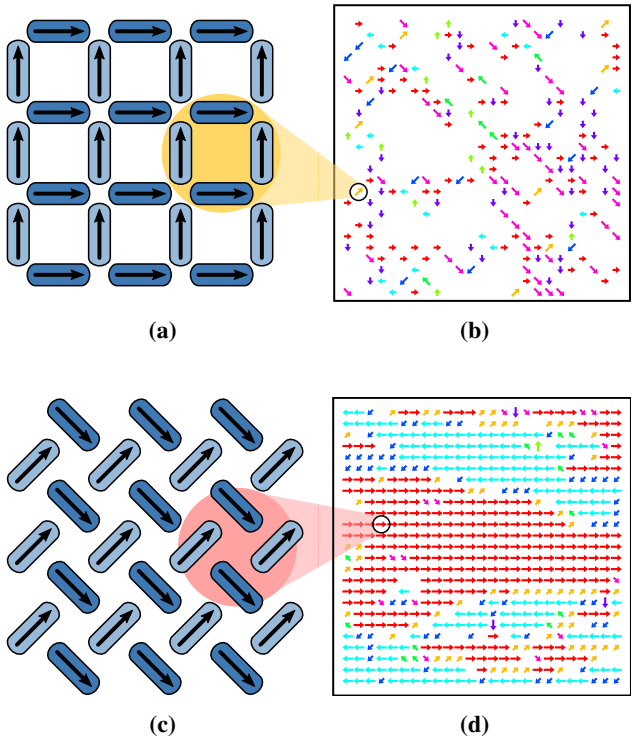


Figure 2.17: The emergent behavior of an ASI is a result of the placement and orientation of the nanomagnets. (a) Square ASI consists of horizontal and vertical magnets arranged on a square lattice. (b) Square ASI favors antiferromagnetic order, resulting in domains of zero net magnetization (white regions). (c) Pinwheel ASI is obtained by rotating each magnet in square ASI by 45° about its center. (d) Pinwheel ASI exhibits long-range ferromagnetic order, supporting formation of domains with coherent magnetization. The systems shown in (b) and (d) are 25×25 square and pinwheel ASI, respectively, each consisting of 1300 magnets. The arrows in (b) and (d) indicate the collective magnetization of the four circled magnets in (a) and (c), respectively. Figure from Paper E (Jensen and Tufte, 2020).

2.7.3 Geometry and emergence

The particular arrangement and orientation of the magnets in an ASI is referred to as the *geometry*, which effectively defines the nature of the magnet-magnet interactions. Fig. 2.17a depicts "square" ASI, which consists of horizontal and vertical magnets arranged on two square lattices. The sub-lattice with vertical magnets is placed at an offset from the sub-lattice with horizontal magnets, as indicated by the different colors. "Pinwheel" ASI is shown in Fig. 2.17c, and is obtained by rotating each magnet in square ASI by 45° about its center.

Geometry is the principal source of emergent phenomena in ASIs. Some geometries result in the emergence of antiferromagnetic ordering, where domains of zero net magnetization are energetically favorable. Fig. 2.17b shows the collective magnetization in square ASI, with the emergence of such antiferromagnetic domains (white regions). Pinwheel ASI, on the other hand, exhibits ferromagnetic order, i.e., the magnets form domains with coherent magnetization of non-zero magnitude, as shown in Fig. 2.17d. It is quite remarkable how such a small change (rotating the magnets), can completely alter the macroscopic behavior, all the while the ferromagnetic material, amount of material, magnet density, magnet shape, underlying lattice and so on stay the same.

Due to the emergence of patterns at higher scales, ASIs can be considered *metamaterials*. When observed at larger scales, magnetic patterns emerge as a result of interactions between the underlying mesoscopic spins. Compared to bulk materials, metamaterials offer considerable control and flexibility, and opens for the design of exotic substrates with unusual physical behavior.

There are a myriad of ways to tune the behavior of ASIs. For example, the lattice spacing (characteristic distance between magnet centers) determines the size of the anti- or ferromagnetic domains: a smaller spacing results in larger domains. Small changes to the geometry can result in fundamentally different behavior, as illustrated by the significantly different properties obtained when square ASI is modified, ever so slightly, to obtain pinwheel ASI. Novel geometries provide a seemingly endless playground for exploration. In addition, there are several ways to tune behavior externally, without altering the system, e.g., through an external magnetic field or temperature.

2.7.4 Fabrication, input and output

Fabrication of ASIs is done using lithographic methods originating from the semiconductor industry. Electron beam lithography (EBL) can be used to create ASI samples with arbitrary geometries in the lab. Nanomagnets are produced on a silicon wafer using a lift-off process with a layer of magnetic material (NiFe) covered by a metal oxidation barrier (Al). Common magnet dimensions are $220\,\text{nm} \times 80\,\text{nm}$ with a thickness in the range 5 nm to 25 nm. While EBL can create patterns with a lateral feature size down to 5 nm, practical limitations in fabrication of ASI systems limit the magnetic spacing to around 10 nm. There are few limitations when it comes to the number of magnets, limited only by the size of the silicon wafer and, most significantly, EBL patterning time.

There are many ways to perturb ASI systems. The use of external magnetic fields is a well-established approach, applied either globally to the entire array or locally to specific areas, e.g., via current-carrying electrical wires. Temperature is routinely used to effectively "anneal" ASI systems to obtain low energy states. Other possibilities include current-induced torques (Brataas et al., 2012), optically induced switching (Le Guyader et al., 2015) and using a scanning probe to manipulate individual nanomagnets (Gartside et al., 2018).

A key feature of ASIs is the ability to directly observe the magnetic state of the individual nanomagnetic spins. It is thus possible to observe phenomena at multiple scales, ranging from the state of individual spins, to spatial patterns of neighboring spins, to the collective state of the entire ensemble. Magnetic force microscopy (MFM) can be used in the lab to resolve magnetization direction of individual nanomagnets, while even higher resolution images can be obtained at a synchrotron using PEEM. Readout techniques beyond the lab include magnetic tunnel junctions (MTJ) and anisotropic magnetoresistance (AMR).

Simulations of ASI systems generally fall into two categories. Micromagnetic simulations capture the internal magnetization dynamics of individual magnets, and hence provide a very accurate picture. With GPU acceleration, ASI ensembles with hundreds of magnets can be simulated micromagnetically within practical time frames (Leliaert et al., 2018). To simulate larger systems, a mesoscopic model of the magnets is needed, where detail is sacrificed for speed. Treating single-domain magnets as point dipoles is a common approach, where the state of each magnet is approximated as a single magnetization vector (Budrikis, 2012).

2.7.5 Computation in ASI

Nanomagnetic computation is inherently energy efficient, due to the use of spin instead of electric charge as information carriers (Salehi Fashami et al., 2011). The energy dissipated in magnetic switching is orders of magnitude lower than transistor-based switching, in some cases approaching thermodynamic limits (Lambson et al., 2011). In addition to low-energy operation, nanomagnets are non-volatile devices which retain state without power, and exhibit radiation hardness.

Nanomagnetic logic (NML), the construction of digital logic using nanomagnetic elements, has attracted considerable interest as an alternative to the electronic digital computer (Imre, 2006; Niemier et al., 2011). Logic gates based on ASI have been proposed (Gypens et al., 2018), including gates that are thermally active (Arava et al., 2019). Still, there are several fundamental challenges which must be solved for NML to be a viable contender, in particular related to reliability of computations and robust magnetic interconnects.

If alternative computing paradigms are considered instead, we may be able to exploit the inherent energy efficiency of nanomagnetic systems, without the engineering difficulties associated with NML. Probabilistic computing based on annealing is one approach, where a computation is encoded in an Ising Hamiltonian, whose energy minima represent a solution. A magnetic system can then be engineered with the same Hamiltonian, and subsequently annealed to find the solution (Debashis et al., 2016; Bhanja et al., 2016; Sutton et al., 2017). Recently, a nanomagnetic Galton board was demonstrated, with the ability to tune the resulting probability distribution (Sanz-Hernández et al., 2021).

In addition to the work presented in this thesis, reservoir computing based on ASI has recently received interest elsewhere (Hon et al., 2021). Reservoirs based on nanomagnetic rings have been proposed, based on similar principles (Dawidek et al., 2021). Also related are reservoirs based on dipolarly coupled nanomagnets (Nomura et al., 2019).

Nanomagnetic systems have long been promoted as promising candidates for future computing devices, but so far the technology has failed to compete with semiconductor digital logic. Perhaps it is as unconventional computing substrates the full potential of nanomagnetic systems will be realized?

Chapter 3

Discussion

The thesis concludes with a discussion of the topics raised by the research questions, how they are addressed by the papers, and identifies important directions for future work. While the discussion is loosely organized around the research questions, there is some overlap in some of the topics, in particular related to reservoir computing (RQ3) and computational properties (RQ1).

Throughout this research, a variety of material properties have been investigated in relation to computation. Towards that end, three material substrates have served as subjects of study, with distinct properties: a complex, unstructured carbon nanotube material (CNT), a simple chaotic circuit (CC) and a family of complex, structured magnetic metamaterials called artificial spin ice (ASI).

We begin the discussion with the first research question (RQ1):

Research question 1

What properties of a material are important for computation, and how do we quantify these properties?

It can be argued that the answer to this question depends on the particular computational task we are interested in. While it is certainly true that some properties are more important for some types of tasks and not so for others, there exist fundamental mechanisms which are needed for *any* type of computation. The primary focus of this research has been on exactly such *intrinsic* properties.

We have considered both *structural* and *emergent* properties. Structural properties include: nonlinearity, dimensionality, local interactions and self-organization. Emergent properties include: rich dynamics, information storage and manipulation, input separation, approximation and fading memory.

3.1 The importance of dynamics

Throughout this research project, *dynamics* has been the principal vehicle for understanding the computational properties of a material. As we have discussed, dynamics is fundamentally what gives rise to intrinsic computational mechanisms such as information storage, transfer and manipulation (Section 2.2). Without dynamics, there cannot be computation (Section 2.1).

Dynamical systems theory provides a solid foundation for our investigation, and forms a bridge between the physical system and its intrinsic computation. Thinking about computation in terms of stability, attractors, chaos and bifurcations offers insight into the kind of computation a physical system does *naturally*.

There is an important distinction between the dynamics of classical dynamical systems theory and the dynamics related to computation. Dynamical systems theory is mainly concerned with the behavior of the *free-running* system, i.e., where the system is left to run on its own, under well-defined conditions. For driven (non-autonomous) dynamical systems, this amounts to fixing all external signals to well-defined functions of time. The principal interest lies on the asymptotic (long-term) behavior of the system.

Computation, in contrast, introduces the notion of an abstract input which is encoded into the domain of the dynamical system (Section 2.1). When a dynamical system is used for computation, input (and output) means there is always a connection between the dynamical system and the environment. Thus, computing is concerned with the behavior of the *input-driven* system, and the relationship between the input and dynamics. In this case, the transient (short-term) dynamics can be very relevant. It is with input-driven dynamics we approach the computational properties of a material. Still, the two perspectives are by no means orthogonal: free-running and input-driven dynamics are typically related.

3.2 Measures of dynamics and computation

By observing the free-running dynamics of a system, we obtain an understanding of its general dynamical properties, the effect of different parameters, etc. We get a birds-eye view of the behavior of the system, which is crucial when considering its potential for computation, deciding how to encode input and decode output, and ultimately finding a suitable computational model. The particular input encodings used in this work have been based largely on what was learned from the

free-running dynamics of the respective system. Papers A, B and C all begin with a study of the free-running dynamics of the corresponding material.

It is with the input-driven system we can really begin to explore the computational properties of a material in greater depth. We have employed several measures to quantify key computational properties, which can all be related to dynamics.

Complexity metrics is an effective, albeit crude, method to quantify the richness of dynamics emitted from a black-box system such as the CNT material. We showed that to isolate the complexity contribution of the material itself, the complexity of the input signal must be taken into account (Paper A).

When the physical system is simple and well-understood, methods from dynamical systems theory can be applied more or less directly to computation. A *bifurcation diagram of transients* is an effective way to visualize system output under different input encoding schemes (Paper B).

For complex systems with many moving parts and a well-defined state (such as ASI), *state space analysis* provides valuable insight into several computational properties (Paper C). The number of unique states visited during a trajectory is an indicator of the dynamical regime, and quantifies richness in behavior. For the input-driven system, the number of unique *final* states indicates the mode of computation, with classification and memory at two extremes. State transition graphs can be used to uncover computational mechanisms in a system, such as information storage.

Finally, reservoir computing metrics allow deeper insight into the computing capabilities of a system (Paper E). *Kernel-quality* (KQ) is effectively a measure of the separation property, and indicates the discriminatory power of material state. *Generalization capability* (GC) quantifies the material's sensitivity to similar inputs – the computational analogy to sensitivity to initial conditions from dynamical systems theory. Measured over time, GC can also reveal the presence of fading memory, a key property for temporal information processing. Combined, KQ and GC provide an overall assessment of computing capacity.

The abovementioned methods by no means exhaust the ways to quantify computational properties. A range of other methods are available to capture similar, overlapping and orthogonal properties, e.g, methods based on information theory (Langton, 1990), statistical complexity (Crutchfield and Young, 1990), memory capacity (Jaeger, 2002) and information processing capacity (Dambre et al., 2012).

3.3 Nonlinearity and dimensionality

Among the structural properties, *nonlinearity* is a prerequisite for complex dynamics and consequently a key ingredient for any non-trivial computation (Section 2.3.6). Hence all the materials considered in this work exhibit nonlinearity at some level. However, it should be mentioned that although CNTs have an inherent nonlinearity, this was not observed directly at the level of our electrical measurements in the bulk material. Still, the complex dynamics observed does indicate that nonlinear mechanisms are at play.

The *dimensionality* of a system is often used as an indication of its potential computing capacity. For a silicon chip, the number of transistors is treated as a measure of inherent computing capabilities. For a material, however, what constitutes a computational element is not so clear. As we have seen with ASI, it is possible to define computational elements at different levels of observation, from individual spins to the collective state of the metamaterial. The situation is similar in bulk materials, where the granularity of measurements to a certain extent defines the computational elements.

In reservoir computing, the computing capacity is bounded by the *output dimensionality* of the reservoir (Dambre et al., 2012). We refer to the output dimensionality as the number of computational nodes. It is an indication of the amount of "freedom" to express different types of computations at the readout layer. In many cases, the number of output nodes may equal the dimensionality of the system, as is the case with RNNs and ASI (Paper C). However, as demonstrated in Paper B, the number of nodes can exceed system dimensionality, e.g., by employing techniques such as time-multiplexing. Such techniques do come at a cost, however, imposing a serial mode of computation which effectively hides properties such as fading memory.

If a material exhibits emergent high-level patterns, it is possible to use an average coarse-grained view of the system as output, which reduces the output dimensionality but with a potential increase in output *resolution*. This was demonstrated for ASI in Paper E, where the effects of coarse-grained outputs were explored in detail. For bulk materials, such as CNT nanocomposites, we typically always obtain a very coarse-grained view of system state, and consequently the number of nodes will be much lower than the dimensionality of the underlying material.

Theoretically, the computing capacity of a reservoir is maximized if all the nodes are linearly independent and have the fading memory property. Of course, the actual computing capacity of a material will typically be well below the theoretical

upper bound. We quantified the computing capacity of a family of ASI reservoirs in Paper E, which was very promising, but still below the theoretical limit. Nevertheless, there are trade-offs between maximizing computing capacity and robustness: redundant node function increases robustness, at the cost of total computing capacity.

3.4 Local interactions and self-organization

Local interactions and *self-organization* are important properties of natural computing systems, and have been highlighted as particularly relevant for material computation (Section 2.3.5). Of the materials explored in this thesis, only ASI clearly demonstrate these properties. Magnetic coupling facilitates local interactions between magnets, which results in the emergence of spatio-temporal patterns at larger scales. We found that magnetic coupling plays a key role in the computational capabilities of ASI: for the number of reachable stable states, the formation of memory and the separation of temporal input patterns (Papers C and E). Although it can be argued that CNTs also exhibit these properties at the lower levels, it was not apparent at the level of our electrical measurements of the bulk material.

Repeatedly, we have seen how self-organization at lower levels can give rise to computational properties at higher levels, i.e., from the "physics" level to the "material" level in Fig. 2.4. The metamaterial properties of ASI has allowed us to explore three levels in the scale hierarchy, from the micro level of internal magnetization, to the meso level of individual spins, to the macro level of collective spin organization. In Paper C we found that micromagnetic state acts as information storage, which gives rise to an apparent memory effect in the dynamics of the mesoscopic spins. Continuing up the hierarchy, in Paper E, we explored how the computational capacity is affected by the scale of observation, from the meso level of individual spins to the self-organized patterns of multiple spins at the macro level.

Local interactions can also be related to dynamics, in that it mediates the connectivity of the network and controls the flow of information. Connectivity is a key parameter for controlling network dynamics. In random boolean networks (Kauffman, 1969), sparse connectivity is associated with stable dynamics, whereas dense connectivity results in a chaotic regime. A critical dynamical regime is found in the transition from sparse to dense connectivity (Aldana et al., 2003). Similar observations have been made in relation to RNNs in RC (Büsing et al., 2010). Intuitively, when connectivity is primarily local, information is more easily maintained over longer time scales, since each node is affected by a limited set of neighbors.

3.5 Controlling computational properties

The computational properties of a material emerge from its underlying physical properties. Throughout this research project, we have uncovered many connections between physical material properties and computational capabilities. While the first research question is concerned with the relationship between physics and computation, the focus of the second research question is the dynamics of this relationship.

Research question 2

How can the computational properties of a material be controlled to tune functionality?

How does a change in physical parameters affect computation? To what degree can the computational properties be *controlled* by turning the available physical knobs? As we discover how the various physical parameters affect computation, a key question is whether the relationships are predictable enough to support reliable control of functionality.

We have argued that the ability to control the computational properties of a material is crucial to obtain a kind of programmability. At its core, programming amounts to changing the *function* of a computer according to the wish of the programmer. Similarly, if the computational properties of a material can be reliably controlled and changed as we see fit, we have the ability to change the function of the material. Granted, changing computational properties may not offer as detailed control over function as writing a computer program. However, it is exactly when we let go of such detailed control we discover what computation a material does *naturally*.

As discussed, computational properties can be modified through changes in the physical parameters. It is natural to distinguish between internal and external physical parameters. *Internal parameters* are tied to the structure of the material itself, and can (typically) only be changed at the point of fabrication. *External parameters*, on the other hand, are accessible from the outside of the material, and can potentially be changed at runtime. From the materials investigated in this research, examples of internal parameters include: the percentage weight of single-walled carbon nanotubes (CNT), the nonlinear characteristics of the Chua diode (CC), the spacing between magnets (ASI), the orientation of magnets (ASI) and the number of magnets (ASI). Examples of external parameters include: the amplitude and fre-

quency of an external driving signal (CNT, CC and ASI), the number and location of input signals (CNT) and output granularity (ASI).

3.6 Controlling dynamics

As discussed earlier (RQ1), dynamics is fundamentally tied to computation and therefore of primary interest also here in RQ2. Finding ways to control and change the dynamics of a material will ultimately give us a powerful knob for controlling its computational properties. Towards that end, we have explored both internal and external parameters.

All the materials investigated in this work have been driven dynamical systems, where an external input signal drives the dynamics of the system and provides a source of energy. In all cases, the properties of the input signal have had a significant effect on the dynamics of the system. As we have seen, parameters such as amplitude, frequency, waveform and input location all affect dynamics in some way. Being external to the material, such parameters offer a convenient set of knobs with which to control dynamics and ultimately the computational properties.

When the frequency of the input signal is sufficiently high, effects at the lower levels come into play. In ASI we saw how frequency of the input signal can be used to excite magnetization dynamics within each nanomagnet (Paper C). These microlevel dynamics give rise to a memory effect at the macro level, where the state of the ensemble exhibits an apparent dependency on input history. This emergent phenomenon is the result both of micro level dynamics and local interactions between magnets at the macro level. As information about past input is maintained in the spatial state of the ensemble, the system effectively separates input into distinct states. We found that the sensitivity of the separation can be tuned by the amplitude of the input signal. Interestingly, the sensitivity can be tuned in a continuous manner, meaning a spectrum of function is available within the system. That such a degree of control is possible through an external parameter is exciting for the prospects of ASIs as computational substrates.

Rich input-driven dynamics are also available in ASI at low input frequencies (Paper E). In this case, the dynamics emerge solely from the local interactions at the macro level. Adjusting the interaction strength allows the dynamics to be controlled to support information storage and manipulation. We saw how the magnetic coupling strength can be used to control the system's sensitivity to different input patterns, i.e., tuning of the separation property. At a critical coupling strength, the

system exhibits a chaotic dynamical response to the input and sensitivity is maximized. In the chaotic regime, the system never forgets past input history. More computationally useful is a regime where past input history is gradually forgotten. We found that increasing the coupling strength allows such a regime to be obtained, i.e., exhibiting the fading memory property. Computational tasks require different sensitivity and amount of memory, hence the ability to reliably control these properties is an important finding.

Rich dynamics is inherently tied to nonlinearity. Some physical systems (such as our CC circuit) offer precise control of nonlinearity and a very fine-grained way to tune node function (Paper B). More often, however, the nonlinearity of a material is the result of its underlying physics where we have limited access. The nonlinear switching characteristics of nanomagnets, for instance, is tied to the properties of the ferromagnetic material itself. Still, the switching characteristics can be influenced to a certain extent by altering the shape of the nanomagnets.

3.7 Dimensionality and scalability

Scalability is concerned with the relationship between the number of computational elements and performance. For a computing system to be scalable, adding physical resources should ultimately result in an increase in performance. Physical resources include the material itself (the amount of "stuff" within), as well as the physical interface to the system (number of input and output signals).

For the CNT material, the number of computational elements is not well-defined, due to the inherent unstructuredness of the nanocomposite. Therefore, it is difficult to say anything about the scalability of such materials, without further detailed inquiry. What we did see, however, was that increasing the density of SWCNT in the nanocomposite resulted in a reduced output complexity (Paper A). In other words, fixing the material volume and adding more "stuff" does not yield scalability.

For structured metamaterials like ASI, the number of internal components is well-defined and countable (number of magnets). In other words, the material volume effectively scales with the number of components. This allows detailed investigations to be conducted on the scalability of the output nodes. How does computing capacity scale with the amount of material being used as output? Instead of observing each internal component directly, can we get away with an aggregate view of system state? In Paper E we found that the large-scale patterns in pinwheel ASI were fairly robust under an aggregate (coarse-grained) view. An interesting

discovery was that computing capacity per output node would, in some cases, *increase* under an aggregate view. This implies that for a given ASI system, there exists an observation level in which computational capacity per output node can be maximized. In other words, controlling output dimensionality can have direct computational benefits.

3.8 Controllability of unstructured and structured systems

For different materials, the amount of control over physical parameters varies greatly. We have characterized the materials in this research as either *structured* or *unstructured*, depending on how orderly their constituent parts are arranged. The presence of structure has important implications for the controllability of parameters, and consequently also computational behavior.

Unstructured materials offer rather limited control over parameters. For the unstructured, bulk CNT material, we found that the frequency and location of input signals plays a significant role in the computational complexity of the emitted output (Paper A). Due to the unstructuredness of the material, there will be large spatial variations in internal structure which ultimately affect the computational function. Correspondingly, correlations between parameters and dynamics are often arbitrary, and ultimately tied to the specific material sample. The amount of control offered over dynamics in such unstructured materials is thus very limited.

Structured materials, on the other hand, offer much more control over physical parameters and consequently also dynamics and computational properties. The availability of physical models opens for more systematic investigations of how different parameters affect computation. With the CC circuit we saw how the nonlinearity of the Chua diode directly influenced task performance, revealing a complex parameter landscape with salient features. For such a simple system, an exhaustive search was possible due to the limited number of parameters. As the number of parameters grow, exhaustive sweeps are no longer viable and we can only explore a subset of the parameter space. Metamaterials such as ASI offer unprecedented control over internal parameters. Although the ASI parameter space is vast, and we have explored only a small part, we have found many physical knobs which directly control computational properties.

3.9 Reservoir computing *in-materio*

Placing a material within a computational model allows useful computational tasks to be solved *in-materio*. Furthermore, a model enables us to analyze and reason about how computation takes place within a material.

The topic of the final research question is reservoir computing *in-materio*, i.e., the use of a reservoir computing (RC) model for material computation.

Research question 3

How can reservoir computing help elucidate and exploit the intrinsic computational properties of a material?

Efficient computing depends on a good correspondence between the physical system and the abstract computational model. In other words, the abstract properties of the model should be accommodated by the physical properties of the system. Because RC is such a generic computational model, its requirements can be fulfilled by many physical systems.

RC is a model inspired by biological computing systems. Hence its properties are inherited from an existing physical system, namely biological neural networks. As we have discussed (Section 2.3), many of the properties of neural networks can be transferred to other physical systems, also non-biological ones. Physical systems consisting of large number of nonlinear nodes governed by local interactions are seemingly abundant in nature. If such physical systems can be reliably controlled, they can potentially be exploited for reservoir computing.

The close connection between RC and dynamical systems theory makes RC particularly suitable for exploring computation in physical systems. If a material can be usefully described as a dynamical systems model, its computational properties can be elucidated from a RC perspective. From our discussion on RQ1, it is clear that a RC perspective opens for powerful analysis and reasoning about the computational properties of a material: from dimensionality and nonlinearity (Section 3.3), to connectivity and dynamics (Section 3.2).

The programming model of RC is very apt for material computation. Programming a reservoir computer involves training the linear readout layer, which is separated from the reservoir itself. This distinct feature relaxes the computational requirements of a material a fair bit. A material reservoir does not need to provide some specific computation directly, but rather its repertoire of intrinsic computations are

combined by the readout layer. Hence, as a computational model, reservoir computing is ideal for exploiting the *natural* computations within a physical system.

In Paper E we applied RC metrics to quantify the computing properties of an ASI. We saw how system parameters affected computational capacity in terms of RC metrics. Since the RC metrics are tied to a well-defined computational model, their results can be directly related to useful computation. For instance, the KQ/GC quality metrics indicate the number of *useful* nodes within the system. They also quantify the expressiveness of system state for performing signal classification.

We have also seen that the requirements of the RC model does not necessarily need to be fulfilled by the physical system itself. In Paper B, a low-dimensional dynamical system could be treated as a virtual high-dimensional system by using time-multiplexing. Furthermore, even though the system then lacks (fading) memory, it can still be exploited for reservoir computing for non-temporal tasks. In a way, this paper illustrates the entire process of reservoir computing *in-materio*: from simulation to physical system, from intrinsic properties to useful computation, and the important choice of input encoding and output decoding.

Although the field of reservoir computing *in-materio* is still young, the results are already very promising. RC is a powerful model for material computation, both for exploiting and understanding the intrinsic computational properties within.

3.10 Conclusions and future work

What kind of materials are ideal for computation is still an open question. In this research project we have explored three materials with distinct properties, ranging from unstructured to structured, from simple to complex, exhibiting various degrees of controllability.

What is clear is the major advantage offered by a *physical model*, as a vehicle to understand the relationship between physical properties and intrinsic computation. Therefore, in assessing the computational potential of a material, the existence of physical models should carry significant weight.

We have also argued for using a well-defined *computational model* in the exploration of material substrates. Reservoir computing is a very general model which can be applied to a range of different physical systems. Still, there are other computational models which may be even more suitable for certain types of materials,

e.g., membrane computing (Păun, 2006) may be more apt for certain chemical materials. While reservoir computing provides an excellent starting point, we should not be predetermined in this choice of model.

RC is a very promising model for material computation, but it only provides a general computational framework. For physical reservoirs, it provides no answer regarding the representation relation, i.e., the choice of input encoding and output decoding. As we have discussed, the properties of physical input signals will often have significant consequences for the computational properties of a material. Although the physical encoding and decoding of signals will inevitably be substrate-dependent, a theoretical treatment of the subject seems pressing.

Of the three materials investigated in this work, ASI is the most promising and therefore been the target of a majority of the work. While the CNT material has many desirable properties in theory, i.e., as a large random network of nonlinear elements, the limited control offered by such unstructured materials makes exploitation difficult in practice. The CC substrate, on the other hand, can be controlled very precisely due to its simple structure. However, its potential is severely limited by the low dimensionality, especially with regards to reservoir computing. ASI seems to offer the best of both worlds: vast parallelism and controllability.

ASIs are high-dimensional, nonlinear dynamical systems governed by local interactions and self-organization. As metamaterials they offer a high degree of control over their internal structure and parameters, which ultimately allows controlling their emergent properties at higher levels. The existence of established physical models and highly developed fabrication techniques make ASIs promising material substrates for future computing devices.

The future of computation in ASI

Within this research project, we have only scratched the surface of this truly fascinating family of metamaterials. There is a lot of future work in store for material computation in ASI.

It is imperative that future work on ASI computing also involves physical realizations of the computing substrate. All the work on ASIs within this research project has been based on software models. Transferring results from models to physical systems requires a lot of practical problems to be solved, in particular regarding input and output. An open question is how susceptible physical ASIs are to noise, and how noise affects their computational properties. Understanding noise in ASI systems is a key topic towards practical realizations.

An important question is how the computational capabilities of ASI systems scales with size. While the ASI systems considered here have been fairly small, ASIs can readily be scaled up to systems with millions of elements. In terms of fabrication, large systems are just as easy to manufacture as are small systems. Simulating large systems will require more computational resources, but the flatspin simulator will easily scale up to a million magnets.

Another key topic is robustness, in terms of device imperfections, component failure and environmental noise. We have hypothesized that large scale patterns, such as the emergent magnetic domains of ASI, should support more robust computation, as the collective behavior will smooth over local imperfections. Future work should investigate whether this is the case, and how the physical properties of the system affect robustness.

Novel ASI geometries provides a seemingly endless playground for the study of emergent computation. Almost all of the established ASI geometries were created as model systems for the study of fundamental physics. What might geometries designed for computation look like? Hybrid geometries, geometries with defects, and more irregular structures are areas that warrant more research. Still, hand-made geometries are the result of the human imagination, with its many inherent biases. An intriguing direction of research are geometries designed by algorithms, which can discover novel geometries with desired computational properties, without the human bias.

Another key topic of research is input encoding. As we have seen, the properties of the input encoding can have a big impact on the dynamics and emergent properties of a material. While the output decoding also plays an important role, it does not directly influence system behavior. For some physical systems, the choice of input encoding may follow fairly naturally, e.g., as part of the driving signal as in the case of our chaotic circuit. For more complex substrates such as ASI, there is a wealth of possibilities when it comes to how to encode input. So far we have considered fairly simple input encoding schemes based on a global magnetic field. Future directions should explore more sophisticated field protocols and the use of local magnetic fields. Here again we are somewhat limited by our human ingenuity. An algorithmic approach to the design of input encodings could be a very interesting pursuit.

Still, there are certain properties of ASIs that are problematic from a computational perspective. The symmetry of magnetic dipole interactions makes controlling the flow of information in magnetic systems challenging. Considering the resulting network of the magnetic interactions, all edges are effectively undirected, which means that information transfer between nodes does not follow immediately from

network structure. Hence, to facilitate information flow in ASI systems, one must rely on emergent dynamic behavior, rather than the structure of the underlying system. Future work should focus on exploring the kind of emergent dynamic behavior that supports information transfer. One potential impact is improving the inherent linear memory, which so far has been quite limited in ASI reservoirs (Hon et al., 2021).

Another feature that is (currently) lacking in ASI systems is adaptation, i.e., mechanisms that allow the underlying structure of the system to be changed dynamically. Intrinsic plasticity is an important feature of adaptive biological systems, which facilitates learning new behavior in response to changes in the environment. Reservoirs with dynamic structure have received relatively little attention, although some promising results have been reported (Schrauwen et al., 2008). Adaptive mechanisms within a material could support similar features, and further extend the computational versatility of the substrate.

Final remarks

Material computation is redefining computing in terms of physical processes and emergence. The world of physics, chemistry and biology provide an abundance of intriguing material systems waiting to be explored. Using physical phenomena for computation can be extremely efficient, encouraging the development of new computational models that naturally exploit them. The endeavor has the potential to result in novel computing devices, whose mode of operation is fundamentally different from anything we have seen so far.

Natural computing provides an endless source of inspiration, demonstrating how computational mechanisms can emerge spontaneously in a range of physical systems. Through a bottom-up exploration of physical substrates, we seek to uncover the fascinating world of computation within. The results may surprise us, challenge our notions of computation, and lead to new directions in computing and beyond.

Bibliography

Aldana, M., Coppersmith, S., and Kadanoff, L. P. Boolean Dynamics with Random Couplings. In Kaplan, E., Marsden, J. E., and Sreenivasan, K. R., editors, *Perspectives and Problems in Nolinear Science: A Celebratory Volume in Honor of Lawrence Sirovich*, pages 23–89. Springer, New York, NY, 2003. ISBN 978-0-387-21789-5. doi: 10.1007/978-0-387-21789-5_2.

Appeltant, L., Soriano, M., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C., and Fischer, I. Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468, 2011. ISSN 2041-1723. doi: 10.1038/ncomms1476.

Arava, H., Leo, N., Schildknecht, D., Cui, J., Vijayakumar, J., Derlet, P. M., Kleibert, A., and Heyderman, L. J. Engineering Relaxation Pathways in Building Blocks of Artificial Spin Ice for Computation. *Physical Review Applied*, 11(5): 054086, May 2019. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.11.054086.

Ashby, W. R. *Design for a Brain: The Origin of Adaptive Behavior*. Chapman & Hall, London, England, 1960. doi: 10.1037/11592-000.

Benioff, P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics*, 22(5):563–591, May 1980. ISSN 1572-9613. doi: 10.1007/BF01011339.

Bertschinger, N. and Natschläger, T. Real-time computation at the edge of chaos in recurrent neural networks. *Neural computation*, 16(7):1413–1436, 2004. ISSN 0899-7667. doi: 10.1162/089976604323057443.

Bhanja, S., Karunaratne, D. K., Panchumarthy, R., Rajaram, S., and Sarkar, S. Non-Boolean computing with nanomagnets for computer vision applications. *Nature Nanotechnology*, 11(2):177–183, 2016. ISSN 17483395. doi: 10.1038/nnano.2015.245.

Bird, J. and Layzell, P. The evolved radio and its implications for modelling the evolution of novel sensors. In *Proceedings of Congress on Evolutionary Computation*, pages 1836–1841, 2002. ISBN 0-7803-7282-4. doi: 10.1109/CEC.2002.1004522.

Boedecker, J., Obst, O., Lizier, J. T., Mayer, N. M., and Asada, M. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012. ISSN 14317613. doi: 10.1007/s12064-011-0146-8.

Boyd, S. and Chua, L. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32 (11):1150–1161, November 1985. ISSN 0098-4094. doi: 10.1109/TCS.1985.1085649.

Brataas, A., Kent, A. D., and Ohno, H. Current-induced torques in magnetic materials. *Nature Materials*, 11(5):372–381, May 2012. ISSN 1476-1122. doi: 10.1038/nmat3311.

Broersma, H., Gomez, F., Miller, J. F., Petty, M., and Tufte, G. Nascence Project: Nanoscale Engineering for Novel Computation Using Evolution. *International Journal of Unconventional Computing*, pages 313–317, 2012.

Budrikis, Z. *Athermal Dynamics of Artificial Spin Ice: Disorder, Edge and Field Protocol Effect*. PhD thesis, The University of Western Australia, 2012.

Büsing, L., Schrauwen, B., and Legenstein, R. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation*, 22(5):1272–1311, 2010. ISSN 0899-7667. doi: 10.1162/neco.2009.01-09-947.

Che, Y., Chen, H., Gui, H., Liu, J., Liu, B., and Zhou, C. Review of carbon nanotube nanoelectronics and macroelectronics. *Semiconductor Science and Technology*, 29(7):073001, May 2014. ISSN 0268-1242. doi: 10.1088/0268-1242/29/7/073001.

Clegg, K. D., Miller, J. F., Massey, K., and Petty, M. Travelling Salesman Problem Solved ' in materio ' by Evolved Carbon Nanotube Device. *Parallel Problem Solving from Nature–PPSN*, XIII:692–701, 2014.

Cook, M. Universality in Elementary Cellular Automata. *Complex Systems*, 15(1): 1–40, 2004.

Coulombe, J. C., York, M. C. A., and Sylvestre, J. Computing with networks of nonlinear mechanical oscillators. *PLOS ONE*, 12(6):e0178663, June 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0178663.

Crutchfield, J. P. The Calculi of Emergence. *Innovation*, 75(April 1993):11–54, 1994. ISSN 01672789. doi: 10.1016/0167-2789(94)90273-9.

Crutchfield, J. P. and Young, K. Computation at the Onset of Chaos. In Zurek, W., editor, *Entropy, Complexity, and the Physics of Information*, volume VIII of *SFI Studies in the Science of Complexity*, pages 223–269. Addison Wesley, Reading, MA, 1990.

Dale, M., Miller, J. F., Stepney, S., and Trefzer, M. A. Evolving Carbon Nanotube Reservoir Computers. In Amos, M. and Condon, A., editors, *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, pages 49–61, Cham, 2016a. Springer International Publishing. ISBN 978-3-319-41312-9. doi: 10.1007/978-3-319-41312-9_5.

Dale, M., Stepney, S., Miller, J. F., and Trefzer, M. Reservoir computing in materio: An evaluation of configuration through evolution. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, December 2016b. ISBN 978-1-5090-4240-1. doi: 10.1109/SSCI.2016.7850170.

Dale, M., Miller, J. F., and Stepney, S. Reservoir Computing as a Model for In-Materio Computing. In Adamatzky, A., editor, *Advances in Unconventional Computing: Volume 1: Theory*, pages 533–571. Springer, Cham, 2017. ISBN 978-3-319-33923-8. doi: 10.1007/978-3-319-33924-5_22.

Dale, M., Miller, J. F., Stepney, S., and Trefzer, M. A. A substrate-independent framework to characterize reservoir computers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2226):20180723, June 2019. doi: 10.1098/rspa.2018.0723.

Dambre, J., Verstraeten, D., Schrauwen, B., and Massar, S. Information Processing Capacity of Dynamical Systems. *Scientific Reports*, 2:514, 2012. ISSN 2045-2322. doi: 10.1038/srep00514.

Dawidek, R. W., Hayward, T. J., Vidamour, I. T., Broomhall, T. J., Venkat, G., Mamoori, M. A., Mullen, A., Kyle, S. J., Fry, P. W., Steinke, N.-J., Cooper, J. F. K., Maccherozzi, F., Dhesi, S. S., Aballe, L., Foerster, M., Prat, J., Vasilaki, E., Ellis, M. O. A., and Allwood, D. A. Dynamically-Driven Emergence in a Nanomagnetic System. *Advanced Functional Materials*, page 2008389, February 2021. ISSN 1616-301X, 1616-3028. doi: 10.1002/adfm.202008389.

Debashis, P., Faria, R., Camsari, K. Y., Appenzeller, J., Datta, S., and Chen, Z. Experimental demonstration of nanomagnet networks as hardware for Ising computing. In *2016 IEEE International Electron Devices Meeting (IEDM)*,

pages 34.3.1–34.3.4. IEEE, December 2016. ISBN 978-1-5090-3902-9. doi: 10.1109/IEDM.2016.7838539.

Dennard, R., Gaensslen, F., Yu, H.-N., Rideout, V., Bassous, E., and LeBlanc, A. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, October 1974. ISSN 1558-173X. doi: 10.1109/JSSC.1974.1050511.

Ditto, W. L., Murali, K., and Sinha, S. Chaos computing: Ideas and implementations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1865):653–664, February 2008. ISSN 1364-503X. doi: 10.1098/rsta.2007.2116.

Ebrahimi, K., Jones, G. F., and Fleischer, A. S. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. *Renewable and Sustainable Energy Reviews*, 31:622–638, March 2014. ISSN 1364-0321. doi: 10.1016/j.rser.2013.12.007.

Esmaeilzadeh, H., Blem, E., Amant, R. S., Sankaralingam, K., and Burger, D. Dark silicon and the end of multicore scaling. In *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, pages 365–376, June 2011.

Fernando, C. and Sojakka, S. Pattern Recognition in a Bucket. *Advances in Artificial Life*, 2801(12):588–597, 2003. ISSN 03029743. doi: 10.1007/978-3-540-39432-7_63.

Fodor, J. A. The Mind-Body Problem. *Scientific American*, 244(1):114–123, 1981. ISSN 0036-8733.

Gallicchio, C., Lukoševičius, M., and Scardapane, S. Frontiers in Reservoir Computing. In *ESANN 2020 - Proceedings*, October 2020.

Gartside, J. C., Arroo, D. M., Burn, D. M., Bemmer, V. L., Moskalenko, A., Cohen, L. F., and Branford, W. R. Realization of ground state in artificial kagome spin ice via topological defect-driven magnetic writing. *Nature Nanotechnology*, 13 (1):53–58, January 2018. ISSN 1748-3387. doi: 10.1038/s41565-017-0002-1.

Gypens, P., Leliaert, J., and Van Waeyenberge, B. Balanced Magnetic Logic Gates in a Kagome Spin Ice. *Physical Review Applied*, 9(3):034004, March 2018. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.9.034004.

Hon, K., Kuwabiraki, Y., Goto, M., Nakatani, R., Suzuki, Y., and Nomura, H. Numerical simulation of artificial spin ice for reservoir computing. *Applied Physics Express*, 14(3):033001, March 2021. ISSN 1882-0778, 1882-0786. doi: 10.35848/1882-0786/abdcd8.

Hopfield, J. J. Neurons, dynamics and computation. *Physics Today*, 47(2):40–47, 1994.

Hornby, G., Globus, A., Linden, D., and Lohn, J. Automated Antenna Design with Evolutionary Algorithms. In *Space 2006*, San Jose, California, September 2006. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-049-9. doi: 10.2514/6.2006-7242.

Horsman, C., Stepney, S., Wagner, R. C., and Kendon, V. When does a physical system compute? *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(2169):20140182–20140182, 2014. ISSN 1364-5021. doi: 10.1098/rspa.2014.0182.

Horsman, D., Kendon, V., Stepney, S., and Young, J. P. W. Abstraction and Representation in Living Organisms: When Does a Biological System Compute? In Dodig-Crnkovic, G. and Giovagnoli, R., editors, *Representation and Reality in Humans, Other Living Organisms and Intelligent Machines*, volume 28, pages 91–116. Springer International Publishing, Cham, 2017. ISBN 978-3-319-43782-8 978-3-319-43784-2. doi: 10.1007/978-3-319-43784-2_6.

Imre, A. Majority Logic Gate for Magnetic Quantum-Dot Cellular Automata. *Science*, 311(5758):205–208, January 2006. ISSN 0036-8075. doi: 10.1126/science.1120506.

Inubushi, M. and Yoshimura, K. Reservoir Computing Beyond Memory-Nonlinearity Trade-off. *Scientific Reports*, 7(1):10199, December 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-10257-6.

Jaeger, H. Short term memory in echo state networks. Technical Report 152, German National Research Center for Information Technology, 2002.

Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. Technical Report 148, German National Research Center for Information Technology, 2001.

Jensen, J. H. and Tufte, G. Reservoir computing with a chaotic circuit. In *ECAL 2017: The Fourteenth European Conference on Artificial Life*, pages 222–229, Cambridge, MA, September 2017. MIT Press. ISBN 978-0-262-34633-7. doi: 10.7551/ecal_a_039.

Jensen, J. H. and Tufte, G. Reservoir Computing in Artificial Spin Ice. In *ALIFE 2020: The 2020 Conference on Artificial Life*, pages 376–383, Online, 2020. MIT Press. doi: 10.1162/isal_a_00268.

Jensen, J. H., Folven, E., and Tufte, G. Computation in artificial spin ice. In *ALIFE 2018: The 2018 Conference on Artificial Life*, pages 15–22, Tokyo, Japan, 2018. MIT Press. doi: 10.1162/isal_a_00011.

Jensen, J. H., Strømberg, A., Lykkebø, O. R., Penty, A., Själander, M., Folven, E., and Tufte, G. Flatspin: A Large-Scale Artificial Spin Ice Simulator. *arXiv:2002.11401 [cond-mat, physics:physics]*, February 2020.

Jones, B., Stekel, D., Rowe, J., and Fernando, C. Is there a Liquid State Machine in the Bacterium Escherichia Coli? *2007 IEEE Symposium on Artificial Life*, pages 187–191, 2007. doi: 10.1109/ALIFE.2007.367795.

Kauffman, S. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, 1969. ISSN 00225193. doi: 10.1016/0022-5193(69)90015-0.

Lambson, B., Carlton, D., and Bokor, J. Exploring the Thermodynamic Limits of Computation in Integrated Systems: Magnetic Memory, Nanomagnetic Logic, and the Landauer Limit. *Physical Review Letters*, 107(1):010604, July 2011. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.107.010604.

Langton, C. G. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, 1990. ISSN 01672789. doi: 10.1016/0167-2789(90)90064-V.

Le Guyader, L., Savoini, M., El Moussaoui, S., Buzzi, M., Tsukamoto, A., Itoh, A., Kirilyuk, A., Rasing, T., Kimel, A. V., and Nolting, F. Nanoscale sub-100 picosecond all-optical magnetization switching in GdFeCo microstructures. *Nature Communications*, 6(1):5839, December 2015. ISSN 2041-1723. doi: 10.1038/ncomms6839.

Legenstein, R. and Maass, W. What makes a dynamical system computationally powerful? In *New Directions in Statistical Signal Processing From Systems to Brain*, pages 127–154. MIT Press, 2005. ISBN 978-0-262-25631-5.

Legenstein, R. and Maass, W. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3):323–334, April 2007. ISSN 0893-6080. doi: 10.1016/j.neunet.2007.04.017.

Leliaert, J., Dvornik, M., Mulkers, J., De Clercq, J., Milošević, M. V., and Van Waeyenberge, B. Fast micromagnetic simulations on GPU - Recent advances made with mumax3. *Journal of Physics D: Applied Physics*, 51(12), 2018. ISSN 13616463. doi: 10.1088/1361-6463/aaab1c.

Levis, D., Cugliandolo, L. F., Foini, L., and Tarzia, M. Thermal Phase Transitions in Artificial Spin Ice. *Physical Review Letters*, 110(20):207206, May 2013. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.110.207206.

Li, Y., Paterson, G. W., Macauley, G. M., Nascimento, F. S., Ferguson, C., Morley, S. A., Rosamond, M. C., Linfield, E. H., MacLaren, D. A., Macêdo, R., Marrows, C. H., McVitie, S., and Stamps, R. L. Superferromagnetism and Domain-Wall Topologies in Artificial "Pinwheel" Spin Ice. *ACS Nano*, 13(2):2213–2222, January 2019. ISSN 1936-0851. doi: 10.1021/acsnano.8b08884.

Lukoševičius, M. and Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. ISSN 15740137. doi: 10.1016/j.cosrev.2009.03.005.

Lykkebø, O. R. and Tufte, G. Comparison and evaluation of signal representations for a carbon nanotube computational device. In *2014 IEEE International Conference on Evolvable Systems*, pages 54–60, 2014. ISBN 978-1-4799-4479-8. doi: 10.1109/ICES.2014.7008722.

Lykkebø, O. R., Nichele, S., and Tufte, G. An Investigation of Square Waves for Evolution in Carbon Nanotubes Material. In *Proceedings of the European Conference on Artificial Life 2015*, pages 503–510, 2015.

Lykkebø, O. R., Jensen, J. H., Penty, A., Strømberg, A., Själander, M., Folven, E., and Tufte, G. Emergent computation on a magnetic lattice. In *International Workshop on Theoretical and Experimental Material Computing*, Tokyo, Japan, June 2019.

Maass, W., Natschläger, T., and Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002. ISSN 0899-7667. doi: 10.1162/089976602760407955.

Maass, W., Legenstein, R. A., and Bertschinger, N. Methods for Estimating the Computational Power and Generalizatoin Capability of Neural Microcircuits. In *Advances in Neural Information Processing Systems 17*, pages 865–872. MIT Press, 2004.

Mack, C. A. Fifty years of Moore's law. *IEEE Transactions on Semiconductor Manufacturing*, 24(2):202–207, 2011. ISSN 08946507. doi: 10.1109/TSM.2010.2096437.

Massey, M. K., Kotsialos, A., Qaiser, F., Zeze, D. A., Pearson, C., Volpati, D., Bowen, L., and Petty, M. C. Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903, April 2015. ISSN 0021-8979. doi: 10.1063/1.4915343.

Mead, C. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10): 1629–1636, 1990. ISSN 00189219. doi: 10.1109/5.58356.

Mengotti, E., Heyderman, L. J., Rodríguez, A. F., Nolting, F., Hügli, R. V., and Braun, H. B. Real-space observation of emergent magnetic monopoles and associated Dirac strings in artificial kagome spin ice. *Nature Physics*, 7(1):68–74, 2011. ISSN 17452473. doi: 10.1038/nphys1794.

Miller, J. and Downing, K. Evolution in materio: Looking beyond the silicon box. In *Proceedings 2002 NASA/DoD Conference on Evolvable Hardware*, pages 167–176, 2002. ISBN 0-7695-1718-8. doi: 10.1109/EH.2002.1029882.

Miller, J. F., Harding, S. L., and Tufte, G. Evolution-in-materio: Evolving computation in materials. *Evolutionary Intelligence*, 7(1):49–67, April 2014. ISSN 1864-5909. doi: 10.1007/s12065-014-0106-6.

Moore, G. E. Cramming more components onto integrated circuits. *Electronicsweek*, 38(8), 1965.

Munakata, T., Sinha, S., and Ditto, W. Chaos computing: Implementation of fundamental logical gates by chaotic elements. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(11):1629–1633, November 2002. ISSN 1057-7122. doi: 10.1109/TCSI.2002.804551.

Munakata, T., Takahashi, J., Sekikawa, M., and Aihara, K. Chaos computing: A unified view. *International Journal of Parallel, Emergent and Distributed Systems*, 25(1):3–16, February 2010. ISSN 1744-5760. doi: 10.1080/17445760902774898.

Murali, K., Lakshmanan, M., and Chua, L. The simplest dissipative nonautonomous chaotic circuit. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(6):462–463, June 1994a. ISSN 10577122. doi: 10.1109/81.295246.

Murali, K., Lakshmanan, M., and Chua, L. BIFURCATION AND CHAOS IN THE SIMPLEST DISSIPATIVE NON-AUTONOMOUS CIRCUIT. *International Journal of Bifurcation and Chaos*, 04(06):1511–1524, December 1994b. ISSN 0218-1274. doi: 10.1142/S0218127494001179.

Murali, K., Sinha, S., and Ditto, W. L. Construction of a reconfigurable dynamic logic cell. *Pramana*, 64(3):433–441, March 2005a. ISSN 0973-7111. doi: 10.1007/BF02704569.

Murali, K., Sinha, S., and Mohamed, I. R. Chaos computing: Experimental realization of NOR gate using a simple chaotic circuit. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 339(1-2):39–44, 2005b. ISSN 03759601. doi: 10.1016/j.physleta.2005.02.044.

Nakajima, K. Muscular-Hydrostat Computers: Physical Reservoir Computing for Octopus-Inspired Soft Robots. In Shigeno, S., Murakami, Y., and Nomura, T., editors, *Brain Evolution by Design: From Neural Origin to Cognitive Architecture*, Diversity and Commonality in Animals, pages 403–414. Springer Japan, Tokyo, 2017. ISBN 978-4-431-56469-0. doi: 10.1007/978-4-431-56469-0_18.

Nichele, S., Laketić, D., Lykkebø, O. R., and Tufte, G. Is There Chaos in Blobs of Carbon Nanotubes Used to Perform Computation? In *FUTURE COMPUTING 2015, The Seventh International Conference on Future Computational Technologies and Applications*, pages 12–17, March 2015. ISBN 978-1-61208-389-6.

Nichele, S., Jensen, J. H., Laketic, D., Lykkebø, O. R. S., and Tufte, G. Dynamics in Carbon Nanotubes for In-Materio Computation. *International Journal On Advances in Systems and Measurements*, 9(1):24–37, 2016.

Niemier, M. T., Bernstein, G. H., Csaba, G., Dingler, A., Hu, X. S., Kurtz, S., Liu, S., Nahas, J., Porod, W., Siddiq, M., and Varga, E. Nanomagnet logic: Progress toward system-level integration. *Journal of Physics: Condensed Matter*, 23(49): 493202, December 2011. ISSN 0953-8984. doi: 10.1088/0953-8984/23/49/493202.

Nikolić, D., Häusler, S., Singer, W., Maass, W., Nikoli, D., and Haeusler, S. Temporal dynamics of information content carried by neurons in the primary visual cortex. *Theoretical Computer Science*, 20:1041–1048, 2007.

Nomura, H., Furuta, T., Tsujimoto, K., Kuwabiraki, Y., Peper, F., Tamura, E., Miwa, S., Goto, M., Nakatani, R., and Suzuki, Y. Reservoir computing with dipole-coupled nanomagnets. *Japanese Journal of Applied Physics*, 58(7):070901, July 2019. ISSN 0021-4922, 1347-4065. doi: 10.7567/1347-4065/ab2406.

Ozturk, M. C. and Principe, J. C. Computing with transiently stable states. *Proceedings of the International Joint Conference on Neural Networks*, 3:1467–1472, 2005. doi: 10.1109/IJCNN.2005.1556092.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., and Massar, S. Optoelectronic Reservoir Computing. *Scientific Reports*, 2:1–6, 2012. ISSN 2045-2322. doi: 10.1038/srep00287.

Pask, G. Physical Analogues to the Growth of a Concept. In *Mechanization of Thought Processes, Symposium*, pages 877–922, 1959.

Păun, G. Introduction to Membrane Computing. In Ciobanu, G., Păun, G., and Pérez-Jiménez, M. J., editors, *Applications of Membrane Computing*, Natural Computing Series, pages 1–42. Springer, Berlin, Heidelberg, 2006. ISBN 978-3-540-29937-0. doi: 10.1007/3-540-29937-8_1.

Piccinini, G. Computing Mechanisms. *Philosophy of Science*, 74(4):501–526, October 2007. ISSN 0031-8248. doi: 10.1086/522851.

Piccinini, G. and Maley, C. Computation in Physical Systems. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2021 edition, 2021.

Pickering, A. *The Cybernetic Brain: Sketches of Another Future*. University of Chicago Press, USA, 2011. ISBN 0-226-66790-1.

Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, January 1999. ISSN 0962-4929, 1474-0508. doi: 10.1017/S0962492900002919.

Pinna, D., Bourianoff, G., and Everschor-Sitte, K. Reservoir Computing with Random Skyrmion Textures. *Physical Review Applied*, 14(5):054020, November 2020. doi: 10.1103/PhysRevApplied.14.054020.

Salehi Fashami, M., Roy, K., Atulasimha, J., and Bandyopadhyay, S. Magnetization dynamics, Bennett clocking and associated energy dissipation in multiferroic logic. *Nanotechnology*, 22(30):309501, July 2011. ISSN 0957-4484, 1361-6528. doi: 10.1088/0957-4484/22/30/309501.

Sanz-Hernández, D., Massouras, M., Reyren, N., Rougemaille, N., Schánilec, V., Bouzehouane, K., Hehn, M., Canals, B., Querlioz, D., Grollier, J., Montaigne, F., and Lacour, D. Tunable stochasticity in an artificial spin network. *Advanced Materials*, 33(17):2008135, 2021. doi: 10.1002/adma.202008135.

Sarpeshkar, R. Analog Versus Digital: Extrapolating from Electronics to Neurobiology. *Neural Computation*, 10(7):1601–1638, October 1998. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976698300017052.

Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J. J., and Stroobandt, D. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7-9):1159–1171, 2008. ISSN 09252312. doi: 10.1016/j.neucom.2007.12.020.

Schurmann, F., Meier, K., and Schemmel, J. Edge of Chaos Computation in Mixed-Mode VLSI - " A Hard Liquid ". In *Advances in Neural Information Processing Systems*, pages 1201–1208. 2004. ISBN 0-262-19534-8. doi: 10.1.1.85.2064.

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., and Gimzewski, J. K. A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, 24 (38):384004, 2013. ISSN 0957-4484. doi: 10.1088/0957-4484/24/38/384004.

Sinha, S. and Ditto, W. Dynamics Based Computation. *Physical Review Letters*, 81 (1):2156–2159, 1998. ISSN 0031-9007. doi: 10.1103/PhysRevLett.81.2156.

Sinha, S., Munakata, T., and Ditto, W. L. Parallel computing with extended dynamical systems. *Physical Review E*, 65(3):036214, February 2002. doi: 10.1103/PhysRevE.65.036214.

Skjærvø, S. H., Marrows, C. H., Stamps, R. L., and Heyderman, L. J. Advances in artificial spin ice. *Nature Reviews Physics*, 2(1):13–28, January 2020. ISSN 2522-5820. doi: 10.1038/s42254-019-0118-3.

Sklenar, J., Lao, Y., Albrecht, A., Watts, J. D., Nisoli, C., Chern, G.-W., and Schiffer, P. Field-induced phase coexistence in an artificial spin ice. *Nature Physics*, 15(2):191–195, February 2019. ISSN 1745-2473, 1745-2481. doi: 10.1038/s41567-018-0348-9.

Stepney, S. The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164, 2008. ISSN 01672789. doi: 10.1016/j.physd.2008.01.028.

Stepney, S. Nonclassical Computation — A Dynamical Systems Perspective. In Rozenberg, G., Bäck, T., and Kok, J. N., editors, *Handbook of Natural Computing*, pages 1979–2025. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-540-92909-3 978-3-540-92910-9. doi: 10.1007/978-3-540-92910-9_59.

Stepney, S. and Kendon, V. The Role of the Representational Entity in Physical Computing. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11493 LNCS, pages 219–231. Springer International Publishing, 2019. ISBN 978-3-030-19310-2. doi: 10.1007/978-3-030-19311-9_18.

Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, second edition, 2015. ISBN 978-0-8133-4910-7.

Strømberg, A., Penty, A., Jensen, J. H., Lykkebø, O. R., Själander, M., Tufte, G., and Folven, E. Unconventional Computing with Artificial Spin Ice. In *Frontiers in Artificial Spin Ice*, Bad Zurzach, 2019.

Sutton, B., Camsari, K. Y., Behin-Aein, B., and Datta, S. Intrinsic optimization using stochastic nanomagnets. *Scientific Reports*, 7:1–9, 2017. ISSN 20452322. doi: 10.1038/srep44370.

Szangolies, J. The Abstraction/Representation Account of Computation and Subjective Experience. *Minds and Machines*, March 2020. ISSN 0924-6495, 1572-8641. doi: 10.1007/s11023-020-09522-x.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, July 2019. ISSN 08936080. doi: 10.1016/j.neunet.2019.03.005.

Thompson, A., Layzell, P., and Zebulum, R. S. Explorations in design space: Unconventional electronics design through artificial evolution. *IEEE Transactions on Evolutionary Computation*, 3(3):167–195, 1999. ISSN 1089778X. doi: 10.1109/4235.788489.

Toffoli, T. Physics and computation. *International Journal of Theoretical Physics*, 21(3):165–175, April 1982. ISSN 1572-9575. doi: 10.1007/BF01857724.

Turing, A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937. ISSN 1460-244X. doi: 10.1112/plms/s2-42.1.230.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., Schrauwen, B., Dambre, J., and Bienstman, P. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5:3541, 2014. ISSN 2041-1723. doi: 10.1038/ncomms4541.

Volder, M. F. L. D., Tawfick, S. H., Baughman, R. H., and Hart, A. J. Carbon Nanotubes: Present and Future Commercial Applications. *Science*, 339(6119): 535–539, February 2013. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1222453.

von Neumann, J. The general and logical theory of automata. In *Cerebral Mechanisms in Behavior: The Hixon Symposium*. John Wiley & Sons, New York, 1951.

Wang, R. F., Nisoli, C., Freitas, R. S., Li, J., McConville, W., Cooley, B. J., Lund, M. S., Samarth, N., Leighton, C., Crespi, V. H., and Schiffer, P. Artificial 'spin ice' in a geometrically frustrated lattice of nanoscale ferromagnetic islands. *Nature*, 439(7074):303–306, 2006. ISSN 14764687. doi: 10.1038/nature04447.

Wiener, N. *Cybernetics: Or Control and Communication in the Animal and the Machine, Second Edition*. MIT Press, Cambridge, 1961. doi: 10.1037/13140-000.

Wolfram, S. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, January 1984. ISSN 0167-2789. doi: 10.1016/0167-2789(84)90245-8.

Wolfram, S. *A New Kind of Science*. Wolfram Media Inc., Champaign, Ilinois, USA, 2002. ISBN 1-57955-008-8.

Part II

Publications

Paper A

Dynamics in Carbon Nanotubes for In-Materio Computation

(Nichele et al., 2016)

Author(s):

Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketić, Odd Rune Lykkebø and Gunnar Tufte

Published in journal:

International Journal on Advances in Systems and Measurements, vol 9 no 1 & 2, year 2016

Paper A
Nichele et al. (2016)

Dynamics in Carbon Nanotubes for In-Materio Computation

Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketić, Odd Rune Lykkebø and Gunnar Tufte

Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
Email: {nichele, johannj, draganal, lykkebo, gunnart}@idi.ntnu.no

Abstract

In-materio computation exploits physical properties of materials as substrates for computation. Evolution-In-Materio (EIM) uses evolutionary search algorithms to find such configurations of the material for which material physics yields desired computation. New unconventional materials have been recently investigated as potential computational mediums. Such materials may intrinsically possess rich physical properties, which may allow a wide variety of dynamics. However, how to access such properties and exploit them to carry out a wanted computation is still an open question. This article explores the dynamics in one particular type of nanomaterials which is used to solve computational tasks. Nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and PolyButyl MethAcrylate (PBMA) are configured so as to undergo evolutionary processes with the goal of performing certain computational tasks. Early experiments showed that rich dynamics may be achieved, which may yield complex computations. Some indications of chaotic behavior were observed so further work was carried out with the aim of examining the dynamics achievable by such nanocomposites. Since it is not an easy task to access the physics at the very bottom of the material, investigation of the material dynamics is kept within the limits imposed by our measurement equipment and the level of observability enabled by it. Presented results show that interesting, complex dynamics is achievable by examined nanocomposites and that it depends on the type of signals used for the material configuration

as well as on the material intrinsic properties such as percentage of SWCNTs in the nanocomposite.

1 Introduction

Computations result from perturbations of some dynamical system. The observable output of the system is the result of its dynamics. Dependent on the type of dynamics exhibited by the system, computations of various complexity levels may be achieved. The type of dynamics depends on the physics of the system and on the way in which the system is manipulated. Our work considers novel nanoscale materials [1] and was carried out within the EU-funded NASCENCE (NAnoSCale Engineering for Novel Computation using Evolution) project [2]. The nanomaterials investigated within the project are nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and polymer molecules (PBMA), and networks of gold nanoparticles. The investigation of nanocomposites is performed under the Evolution-In-Materio (EIM) scenario [3, 4].

EIM is a novel approach to designing computing devices where various materials are used as computational substrates. It is one approach that may emerge as an answer to the challenges of today's widely accepted semiconductor technology. Digital computers based on silicon technology are designed using a conventional top-down process by human engineers. Engineering of such processors poses technological challenges due to scaling down. Various design techniques are applied in order to sustain scaling down of the semiconductor technology but it is becoming increasingly difficult to fabricate transistors at the nanoscale.

This has motivated efforts towards novel technologies that will assume not only new computational substrates but also novel principles of the design of computing devices and their usage. EIM is a bottom-up approach in which the physics of a computing substrate is used to produce computations of interest. Different computational substrates have been previously explored such as liquid crystals and Field Programmable Gate Arrays (FPGAs) [5–7]. The configuration of the computing substrate, i.e., some material, undergoes evolutionary changes until some desired response of the material is achieved according to the computational task at hand. The digital computer accesses the material via a special board, which allows the Evolutionary Algorithm

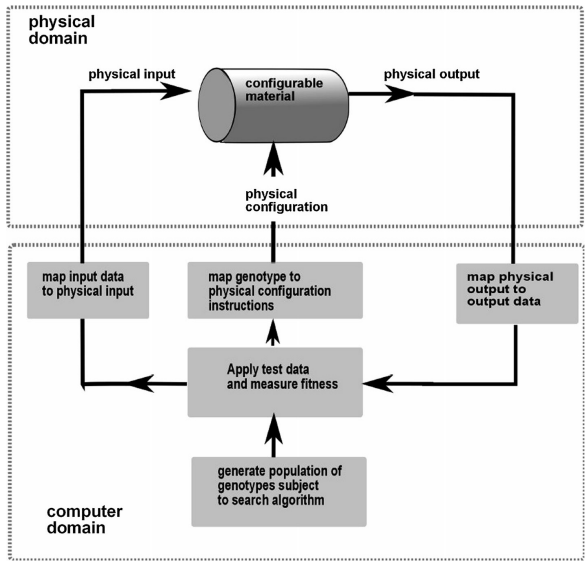


Figure 1. Principle of EIM illustrating the separation of an analog/physical domain where the material operates and the digital domain of computers, from [3].

(EA) to apply configuration and input signals and read the material response which will guide the evolutionary search.

Figure 1 illustrates an EIM system. Three main entities can be distinguished: a digital computer, the material and the interface between the two. The system clearly shows the separation of an analog/physical domain in which materials operate and a digital domain in which the computer responsible for input/output mapping and configuration operates. In all such systems an interface is needed for bidirectional translation of signals between digital signals of the computer and analog signals in the physical domain of the material. As mentioned, the digital computer is used for running the EA, which generates a population of genomes, and translates each genome into suitable analog signals which can be sent to the interface board.

Further, the response of the material for a given configuration and input signals is translated from analog form as produced by the material to its corresponding digital value so that the computer can calculate the fitness value of the genome. The fitness value guides evolutionary search towards a solution to the problem at hand.

In order to produce interesting behavior under the EIM scenario, it is required that the material is able to exhibit rich dynamics. The richness of

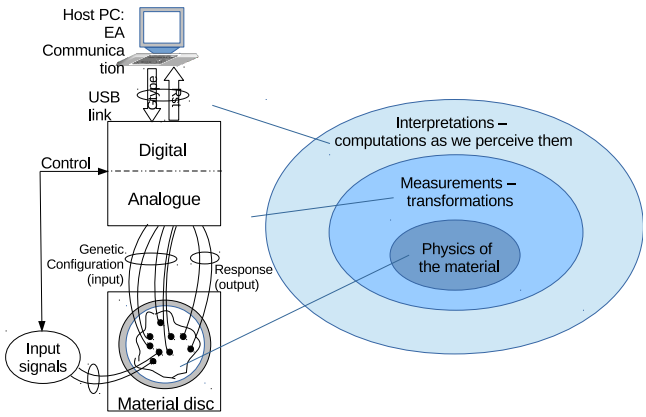


Figure 2. Conceptual domains of the computing system.

the exhibited dynamics can be attributed to the physical properties of the material. In a way it can be said that EIM manipulates the material so as to produce rich dynamics. The material blob is treated as a black box and EAs are used to "program" the material to solve a problem at hand.

Such a black box hybrid approach has been shown successful for a number of computational problems [8–13]. At the current state of research, it is not clearly understood what the exploited physical properties are and what the best way of exploring them is, e.g., what number of inputs and outputs and which types of signals - electrical (static voltages, sinusoidal waves, square waves) or even of some other kind such as temperature or light. The solved problems serve as a proof of concept that an EIM approach may be used for solving computational problems and indicates that it may be competitive in terms of computational time, size, and energy consumption. However, scaling-up to solve larger instances of a problem requires a better understanding of the dynamics exhibited by the material. In other words, the black box needs to be opened so that the underlying physical properties of the material are well understood. The number of used input electrodes, configuration signals available, etc. will directly affect the evolutionary search space.

Observing dynamics and its emergent complexity in computational materials is not an easy pursuit. Observability is limited by what output can be measured from the material and at which scale. At some scales we are

not able to directly observe physical effects present in the materials, e.g., quantum effects due to mechanisms of electron transmission through carbon nanotubes. Therefore, we are limited to use signals which can be observed and measured. Figure 2 illustrates the taken approach to observe, exploit and gain an understanding of the dynamics of EIM systems. At the lowest level we have the physics of the material where computations happen, but due to nanoscale and even quantum effects, what is captured by our instruments will at best be just an approximation. In other words, the lowest level is inaccessible and must be treated as a black box. At the second level, the level of measurements and transformations, physical properties and dynamics are observable in the analog domain. This level can be explored to gain insight into the electrical properties of the material. The top level is the level of interpretations, i.e., computations as we perceive them. So, as shown in Figure 2, the dynamics of the analog signals are interpreted and transferred to data, i.e., the computational input – output mapping is performed. The top level is the level which is explored for computation. Here, it is important to note that the observations on the top level emerge as a result of all underlying dynamics.

The work presented in this paper includes a specific approach, as illustrated in Figure 2, to investigate the dynamics of the material at hand. The approach considers the complexity of the input - output mapping performed by the material for computation. Complexity is hard to measure even when well defined as, for example, Kolmogorov complexity [14]. Some approximations are needed if we want to obtain quantitative measures. In this work, we adopt *compressibility* as a measure of complexity.

This paper, which is an extended version of [1], is organized as follows: Section 2 provides background on EIM and position of the NASCENCE project within the field. Section 3 presents experimental platform Mecobo, which was developed within NASCENCE project, and which is used in our EIM experiments. Also, an experimental setup is explained as well as the material which was used in the experiments. Moreover, the section provides some background on different computational domains which can be distinguished under EIM computing scenario. Further, Section 4 provides some initial results, presented in [1], which demonstrate interesting behaviors of the investigated material. Section 5 presents experiments which were conducted with the aim of investigating material dynamics in a greater detail. A measure of complexity is introduced which is used as the description of material behavior, the three sets of experiments are described followed by the results and the discussion which relates results to theoretical foundations. Finally,

Section 6 provides conclusion about the presented experiments and exhibited material dynamics within EIM computing.

2 Background – Evolution-in-Materio

The term Evolution-in-Materio was introduced by Julian Miller and Keith Downing in 2002 [4]. The general concept of EIM is that physical systems may intrinsically possess properties which may be exploited for computation.

2.1 Pioneering work

Early work on manipulation of physical systems for computation is related to the work of Gordon Pask [15], a classical cyberneticist whose pioneering work dates back to the 1950s. He tried to grow neural structures, dendritic wires, in a metal-salt solution by electrical stimulation [7]. His goal was to self-assemble a wiring structure within the material in order to carry out some sort of signal processing embedded in the material. He was able to alter the position and structure of the wiring filaments, and thus the behavior of the system. This was achieved by external influence, which consisted in applying different currents on electrodes in the metal-salt solution. This early version of material manipulation was done without aid of computers and different electrical configurations were tested manually. Stewart [16] later defined such a process as manufacturing logic "by the pound, using techniques more like those of a bakery than of an electronics factory".

2.2 Analog computers, FPGAs and liquid crystal

Later, Mills constructed an analog computer which he called Kirchhoff-Lukasiewicz Machine (KLM) [17]. The construction was done by connecting a conductive polymer material to logical units. The analog computation was carried out by placing current sources and current sinks into the conductive foam layer and reading the output from the logical units. One could argue that such machines were not easy to program due to the manual placement of connections into the material. On the other hand, some advantages of performing computation directly in the material substrate became obvious, e.g.,

a large number of partial differential equations were solved within nanoseconds.

In 1996, Thompson used intrinsic evolution to produce electrical circuits in FPGAs [5]. In his well-known experiment, he demonstrated that artificial evolution can be used to exploit physical properties of FPGAs to build working circuits, e.g., a frequency discriminator circuit. He found out that placing the circuit in a different part of the chip or disconnecting some unused modules would result in a non-working solution. Moreover, he was unable to replicate the chip behavior in simulation because evolution had exploited underlying physical properties of the FPGA. In fact, changing the FPGA with a similar model from the same producer would result in slightly different behavior. Thompson described such a process as "removing the digital design and letting evolution do it".

In [4], Miller and Downing suggested several materials which may be suitable for EIM, liquid crystals being among them. Simon Harding [18] later demonstrated that it was indeed possible to apply EIM on liquid crystals to evolve several computational devices: a tone discriminator [19], logic gates [20], and robot controllers [6]. Liquid crystal is a movable material where voltages affect orientation of the crystals. The movability was problematic since the material would undergo permanent changes during evolution. This led to unstable solutions that worked only once. Nevertheless, he showed that it was possible to quickly reach a working solution again by re-running the evolutionary algorithm for a couple of generations [19].

2.3 The NASCENCE project and recent work

Recently, the NASCENCE project [2] addressed nanomaterials and nanoparticles for EIM with the long term goal to build information processing devices exploiting such materials without the need to reproduce individual components. In particular, investigated nanomaterials included nanocomposites made of SWCNTs and polymer molecules and nanoparticle networks, in particular gold coated nanoparticles. Several hard-to-solve computational problems have been solved as proof of concept, e.g., Traveling Salesman [8], logic gates [9], bin packing [10], machine learning classification [11], frequency classification [12], function optimization [13] and robot controllers [21]. The SWCNT materials from the project are the subject of our investigation in this paper.

2.4 Interpretation and computation

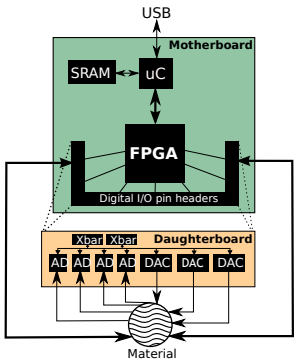
As stated, EIM has been used to solve a variety of problems. However, these results are all limited to a specific problem domain. To assess the potential computational power available in a material, we need a more general measurement. One way is to view *complexity* as indication of potential computational power [22].

Kolmogorov complexity [14, 23] is well-defined but incomputable in theory. However, it is possible to use measures such as compressibility to approximate complexity to some extent [24–27]. In fact, strings that are hardly compressible have a presumably high Kolmogorov complexity. Complexity is then proportional to the compression ratio.

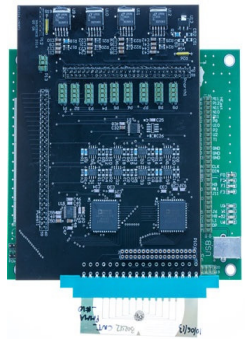
High measurable complexity of output data or a high complexity ratio between input and output data may not always be a desired property. In classifier systems, such as Thompson's frequency discriminator [5], the output may be a binary response to a complex input signal. In this case the complexity ratio between output and input is very low. However, the computation, i.e., internal state transitions in the underlying physics of the material, is still a complex process but the complexity is unobservable since we only observe the input and output signals.

3 A platform for experiments and understanding of EIM systems

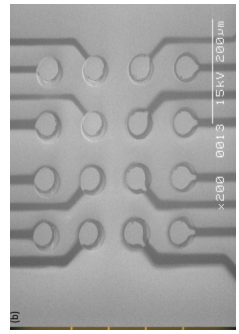
The conceptual idea of exploiting physics for computation requires a physical device, i.e., the material. In most EIM works, an intrinsic approach has been taken – computation is a result of real physical processes and the evaluation is a result of the performance of a physical system. An intrinsic approach allows access to all inherent physical properties of the material [3]. An analog computation [28] is a possibility, however, in this work a hybrid approach is taken. The hybrid approach includes the computational matter in a mixed signal system using a digital computer to configure and communicate with the material. Such an approach enables the computational power of the material with the ease of programmability of digital computers [2]. In a hybrid approach, observability is an issue, i.e., ensuring that the data from the material is observable and sound without using more computational power for the observation than the actual computation [29].



(a) Block diagram of the Mecobo hardware interface.



(b) Picture of the Mecobo motherboard with mixed signal daughter board.



(c) Electrode array, microscopic view. (Source: documentation of the NASCENCE project).

Figure 3. Overview of the Mecobo hardware interface.

3.1 NASCENCE's Mecobo: an experimental platform for EIM

A hybrid approach requires an interface between the digital world of computers and the analog world of materials. The Mecobo experimental platform [30] from the NASCENCE project is a hardware/software platform implementing the conceptual Evo-Materio-system shown in Figure 1.

Figure 3 shows an overview of the hardware interface: a Mecobo platform and microelectrode array on the material slide. A block diagram of the Mecobo platform is shown in Figure 3a. Configuration specification, i.e., genotypes, are loaded from a PC to Mecobo over a USB port. A microcontroller communicates with the USB interface and with an FPGA via an internal bus. The FPGA can be interfaced to the materials directly or, as shown in the figure, use a daughter board to extend the range of possible signals.

A picture of the Mecobo hardware is presented in Figure 3b. In the picture, the Mecobo is shown with a mixed signal daughter board and a material sample on a glass slide plugged in. Electrical connection between the material on the slide and the board is realized by the microelectrode array. A microscopic view of the microelectrode array before material disposition is shown in Figure 3c.

Mecobo is capable of controlling close to 100 individually configurable input/output signals (pins), which can be connected to the material. Each signal is described by parameters at a given point in time. For example, a pin can be programmed as a recording pin from time 0 to $100ms$, or as an output pin of square waves of some frequency from 0 to $1000ms$, or as an output pin of a constant voltage level, e.g., $2.7V$ from time 0 to $1500ms$ etc. Mecobo is connected to a host PC over USB and communicates via a Thrift server [31]. Communication based on Thrift technology also enables access to Mecobo remotely over the Internet. The maximum analog sampling frequency of the Mecobo board is $500kHz$. Input signals may be static voltages or periodic (e.g., square, sinusoidal) waves ranging in frequency between $400Hz$ and $25MHz$. For more details on Mecobo and an overview of the full range of programmable properties of the platform, see [30].

3.2 Explaining computations within EIM

It can be said that computations are based on transformations of a system, so that the system input(s) and output(s) are related in some functional way. This functional relation can be expressed by a simple formula:

$$y = F(x) \tag{1}$$

where x and y correspond to an input and output of the system, respectively, and, in general, they are considered to be multidimensional and represented by vectors.

One way of analysis, more formally addressed within the system theory [32, 33] assumes that the system state is described by a set of variables that move through a state space.

For an EIM scenario, a better look into the state space of the system needs clarification of what is meant by system variables [34]. According to the explanation of different domains of computation as described in Section 1, the variables of the system belong to the *domain of measurements* as schematically shown in Figure 2. The voltages and the set of properties which define them in this domain, i.e., amplitude, frequency and phase, can be represented with:

$$v_i = a_i \cdot func_p(f_i, \phi_i) \tag{2}$$

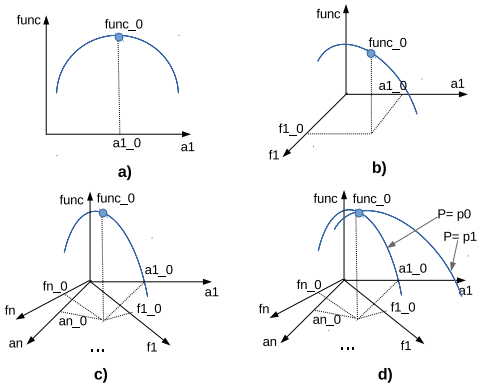


Figure 4. CNT computing system within a system theory framework.

where v_i is voltage on the i-th electrode, a_i the amplitude, $func_p$ some periodic function, f_i frequency of the function $func_p$ and, finally, ϕ_i the phase of the voltage, all referring to the i-th electrode. The symbols are left lower case to remind that all of these values can be time varying.

Let us now consider an example in which for a system to perform functionality $func\_0$, for the input x_0, an output value y_0 is desired (Figure 4 *a*)). For simplicity, the variables on each of the axes are assumed to be scalars. When different configuration voltages are applied to the material, they change the system variables so that it passes through various states in the state space along some trajectory. Further, let us assume that only one electrode is used for configuration voltage and only one voltage parameter is changed, for example amplitude. By changing the amplitude along the a_1 axis different input-output mappings will be performed by the system. EIM would then search through the space until $func\_0$ point is reached. If also the frequency of the voltage v_1 is changed, then the state space could be searched along two axes as shown in Figure 4 *b*). And even further, if more than one electrode is used for configuring the material, then, in general, the space would look something like in Figure 4 *c*) and would be searchable along high number of axes, the limitation being only the physical number of electrodes in the system. Moreover, the state space may grow due to the change in some parameter, like temperature or light, as shown in Figure 4 d), which may all increase the size of the state space to search for the solution.

Table 1. Different materials used in the experiments.

Material	**SWCNT Concentration,** $wt\%$
B09S12	0.53%
B15S03	1.25%
B15S04	1.50%
B15S08	5.00%

4 A Detailed View of Material Dynamics

Experiments are performed on SWCNT mixed with PBMA on a micro electrode array supplied by Durham University. Material samples and micro electrode arrays are produced in a process where SWCNT-PBMA mixture is dissolved in anisole (methoxy benzene). The material samples are prepared on $4x4$ grids of gold micro-electrode arrays with pads of $50\mu m$ and pitch of $100\mu m$, see Figure 3c. The preparation is done by dispensing $20\mu L$ of the material onto the electrode area. The concentration of SWCNTs varies as shown in Table 1 where the material samples used in our experiments are listed. The SWCNT mixed with PBMA material dispersed over electrode array is baked for $30min$ at $90C°$. The solvent dries out and leaves a thick film of immovable SWCNTs supported by polymer molecules. The substrate is cooled slowly over a period of 1h. This process leaves a variable distribution of nanotubes across the electrodes. Typically, carbon nanotubes are 30% metallic and 70% semi-conducting, while PBMA creates insulation areas within nanotube networks. Such electrical properties of the material may allow non-linear current versus voltage characteristics.

The coverage of gold microelectrodes with randomly dispersed nanotubes varies and some of the electrodes may even be left with little or no coverage, as visible in the Scanning Electron Microscope (SEM) image in Figure 5.

Initial investigation of the material response to various input signals showed several interesting behaviors in the material [1]. The goal was to gain insight into the material dynamics to identify suitable ways in which the material can be manipulated to perform computation.

As mentioned, EIM requires an interface between a digital computer which runs the EA and the material whose physics undergoes analog processes. This interface is typically provided by the Mecobo board. However, in order to better understand the underlying properties of the material and its

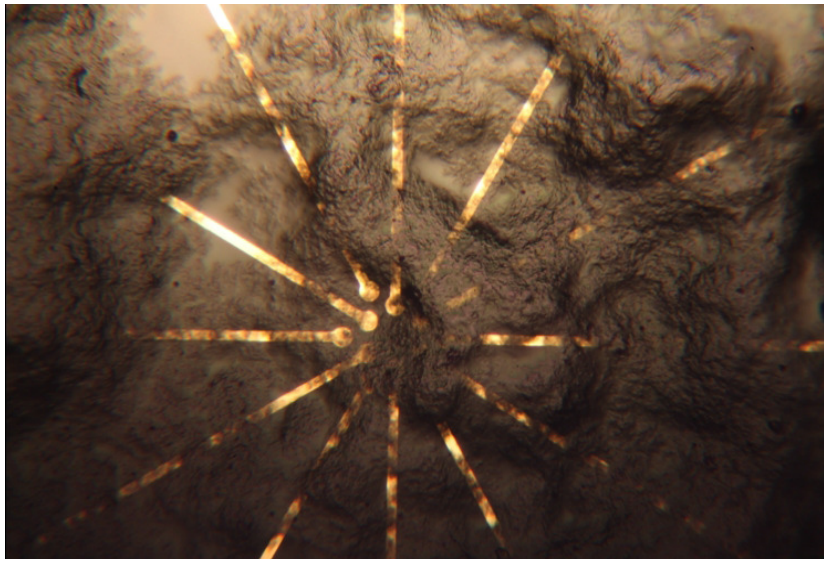


Figure 5. SEM image of gold electrode array with different coverage of nanotubes. Adopted from [9].

responses, it is necessary to use more precise instruments. In these experiments, oscilloscopes and signal generators were used to get a more detailed view of the material dynamics.

4.1 Experimental Setup

In the experiments herein, we connect a material slide to a Hewlett Packard 33120A $15MHz$ function / arbitrary waveform generator (used as input) and an Agilent 54622D $100MHz$ mixed signal oscilloscope (used as output). Input signals are square waves at different frequencies from the signal generator and the output signals are recorded on the oscilloscope.

The input / output pins were chosen so that there would be an equal distance between microelectrode pads within the microelectrode array (Figure 3c). The placement of input and output signals on the material slide is shown in Figure 6, where the input probe (from the signal generator) is placed on pin #2 (IN) and the two output probes (to the oscilloscope) are connected to pins #9 (OUT1) and #7 (OUT 2).

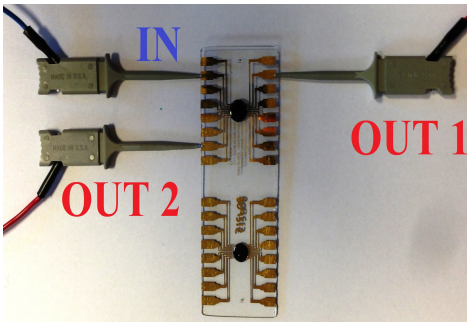
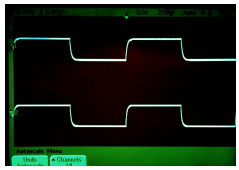


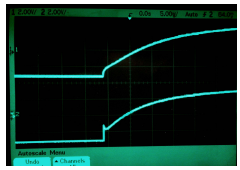
Figure 6. Material slide and pins connected to signal generator (IN) and oscilloscope (OUT).

4.2 Results and discussion

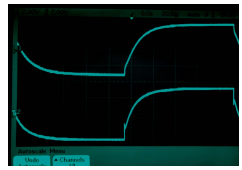
Figure 7 presents the experimental results. In particular, Figures 7a) show several snapshots of the material response on two different pins at different frequencies, ranging from $1KHz$ (Figure a1) to $14MHz$ (Figure a12). At $1KHz$ the signals may seem similar (a1), where the material charges-up and subsequently discharges, but in a zoomed in snapshot, i.e., where a part of the response is shown at a higher resolution (a2), a voltage spike is visible on the second probe which is not present on the first probe. This is better visible at $5KHz$ (a3), $30KHz$ (a4) and $100KHz$ (a5), where it is possible to notice that on the rising front there is a sudden voltage increase/drop. The material behavior is capacitor-like. Starting from $500KHz$ (a6), which is also zoomed in (a7), the second probe signal is similar to a square wave (most of the harmonic frequencies are passed) while the first probe acts more like a filter. The difference is caused by different concentrations of CNTs between the IN-OUT electrodes, i.e., different paths the current is enabled to follow between the electrodes. In both cases, there is a resonance phase which results in a deterministic yet semi-chaotic waveform. This may be the effect of some conducting sub-networks in the material that are enabled at specific frequencies and disabled at others. At 2, 5 and $8.5MHz$ the measured voltage decreases while frequency increases. At $10MHz$ (a11) a strange phenomenon is observed where both signals show a voltage increase. The effect is more prominent on the first output. We ascribe such behavior to be due to a feedback effect where harmonics of some frequencies are fed again into the material by some nanotube sub-networks. At $14MHz$ (a12) the signal on the second probe is sinusoidal, i.e., only one harmonic is present. As such, it may



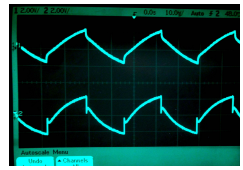
(a1) $1kHz$ (scale: $5V, 200\mu s$)



(a2) $1kHz$ (zoom, $2V, 5\mu s$)

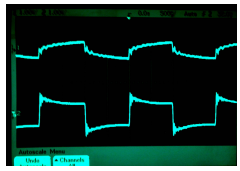


(a3) $5kHz$ (scale: $2V, 20\mu s$)

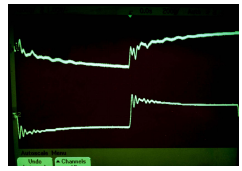


(a4) $30kHz$ (scale: $2V, 10\mu s$)

(a5) $100kHz$ (scale: $1V, 2\mu s$)



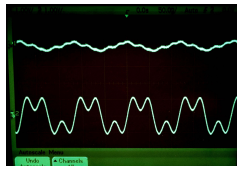
(a6) $500kHz$ (scale: $1V, 500ns$)



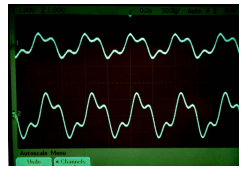
(a7) $500kHz$ (zoom, $500mV, 200ns$)

(a8) $2MHz$ (scale: $1V, 100ns$)

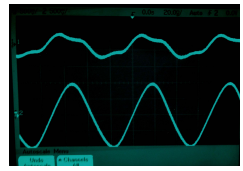
(a9) $5MHz$ (scale: $1V, 50ns$)



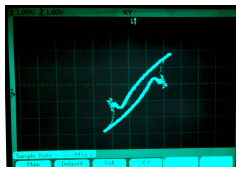
(a10) $8.5MHz$ (scale: $1V, 50ns$)



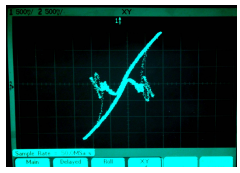
(a11) $10MHz$ (scale: $1V, 50ns$)



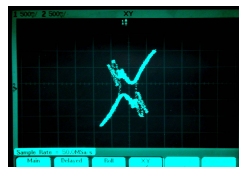
(a12) $14MHz$ (scale: $500mV, 20ns$)



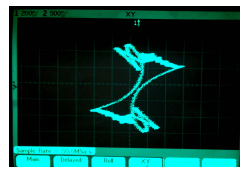
(b1) $30kHz$ (scale: $1V, 1V$)



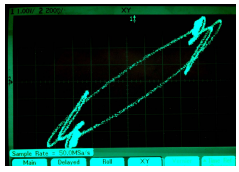
(b2) $60kHz$ (scale: $500mV, 500mV$)



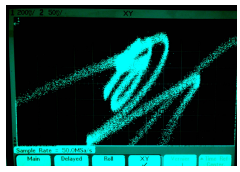
(b3) $100kHz$ (scale: $500mV, 500mV$)



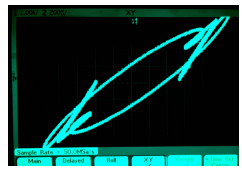
(b4) $500kHz$ (scale: $200mV, 200mV$)



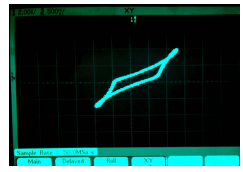
(c1) $350kHz$ (scale: $1V, 1V$)



(c2) $2MHz$ (zoom, $200mV, 50mV$)



(c3) $5MHz$ (scale: $1V, 200mV$)



(c4) $10MHz$ (scale: $2V, 500mV$)

Figure 7. Oscilloscope screenshots. The resolution is indicated in parentheses. The resolutions have been chosen so as to be able to show interesting results at different scales.
(a) Voltage responses on 2 different pins with input square wave at different frequencies.
(b) XY plots, X (OUT1) is plotted against Y (OUT2) at different frequencies.
(c) XY plots, X (IN) is plotted against Y (OUT1) at different frequencies.

be concluded that with a single square wave input it is possible to observe a rich variety of behaviors while the frequency spectrum is traversed.

As the system produces uniform, stable, and semi-chaotic behaviors, it is of particular interest to visualize input-output responses and output-output relations in order to better understand traversed trajectories and attractors. For this purpose, XY plots are shown in Figure 7b), where OUT1 is plotted against OUT2 and Figure 7c), where IN is plotted against OUT1. In Figure 7b1), some orbits are present at $30KHz$. Similar orbits are visible at $60KHz$ (b2) and $100KHz$ (b3), moving towards opposite corners to those where the impulse is. After each impulse, there is a semi-chaotic orbit that relaxes before the next impulse arrives, as the semi-chaotic behavior is annihilated by the lack of energy in the material, until the arrival of the next impulse. This suggests that chaotic behavior may be present, yet particularly difficult to observe.

XY plots between input and output are shown in Figure 7c). These figures represent the phase space of the system (input-output pin pair). Figure 7c1) is obtained at $350Khz$. Several oscillating orbits are present, which are zoomed-in at $2MHz$ (c2). The same effect is observed for frequencies up to $5MHz$ (c3) while for frequencies around $10MHz$ and higher we observe a hysteresis loop, which indicates that some saturation may have been reached in the material. Some sort of non-linearity seems present, which is always a good indicator that the system may achieve complex behavior.

To summarize this set of results, even if a single square wave input signal is used, the resulting output shows a variety of behaviors. Square waves [35] produce richer dynamics than what may be achieved by a single static voltage or by a sinusoidal wave. Such richness of the response is due to the rich spectrum of the square waves which contains a variety of harmonics. In particular, some of the nanotube sub-networks may be sensitive to certain frequencies. Therefore, square waves may be better suited to penetrate the material and exploit the nanocomposite's intrinsic properties.

5 A Complexity View of Material Dynamics

The initial experiments with the oscilloscope measurements gave valuable insight into the different dynamics available in the material. However, such detailed measurements only give a very narrow view of the possible behaviors of the system. In order to get a broad picture of the space of possible material

dynamics, one has to sacrifice some amount of detail. By using the Mecobo hardware platform (Section 3) we are able to explore material dynamics at a higher level.

Mecobo allows us to scan a much wider range of signal frequencies, explore a myriad of different material locations and easily analyze the results on a PC. For these experiments we use the digital signal generator on Mecobo to generate square waves as input signals. The output signal is sampled as analog voltage using the on-board AD converter (Figure 3).

Complexity of the input/output signal is used as metric to classify different types of material dynamics. We use compressibility as an estimate of complexity as described in Section 2.4. Since we are primarily interested in the complexity contribution of the material (and not the complexity of the input signal itself), we adapt the *complexity ratio*:

$$C_r = \frac{C_o}{C_i}$$

where C_o is the complexity of the output signal and C_i is the complexity of the input signal.

We present three sets of experiments where the computational complexity of the material is explored:

1. Complexity as number of input signals are increased
2. Complexity as function of one input frequency
3. Complexity as function of two input frequencies

5.1 Experimental Setup

For all the experiments, a set of input signals are sent through the material and a single output signal is recorded. The input signals are digital square waves in the range $400Hz$ to $25kHz$. The amplitude of the square waves is $0-3.3V$, which means that the material is exposed to a sharp rise and fall of the signal in this range. The duty cycle is held constant at 50%.

The output signal is recorded as analog voltage over time and sampled at a frequency of $500kHz$ for 10ms resulting in an output buffer of 5000 samples.

In order to compare the analog output signal to the digital input signal, we digitize the output signal by using the mean voltage as digital threshold. In other words, samples above the mean correspond to logical 1 and samples below the mean correspond to logical 0. To reduce sensitivity to noise, we apply hysteresis so that transitions between logic levels happen only if the analog voltage crosses the mean by a noise margin.

Complexity is estimated by compressing the sample buffer with zlib (zlib is based on LZ77 [36]) and calculating the length of the compressed string. Input complexity C_i is calculated based on a set of ideal square waves sampled at the same frequency as the output signal ($500kHz$).

All the experiments are repeated for the different material samples listed in Table 1.

5.1.1 Complexity as number of input signals are increased

In the first experiment, the number of input pins are increased from 1 to 15. Input pins are selected at random and for each input pin a random frequency is chosen in the range of $400Hz - 25kHz$. The output signal is recorded from pin #0. The experiment is repeated 100 times for each number of input pins resulting in 1500 output signals.

5.1.2 Complexity as function of one input frequency

The second set of experiments provides a more detailed view of a subset of the first experiment by traversing the input frequency spectrum. Frequencies are increased from $400Hz - 25kHz$ in steps of $1000Hz$ resulting in 25 different input frequencies. The number of input pins are again increased from 1 to 15 but the same frequency is now applied to all input pins. In addition, both input pins and output pins are selected at random. For each number of input pins and for each frequency, the experiment is repeated 100 times resulting in 37500 output signals.

5.1.3 Complexity as function of two input frequencies

In the third experiment, we again traverse the same input frequency spectrum ($400Hz - 25kHz$), but this time for two input pins. In other words, the frequency spectrum is traversed in two dimensions resulting in 25^2 pairs of

input frequencies. Both input pins and output pins are selected at random. The experiment is repeated 10 times for each set of input/output pins.

5.2 Results and discussion

5.2.1 Complexity as number of input signals are increased

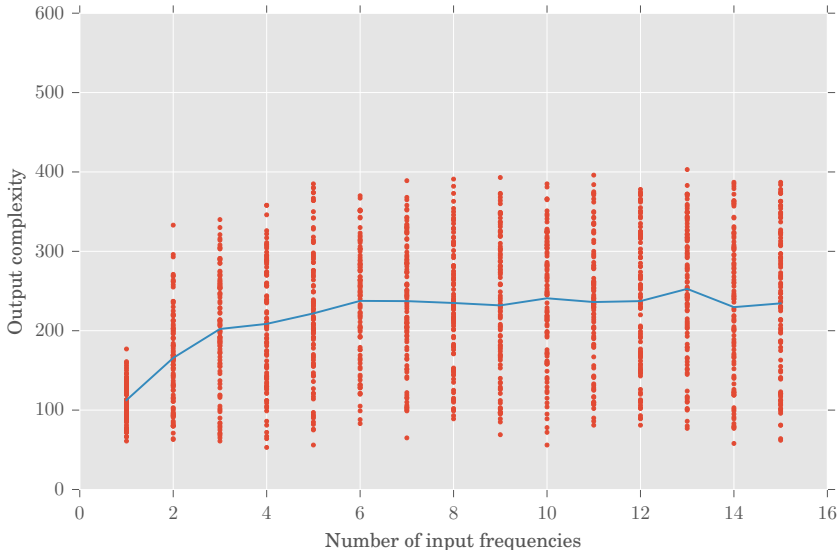
Figure 8 shows output complexity C_o measured over the range of 1-15 input frequencies. The blue line shows the mean output complexity value for each of the 100 data points. As shown in the plots, the output complexity increases with the number of input signals. There appears to be a fairly sharp rise in complexity as the number of square wave inputs are increased from 1 to 4. After this point the output complexity appears to saturate.

The scatter plot shows a fairly high variation in output complexity when the number of input signals exceeds one. This indicates that the materials exhibit a rich variety in output depending on the frequency and/or the choice of input pins.

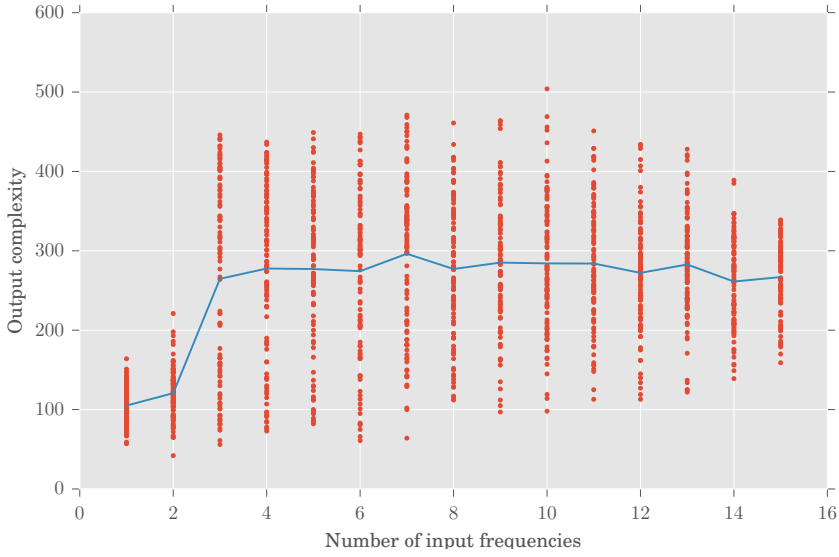
A more detailed view is obtained when output complexity is plotted against input complexity (Figure 9). In these plots, it becomes clear that the input complexity C_i increases almost linearly with the number of input signals. Output complexity, however, saturates quickly above 3-4 input signals. In other words, above this level the added complexity from the input signal is not observed at the output.

Again the richness of output complexity can be observed. The output signal is generally less complex than the input signal, which indicates that the material acts as a filter or stable attractor. However, there are situations where the complexity of the output signal exceeds that of the input signal. The input complexity is estimated from *ideal* square waves, which are not directly comparable to the signals generated by the hardware platform. However, the estimate does give an indication that the materials exhibit rich dynamics.

From Figures 8 and 9 it appears as if higher concentrations of SWCNTs result in higher output complexity. Such a trend is counter-intuitive, since as concentration increases the electrical resistance of the material is reduced. As resistance goes towards zero the material should act more like a wire, which means that the input signals should pass through unaltered. If multiple input signals are sent through a wire, the output signal would simply be the sum

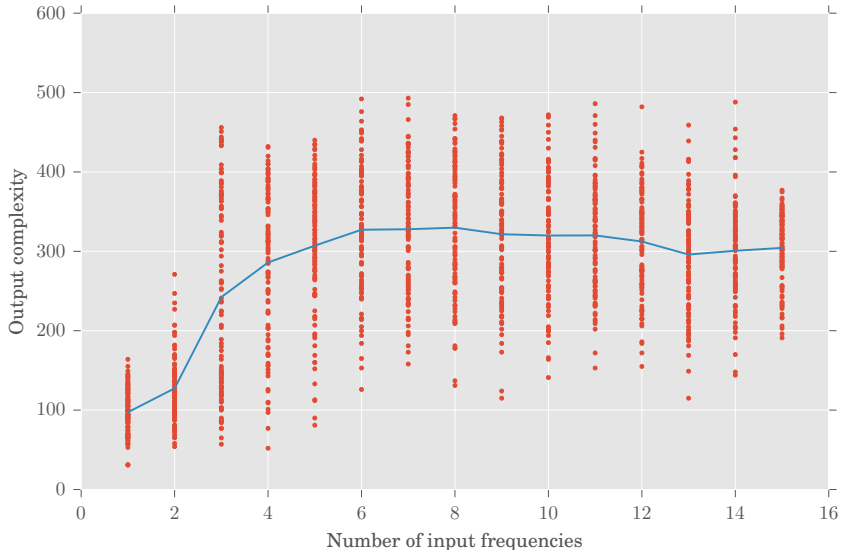


(a) Material B09S12 (0.53 wt% SWCNT)

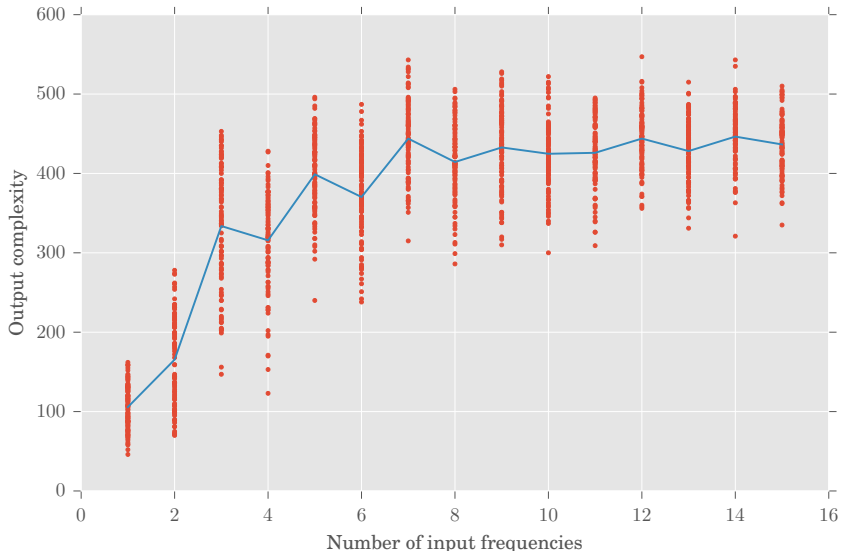


(b) Material B15S03 (1.25 wt% SWCNT)

Figure 8. Output complexity as the number of input frequencies are increased from 1 to 15 for four different material samples. The red scatter plot shows individual measurements while the blue line indicates the mean values for each of the 100 data points.

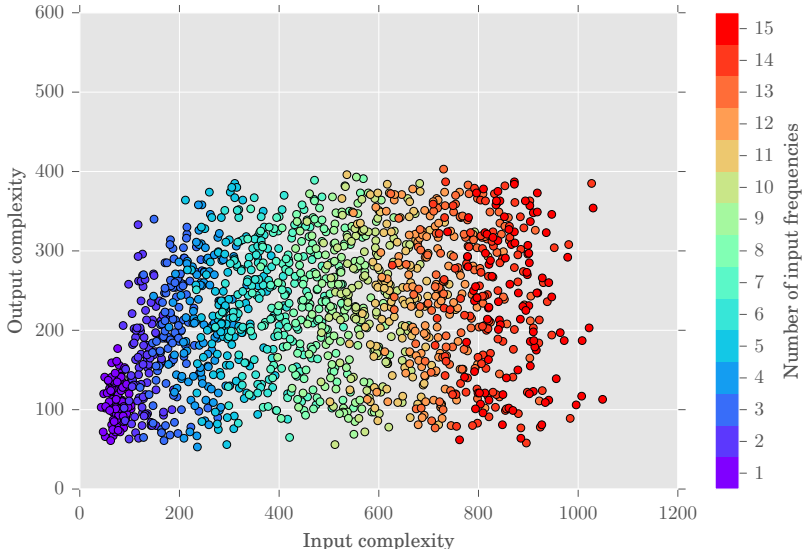


(c) Material B15S04 (1.50 wt% SWCNT)

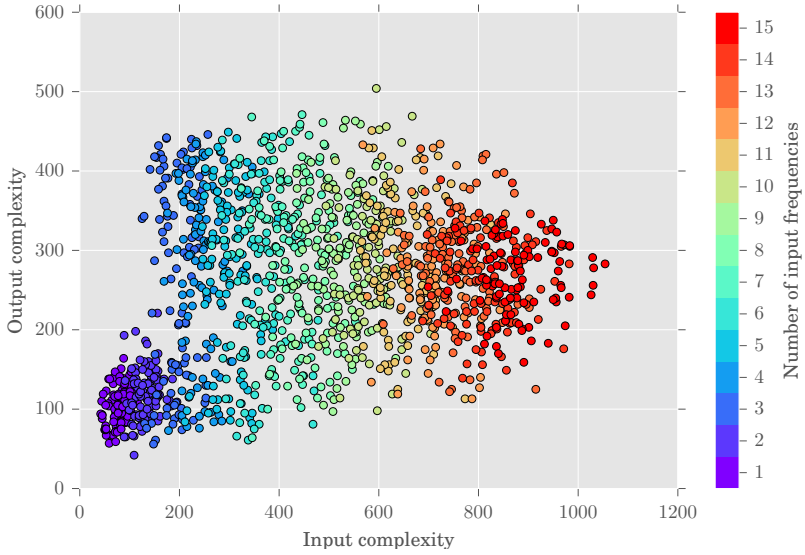


(d) Material B15S08 (5.00 wt% SWCNT)

Figure 8 (cont.). Output complexity as the number of input frequencies are increased from 1 to 15 for four different material samples. The red scatter plot shows individual measurements while the blue line indicates the mean values for each of the 100 data points.

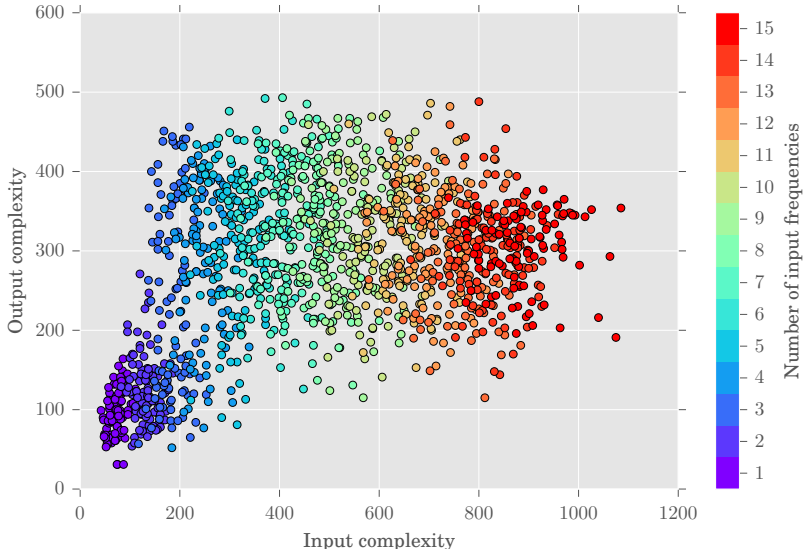


(a) Material B09S12 (0.53 wt% SWCNT)

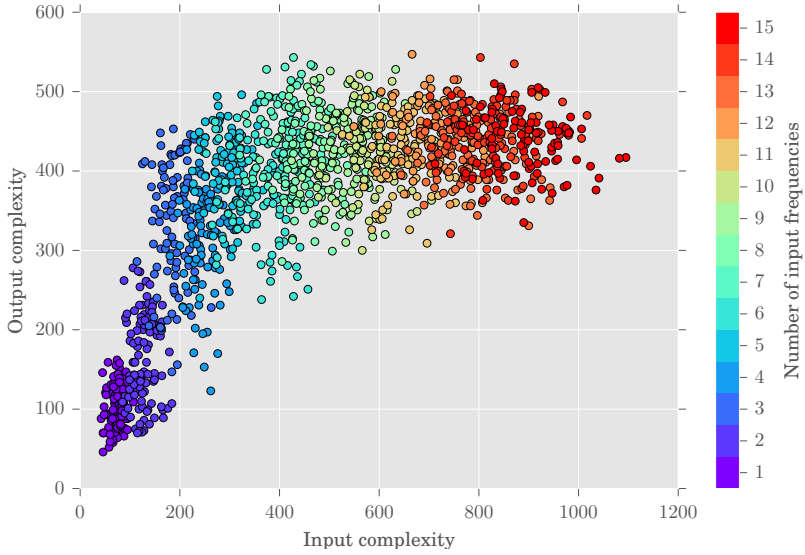


(b) Material B15S03 (1.25 wt% SWCNT)

Figure 9. Input vs output complexity as number of input frequencies are increased from 1 to 15 for four different material samples. The dots are colored according to the number of input frequencies used.



(c) Material B15S04 (1.50 wt% SWCNT)



(d) Material B15S08 (5.00 wt% SWCNT)

Figure 9 (cont.). Input vs output complexity as number of input frequencies are increased from 1 to 15 for four different material samples. The dots are colored according to the number of input frequencies used.

of the input signals. Therefore, it would be interesting to investigate how closely the output signal resembles the sum of the input signals.

Figure 10 plots input vs output complexity when the input signals are summed together before C_i is estimated. For the material with high SWCNT concentration (B15S08, Figure 10b) there is now a clear linear relationship between input complexity and output complexity. In other words, this material appears to behave much like a wire that simply sums the input signals together in some way. Lower SWCNT concentrations, however, display more diverse behavior as can be seen in Figure 10a, where there is no clear linear relationship between C_i and C_o.

5.2.2 Complexity as function of one input frequency

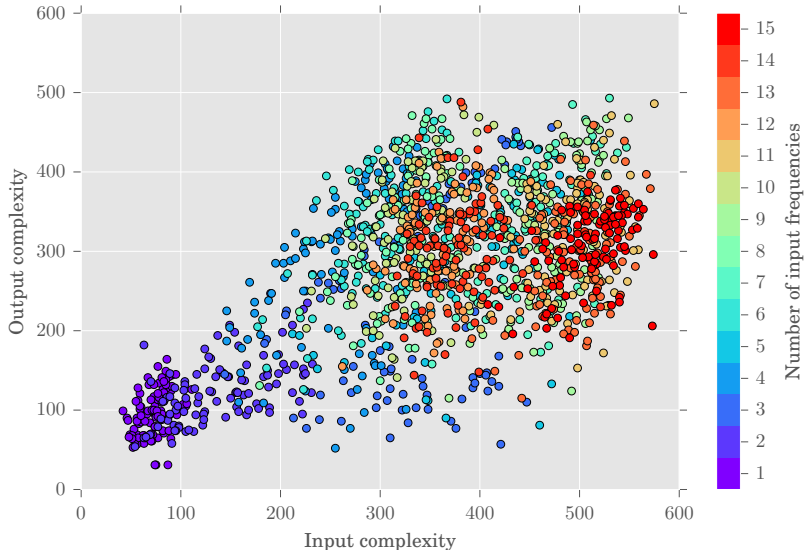
Figure 11 shows the mean *complexity ratio* C_r over the range of input frequencies applied to the four material samples. From the plots it is evident that C_r is highly dependent on the input frequency with spikes at certain frequencies. Complexity appears to be fairly consistent across the four material samples, i.e., the materials are sensitive to the same frequencies.

Applying the input frequency to more pins does not seem to affect the mean complexity by much. However, there is a clear reduction in complexity variation, as can be seen from Figure 12, where standard deviation of the complexity ratio is shown. One possible explanation is that the input signal is effectively amplified as it is applied to more input pins.

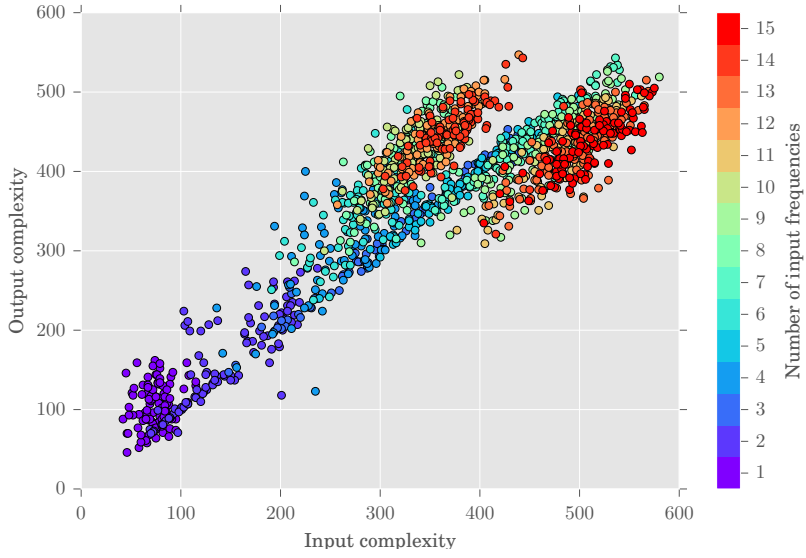
Another trend that can be seen from the plots in Figure 12 is an inverse relationship between complexity variation and the SWCNT concentration, i.e., more uniform output complexity with increased SWCNT concentration. This may be due to the fact that higher SWCNT concentration leads to a lower electrical resistance in the material and thus more pathways for the input signal to reach the output pin. However, one exception can be observed for the B15S04 sample where a higher variation is found when the frequency is applied to only one input pin. This likely indicates that one electrode is only partially connected to the material in this particular sample.

5.2.3 Complexity as function of two input frequencies

By sweeping the two input frequencies applied to the material we get a more detailed view of some of the results from the first experiment. Figure 13

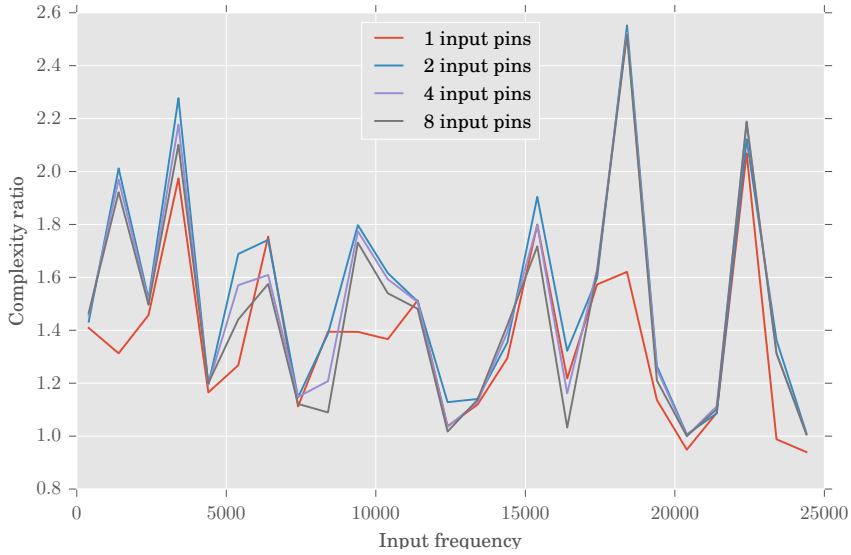


(a) Material B15S04 (1.50 wt% SWCNT)

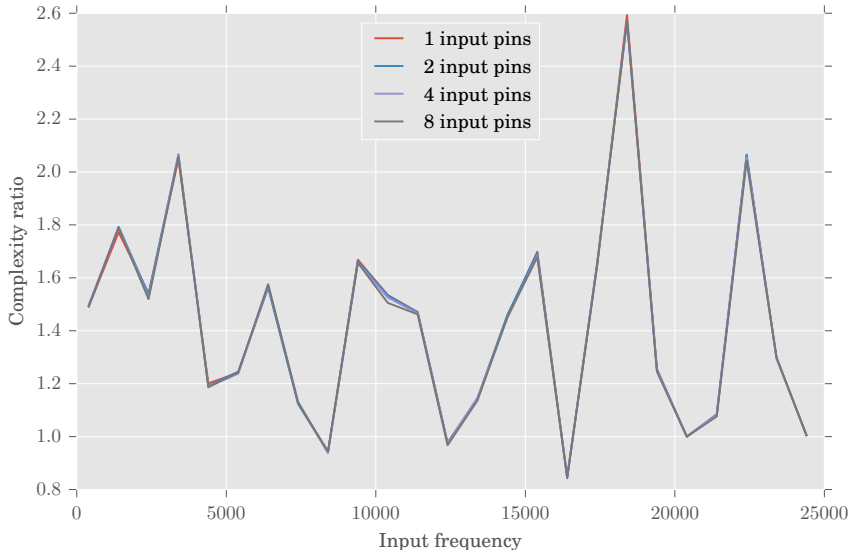


(b) Material B15S08 (5.00 wt% SWCNT)

Figure 10. Input vs output complexity when the input signals are summed together before input complexity is estimated. Results from two material samples with different SWCNT concentrations are shown. The dots are colored according to the number of input frequencies used.

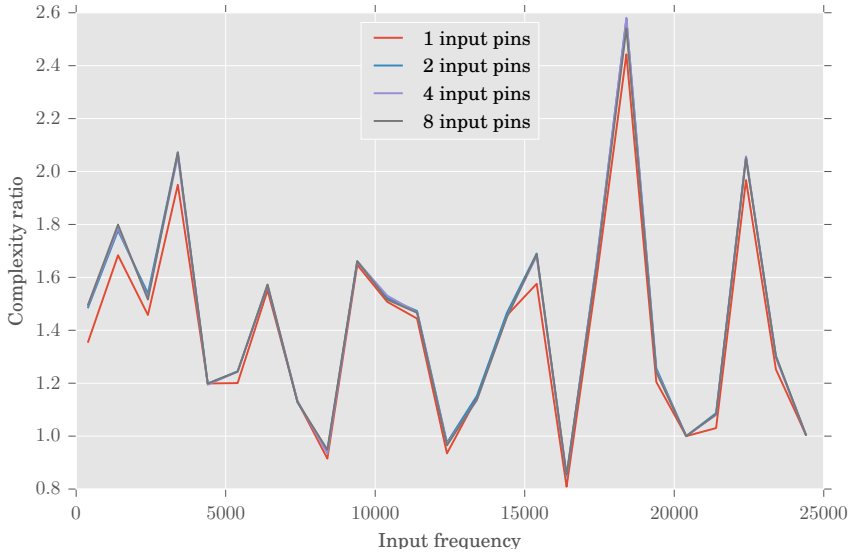


(a) Material B09S12 (0.53 wt% SWCNT)

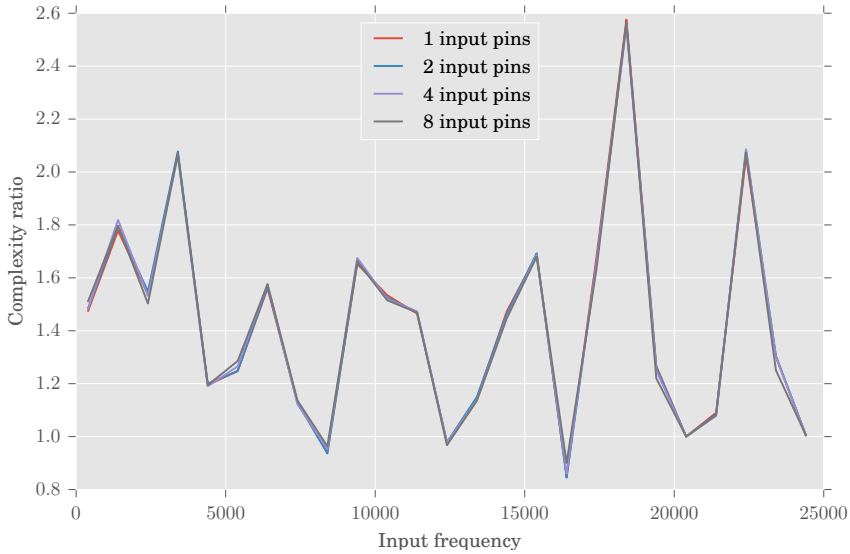


(b) Material B15S03 (1.25 wt% SWCNT)

Figure 11. Mean complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

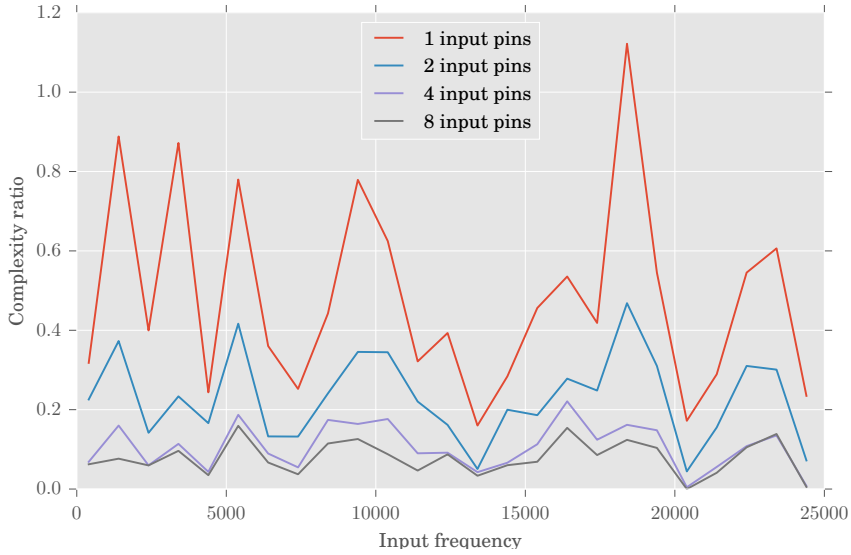


(c) Material B15S04 (1.50 wt% SWCNT)

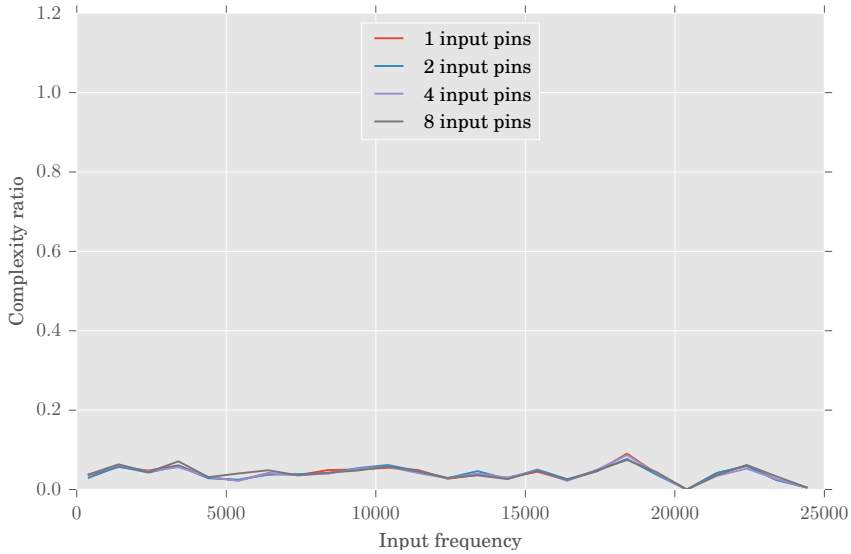


(d) Material B15S08 (5.00 wt% SWCNT)

Figure 11 (cont.). Mean complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

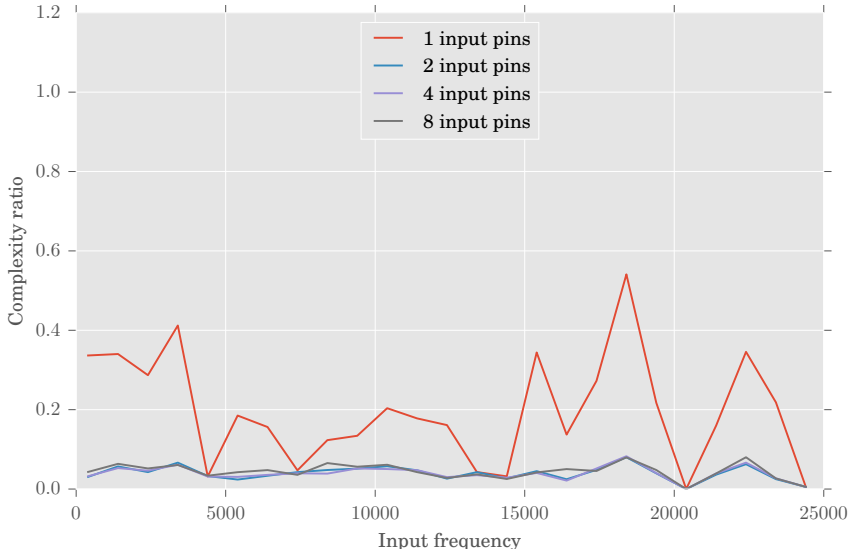


(a) Material B09S12 (0.53 wt% SWCNT)

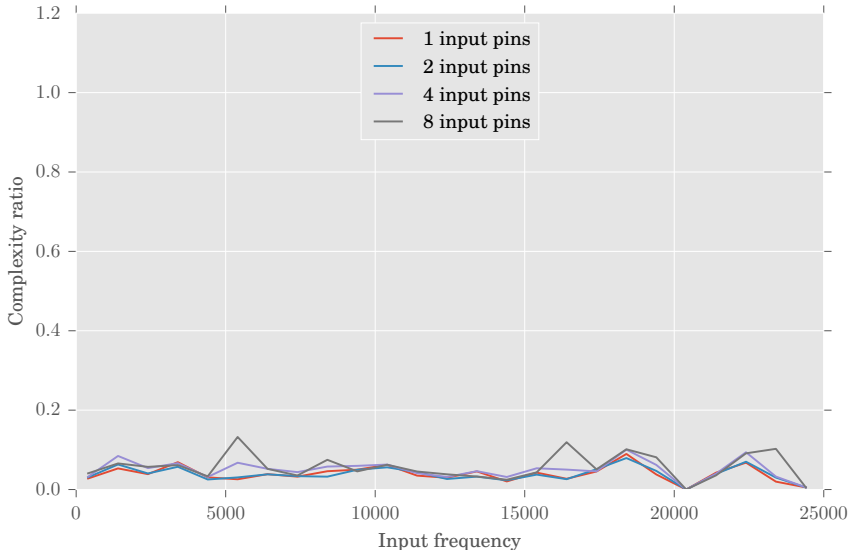


(b) Material B15S03 (1.25 wt% SWCNT)

Figure 12. Standard deviation of complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

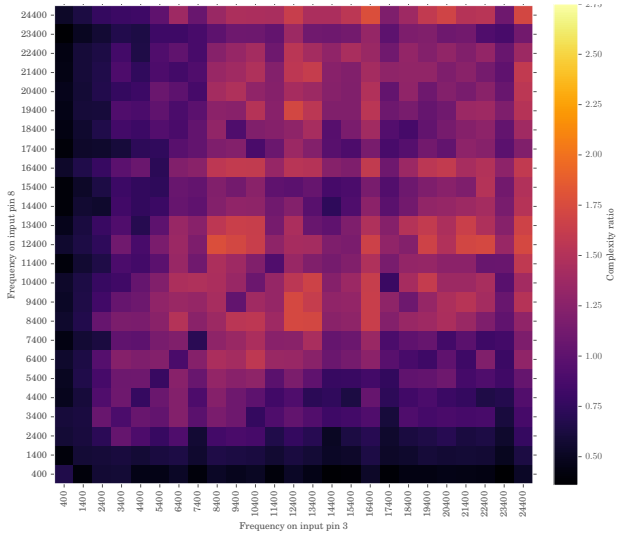


(c) Material B15S04 (1.50 wt% SWCNT)

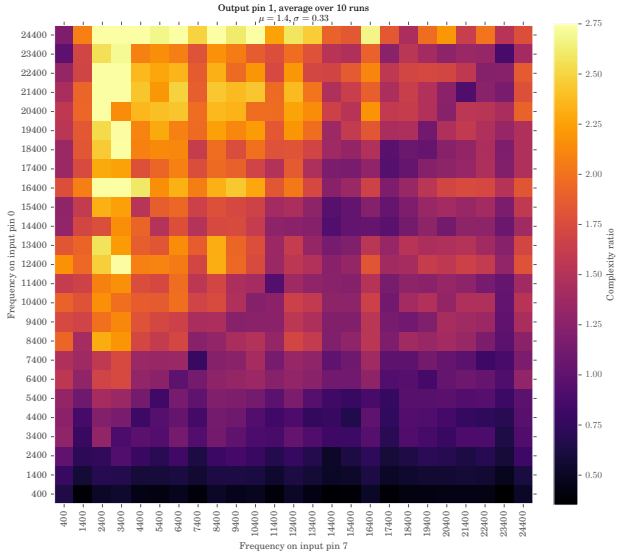


(d) Material B15S08 (5.00 wt% SWCNT)

Figure 12 (cont.). Standard deviation of complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

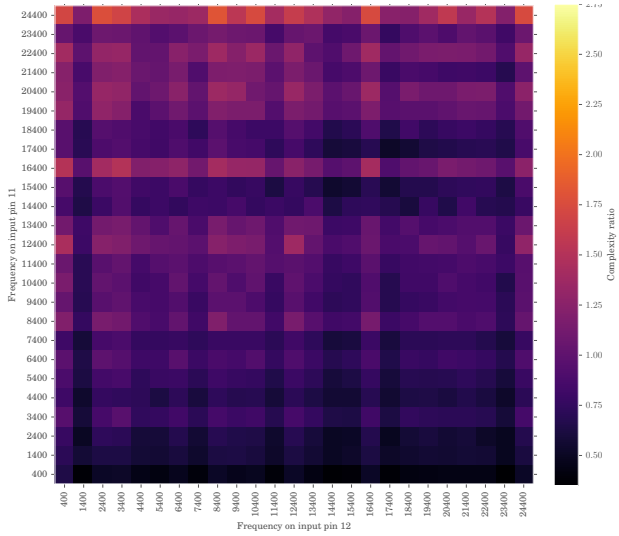


(a) Material B09S12 (0.53 wt% SWCNT), input pins 3 and 8, output pin 15

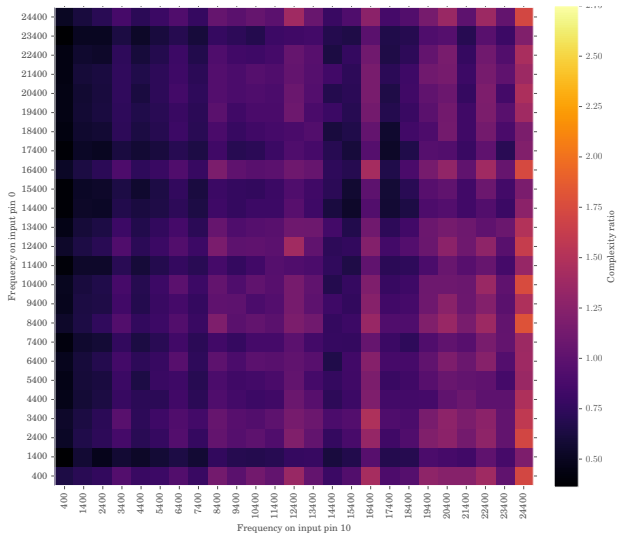


(b) Material B09S12 (0.53 wt% SWCNT), input pins 7 and 0, output pin 1

Figure 13. Complexity ratio as function of two input frequencies (X and Y axes). The heat maps shows complexity ratio C_r averaged over 10 runs. Colors range from dark purple (low complexity) to bright yellow (high complexity). Four heat maps are shown for two material samples: B09S12 (13a-13b) and B15S08 (13c-13d). Each heat map shows complexity when input is applied to different input/output pins.



(c) Material B15S08 (5.00 wt% SWCNT), input pins 12 and 11, output pin 10



(d) Material B15S08 (5.00 wt% SWCNT), input pins 10 and 0, output pin 6

Figure 13 (cont.). Complexity ratio as function of two input frequencies (X and Y axes). The heat maps shows complexity ratio C_r averaged over 10 runs. Colors range from dark purple (low complexity) to bright yellow (high complexity). Four heat maps are shown for two material samples: B09S12 (13a-13b) and B15S08 (13c-13d). Each heat map shows complexity when input is applied to different input/output pins.

depicts complexity ratio as a heat map where the two input frequencies are swept in the X and Y axes and color represents complexity. The colors range from dark purple (low complexity) to bright yellow (high complexity). As with one input signal, the heat maps show clearly that the complexity landscape is dependent on the selection of input frequencies.

Figures 13a and 13b depict complexity for the same material sample B09S12, but with different selection of input and output pins. As can be seen, the two heat maps display clear differences in complexity ratio, where the latter pin configuration (13b) generally exhibits more complex output. However, this is not always the case, as can be seen in Figures 13c and 13d, where different input locations result in quite similar complexity landscapes.

6 Conclusion

The general ideas, experiments, and results presented relates to dynamics performed by SWCNT and PBMA nanocomposites, which may be exploited by EIM. The materials and experimental system (as presented in Section 2) has shown promising computational behavior on a variety of problems. In this work, the behaviors are related to measurable dynamic behavior. That is, the experiments are designed to capture dynamic properties of the materials as to gain an understanding of what inherent dynamics are observable in an EIM setting. The approach taken is to view the material, i.e., physical system, as a hierarchical information processing device (Figure 2). At the bottom level the physical dynamics, i.e., quantum effects due to mechanisms of electron transmission through carbon nanotubes, are not observable within a reasonable resource usage. As such, the lowest level is treated only at a conceptual level. Dynamics at the bottom level are only observed as resulting voltages in the analogue domain. The information available at this level is exploited to gain insight into the electrical properties of the material when exposed to dynamic input stimuli. At the top level the material is interpreted as a discrete dynamical system. However, the observable dynamics at this discrete level is a result of all the underlying physics.

As stated by Miller et al. [3]: "...exploit the intrinsic properties of materials, or “computational mediums”, to do computation, where neither the structure nor computational properties of the material needs to be known in advance". The statement may indicate that any material can be looked at as a black-box. However, insight into what properties are available for

evolution provides knowledge on how to construct a successful EIM system. Our findings show that the materials exhibit rich dynamical properties observable at the analogue level. Figure 7 shows the behavior at an (close to) analogue time and voltage scale. The properties of these behaviors are available for exploitation by evolution, even if not explicitly controllable from the top discrete digital domain.

At the top level, the abstract measurements of complexity shows how such a measurement can indicate what computational problems the EIM system may handle. Especially, the experimental results from Figure 13 show that the materials tested include behavior found in classifier systems, such as Thompson's frequency discriminator [5] (generally a trend of reduced complexity as illustrated in Figure 13d). From the same experiment, Figure 13b shows an increase in complexity generated by the dynamics of the material. A clear indication of a system which has more internal (observable) states than of the input data.

Our results also reveal several specific properties of the SWCNT materials used. In particular, as the number of input signals grows, a saturation of output complexity is reached. From an EIM perspective this is interesting, since it implies that information is filtered when many input signals are applied. The results also show a wide variety in output complexity depending on input frequency and selection of input/output pins. An indication that the materials are capable of many different modes of operation.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP/2007-2013) under grant agreement number 317662.

References

[1] S. Nichele, D. Laketić, O. R. Lykkebø, and G. Tufte, "Is there chaos in blobs of carbon nanotubes used to perform computation?" in *FUTURE COMPUTING 2015, The Seventh International Conference on Future Computational Technologies and Applications.* ThinkMind, 2015, pp. 12–17.

[2] H. Broersma, F. Gomez, J. Miller, M. Petty, and G. Tufte, "Nascence project: nanoscale engineering for novel computation using evolution," *International journal of unconventional computing*, vol. 8, no. 4, pp. 313–317, 2012.

[3] J. Miller, S. Harding, and G. Tufte, "Evolution-in-materio: evolving computation in materials," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 49–67, 2014.

[4] J. Miller and K. Downing, "Evolution in materio: Looking beyond the silicon box," in *The 2002 NASA/DoD Conference on Evolvable Hardware*, A. Stoica, J. Lohn, R. Katz, D. Keymeulen, and R. S. Zebulum, Eds., Jet Propulsion Laboratory, California Institute of Technology. Alexandria, Virginia: IEEE Computer Society, 15-18 July 2002, pp. 167–176.

[5] A. Thompson, *Hardware evolution - automatic design of electronic circuits in reconfigurable hardware by artificial evolution.* CPHC/BCS distinguished dissertations, 1998.

[6] S. Harding and J. Miller, "Evolution in materio : A real-time robot controller in liquid crystal," in *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, J. Lohn, D. Gwaltney, G. Hornby, R. Zebulum, D. Keymeulen, and A. Stoica, Eds. Washington, DC, USA: IEEE Press, 29 June-1 July 2005, pp. 229–238.

[7] P. Cariani, "To evolve an ear: epistemological implications of Gordon Pask's electrochemical devices," *Systems Research*, vol. 10, no. 3, pp. 19–33, 1993.

[8] K. Clegg, J. Miller, K. Massey, and M. Petty, "Travelling salesman problem solved "in materio" by evolved carbon nanotube device," in *Parallel Problem Solving from Nature - PPSN XIII*, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, Eds. Springer International Publishing, 2014, vol. 8672, pp. 692–701.

[9] A. Kotsialos, K. Massey, F. Qaiser, D. Zeze, C. Pearson, and M. Petty, "Logic gate and circuit training on randomly dispersed carbon nanotubes." *International journal of unconventional computing.*, vol. 10, no. 5-6, pp. 473–497, September 2014.

[10] M. Mohid, J. Miller, S. Harding, G. Tufte, O. R. Lykkebø, K. Massey, and M. Petty, "Evolution-in-materio: Solving bin packing problems using materials," in *The 2014 IEEE Conference on Evolvable Systems - ICES, IN PRESS*. IEEE Computer Society, 2014.

[11] M. Mohid, J. Miller, S. Harding, G. Tufte, O. Lykkebø, M. Massey, and M. Petty, "Evolution-in-materio: Solving machine learning classification problems using materials," in *Parallel Problem Solving from Nature – PPSN XIII*, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, Eds. Springer International Publishing, 2014, vol. 8672, pp. 721–730.

[12] M. Mohid, J. Miller, S. Harding, G. Tufte, O. R. Lykkebø, K. Massey, and M. Petty, "Evolution-in-materio: A frequency classifier using materials," in *The 2014 IEEE Conference on Evolvable Systems - ICES, IN PRESS*. IEEE Computer Society, 2014, pp. 46–53.

[13] M. Mohid, J. Miller, S. Harding, G. Tufte, O. Lykkebo, M. Massey, and M. Petty, "Evolution-in-materio: Solving function optimization problems using materials," in *Computational Intelligence (UKCI), 2014 14th UK Workshop on*, D. Neagu, M. Kiran, and P. Trundle, Eds. IEEE, September 2014, pp. 1–8.

[14] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems of information transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[15] G. Pask, "Physical analogues to growth of a concept," *Mechanisation of Thought Processes*, pp. 877–922, 1959.

[16] R. Stewart, "Electrochemically active field-trainable pattern recognition systems," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 5, no. 3, pp. 230–237, 1969.

[17] J. W. Mills, "Polymer processors," Technical Report TR580, Department of Computer Science, University of Indiana, Tech. Rep., 1995.

[18] S. L. Harding and J. F. Miller, "Evolution in materio: Computing with liquid crystal," *Journal of Unconventional Computing*, vol. 3, no. 4, pp. 243–257, 2007.

[19] S. Harding and J. F. Miller, "Evolution in materio: A tone discriminator in liquid crystal," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2. IEEE, 2004, pp. 1800–1807.

[20] ——, "Evolution in materio: Evolving logic gates in liquid crystal," in *In Proceedings of the workshop on unconventional computing at ECAL 2005 VIIIth European.* Beckington, UK, 2005, pp. 133–149.

[21] M. Mohid and J. Miller, "Evolving robot controllers using carbon nanotubes," in *The 2015 European Conference on Artificial Life.* The MIT Press, 2015.

[22] T. Kowaliw, "Measures of complexity for artificial embryogeny," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation.* ACM, 2008, pp. 843–850.

[23] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications.* Springer Science & Business Media, 2013.

[24] S. Nichele and G. Tufte, "Measuring phenotypic structural complexity of artificial cellular organisms," in *Innovations in Bio-inspired Computing and Applications.* Springer, 2014, pp. 23–35.

[25] M. Hartmann, P. K. Lehre, and P. C. Haddow, "Evolved digital circuits and genome complexity," in *Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on.* IEEE, 2005, pp. 79–86.

[26] P. K. Lehre and P. C. Haddow, "Developmental mappings and phenotypic complexity." in *IEEE Congress on Evolutionary Computation (1).* Citeseer, 2003, pp. 62–68.

[27] H. Zenil and E. Villarreal-Zapata, "Asymptotic behavior and ratios of complexity in cellular automata," *International Journal of Bifurcation and Chaos*, vol. 23, no. 09, p. 1350159, 2013.

[28] B. J. MacLennan, "A review of analog computing. technical report utcs-07-601," University of Tennessee, Knoxville, Tech. Rep., 2007.

[29] H. J. Bremermann, *Self-Organizing Systems-1962.* Spartan Books, 1962, ch. Optimization through Evolution and Recombination, pp. 93–106.

[30] O. R. Lykkebø, S. Harding, G. Tufte, and J. Miller, "Mecobo: A hardware and software platform for in materio evolution," in *Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014, Proceedings*, 2014, pp. 267–279.

[31] A. S. Foundation, "Apache thrift," accessed: 2016-06-07. [Online]. Available: https://thrift.apache.org/

[32] W. R. Ashby, *Design for a Brain, the origin of adaptive behaviour*. Chapman & Hall Ltd., 1960.

[33] L. von Bertalanffy, *General System Theory*. George Braziller, Inc., Revised edition, Fourth printing, 1973.

[34] D. Laketić, G. Tufte, O. R. Lykkebø, and S. Nichele, "An explanation of computation – collective electrodynamics in blobs of carbon nanotubes," in *EAI, BICT 2015, 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. ACM Digital Library, 2015, pp. 1–6.

[35] O. R. Lykkebø, S. Nichele, and G. Tufte, "An investigation of square waves for evolution in carbon nanotubes material," *Proceedings of the European Conference on Artificial Life*, pp. 503–510, 2015.

[36] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

Paper B

Reservoir Computing with a Chaotic Circuit
(Jensen and Tufte, 2017)

Author(s):
Johannes H. Jensen and Gunnar Tufte

Published at conference:
ECAL 2017: The Fourteenth European Conference on Artificial Life

Paper B
Jensen and Tufte (2017)

Reservoir Computing with a Chaotic Circuit

Johannes H. Jensen, Gunnar Tufte

Department of Computer Science,
Norwegian University of Science and Technology, Trondheim, Norway

Abstract

Reservoir Computing has been highlighted as a promising methodology to perform computation in dynamical systems. This makes Reservoir Computing particularly interesting for exploiting physical systems directly as computing substrates, where the computation happens "for free" in the rich physical domain. In this work we consider a simple chaotic circuit as a reservoir: the Driven Chua's circuit. Its rich variety of available dynamics makes it versatile as a reservoir. At the same time, its simplicity offers insight into what physical properties can be useful for computation. We demonstrate both through simulation and in-circuit experiments, that such a simple circuit can be readily exploited for computation. Our results show excellent performance on two non-temporal tasks. The fact that such a simple nonlinear circuit can be used, suggests that a wide variety of physical systems can be viewed as potential reservoirs.

1 Introduction

Most dynamical systems in nature are nonlinear (Strogatz, 2015). An abundance of these systems show complex dynamic behavior, giving rise to phenomena such as self-organization, robustness, adaptivity, learning and intelligence (Mainzer, 2007).

It is argued that many natural systems perform some form of *intrinsic computation*. Neural systems and self-organizing cellular structures are examples

of large networks of simple electrochemical nodes that together process large amounts of information in support of the organism (Toffoli, 2004).

Transforming such intrinsic information processing to the artificial realm has been a key topic towards reproducing lifelike systems. Dynamics in the brain and neural systems has been targeted from the early days of cybernetics (Wiener, 1961; Ashby, 1960) to today's ongoing Human Brain Project (Markram, 2012). The processes of cellular communication also includes dynamic behavior that can be exploited toward mimicking natural information processing in artifacts, e.g. self-replication of structure (Langton, 1984).

An often neglected aspect about these phenomena is that they occur in *physical* systems. Through millions of years, computation has evolved bottom-up with electrical and chemical mechanisms as the basic building blocks. Evolution has discovered ways of doing computation by exploration and exploitation of the natural processes available in the physical substrate. Can we similarly exploit the underlying physics of matter to perform computation with such desirable properties as self-organization, robustness, vast parallelism and adaptivity?

To be successful, we need to consider computation both from a physical and a dynamical systems perspective. What types of dynamical/physical systems are suitable for computation? How do we "program" such systems (what are the inputs and outputs)? How should we define computation in terms of fixed points, attractors and trajectories? What's the role of bifurcations and chaos? (Stepney, 2012)

Finding natural models of computation would pave the way for exploiting physical systems directly for information processing. Such devices would be highly efficient, since computation happens "for free" directly in the substrate as the result of intrinsic physical properties. Complex materials with a vast number of interconnected elements could be exploited for immense parallel processing at the nanoscale. *Evolution in-materio* has indeed shown that computation can be evolved in physical matter (Miller et al., 2014).

Reservoir Computing (RC) has emerged as a promising technique for exploiting a dynamical system (the "reservoir") for computation. It is difficult to know how the output of a dynamical system should be interpreted to effectively perform useful computation. RC offers a flexible solution to this problem by utilizing a *readout* layer that is trained to produce some desired function as a linear combination of reservoir states. By virtue of its complex

dynamics, the reservoir provides a rich repertoire of nonlinear transformations that can be utilized by the readout.

The study of nonlinear dynamics has shown that surprisingly simple systems can exhibit complex behavior. In this work, we present one of the simplest physical reservoirs: a chaotic circuit. We show both through simulations and in-circuit experiments that the rich dynamics of the circuit can be exploited for solving computational tasks.

Typically, reservoirs are complex systems with many state variables. However, even low-dimensional systems can be used effectively as reservoirs. Appeltant et al. (2011) show that by multiplexing output in time rather than space, a single nonlinear node can act as a virtually high dimensional system.

RC's primary success story has been processing temporal (time-dependent) signals, where the reservoir serves as a nonlinear memory. Appeltant et al. (2011) employ delayed feedback to provide the nonlinear node with memory of past states. Our reservoir is conceptually simpler in that it does not employ delayed feedback, and consequently has very limited memory.

It is often argued that a reservoir performs optimally when its dynamics lie on the "edge of chaos" (Bertschinger and Natschläger, 2004). However, a chaotic reservoir can still be used as long as the input is sufficiently large to drive its dynamics out of the chaotic regime (Ozturk and Principe, 2005).

For non-temporal tasks, a chaotic system can be more readily exploited as long as it can be reset between inputs. Goh and Crook (2007) demonstrate how the transients of the Lorenz attractor can be used for pattern recognition. All transients will eventually diverge in a chaotic attractor, effectively separating all inputs. However, transients will remain similar for an initial period of time. By selecting at which point(s) in time the transients are observed, the sensitivity of the classification can be adjusted. This selection, of course, can be done automatically by the readout.

This paper is organized as follows: we begin by discussing the relevant background theory and related work. Next, we describe the chaotic circuit and our approach for using it as a reservoir. We then explain our experimental setup, followed by results and discussion. Finally we conclude with a discussion of future work.

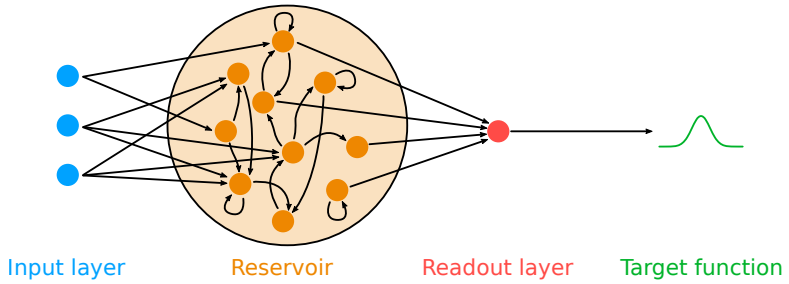


Figure 1: Reservoir Computing architecture.

2 Background

Inspired by the wide array of intrinsic physical computation found in nature, *Evolution in-materio* takes a bottom-up approach to evolve computation in physical matter (Miller et al., 2014). These efforts have been rather successful and a variety of computational devices have been evolved, e.g. a tone discriminator and robot controller in liquid crystal (Harding and Miller, 2004, 2005) and logic gates evolved in carbon nanotubes (Lykkebø et al., 2014; Massey et al., 2015) and gold nanoparticles (Bose et al., 2015).

Stepney (2008) argues that the time is ripe to climb the "neglected pillar of material computation". To be successful we must find computational models that are natural for the physical substrate. Stepney further argues that model-free evolutionary search provides limited insight into what (physical) mechanisms are being exploited and for what reason.

Links between dynamical systems and information theory were established over half a century ago when Shannon entropy was used to describe uncertainty in nonlinear dynamics (see Crutchfield et al. (2010) for a historical discussion).

Shaw (1981) argues that attractors act as information sinks: in the joining of trajectories, information about the state history of the system is lost. Chaos conversely acts as an information source: divergence of trajectories brings into view new information not present in the initial conditions.

It has been suggested that chaos plays an important role in natural systems by providing a rich repertoire of dynamics that may be utilized for increased performance (Sinha and Ditto, 1998).

Reservoir Computing has its roots in neural network research, where it was discovered that the rich dynamics and memory capabilities of recurrent neural networks (RNNs) could be exploited without any training of the network. Good performance could be obtained with a random network coupled with a linear readout layer trained on the activations of the RNN nodes (Jaeger, 2001; Maass et al., 2002).

Figure 1 shows the typical RC architecture with three distinct parts: the input layer, the reservoir with its many recurrent nodes, and the readout layer. Note that the readout is the only trained part, both the input layer and the reservoir remain unchanged.

Formally, the reservoir transforms a low-dimensional time-dependent input $u(t)$ into a high-dimensional state vector $\boldsymbol{x}(t)$ which is more readily processed by the linear readout. The reservoir acts as a nonlinear kernel with memory, maintaining a rich, high dimensional, nonlinear transformation of input history.

RNN reservoirs have outperformed state of the art methods in a wide range of challenging temporal tasks such as speech recognition and time series prediction. For an overview of RC methods, see Lukoševičius and Jaeger (2009).

The reservoir can be any kind of dynamical system, as long as it can be perturbed by input and its output observed. This has inspired a diverse range of physical systems used as reservoirs. Examples include an optoelectronic system (Paquot et al., 2012), a photonics chip (Vandoorne et al., 2014), nanoscale switch networks (Sillin et al., 2013), the bacterium Escherichia coli (Jones et al., 2007) and even a bucket of water (Fernando and Sojakka, 2003).

3 System description

In this work we consider a simple nonlinear circuit as a reservoir. We have chosen the Driven Chua's circuit introduced by Murali et al. (1994a,b) since it is one of the simplest circuits with a rich variety of dynamics. The circuit (Figure 2) consists of only a handful of components: three linear elements (a resistor, an inductor and a capacitor) and a nonlinear resistor (a Chua's diode). An external periodic forcing signal $f(t)$ drives the dynamics of the circuit.

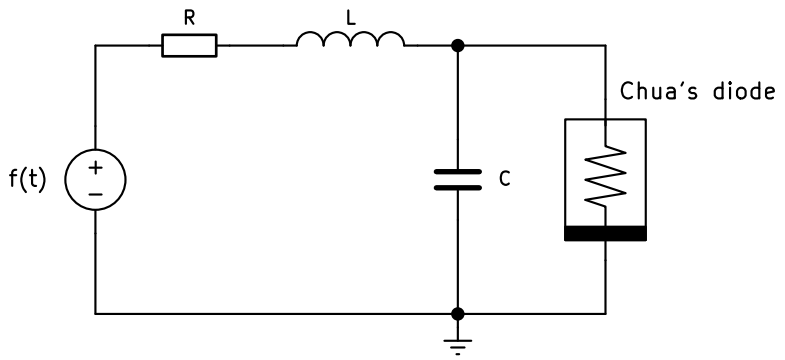


Figure 2: Driven Chua's circuit, adapted from Murali et al. (1994a)

The circuit is described by the following set of normalized differential equations:

$$\begin{aligned} \dot{x} &= y - g(x) \\ \dot{y} &= -\beta y - \beta x + F f(\omega t) \\ g(x) &= bx + 0.5(a-b)[|x+1| - |x-1|] \end{aligned} \tag{1}$$

where x corresponds to the voltage across the capacitor C and y corresponds to the current through the inductor L. The term $Ff(\omega t)$ is the external forcing signal with amplitude F and angular frequency ω. Note that in the original paper, the periodic driving force was sinusoidal, i.e. $f(t) = sin(t)$. Here we generalize the forcing term to include any type of periodic function.

$g(x)$ is the equation for the Chua's diode which has a piecewise-linear current-voltage characteristic as shown in Figure 3. The three linear regions have slopes a and b as shown with breakpoints at ± 1.

The dynamics of the circuit depends on the parameters β, a, b, ω and F. Figure 4 shows the bifurcation diagram as we increase the amplitude F of the forcing signal. Several interesting phenomena can be observed, such as period-doubling bifurcations, chaos and periodic windows.

For the current study we consider the case where the slopes of the Chua's diode are in the range $a < -1$ and $-1 < b < 0$. In the absence of the external forcing ($F = 0$), the system has three fixed points in this case: an unstable fixed point at the origin and two stable fixed points P^+ and P^-.

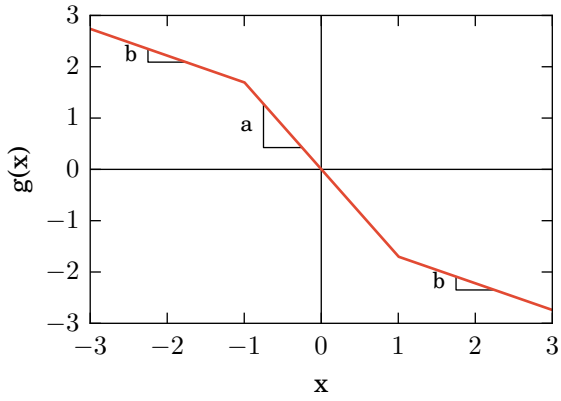


Figure 3: Current-voltage characteristic of Chua's diode. Given the (normalized) voltage x across the diode, the current through the diode is given by $g(x)$. The characteristic slopes a and b from equation (1) are also depicted.

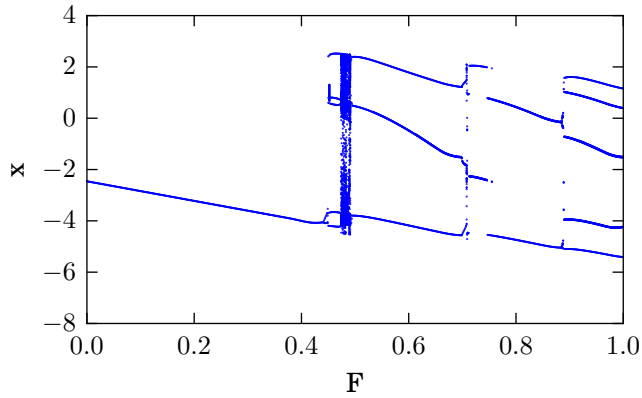


Figure 4: Bifurcation diagram as the forcing amplitude F is increased from 0.0 to 1.0. The system shown has $\beta = 1.0$, $a = -1.70$ $b = -0.52$ and $\omega = 0.7$.

For the first part of this work we simulate the system numerically using the normalized version of the equations. When implementing the circuit physically we translate back to the corresponding circuit equations.

3.1 Input/output encoding

Before any dynamical system can be used as a reservoir, we need to decide how to perturb the system with input (the input encoding), and how to observe the corresponding response (the output decoding). Since we also wish to realize the circuit experimentally, we must keep in mind the physical constraints of the system, as well as the limitations of electrical components and test equipment.

First we tackle the question of input encoding: where in our system do we apply the input signal $u(t)$ and how should this signal be encoded? Note that the focus here is on non-temporal tasks, where the input u does not depend on time.

A common approach is to apply input as the initial conditions (x_0, y_0) of the system. This is possible in simulation where we are free to choose initial conditions, however in a physical system this may not be the case. For instance, forcing a physical system towards an unstable fixed point may be impossible.

In this work, we instead apply input as part of the external forcing signal $Ff(\omega t)$. We map the input u to the amplitude F of the forcing signal in a linear range $[F_{min}, F_{max}]$, i.e. u is first normalized to the range $[0, 1]$ then $F = F_{min} + (F_{max} - F_{min})u$. Instead of sinusoidal forcing, we use square waves since they are more easily generated with our test equipment, i.e. $f(t) = sgn(sin(t))$. For each input u, the system is perturbed for N periods of the forcing signal. Between each input the system is reset to start in the same stable fixed point.

The output of our reservoir is the (discretized) transients of the system during the fixed perturbation period. We can visualize the output as a bifurcation diagram of transients, as shown in Figure 5. With such an output mapping, the reservoir produces many nonlinear transformations as a function of the input F. Compared to the long-term dynamics of the same system (Figure 4), the transients produce a richer repertoire of nonlinear functions.

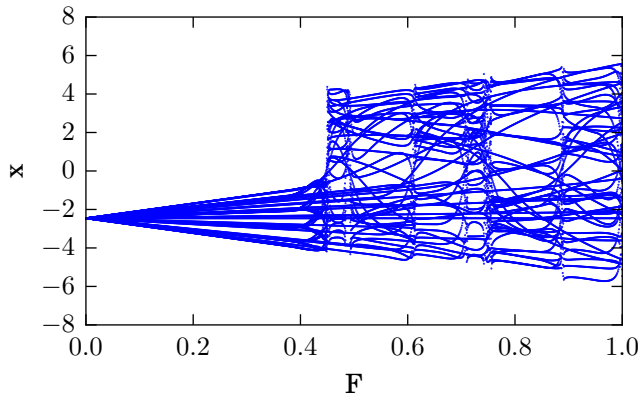


Figure 5: Bifurcation diagram of transients as the forcing amplitude F is increased from 0.0 to 1.0. The system shown has $\beta = 1.0$, $a = -1.70$, $b = -0.52$ and $\omega = 0.7$, sampled with $k = 10$ for $N = 5$ periods.

Specifically, we observe the variable $x(t)$ while the system is perturbed by input. We record $x(t)$ at a fixed sampling rate $k\omega$, always an integer multiple of the forcing frequency, to obtain kN discrete output samples from the system. The reservoir state vector is thus $\boldsymbol{x} = [x(0), x(\tau), x(2\tau), ..., x(kN\tau)]$ where $\tau = 2\pi/k\omega$ is the sample interval. Note that with $k = 1$ we get the Poincaré map.

4 Experimental setup

The dynamics of our reservoir depends on several parameters which will affect performance. For the current study we fix $\beta = 1.0$, $\omega = 0.7$ and vary the slopes in the intervals $a = (-2, -1)$ and $b = (-1, 0)$.

From Figure 5 we observe a linear region from $F = 0$ up to the first bifurcation point (here at $F = 0.45$). As a reservoir, such a linear region is uninteresting (since any linear function can be constructed by the readout layer alone) so we should set F_{min} past the first bifurcation point. The width of the linear region depends on the parameters a, b, and ω. Thus for a given set of these parameters, there exists a (problem-specific) optimal value for F_{min} and F_{max}. However, to reduce the number of parameters we need to explore, we experimentally fix $F_{min} = 0.5$ and $F_{max} = 1.0$.

We perturb the reservoir with $N = 5$ periods of square waves and record

$kN = 100$ output samples. Between each input, the system is reset to a stable fixed point, i.e. we set the initial condition $(x_0, y_0) = P^+$.

4.1 Tasks

We evaluate the performance of the reservoir on two non-temporal tasks: nonlinear regression and nonlinear classification.

The goal of the regression task is to approximate the 7th degree polynomial

$$y = (x-3)(x-2)(x-1)(x)(x+1)(x+2)(x+3)$$

in the range $(-3, 3)$. Such a smooth function was selected to test the reservoir's generalization capabilities and to investigate the effect of bifurcations and chaos on reservoir performance.

For classification we use the classical "circles" dataset with two classes organized in concentric rings (Ben-Hur et al., 2001). This is a simple dataset which requires a nonlinear decision surface. Being two-dimensional, it can be easily visualized which enables graphical analysis. We reset the reservoir between the application of each input feature, i.e. the x and y coordinates of each point are independently transformed by the reservoir. Furthermore, the number of samples was set to $kN = 50$ for each feature resulting in a total of 100 output samples.

For each of the tasks, 10 000 examples were generated of which 75% were used for training and 25% for validation. As performance metric we have used the normalized root mean square error (NRMSE) for regression and accuracy for classification.

Ridge regression was used for the readout layer, which is widely adopted within the RC community as it reduces overfitting thanks to a regularization term (Hoerl and Kennard, 1970).

4.2 Simulation experiments

First we evaluate reservoir performance in simulation, i.e. the governing equations (1) were integrated numerically.

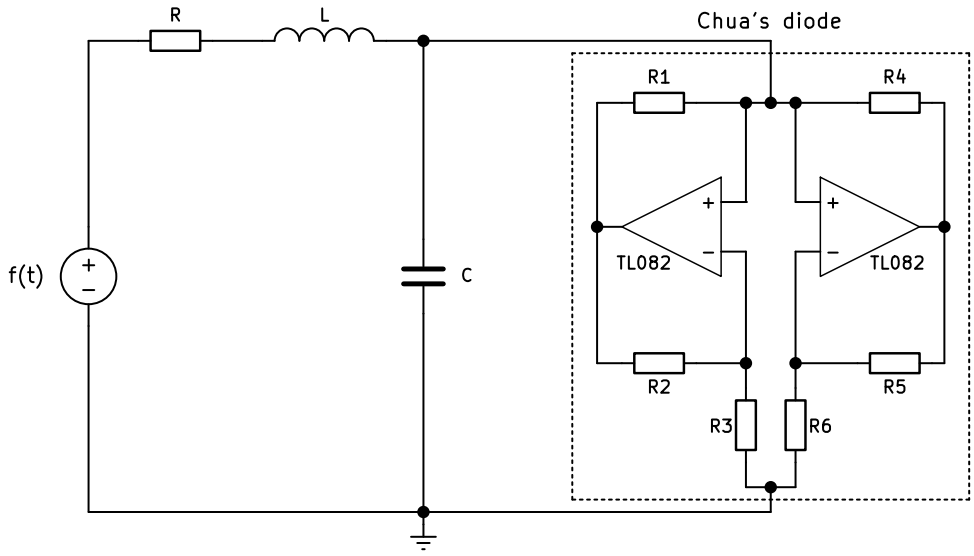


Figure 6: Driven Chua's circuit implemented using an active version of Chua's diode based on two op-amps.

We sweep the slopes of the Chua's diode in the intervals $a = (-2, -1)$, $b = (-1, 0)$ and evaluate performance on the regression task. We use ten-fold cross validation on the training set to evaluate reservoir performance.

We then have a closer look at the best performing reservoir by analysing the approximated polynomial on the validation set.

Next we apply the same reservoir on classification, to demonstrate that the reservoir can be re-used for a different task.

4.3 Circuit experiments

Based on the sweeps from simulation, we select a good performing reservoir which we implement on a printed circuit board. We use high-quality components with low tolerances whenever possible. Some components are not readily available with low tolerances (such as the inductor L), so these must be measured.

Figure 6 shows our circuit implementation. A passive version of Chua's diode doesn't exist, but an active version can be implemented using two op-amps

(Kennedy, 1992) which is what we have done here. This implementation allows the slopes to be set by the choice of resistors $R_1 - R_6$.

Given the normalized set of parameters $\beta = 1.0$, $a = -1.70$, $b = -0.52$ and $\omega = 0.7$ (the best performing reservoir from the sweeps), we calculate values for the circuit components as follows: select $L \approx 14mH$ (measured) and $C = 10nF$. Calculate $R = \sqrt{\beta L/C} = 1185\Omega$ and determine the frequency of the forcing signal $\Omega = \omega/RC \approx 9.4kHz$. The required slopes of the Chua's diode are then $G_a = Ga \approx -1.435mS$ and $G_b = Gb \approx -0.439mS$ where $G = 1/R$.

For Chua's diode we adjust the breakpoints $B_p = \pm 0.8V$ so that the dynamics of the circuit stays within the range of our ADC. Adjustable positive and negative power supplies of the op-amps allows fine-tuning these breakpoints.

The slopes of the Chua's diode are determined by resistors $R_1 - R_6$. Following the design procedure in Kennedy (1992), we set $R_1 = R_2 = 100\Omega$, $R_3 = E_{sat}/((B_p - E_{sat})G_b - B_pG_a) \approx 1915\Omega$, $R_4 = R_5 = E_{sat}/(B_p(G_b - G_a)) \approx 10052\Omega$ and $R_6 = E_{sat}/((E_{sat} - B_p)(G_b - G_a)) \approx 1116\Omega$ where $E_{sat} = 8.0V$ is the saturation level of the op-amps.

After application of input, the circuit may end up in either of the two stable fixed points P^+ and P^-. To make sure the circuit starts in the fixed point P^+ before each input, we use the following reset procedure: first a constant positive voltage $F > -a - 1$ is applied at the input $f(t)$. This has the effect of destroying the other two fixed points, leaving a single fixed point close to P^+ which the system will approach. Next, the constant voltage is removed ($F = 0$), causing the the system to return to the nearest fixed point P^+ as desired. The duration of the reset period must be sufficiently large ($\gg 1/\Omega$) to allow transients to settle.

To interface with the circuit we use the Mecobo platform (Lykkebø et al., 2014). The board can generate analog voltage signals using an onboard DAC and record analog voltages with an onboard ADC. For our experiments we use a 12-bit DAC (AD5308) with range set to $\pm 1.024V$ and a 13-bit ADC (AD7327) with a range of $\pm 5V$.

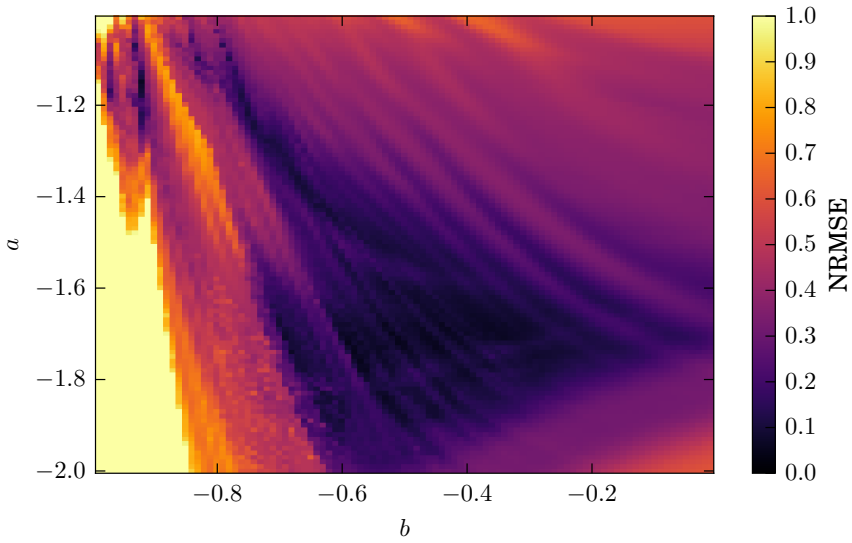


Figure 7: Score on the regression task as a function of the slopes a and b. Mean NRMSE from ten-fold CV is shown, where values have been capped to 1.0.

5 Results

5.1 Regression

Figure 7 shows the mean NRMSE of the reservoir plotted as a function of the slopes a and b. There is a fairly large region of parameter space with NRMSE below 0.1. The best performing reservoir has slopes $a = -1.70, b = -0.52$ and a NRMSE of $0.05(\pm 0.02)$. We can also see that performance is fairly smooth as a function of the two parameters.

Figure 8a shows the approximated polynomial on the validation set obtained with the best performing reservoir. For a majority of the input range, the approximation is almost perfect. There is however two noisy regions at around $x = -0.75$ and $x = 1.45$. The NRMSE score on the validation set is 0.07.

With the circuit reservoir we obtain an NRMSE score of 0.16 on the validation set (Figure 8b). Although the overall shape of the approximation resembles the desired polynomial, there is a fair amount of noise present. Note that no filtering has been performed on the sampled voltage data.

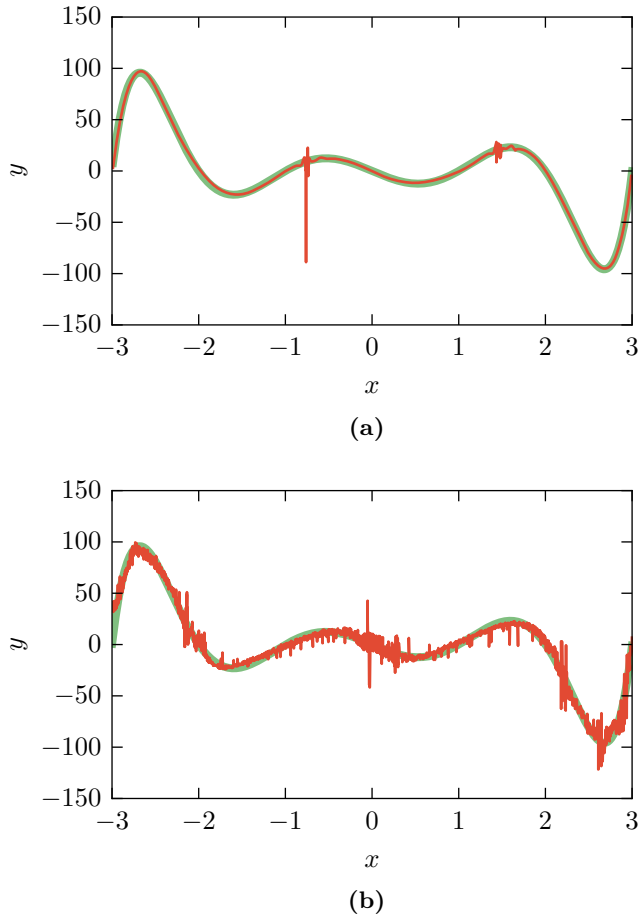


Figure 8: Approximated polynomial (red line) on validation set: (a) simulation NRMSE=0.07 (b) circuit NRMSE=0.16. The target polynomial is shown in green.

5.2 Classification

On the classification task, the simulated reservoir obtains perfect accuracy. Figure 9a shows the decision surface for the two classes, with the validation data superimposed. As can be seen, the decision surface is smooth and there is a sizeable margin separating the two classes.

Figure 9b shows the classification results with the circuit reservoir. Excellent performance is obtained with only one misclassified point (99.9% accuracy). However, the decision surface is markedly more noisy compared to simulation.

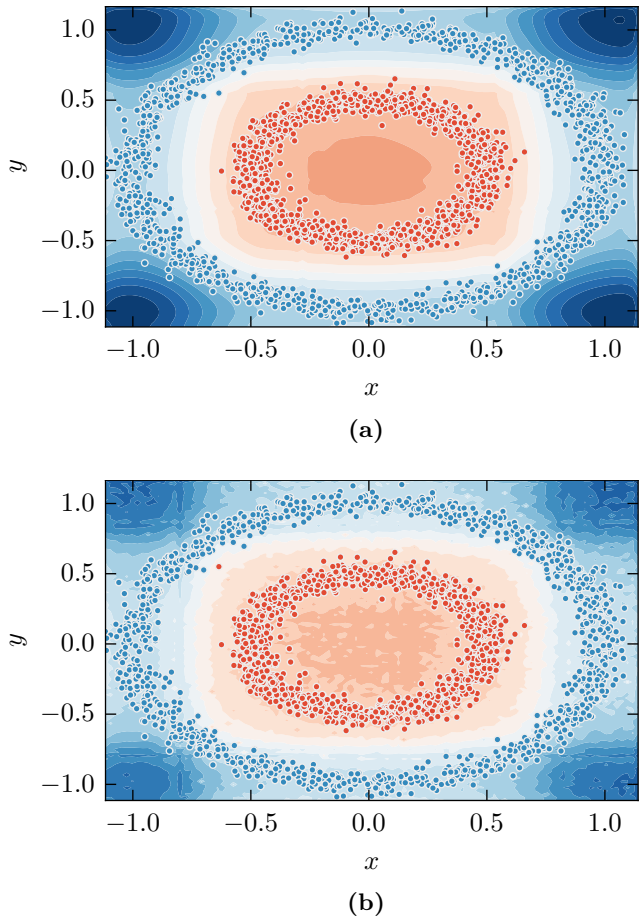


Figure 9: Decision surface and classification result on validation set: (a) simulation 100% accuracy (b) circuit 99.9% accuracy.

6 Discussion

Our results demonstrate that a chaotic circuit can be exploited for reliable computation within a reservoir computing framework.

The a and b parameter sweeps revealed a sizeable region of parameter space where good performance was obtained. Furthermore, the performance landscape has many smooth features, which looks promising for further exploration of the parameter space with stochastic search methods.

The sweep was performed with fixed values for F_{min} and F_{max} which, as

highlighted earlier, may be a suboptimal choice depending on the slopes. It is likely that the overall performance could be improved quite substantially if F_{min} and F_{max} was tuned individually for each pair of slopes.

Although good results were obtained on the polynomial task, there were two noisy regions in the approximation. They were also present on results from the training set, although not as pronounced. These regions are likely caused by instability close to bifurcation points, where chaotic dynamics dominate even the initial transients. The locations of the noisy regions ($x = -0.75$ and $x = 1.45$) correspond to $F \approx 0.69$ and $F \approx 0.87$ with our input encoding. This is very close to two narrow chaotic bands which can be seen in the bifurcation diagram (Figure 4).

For the classification task, near-perfect performance was obtained with a very smooth decision surface. This result demonstrates that the same reservoir can be re-used for two quite different tasks, by re-training the readout layer only. This is particularly relevant for physical reservoirs whose properties cannot easily be changed to suit a particular task.

Our experimental results with the circuit implementation of the reservoir revealed comparable performance to that of simulation. However, there is clear performance degradation caused by noise, especially on the regression task. Classification seems more robust to this noise where we can get away with a rather rugged decision surface.

We can attribute much of the noise due to sampling errors which will be particularly large in regions of the signal with steep slopes. Although noise will always be a problem when dealing with a physical system, it can have an even more pronounced effect in sensitive chaotic systems.

Finally, the dynamics of our circuit implementation did deviate somewhat from simulation, likely due to nonidealities in the circuit components. However, good performance was still obtained, which illustrates the robustness and power of the Reservoir Computing methodology.

7 Conclusion

In this work we have shown that a simple chaotic circuit can be used effectively as reservoir. We demonstrate that its rich dynamics can be exploited

directly for computation, both in simulation and with a circuit implementation. To the best of our knowledge, this is the first time a chaotic circuit has been used as a reservoir.

Here we have restricted the scope to non-temporal tasks. Future work will explore ways in which the chaotic circuit can be used to process temporal signals as well. Using the reservoir for more difficult tasks should also be attempted, e.g. with real-world datasets that contain more noise and/or require a more complex decision surface.

Different input/output encoding schemes should be investigated to obtain a richer repertoire of nonlinear transformations. Further exploration of parameter space through e.g. evolutionary search is likely to find better performing reservoirs.

To further investigate reservoirs within the same family, i.e. with a single nonlinear node, the Chua's diode could be replaced with other nonlinear elements. Memristive devices could potentially serve as nonlinear memory, making the circuit applicable to tasks such as speech recognition.

Our chaotic circuit can be viewed as an electrical analogy of any physical system with similar dynamic properties. The methods and results presented herein should therefore be transferable to any natural system that can be manipulated to behave within the desired dynamic regime. Given that such a simple physical system can be exploited, a wide variety of natural systems can be viewed as potential reservoirs. The grand goal of exploiting intrinsic properties of matter for computation seems within reach, paving the way towards highly efficient computation at the nanoscale.

References

Appeltant, L., Soriano, M., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C., and Fischer, I. (2011). Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468.

Ashby, W. R. (1960). *Design for a Brain: The Origin of Adaptive Behavior*. Chapman & Hall, London, England.

Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research*, 2:125–137.

Bertschinger, N. and Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural computation*, 16(7):1413–1436.

Bose, S. K., Lawrence, C. P., Liu, Z., Makarenko, K. S., van Damme, R. M. J., Broersma, H. J., and van der Wiel, W. G. (2015). Evolution of a designless nanoparticle network into reconfigurable Boolean logic. *Nature Nanotechnology*, 10(12):1048–1052.

Crutchfield, J. P., Ditto, W. L., and Sinha, S. (2010). Introduction to focus issue: Intrinsic and designed computation: Information processing in dynamical systems-beyond the digital hegemony. *Chaos*, 20(3).

Fernando, C. and Sojakka, S. (2003). Pattern Recognition in a Bucket. *Advances in Artificial Life*, 2801(12):588–597.

Goh, W. J. and Crook, N. (2007). Pattern Recognition using Chaotic Transients. In *ESANN*, number April, pages 25–27.

Harding, S. and Miller, J. (2005). Evolution In Materio : A Real-Time Robot Controller in Liquid Crystal. In *2005 NASA/DoD Conference on Evolvable Hardware (EH'05)*, number January, pages 229–238. IEEE.

Harding, S. and Miller, J. F. (2004). Evolution in materio: A tone discriminator in liquid crystal. *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, 2:1800–1807.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Bias Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. Technical Report 148, German National Research Center for Information Technology.

Jones, B., Stekel, D., Rowe, J., and Fernando, C. (2007). Is there a Liquid State Machine in the Bacterium Escherichia Coli? *2007 IEEE Symposium on Artificial Life*, pages 187–191.

Kennedy, M. P. (1992). Robust OP Amp Realization of Chua's Circuit. *Frequenz*, 46(3-4):66–80.

Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):135–144.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Lykkebø, O. R., Harding, S., Tufte, G., and Miller, J. F. (2014). Mecobo: A Hardware and Software Platform for In Materio Evolution. In Ibarra, O. H., Kari, L., and Kopecki, S., editors, *Lecture Notes in Computer Science 2014*, volume 8553 of *Lecture Notes in Computer Science*, pages 267–279. Springer International Publishing, Cham.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560.

Mainzer, K. (2007). *Thinking in Complexity*. Springer Berlin Heidelberg, Berlin, Heidelberg, 5 edition.

Markram, H. (2012). The Human Brain Project. *Scientific American*, 306(6):50–55.

Massey, M. K., Kotsialos, A., Qaiser, F., Zeze, D. A., Pearson, C., Volpati, D., Bowen, L., and Petty, M. C. (2015). Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903.

Miller, J. F., Harding, S. L., and Tufte, G. (2014). Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence*, 7(1):49–67.

Murali, K., Lakshmanan, M., and Chua, L. (1994a). The simplest dissipative nonautonomous chaotic circuit. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(6):462–463.

Murali, K., Lakshmanan, M., and Chua, L. (1994b). BIFURCATION AND CHAOS IN THE SIMPLEST DISSIPATIVE NON-AUTONOMOUS CIRCUIT. *International Journal of Bifurcation and Chaos*, 04(06):1511–1524.

Ozturk, M. C. and Principe, J. C. (2005). Computing with transiently stable states. *Proceedings of the International Joint Conference on Neural Networks*, 3:1467–1472.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., and Massar, S. (2012). Optoelectronic Reservoir Computing. *Scientific Reports*, 2:1–6.

Shaw, R. (1981). Strange Attractors, Chaotic Behavior, and Information Flow. *Zeitschrift für Naturforschung A*, 36(1):81–87.

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., and Gimzewski, J. K. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, 24(38):384004.

Sinha, S. and Ditto, W. (1998). Dynamics Based Computation. *Physical Review Letters*, 81(1):2156–2159.

Stepney, S. (2008). The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164.

Stepney, S. (2012). *Nonclassical Computation — A Dynamical Systems Perspective*. Springer Berlin Heidelberg.

Strogatz, S. H. (2015). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Steven H. Strogatz)*. Westview Press, 2 edition.

Toffoli, T. (2004). Nothing makes sense except in light of evolution. *International Journal of Unconventional Computing*, 1:3–29.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., Schrauwen, B., Dambre, J., and Bienstman, P. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5:3541.

Wiener, N. (1961). *Cybernetics: or Control and Communication in the Animal and the Machine, Second Edition*. MIT Press, Cambridge.

Paper C

Computation in artificial spin ice

(Jensen et al., 2018)

Author(s):
Johannes H. Jensen, Erik Folven and Gunnar Tufte

Published at conference:
ALIFE 2018: The 2018 Conference on Artificial Life

Paper C
Jensen et al. (2018)

Computation in artificial spin ice

Johannes H. Jensen[1], Erik Folven[2] and Gunnar Tufte[1]

[1]Department of Computer Science,
[2]Department of Electronics and Telecommunications,
Norwegian University of Science and Technology, Trondheim, Norway

Abstract

We explore artificial spin ice (ASI) as a substrate for *material computation*. ASI consists of large numbers of nanomagnets arranged in a 2D lattice. Local interactions between the magnets gives rise to a range of complex collective behavior. The ferromagnets form large networks of nonlinear nodes, which in many ways resemble artificial neural networks. In this work, we investigate key computational properties of ASI through micromagnetic simulations. Our nanomagnetic system exhibits a large number of reachable stable states and a wide range of available dynamics when perturbed by an external magnetic field. Furthermore, we find that the system is able to store and process temporal input patterns. The emergent behavior is highly tunable by varying the parameters of the external field. Our findings highlight ASI as a very promising substrate for *in-materio* computation at the nanoscale.

1 Introduction

Intelligent systems in nature are closely coupled to physics. Through bottom-up exploration and exploitation of physical processes, evolution has found ways to achieve self-organized computation. Such natural computation ultimately results in intelligent behavior (Mainzer, 2007). Furthermore, natural computation is extremely efficient: Our brain contains billions of processing elements (neurons) but consumes only 20W.

Artificial intelligent systems, e.g. artificial neural networks, are abstracted far away from the physical and chemical domain. As such, physical properties

are not exploitable for computation. Consequently, running these abstract models is inherently inefficient and requires massive server farms consuming megawatts of power.

Material computation places physics back in the front seat and views computation as an inherently physical process. Within a physical system (the material) lies an inherent capacity for computation. The goal is to find ways to exploit this inherent computing power. If accomplished, we can build extremely efficient computing devices based on principles found in biological systems: vast parallelism, self-organization, robustness and adaptation (Stepney, 2008).

The key principles of material computation date back to the early days of cybernetics (Ashby, 1960; Pask, 1959) and artificial life (Langton, 1990): a large number of nodes, nonlinearity, local interactions and rich dynamics. More recently the field of reservoir computing (Jaeger, 2001) has proven that complex physical systems with such properties can be readily exploited, by training a readout layer to map system output to the target problem (Dale et al., 2017; Jensen and Tufte, 2017; Sillin et al., 2013).

Hopfield (1982) was early to establish links between neural network models and physics. Hopfield networks are defined in terms of an energy function whose many local minima represent memorized states. Recently there has been an increased interest in energy-based neural network models since they are physically plausible learning architectures (Scellier and Bengio, 2017).

Hopfield energy provides a direct link between neuronal computation and ferromagnetic systems. The energy function is isomorphic with an Ising model where atomic spins take the place of neurons, while exchange coupling between spins is analogous to synaptic connections. Hopfield showed that important properties for computation spontaneously arise in such systems with many nonlinear nodes.

Artificial spin ice (ASI) is a class of ferromagnetic metamaterial which consists of large arrays of coupled nanomagnets. ASI exhibits key properties for material computation: a large number of nonlinear elements whose local interaction gives rise to complex collective behavior. Furthermore, the physical fabrication of ASI systems has been well-established over the last decade. It is thus an intriguing substrate for neuronal material computation.

In this work we investigate key properties for material computation in an ASI substrate. Through detailed micromagnetic simulations, we explore the range of dynamics available and the number of reachable states when ASI

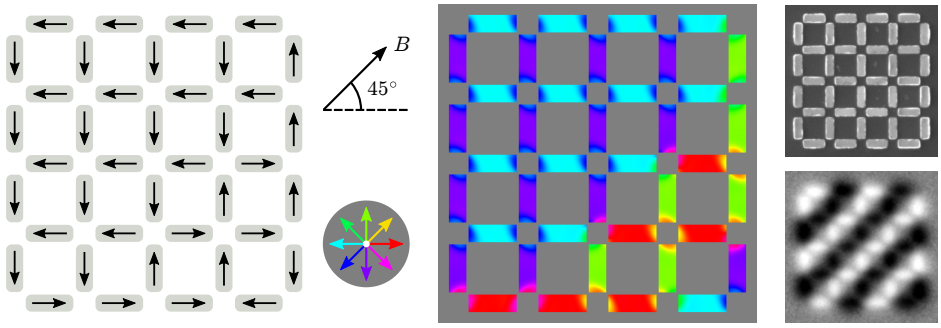


Figure 1: Square artificial spin ice consists of horizontal and vertical nanomagnets arranged in a 'square' pattern. For this work we study the 4x4 square spin ice which consists of 40 nanomagnets (a). Since the nanomagnets are single-domain, the internal magnetization will align left/right for horizontal magnets and up/down for vertical magnets (b). Colors indicate the direction of magnetization as shown in the color map. We perturb the array with an external magnetic field B at an angle of 45° as shown. Microscopic imaging of our fabricated spin ice (c) enables a view of the geometry (top) and magnetization (bottom) of each individual nanomagnet. The size of each magnet is 220x80 nm.

is subject to an oscillating magnetic field. Finally we demonstrate how the system can be exploited for temporal pattern classification.

This paper is organized as follows: we begin with a short introduction to the field of artificial spin ice. Next, we discuss how ASI relates to established models of computation. We introduce our methods and simulation experiments, followed by results and discussion. Finally we conclude our findings and discuss future work.

2 Artificial spin ice

Artificial spin ice is a class of metamaterial which consists of nanoscale ferromagnetic islands (nanomagnets) arranged in a 2D lattice. Figure 1a shows the spin ice system used in this study, which consists of 40 nanomagnets arranged in a 4x4 square lattice.

For nanomagnets smaller than some critical dimension, the individual spin moments will tend to align in the same direction, i.e. a *single-domain* state. When the magnets are elongated, the ground state magnetization will be

oriented homogeneously in either of the two directions defined by the long-axis of the magnet. Figure 1b shows how the internal magnetization of each nanomagnet is oriented in a single-domain ground state. Magnetization reversal, or *switching* in such nanoscale magnets is an inherently nonlinear process (Stoner and Wohlfarth, 1948).

Nanomagnets in the lattice will interact through local dipolar coupling. The interactions depend on the geometrical arrangement of the lattice. This makes it possible to create a geometrically frustrated system, i.e. a system in which all competing interactions cannot be satisfied at the same time. Many geometric patterns giving rise to geometrical frustration have been demonstrated. In this work we focus on the 'square' lattice as shown in Figure 1.

Originally, the field of ASI was developed to build nanotechnological model systems for fundamental studies of geometrical frustration (Wang et al., 2006). It has since grown to encompass a wide variety of phenomena ranging from effects of quenched disorder, thermally activated dynamics, microwave frequency responses, magnetotransport properties, and magnetic phase transitions (Marrows, 2016).

Owing to the semiconductor industry, the fabrication routes for artificial spin ice devices are already highly developed. With electron beam lithography, extended arrays consisting of millions of elements can readily be fabricated. The variety of geometrical arrangements that can be realized is only limited by the creativity of the designer and a plethora of different lattices have been created and explored in existing literature (Heyderman and Stamps, 2013).

The development of microscopy techniques with lateral resolution at the nanoscale and magnetic contrast enables direct imaging of the magnetization of individual nanomagnet elements (Figure 1c). At the macroscale, the collective state of the array can be observed using well-established magnetometry techniques. Reading techniques beyond the lab include magnetic tunnel junction based approaches similar to those found in Magnetic RAM and conventional hard drives.

There are many ways to manipulate the nanomagnet elements within an ASI. External magnetic fields is a well-established approach, applied either globally to the entire array or locally to specific areas, e.g. via current-carrying electrical wires. Other possibilities include current-induced torques (Brataas

et al., 2012), optically induced switching (Le Guyader et al., 2015) and using a scanning probe to manipulate individual nanomagnets (Gartside et al., 2018).

Recent developments of GPU accelerated simulation frameworks combined with increased computational resources have opened for simulation studies of extended nanomagnet ensembles, which greatly aids the exploration of novel artificial spin ice configurations.

3 Computation in artificial spin ice

ASI exhibits key properties for material computation, i.e. a large number of nodes, nonlinearity and local interactions. However, the question of how ASI relates to models of computation needs to be answered. Here we discuss the relation between ASI and one model of computation: artificial neural networks (ANNs).

ASI has a range of properties which are analogous to ANNs. Like neural networks, ASI consists of a large number of nonlinear nodes (nanomagnets) which are connected together in a network (dipolar coupling). Magnets are non-volatile devices, so each node exhibits long term memory. As with neural networks, computation in ASI is thus closely coupled with memory.

Network dynamics is the result of local interactions between neighboring magnets, much like recurrent neural networks. Geometry imposes certain limitations on the network topology, e.g. fully connected networks are not realizable in ASI. In this aspect ASI shares similarities with cellular neural networks (Chua and Yang, 1988).

A major difference between ASI and ANNs is the absence of intrinsic synaptic plasticity: the coupling between magnets is determined by their geometrical arrangement which is fixed. ASI is thus analogous to a network with fixed weights. The coupling strength can however be modified externally, e.g. through local magnetic fields. Reservoir computing has proven that even random fixed-weight networks can be exploited to solve many useful problems (Jaeger, 2001).

3.1 Microstates and macrostates

The state of a physical system can be understood in terms of hierarchies, i.e. the level of observation. We can distinguish between the *microstate* and the *macrostate*. The microstate is the state at the most detailed level, i.e. down to the smallest particle. Usually we do not have access to the microstate. Typically we only have access to the macrostate: the state of the system observed at some higher level. As observers, we are free to choose how and at what level we measure the macrostate.

In the case of ASI, the microstate is defined by the magnetic moments of the atoms within a nanomagnet. In simulation we can directly observe the microstate, something which is not possible in the real physical system. Figure 1b shows the microstate of our simulated 4x4 square spin ice.

Regarding the macrostate of ASI, a natural choice is to define a macrostate based on the average magnetization within a nanomagnet. For the single domain magnets used here, the magnetization will align in one of two directions, i.e. a *binary* macrostate. The macrostate of the entire array can thus be represented with N bits, where N is the number of nanomagnets. Figure 1a shows the macrostate derived from the corresponding microstate in Figure 1b. The macrostate as can be observed in the real physical system can be seen in the bottom of Figure 1c.

Macrostates are in general *degenerate*, i.e. there are many microstates which map to the same macrostate. Information about the true state of the system is therefore hidden. States that look identical at the macro level may in reality be different at the micro level.

3.2 Energy-based computation

Energy-based models such as Hopfield networks define computation as movement through an energy landscape. Each valley is a local energy minimum and represent stable ground states of the system. Given an arbitrary initial state, the system will settle into the nearest valley. If there are many such valleys, the system has a large number of stable states.

If we perturb the system with sufficient force (input), the system may escape the valley and transition to a nearby state. This movement through state space is a form of intrinsic physical computation. Computation is thus

closely linked to dynamics, i.e. complex dynamics results in complex computations (Langton, 1990).

When ASI is subject to a cyclic external magnetic field, complex switching behavior can occur. The behavior will depend critically on the strength of the magnetic field. If the field is too weak, there is not enough energy to leave the local energy minimum so no switching occurs. If the field is too strong, all the magnets will simply follow the field and we get trivial switching behavior. However, when the field strength is at some critical value, the local magnetic fields around each magnet will affect the switching of neighboring magnets. At this critical field strength we may find complex dynamics. Gilbert et al. (2015) demonstrated experimentally that for slowly varying fields whose strength is just above the array coercivity, an ASI will go through several transient states before settling in a stable ground state.

Little is however known about the switching dynamics of ASI at high frequencies. It seems likely that increasingly complex dynamics occur at higher frequencies where phenomena such as spin waves come into play. Indeed, studies of magnetic resonance has revealed complex frequency dependence of spin ice systems in the GHz range (Jungfleisch et al., 2016, 2017). Furthermore, a system of two interacting dipoles exhibits chaotic dynamics when subject to a time-dependent external magnetic field (Urzagasti et al., 2015).

4 Methods

We have argued that artificial spin ice exhibits key properties for material computation. Several properties are fulfilled by definition: a large number of nodes, nonlinearity and local interactions. However, the availability of complex dynamics within ASI is still largely unexplored. We need a reliable way to excite and control such complex dynamics.

Here we investigate magnetization dynamics when ASI is perturbed by an external magnetic field of high frequency. Next we demonstrate how such complex dynamics can be exploited for computation.

As subject of study we use the 4x4 square spin ice depicted in Figure 1 consisting of 40 permalloy nanomagnets. The size and spacing of the magnets is similar to previous studies (Wang et al., 2006): each magnet is 220x80 nm with a thickness of 25 nm, and the lattice spacing between each magnet is 320 nm.

We start with an initially polarized array such that all horizontal magnets point to the right and vertical magnets point up. The initial state is easily obtained by first saturating the array with a strong magnetic field at 45° which is then reduced to zero.

We perturb the array with a time-varying external magnetic field $B(t) = A\sin(\omega t)$ with amplitude A and frequency ω. The field is applied at a constant angle of 45° as shown in Figure 1.

The magnets in our experiments are single domain, so we adopt a binary macrostate based on the average magnetization. For horizontal magnets, let the state be 1 if the magnetization points to the right and 0 if it points to the left. Similarly for vertical magnets, the state is 1 if the magnetization points up and 0 if it points down. Since the array consists of 40 magnets, the macrostate of the entire array can be represented as a 40 bit vector. Hence the system has a state space of 2^{40} unique states.

All experiments were conducted using the MuMax3 micromagnetic simulator (Vansteenkiste et al., 2014). Key material parameters used are $M_{sat} = 860 \times 10^3$ A/m, $A_{ex} = 13 \times 10^{-12}$ J/m and $\alpha = 0.01$. A lateral cell size of 5x5 nm^2 was used throughout, which is less than the exchange length ($L_{ex} = 5.3$ nm).

5 Experiment 1: Complex dynamics

First we investigate the dynamics of the array when the sinusoidal external field has constant amplitude and frequency. In the following, we investigate the impact of field amplitude A and frequency ω on the dynamics of the system.

We perturb the array with 100 periods of the external field and sample the state of the system at the end of each period. We then count the number of unique states S visited during this time ($1 \leq S \leq 100$). A large number of unique states indicates complex dynamics while a low number indicates stability. A hallmark of chaos is aperiodic long-term behavior where state space trajectories never repeat, hence a large number of unique states is an indication of chaotic dynamics.

For weak fields, we expect $S = 1$ since none of the magnets will switch. For strong fields, we also expect $S = 1$ as all magnets will switch in unison with the field, and return back to the same polarized state after one period. Only

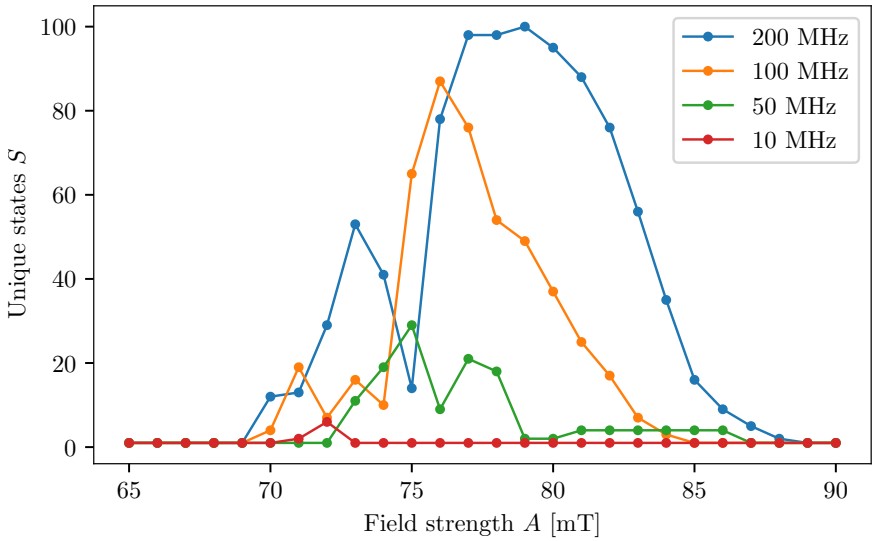


Figure 2: Number of unique states S visited as a function of the external field strength A, for four different frequencies.

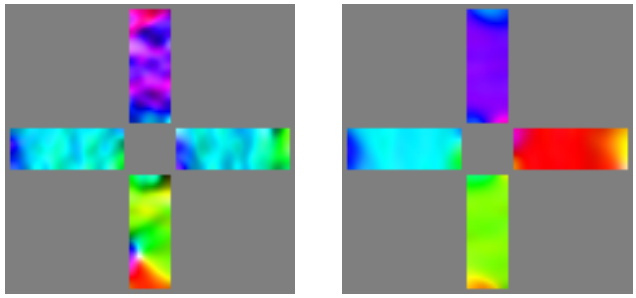
for intermediate field strengths close to the array coercivity H_c can we hope to find complex switching dynamics and consequently $S \gg 1$. For our system we estimate $H_c \approx 75$ mT measured at 45°.

We vary the field amplitude A around H_c and count the number of states for each A. This sweep is repeated for a set of frequencies.

5.1 Results and discussion

Figure 2 shows the number of unique states S as a function of the external field strength A. We plot the number of states for four different frequencies: 10 MHz, 50 MHz, 100 MHz and 200 MHz.

At 10 MHz we see at most 6 unique states which is in agreement with Gilbert et al. (2015). When the frequency is increased to 50 MHz, the number of states reaches a maximum of 29 for $A = 75$ mT. At 100 MHz there is a significant increase in number of states with a maximum of 87 for $A = 76$ mT. As we increase frequency further to 200 MHz, we observe a saturation in the number of states ($S = 100$) for $A = 79$ mT.



(a) (b)

Figure 3: Snapshots of the microstates for 200 MHz (a) and 100 MHz (b). The color map is identical to Figure 1.

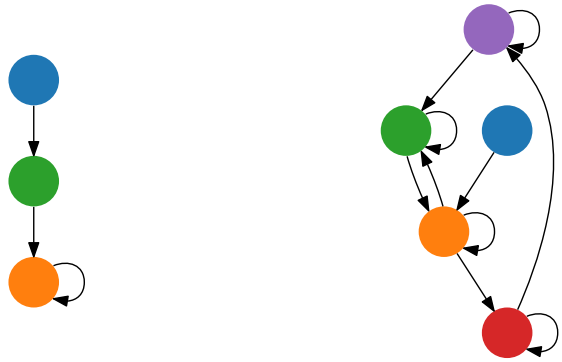
As expected, the bell-like curves are all centered around the array coercivity $H_c \approx 75$ mT. A general trend is that the number of unique states increases with frequency. The curves also become wider with increased frequency.

The large number of unique states seen at frequencies 100 MHz and above is the product of long transients. It is thus likely that dynamics are chaotic in these cases. The results also indicate that the dynamics get more chaotic as frequency is increased. Videos of the dynamics are available on our website[1].

These results demonstrate that ASI has a large number of *reachable* macrostates. Furthermore, the states can be reached by the straightforward application of a cyclic magnetic field, as long as the field strength is close to a critical value and the frequency is sufficiently high.

We get a better understanding of these results by examining the microstate. Figure 3 shows snapshots of the microstate taken at the end of the field cycle, i.e. at zero amplitude. Figure 3a shows the microstate at 200 MHz, where we can see that spin wave dynamics have not yet settled. These micro level dynamics become very turbulent if frequency is increased further to 1 GHz (not pictured). Hence at high frequencies, the spins within the nanomagnet have not yet aligned in one direction. In other words, the system has not reached a stable equilibrium and consequently the macrostate will be in flux. Figure 3b shows the microstate at 100 MHz, where we see that the spin waves have mostly settled at the end of the field cycle, resulting in a stable macrostate.

[1] `https://www.ntnu.edu/socrates/magnets`



(a) (b)

Figure 4: Example state transition graphs for 10 MHz (a) and 50 MHz (b).

What effect do micro level dynamics such as spin waves have on dynamics at the macro level? An interesting question is whether macro level dynamics are memoryless, i.e. does the next macrostate depend only on the current macrostate? We answer this with an analysis of the state transition graphs derived from the observed macro level dynamics (Figure 4). Here nodes represent macrostates and edges are the transitions between them.

At low frequencies (10 MHz) we find that all nodes have at most one outgoing edge, i.e. the dynamics are memoryless. An example is shown in Figure 4a where the system settles in a stable state after a transition through an intermediate state.

At higher frequencies (50 MHz and above) most graphs contain nodes with more than one outgoing edge, as shown in Figure 4b. Here the next macrostate can not be determined without knowing the history of previous macrostates. This is information which is hidden in the microstate.

This suggests that the microstate provides a means of information storage, which is indirectly observed in the history dependent dynamics at the macro level. Indeed it has been shown that chaos both generates and stores information (James et al., 2014).

In summary, these results demonstrate that rich micro level dynamics are available in the ASI system. Dynamics at the micro level give rise to phenomena at the macro level, i.e. a large number of distinct macrostates and complex state transition patterns. Crucially, the dynamics appear to be

highly tunable, e.g. by varying the amplitude and/or frequency of an external magnetic field.

6 Experiment 2: Temporal pattern classification

Results from the first experiment show that information about state history is embedded in the microstate, which suggests that the system may be able to store and process temporal input.

Let us assume that transitions between macrostates is sensitive not only to the history of past states, but also to the history of the external field. We can exploit this property for computation by encoding input as part of the time-varying external field.

For each input, does the system end up in a unique macrostate? If so, the system essentially acts as *memory*, transforming a temporal input to a unique spatial state. The function is that of maximum discrimination: each input is mapped to a unique output state. Hence, information in the input is preserved by the system.

On the other hand, if the system ends up in only a handful of states, the functionality is some form of *classification*. The input-output relations are many to one, i.e. many inputs map to the same output state. This mapping may be arbitrarily complex, but information about the input is always lost in the mapping process.

What about the case in-between these two extremes? Here the functionality is a mixture of memory and classification where some information is retained and some information is lost in the mapping.

Can we find these different modes of computation in square artificial spin ice? To test this, we consider as input bit strings of length $N = 1$ to $N = 8$. For each N we apply all possible 2^N inputs and record the final macrostate of the array. We then count the number of unique such states to determine what mode of computation is performed: memory, classification or a mixture of the two.

To encode the bit strings we employ amplitude modulation of the external magnetic field as shown in Figure 5. Each bit corresponds to one cycle of the external field, where 1 maps to an amplitude A_{hi} while 0 maps to an amplitude A_{lo}.

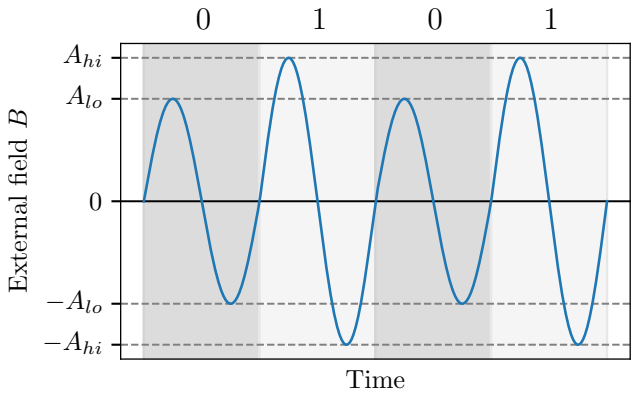


Figure 5: Bit strings are encoded into the external magnetic field with amplitude modulation. Each bit corresponds to one cycle of the external field, where 1 maps to an amplitude A_{hi} while 0 maps to an amplitude A_{lo}. The figure shows how the bit string 0101 is encoded.

We set the frequency of the external field to 100 MHz in order to obtain both complex dynamics and a stable macrostate at the end of each period. For the current study we fix $A_{lo} = 70$ mT and vary A_{hi} in the range where complex switching dynamics was found in the previous experiment, i.e. 70 mT $< A_{hi} \leq 84$ mT.

6.1 Results and discussion

Figure 6 shows the number of unique final states S as a function of the number of bits N in the input string. For comparison we plot the function 2^N which is the number of different input values and thus the theoretical maximum. We plot the number of unique states for a selection of A_{hi} values which resulted in distinct behavior, namely 76 mT, 79 mT, 81 mT and 84 mT.

A general trend for all the values of A_{hi} is that the number of states increases with the number of input bits. The rate of increase is however quite different for distinct values of A_{hi}. Another observation is that the number of states do not appear to saturate.

Recall from the first experiment that $A = 76$ mT was the amplitude which produced the highest number of states at 100 MHz. Here we see that this particular value for A_{hi} also results in the highest number of unique final

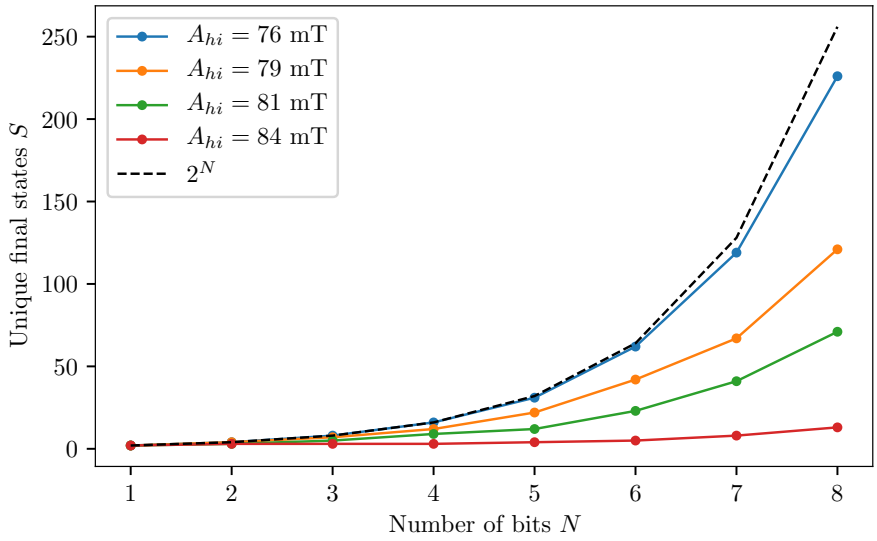


Figure 6: Number of unique final states S as a function of number of bits N in the input string (100 MHz input frequency).

states, closely tracking the theoretical maximum 2^N. In this regime the system acts primarily as memory, mapping almost all temporal input patterns to unique macrostates. At the maximum $N = 8$ the system maps the 256 different input patterns to 226 different states.

On the other extreme we have $A_{hi} = 84$ mT which results in only a handful of unique states. In this regime the system acts as a form of classifier. When the number of input bits is 8, the system maps the 256 different input patterns to 13 distinct states.

Interestingly, we find that certain values of A_{hi} result in functionality somewhere in between memory and classification. For $A_{hi} = 79$ mT we can see that the number of states follows roughly in the middle of the previous extremes, while for $A_{hi} = 81$ mT the state count is somewhat lower.

We may understand these modes of intrinsic computation in terms of dynamical systems theory. Classification is equivalent to entering an attractor, where the number of attractors is equal to the number of classes. Memory can be explained by chaos where sensitivity to initial conditions means that every distinct input will result in a unique trajectory through state space (Crook, 2007; Shaw, 1981).

The sensitivity of the system will determine how quickly nearby trajectories diverge and consequently whether similar input patterns will end up in the same macrostate after all the bits have been applied. By adjusting the field amplitude A_{hi} we are essentially tuning the sensitivity of the system, resulting in different modes of computation.

These results demonstrate only one way to exploit ASI for computation with a temporal input encoding. Here we are exploiting the rich repertoire of dynamics available in the system. Combined with the large number of stable states, the system is able to effectively distinguish between temporal patterns. Furthermore, we show that the sensitivity of discrimination is highly tunable which yields different modes of computation.

7 Conclusion and future work

In this work we have explored artificial spin ice (ASI) as a highly promising substrate for material computation. The ferromagnetic material exhibits key intrinsic properties for computation: a large number of nodes, nonlinearity, local interactions and rich dynamics. Fabrication methods for ASI are already highly developed, making such computational properties readily available *in-materio*.

Through micromagnetic simulations of square ASI, we have found a large number of reachable stable states and a wide range of available dynamics when ASI is perturbed by an oscillating magnetic field. Furthermore, the dynamics are highly tunable by varying the amplitude and frequency of the magnetic field.

We have also demonstrated how the complex dynamics and many stable states in ASI can be exploited for temporal pattern classification. By tuning the sensitivity of the system, different modes of computation can be obtained: memory, classification or a mixture of the two.

The results in this work have been derived from micromagnetic simulations. A natural next step is to replicate these results experimentally with physical realizations of the spin ice. As shown in Figure 1c, the physical parameters of the ASI are all well within limits of what can be fabricated and measured in our lab today.

Our experiments so far have been purely deterministic without noise. A key question is how the system behaves in the presence of noise, e.g. thermal

noise at room temperature or electrical noise from measurement equipment. Results indicate that the system may be very sensitive to small changes, i.e. chaotic dynamics will be susceptible to noise. Hence, it is critical to obtain a measure of this sensitivity.

The ASI system used in this study has been rather small, consisting of only 40 magnets. However, scaling up to millions of magnets can easily be achieved in physical realizations. A key question is then how scale affects the behavior of larger arrays. Micromagnetic simulations of such large systems is not computationally feasible, hence it is crucial to establish simulation models at higher levels of abstraction.

The focus of this work has been on the basic computational properties of ASI. As such, we have not focused on any specific application in our experiments. Future work will include the application of techniques such as reservoir computing to exploit ASI for useful tasks. Reservoir computing in spin ice could enable robust, massively parallel magnetic processing at the nanoscale.

The work presented herein only scratches the surface of what is possible with ASI systems. There is a wealth of parameters worth exploring, e.g. magnet size, shape, spacing and geometry, together with methods for perturbation and observation of dynamics. Furthermore, developing ways to efficiently exploit the intrinsic computing power in such systems is critically important. The field of ASI computation is ripe for exploration, towards vastly parallel and energy efficient computing substrates.

Acknowledgements

This work was funded in part by the Norwegian Research Council under the SOCRATES project, grant number 270961.

References

Ashby, W. R. (1960). *Design for a Brain: The Origin of Adaptive Behavior*. Chapman & Hall, London, England.

Brataas, A., Kent, A. D., and Ohno, H. (2012). Current-induced torques in magnetic materials. *Nature Materials*, 11(5):372–381.

Chua, L. and Yang, L. (1988). Cellular neural networks: theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257–1272.

Crook, N. (2007). Nonlinear transient computation. *Neurocomputing*, 70(7-9):1167–1176.

Dale, M., Miller, J. F., and Stepney, S. (2017). Reservoir Computing as a Model for In-Materio Computing. In Adamatzky, A., editor, *Advances in Unconventional Computing. Emergence, Complexity and Computation, vol 22.*, chapter 22, pages 533–571. Springer, Cham.

Gartside, J. C., Arroo, D. M., Burn, D. M., Bemmer, V. L., Moskalenko, A., Cohen, L. F., and Branford, W. R. (2018). Realization of ground state in artificial kagome spin ice via topological defect-driven magnetic writing. *Nature Nanotechnology*, 13(1):53–58.

Gilbert, I., Chern, G.-W., Fore, B., Lao, Y., Zhang, S., Nisoli, C., and Schiffer, P. (2015). Direct visualization of memory effects in artificial spin ice. *Physical Review B*, 92(10):104417.

Heyderman, L. J. and Stamps, R. L. (2013). Artificial ferroic systems: novel functionality from structure, interactions and dynamics. *Journal of Physics: Condensed Matter*, 25(36):363201.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. Technical Report 148, German National Research Center for Information Technology.

James, R. G., Burke, K., and Crutchfield, J. P. (2014). Chaos forgets and remembers: Measuring information creation, destruction, and storage. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 378(30-31):2124–2127.

Jensen, J. H. and Tufte, G. (2017). Reservoir computing with a chaotic circuit. In *Proceedings of the 14th European Conference on Artificial Life ECAL 2017*, pages 222–229, Cambridge, MA. MIT Press.

Jungfleisch, M. B., Sklenar, J., Ding, J., Park, J., Pearson, J. E., Novosad, V., Schiffer, P., and Hoffmann, A. (2017). High-Frequency Dynamics Modulated by Collective Magnetization Reversal in Artificial Spin Ice. *Physical Review Applied*, 8(6):064026.

Jungfleisch, M. B., Zhang, W., Iacocca, E., Sklenar, J., Ding, J., Jiang, W., Zhang, S., Pearson, J. E., Novosad, V., Ketterson, J. B., Heinonen, O., and Hoffmann, A. (2016). Dynamic response of an artificial square spin ice. *Physical Review B*, 93(10):100401.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37.

Le Guyader, L., Savoini, M., El Moussaoui, S., Buzzi, M., Tsukamoto, A., Itoh, A., Kirilyuk, A., Rasing, T., Kimel, A. V., and Nolting, F. (2015). Nanoscale sub-100 picosecond all-optical magnetization switching in GdFeCo microstructures. *Nature Communications*, 6(1):5839.

Mainzer, K. (2007). *Thinking in Complexity*. Springer Berlin Heidelberg, Berlin, Heidelberg, 5 edition.

Marrows, C. (2016). Experimental Studies of Artificial Spin Ice. *arXiv preprint arXiv:1611.00744*.

Pask, G. (1959). Physical Analogues to the Growth of a Concept. In *Mechanization of Thought Processes, Symposium*, pages 877–922.

Scellier, B. and Bengio, Y. (2017). Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. *Frontiers in Computational Neuroscience*, 11:24.

Shaw, R. (1981). Strange Attractors, Chaotic Behavior, and Information Flow. *Zeitschrift für Naturforschung A*, 36(1):81–87.

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., and Gimzewski, J. K. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, 24(38):384004.

Stepney, S. (2008). The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164.

Stoner, E. C. and Wohlfarth, E. P. (1948). A Mechanism of Magnetic Hysteresis in Heterogeneous Alloys. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 240(826):599–642.

Urzagasti, D., Becerra-Alonso, D., Pérez, L. M., Mancini, H. L., and Laroze, D. (2015). Hyper-chaotic Magnetisation Dynamics of Two Interacting Dipoles. *Journal of Low Temperature Physics*, 181(5-6):211–222.

Vansteenkiste, A., Leliaert, J., Dvornik, M., Helsen, M., Garcia-Sanchez, F., and Van Waeyenberge, B. (2014). The design and verification of MuMax3. *AIP Advances*, 4(10):107133.

Wang, R. F., Nisoli, C., Freitas, R. S., Li, J., McConville, W., Cooley, B. J., Lund, M. S., Samarth, N., Leighton, C., Crespi, V. H., and Schiffer, P. (2006). Artificial 'spin ice' in a geometrically frustrated lattice of nanoscale ferromagnetic islands. *Nature*, 439(7074):303–306.

Paper D

flatspin: A Large-Scale Artificial Spin Ice Simulator

(Jensen et al., 2020)

Author(s):

Johannes H. Jensen, Anders Strømberg, Odd Rune Lykkebø, Arthur Penty, Magnus Själander, Erik Folven and Gunnar Tufte

In review:

preprint in arXiv:2002.11401 [cond-mat, physics:physics]

Paper D
Jensen et al. (2020)

flatspin: A Large-Scale Artificial Spin Ice Simulator

Johannes H. Jensen, Anders Strømberg, Odd Rune Lykkebø, Arthur Penty, Magnus Själander, Erik Folven, and Gunnar Tufte

Norwegian University of Science and Technology, Trondheim, Norway

Abstract

We present flatspin, a novel simulator for systems of interacting mesoscopic spins on a lattice, also known as artificial spin ice (ASI). Our magnetic switching criteria enables ASI dynamics to be captured in a dipole model. Through GPU acceleration, flatspin can simulate realistic dynamics of millions of magnets within practical time frames. We demonstrate flatspin's versatility through the reproduction of a diverse set of established experimental results from the literature. In particular, magnetization details of "pinwheel" ASI during field-driven reversal have been reproduced, for the first time, by a dipole model. The simulation framework enables quick exploration and investigation of new ASI geometries and properties at unprecedented speeds.

1 Introduction

An artificial spin ice (ASI) is an ensemble of nanomagnets arranged on a lattice, coupled through magnetic dipole-dipole interactions. The vast variety of emergent collective behaviors found in these systems have generated considerable research interest over the last decade [1, 2]. Using modern nanofabrication techniques, emergent phenomena can be facilitated through direct control of the ASI geometry, e.g., collective ferromagnetic/antiferromagnetic

ordering [3], Dirac strings [4], and phase transitions [5, 6]. ASIs offer a unique model system for exploring fundamental physics, since magnetic microscopy enables direct observation of their internal state. There is also a growing interest in ASIs as building blocks for novel devices [7].

Micromagnetic simulations of ASI have been limited to a handful of nanomagnets due to excessive computational cost. Although physically accurate, such high fidelity simulations are unable to capture large-scale emergent phenomena, such as the size of magnetically ordered domains and long-range order. To simulate large ASI systems, an established approach is to sacrifice fidelity for speed by employing a dipole model, i.e., treating each nanomagnet as a single macro spin approximated by a point dipole [8]. Traditionally, Monte Carlo methods have been used in conjunction with the dipole approximation to search for low energy configurations [9, 10] or study statistical measures such as vertex populations [8]. However, Monte Carlo methods are inherently stochastic and better suited for ensemble statistics rather than detailed dynamics [11].

flatspin is a simulator for large ASI systems based on a dipole approximation with the ability to capture realistic dynamics, inspired by the work of Budrikis [8]. The flexibility of flatspin enables quick exploration of ASI parameters and geometries. Through GPU acceleration, flatspin can capture realistic dynamics of millions of magnets within practical time frames.

In this paper, we present the motivation and design of flatspin. We demonstrate good agreement between flatspin and a variety of published experimental results. We show that flatspin can capture dynamic behaviors observed experimentally, which have previously eluded modeling [12].

2 The flatspin magnetic model

In this section, we describe the dipole model and the underlying physical assumptions of flatspin. The model is designed to simulate the ensemble state-by-state evolution, i.e., *dynamics*, of two-dimensional ASI. In short, magnets are modeled as point dipoles (section 2.1), and each dipole can be affected by three types of external influence: magnetic dipole-dipole coupling (section 2.3), an applied external magnetic field (section 2.4), and thermal fluctuations (section 2.5). The switching of spins is determined using a generalized Stoner-Wohlfarth model (section 2.6). Imperfections in the ASI are

introduced as different coercive fields, set per spin (section 2.7). Dynamics are modeled using a deterministic single spin flip strategy (section 2.8).

2.1 Magnets as dipoles

ASI systems are physically realized as elongated islands of a ferromagnetic material, arranged on a two-dimensional lattice. The magnets are made small enough to exhibit a single ferromagnetic domain, i.e., coherent magnetization throughout the magnet. The single domain is stable as the energy cost associated with domain walls exceeds the cost associated with the demagnetization energy [13, 14]. Since a magnet has coherent magnetization, it can be approximated by a single mesoscopic spin and the magnetic state can be represented by a single vector $\mathbf{m}$.

The magnets will exhibit an in-plane shape anisotropy due to their small thickness, as well as a magnetization direction defined by their elongated shape. Hence individual magnets can be approximated by classical macro spins with a twofold degenerate ground state defined by the elongated shape of the individual elements. Due to the two degenerate ground state configurations, we approximate each magnet as a magnetic dipole with *binary* magnetization, i.e., a macro spin, $s_i \in \{-1, +1\}$.

As illustrated in Fig. 1, each magnetic dipole is modelled with a position $\mathbf{r}_i$ and rotation θ_i, which together define the ASI geometry. Furthermore, each magnet is assigned a coercive field, $h_c^{(i)}$, describing its resistance to switching (see section 2.6). Using reduced units, the magnetization vector of a single magnet can be expressed as

$$\mathbf{m}_i = s_i \hat{\mathbf{m}}_i \tag{1}$$

where $\hat{\mathbf{m}}_i$ is the unit vector along $\mathbf{m}_i$.

2.2 Magnetic Fields and Temperature

External fields and temperature are modeled as a combination of effective magnetic fields. The total field, $\mathbf{h}_i$, affecting each magnet i is the sum of three components:

$$\mathbf{h}_i = \mathbf{h}_{\text{dip}}^{(i)} + \mathbf{h}_{\text{ext}}^{(i)} + \mathbf{h}_{\text{th}}^{(i)}, \tag{2}$$

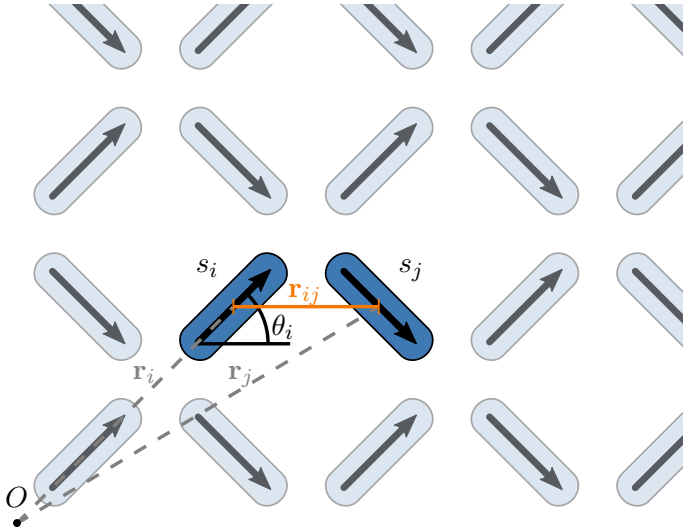


Figure 1: The representation of nanomagnets as spins s_i and associated quantities: angle θ_i and distance to neighbor j, $\mathbf{r}_{ij}$. Note that the magnetization of spin i is given by its spin, s_i, and orientation, θ_i.

where $\mathbf{h}_{\text{dip}}^{(i)}$ is the local magnetic field from neighboring magnets (magnetic dipole-dipole interactions), $\mathbf{h}_{\text{ext}}^{(i)}$ is a global or local external field, and $\mathbf{h}_{\text{th}}^{(i)}$ is a random magnetic field representing thermal fluctuations in each magnetic element. The contributions from each field is described in the following sections.

2.3 Magnetic dipole-dipole interactions

The individual magnets, or spins, are coupled solely through dipole-dipole interactions. Each spin, i, is subject to a magnetic field from all neighboring spins, $j \neq i$, given by

$$\mathbf{h}_{\text{dip}}^{(i)} = \alpha \sum_{j \neq i} \frac{3\mathbf{r}_{ij}(\mathbf{m}_j \cdot \mathbf{r}_{ij})}{|\mathbf{r}_{ij}|^5} - \frac{\mathbf{m}_j}{|\mathbf{r}_{ij}|^3}, \tag{3}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the distance vector from spin i to j, and α scales the dipolar coupling strength between spins. The coupling strength α is given by $\alpha = \frac{\mu_0 M}{4\pi a^3}$, where a is the lattice spacing, M is the absolute magnetic moment of a single magnet, and μ_0 is the vacuum permeability. The distance $\mathbf{r}_{ij}$ is given in reduced units of the lattice spacing.

The dipole field present at each spin's location is calculated by summing the dipole field contributions from spins in its neighborhood. The size of the neighborhood is user-configurable and defined in units of the lattice spacing. In some geometries, such as square ASI, short range interactions dominate the contributions to $\mathbf{h}_{\mathrm{dip}}$ [15, 16], in which case the neighborhood size can be relatively small, for a benefit of increased efficiency. For geometries where long range interactions are significant, a larger neighborhood is required, e.g., pinwheel ASI [17].

2.4 External field

Applying an external magnetic field is the primary mechanism for altering the state of an ASI in a controlled manner. The external field can either be set locally on a per-spin basis, $\mathbf{h}_{\mathrm{ext}}^{(i)}$, globally for the entire system, $\mathbf{h}_{\mathrm{ext}}$, or as a spatial vector field, $\mathbf{h}_{\mathrm{ext}}(\mathbf{r})$.

Time-dependent external fields are supported, i.e., $\mathbf{h}_{\mathrm{ext}}$ is a discrete time series of either local, global or spatial fields. A variety of time-dependent external fields are provided, including sinusoidal, sawtooth and rotational fields. More complex field-protocols can be generated, e.g., for annealing purposes or probing dynamic response.

2.5 Thermal field

Thermal fluctuations are modeled as an additional local field, $\mathbf{h}_{\mathrm{th}}^{(i)}$, applied to each magnet individually. Two orthogonal components of the field are independently drawn from the Normal distribution $\mathcal{N}(0, \sigma_{\mathrm{th}}^2)$. The simulated temperature T is closely related to the value of σ_{th}, where a large σ_{th} corresponds to higher temperatures.

When the material and geometric properties of the magnetic islands are known, it is possible to choose a σ_{th} to match a desired thermal behavior. In some cases, such as for no thermal field and for thermal protocols with no absolute reference point, it is useful to note that $\sigma_{\mathrm{th}} = 0$ implies $T = 0$, and that T is a monotonically increasing function of σ_{th}.

2.6 Switching

Magnetization reversal, or *switching*, may take place when a magnet is subjected to a magnetic field or high temperature. If the field is sufficiently strong (stronger than some critical field) and directed so that the projection onto $\mathbf{m}_i$ is in the opposite direction to $\mathbf{m}_i$, the magnetization $\mathbf{m}_i$ will switch direction.

The critical field strength is referred to as the coercive field h_c. For elongated magnets, h_c depends on the angle between the applied field $\mathbf{h}_i$ and $\mathbf{m}_i$. As illustrated in Fig. 2a, the *easy axis*, where the magnetization favors alignment, lies along the long axis of the magnet, whereas the *hard axis* is perpendicular to the long axis. The external field can be decomposed into two components, $\mathbf{h}_\parallel$ and $\mathbf{h}_\perp$, corresponding to the field component parallel and perpendicular to the easy axis, respectively. We denote the coercive field strength along the hard axis as h_k.

A *switching astroid* is a polar plot of h_c at different angles, with $h_\perp$ on the horizontal axis and $h_\parallel$ on the vertical axis. It is a description of $\mathbf{h}_\mathrm{c}$, or $h_\mathrm{c}(\phi)$. For any applied field $\mathbf{h}_i$ that is outside the switching astroid, the magnet will switch (given that the projection of $\mathbf{h}_i$ onto $\mathbf{m}_i$ is oppositely aligned with respect to $\mathbf{m}_i$).

Fig. 2c shows the normalized switching astroid for an elliptical magnet (Fig. 2a) as obtained from micromagnetic simulations (red dots). Notice how h_c is the smallest at a 45° angle, indicating that a field directed at 45° to a magnet's principal axes will require the least field strength in order to switch its magnetization.

The Stoner-Wohlfarth (SW) model captures the angle dependent switching characteristic of single-domain elliptical magnets [18]. The characteristic SW astroid is shown in Fig. 2c (blue line) and is described by the equation

$$\left(\frac{h_\parallel}{h_k}\right)^{2/3} + \left(\frac{h_\perp}{h_k}\right)^{2/3} = 1. \tag{4}$$

In the SW model, switching may occur when the left hand side of Eq. (4) is greater than one.

The astroid obtained from micromagnetic simulations and the SW astroid (Fig. 2c) are nearly identical, indicating that the SW model is a simple and valid description of switching in elliptical nanomagnets.

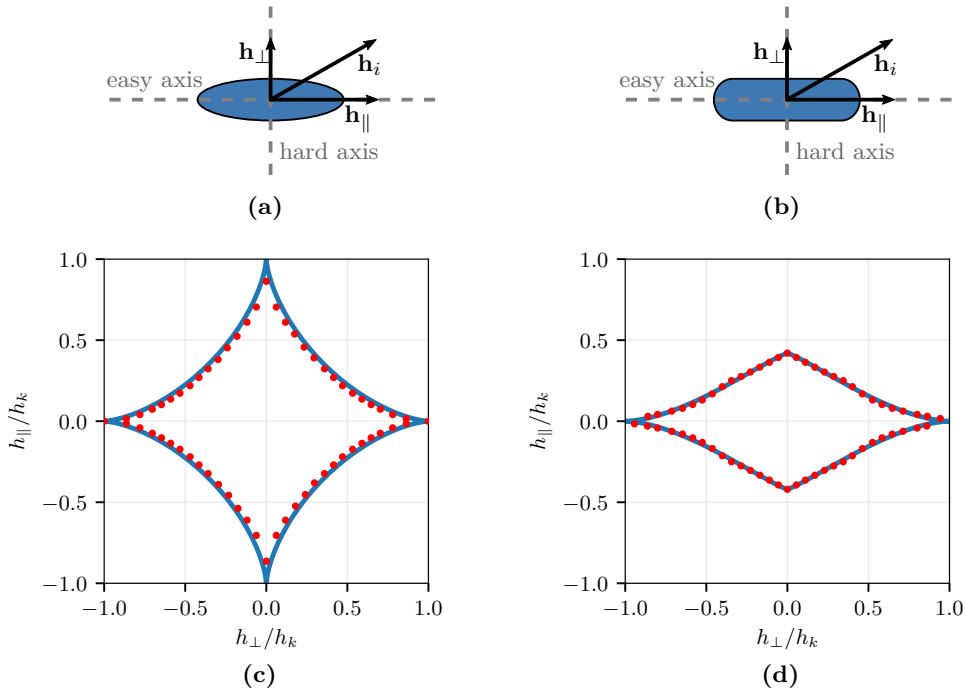


Figure 2: Top: Schematic showing hard and easy axes of (a) an elliptical magnet and (b) a rectangular stadium-shaped magnet, as well as the total field acting on the magnet, $\mathbf{h}_i$, with its parallel and perpendicular components, $\mathbf{h}_\parallel$ and $\mathbf{h}_\perp$, respectively. Bottom: Switching astroid for (c) an elliptical magnet and (d) a rectangular stadium-shaped magnet. Red dots show the coercive field obtained from micromagnetic simulations. The blue line in (c) shows the Stoner-Wohlfarth astroid. The blue line in (d) shows the generalized Stoner-Wohlfarth astroid with parameters $b = 0.42$, $c = 1$, $\beta = 1.7$, and $\gamma = 3.4$ in Eq. (5). The astroids have been normalized with respect to h_k.

However, the SW model is only accurate for elliptical magnets, other magnet shapes typically have quite different switching characteristics. Fig. 2d shows the switching astroid for rectangular stadium-shaped magnets (red dots), which is the shape commonly used in most fabricated ASIs (Fig. 2b). Notice how the astroid is asymmetric: rectangular magnets switch more easily along the easy axis than the hard axis.

To capture the asymmetric switching characteristics of non-elliptical magnets, we have generalized the SW switching model to allow an asymmetry between easy and hard axes. Additionally, the model has been extended to

support tuning of the curvature of the extrema. In the generalized model, the switching threshold is given by

$$\left(\frac{h_{\parallel}}{bh_k}\right)^{2/\gamma} + \left(\frac{h_{\perp}}{ch_k}\right)^{2/\beta} = 1, \tag{5}$$

where b, c, β and γ are parameters which adjust the shape of the astroid: b and c define the height and width, respectively, while β and γ adjust the curvature of the astroid at the easy and hard axis, respectively. Introducing these new parameters allows for tuning of the switching astroid to fit with the shape of nanomagnets used in ASIs. With $b = c = 1$ and $\beta = \gamma = 3$, Eq. (5) reduces to Eq. (4), i.e., the classical Stoner-Wohlfarth astroid is obtained (valid for elliptical magnets).

By tuning the parameters of the generalized SW model, we can obtain the asymmetric switching astroid shown in Fig. 2d (blue line). The astroid is in good agreement with results obtained from micromagnetic simulations (red dots).

In flatspin, the generalized SW model is used as the switching criteria, i.e., a spin may flip if the left hand side of Eq. (5) is greater than one. Additionaly, the projection of $\mathbf{h}_\mathrm{i}$ onto $\mathbf{m}_i$ must be in the opposite direction of $\mathbf{m}_i$:

$$\mathbf{h}_i \cdot \mathbf{m}_i < 0. \tag{6}$$

2.7 Imperfections and disorder

Due to manufacturing imperfections there will always be a degree of variation in the shape and edge roughness of nanomagnets. This variation can be thought of as a disorder in the magnets' inherent properties. Rough edges and a slightly distorted geometry can affect the magnets' switching mechanisms, with defects pinning magnetization and altering the coercive field for each magnet.

In flatspin we model this variation as disorder in the coercive fields. The coercive field is defined individually for each magnet, and a distribution of values can be used to introduce variation. A user-defined parameter, k_disorder, defines the distribution of coercive fields, i.e., $h_k^{(i)}$ is sampled from a normal distribution $\mathcal{N}(h_k, \sigma)$, where $\sigma = k_\mathrm{disorder} \cdot h_k$ (while ensuring $h_k^{(i)}$ is always positive).

2.8 Dynamics

flatspin employs deterministic single spin flip dynamics. At each simulation step, we calculate the total magnetic field, $\mathbf{h}_i$, acting on each spin. Next, we determine which spins *may* flip according to the switching criteria Eqs. (5) and (6). Finally we flip the spin where $\mathbf{h}_i$ is the furthest outside its switching astroid, i.e., where the left hand side of Eq. (5) is greatest. Ties are broken in a deterministic, arbitrary manner, although with non-zero disorder such occurrences are rare. The above process is repeated until there are no more flippable spins.

This relaxation process is performed with constant external and thermal fields. To advance the simulation, the fields are updated and relaxation is performed again. Hence a simulation run consists of a sequence of field updates and relaxation processes.

The dynamical process makes three main assumptions:

1. The external field is quasi-static compared to the time scale of magnet switching.
2. Magnet switching is sequential.
3. The magnet experiencing the highest effective field compared to its switching threshold is the first to switch.

Assumption 1 means the model holds for low frequency external fields, i.e., where switching settles under a static field. The switching mechanics of nanomagnets are typically in the sub nanosecond range [19, 20], and experimental setups often employ external magnetic fields which can be considered static at this time scale. At high applied field frequencies, more complex physical phenomena such as spin waves will have a non-negligible effect on switching dynamics, which are not modeled in flatspin.

Assumption 2 holds if the coercive fields $h_\mathrm{c}^{(i)}$, and total field $\mathbf{h}_i$, of the magnets are sufficiently non-uniform, so that there will always be a single magnet which will flip first. It is assumed to be unlikely that two magnets will have the same $h_\mathrm{c}^{(i)}$ *and* $\mathbf{h}_i$ simultaneously. However, in those rare cases where two magnets are equally close to switching, overlapping switching events may occur in a physical system.

Assumption 3 relies on the fact that all changes in the magnetic fields are effectively continuous, and the change is unidirectional within a simulated time

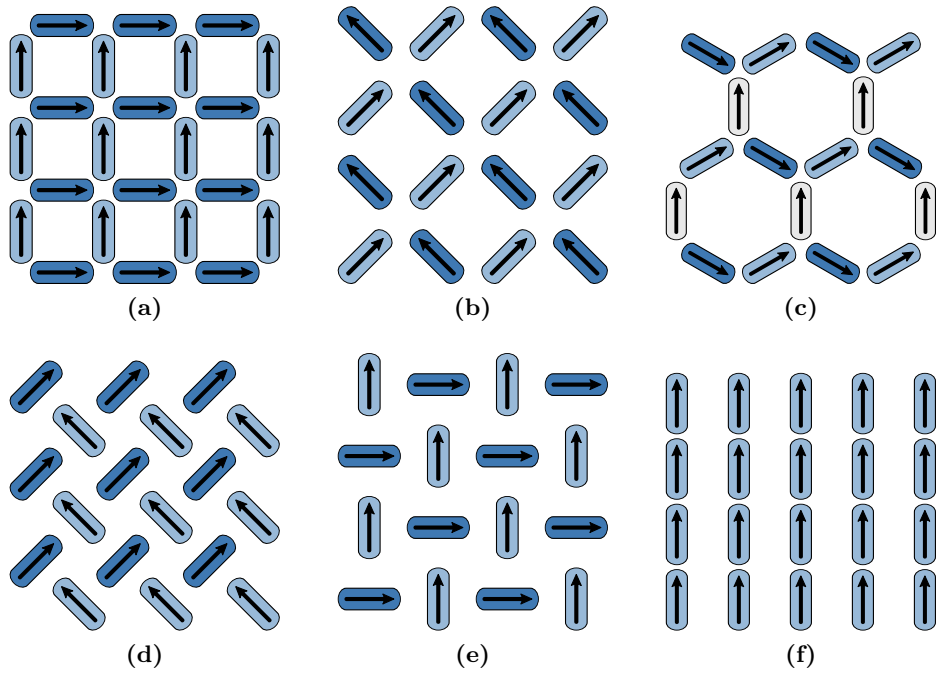


Figure 3: flatspin includes the most common ASI geometries: (a) Square (closed edges), (b) Square (open edges), (c) Kagome, (d) Pinwheel "diamond", (e) Pinwheel "lucky-knot", and (f) Ising.

step, i.e., a *quasi-static field*. Since evaluation happens in discrete time, there will be cases where several magnets are above their corresponding switching thresholds simultaneously. In those cases, the magnet furthest above its switching threshold will have been the first to have crossed the threshold under a quasi-static field. Furthermore, if the angle of the external field is constant, the switching order is invariant to the time resolution of the external field.

2.9 Geometries

The particular spatial arrangement of the magnets is referred to as the *geometry*. A wide range of ASI geometries have been proposed in the literature. Fig. 3 depicts the geometries included in flatspin, which are the most commonly used ASI geometries: square [15], kagome [21, 22], pinwheel [12, 17] and ising [23].

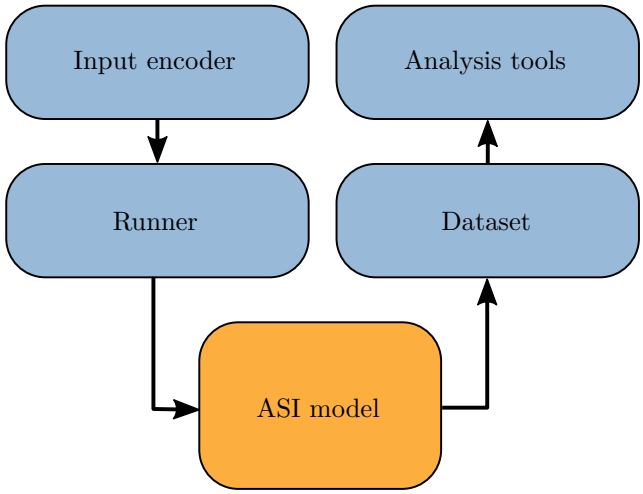


Figure 4: Overview of the flatspin architecture, with arrows indicating data flow.

Geometries are often decomposed into two or more "sublattices", where the magnets within one sublattice are all aligned (have the same rotation). In Fig. 3, the sublattice a magnet belongs to is indicated by its color. As can be seen, both square and pinwheel ASIs have two perpendicular sublattices, whereas kagome has three sublattices.

flatspin can be used to model any two-dimensional ASI comprised of identical elements. New geometries can easily be added by extending the model with a new set of positions $\mathbf{r}_i$ and rotations θ_i.

3 Simulation framework

In addition to a magnetic model, flatspin provides a flexible framework for running simulations, storing results, and performing analysis.

Fig. 4 illustrates the overall architecture of flatspin. The *ASI model* has been described in detail in section 2. Conceptually, the ASI model describes the physical system under study. The rest of the components are tools used by the experimenter to interact with the ASI and observe the results. In this section we briefly describe each of these components.

The *input encoder* translates a set of input values to a series of external fields. Encoders provide a flexible way to define field protocols, and have

been designed with neuromorphic computing in mind. A range of encoders are included, e.g., sinusodial, sawtooth and rotational fields.

The responsibility of the *runner* component is to perturb the ASI model according to the field protocol, and save the results. The model, which is fully parametric, receives parameters from the runner, enabling automated parameter sweeps. In addition, there is support for distributed running of simulations on a compute cluster.

Results are stored in a well-defined *dataset* format which makes the analysis of a large numbers of simulations straightforward. A suite of *analysis tools* are included, e.g., for plotting results, visualizing ensemble dynamics and analysis of vertex populations.

flatspin is written in Python and utilizes OpenCL to accelerate calculations on the GPU. OpenCL is supported by most GPU vendors, hence flatspin can run accelerated on a wide variety of platforms. The simulator may also run entirely on CPUs in case GPUs are not available, albeit at a reduced speed.

flatspin is open-source software and released under a GNU GPL license. For more information see the website [24] and User Manual [25].

4 Validation of flatspin

To evaluate the ASI model, flatspin simulations were compared to established experimental results from the literature, as well as micromagnetic simulations. In particular we investigate phenomena such as Dirac strings in kagome ASI, the size of crystallite domains in square ASI, and superferromagnetism in pinwheel ASI. Finally, we compare the switching order from flatspin simulations with that of micromagnetic simulations, and investigate the effect of varying lattice spacings.

4.1 Dirac strings in kagome ASI

To assess the ability of flatspin to reproduce fine-scale patterns, we consider the emergence of Dirac strings in a kagome ASI (Fig. 3c). Applying a reversal field to a polarized kagome ASI results in the formation of monopole-antimonopole pairs [26]. These pairs are joined by a "string" of nanomagnets

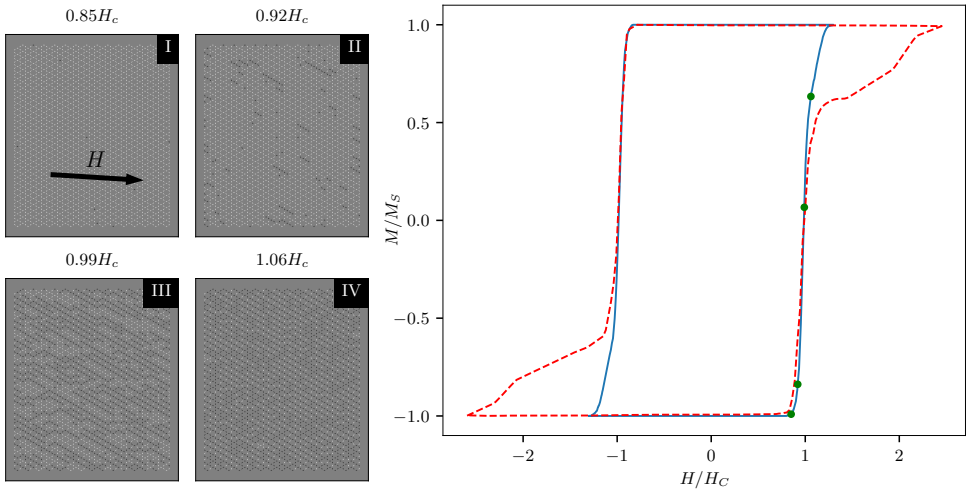


Figure 5: Left: snapshots of the evolution of a kagome ASI at selected field values. Right: Comparison of the hysteresis curve of the simulated ensemble (blue line) against a sketch of the hysteresis curve from the experimental results [26] (red dashed line). Green dots indicate the points at which the snapshots are sampled from.

which have flipped due to the reversal field. As the strength of the reversal field increases, the strings elongate until they fill the array.

We closely follow the methodology set out in an experimental study of Dirac stings in kagome ASI [26], in which a room temperature kagome ASI undergoes magnetization reversal. We start with an array of 2638 magnets (29×29 hexagons) polarized to the left and apply a reversal field $\mathbf{H}$ to the right with a slight, downward offset of 3.6°. This offset breaks the symmetry, such that one of the sublattices is now least aligned with the field, resulting in an increased coercive field on this "unfavored" sublattice.

Micromagnetic simulations of magnets of size $470\,\text{nm} \times 160\,\text{nm} \times 20\,\text{nm}$ yield the following estimation of flatspin parameters: $\alpha = 0.00103, h_k = 0.216, \beta = 2.5, \gamma = 3, b = 0.212, c = 1$. The temperature and disorder parameters $\sigma_{\text{th}} = 0.002$ and $k_{\text{disorder}} = 0.05$ were determined empirically through qualitative comparison with the experimental results [26].

The time evolution snapshots of Fig. 5 demonstrate a strong, qualitative similarity to the results of Mengotti *et al.* [26]. We see Dirac strings developing with a preference to lie along the two sublattices most aligned with the field angle. Furthermore, in the final image, we see the vast majority of unflipped

magnets (excluding the edges) are on the unfavored sublattice, in accordance with both experimental and simulated results from the literature.

Also in Fig. 5, we see the hysteresis of the simulated ensemble (solid line) is similar to that of Mengotti *et al.* [26] (dashed line) in some sections, but differs near the extrema. The hysteresis can be understood in two phases. The first phase, at roughly $M/M_S \in [-0.6, 0.6]$, is dominated by the lengthening of the Dirac strings, with almost no activity occurring on the unfavored sublattice. At $M/M_S < -0.6$ and $M/M_S > 0.6$, the ensemble enters a second phase in which the Dirac strings have fully covered the array, and change in magnetization is dominated by switching on the unfavored sublattice. Clearly we see good agreement, within phase one, between our simulated hysteresis and the experimental results. Furthermore, there is a clear phase transition (characterized by a sharp decrease in gradient) in our hysteresis very close to the transition in the experimental hysteresis. Notably however, although the phase transitions occur at a similar time, the change in gradient is less pronounced in our simulated hysteresis. This disparity indicates that, in the second phase, the magnets on the unfavored sublattice flip more easily in our simulation than in the experimental data.

The accuracy of the point dipole approximation is known to suffer when considering kagome ASI. Specifically, it has been shown to underestimate the coupling coefficient of the nearest neighbors by approximately a factor of 5 [27]. Despite this, we observe flatspin accurately reproduces snapshots of the time evolution behavior, while also capturing salient features of the ensemble hysteresis curve.

4.2 Domain size in square ASI

In order to demonstrate simulation of large-scale behavior, we have reproduced the emergence of large domains of magnetic order in square ASI, similar to experimental results of Zhang *et al.* [9]. One of the main advantages of flatspin over typical alternatives is the scalability and high throughput of large systems with many magnets. Some emergent ASI phenomena require large systems in order to be fully quantified and studied with high fidelity, such as the domain size of magnetic charge crystallites. For ASIs with strongly coupled magnets, typical domain sizes can become too large for direct experimental observation. Thus, an accurate estimate of the domain size for ASIs with a small lattice constant is, in part, limited by the number of directly observable magnets.

For a given range of lattice constants covering both strongly coupled ASIs and weakly coupled ASIs, a corresponding range of large to small magnetic order coherence lengths is expected. In this study, we consider square ASI (closed edges, Fig. 3a) with different lattice constants, a, ranging from 320 nm to 880 nm.

50×50 square ASIs were annealed in flatspin using a thermal protocol of exponentially decreasing σ_{th}. A switching astroid for 220 nm × 80 nm × 25 nm was obtained through micromagnetic simulations, described by generalized astroid parameters $b = 0.4$, $c = 1.0$, $\beta = 3.0$, and $\gamma = 3.0$. Additionally, $h_k = 0.186$, $k_{\mathrm{disorder}} = 0.05$, and a neighbor distance of 10 magnets was used. The thermal protocol was chosen such that the total dipole interaction energy was not significantly reduced by increasing simulation steps.

In the annealed state, the spin-spin correlation as a function of their lateral separation was calculated across the ensembles. Analysis of the average correlation of annealed states provides insight about the typical coherence length of magnetic order, i.e., magnetic charge crystallite size, or domain size. Here, the correlation of two spins is defined as +1 (-1) if their dipole interaction is minimized (maximized). Averaging correlation across distinct types of spin pairs, in the annealed ASI, gives a measure of how coherent the ASI is at that particular neighbor separation. How quickly the average correlation decreases as a function of separation can be used to estimate the characteristic domain size. In particular, it can be argued that the separation where the correlation falls below $1/e$ is the characteristic domain radius [9, 28].

Typical domain structures and correlation results can be seen in Fig. 6. The domains shown in Fig. 6a and the correlation curves in Fig. 6b are, for the most part, in good agreement with experimental results [9]. A qualitative comparison of the domain sizes and structures in Fig. 6 shows that the domains tend to be larger, with smoother domain boundaries, for smaller a. The analysis of coherence as a function of separation also shows identical trends and similar values, where an increase in a leads to low correlation, even between nearest neighbors.

For cases where $a < 400$ nm, domains do not significantly increase in size when the lattice constant is further reduced. This discrepancy with the experimental results is not completely unexpected: the point-dipole approximation is known to underestimate nearest-neighbor interaction for magnets placed close together [27]. In addition, a stronger interaction between spins would cause each spin flip to contribute a greater change in the total dipole

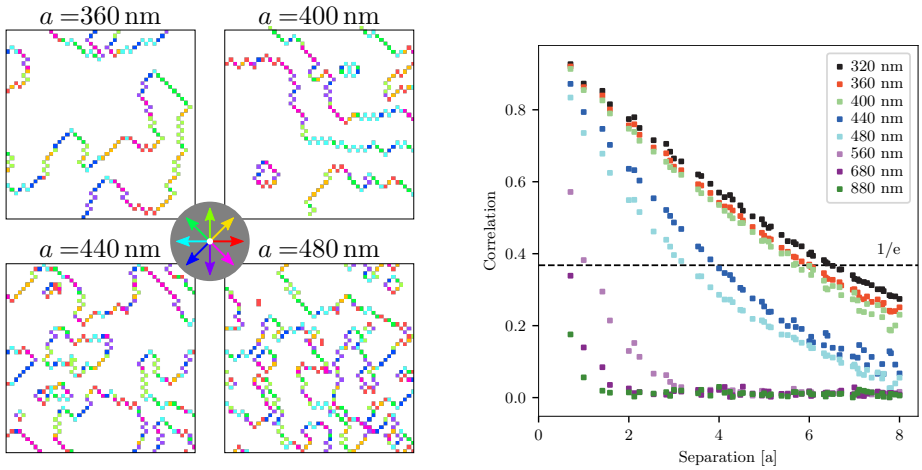


Figure 6: (a) Maps showing the net magnetization of the ASI vertices, for an annealed 50×50 square ASI with the given lattice constant a. The white regions have zero net magnetization and thus correspond to a coherent domain of Type I vertices. Colored regions have a non-zero net magnetization, direction indicated by the color wheel, and correspond to Type II or III vertices. (b) The absolute value of spin-spin correlation at a given separation for square ASIs of different lattice constants, a. Also indicated is a $1/e$ threshold of correlation (dashed line).

energy. This makes a gradual descent towards the ground state by random spin flips (the thermal fluctuations as modeled by flatspin) harder to achieve. These issues may be addressed by increasing the coupling parameter α for nearest neighbor spins, and by a longer and slower annealing protocol. A longer and slower annealing protocol will inevitably come at the cost of longer computation times. In a future version of flatspin, it might be beneficial to include other thermal effects such as a declining saturation magnetization of the constituent material.

These results show that flatspin provides sufficient flexibility, fidelity and performance required to reproduce experimentally observed large-scale emergent behavior in ASIs.

4.3 Superferromagnetism in pinwheel ASI

In this section, we use flatspin to reproduce the dynamic behavior of pinwheel ASI, which had yet to be demonstrated with a dipole model [12]. We find that

our switching criteria plays a key role in replicating magnetization details during the field-driven array reversal.

Pinwheel ASI is obtained by rotating each island in square ASI some angle about its center. A rotation of 45 degrees results in a transition from antiferromagnetic to ferromagnetic order [17]. The dynamics of pinwheel ASI in many ways resemble continuous ferromagnetic thin films, with mesoscopic domain growth originating from nucleation sites, followed by coherent domain propagation and complete magnetization reversal [12].

Here we demonstrate that flatspin is able to replicate the experimental reversal processes presented in Li *et al.* [12], where pinwheel "diamond" ASI (Fig. 3d) is subject to an external field at different angles. A key result is that the angle θ of the external field controls the nature of the reversal process. When θ is small (equally aligned to both sublattices), reversal happens in a single avalanche, whereas when θ is large (more aligned to one sublattice), reversal happens in a two-step process where one sublattice switches completely before the other. Previous attempts at capturing this behavior in a dipole model have proven unfruitful [12].

To replicate this process in flatspin, an asymmetric switching astroid is required, i.e., the threshold along the parallel component is reduced by setting $b < 1$ in Eq. (5). From micromagnetic simulations of a single $470 \times 170 \times 10\,\mathrm{nm}$ magnet we obtain the following characteristic switching parameters: $b = 0.28$, $c = 1.0$, $\beta = 4.8$ and $\gamma = 3.0$. Other simulation parameters include $\alpha \approx 0.00033$, $h_k = 0.098$, $k_{\mathrm{disorder}} = 0.05$ and a neighbor distance of 10.

Figs. 7a to 7d show hysteresis loops and array snapshots when the field is aligned with the array ($\theta = 0°$ and $\theta = -6°$). As can be seen, the results from flatspin (Figs. 7b and 7d) are qualitatively very similar to experimental results (Figs. 7a and 7c). In all cases, the ASI undergoes reversal in a single avalanche. Reversal begins at a small number of nucleation points close to the edge, followed by domain growth and domain wall movement perpendicular to the direction of the field. The simulated system appears to have an anisotropy axis of 0° as opposed to −6° observed experimentally. Hence Fig. 7b is most similar to Fig. 7c and Fig. 7d is most similar to Fig. 7a. It should be noted that the non-zero anisotropy axis found experimentally has not yet been explained.

Figs. 7e and 7f show the hysteresis loops and array snapshots when the field is misaligned with the array ($\theta = 30°$). Again, flatspin simulations (Fig. 7f)

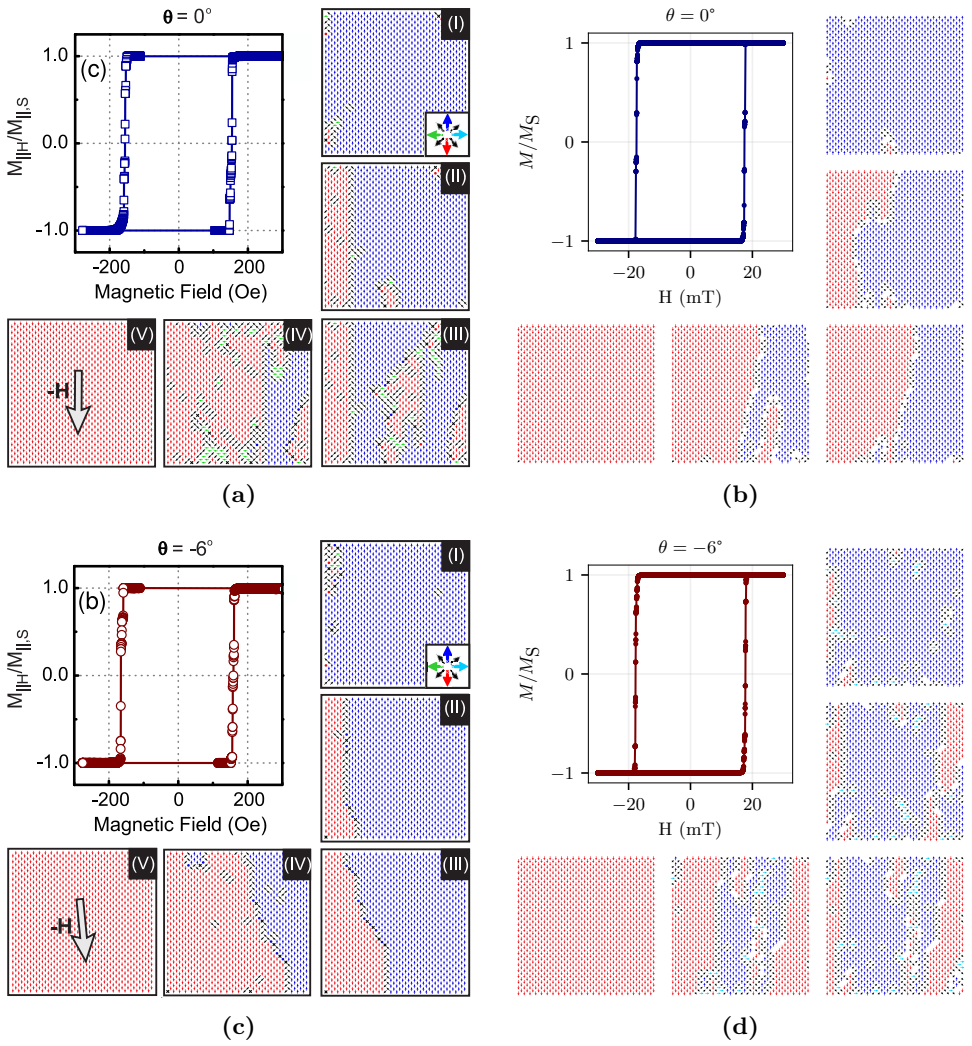
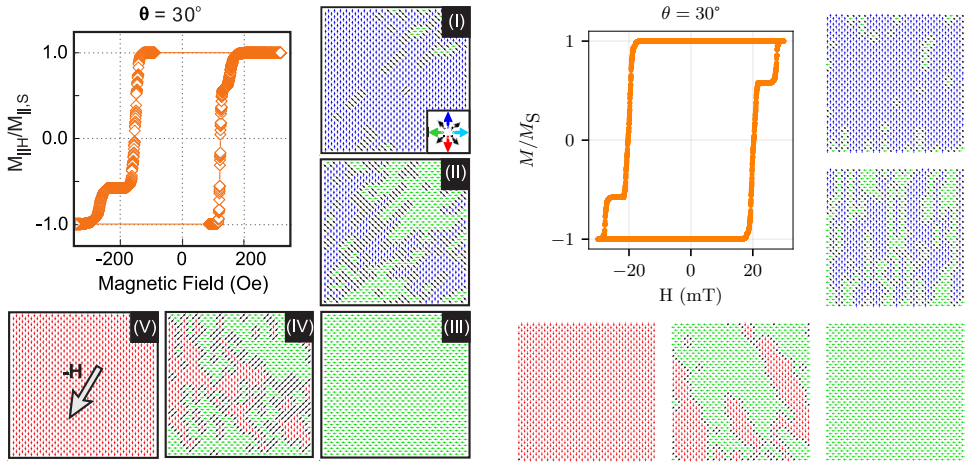


Figure 7: Hysteresis loop and snapshots of the pinwheel units for various angles θ of the applied field. Figures (a), (c) and (e) show experimental results, adapted from Li *et al.* [12], Copyright ©2018 American Chemical Society, CC-BY-4.0 `https://creativecommons.org/licenses/by/4.0/`. Figures (b), (d) and (f) show results from flatspin simulation (white indicates zero net magnetization).



(e) **(f)**

Figure 7 (cont.): Hysteresis loop and snapshots of the pinwheel units for various angles θ of the applied field. Figures (a), (c) and (e) show experimental results, adapted from Li *et al.* [12], Copyright ©2018 American Chemical Society, CC-BY-4.0 `https://creativecommons.org/licenses/by/4.0/`. Figures (b), (d) and (f) show results from flatspin simulation (white indicates zero net magnetization).

replicate key features observed experimentally (Fig. 7e). Reversal now happens in two steps: the sublattice whose magnets have their easy axis most aligned with the field will switch first, followed later by the other sublattice. This two-step reversal process results in an emergent rotation of the collective magnetization. The magnetization is constrained to follow the orientation of the magnets, resulting in reversal via stripe patterns at 45°.

Li *et al.* [12] report they were unable to replicate the magnetization details using a point-dipole Monte Carlo model. One crucial difference between flatspin and their dipole model is the switching criteria. They use the simpler criteria $\mathbf{h}_i \cdot \mathbf{m}_i < h_k^{(i)}$, which considers only the parallel field component and will be largely inaccurate for fields which are not aligned with the magnet's easy axis. Indeed, using this simpler switching criteria in flatspin results in a very different reversal process and magnetization details (not shown).

4.4 Comparison to micromagnetic single-spin switching order

Micromagnetic simulations in MuMax3 [29] have been shown to agree with experiment due to high simulation fidelity. It is, therefore, of interest to study how well flatspin agrees with MuMax3 at the level of detail expressed in flatspin.

Here we evaluate the switching strategy outlined in Section 2.8, by comparing the switching orders obtained in flatspin and MuMax3, of a square ASI as it undergoes reversal by an external field. As a similarity measure, Spearman's rank correlation coefficient ρ [30] is used, where a value of 1 indicates perfect correlation and 0 indicates no correlation between switching orders.

In the weakly coupled regime, the switching order is dictated by the coercive fields of each individual magnet. In flatspin, the coercive field can be set directly by modifying $h_{\mathrm{k}}^{(i)}$. In MuMax3, we control the coercive field implicitly, by varying the first-order, uniaxial, magnetocrystalline anisotropy, $K_{\mathrm{U1}}^{(i)}$ of each magnet. Given a set of randomly drawn $K_{\mathrm{U1}}^{(i)}$ values, the corresponding $h_{\mathrm{k}}^{(i)}$ values were obtained by a linear map.

The system we considered was a 4×4 square (closed) ASI, each magnet measuring $220\,\mathrm{nm} \times 80\,\mathrm{nm} \times 25\,\mathrm{nm}$. flatspin was run with parameters $b = 0.38$, $c = 1$, $\beta = 1.5$, and $\gamma = 3.2$. In both simulators, we applied a gradually increasing reversal field at $\theta = 44°$.

At a certain point the dipolar interactions begin contributing to the switching order. To verify that flatspin still captures switching dynamics, we perform a comparision of the switching orders for all pairs of lattice spacings in both simulators.

Fig. 8a shows the correlations for each pair of lattice spacings as an average over 32 different square ASIs. We observe a clear linear relationship between the two simulators, with higher lattice spacings exhibiting higher correlation. The non-zero y-intercept in the heatmap indicates that, as suspected, the coupling strength is slightly underestimated by the dipole approximation employed in flatspin, in particular for lower lattice spacings. For example, flatspin with 300 nm lattice spacing is most similar to MuMax3 with 380 nm.

The red line in Fig. 8b traces the ridge in the heatmap, i.e., the highest ρ, for each flatspin lattice spacing. As can be seen, a near-perfect agreement between the simulators is found in the weakly coupled regime (high lattice

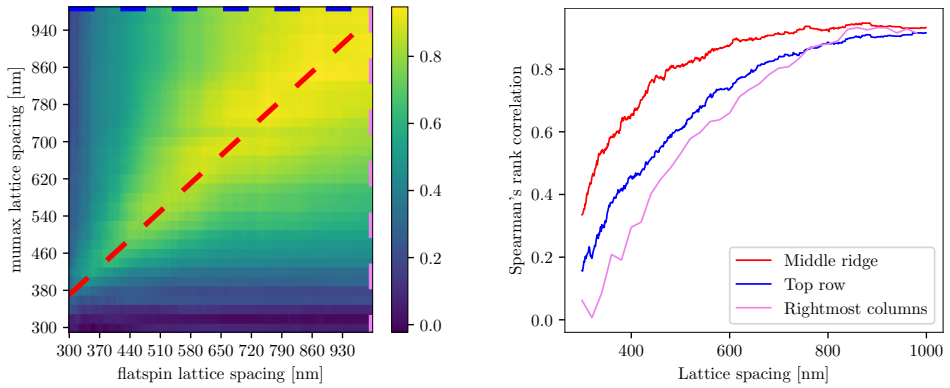


Figure 8: (a) Spearman's rank correlation coefficients ρ averaged over 32 different square ASIs, evaluated for different lattice spacings in flatspin and MuMax3. The red line shows the approximate maximum ridge line through the heatmap. (b) The red line shows the true maximum ρ for the lattice spacing pairs. The blue and violet lines show projections of the top row and rightmost column of (a), respectively.

spacing). As lattice spacings decrease, the effect of dipole interactions become apparent. Below 450 nm the correlation drops. Since flatspin does not account for the micromagnetic state, complete correlation is not expected.

The particular selection of $h_{\mathrm{k}}^{(i)}$ values introduces an *inherent* bias in the switching order. One might expect that this bias causes the dipole interactions to have a negligible impact on the switching order, leading to an inflated correlation between flatspin and MuMax3, regardless of lattice spacing.

The violet and blue lines, (rightmost column and top row from Fig. 8a), show the deviation from the inherent switching order in MuMax3 and flatspin, respectively. Their rapid decline confirms that the switching order is not dominated by the inherent bias for highly coupled systems. The red line clearly shows a stronger agreement between MuMax3 and flatspin (higher than the blue and violet lines), even at smaller lattice spacings.

5 Performance

Although the total simulation time will depend on many factors, it is of interest to measure how simulation time scales with the number of spins. As

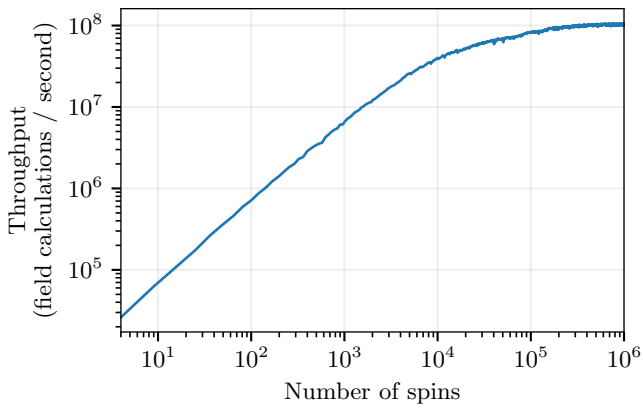


Figure 9: The throughput (number of field calculations per second) as a function of number of spins. Throughput is averaged over 100 simulations of each size. The test was performed on an NVIDIA Tesla V100 GPU with 32GB of RAM. Note the logarithmic scale of the axes.

the number of spins are increased, simulation time will be largely dominated by the calculation of the effective field, $\mathbf{h}_i$, acting on each of the N spins in the lattice. Computing time for $\mathbf{h}_{\text{dip}}^{(i)}$ depends on the number of neighbors around spin i, which is typically constant for all spins except the ones at the edges of the geometry. For large N, the number of edge magnets is negligible (in the common ASI geometries). Computing $\mathbf{h}_i$ for all spins will take $\mathcal{O}(N)$ time, i.e., computation time grows no faster than linear in N.

Fig. 9 shows the throughput (number of field calculations per second) as a function of the number of spins. Here a field calculation is defined as the computation of $\mathbf{h}_i$ for a single spin i, hence for N spins there will be N such field calculations. The geometry used was square ASI (open edges) using a standard 8 spin neighborhood for calculating $\mathbf{h}_{\text{dip}}^{(i)}$. The throughput was averaged over 100 simulations of each size. The test was performed on an NVIDIA Tesla V100 GPU with 32GB of RAM.

At around 200 000 spins, the throughput saturates at 10^8 field calculations per second. On our test setup, computing $\mathbf{h}_i$ for one million spins takes approximately 10 ms. Above 200 000 spins we are able to fully utilize the GPU resources.

To simulate the reversal of an ASI by a gradually increasing external field, at least one field calculation per spin flip is required, i.e., at least N field calculations. If the external field gradually changes with a resolution of K

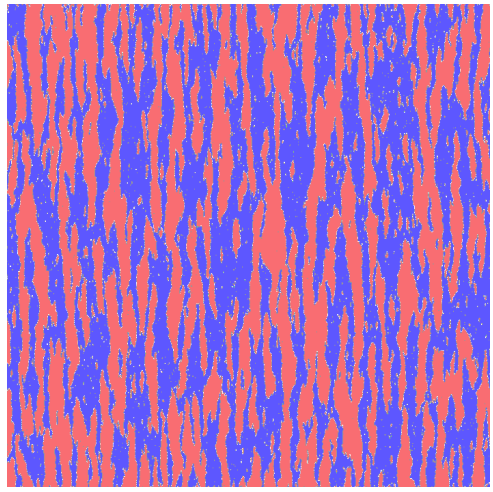


Figure 10: A snapshot from flatspin simulations of 708×708 pinwheel ASI with more than one million magnets, as it undergoes reversal by an external field. The angle of the external field is $\theta = 0°$.

values, the worst case will be when all spins flip during a single field value. In this case the number of field calculations required will be $N + K - 1$ since there will be $K - 1$ field calculations which results in no spin flips.

The total simulation time depends largely on the particular experimental setup, parameters and other system characteristics. Time will be spent on things other than field calculations, e.g., organizing and writing results to storage. Hence the total simulation time will be longer than predicted by field calculations alone. As an example, the simulations from section 4.3 of 25×25 pinwheel ASI with 1250 magnets took approximately 6 seconds with $K = 2500$, for one reversal.

Fig. 10 shows a snapshot from flatspin simulations of 708×708 pinwheel ASI as it undergoes reversal by an external field. The ability to simulate such large systems allows an experimenter to explore phenomena at much larger scales than can be directly observed experimentally. With more than one million magnets, the simulation of array reversal took several days to complete. A video of the full reversal is available as Supplemental Material[1].

[1] See Supplemental Material at [URL will be inserted by publisher] for a video of the reversal of 708x708 pinwheel ASI.

6 Conclusion

flatspin is a highly effective simulator for ASI ensemble dynamics. At its heart lies a robust magnetic model based on dipole-dipole interactions with a switching criteria based on a generalized Stoner-Wohlfarth model. Accompanying the model is a toolbox of useful input encoders and analysis tools. The model includes several common ASI geometries, and there are no inherent limits to the range of possible geometries.

The flatspin ASI model has been verified against micromagnetic simulations and experimental results from the literature. On a detailed level, we found good agreement between micromagnetic simulations and flatspin in terms of magnet switching order. Emergent fine-scale patterns in kagome ASI were replicated successfully, where the formation of Dirac strings matched experimental results. Large-scale domain sizes in square ASI were reproduced, and good agreement was found between flatspin and experimental results. Finally, using flatspin, the experimental magnetization reversal of pinwheel ASI was reproduced for the first time in a dipole model.

Through GPU acceleration, flatspin scales to large ASI systems with millions of magnets. High speed, parallel computation allows for a large number of ASI simulations to be executed, enabling quick exploration of parameters and novel geometries. The flexibility and performance offered by flatspin opens for unprecedented possibilities in ASI research.

Acknowledgements

This work was funded in part by the Norwegian Research Council under the SOCRATES project, grant number 270961. Simulations were executed on the NTNU EPIC compute cluster [31].

References

[1] S. H. Skjærvø, C. H. Marrows, R. L. Stamps, and L. J. Heyderman, Advances in artificial spin ice, Nature Reviews Physics **2**, 13–28 (2020).

[2] L. J. Heyderman and R. L. Stamps, Artificial ferroic systems: novel functionality from structure, interactions and dynamics, Journal of Physics: Condensed Matter **25**, 363201 (2013).

[3] J. Sklenar, Y. Lao, A. Albrecht, J. D. Watts, C. Nisoli, G.-W. Chern, and P. Schiffer, Field-induced phase coexistence in an artificial spin ice, Nature Physics **15**, 191–195 (2019).

[4] D. J. P. Morris, D. A. Tennant, S. A. Grigera, B. Klemke, C. Castelnovo, R. Moessner, C. Czternasty, M. Meissner, K. C. Rule, J.-U. Hoffmann, K. Kiefer, S. Gerischer, D. Slobinsky, and R. S. Perry, Dirac strings and magnetic monopoles in the spin ice Dy2Ti2O7, Science **326**, 411–414 (2009).

[5] D. Levis, L. F. Cugliandolo, L. Foini, and M. Tarzia, Thermal Phase Transitions in Artificial Spin Ice, Physical Review Letters **110**, 207206 (2013).

[6] L. Anghinolfi, H. Luetkens, J. Perron, M. G. Flokstra, O. Sendetskyi, A. Suter, T. Prokscha, P. M. Derlet, S. L. Lee, and L. J. Heyderman, Thermodynamic phase transitions in a frustrated magnetic metamaterial, Nature Communications **6**, 8278 (2015).

[7] J. H. Jensen, E. Folven, and G. Tufte, Computation in artificial spin ice, in *The 2018 Conference on Artificial Life* (MIT Press, Tokyo, Japan, 2018) pp. 15–22.

[8] Z. Budrikis, *Athermal dynamics of artificial spin ice: disorder, edge and field protocol effects*, Ph.D. thesis, The University of Western Australia (2012).

[9] S. Zhang, I. Gilbert, C. Nisoli, G. W. Chern, M. J. Erickson, L. O'Brien, C. Leighton, P. E. Lammert, V. H. Crespi, and P. Schiffer, Crystallites of magnetic charges in artificial spin ice, Nature **500**, 553–557 (2013).

[10] X. Ke, J. Li, C. Nisoli, P. E. Lammert, W. McConville, R. F. Wang, V. H. Crespi, and P. Schiffer, Energy Minimization and ac Demagnetization in a Nanomagnet Array, Physical Review Letters **101**, 037205 (2008).

[11] Y. Bar-Yam, *Dynamics of Complex Systems* (Perseus Books, USA, 1997).

[12] Y. Li, G. W. Paterson, G. M. Macauley, F. S. Nascimento, C. Ferguson, S. A. Morley, M. C. Rosamond, E. H. Linfield, D. A. MacLaren, R. Macêdo, C. H. Marrows, S. McVitie, and R. L. Stamps, Superferromagnetism and Domain-Wall Topologies in Artificial "Pinwheel" Spin Ice, ACS Nano , acsnano.8b08884 (2019).

[13] J. Frenkel and J. Doefman, Spontaneous and induced magnetisation in ferromagnetic bodies, Nature **126**, 274–275 (1930).

[14] C. Kittel, Theory of the structure of ferromagnetic domains in films and small particles, Phys. Rev. **70**, 965–971 (1946).

[15] R. F. Wang, C. Nisoli, R. S. Freitas, J. Li, W. McConville, B. J. Cooley, M. S. Lund, N. Samarth, C. Leighton, V. H. Crespi, and P. Schiffer, Artificial 'spin ice' in a geometrically frustrated lattice of nanoscale ferromagnetic islands, Nature **439**, 303–306 (2006), 0601429 [cond-mat] .

[16] C. Fan and F. Y. Wu, Ising model with second-neighbor interaction. i. some exact results and an approximate solution, Phys. Rev. **179**, 560–569 (1969).

[17] R. Macêdo, G. M. Macauley, F. S. Nascimento, and R. L. Stamps, Apparent ferromagnetism in the pinwheel artificial spin ice, Physical Review B **98**, 014437 (2018).

[18] C. Tannous and J. Gieraltowski, The stoner–wohlfarth model of ferromagnetism, European Journal of Physics **29**, 475–487 (2008).

[19] R. Kikuchi, On the minimum of magnetization reversal time, Journal of Applied Physics **27**, 1352–1357 (1956).

[20] P. R. Gillette and K. Oshima, Magnetization reversal by rotation, Journal of Applied Physics **29**, 529–531 (1958).

[21] M. Tanaka, E. Saitoh, H. Miyajima, T. Yamaoka, and Y. Iye, Magnetic interactions in a ferromagnetic honeycomb nanoscale network, Physical Review B **73**, 052411 (2006).

[22] Y. Qi, T. Brintlinger, and J. Cumings, Direct observation of the ice rule in an artificial kagome spin ice, Physical Review B **77**, 094418 (2008), arXiv:0802.0034 .

[23] M. Saccone, A. Scholl, S. Velten, S. Dhuey, K. Hofhuis, C. Wuth, Y.-L. Huang, Z. Chen, R. V. Chopdekar, and A. Farhan, Towards artificial ising spin glasses: Thermal ordering in randomized arrays of ising-type nanomagnets, Phys. Rev. B **99**, 224403 (2019).

[24] flatspin contributors, flatspin website, `https://flatspin.gitlab.io` (2020).

[25] flatspin contributors, flatspin user manual, `https://flatspin.readthedocs.io/en/latest/` (2020).

[26] E. Mengotti, L. J. Heyderman, A. F. Rodríguez, F. Nolting, R. V. Hügli, and H.-B. Braun, Real-space observation of emergent magnetic monopoles and associated dirac strings in artificial kagome spin ice, Nature Physics **7**, 68 (2011).

[27] N. Rougemaille, F. Montaigne, B. Canals, A. Duluard, D. Lacour, M. Hehn, R. Belkhou, O. Fruchart, S. El Moussaoui, A. Bendounan, *et al.*, Artificial kagome arrays of nanomagnets: a frozen dipolar spin ice, Physical Review Letters **106**, 057209 (2011).

[28] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, 1999).

[29] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. Van Waeyenberge, The design and verification of MuMax3, AIP Advances **4**, 107133 (2014), 1406.7635 .

[30] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference: Revised and Expanded* (CRC press, 2014).

[31] M. Själander, M. Jahre, G. Tufte, and N. Reissmann, EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, arXiv (2019), arXiv:1912.05848 [cs.DC], arXiv:1912.05848 [cs.DC] .

Paper E

Reservoir Computing in Artificial Spin Ice

(Jensen and Tufte, 2020)

Author(s):
Johannes H. Jensen and Gunnar Tufte

Published at conference:
ALIFE 2020: The 2020 Conference on Artificial Life

Paper E
Jensen and Tufte (2020)

Reservoir Computing in Artificial Spin Ice

Johannes H. Jensen and Gunnar Tufte

Department of Computer Science,
Norwegian University of Science and Technology, Trondheim, Norway

Abstract

Artificial spin ice (ASI) are systems of coupled nanomagnets arranged on a 2D lattice. ASIs are promising computing substrates due to the rich variety of emergent behavior, accompanied by considerable control and flexibility. Computational models may exploit the small-scale dynamics of the individual elements, or large-scale emergent behavior of the resulting metamaterial. We investigate the computational capabilities of "pinwheel" ASI, whose emergent ferromagnetic patterns can be observed at different scales. Within a reservoir computing framework, we examine how key system parameters affect performance using well-established reservoir quality metrics. As reservoir output, we consider system state at different granularities, ranging from individual magnets to the collective state of multiple magnets. Our results show that pinwheel ASI exhibits excellent computing capacity, including evidence of fading memory. Interestingly, a wide range of output granularities result in good performance, offering new insights into the scalability and robustness of reservoirs based on self-organized collective behavior. The apparent flexibility in output granularity show that ASIs have computational properties at different abstraction levels, from the small-scale dynamics of simple elements, to the large-scale spatial patterns of the metamaterial.

1 Introduction

In ASI, each nanomagnet behaves as a macrospin, analogous to the atomic spins in bulk materials. Collectively, the macrospins form a magnetic meta-

material, whose emergent properties can be controlled directly by the placement, orientation and shape of the nanomagnets.

Coupled nanomagnetic systems are ideal for studies related to self-organization and emergence. A wide range of emergent phenomena has been discovered in ASIs, e.g., collective ferromagnetic/antiferromagnetic ordering (Sklenar et al., 2019), domain wall propagation (Li et al., 2019), avalanche dynamics (Mengotti et al., 2011), and phase transitions (Levis et al., 2013).

Furthermore, established nanofabrication methods make ASIs readily available for real-world exploration. Unlike atomic spins, the mesoscopic size of the nanomagnets enables direct observation of the macrospin states through magnetic microscopy. Micromagnetic simulations are feasible for smaller systems (Leliaert et al., 2018; Jensen et al., 2018), while large-scale behavior can be captured by mesoscopic models (Jensen et al., 2020).

As systems of coupled spins, ASIs are natural substrates for neuromorphic computing. Like biological computing systems, the coupled nanomagnets form large spatial networks of nonlinear nodes, where computation is closely linked to memory. Computation in neuromorphic systems is inherently parallel, the result of interactions between large numbers of simple elements.

An alternative view is ASI as a metamaterial: when observed at larger scales, magnetic patterns emerge as a result of the underlying macrospin interactions. The metamaterial view is a natural fit for material computation (Stepney et al., 2018). Compared to bulk materials, metamaterials offer considerable control and flexibility, and opens for the design of exotic substrates with unusual physical behavior. Furthermore, computation with large-scale emergent phenomena offers an inherent robustness, as small differences in the underlying state are washed out in the aggregate view.

Here, we explore the computation arising from these alternate ASI views. By observing the system at different scales (adjusting the amount of "squinting"), it is possible to move gradually between the two views: at the smallest scale we have the network of spins, while at larger scales we approach the metamaterial. How does the scale of observation affect computation? This question has practical implications for computing devices based on ASI, where the readout of magnetic state necessitates sensor circuitry with an associated cost. Note that the same is true for all physical computing devices: readout of state has a cost which scales with a growing number of outputs.

Specifically, we investigate the computational properties of “pinwheel” ASI within a Reservoir Computing (RC) framework. Using established RC met-

rics, we study how key system parameters affect performance. We consider different output granularities to define the reservoir nodes, ranging from single magnets to the aggregate of multiple magnets.

2 Background

2.1 Artifical Spin Ice

ASIs have received considerable interest over the last decade, primarily as a model system for the study of fundamental physics. The name "artificial spin ice" stems from the use of engineered systems to mimic the arrangement of molecules in water ice. Established nanofabrication techniques coupled with the ability to directly observe macrospin states, has enabled the study of a wide range of physical phenomena in ASI (Skjærvø et al., 2020).

In ASI systems, each nanomagnet behaves as a binary mesoscopic spin. The small size ensures a uniform internal magnetization (a single-domain state), while an elongated shape constrains the orientation of the magnetization to lie along the long axis (a binary state).

The artificial spins are coupled via the magnetic dipole-dipole interaction: each magnet is subject to the stray magnetic field of neighboring magnets.

The particular arrangement and orientation of the magnets is referred to as the *geometry*, which effectively defines the nature of the magnet-magnet interactions. Fig. 1a depicts square ASI, which consists of horizontal and vertical magnets arranged on two square lattices. The sub-lattice with vertical magnets is placed at an offset from the sub-lattice with horizontal magnets, as indicated by the different colors. Pinwheel ASI is shown in Fig. 1c, and is obtained by rotating each magnet in square ASI by 45° about its center.

Some geometries result in antiferromagnetic ordering, where domains of zero net magnetization are energetically favorable. Fig. 1b shows the collective magnetization in square ASI, with the emergence of antiferromagnetic domains (white regions). In antiferromagnetic systems, only the boundaries of the domains have an observable magnetization at larger scales. Pinwheel ASI, on the other hand, exhibits ferromagnetic behavior, i.e., the magnets form domains with coherent magnetization of non-zero magnitude. Fig. 1d

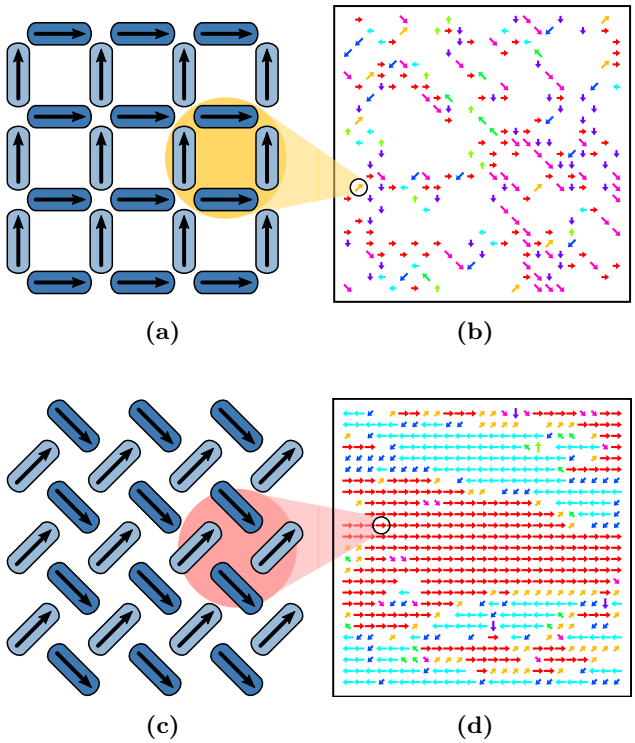


Figure 1: The emergent behavior of ASI is a result of the placement and orientation of the nanomagnets. (a) Square ASI consists of horizontal and vertical magnets arranged on a square lattice. (b) Square ASI favors antiferromagnetic order, resulting in domains of zero net magnetization (white regions). (c) Pinwheel ASI is obtained by rotating each magnet in square ASI by 45° about its center. (d) Pinwheel ASI exhibits long-range ferromagnetic order, supporting formation of domains with coherent magnetization. The systems shown in (b) and (d) are 25×25 square and pinwheel ASI, respectively, each consisting of 1300 magnets. The arrows in (b) and (d) indicate the collective magnetization of the four circled magnets in (a) and (c), respectively.

shows emergent ferromagnetic patterns as found in pinwheel ASI. Ferromagnetic domains are also clearly visible at large scales, making pinwheel ASI ideal for our study.

There are a myriad of ways to tune the behavior of ASIs. For example, the lattice spacing (distance between magnet centers) determines the size of the anti- or ferromagnetic domains: a smaller spacing results in larger domains. Small changes to the geometry can result in fundamentally different behavior. Novel geometries provide a seemingly endless playground for exploration of self-organization and emergence *in-materio*. In addition, there are several ways to tune behavior externally, without altering the system, e.g., through an external magnetic field or temperature.

2.2 Reservoir Computing

Reservoir Computing (RC) is a methodology which allows a dynamical system to be exploited for computation (Jaeger, 2001; Maass et al., 2002). The key component is the dynamical system, which is referred to as the *reservoir*. An input signal perturbs the reservoir, which, as a result of its inherent properties, produces a complex dynamic response. The reservoir functions as a nonlinear kernel with memory, maintaining a rich repertoire of nonlinear input transformations. Subsequently, a linear *readout layer* is trained to produce some desired function as a weighted sum of reservoir states. Crucially, the readout layer is the only trained part of the system, i.e., both the input layer and the reservoir remains unchanged.

State of the art performance has been obtained using RC methods for a variety of tasks, both with classical neural reservoirs (Lukoševičius and Jaeger, 2009) as well as a range of physical reservoirs (Tanaka et al., 2019). A variety of magnetic reservoirs have been proposed, such as magnetic tunnel junctions (Furuta et al., 2018), spin torque oscillators (Torrejon et al., 2017) magnetic skyrmions (Prychynenko et al., 2018), magnetic thin-films (Nakane et al., 2018) and dipole coupled nanomagnets (Nomura et al., 2019). The latter two examples bear some resemblance to ASIs, as magnetic metamaterials consisting of dipole coupled nanomagnets.

Good reservoirs are nonlinear, high dimensional dynamical systems with rich dynamics. Interactions between nodes in the reservoir facilitates the formation of nonlinear memory, i.e., where the state of a node is a nonlinear function of current and previous inputs. Crucially, the reservoir should have the

echo-state property which, informally, means the reservoir gradually forgets over time.

ASIs are promising reservoirs since they exhibit many of the above-mentioned properties. Magnetic switching is inherently nonlinear, hence a large number of magnets is a high-dimensional nonlinear system. Magnetic dipole-dipole interactions enable the flow of information between nodes, with the potential for memory formation. Reservoir state can be observed directly as the state of individual spins, or through emergent patterns at coarser granularities.

As dynamical systems, ASIs exhibit a large number of attractors, due to the highly degenerate energy landscape. Earlier work has shown that different attractors can be reached by encoding input as a global external magnetic field (Jensen et al., 2018). Consequently, the system state forms a spatial representation of input history, i.e., exactly the kind of behavior sought in a reservoir.

2.3 Reservoir quality

A range of methods have been proposed to evaluate reservoir quality, ranging from benchmark tasks such as speech recognition and signal classification, to more generic measures such as memory capacity (Jaeger, 2002) and information processing capacity (Dambre et al., 2012).

In this work, we employ two generic measures related to signal classification, namely the kernel-quality and generalization-capability (Legenstein and Maass, 2005).

Kernel-quality is a measure of how well the reservoir is able to separate temporal input patterns. It is estimated by perturbing the reservoir with m different input signals. At the end of each signal, the reservoir states are recorded as the columns of an $n \times m$ matrix M_K where n is the number of reservoir nodes. Computing the rank K of this matrix gives a measure of kernel-quality (higher is better). If the kernel rank $K = m$, then it is guaranteed that *any* assignment of target outputs can be implemented by a linear readout. If $K < m$, the kernel rank can still be viewed as a measure of computational power, since it is a measure of the number of "degrees of freedom" the readout has available.

Kernel-quality is insufficient alone as a measure of reservoir quality. A complementary property is the reservoir's ability to generalize to new unseen

input signals. Generalization-capability is measured the same way as kernel-quality, except the $n \times m$ matrix M_G is now the reservoir states after seeing m *similar* input signals. We wish for the generalization rank G of this matrix to be low, meaning the reservoir states are similar and should generalize well.

A good reservoir maximizes K while minimizing G, hence a combined measure of computing capacity Q can be obtained by simply taking the difference: $Q = K - G$ (higher is better). Q is a measure of the usable nodes in the reservoir, i.e., nodes with both good kernel-quality and generalization-capability.

Theoretically, the information processing capacity is bounded by the number of reservoir nodes n (Dambre et al., 2012). Since the rank of a matrix is bounded by its smallest dimension, one should choose $m \geq n$ to avoid saturation of the measures before the theoretical limit. Thus $K, G \in [1, n]$ and $Q \in [1-n, n-1]$, where $Q = n-1$ indicates the best possible performance, while $Q \leq 0$ indicates a reservoir with no usable computing capacity.

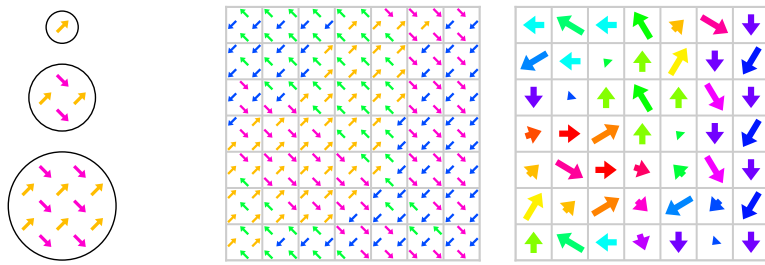
When comparing reservoirs with different numbers of nodes, the normalized rank r provides a measure of computational power per node: $r = R/n$ where R is the matrix rank. We denote the normalized versions of K, G and Q as k, g and q, respectively. Thus, $k, g \in [1/n, 1]$ and $-1 < q < 1$ with $q = 1 - 1/n$ indicating the best possible performance while $q \leq 0$ represents a reservoir with no usable capacity (Haynes et al., 2015).

3 Methods

3.1 Magnetic model

For our computational study, we use the flatspin ASI simulator, which enables fast simulations of dynamics in coupled spin systems (Jensen et al., 2020). In flatspin, magnets are modeled as point dipoles with binary state. Each dipole is affected by neighboring magnets through magnetic dipole-dipole interactions, as well as a global external field.

Dynamics in flatspin are deterministic, modeled as a series of single spin flips. A spin may flip (switch state) if the total magnetic field acting on it is sufficiently strong, i.e., exceeds its intrinsic coercive field, and is directed in the opposite direction of its magnetization.



(a) **(b)** **(c)**

Figure 2: (a) A reservoir node contains one or more spins: shown here are nodes with one, four and 12 spins. (b) 10×10 pinwheel ASI has 220 spins, shown here with a 7×7 grid superimposed. Summing the magnetization of the spins in each cell results in the coarse-grained view shown in (c). The length of the arrows indicates the magnitude of the magnetization.

The global parameter α scales the strength of the dipole-dipole interactions. A large value of α denotes a high degree of coupling between the spins. An increase in α is equivalent to reducing the lattice spacing between all magnets.

3.2 Input encoding

As input we consider temporal binary patterns, i.e., the input is a function $u(t) \in \{0,1\}$ for discrete time $t = 0..T$. For each input bit we cycle the external field at a fixed field strength H at an angle determined by the input bit: ϕ_0 for 0 and $\phi_1 = \phi_0 + 90°$ for 1. The 90° offset ensures both 0 and 1 will perturb the system with the same amount of force (due to the pinwheel geometry). To break symmetry, we set $\phi_0 = 7°$, which causes each input bit to affect magnets in one sub-lattice slightly more than the other. We use a small angle to still allow switching to occur in both sub-lattices.

3.3 Output granularity

As reservoir output we record the magnetization of the ASI. The number of reservoir nodes n depends on the granularity of observation (the level of "squinting"). Fig. 2a illustrates nodes containing different numbers of magnets: a single spin, four magnets and twelve magnets. Each group of

magnets results in two reservoir nodes, one for each vector component of the collective magnetization.

At the finest granularity, we resolve the binary state of individual spins, i.e., the number of reservoir nodes n equals the number of spins N.

To define coarse-grained nodes, we superimpose a regular $S \times S$ grid onto the ASI, as shown in Fig. 2b. The magnetization of the spins within each grid cell is then summed to produce an aggregate output vector, as shown in Fig. 2c. For an $S \times S$ grid, we obtain $n = 2S^2$ reservoir nodes, as each grid cell results in two nodes.

The grid won't necessarily align with the underlying ASI geometry, thus the number of magnets within each grid cell may vary. This can be seen in Fig. 2b, where some cells contain four magnets while others contain five.

A decrease in the number of grid cells results in an increase in state resolution, as nodes can take more possible values. Hence, a coarse-grained view offers more computational power per node, at the cost of fewer nodes.

When multiple magnets are aggregated, the reservoir state is effectively degenerate: there will be multiple spin configurations which produce the same vector sum.

3.4 Experiment setup

We consider systems of 10×10 pinwheel ASI, consisting of $N = 220$ stadium-shaped nanomagnets with dimensions $220\,\text{nm} \times 80\,\text{nm} \times 20\,\text{nm}$, and parameters $h_k = 200\,\text{mT}$, $b = 0.41$, $c = 1.0$, $\beta = 1.5$ and $\gamma = 3.9$.

Due to manufacturing imperfections there will always be variation in the coercive fields of the magnets. Hence, we apply a disorder of 5% to the coercive fields $h_k^{(i)}$ of each magnet i, i.e., the coercive fields are sampled from a normal distribution with mean h_k and standard deviation $0.05h_k$. We define an ASI *sample* as a set of N coercive fields $\{h_k^{(i)}\}$.

We start with an initially polarized ASI, such that the total magnetization is saturated towards the right (as illustrated in Fig. 1c). Next, the input signal is applied through the external field. For kernel-quality we use $m = 220$ random binary input signals, each 100 bits in length. For generalization-capability, we use $m = 220$ random binary input signals where the first 40 bits are random and the remaining 60 bits are equal across the signals. Hence

the generalization rank, at the end of the input signal, is a measure of how sensitive the system is to inputs older than 60 time steps.

In the following, we vary the strength of the external field H and the coupling strength α. For each experiment we generate 30 ASI samples, and take the average rank.

3.4.1 Full visibility

First, we consider reservoirs with full visibility of the 220 magnets as output ($n = 220$ reservoir nodes). For each ASI sample, we sweep the coupling strength α and the strength of the external field H, and measure the corresponding K, G and Q. We sweep 16 values of α in the range 3e−5 to 3e−3, which roughly corresponds to lattice spacings from 1000 nm to 215 nm. For each α value, we sweep 16 values of H in the range 66 mT to 81 mT.

3.4.2 Output granularity

Next, we investigate how the output granularity affects performance. Grids of size 1×1 to 10×10 are superimposed onto the ASI, resulting in $n = 2$ to $n = 200$ reservoir nodes. For each n we calculate the corresponding K, G and Q. When comparing performance across different number of nodes n, we use the normalized rank measures k, g and q. We maintain the same number of input patterns $m = 220$, i.e., independently of n.

4 Results

4.1 Full visibility

Fig. 3 shows the results of the parameter sweep of H and α, as heatmaps of the average kernel rank $\overline{K}$, generalization rank $\overline{G}$ and computing capacity $\overline{Q} = \overline{K} - \overline{G}$. Each cell in the heatmap is the average of the 30 different ASI samples.

All measures exhibit a ridge line in the H–α plane, which drops quickly for low α values. The ridge shows an apparent linear relationship between H and α, in terms of computational performance.

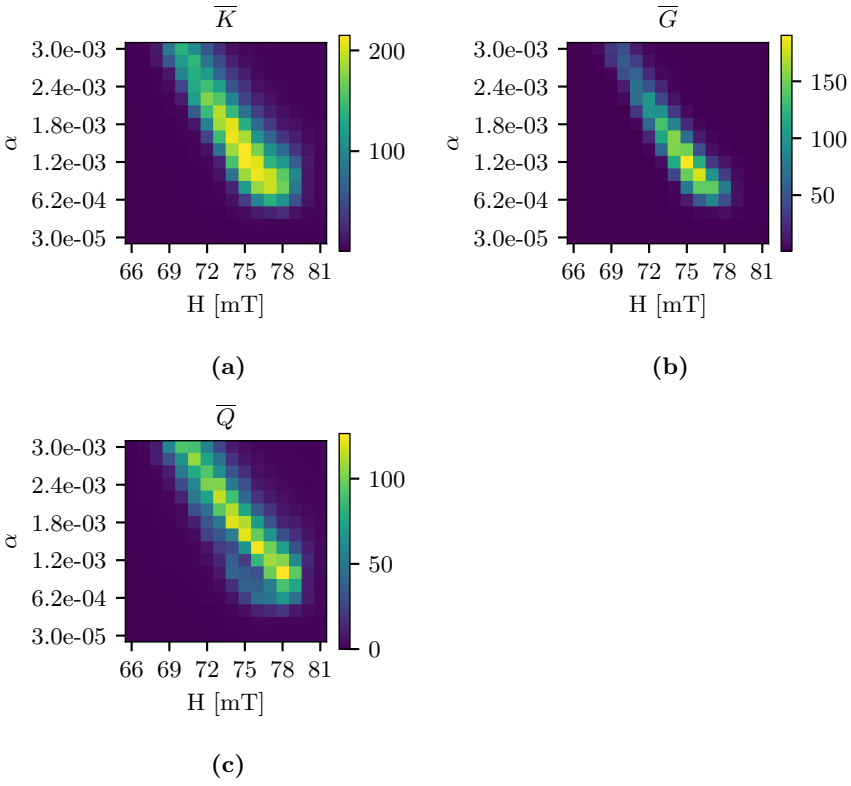


Figure 3: Average (a) kernel rank $\overline{K}$, (b) generalization rank $\overline{G}$ and (c) computing capacity $\overline{Q} = \overline{K} - \overline{G}$, as a function of the parameters H and α.

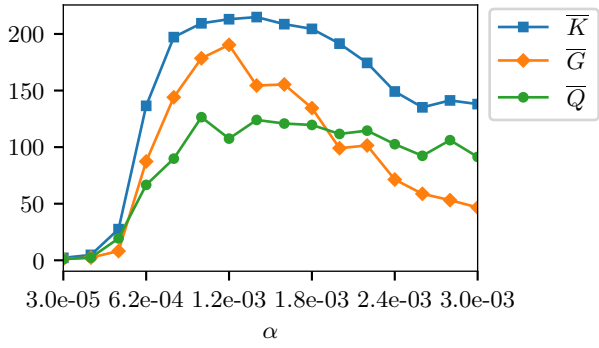


Figure 4: Average kernel rank $\overline{K}$, generalization rank $\overline{G}$ and computing capacity $\overline{Q}$, along the ridge lines of Fig. 3, i.e., for each α value, the highest value of $\overline{K}$, $\overline{G}$ and $\overline{Q}$ is plotted.

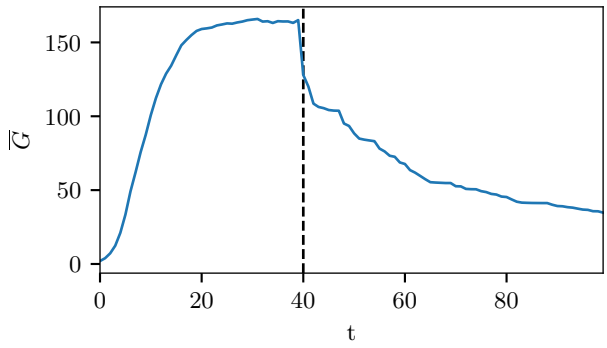


Figure 5: Average generalization rank $\overline{G}$ over time. The dashed line marks where the input signals transition from being different to being identical.

As can be seen in Fig. 3a, kernel rank $\overline{K}$ is generally high along the ridge. In Fig. 3b, a similar but thinner ridge is apparent for the generalization rank $\overline{G}$. The $\overline{K}$ and $\overline{G}$ ridges are in the same location of the H–α plane. In Fig. 3c, the ridge line of their difference $\overline{Q}$ is shifted slightly to the right.

Fig. 4 plots $\overline{K}$, $\overline{G}$ and $\overline{Q}$ along the ridge lines in Fig. 3, as a function of α, i.e., for each α value, the highest value of $\overline{K}$, $\overline{G}$ and $\overline{Q}$ is plotted. A general trend is a decline in both $\overline{K}$ and $\overline{G}$ as α is increased. $\overline{K}$ nearly saturates for $1\text{e}{-3} < \alpha < 2\text{e}{-3}$ with ranks as high as 215 on average (recall that the maximum rank is 220). The $\overline{Q}$ ridge, on the other hand, is fairly flat as a function of α, with an apparent maximum for $\alpha = 1.02\text{e}{-3}$ and $H = 78\,\text{mT}$. However, we note that the standard deviation of K (and hence Q) is significantly higher for large values of α.

Fig. 5 shows how $\overline{G}$ evolves over time, i.e., measured after being perturbed with each of the 100 input bits, for $\alpha = 1.02\text{e}{-3}$ and $H = 78\,\text{mT}$. Recall that after the first random 40 bits, the input signals are identical for the remaining 60 bits. As can be seen in Fig. 5, the average rank drops quickly at $t = 40$, after which there is a somewhat gradual decline. Inspecting the trajectories of G for the individual ASI samples reveals that there are variations in the behavior: for some samples, the rank drops quickly, while others exhibit a more gradual decline.

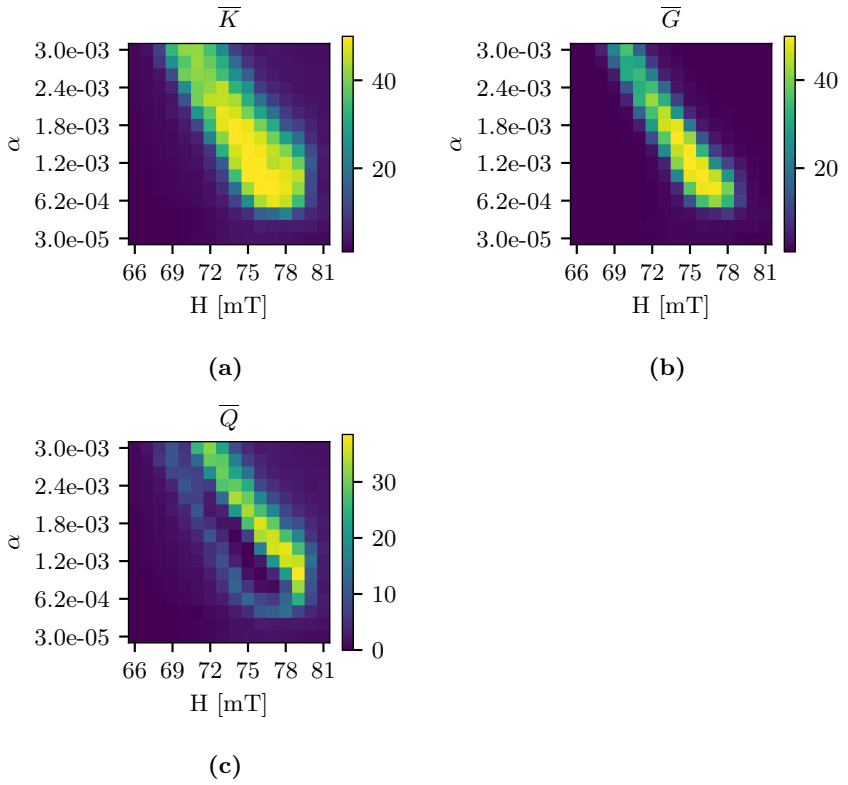


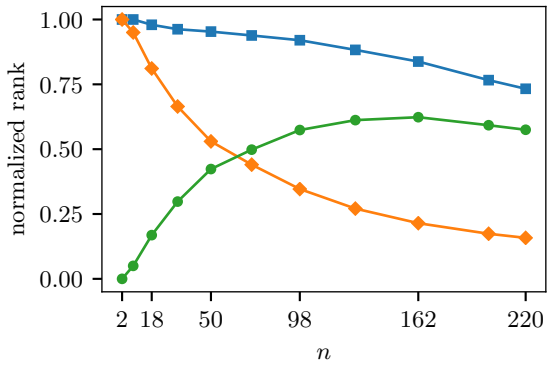
Figure 6: Average (a) kernel rank $\overline{K}$, (b) generalization rank $\overline{G}$ and (c) computing capacity $\overline{Q} = \overline{K} - \overline{G}$, as a function of the parameters H and α, using an output granularity of $n = 50$ nodes (5×5 grid).

4.2 Output granularity

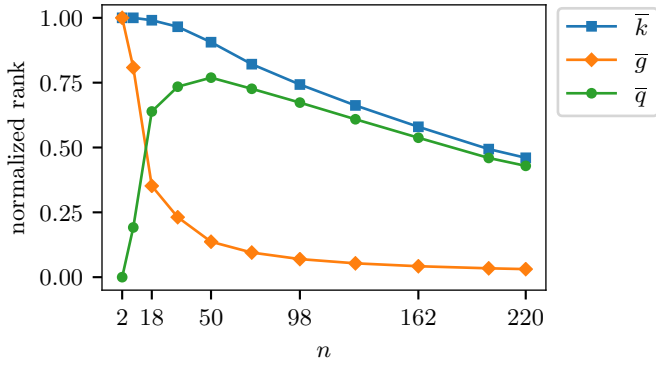
Fig. 6 shows similar heatmaps of $\overline{K}$, $\overline{G}$ and $\overline{Q}$, but using a coarse-grained output with a 5×5 grid ($n = 50$ nodes). Compared to the heatmaps using full visibility of all magnets (Fig. 3), both $\overline{K}$ and $\overline{G}$ exhibit wider ridge lines. As a consequence, the $\overline{Q}$ ridge is shifted diagonally towards higher H and α values. Saturation of $\overline{K}$ is still obtained for large regions of parameter space (here the maximum rank is 50).

Comparing the heatmaps from all granularities (not shown), a general trend is that, as output becomes more coarse-grained, the $\overline{Q}$ ridge line moves diagonally in the heatmaps towards higher H and α.

Fig. 7 shows the average normalized rank measures $\overline{k}$, $\overline{g}$ and $\overline{q} = \overline{k} - \overline{g}$, as



(a)



(b)

Figure 7: Average normalized kernel rank $\overline{k}$, generalization rank $\overline{g}$ and computing capacity $\overline{q} = \overline{k} - \overline{g}$, as a function of the number output of nodes n. The reservoir parameters are $\alpha = 1.02\text{e}{-3}$ with (a) $H = 78\,\text{mT}$ and (b) $H = 79\,\text{mT}$.

the number of reservoir nodes n are increased (resulting from the increasing number of grid cells). For completeness, the plots also include results with full visibility at $n = 220$.

Fig. 7a shows the measures for $\alpha = 1.02\text{e}{-3}$ and $H = 78\,\text{mT}$, i.e., the parameters with the best performance from the full visibility experiment. As can be seen, a decrease in n results in poor generalization $\overline{g}$ and hence a decrease in performance per node $\overline{q}$. There's an apparent peak of $\overline{q} \approx 0.6$ for $n = 162$.

Fig. 7b shows the same plot for an increased field strength of $H = 79\,\text{mT}$. With a stronger field, much better generalization is obtained. In this case, we observe an *increase* in $\overline{q}$ as n is decreased, with an apparent peak at $\overline{q} \approx 0.8$ for $n = 50$.

5 Discussion

5.1 Full visibility

Our investigation of the H–α parameter space demonstrates salient features of ASI reservoirs. For most values of α, there exists a corresponding critical field strength H, which is neither too weak (resulting in little activity), nor too strong (causing all magnets to switch).

Clearly, spin interactions play a crucial role in the formation of a complex dynamic response, since low α values result in poor kernel-quality. Intuitively, in an uncoupled system, the state of the spins will only be affected by the current input. Memory formation requires sufficient flow of information between the spins. If the coupling is too weak, the current state will be completely overwritten by new input, and all history of previous inputs is lost.

The saturation of kernel rank, shown in Fig. 3a, demonstrates excellent input separation for large regions of parameter space. In these cases, the ASI states contain sufficient information to discriminate between all the input signals. However, the reservoirs with the highest kernel-quality suffer from poor generalization-capability. High generalization rank is evidence of chaotic dynamics, where the initial difference in states persists for a long time.

The measure of computing capacity Q (Fig. 3c) exhibited a ridge line which, compared to the ridge lines of K and G (Fig. 3a and Fig. 3b), is shifted

slightly towards stronger fields (larger H). A stronger external field will consequently result in more spin flips per input, thus overwriting more of the system state. Indeed, chaotic reservoirs may still be used successfully, as long as the input is sufficiently strong to drive its dynamics out of the chaotic regime (Ozturk and Principe, 2005).

The decrease in kernel-quality as function of coupling strength α is expected, since the size of ferromagnetic domains increase with α, and hence there is less variation in the spin states. Smaller domains result in more diverse spatial patterns, and consequently a richer repertoire of input transformations. However, the parameter regions with the highest kernel-quality have very poor generalization-capability, resulting in poor overall performance. As a result, the capacity measure Q predicts no significant difference in performance between loosely and highly coupled systems. Still, the higher variance observed for large α values is evidence that the particular ASI sample, i.e., the set of coercive fields, plays a more important role for highly coupled ASIs compared to the loosely coupled systems.

The observed gradual decrease in generalization rank over time is clear evidence of fading memory, where past input is gradually forgotten over time. The plot in Fig. 5 is remarkably similar to the time-wise separation observed in neural microcircuits (Maass et al., 2004). Although there are variations in the behavior, the results indicate that ASI reservoirs can indeed exhibit the echo-state property.

5.2 Output granularity

Our results revealed that a change of output granularity affects the performance landscape in the H–α parameter space. Parameters which perform well with full visibility of all spins perform poorly with a coarse-grained view.

As the number of spins per node is increased, the areas in the parameter space with good performance move towards stronger fields and higher coupling (compare Figs. 3c and 6c). Interestingly, a small increase in field strength seems to be sufficient to improve performance under a coarse-grained view (Figs. 7a and 7b).

Under a coarse-grained view, one might expect that the more strongly coupled systems would have a benefit, since larger magnetic domains would still be visible without significant information loss. However, we find no evidence

of this in our results. In fact, the strongly coupled systems appear to perform worse, regardless of output granularity.

Since the computational capacity is bounded by the number of reservoir nodes (Dambre et al., 2012), it would seem like full visibility of all spins is always beneficial. However, in any physical reservoir there will be a cost associated with measurement of state, placing practical limits on the number of output nodes. Additionally, a coarse-grained view brings some additional benefits, which we discuss below.

The normalized rank measures (Figs. 7a and 7b) revealed that, for a system consisting of binary elements, the computing capacity per node $\overline{q}$ can be *increased* by combining multiple elements into one node. The increase can be attributed primarily to an increase in the degrees of freedom per node, i.e., as a node can take more possible values. This was confirmed by thresholding the aggregate values, effectively making the grid cells "super-spins", which resulted in a fairly flat $\overline{q}$ across the different granularities (not shown).

For a given number of reservoir nodes, it should be possible to maximize performance by scaling up the underlying system, while maintaining a fixed-size coarse-grained view.

Another potential benefit of a coarse-grained view is robustness: the output will be less sensitive to small differences in the underlying spin state. If a spin inadvertently flips, e.g., due to noise, its immediate effect will be small under a coarse-grained view. With full visibility, however, the readout may be more sensitive to a single spin flip.

The results show that, at least for pinwheel ASI, there is a great degree of freedom in choosing the output granularity. We may observe the system at a range of different scales, and still obtain good performance.

6 Conclusion

ASIs are promising computing substrates due to the wide variety of emergent behavior, which can be directly controlled by the system geometry. We have shown how the inherent properties of pinwheel ASI result in complex spatio-temporal patterns that can be readily exploited for computation. Our experiments demonstrate excellent computing capacity in terms of well-established reservoir quality measures. We find clear evidence of fading memory, suggesting the presence of the crucial echo-state property.

An exciting finding is that good performance can also be obtained with a coarse-grained metamaterial view of the system. Although the size of our magnetic system was fixed, our results indicate that ASI reservoirs are scalable, both in terms of the number of nodes as well as the computing capacity per node. The apparent flexibility in output granularity show that ASIs have computational properties at different abstraction levels, from the small-scale dynamics of simple spins, to the large-scale spatial patterns of the metamaterial. The fact that meaningful computations can be obtained with a very coarse-grained view of the substrate, shows that physical ASI reservoirs are not only possible, but also practical.

Acknowledgements

This work was funded in part by the Norwegian Research Council under the SOCRATES project, grant number 270961. Thanks to Arthur Penty for insightful comments and interesting discussions. Simulations were run on the NTNU EPIC GPU cluster (Själander et al., 2019).

References

Dambre, J., Verstraeten, D., Schrauwen, B., and Massar, S. (2012). Information Processing Capacity of Dynamical Systems. *Scientific Reports*, 2:514.

Furuta, T., Fujii, K., Nakajima, K., Tsunegi, S., Kubota, H., Suzuki, Y., and Miwa, S. (2018). Macromagnetic Simulation for Reservoir Computing Utilizing Spin Dynamics in Magnetic Tunnel Junctions. *Physical Review Applied*, 10(3):034063.

Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I., and Gauthier, D. J. (2015). Reservoir computing with a single time-delay autonomous Boolean node. *Physical Review E*, 91(2):020801.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. Technical Report 148, German National Research Center for Information Technology.

Jaeger, H. (2002). Short term memory in echo state networks. Technical Report 152, German National Research Center for Information Technology.

Jensen, J. H., Folven, E., and Tufte, G. (2018). Computation in artificial spin ice. In *ALIFE 2018: The 2018 Conference on Artificial Life*, pages 15–22, Cambridge, MA. MIT Press.

Jensen, J. H., Strømberg, A., Lykkebø, O. R., Penty, A., Själander, M., Folven, E., and Tufte, G. (2020). Flatspin: A Large-Scale Artificial Spin Ice Simulator. *arXiv:2002.11401 [cond-mat, physics:physics]*.

Legenstein, R. and Maass, W. (2005). What makes a dynamical system computationally powerful? In *New Directions in Statistical Signal Processing From Systems to Brain*, pages 127–154. MIT Press.

Leliaert, J., Dvornik, M., Mulkers, J., De Clercq, J., Milošević, M. V., and Van Waeyenberge, B. (2018). Fast micromagnetic simulations on GPU - Recent advances made with mumax3. *Journal of Physics D: Applied Physics*, 51(12).

Levis, D., Cugliandolo, L. F., Foini, L., and Tarzia, M. (2013). Thermal Phase Transitions in Artificial Spin Ice. *Physical Review Letters*, 110(20):207206.

Li, Y., Paterson, G. W., Macauley, G. M., Nascimento, F. S., Ferguson, C., Morley, S. A., Rosamond, M. C., Linfield, E. H., MacLaren, D. A., Macêdo, R., Marrows, C. H., McVitie, S., and Stamps, R. L. (2019). Superferromagnetism and Domain-Wall Topologies in Artificial "Pinwheel" Spin Ice. *ACS Nano*, 13(2):2213–2222.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Maass, W., Legenstein, R. A., and Bertschinger, N. (2004). Methods for Estimating the Computational Power and Generalizatoin Capability of Neural Microcircuits. In *Advances in Neural Information Processing Systems 17*, pages 865–872. MIT Press.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560.

Mengotti, E., Heyderman, L. J., Rodríguez, A. F., Nolting, F., Hügli, R. V., and Braun, H. B. (2011). Real-space observation of emergent magnetic monopoles and associated Dirac strings in artificial kagome spin ice. *Nature Physics*, 7(1):68–74.

Nakane, R., Tanaka, G., and Hirose, A. (2018). Reservoir Computing With Spin Waves Excited in a Garnet Film. *IEEE Access*, 6:4462–4469.

Nomura, H., Furuta, T., Tsujimoto, K., Kuwabiraki, Y., Peper, F., Tamura, E., Miwa, S., Goto, M., Nakatani, R., and Suzuki, Y. (2019). Reservoir computing with dipole-coupled nanomagnets. *Japanese Journal of Applied Physics*, 58(7):070901.

Ozturk, M. C. and Principe, J. C. (2005). Computing with transiently stable states. *Proceedings of the International Joint Conference on Neural Networks*, 3:1467–1472.

Prychynenko, D., Sitte, M., Litzius, K., Krüger, B., Bourianoff, G., Kläui, M., Sinova, J., and Everschor-Sitte, K. (2018). Magnetic Skyrmion as a Nonlinear Resistive Element: A Potential Building Block for Reservoir Computing. *Physical Review Applied*, 9(1):014034.

Själander, M., Jahre, M., Tufte, G., and Reissmann, N. (2019). EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure. *arXiv:1912.05848 [cs]*.

Skjærvø, S. H., Marrows, C. H., Stamps, R. L., and Heyderman, L. J. (2020). Advances in artificial spin ice. *Nature Reviews Physics*, 2(1):13–28.

Sklenar, J., Lao, Y., Albrecht, A., Watts, J. D., Nisoli, C., Chern, G.-W., and Schiffer, P. (2019). Field-induced phase coexistence in an artificial spin ice. *Nature Physics*, 15(2):191–195.

Stepney, S., Rasmussen, S., and Amos, M., editors (2018). *Computational Matter*. Natural Computing Series. Springer International Publishing.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123.

Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., Bortolotti, P., Cros, V., Yakushiji, K., Fukushima, A., Kubota, H., Yuasa, S., Stiles, M. D., and Grollier, J. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664):428–431.

ISBN 978-82-326-6942-4 (printed ver.)
ISBN 978-82-326-5163-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology