HAL
open science

Rhythms and oscillations : a vision for nanoelectronics

Damir Vodenicarevic

To cite this version:

Damir Vodenicarevic. Rhythms and oscillations : a vision for nanoelectronics. Micro and nanotechnologies/Microelectronics. Université Paris Saclay (COmUE), 2017. English. NNT : 2017SACLS518 . tel-01695747

HAL Id: tel-01695747

https://theses.hal.science/tel-01695747v1

Submitted on 29 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université PARIS-SACLAY

NNT : 2017SACLS518

Thèse de doctorat

UNIVERSITÉ PARIS SUD

Comprendre le monde, construire l'avenir

Rhythms and Oscillations: a Vision for Nanoelectronics

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°575 EOBE
Electrical, optical, bio : physics and engineering
Spécialité : électronique et optoélectronique, nano- et microtechnologies

Thèse présentée et soutenue à Orsay, le 15/12/2017, par

Damir Vodenicarevic

Composition du Jury :

Gilles Sassatelli Directeur de Recherche, CNRS/Université Montpellier 2 (LIRMM)	Président
Ian O'Connor Professeur des Universités, Ecole Centrale Lyon (INL)	Rapporteur
Alexandre Pitti Maître de Conférences, CNRS/Université Cergy-Pontoise (ETIS)	Rapporteur
Denis Crété Ingénieur de Recherche, UMR CNRS/Thales	Examinateur
Julie Grollier Directrice de Recherche, UMR CNRS/Thales	Examinatrice
Damien Querlioz CR1, CNRS/Université Paris-Sud (C2N)	Directeur de thèse
Nicolas Locatelli Professeur Agrégé en CPGE, Lycée Gustave Eiffel, Cachan	Invité

Thèse effectuée au sein du **Centre de Nanosciences et de Nanotechnologies**
de l'Université Paris-Sud
Bâtiment 220, rue André Ampère
91405 Orsay cedex
France

Ce travail a bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche
au titre du programme Investissements d'Avenir portant la référence
ANR-10-LABX-0035: Labex NanoSaclay

À la santé du colonel, tout particulièrement.

Acknowledgements

Many people contributed to the success of this work, and deserve to be acknowledged for their direct or indirect contributions.

First and foremost, I would like to thank my parents Lilijana and Muhamed Vodenicarevic, who fled a country at war with a one year-old baby to start a new life from scratch in France. You left everything behind for me, and despite the difficult context, you managed to give me the taste for science, and stay the best people and the best parents I know, and for all of that, I am proud of being your son. I am also infinitely grateful to Sabira Sabanovic and Martine Villard for investing enormous efforts and taking huge risks to allow me and my family to settle in beautiful France, and for always being so dedicated in helping people in need. Thanks to my cousin Emir Hrustic with whom I had a great childhood and learned many useful skills. Thanks to Clara Rabier for her support. Huge thanks to Farsane Tabataba-Vakili for supporting me in the end of my thesis.

This thesis also directly owes its quality to the people from the Integnano research group. First of all, I am extremely grateful to my director Damien Querlioz for his pedagogy, knowledge, methods, managing skills, understanding, patience, dynamism, attention, exemplary supervision, and his sincere will to make his students succeed in their thesis, but also in their lives. Then, I would like to thank Nicolas Locatelli who invested time and effort to teach me a lot about device physics, seeded many of the ideas of this work, and contributed directly to the contents of this thesis. Thanks to my "first generation" PhD officemate Alice Mizrahi for insightful, fun, and productive discussions, but also for amazing parties. Also, thanks to my other first generation officemate Adrien Vincent for the infinite amount of time he invested in helping me whenever I needed something, for high quality discussions, and for providing the LaTeX template of this thesis. Thanks to Joseph Friedman for fascinating discussions, mac and cheese, and nice experimental work time. Big kudos to my roommate Christopher Bennett for amazing discussions, and for all the exciting side projects, parties and trips that made my PhD time memorable. I would also like to thank the "new generation" of PhD students: Tifenn Hirtzlin and Maxence Ernoult for wonderful discussions, and keeping the group as lively and productive as I found it when I arrived. Thanks to Laurie Calvet for her advice, and cool brainstorming sessions. Big thanks to Liza Herrera Diez for cheerful and inspiring discussions, heartwarming parties, and beautiful paintings. Thanks to Jacques-Olivier Klein and Nicolas Vernier for their help with experiments, and with the rehearsals of my defense. Thanks to Eric Fadel, Guillaume Hocquet and Qifan Wu. I am also grateful to the people I met in the broader C2N laboratory, including Thibaut Devolder, Joo-Von Kim, Sohini Manna, Jérôme Larroque, Arnaud Bournel,

Brice Davier, Dafiné Ravelosona, Eric Cassan, Ivan Saychenko, Jérôme Saint-Martin, Philippe Dollfus, and my interim director Arnaud Bournel. Thanks to the C2N administrative personnel and the EOBE doctoral school including Marie-Pierre Caron, Laurence Stephen and Sophie Bouchoule.

Huge thanks to our collaborators from CNRS/Thales, in particular Julie Grollier who contributed greatly to the quality of this thesis, but also to Jacob Torrejon, Steven Lequeux, Daniele Pinna, Mathieu Riou, Philippe Talatchian, Miguel Romera, Flavio Abreu Araujo, Joao Sampaio, and Sören Boyn for extemely useful discussions and fun times. Thanks to A. Fukushima, K. Yakushiji, H. Kubota and S. Yuas from AIST Japan for fabricating the samples used in my experiments. Also big thanks to Fabien Alibart, Selina La Barbera, Omid Kavehei, Massimiliano Di Ventra, Manan Suri, Pierre Bessière, Jacques Droulez, Emmanuel Mazer, and Sandip Tiwari for interesting and productive discussions and collaborations. Thanks to Mark D. Stiles for inviting me to present my work at NIST.

Thanks to the people I fruitfully collaborated with at the Capo Caccia workshop, including Julien Martel, Giacomo Indiveri, Tobi Delbruck, Gabriela Michel, Rodney Douglas, Sergio Solinas, Saray Soldado, Christian Pehle, Alex Rast, Sergio Martin del Campo, Fredrik Sandin and Vincent Lorrain.

Thanks to all the pseudonymous Electrolab Hackerspace members, especially Zenos and Crafty. Thanks to Nadim Kobeissi for his underground courses.

Finally, I thank Ian O'Connor and Alexandre Pitti for carefully reading my manuscript and providing thorough and useful reviews. I also thank the other members of the jury: Denis Crété, Julie Grollier and Gilles Sassatelli for generously taking the time and effort to evaluate my defense, and ask important questions. Thanks to Nicolas Locatelli for his heartwarming presence during my defense. And thanks to all the people who came to support me.

Contents

Introduction **1**

1 The computational power of oscillations **5**

1.1 Coupled oscillations and synchronization 6

1.1.1 Intuitive definition of coupled oscillators and synchronization 6

1.1.2 Oscillations in the brain 9

1.2 Understanding auto-oscillator systems 13

1.2.1 Relaxation oscillators 13

1.2.2 Kuramoto oscillators and the intuition behind phase dynamics 14

1.2.3 Noise and stochasticity 22

1.2.4 Non-linear oscillators: the Landau-Stuart model 23

1.3 Available nano-oscillators 24

1.3.1 Relaxation nano-oscillators based on physical or chemical state switching 24

1.3.2 CMOS ring oscillators 25

1.3.3 Mechanical nano-oscillators 26

1.3.4 Josephson junction nano-oscillators 28

1.3.5 Magnetic nano-oscillators 29

1.3.6 Other types of oscillators 31

1.4 Computing with oscillators 32

1.4.1 Binary computing with oscillators 32

1.4.2 Image segmentation 32

1.4.3 Vertex coloring 34

1.4.4 Degree of matching computation 34

1.4.5 Auto-associative memories and Hopfield networks 35

1.4.6 Hetero-associative memories and pattern classification 37

1.4.7 Feed-forward artificial neural networks 38

1.4.8 Reservoir computing 40

1.5 Summary of the chapter 41

2 Using coupled oscillators for pattern classification **43**

2.1 Introduction 44

2.2 Defining a nano-oscillator based pattern classifier 45
2.3 A robust synchronization detection scheme 47
2.3.1 Schemes for evaluating synchronization patterns 48
2.3.2 Equivalence of the detection schemes 49
2.3.3 Pattern recognition and comparison of the readout schemes 50
2.3.4 Sensitivity of the readout schemes to noise and parameters 54
2.3.5 Choosing the right detector 58
2.4 Is this architecture nanodevice-ready ? 58
2.4.1 Behavior under the presence of noise 59
2.4.2 Effects of natural frequencies variability 60
2.4.3 Effects of coupling constants variability 61
2.4.4 Effects of coupling phase shifts 62
2.4.5 Effects of oscillator non-linearity 64
2.4.6 System scalability 65
2.4.7 Effects of geometrical constraints 67
2.4.8 Advantages and limitations of the architecture 70
2.5 Possible improvements and alternative architectures 71
2.5.1 Improving the architecture 71
2.5.2 An alternative “EEG-like” readout method 71
2.6 Summary of the chapter 73

3 Case of the spintronic vortex nano-oscillator 75
3.1 Introduction 76
3.1.1 Features of the vortex spin-transfer nano-oscillator 76
3.1.2 Description of the device 77
3.2 Defining a model of the device 78
3.2.1 The vortex spin-transfer oscillator model 78
3.2.2 Model implementation 79
3.2.3 CMOS-compatible natural frequency tuning 81
3.2.4 Synchronization to an oscillating signal 82
3.2.5 Limitations of the model and subsequent improvements 84
3.2.6 Adjusting the model to specific devices 84
3.3 Electrically coupling nano-oscillators in a circuit 85
3.3.1 Coupling oscillators in series 85
3.3.2 Analytical interpretation of the circuit 86
3.4 Link with general oscillator models 88
3.4.1 Coupled phase-amplitude equations 88
3.4.2 From Thiele to Landau-Stuart 90
3.4.3 Linear approximation: connection with the Kuramoto model 97
3.5 A circuit implementing the pattern recognition architecture 99

3.6 Summary of the chapter 101

4 Machine learning with oscillator-based architectures 103

4.1 Introduction 104
4.2 An offline learning algorithm for oscillator-based classification 105
4.2.1 Supervised machine learning principles 105
4.2.2 Defining the output of the oscillator-based classifier 106
4.2.3 Choosing an error function for oscillator-based classification 106
4.2.4 Differentiation of the error 108
4.2.5 Jacobian computation methods 109
4.2.6 Learning binary classification 114
4.2.7 Classification capabilities 115
4.2.8 Performance on a simple task 116
4.2.9 Freeing the coupling strengths 117
4.3 Making the learning more flexible 117
4.3.1 Extended architecture 118
4.3.2 Learning capabilities 119
4.3.3 Performing a canonical machine learning task 119
4.4 Discussion on the extended oscillator-based classifier 124
4.5 Alternative “EEG-like” readout and adapted learning approaches 125
4.5.1 Neural networks based on synchronized pairs of oscillators 125
4.5.2 EEG-like signal readout in the extended oscillator-based classifier 129
4.5.3 Definition of the classification architecture with EEG-like readout 129
4.5.4 Bio-inspired learning using genetic algorithms 130
4.5.5 Results on the Iris dataset 134
4.5.6 Remarks and improvements on the architecture 134
4.6 Summary of the architectures and offline learning methods 135
4.7 Towards full online learning 137
4.8 Summary of the Chapter 139

5 RNG and computing with stochastic oscillators 141

5.1 Introduction 142
5.2 Random bitstreams from superparamagnetic tunnel junctions 144
5.2.1 The stochastic behavior of superparamagnetic tunnel junctions 144
5.2.2 Optimizing the quality of random bits 147
5.2.3 Scaling capabilities in terms of speed and energy consumption 150
5.2.4 Sensitivity of the random bitstream generator to perturbations 153
5.2.5 Comparison table with other hardware random bitstream generators . . . 155
5.3 Using superparamagnetic tunnel junctions for unconventional computing 157
5.3.1 Introduction to stochastic computing 157

5.3.2 Performing Bayesian inference with stochastic computing using probabilistic bitstream generators . . . 158
5.3.3 Implementing Bayesian inference for message classification . . . 161
5.4 Conclusion . . . 163

Conclusions and future work **165**

List of publications and contributions **169**

A Genetic algorithm for learning with the "EEG-like" architecture **173**
A.1 Initialization . . . 174
A.2 Evaluation . . . 174
A.3 Selection . . . 175
A.4 Sexual reproduction and mutations . . . 175

B Effects of XOR whitening on bitstream probability and auto-correlation **177**
B.1 Auto-correlation . . . 178
B.2 Mean value . . . 180

Synthèse en français **187**

Bibliography **212**

List of Figures

1 The computational power of oscillations **5**

1.1 Fireflies synchronize their blinking. 8
1.2 Firefly species have different natural frequencies. 8
1.3 Schematic of a neuron. 9
1.4 Half-center oscillator. 10
1.5 Example electroencephalogram. 12
1.6 Simple example of a relaxation oscillator. 14
1.7 Phase lag and phase advance in two coupled Kuramoto oscillators 16
1.8 Two coupled kuramoto oscillators at different natural frequencies. 17
1.9 Mean frequencies of two coupled Kuramoto oscillators. 18
1.10 Complex behavior of Kuramoto oscillators on a 2D lattice. 21
1.11 Linear vs non-linear oscillator. 23
1.12 Oxide-based auto-oscillator. 25
1.13 CMOS ring oscillator. 26
1.14 Nano-electromechanical auto-oscillator. 27
1.15 Schematic of a magnetic tunnel junction. 29
1.16 Tunnel magneto-resistance effect. 30
1.17 Image segmentation with locally coupled oscillators. 33
1.18 Degree of matching measure. 35
1.19 Auto-associative memory example. 36
1.20 Classification by the degree of matching comparison. 38
1.21 A simple example of neural network for classification. 39
1.22 Working principle of reservoir computing. 40

2 Using coupled oscillators for pattern classification **43**

2.1 Reference oscillator-based pattern classification architecture 45
2.2 Dummy classification task with the reference architecture. 46
2.3 Schematic of the detectors. 49
2.4 Redout scheme comparison on a simple 3-oscillator system. 50

2.5 Evaluation of the detectors on the pattern recognition architecture. 51
2.6 Synchronization map filtering. 52
2.7 Influence of the coupling on detection. 54
2.8 Influence of noise on detection. 55
2.9 Influence of the detection threshold. 56
2.10 Influence of the detection time. 57
2.11 Effect of noise. 59
2.12 Effect of natural frequency variability. 61
2.13 Effect of coupling variability. 62
2.14 Effect of phase shift. 63
2.15 Effect of non-linearity. 65
2.16 Scaling of the architecture. 66
2.17 Effect of spacially decaying couplings. 68
2.18 Effect of spatially increasing phase shifts. 69
2.19 Maps of the EEG-like signals. 72

3 Case of the spintronic vortex nano-oscillator 75
3.1 Description of the vortex nano-oscillator. 77
3.2 Resistance oscillations under a current bias. 82
3.3 Current-based frequency tuning. 83
3.4 Synchronization to an external sinal. 83
3.5 Electrically coupling 2 nano-oscillators. 86
3.6 Electrically coupling N nano-oscillators. 86
3.7 Uncorrected Landau-Stuart model. 95
3.8 Coupling-corrected Landau-Stuart model. 96
3.9 Linear approximation with the Kuramoto model. 98
3.10 Simulation of the whole architecture with nano-oscillators. 100

4 Machine learning with oscillator-based architectures 103
4.1 Oscillator pair error contribution. 107
4.2 Block view of the offline oscillator-based classifier learning model. 108
4.3 Differentiable oscillator network operator 112
4.4 Diagram of the binary classifier 114
4.5 Response map during learning. 115
4.6 Classifying off-diagonal regions by tuning core natural frequencies only. 116
4.7 Maps after learning by tuning natural frequencies and couplings. 117
4.8 Extended binary classification architecture. 118
4.9 Basic off-diagonal classification with the extended classifier. 120
4.10 Classifying complex regions with the extended architecture. 121

4.11 Offline learning of the Iris classification task. 122
4.12 Iris-Versicolor classification rate vs number of oscillators. 123
4.13 Impact of noise on Iris-Versicolor classification rate. 124
4.14 Simple neural network. 126
4.15 Schematic of the oscillator-based neuron. 126
4.16 Iris test set classification using oscillator-based neurons. 128
4.17 MNIST test set classification rates in the presence of noise. 128
4.18 Extended oscillator-based classifier with EEG-like readout. 129
4.19 Evolution on a simple classification task. 132
4.20 A more complex 2D classification task, and influence of noise. 133
4.21 Vowel classification with STNOs (by Julie Grollier's team) 139

5 RNG and computing with stochastic oscillators 141
5.1 Description of the superparamagnetic tunnel junction. 145
5.2 Statistics of the measured bitstreams. 147
5.3 Generated random bitstream quality. 148
5.4 Sampling rate and readout circuitry. 151
5.5 Read disturb effect. 152
5.6 Sensitivity to environmental perturbations and crosstalk. 153
5.7 Crosstalk through dipolar interaction. 154
5.8 Stochastic multiplier compared to a classical digital one. 157
5.9 C-element symbol, equivalent circuit and truth table. 159
5.10 C-element based Bayesian inference with stochastic MTJ RNGs. 160
5.11 E-mail message classification with the measured bitstreams. 162
5.12 Energy consumption of spam detection. 163

Introduction

"Constraint inspires creativity."

Biz STONE

"TRADITIONAL *computing paradigms are currently reaching fundamental limits. Alternative computing approaches exploiting the rich dynamics of coupled oscillators could provide a solution to this problem.* "

FROM face and speech recognition, movie recommendations and text translation to autonomous vehicles, targeted advertising and fraud detection, there is a colossal demand for human-like "artificial intelligence" in this era of massive data processing.

Personal or corporate computers, hand-held devices and even everyday objects and appliances are increasingly required to go beyond logic and arithmetic operations that represent the realm of traditional computing, and to perform cognitive tasks in which humans usually excel. Recent breakthroughs in machine learning algorithms have opened a wide range of applications that were long thought to be out of reach for computers. Image recognition algorithms now outperform humans in several situations [1], and in 2016 a computer won against a human grandmaster at the strategic Go game [2].

Computer architectures and underlying complementary metal-oxide semiconductor (CMOS) transistor technologies have been optimized for decades to perform billions of successive logic and arithmetic operations per second with high precision and without error. At the architectural level, computers use separate units for memory and computation, which induces a communication overhead called the von Neumann bottleneck.

Most cognitive algorithms, however, have fundamentally different requirements. They function optimally in a massively parallel and distributed way, and are tolerant to noise and errors, which makes traditional computers ill suited for such algorithms. Efforts are being made to increase parallelism in computers with the advent of thousand-core graphics processing units (GPU), to gain efficiency by reducing floating point precision, and to accelerate certain operations used by cognitive algorithms using specialized hardware such as tensor processing units (TPU). Despite those improvements, supercomputers running machine learning algorithms may require kilowatts ($\approx 600\,\mathrm{kW}$ for the Go game) or more to perform tasks that the human brain executes with 20 watts [3] using unreliable neurons arranged in a parallel and distributed topology that fuses computing and memory.

With information technology representing 10% of the worldwide electricity consumption in 2013 [4], CMOS technology reaching scaling limits [5], and increasing demand for low-power cognitive computing in hand-held devices and Internet of things (IoT), these observations have revived interest into alternative, and often brain-inspired computing systems. Many of these alternative schemes employ dynamical systems as a computing substrate, exploiting the natural energy minimization in physical systems to perform computations inspired by the dynamics observed in the brain.

In particular, some approaches seek to leverage the dynamics of coupled oscillators for computing. Oscillatory phenomena are very common in nature, and have been observed at different scales in the human brain where they seem to have a computational role. Mathematical studies also show that even small networks of a few coupled oscillators exhibit complex dynamics including chaos, phase transitions and synchronizations. Moreover, recent advances in nanotechnology have allowed the fabrication of nano-scale, low-power and highly integrable oscillator devices. These advances have triggered a surge of interest into the design

of oscillator-based computing chips that could perform cognitive tasks efficiently by exploiting the rich physics of coupled nano-oscillators.

However, proposed oscillator-based computing systems are not yet competitive with traditional CMOS due to the limitations of their architectures, the intrinsic constraints of nano-oscillator technologies, and lack of powerful algorithms to run them in real-life contexts.

THIS thesis proposes and investigates new oscillator-based computing architectures compatible with current nano-oscillator technologies, and provides specially designed machine learning algorithms allowing them to process real-life cognitive tasks efficiently.

The first chapter of the thesis provides an intuition of oscillator dynamics, and their potential computational role in the human brain. The mathematical origin of the computational capabilities of coupled oscillators is then explained through simple models. Using these concepts, existing nano-oscillator technologies are detailed, together with their potential for oscillator-based computing. An overview of existing computing architectures using those oscillators is then provided.

The second chapter presents a promising oscillator-based pattern classification idea that was originally proposed in a purely mathematical context. A readout circuitry is designed and its robustness investigated in order to propose a full-featured oscillator-based pattern classification architecture. A thorough simulation analysis is then performed to investigate the compatibility of this architecture to the constraints of nano-technologies, as well as its scalability. The chapter ends by presenting possible improvements and alternative versions of the architecture.

The third chapter presents a particular nano-oscillator technology: the spintronic vortex nano-oscillator. It establishes a model of the device compatible with circuit simulation tools, and uses it to show the compatibility of the device with the requirements of the proposed oscillator-based classification architecture. A circuit coupling multiple oscillators to implement the oscillator-based classifier of Chapter 2 is then designed. A mathematical description of the system, together with circuit simulations are used to investigate the compatibility of its behavior to the one described in the case of generic oscillators in the second chapter.

The fourth chapter investigates learning algorithms that can be applied to the oscillator-based classifier in order to train it to solve real-life classification tasks. After introducing machine learning principles and terminology, the theory of the oscillator-based classifier architecture is translated into the framework of machine learning. A novel learning algorithm tailored to this architecture is proposed, and its operation is tested on typical machine learning tasks. Modifications of the architecture are then proposed to increase its computational power and their effect is investigated on different classification tasks. Algorithms for an alternative version of the architecture are also proposed and tested.

The fifth and last chapter focuses on a radically different approach. A true hardware random bitstream generator based on purely stochastic nano-oscillators is proposed and investi-

gated experimentally. The quality of the resulting bitstream is evaluated to confirm the validity of the approach, and the effects of different environmental perturbations are investigated. The generated random bits are then used in the context of probabilistic computing to feed a simulated architecture performing low-power e-mail classification as spam or not spam.

Chapter 1

The computational power of oscillations

> "If you have pendulum clocks on the wall and start them all at different times, after a while the pendulums will all swing in synchronicity. The same thing happens with heart cells in a Petri dish: they start beating in rhythm even when they're not touching one another."
>
> Henry Louis MENCKEN

"COUPLED *oscillations are ubiquitous in nature and follow complex synchronization dynamics. This Chapter explains how inspiration can be drawn from oscillations present in the brain in order to leverage modern nanoscale oscillators for efficient computing.* "

THIS CHAPTER defines the context of the thesis and its motivations. After introducing coupled oscillators and the complex synchronizations they exhibit, it presents existing nano-oscillator technologies capable of producing these behaviors on chip, and reviews architectures leveraging their dynamics to achieve unconventional computing tasks.

Coupled oscillations and synchronization

In 2000, people walking at similar paces on the newly opened London's Millennium bridge caused vibrations that spontaneously synchronized their step, which further amplified the vibrations until the bridge started to sway [6]. This phenomenon intuitively illustrates how oscillators (the humans walking) can synchronize when they interact (through the vibrations they induce on the bridge).

Such coupled oscillations and synchronizations are ubiquitous in the environment: clapping audiences spontaneously synchronize [7], circadian rhythms align to daylight cycles [8], swarms of crickets end up chirping in unison [9], metronomes on the same table synchronize [8], alternating current power grids stabilize by aligning their phases [10, 11]. Such phenomena are also observed at higher scales such as stock markets [8] and ecosystems themselves [12].

This section provides an intuitive introduction to coupled oscillations and synchronizations, as well as their potential role in the human brain.

Intuitive definition of coupled oscillators and synchronization

Resonators and auto-oscillators

All systems with a periodic or quasi-periodic behavior can be seen as generalized oscillators. Their frequencies are defined as the number of periods per second, and their state is defined by an internal value called the phase. The way in which their oscillations are sustained separate them into two main categories: resonators and auto-oscillators.

In resonators, oscillation is induced and maintained by a periodic source of power acting on a system at a given frequency that does not depend on the state of the system [8]. A simple example is a child on a swing being pushed by a blind parent, that is at a constant frequency and regardless of the position of the swing. The swing then exhibits sustained oscillations only when the frequency of the pushes is close to its resonant frequency, which is defined by the resonator parameters such as the length of the swing ropes. Because they amplify the oscillations of a periodic source around their resonant frequency, resonators are mainly used as amplifiers or filters. For example guitar bodies resonate to amplify the sound of the strings, while RLC filters can discard frequencies that are far from their resonance. However, they always require a modulated power source to oscillate.

On the contrary, auto-oscillators only require a constant power source but its effect de-

pends on the state of the oscillator, which allows sustained oscillations. The net effect of the power source is effectively modulated by the oscillator itself, and an isolated auto-oscillator exhibits self-sustained oscillation at its natural frequency which is defined by its parameters [8]. A typical example is the battery-powered pendulum clock which triggers a magnetic push every time it reaches an extremal position. Other examples of auto-oscillators include direct current motors, menstrual cycles, neurons, or blinking fireflies [8]. Note that an auto-oscillator can be built by coupling a resonator with a feedback system.

Oscillator coupling and synchronization

It is convenient to introduce the concepts of oscillator coupling and synchronization through the example of fireflies. Male individuals from most firefly species use periodic light signals (bioluminescent blinking) to signal their presence to females. It has been observed that the males from a given species quickly synchronize their signals and start blinking in unison to allow females of the same species to locate them more efficiently [13]. Each blinking firefly can be modeled as an auto-oscillator as it has an internal self-sustained periodic clock advancing at its natural frequency and making the firefly blink at the end of each period. Fireflies from the same species have similar natural frequencies. As shown in Figure 1.1, a firefly can perceive the blinking of neighboring fireflies and every time it perceives a signal, it takes an action depending on the current state of its own clock. If it happens in the first half-period, it means that the firefly blinked before its neighbor, to which it responds by slightly nudging its clock back in time to reduce its advance. On the contrary, if a neighboring flash is detected during the second half-period, the clock is nudged forward in time to catch up and blink slightly sooner the next time [8, 14]. Through this mechanism, neighboring fireflies end up synchronizing until all the males of the species in the area blink in unison. This is a typical example of coupled oscillators that synchronize in frequency locking (they blink at the same frequency), and phase alignment (they blink at the same time).

The amplitude by which firefly i nudges its clock when firefly j flashes corresponds to the coupling strength from oscillator j to oscillator i. Typically, the stronger the coupling strength, the more oscillators interact, and the more effectively they can synchronize. The coupling strength is a critical parameter as it determines the dynamics of an ensemble of coupled oscillators. For example, heart pacemaker cells need to pulsate in perfect synchrony to avoid fibrillation and are therefore strongly coupled, unlike some areas in the brain where excessive synchronization between neurons causes epilepsy [9].

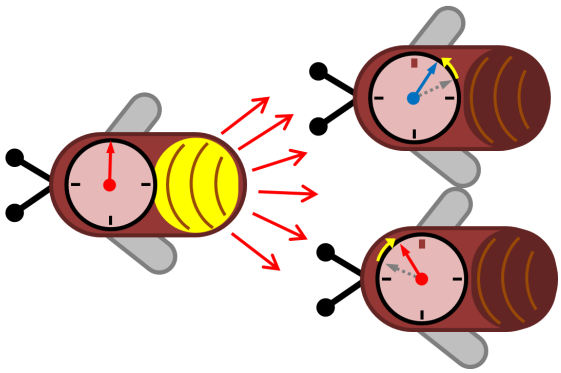


Figure 1.1: The internal clock of the firefly on the left reaches the end of a period which causes the firefly to blink. When they perceive it, the neighboring fireflies on the right adjust their internal clocks: the top one is in advance so it nudges its clock back in time while the bottom one lags behind and nudges its clock forward. This brings them closer to synchrony with the firefly that blinked.

Another important concept in oscillator networks is natural frequency mismatch. When two fireflies have clocks running at different speeds, they behave like auto-oscillators with different natural frequencies. The difference between their natural frequencies is called the natural frequency mismatch. At low mismatch, synchronization is harder but the two oscillators can still synchronize by locking at a common frequency. However, synchronization becomes impossible above a threshold mismatch value. In the case of fireflies, this threshold has evolved so that males of the same species can synchronize despite small variations in their clock speeds, but Figure 1.2 shows that different species have very distinct natural frequencies, thus leveraging this property to minimize interference.

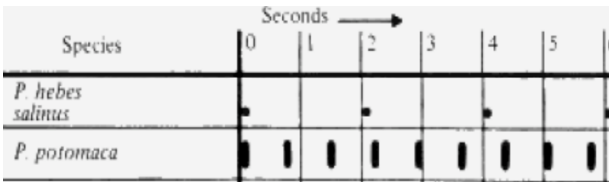


Figure 1.2: Time trace of the flashes recorded on two different firefly species. The two species have distinct natural frequencies which prevents synchronization between them. Reproduced from [15].

Even though the firefly example provides the basic intuition behind coupled oscillators, it represents a simple case. Networks of oscillators coupled in more complex topologies have been shown to exhibit advanced synchronization behaviors, chaos and multi-scale oscillation states [16, 17]. Many of these behaviors have also been observed in the human brain, which suggests that the brain could be using these complex dynamics for functional purposes.

Oscillations in the brain

From single neuron fluctuations to whole brain rhythmic signals, oscillations have been observed in the brain at multiple scales and linked to its activity.

Neurons and synapses

Neurons are living cells thought to be the main information processing units of the brain. The human brain contains a network of about 100 billion highly interconnected neurons [18], which integrate, transform and transmit electrochemical signals from multiple neurons or sensory organs, towards multiple other neurons or motor organs. This neural network is highly recurrent and exhibits diverse connectivity topologies at multiple scales. Neurons are connected to each other through nano-scale electro-chemical channels called synapses through which information typically travels unidirectionally, from the pre-synaptic neuron to the post-synaptic neuron.

Figure 1.3 provides a simple high level schematic of a neuron and shows its three main functional parts: the synapses, the dendrites, and the axon. Dendrites are tree-shaped ramifications of the cell body that receive information from other neurons through synapses and convey it to the cell. The axon is a single long projection of the cell that transmits signals from the cell body to post-synaptic neurons, and therefore acts as an outbound information channel.

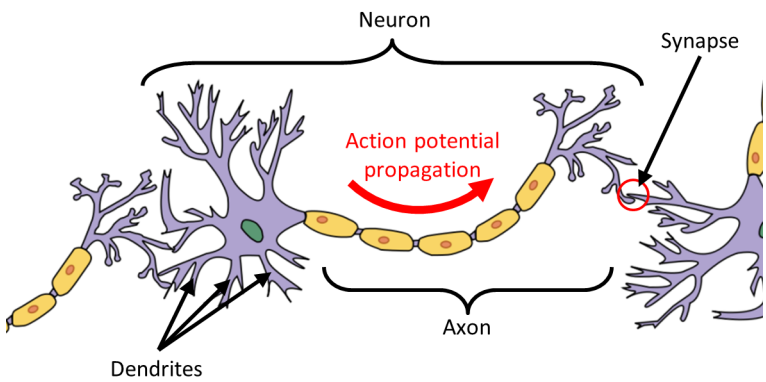


Figure 1.3: Schematic centered on a neuron connected to two other (cropped) neurons. The axon visible on the left can transmit an action potential to the neuron in the middle through a synaptic connection. This event is then channeled by the dendrites of the neuron in the center and integrated in its soma, along with other events received from other synapses. This integration can induce an action potential that is then transmitted by the axon towards the dendrites of the neuron on the right.

More precisely, the neuron is an excitable cell that maintains a gradient of ionic concentrations across its outer membrane, and therefore a voltage called membrane potential. When a limited voltage perturbation occurs, the resting state potential is progressively restored by ion pumps. However, when a sufficiently important positive voltage perturbation (depolarization) occurs, an electrochemical pulse called an action potential is generated, quickly travels

through the axon, and activates the synapses at its extremity. This emission of an action potential is called firing or spiking. As they are connected to the dendrites of post-synaptic neurons, those activated synapses then perturb the voltages of these neurons in turn. If the combined depolarization induced by all its pre-synaptic neurons is sufficient, a post-synaptic neuron fires as well. Once a neuron has fired, it enters a depletion period called the refractory period and during which it is unable to fire. After the refractory period, the neuron returns to its resting state [19].

Synapses can be excitatory and contribute to post-synaptic neuron firing, or inhibitory and prevent the post-synaptic neuron from firing. The amplitude of the perturbation they induce is also an important factor as different synapses have different weights in the integration process of post-synaptic neurons. The total effect of an activated synapse on its post-synaptic neuron varies from synapse to synapse, and can also depend on brain-scale hormonal signals or on the types and states of pre and post-synaptic neurons [19, 20]. Furthermore, the properties of synapses and neurons can change on longer time scales as part of the learning capabilities of the brain [19, 21].

The interplay of those excitatory and inhibitory phenomena in complex and recurrent neural networks gives rise to oscillations at multiple scales that could be a means of information processing.

Neural oscillations

Periodically firing neurons are common in human and animal nervous systems, and all neurons can be seen as auto-oscillators as they fire periodically when subject to a constant stimulus. However, certain neurons do not require external stimulation to fire periodically. Ensembles of interacting neurons can also show auto-oscillatory group behavior at a higher scale.

Neurons of a particular class, called endogenous oscillators, spontaneously fire periodically due to fluctuations of their own membrane potentials, and continue to oscillate when isolated from other neurons. These neurons behave like auto-oscillators, and their internal clocks can be accelerated or slowed down by other neurons, thus allowing synchronization [22–24].

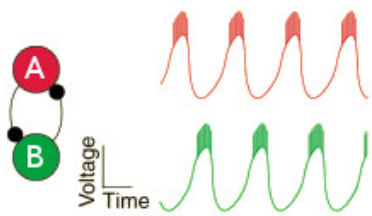


Figure 1.4: Schematic of a half-center oscillator composed of two coupled neurons firing alternatively, and the time traces of their membrane voltages. Adapted from [25].

In other cases, the interaction between multiple neurons gives rise to higher scale auto-oscillatory behaviors. A well-studied example is the half-center oscillator, a simple network of

two coupled neurons commonly found in the nervous systems of animals [26]. Figure 1.4 shows a half-center oscillator composed of neurons A and B. The effective interactions between the two neurons are both excitatory and inhibitory. When neuron A fires, it inhibits neuron B, preventing it from firing. However, this incoming spike also triggers a slow depolarization process in neuron B, eventually leading to neuron B firing. The same process then happens to neuron A. The membrane voltages of the two neurons plotted in Figure 1.4 show the periodic and alternating behavior of the two neurons. Half-center oscillators behave like auto-oscillators, and in the case when external inhibitory or excitatory signals are applied to their neurons, their frequencies decrease or increase respectively, which allows coupled half-center oscillators to interact and synchronize [25, 27].

At higher scales, oscillations involving large numbers of neurons can also be studied by measuring mean electric or magnetic fields around given areas of the brain (electroencephalography/EEG, magnetoencephalography/MEG) or by detecting increased blood supply resulting from local neural activity (functional magnetic resonance imaging, positron emission tomography, near-infrared spectroscopy). These techniques allowed the observation of both local oscillations and disparate groups of neurons oscillating in synchrony throughout the brain [28].

Figure 1.5 presents oscillations measured by electroencephalography (EEG) by detecting large scale oscillations of the average electric potential of entire areas of the brain. The observed oscillations can be used to detect the awareness state, sleep cycles, diagnose epilepsy and brain damage [29], or even control robots by thought [30]. This suggests that oscillatory neural activity is tightly linked to the function of the brain.

One example of a well studied neural oscillator is the circadian clock, which allows humans and other mammals to keep track of the time of day. The circadian clock is an ensemble of about 20,000 neurons of which many fire in synchrony and generate a 24-hour rhythm. This network behaves as an auto-oscillator that actively synchronizes to day-night cycles [31]. Other important examples of pace keeping auto-oscillators observed in the nervous system of most vertebrates are central pattern generators. They consist of an ensemble of neurons, often including endogenous and half-center oscillators, that oscillate at a tunable natural frequency to coordinate rhythmic motor actions, including for example swimming and walking. Their operation involves highly diverse and tunable synchronization capabilities [24].

Oscillatory behaviors are also associated to advanced cognitive activity and data processing. For example, gamma oscillations, which are 30 - 100 Hz neural oscillations observed in large areas of the brain, were linked to attention, memory and motor mechanisms [32–34]. These observations have motivated an important research effort to understand the way information could be encoded and processed by neural oscillations.

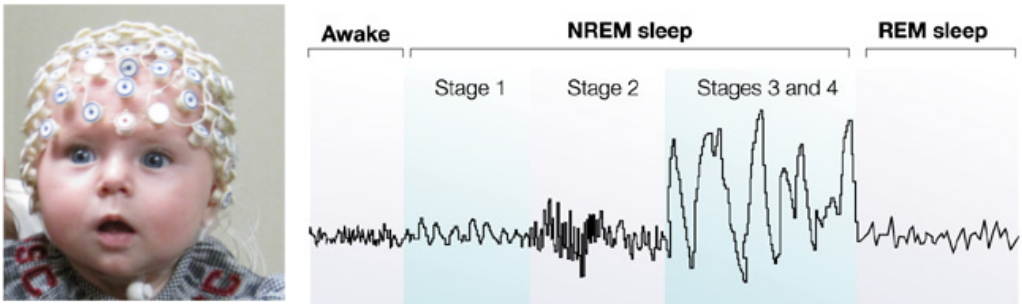


Figure 1.5: An example of electroencephalogram (EEG): time signal of the voltage measured on the scalp of a sleeping patient, showing different oscillatory patterns at different sleep stages. Credits: Univ. South Carolina Infant Development Laboratory and Nature Group.

Oscillator-based information processing in the brain

Contrary to the transistors used by computers, neurons are slow (hundreds of activations per second at most) but activate in a massively parallel way, show a high degree of variability as each neuron is substantially different from the others, and appear to be noisy and unreliable. It was shown that synapses have a probabilistic nature and can fail to transmit a signal more than 90% of the time [35]. Moreover, significant random noise has been observed in neural activity [36]. The way the brain robustly encodes and processes information is therefore fundamentally different from traditional computers, and coupled oscillations and synchronizations are often considered as one of the mechanisms used by the brain to achieve this purpose [19].

It is likely that multiple coding schemes are used by the nervous system. The simplest one is rate coding which consists in encoding an analog value as the spiking frequency of neurons. Early experiments have shown that the sensory neurons of a muscle produce spikes at a frequency that increases with the weight applied on the muscle [37]. This coding scheme is believed to be very common in sensory and motor neurons, and is also present in multiple other areas of the brain [19].

Information can also be encoded in the joint activity of an ensemble of neurons sensitive to different stimuli (population coding). It has been observed for example that the synchronization of different subsets of neurons encodes frequency-domain information when listening to spoken vowels [38].

Temporal coding, on the other hand, appears to encode information in the timing of emitted spikes, and phase-of-fire coding in particular can represent information as the timing of a spike relative to a background oscillation. Activity consistent with this mechanism appears in the hippocampus of mice and is thought to be a means of encoding the distance to visually observed landmarks. It is known as the theta precession. A background neural theta oscillation (4 - 8 Hz) is present in this area of the brain. As the mouse approaches the landmark, some of its neurons start to fire, and the closer it gets, the closer their spikes align with the background theta oscillation [20, 39].

Those neural coding schemes usually involve large populations of neurons which makes them statistically robust to individual neuron variability, noise and lack of reliability. Moreover, oscillators made of large synchronized ensembles of neurons are less subject to noise and behave in a more stable and deterministic way than single neurons [40].

However, noise and probabilistic behaviors can also be seen as assets for efficient data processing in the brain [36]. Studies have shown that synaptic stochasticity can improve information transmission [41], filter redundant information [42], and that noise can induce or improve the synchronization of neural oscillations [43–47].

Coupled oscillations in the brain appear to be linked to data processing tasks, which suggests that the brain could be leveraging the complex dynamics of oscillator networks for computational purposes. This is a major source of inspiration for the design of neuro-inspired unconventional computing architectures but requires an understanding of oscillator theory, as well as oscillator network modeling and simulation techniques.

Understanding auto-oscillator systems

In 1665, Christiaan Huygens first recognized the phenomenon of oscillator synchronization [8]. Since then, important theoretical and experimental research efforts have been deployed to understand and theorize oscillators and systems of coupled oscillators. As a result, various types of oscillators showing different behaviors have been identified, modeled and simulated.

Relaxation oscillators

Relaxation oscillators as defined by van der Pol [48] are a very common type of oscillator involving dynamics at two distinct time scales: each period contains intervals of slow (integration) and fast (firing) motion. Their oscillations typically do not resemble a smooth sine wave, but a sequence of pulses. A pedagogical example of a relaxation oscillator is the water tank presented in Fig. 1.6(a). The tap water fills the tank, progressively increasing its water level (integration). When it reaches a threshold, the tank is emptied quickly (firing), and the cycle repeats. The water outflow follows a periodic spike pattern as shown in Fig. 1.6(b).

The firefly example in Section 1.1.1.2 is also a relaxation oscillator: the internal clock of each individual runs until a certain threshold that resets it and triggers a short light pulse, which gives rise to periodic flashes.

Relaxation oscillator models applied to neuroscience are called integrate-and-fire oscillator models. The auto-oscillator neurons mentioned in Section 1.1.2.2 are examples of integrate-and-fire neural oscillators: their dynamics alternate between a slow voltage integration time and, after a threshold is reached, a sudden (2ms) action potential that resets the voltage, generating a periodic spike train in the process.

Relaxation oscillators have been used to model various phenomena such as geysers, ther-

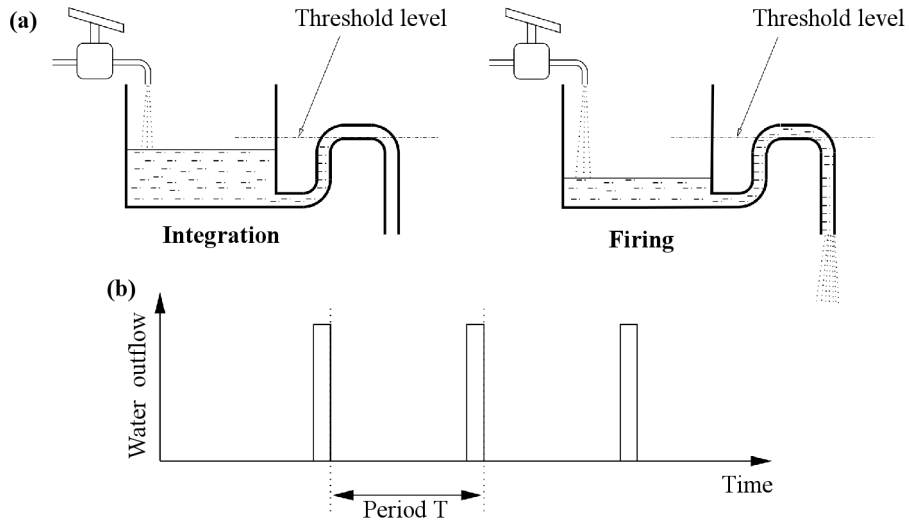


Figure 1.6: A simple example of a relaxation oscillator. (a) Schematic of the water tank oscillator. (b) Water outflow as a function of time. Adapted from [8].

mostats [49] or heartbeats [50]. Because their dynamics span multiple time scales and can display discontinuities, their analysis and modeling can be impractical. Their behavior has therefore often been approximated by harmonic oscillator models otherwise used to describe sine-shaped oscillations [51]. The most widely used and generic harmonic oscillator model is the Kuramoto model.

Kuramoto oscillators and the intuition behind phase dynamics

The Kuramoto model [52] is one of the most widely used and generic harmonic oscillator models, and is famous for admitting analytical solutions in certain cases (see details in Section 1.2.2.3). It captures the two main properties of auto-oscillators: the natural frequency and inter-oscillator coupling. Despite its simplicity, it is able to predict advanced behaviors in complex networks of oscillators and was successfully used to describe a wide range of oscillators such as chemical reactions, firefly blinking, or earthquakes [53–57]. In particular, it is an important tool in neuroscience where it was used to model multiple neural oscillation phenomena including central pattern generators and circadian clocks [8, 51, 58, 58–61]. The Kuramoto model is therefore a valuable approach for modeling networks of oscillators in order to leverage their computational power.

Constructing the Kuramoto model

The Kuramoto model can be readily constructed from the basic intuition behind coupled auto-oscillators provided in Section 1.1.1. As a starting example, an auto-oscillator A can be seen as a clock with a needle continuously rotating anti-clockwise in a smooth fashion. The angle of the needle θ_A (in radians) represents the phase of the oscillator. When nothing disturbs it, the

clock runs at a constant number of turns per unit time: its natural frequency f_A^0 (in Hz), which is an intrinsic property of the oscillator. The evolution of its phase can then be written as:

$$\frac{\dot{\theta}_A}{2\pi} = f_A^0 \quad \text{with} \quad \theta_A(t=0) = \theta_A^0, \tag{1.1}$$

where $\dot{\theta}_A$ is the time derivative of the phase of oscillator A, and therefore its angular frequency. θ_A^0 is the initial phase of the oscillator (the angle of its needle at $t = 0$). This is the Kuramoto model for one oscillator.

Note : Phase is free

Note that the phase increases linearly with time, eventually going beyond 2π and reaching arbitrarily large values. Even though $\theta = \pi$ and $\theta = \pi + 2\pi$ correspond to the same state of the system, resetting the phase at each period would involve a discontinuous jump in the value of θ, and hamper its smooth description as well as the proper definition of its time derivative $\dot{\theta}$. This implies that the phase of the oscillator effectively accumulates over time.

The next step in this construction is to take into account the coupling between two interacting oscillators A and B in order to capture their synchronization capabilities. As presented in the simple firefly example in Section 1.1.1.2, the clock B is expected to accelerate clock A when it lags behind B, and to slow it down when it is ahead of B. As the two clocks are periodic, deciding whether clock B lags behind or is ahead of clock A implies evaluating the sinus of the phase difference $\sin(\theta_B - \theta_A)$. Figure 1.7 shows that if the phase difference lies in the half-period $\sin(\theta_B - \theta_A) > 0$, then A lags behind B and needs to accelerate (positive contribution to $\dot{\theta}_A$), while if it lies in the other half-period $\sin(\theta_B - \theta_A) < 0$, then A is ahead of B and needs to slow down (negative contribution to $\dot{\theta}_A$). These contributions bring $\sin(\theta_B - \theta_A)$ closer to 0, in which case the two clocks show the same time. The same situation applies to the clock B, and the two oscillators can synchronize.

The equations can then be rewritten to include this coupling term which yields the two-oscillator Kuramoto model:

$$\begin{cases} \frac{\dot{\theta}_A}{2\pi} = f_A^0 + k_{A,B}\sin(\theta_B - \theta_A) \\ \frac{\dot{\theta}_B}{2\pi} = f_B^0 + k_{B,A}\sin(\theta_A - \theta_B) \end{cases} . \tag{1.2}$$

$k_{i,j}$ (in Hz) is the coupling strength from oscillator j to oscillator i which quantifies the amplitude of the frequency boost that oscillator j can cause on oscillator i. The higher $k_{A,B}$ is, the more clock B influences clock A. f_A^0 and f_B^0 are the natural frequencies of oscillators A and B respectively: they are the frequencies at which they oscillate when they don't interact with each other (when $k_{A,B} = k_{B,A} = 0$).

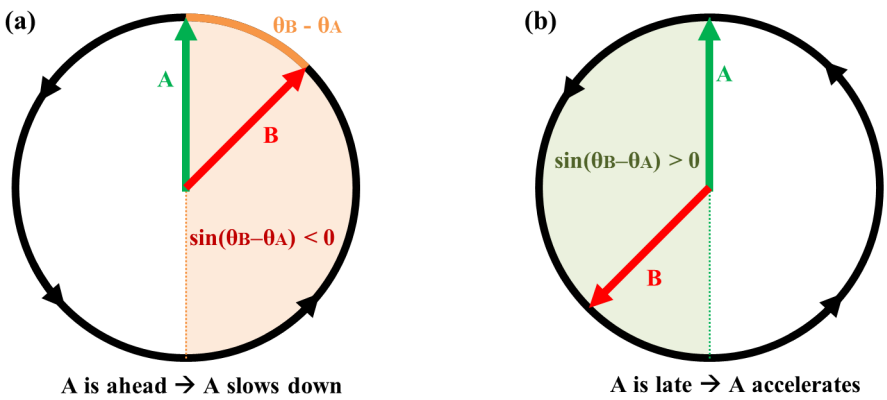


Figure 1.7: Phase response of an oscillator A subject to the influence of another oscillator B in the Kuramoto model. **(a)** When the phase difference $\theta_B - \theta_A$ lies in the half-period $\sin(\theta_B - \theta_A) < 0$, it means that oscillator A is ahead of B and slows down. When the phase difference lies in $\sin(\theta_B - \theta_A) > 0$, A is lagging behind B and therefore it accelerates.

This approach generalizes to a network of N coupled oscillators, which implies N phase equations, one per oscillator. The phase equation for oscillator i then reads:

$$\frac{\dot{\theta}_i}{2\pi} = f_i^0 + \sum_j^N k_{i,j} \sin(\theta_j - \theta_i) \,. \tag{1.3}$$

In that case, $k_{i,j}$ is the $N \times N$ coupling matrix of the network, which defines the coupling strength from each oscillator j to each other oscillator i in the network. This model is the simplest form of the N-oscillator Kuramoto model and successfully describes the dynamics of many coupled auto-oscillators.

Synchronization of two Kuramoto oscillators

To provide a better understanding of the phase dynamics in Kuramoto networks, I have simulated the simple case of the two coupled Kuramoto oscillators shown in Figure 1.8, following equations 1.2, and for different natural frequencies. The couplings are symmetric and defined as $k = k_{A,B} = k_{B,A} = 10$MHz. This two-oscillator case gives important insight into the influence of natural frequency mismatch and into the principles of synchronization and entrainment.

Figure 1.8(b) shows the instantaneous frequencies of the two oscillators $\frac{\dot{\theta}_A(t)}{2\pi}$ and $\frac{\dot{\theta}_B(t)}{2\pi}$ as functions of time in full lines. The dotted lines represent their natural frequencies $f_A^0 = 200$MHz and $f_B^0 = 100$MHz. In this first example, the instantaneous frequencies of the two oscillators fluctuate in time around their natural frequencies due to the coupling term in equations (1.2). However, as in the case of fireflies (Section 1.1.1.2), the two oscillators can't reach a common frequency because their natural frequencies differ sensibly ($f_A^0 - f_B^0 = 100$MHz).

In Figure 1.8(c), the natural frequency mismatch is lower: $f_A^0 - f_B^0 = 30$MHz. The instantaneous frequencies of the two oscillators appear to be more efficiently pulled together, as they

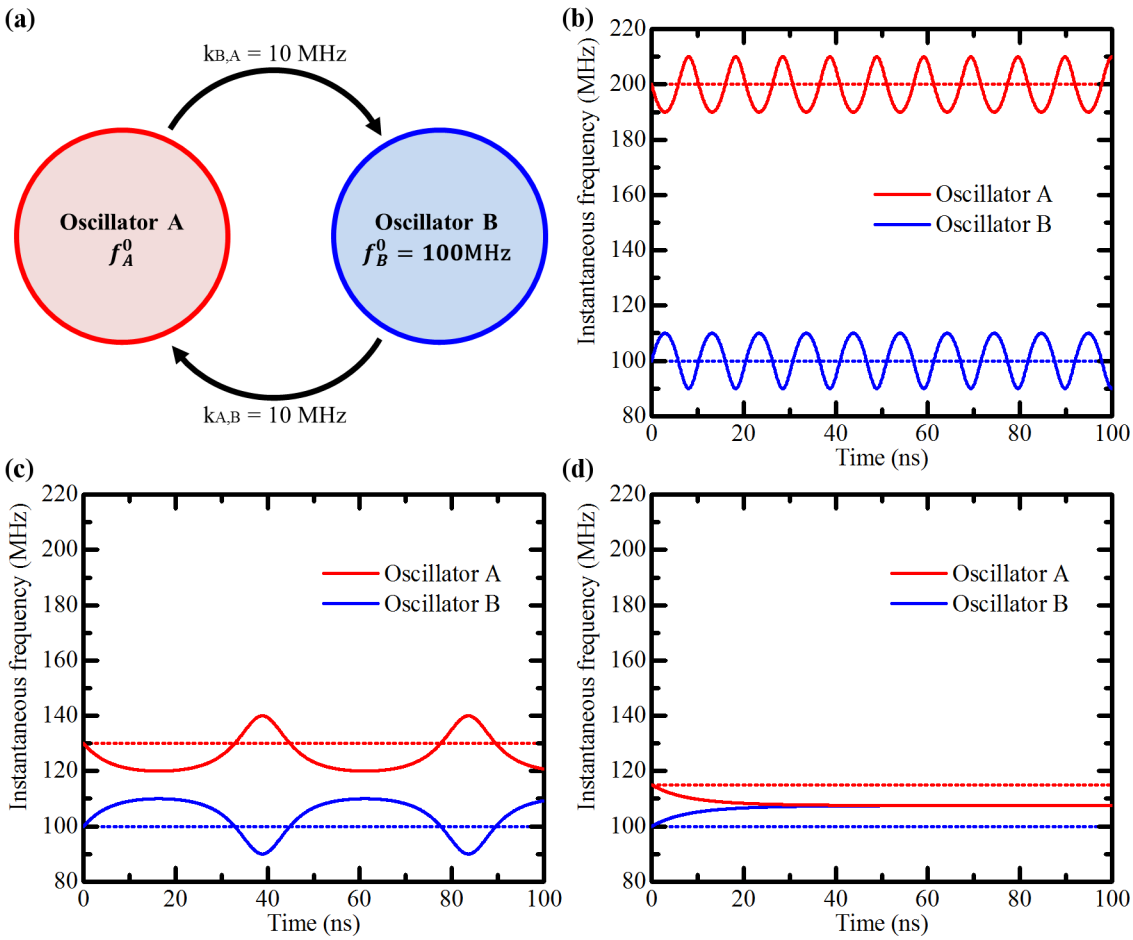


Figure 1.8: Two coupled kuramoto oscillators at different natural frequencies. **(a)** Schematic diagram. **(b)** Instantaneous frequencies of the oscillators (full line) and natural frequencies (dotted lines) $f_A^0 = 200$MHz and $f_B^0 = 100$MHz. **(c)** Another simulation for $f_A^0 = 130$MHz and $f_B^0 = 100$MHz **(d)** Simulation for $f_A^0 = 115$MHz and $f_B^0 = 100$MHz.

spend more time close to each other. However, phase jumps still occur periodically. This frequency pulling phenomenon is called **entrainment**.

Figure 1.8(d) shows a case with an even lower natural frequency mismatch $f_A^0 - f_B^0 = 15$MHz. After a transient period (about 50ns), the frequencies of the two oscillators lock to a common frequency (107.5MHz). This is a typical example of **synchronization** with frequency locking.

This simple example illustrates that the behavior of the instantaneous frequencies of two coupled oscillators depends on their natural frequency mismatch: the closer the oscillators are to synchronization, the more their frequencies are attracted to each other on average. To properly visualize and quantify this phenomenon, the mean frequencies of the oscillators can be evaluated as functions of their natural frequency mismatch.

For this purpose, I have simulated the same system by varying the natural frequency of oscillator A while keeping $f_B^0 = 100$MHz constant, and evaluated the mean instantaneous frequencies of the two oscillators. To achieve high precision, the simulation was run for $T = 10$s

and the frequencies were averaged over the last 5 seconds to reject any transient period. Figure 1.9 shows the mean frequencies of the two oscillators in full lines and their natural frequencies as dotted lines, as functions of f_A^0.

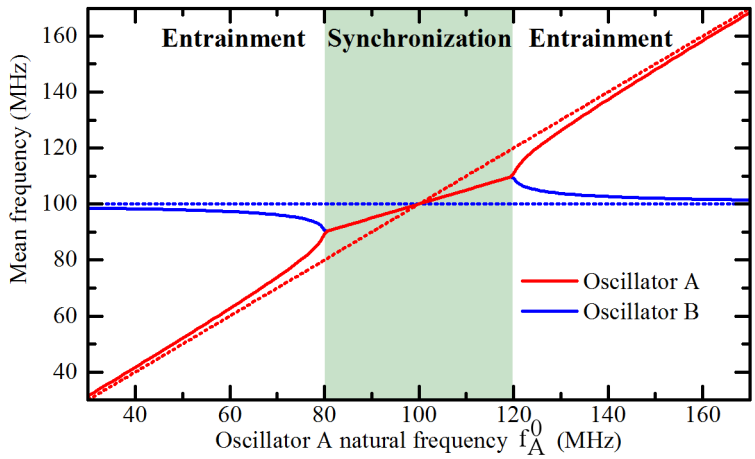


Figure 1.9: Mean frequencies (plain lines) and natural frequencies (dotted lines) of two coupled Kuramoto oscillators A and B as functions of the natural frequency of oscillator A.

The simulation shows that when the natural frequency mismatch is high (for example $f_A^0 <$ 50MHz), the mean frequencies of the oscillators stay almost equal to their natural frequencies. This shows that even though the instantaneous frequencies fluctuate in time due to the coupling, the average influence of an oscillator on the frequency of the other is low when the mismatch is high.

As f_A^0 gets closer to f_B^0, the mean frequencies get closer to each other (increasing entrainment) until $f_A^0 = 80$MHz where the two oscillators synchronize. The synchronization region is centered around f_B^0 and spans 20MHz. This can be understood through equations 1.2 where the frequency of oscillator A can be pulled down to $f_A^0 - k_{A,B} = 190$MHz at most and the frequency of oscillator B can go up to $f_B^0 + {}_{B,A} = 110$MHz, which can bring their frequencies closer to each other by $k_{A,B} + k_{B,A} = 20$MHz. This corresponds to the maximum natural frequency mismatch that allows them to reach a common frequency by interacting with each other and to synchronize.

These observations therefore provide two important intuitions on the dynamics of Kuramoto auto-oscillators that can be leveraged for computational purposes:

- the entrainment between two oscillators increases with their coupling strength but decreases with their natural frequency mismatch.
- two oscillators can synchronize when their total coupling strength is higher than their natural frequency mismatch.

Analytical predictions in Kuramoto networks

Since its first formulation [52], the Kuramoto model has been an important research subject in mathematics. The theory of Kuramoto networks is now very elaborate and can provide important predictive tools to build powerful oscillator systems, but key aspects of even simple networks remain open questions.

An important observation is that the Kuramoto equations are fully scale invariant [8] and are only sensitive to frequency or phase differences relatively to coupling strengths. This means that shifting all natural frequencies or all phases in a Kuramoto system by the same amount does not affect the dynamics. Moreover, multiplying both the natural frequencies and the couplings by the same factor also leads to a fully equivalent behavior. This scale and shift invariances therefore allow the Kuramoto model to be used consistently to describe different types of oscillators running at different speeds.

Networks of Kuramoto oscillators have also been shown to exhibit high sensitivity to initial conditions in certain cases, and even chaos [8, 16]. Theoretical approaches provide ways to predict complex behaviors in networks of coupled oscillators, but those predictive capabilities heavily depend on the structure and size of the network.

On one side, the two-oscillator problem presented above is fully solved analytically: closed form expressions of the phases, average frequencies and synchronization ranges are available given the natural frequencies and coupling values [8, 62]. Analytical predictions of the equilibrium points of the 3 and 4 oscillator Kuramoto models have also been published recently [63].

On the other side, the case of an infinite number of oscillators $N \to \infty$ was also successfully studied. Kuramoto himself [52] provided analytical expressions allowing him to fully solve the problem of a continuum of uniformly coupled oscillators with suitable distributions of natural frequencies. The critical coupling value that induces a phase transition into a fully synchronized state of such a network is also known but the stability of steady state solutions is still a subject of research [11, 53, 56].

An important parameter that describes the synchronization state of a network with an infinite number of oscillators $N \to \infty$ is the order parameter r defined by:

$$re^{i\psi} = \frac{1}{N}\sum_j e^{i\theta_j}, \tag{1.4}$$

where ψ is the mean phase of the oscillators. The term r corresponds to the absolute value of the mean of the complex phases of the oscillators and quantifies the phase-coherence of the system. It converges to a stable value for an infinite number of oscillators and reaches its maximum $r = 1$ when the phases of all oscillators add up constructively, or its minimum $r = 0$ when their phases take balanced values in $[0; 2\pi)$.

Despite these advances, the case of a finite $4 < N < \infty$ number of oscillators, which is the situation relevant for this thesis, remains elusive. As Strogatz notes, "As of March 2000, there are no rigorous convergence results about the finite-N behavior of the Kuramoto model" [53]. This statement remains true today. Approximations have been provided to predict the oscillator phases in special cases such as full synchronization of all the oscillators in a network [64]. These approximations remain however impractical for applications.

One of the reasons of this lack of solutions is that the order parameter r behaves in a more complex way in the finite-N case. Strogratz [53] writes: “Whereas Kuramoto’s approach had relied on the assumption that r was strictly constant, [...] nothing like that could be strictly true for any finite N.” This suggests that more complex, and fluctuating synchronization states are present in the finite-N case, making it the regime with the richest dynamics.

It is also important to note that, as the order parameter arises from the mean activity (mean field) of a set of oscillators, it may be linked to neural oscillations in the brain, and more specifically to EEG/MEG signals. Cabral *et al.* [61] note that in their model of neural oscillations “the synchrony degree of the system (estimated using the Kuramoto order parameter) is strongly correlated with the envelopes of 10.5-21.5Hz oscillations” corresponding to MEG signals.

As the theory of finite networks of Kuramoto oscillators does not provide enough predictive power or is impractical for most applications, numerical simulations have proven to be an important approach for the study of complex oscillator systems.

Simulating Kuramoto networks

A Kuramoto network is described by a set of nonlinear differential equations that can be readily integrated using standard numerical iterative methods.

Through their inherent flexibility and ease of use, numerical integration approaches have provided important insight into the rich behavior of arbitrary Kuramoto networks by revealing both expected and unforeseen complex phenomena. For example, simulation was used to analyze complex partially synchronized states in Kuramoto networks [53]. It has also been used to study different topologies of oscillator network connectivity and to successfully reproduce complex behaviors observed in the brain [60] or in networks of nano-oscillators [17].

Figure 1.10 shows the phases of a two-dimensional network of locally coupled Kuramoto oscillators simulated by Flovik *et al.* [17], providing a visual example of its complex dynamics including the presence of vortices (Figure 1.10(c)). Figure 1.10(a) shows how the order parameter r increases from 0 without couplings ($k = 0$) to almost 1 for strong couplings ($k \geq 35$MHz). Snapshots taken at increasing coupling strengths are presented in Figure 1.10(b), and show phase transitions towards more coherent states as the couplings increase.

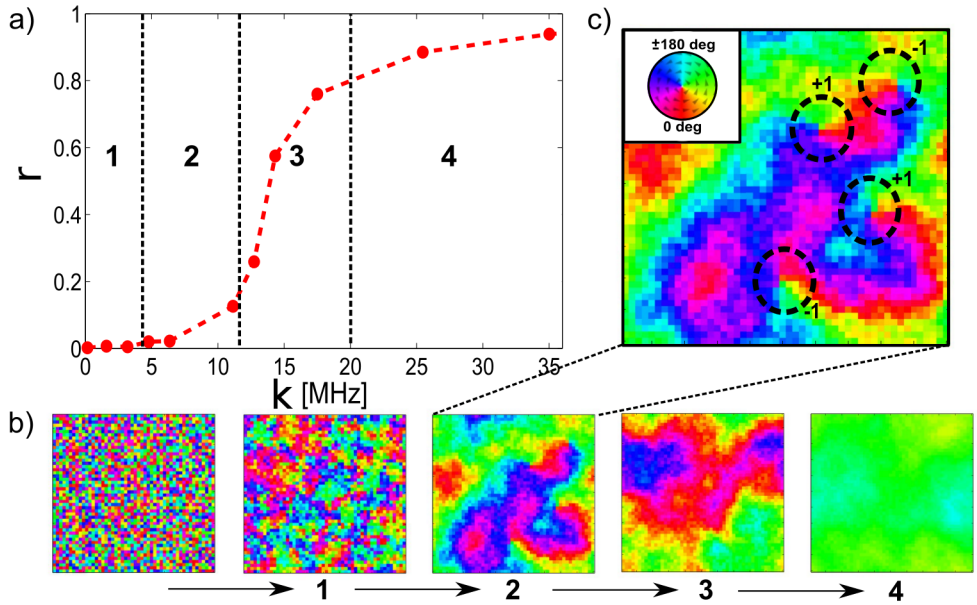


Figure 1.10: Simulation of a two-dimensional lattice of Kuramoto oscillators showing **(a)** the evolution of the order parameter r as the coupling strength k increases, **(b)** maps of the relative phases of the oscillators with increasing coupling strengths (1 to 4), and **(c)** the relative phases of the oscillators in one of the maps, highlighting the presence of vortices (dashed circles) with their polarities. Figure adapted from [17].

Integration method

In this thesis, simulations are carried out using the popular fourth order Runge Kutta method as it provides high accuracy at reasonable computing times. The choice of the integration time step dt has important implications: too high values degrade the accuracy of the results while too low values considerably slow down the simulation without providing notable accuracy gains. In my simulations, I set the value of dt by reducing it until the simulation results stopped changing, then further divided its value by 10.

In summary, numerical results have shown that the behavior of Kuramoto oscillators is the most complex, rich and computationally promising in the case of finite-N weakly coupled oscillator networks allowing multiple subsets of synchronized and free-running oscillators. Such networks exhibit complex behaviors and elude analytical description despite the simplicity of the underlying model. Furthermore, the Kuramoto model can also be extended to capture a larger part of the dynamics present in physical oscillators such as noise and extra non-linearities.

Noise and stochasticity

Noise is inherent to all physical systems above absolute zero temperatures, and can have important effects on the dynamics of oscillators by randomly perturbing their phases. It must therefore be taken into account for the design of oscillator-based systems.

Experimentally, the noise present in an isolated auto-oscillator is typically quantified by the Full Width at Half Maximum (FWHM, also called linewidth) of its power spectrum as measured by a spectrum analyzer [65, 66]. Theoretically, noise in a Kuramoto oscillator is usually introduced as a Brownian motion (random walk) of its phase resulting from white noise, and can be set up to match experimental FWHM values for realism.

Noise has also been shown to reduce the stability of synchronization between oscillators. High levels of noise can even prevent it altogether. This is mainly due to the fact that Gaussian noise can agitate the phase of an oscillator to a point where it has to re-synchronize by skipping one period. These noise-induced "phase slips" occur more frequently for higher noise levels [8].

SDE integration

When noise is present, the Kuramoto model becomes a system of stochastic differential equations (SDE). In that case, I used the Euler-Maruyama SDE integration method instead of fourth order Runge Kutta. The iterative update equations of the phases in the noisy Kuramoto model are then:

$$\theta_i \leftarrow \theta_i + \left[2\pi \times f_i^0 + \sum_j 2\pi \times k_{i,j} \sin(\theta_j - \theta_i) + \sqrt{\frac{2\pi \times \mathrm{FWHM}}{dt}} \times \mathcal{N}(0,1) \right] \times dt,$$

where dt is the timestep and $\mathcal{N}(0,1)$ a random number generator following a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$.

Also note that the noisy two-oscillator Kuramoto model can be fully solved analytically and the expression of the mean frequencies of the two oscillators as continued fractions can be found in [8] p.242. I used this expression to verify the stochastic simulations.

Despite its locally negative effect on synchronization, noise can also be an agent of self-organization. It has been shown that moderate noise helps stabilize global synchronization in certain Kuramoto network topologies showing multiple stationary states [67], a phenomenon reminiscent of noise-induced synchronization of neural oscillations (Section 1.1.2.2).

When the noise amplitude is too high, oscillators behave in a purely stochastic way, which is out of the scope of the Kuramoto model. **However, stochastic oscillators show promising computational capabilities and are the focus of Chapter 5.**

Non-linear oscillators: the Landau-Stuart model

The Kuramoto oscillator is qualified as isochronous or **linear** as the state of the isolated oscillator is fully described by its phase θ and its frequency does not depend on the amplitude of the oscillations. The state of a Kuramoto oscillator can be seen as evolving along a circle (Figure 1.11(a)), the radius of which is arbitrary and constant as it does not affect the dynamics. This system owes its linearity to a proportional response to any applied perturbation.

On the contrary, the frequency of **non-linear** (or non-isochronous) oscillators depends on their amplitude. For example, the frequency of the heartbeat in mammals changes with heartbeat amplitude [68]. Neurons in animal brains, the locomotion of fish and reptiles [69], and certain nano-oscillators [70] are also well described as non-linear oscillators.

The state of a non-linear oscillator typically converges to a stable oscillation orbit called the limit cycle along which it oscillates at the natural frequency f^0, when isolated from any source of perturbations. The stable amplitude ρ^0 of the oscillations corresponds to the radius of this cycle. As shown in Figure 1.11(b), the non-linear oscillator responds to an amplitude perturbation by relaxing back to its limit cycle, altering the phase in the process.

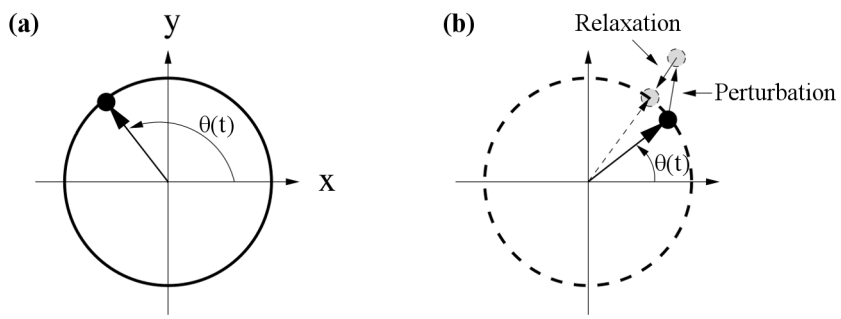


Figure 1.11: **(a)** Linear oscillator state evolving along a circle. **(b)** Non-linear oscillator relaxing back to its limit cycle (dashed circle) after a perturbation. Figure adapted from [8].

The equation that "describes the simplest nonlinear oscillator" as stated in [52] is the Landau-Stuart equation [8] which assumes a quadratic influence of the amplitude on the frequency. Without coupling terms, the amplitude ρ and the phase θ of a Landau-Stuart oscillator follow:

$$\begin{cases} \dot{\rho} &= \gamma({\rho^0}^2 - \rho^2)\rho \\ \dot{\theta} &= 2\pi f^0 + \nu\gamma(\rho^2 - {\rho^0}^2) \end{cases}, \tag{1.5}$$

where γ is the damping coefficient for radius deviation that quantifies how fast the amplitude ρ is pulled back towards its stable value ρ^0 after a perturbation. ν is the dimensionless nonlinear frequency shift that defines the non-linearity: it quantifies how much the frequency shifts from the natural frequency f^0 due to a radius deviation. At $\nu = 0$, the amplitude has no effect on the frequency and the oscillator behaves like a linear Kuramoto oscillator.

The Kuramoto model and its non-linear generalization, the Landau-Stuart model, are fun-

damental for the work in this thesis as they allow evaluating the behavior of arbitrary systems composed of linear or non-linear oscillators. In order to design a neuro-inspired oscillator-based computing architecture, the oscillator model needs to be tailored to the chosen physical oscillator technology. The diverse zoology of potentially suitable physical oscillators is presented in the next section.

Available nano-oscillators

In this section, I review and summarize the main types of nanoscale auto-oscillators currently available and potentially usable in oscillator-based computing architectures.

Relaxation nano-oscillators based on physical or chemical state switching

This particular type of nano-oscillator typically consists of a pair of electrodes enclosing an oxide film, amorphous semiconductor or Mott insulator with an electrical resistance that depends on its physical or chemical state. This state is in turn affected by the voltage applied at the device terminals. Voltages above a high threshold V_{th} trigger an abrupt transition to a high conduction state, resulting in a sudden drop of the device resistance. This behavior is called "threshold switching". This transition is reversible and the device resets to its higher resistance state after a short time when the applied voltage goes below a low threshold.

Such a threshold switching device can be connected to a capacitor in order to produce a relaxation auto-oscillator. To illustrate this approach, Figure 1.12 shows the results of Sharma *et al.* [71] on tantalum oxide nano-oscillators: Figure 1.12(a) presents a circuit including a threshold switching device in parallel with a capacitor that behaves as an auto-oscillator, and Figure 1.12(b) shows the time trace of the output voltage V_{out} and current I_{out} of the circuit. As the parasitic capacitor $C_{\text{paras.}}$ in parallel with the device is connected to the voltage source V_S through a resistor R_S, the voltage at the device terminals progressively increases. When the voltage reaches $V_{\text{th}} \approx 6\,\text{V}$, the device suddenly becomes conductive ($\approx 300\,\Omega$), generating a current spike through R_{ch2} and a quick discharge of $C_{\text{paras.}}$. The device stays conductive until the voltage is sufficiently low ($\approx 2\,\text{V}$) to allow it to reset to a high resistance state ($\approx 100\,\text{k}\Omega$). The cycle then repeats, producing a self-sustained periodic train of voltage spikes at the terminals of R_{ch2}.

This circuit uses $700 \times 700\,\text{nm}^2$ wide CMOS-compatible nano-devices and behaves as an auto-oscillator with a natural frequency tunable from 30 kHz to 300 MHz by adjusting the supply current using a MOSFET transistor. Sharma *et al.* [71] have also experimentally demonstrated electrical coupling between two of these nano-oscillators both using direct electrical connections and capacitive coupling, successfully achieving synchronization. The authors argue such devices could be scaled down to 3 nm and reach gigahertz frequencies.

Unfortunately, data on the total power consumption of such oscillators is scarce (Pick-

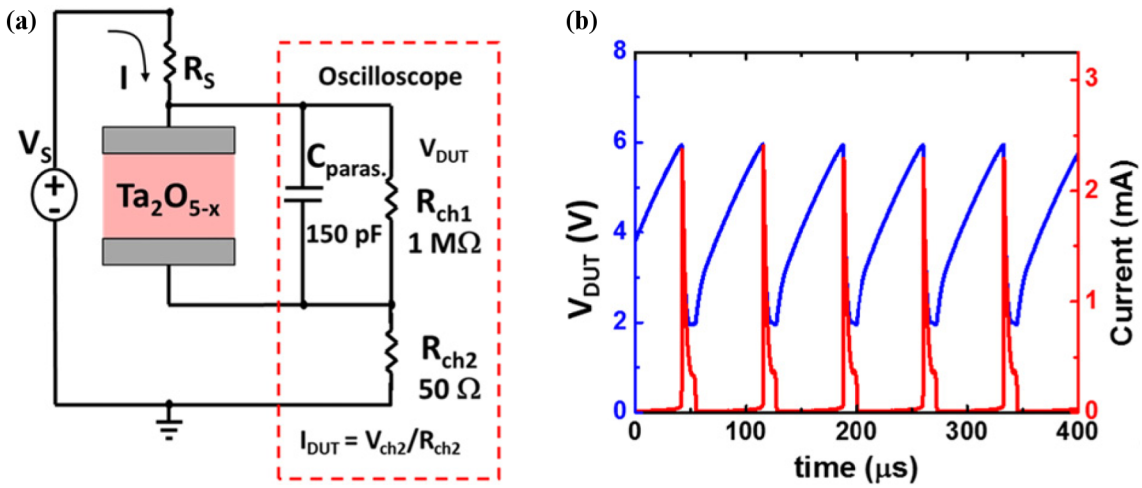


Figure 1.12: Figure adapted from [71] showing a tantalum oxide relaxation auto-oscillator. **(a)** Auto-oscillator circuit. **(b)** Current and voltage as functions of time.

ett *et al.* [72] show single switching with less than 100 fJ). Efficient coupling circuits involving more than two oscillators have yet to be designed, and due to their operation based on physical transitions, limited device lifetimes are expected.

Similarly, architectures implementing spiking units inspired by biological neurons were built using $100 \times 100\,\text{nm}^2$ Nb_2O_5 devices [73] or by lacunar spinel compounds [74].

These relaxation oscillators were modeled using custom compact models depending on the component but are also compatible with the the more general Van der Pol relaxation oscillator model [8, 75]. The synchronization dynamics of large populations of such oscillators can also be studied with the Kuramoto model [56].

With experimentally demonstrated wide frequency tuning, two-device coupling and synchronization capabilities, relaxation nano-oscillators based on intrinsic threshold switching are promising devices for the implementation of oscillator-based computing architectures [76], but suffer from lack of efficient multi-device coupling methods and shorter lifetimes.

CMOS ring oscillators

CMOS ring oscillators leverage the finite propagation time of a signal through logic gates made with transistors in order to implement auto-oscillators. Because they are built with transistors only, they are a practical source of oscillating signals inside integrated circuits, and are heavily used for this purpose. CMOS ring oscillators are typically made of an odd number of NOT logic gates arranged in a ring as shown in Figure 1.13. The odd number of not gates, together with propagation delays, prevents a stable state from establishing and induces permanent self-sustained voltage oscillations in the circuit.

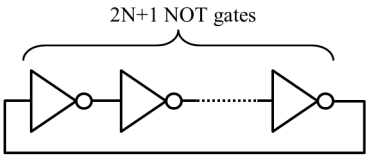


Figure 1.13: Simplified schematic of a typical CMOS ring oscillator.

Using additional control transistors, it is possible to tune the natural frequency of a CMOS ring oscillator to build a voltage-controlled oscillator that can reach multi-gigahertz frequencies. Natural frequencies are tunable across wide ranges, for example from 13 Hz to 407 MHz [77] or from 100 MHz to 3.5 GHz [78]. The size of a single oscillator heavily depends on the number of gates, the control circuitry and the choice of CMOS technology. An example implementation in 0.18 µm CMOS requires $49 \times 28\,\mu m^2$ [79]. Power requirements also vary heavily depending to the circuit and technology. However, low energy designs are available, and consume for example 6.75 µW at 124 MHz [80]. Ring oscillators can be modeled as non-linear oscillators [81], and multiple ring oscillators can be coupled to achieve synchronization [82].

All these features suggest that ring oscillators could be used for all-CMOS oscillator-based computing architectures, despite relatively high circuit area requirements.

Mechanical nano-oscillators

Nano-electromechanical systems (NEMS) are nano-scale systems containing moving (vibrating) mechanical parts. NEMS technology can be used to build nano-scale bridges, cantilevers or more complex structures that can mechanically oscillate in one or more dimensions. Their position, angle or deformation can both be measured and actuated through various techniques, the most common ones being electrostatic interaction and piezoelectric effects. NEMS can be co-integrated with CMOS and are already present in a wide range of commercial devices including inertial measurement units (accelerometers, gyroscopes), micro-mirror arrays for DLP video projectors or ink jet printer heads [83].

At first order, a typical NEMS cantilever behaves like a spring, and therefore a resonator. In order to turn it into an auto-oscillator, it is necessary to use support circuitry that performs a readout of the current position of the cantilever, and actuates the motion of the cantilever depending on the readout in a feedback loop, as introduced in Section 1.1.1.1. To illustrate this concept, Figure 1.14 presents the experimental work of Philippe *et al.* [84]: a NEMS cantilever with co-integrated CMOS feedback circuitry, effectively implementing an auto-oscillator.

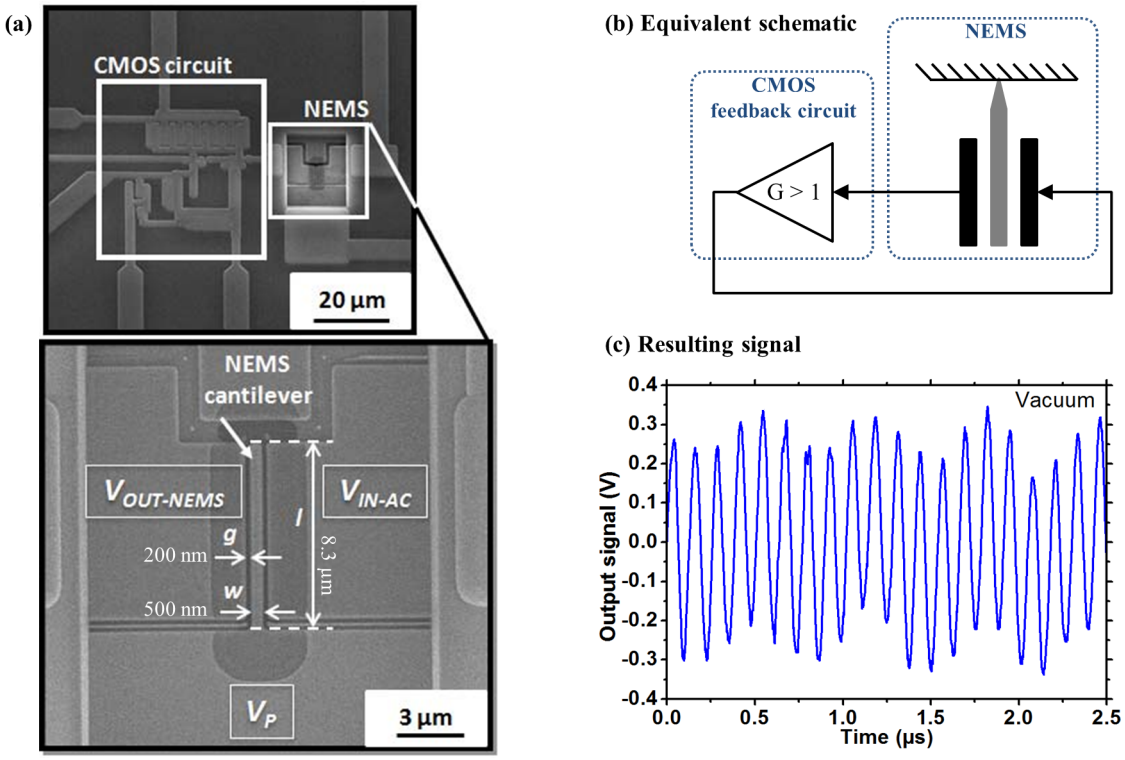


Figure 1.14: Figure adapted from [84]. **(a)** SEM micrograph of a NEMS-CMOS auto-oscillator. **(b)** Simplified schematic of the oscillator. **(c)** Output voltage as a function of time.

Figure 1.14(a) shows a micrograph of the system with a zoom on the NEMS cantilever, and Figure 1.14(b) shows an equivalent schematic of the system. The position of the cantilever is detected using the changes in capacity between the cantilever and the left electrode (left black rectangle on the schematic). This signal is amplified by a minimal CMOS amplifier that then actuates the cantilever in return using electrostatic interaction through the right electrode (right black rectangle on the schematic). Figure 1.14(c) presents the resulting time trace of the voltage signal generated by this auto-oscillator and shows sustained oscillations at about 8 MHz. The area occupied by the whole circuit on chip using 0.35 µm technology is $50 \times 70\,\mu m^2$, which is still relatively large.

NEMS auto-oscillator frequencies can reach hundreds of MHz [85] and natural frequency tuning was demonstrated using multiple approaches including dynamically adjusting the feedback circuit [86] or the stiffness of the moving part using piezoelectric materials [87]. These oscillators also show very low phase noise levels. For example, Feng*et al.* [85] measured FWHM = 9 kHz at $f^0 = 428$ MHz.

Moreover, coupling and synchronization of multiple NEMS oscillators has been demonstrated experimentally. The most straightforward coupling approach is mechanical coupling [88] where oscillators interact with each other through vibrations. As the magnitude of the vibrations decreases with distance, this is mostly a local coupling scheme that involves neighboring

oscillators. Nearest-neighbor coupling can also be achieved through optical coupling [89]. Finally, electrical coupling can achieve local but also global coupling by actuating the oscillators using the sum of the signals measured on individual oscillators [90]. NEMS oscillators have been successfully modeled using the Kuramoto model [91, 92] but also more complex models describing their dynamics more precisely [90].

With their industry standard fabrication techniques, CMOS integration, low noise and demonstrated natural frequency control, coupling and synchronization, NEMS oscillators are promising devices for oscillator-based computing architectures, but tend to require high circuit areas.

Josephson junction nano-oscillators

A Josephson junction consists of two superconductors separated by a thin insulating layer. When subject to a constant voltage, the current through a Josephson junction oscillates at a well-defined fundamental frequency, $\nu = V/\Phi_0$, where $\Phi_0 \approx 2.07$mV/THz and V is the time-averaged voltage across the junction [93]. Under certain voltage and current conditions, the output signal of the oscillator is nearly sinusoidal. The natural frequencies of such oscillators are also voltage-tunable from hundreds of kHz to more than 1 THz [94], raising considerable interest for terahertz applications [95, 96].

Note that a single Josephson junction has a broad power spectrum, but multiple junctions can be coupled together to synchronize and behave as one spectrally pure oscillator. Benz *et al.* [97] show that junctions with FWHM $\approx$ 160 MHz can be coupled in a 10 grid to produce a 88.8 GHz signal with FWHM = 13 kHz. However, this requires more circuit area per oscillator. Nevertheless, this limitation is compensated by the extreme scalability of Josephson junctions, and experimental results presented in [98] suggest that Josephson junctions could be scaled down to the nanometer.

Electrical coupling and synchronization of multiple Josephson junctions has also been demonstrated [93, 99–101] and coupled Josephson junction oscillators have been successfully modeled using the Kuramoto model [102].

An important drawback of Josephson junctions is that they can not operate at room temperature but require an important cooling effort. High critical temperature superconductors allow operation at temperatures up to 90 K [103], but still require liquid nitrogen cooling.

As a result, Josephson junctions are a promising device for implementing oscillator-based computation architectures but come with a high cooling overhead. These devices are therefore not adapted to low-area/low-energy applications, but show interesting potential for high supercomputers.

Magnetic nano-oscillators

Magnetic nano-oscillators leverage the magnetization dynamics of materials and the spin of electrons in order to achieve auto-oscillation at the nano scale. Most magnetic nano-oscillators rely on the CMOS-compatible magnetic tunnel junction (MTJ) technology at the heart of novel magnetic random access memories [104]. In their most basic form illustrated in Figure 1.15, MTJs consist of a nano-pillar (≈ 100 nm in diameter) containing a thin insulating barrier (called the tunnel barrier) sandwiched between a pinned ferromagnet with constant magnetization, and a "free" ferromagnet with a magnetization capable of oscillating.

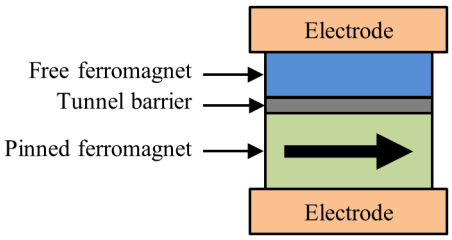


Figure 1.15: Simplified schematic of a magnetic tunnel junction.

Electrons carry a small quantity of angular momentum called the spin. In a typical electronic current, half of the electrons have a spin "up", and half have a spin "down". When electrons flow through the junction, they get spin-polarized by the pinned ferromagnet, causing an excess of electrons with either spin to reach the free ferromagnet. Such a spin-polarized current applies in turn a magnetic torque [105] on the free ferromagnet. This phenomenon is called the spin-transfer torque effect.

Current-induced spin transfer torque can cause oscillations of the magnetization in the free ferromagnet. Oscillators using this approach are called spin transfer nano-oscillators (STNOs). Different magnetization distributions and oscillation regimes exist. **The particular case of a vortex magnetization core precessing along an orbit is studied in Chapter 3. Stochastic oscillations can also be spontaneously induced by noise in unstable magnetic tunnel junctions, which are studied in Chapter 5.**

As the tunnel barrier is thin enough for electrons to cross it by quantum tunneling, MTJs behave as electrical resistors. Moreover, the tunnel magneto-resistance effect [106] (TMR) causes the resistance of magnetic tunnel junctions to depend on the the free layer magnetization.

As illustrated in Figure 1.16, the presence of a net magnetization is linked to an asymmetry in the densities of spin up ($n^{\uparrow}$) and spin down ($n^{\downarrow}$) electronic states. The electronic currents for the two spin orientations can be seen as two parallel currents (red arrows in the figure). When the two sides of the tunnel junction are magnetized in the same direction (top half of the figure), the excess of electrons in the favored spin direction on one side of the tunnel barrier matches the excess of available states for the same spin direction on the other side of the barrier, allowing a strong spin-polarized electronic current to flow. On the contrary, when the

magnetizations of the two ferromagnets are in opposite directions (bottom half of the figure), an excess of states for a given spin on one side of the barrier is matched with a shortage of states on the other side. This results in low currents for both spins. The resistance of of the device can therefore be modeled as a pair of resistors in parallel (right side of the figure), each corresponding to one electronic spin. When the ferromagnet magnetizations are aligned, the device is equivalent to a low-value resistor and a high-value resistor in parallel. When the magnetizations are in opposite directions, the device is equivalent to two high-value resistors in parallel.

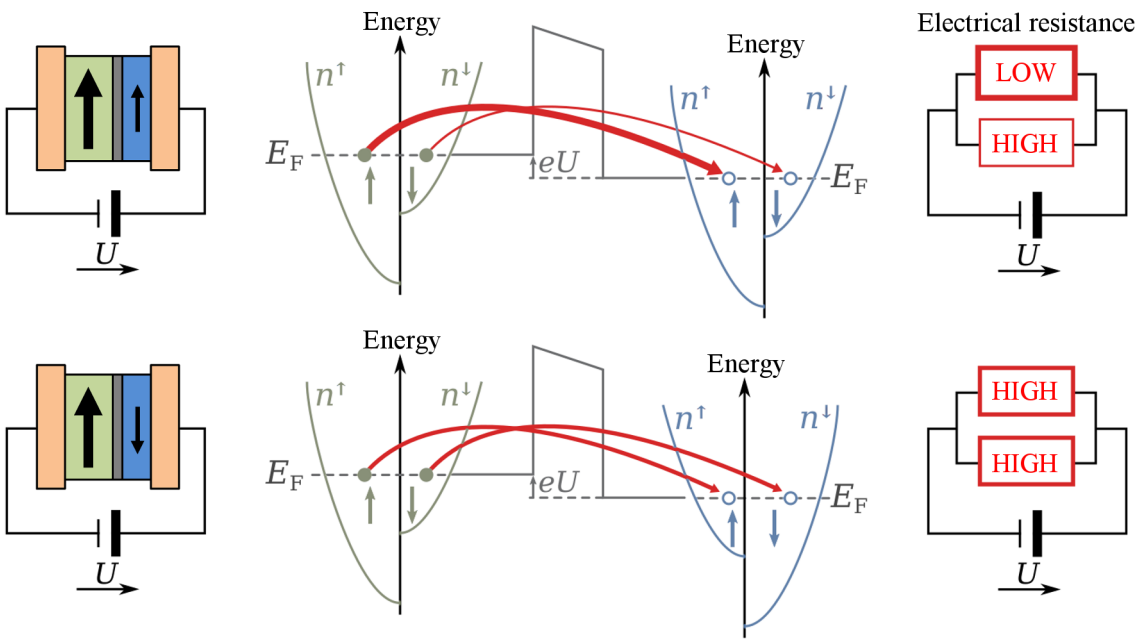


Figure 1.16: Schematic description of the Tunnel Magneto-Resistance effect (TMR), adapted from [107]. MTJs under a constant voltage bias U are shown on the left in the cases where the magnetization of the free ferromagnet is aligned with the pinned ferromagnet (top), and in the case where they are in opposite directions (bottom). The central part of the figure shows the corresponding densities of electronic states ($n^\uparrow$ for spin up states, and $n^\downarrow$ for spin down) in the pinned (left) and free ferromagnets (right). The red arrows are electron tunneling currents and E_F are the Fermi levels. The right part of the figure shows the equivalent resistance model.

As a result, resistance is minimal when the free layer magnetization is aligned with the pinned layer magnetization, and maximal when they are in opposite directions, with intermediate cases resulting in intermediate resistance values. The resistance of the MTJ therefore oscillates together with the magnetization of the free layer, allowing signal readout.

Natural frequency control through voltage or current biases allows wide tuning ranges, typically from 1 to 5 GHz [108] or for 100s of MHz in the sub-gigahertz range for vortex oscillators (see Chapter 3). Moreover, Bonetti *et al.* [109] experimentally observed 46 GHz oscillations in 100 nm nano-contact STNOs under strong magnetic fields (1.4 T) at an angle of 30° with respect

to the device plane. The authors also theoretically predict that frequencies beyond 65 GHz could be reached using perfectly in-plane magnetic fields.

Magnetic oscillators can be coupled electrically [110] in arbitrary topologies, through dipolar interactions resulting in local couplings [111], or through exchange interaction and spin waves [112]. Synchronization of multiple magnetic nano-oscillators was also shown both theoretically and experimentally [112–116].

Depending on their structure, technology and operation regime, these nano-oscillators show different levels of phase noise. Experimentally observed values for vortex STNOs (see Chapter 3) are around FWHM $\approx$ 1 MHz at $f^0 \approx 500$ MHz [117]. Phase noise can be further pushed below 1 Hz at $f^0 = 7.344$ GHz for STNOs stabilized by external circuitry [118]. Certain types of magnetic oscillators have been modeled using specialized models [119] but also using the generic Kuramoto model [17].

All these properties make magnetic nano-oscillators attractive as microwave sources for telecommunications [108, 120, 121], but also promising candidates for oscillator-based computing architectures.

Other types of oscillators

Other types of small form factor oscillators exist, and could also present attractive features for oscillator-based computing.

Traditional capacitor-inductor resonators coupled with feedback circuitry are currently used as auto-oscillators for a wide range of applications, but have limited integration capabilities. Negative differential resistances (e.g. Gunn diode, Tunnel diode) coupled to resonant circuits also show promising results for auto-oscillator based Terahertz generation [122] and could be potential candidates for oscillator-based computing.

Optical oscillators using light-matter interactions show interesting potential for high speed computation, and have been shown to synchronize [123, 124]. Research on using these oscillators for computing is however still at a preliminary stage [125].

Chemical oscillators also present significant potential for computation, and populations of chemical oscillators show synchronization capabilities [126]. However, chemical oscillators still require research efforts on miniaturization and lifetime.

Biological auto-oscillators including certain types of neurons or groups of neurons such as the endogenous oscillators presented in Section 1.1.2.2 could be used for computation. Computation using living neuron cultures on silicon chips was achieved, for example DeMarse *et al.* presented a “brain in a dish” controlling a flight simulator [127]. However, such approaches require expensive life support systems, and the shortened lifetime of neurons in such conditions is a major limitation.

Alternatively, molecular motors such as the ones found in living cells could be used as auto-oscillators. Research on this approach is however at a very early stage and consists mainly of theoretical predictions [128].

Computing with oscillators

Several computing systems leveraging the dynamics of oscillators have been proposed and implemented. This section describes the main oscillator-based computing approaches.

Binary computing with oscillators

Oscillator-based binary computers, called parametrons, were among the first computers used in Japan in the late 1950s, before being superseded by faster and more scalable semiconductor-based computers in the 1960s. Parametrons perform classical binary operations using coupled oscillators based on macro-scale inductors all running at the same natural frequencies and subject to a global external oscillating signal at the double of this frequency. This approach exploits the capability of these oscillators to achieve two possible stable states of synchronization with the external signal. Due to higher order interactions, the oscillators spontaneously lock to the external signal with two possible relative phases (0 or π), allowing each oscillator to represent a bit 0 or 1. Different types of couplings between oscillators implement a complete set of logic functions that manipulate these bits, allowing full-featured computation [129, 130]. With CMOS reaching its limits, parametrons have recently seen a new surge of interest.

Quantum flux parametrons implemented with superconducting Josephson Junction oscillators (see Section 1.3.4) show no resistive losses and extremely energy-efficient, fast and thermodynamically reversible bit state switching. This physical reversibility, together with bijective logic gates that do not induce dissipation through loss of information, could lead to fully reversible super-computers with energy requirements below the Landauer limit, which is the lower bound on the energy consumption of dissipative computers. Fully reversible quantum flux parametron logic gates have been demonstrated experimentally [131], but full computing systems have not been demonstrated yet.

Parametrons can also be implemented with a wide range of room temperature auto-oscillator technologies [132]. For example, they have been built and characterized using oxide-based relaxation oscillators (introduced in Section 1.3.1) [71, 133] and nano-electro-mechanical oscillators (introduced in Section 1.3.3) [134].

Parametron computers built with nano-oscillators show promising capabilities for general purpose computing based on standard binary operations. However, while performing the same tasks, their current implementations are still slower and less energy-efficient than their CMOS counterparts.

Image segmentation

Low level image segmentation consists in isolating regions of a picture based on certain criteria such as brightness or contrast. It is one of first processing steps used by a high number of

image processing algorithms [135], allowing for example the detection of foreground objects in a picture.

Spatially local coupling of oscillators can be achieved through dipolar, mechanical, electrical or optical interactions between neighboring oscillators. Moreover, their synchronization also depends on the differences between their tunable natural frequencies. As a result, the synchronization dynamics of a two-dimensional grid of locally coupled oscillators naturally shows both space and value dependencies, which can be leveraged to implement brightness-based image segmentation. Figure 1.17(a) shows an example of oscillator network where the natural frequency of each oscillator is controlled by the corresponding pixel of a picture, and the oscillators are bidirectionally coupled to their close neighbors (we assume a uniform coupling strength k). Figure 1.17(b) shows a map of the average frequencies of the simulated Kuramoto oscillator network without coupling ($k = 0$). In that case, the average frequencies of the oscillators match their natural frequencies, and therefore the brightness of the original image pixels. When couplings are present ($k > 0$), neighboring oscillators with similar natural frequencies synchronize to a common average frequency. On the average frequency map Figure 1.17(c) for $k = 30$ MHz, contiguous pixel regions of similar brightness are mapped to a single common brightness value. This result corresponds to an image segmentation according to brightness and allows simplifying an image as a pre-processing step.

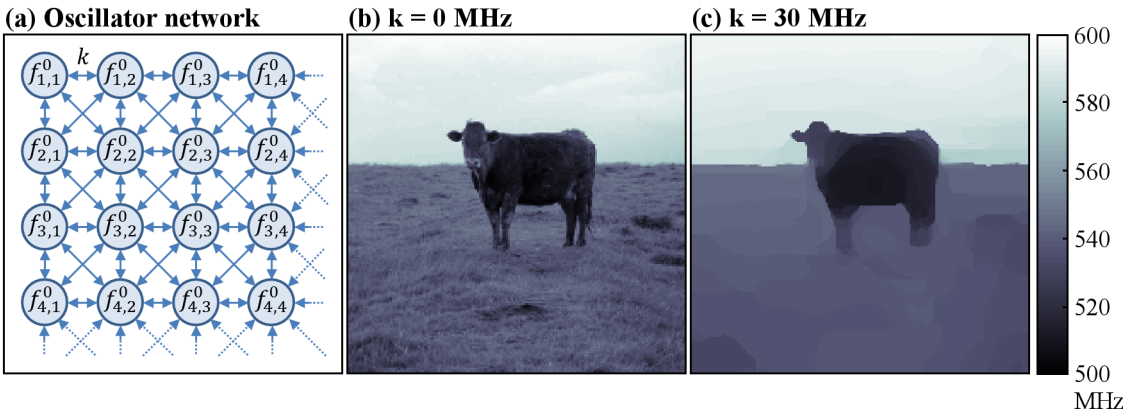


Figure 1.17: Segmentation with locally coupled oscillators. Each oscillator represents a pixel, its natural frequency is set according to the original image pixel brightness, and its resulting mean frequency is the segmentation brightness output. **(a)** Neighbor coupling scheme. **(b)** Clustered image output without coupling ($k = 0$MHz). **(c)** Clustered image output with $k = 30$MHz couplings.

I performed this simulation as a side-project during the 2016 Capo-Caccia neuromorphic engineering workshop, but this approach was already described in the litterature [136]. Similar and more advanced oscillator-based image segmentation and contrast detection approaches have been proposed using different oscillator technologies among which neural oscillators, chemical oscillators, micro-electro-mechanical oscillators [137] or spin transfer torque nano-oscillators [138].

Using the intrinsic dynamics of oscillators to perform image segmentation is therefore an attractive approach for efficient image pre-processing.

Vertex coloring

Vertex coloring, is a NP-hard problem that consists in attributing a color to each node of a graph so that two connected nodes do not get the same color, while minimizing the total number of different colors. Finding an approximate coloring solution plays an important role in resource allocation and scheduling, including program compilation or frequency attribution in telecommunication [139].

In 2017, Parihar *et al.* [140] proposed a reformulation of this problem that can be solved approximately by capacitively coupled oxide-based relaxation oscillators with individually tunable dense couplings, and presented a working experimental implementation for small graphs. With the chosen coupling scheme, two connected oscillators synchronize out of phase, i.e. their phases "repel" each other. When a full oscillator network, this "repelling effect" leads to a stable state of the phases from which an approximate solution can be easily extracted. Note that using oscillators for graph coloring was previously theorized in 1998 by Wu *et al.* [141] for 2 colors, and generalized to k colors in 2009 by Lee [142].

Vertex coloring is therefore an interesting example of computing with intrinsic nano-oscillator dynamics. However, the number of colors the architecture can represent is limited by the phase noise and variability of the oscillators. Moreover, the number of independently tunable connections between oscillators grows quadratically with the number of nodes in the graph, which represents a serious scalability limitation.

Degree of matching computation

Computing a distance measure between two vectors can be computationally heavy for traditional computers. For example, computing the $L_2^2 = ||\boldsymbol{A} - \boldsymbol{B}||_2^2 = \sum_i^N (a_i - b_i)^2$ distance between N-dimensional vectors $\boldsymbol{A} = \{a_i\}_{i=1..N}$ and $\boldsymbol{B} = \{b_i\}_{i=1..N}$ requires as many multiplications as there are dimensions, followed by an accumulation operation.

As explained in Section 1.2.2.3, the order parameter of a network of coupled oscillators naturally indicates the level of synchronization in the network, and can be used to obtain an approximate similarity measure called "degree of matching" (DOM) [143]. Among others, Yogendra *et al.* [138] proposed and simulated such an architecture with N coupled spin transfer torque nano-oscillators where the natural frequency of the i-th oscillator is shifted away from a common value proportionally to $a_i - b_i$. A simple circuit then extracts the degree of matching using the mean field signal of the oscillators. Figure 1.18(a) shows a simplified representation of this network in the $N = 4$-dimensional case.

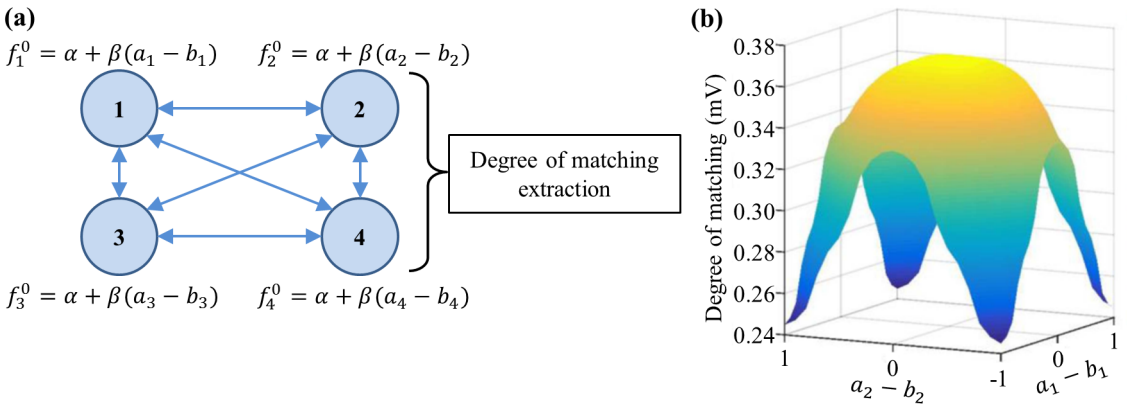


Figure 1.18: Degree of matching measure using 4 coupled oscillators. **(a)** Simplified view of the oscillator network. **(b)** Degree of matching as a function of $a_1 - b_1$ and $a_2 - b_2$ when $a_3 = b_3$ and $a_4 = b_4$. Adapted from [138].

Figure 1.18(b) shows the output degree of matching as a function of $a_1 - b_1$ and $a_2 - b_2$. The more vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ are similar ($a_i \approx b_i$), the closer the natural frequencies are to each other, inducing a higher level of synchronization in the network, and a higher degree of matching. The opposite of the parabola-shaped degree of matching, −DOM, can then be rescaled to resemble the $L_2^2(\boldsymbol{A}, \boldsymbol{B}) = \sum_i^N (a_i - b_i)^2$ function for small distances. As a result, the authors propose to use this approach to approximate the L_2^2 distance between two similar vectors by leveraging oscillator dynamics.

Note that they also propose the computation of the dot product $\boldsymbol{A} \cdot \boldsymbol{B}$ by using the expansion $\boldsymbol{A} \cdot \boldsymbol{B} = -\frac{1}{2}\left(L_2^2(\boldsymbol{A}, \boldsymbol{B}) - L_2^2(\boldsymbol{A}, \boldsymbol{0}) - L_2^2(\boldsymbol{0}, \boldsymbol{B})\right)$.

Similar schemes have been proposed in multiple other papers [136, 144, 145], and implementations with other kinds of oscillators including oxide-based relaxation oscillators [146] have been imagined. It is therefore possible to naturally implement approximate distance evaluation using oscillator dynamics without having to explicitly compute any sum or product.

Auto-associative memories and Hopfield networks

Auto-associative memory corresponds to the capability of fully retrieving a known pattern when given an incomplete, deformed or noisy version of this pattern. For example, when provided with the incomplete quote “to be, or not to –”, the brain spontaneously and effortlessly associates it to the corresponding known full pattern “to be, or not to be”.

Auto-associative memories are important for a wide range of applications including auto-completion, de-noising preprocessing or error correction [147]. However, traditional computers do not perform well on such tasks which typically require massively distributed computation. As a result, alternative auto-associative memory schemes leveraging device physics have been proposed. These approaches borrow the ideas of Hopfield networks, which consist in

tuning the interactions between a set of binary units so that their final states match known stored patterns, and apply them to dynamical systems by engineering the attractors of a physical system to match stored patterns. When the state of the system is set according to a noisy or incomplete input pattern, the dynamics converge to the closest attractor, naturally reconstructing the full pattern in the process [147].

Multiple oscillator-based Hopfield-like auto-associative memories have been proposed since the original theoretical formulation of the approach by Hoppensteadt *et al.* [148]. As an example, Maffezzoni *et al.* [149] present an auto-associative memory for boolean words (binary vectors $\boldsymbol{x} = \{\pm 1\}$) using coupled MEMS oscillators modeled using the Kuramoto model and inspired by a work from Hoppensteadt *et al.* [150]. First, the "correlation" matrix of each bit i with each other bit j is computed and averaged across all M known examples:

$$k_{i,j}^{\text{rec}} = \frac{1}{M} \sum_{m=1}^{M} x_i^{(m)} x_j^{(m)} . \tag{1.6}$$

In order to perform auto-association on an unknown binary vector x^{test}, its correlation matrix $k^{\text{test}} = x_i^{(\text{test})} x_j^{(\text{test})}$ is applied as the coupling matrix between the oscillators in the network. As a result of synchronization, the phase of each oscillator converges to $\pm\pi$ with respect to a reference oscillator, which closely matches the presented vector bits. After a stabilization period, the coupling matrix is set to k^{rec} and the dynamics converge towards the corresponding stored attractor. After stabilization, the relative phases of the oscillators encode the bits of the reconstructed vector. Figure 1.19 shows an example reconstruction result of an image containing a noisy version of one of the stored digits: "2".

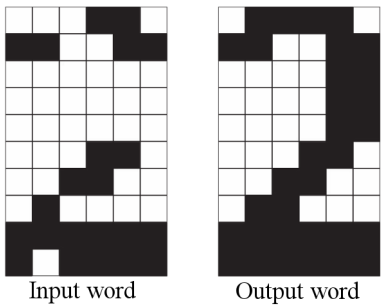


Figure 1.19: MEMS-based auto-associative memory reconstructing a picture of the digit "2". Adapted from [149].

This architecture leverages intrinsic oscillator dynamics to implement an auto-associative memory. However, it still requires a quadratically large number of tunable amplifier-based couplings, as well as the capability to read individual oscillator phases. The couplings need to be tuned dynamically during the computation, which could be done by applying arbitrary waveforms [151], but remains impractical at the nano-scale.

Similar architectures were also proposed for other oscillator technologies including spin transfer torque nano-oscillators [138], laser oscillators [125].

With the Hopfield approach, the maximal number of orthogonal patterns that can be stored in a system of N oscillators while still achieving recognition with low error rates is approximately $0.138N$ but is usually lower in practical implementations [152]. Oscillator-based auto-associative memories therefore offer promising capabilities for fast and efficient operation, but their scalability is limited.

Hetero-associative memories and pattern classification

Hetero-associative memories involve associating an incomplete or noisy pattern of one type to a corresponding pattern of another type. As an example, the human brain can associate a particular smell to visual memories from past experiences involving similar smells [153].

Moreover, this mechanism of the brain is thought to be correlated with certain types of neural oscillations [154], suggesting that oscillator-based approaches could be particularly well suited for hetero-associative memory implementation.

Pattern classification

In the particular case in which an input pattern is associated to a "label", for example when the picture of a face is associated to the name of the person, hetero-associative memory corresponds to pattern classification. Classification consists in automatically recognizing a presented pattern and tagging it with the right label. For example, it can consist in classifying presented pictures as "cat" or "dog" if they contain the corresponding animal, or classifying a tumor as "benign" or "malign" given some of its properties. Classification algorithms are ubiquitous in various contexts including object recognition for self-driving cars [155], biometric identification [156] or credit scoring [157].

Classification tasks typically require important computational resources, and leveraging the naturally complex dynamics of coupled nano-oscillators to perform classification efficiently is therefore highly attractive.

Oscillator-based classification

A straightforward way of performing classification is to use the degree of matching (Section 1.4.4) to compare an input pattern to stored examples of each class and to output the label of the best matching one. Evaluating the degree of matching against multiple stored examples can be done in a parallel way using multiple coupled oscillator circuits. Circuit complexity can be further reduced by using a hierarchical architecture, and achieves promising results on image recognition [158].

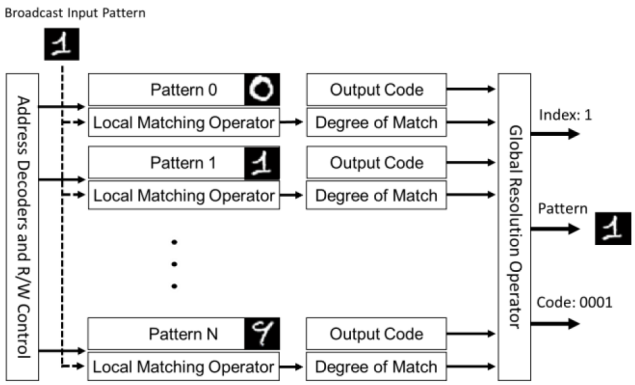


Figure 1.20: Classification architecture using the degree of matching to find the index of the stored pattern that matches the input pattern the best. Adapted from [158].

Figure 1.20 shows a classification architecture by Fang *et al.* [158] that uses CMOS ring oscillators. Degrees of matching are computed between the input pattern and each stored pattern, and a "Global Resolution Operator" resolves the label of the best matching stored pattern with high accuracy. Similar architectures have been proposed for other types of oscillators including spin transfer torque nano-oscillators [159] or MEMS [150].

This type of oscillator-based classifier only requires tuning natural frequencies and reading degrees of matching, which is convenient for implementations based on nano-devices. However, a set of training examples needs to be stored, and the number of coupling and readout circuits increases rapidly with the number of examples to compare with. This seriously limits the scalability of such an architecture.

An alternative classification architecture proposed in a theoretical context by Vassilieva *et al.* [160] solves this scalability problem by using synchronization readouts of a network of tuned oscillators as the ''label" associated to the presented pattern. It only requires tuning natural frequencies, detecting synchronizations, and the number of oscillators does not depend on the number of examples. **In Chapter 2, I detail this architecture, adapt it to the constraints of nano-oscillators and characterize its performance and robustness.**

Feed-forward artificial neural networks

Definition of an artificial neural network

Currently, the most popular and powerful machine learning models for cognitive tasks are artificial neural networks, which are loosely inspired from biological neural networks. Figure 1.21 shows a artificial neural network in its simplest feed-forward form, that classifies a set of features (height, weight, head size) into two classes (cat or dog). It consists of successive layers (2 in the figure) of non-linear transformations (neurons) on data to perform the task. The vector of outputs $\boldsymbol{h}$ of the neural network gives the degree of confidence that the presented set

of features corresponds to a given class. In Figure 1.21, the neurons performing non-linear transformations are represented as blue circles. The input of each layer is a linear combination of the outputs of the previous layer (or the input data for the first layer) and a constant bias. These linear combinations (red lines) are encoded as weight matrices, and the ensemble of all the elements of all those matrices constitutes the parameter vector $\mathbf{\Theta}$ of the model.

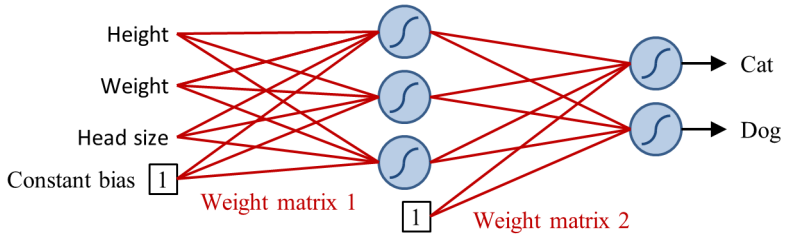


Figure 1.21: A simple two-layer neural network for classification. Blue circles are non-linear transfer functions (neurons), and red lines are tunable weight matrices.

The parameters of the model are then optimized using a learning algorithm so that the network provides the right answer on a set of known examples. After this learning step, the network is able to generalize and classify new, unknown inputs. **Chapter 4 focuses on this learning procedure.**

Artificial neural networks play an important role in natural data processing and cognitive tasks such as language translation, or image classification. However, these networks are currently run by traditional computers or graphic cards using an important number of floating point operations to simulate non-linear transformations.

Implementation with oscillators

As oscillators show intrinsically non-linear responses, they represent an attractive approach for implementing non-linear neurons in artificial neural networks. Approaches implementing artificial neural networks using oscillator-based neurons require the theoretical knowledge of the exact transfer function that the neuron implements. As explained in Section 1.2.2.3, usable analytical predictions are scarce in coupled oscillator models. As a consequence, oscillator-based neurons need to use extremely basic oscillator dynamics that are well-described by theory.

In the literature, I have found only two types of oscillator-based neuron designs. They both implement a simple threshold function that yields a high output if the input is above a given threshold value, and a low output otherwise.

In a work from 2013, Kaluza [161] proposes a simple threshold neuron made of a pair of synchronized oscillators described by a modified Kuramoto model with doubled frequencies. The input of the oscillator tunes the coupling strength of the connection between the two oscillators, and the output corresponds to the phase difference between them. Similarly to the parametrons introduces in Section 1.4.1, and due to the special oscillator model being used,

the oscillators synchronize with a relative phase of π if the coupling strength (and therefore the input) is beyond a threshold, which corresponds to output "1", or with a relative phase of 0 (output "0") otherwise.

The drawbacks of this scheme are that it uses an exotic oscillator model, and relies on tuning the coupling between the oscillators in real time.

In their 2015 paper, Yogendra *et al.* [162] propose a threshold neuron that uses a pair of bidirectionally coupled spin transfer torque nano-oscillators with similar baseline natural frequencies. The input detunes the natural frequency of one of the oscillators. The two oscillators stay synchronized for low input values, and unlock when the input crosses a threshold. The readout is opposite to the degree of matching (see Section 1.4.4). The output is 0 when the two oscillators are synchronized, and 1 when a sufficiently high input desynchronizes them.

Contrary to Kaluza's neuron, this approach does not rely on dynamically tuned couplings, and functions with realistic oscillator models. Note that the authors also introduce an alternative design that replaces the reference oscillator by a radio-frequency current source. This requires a function generator but a single signal can be shared by multiple neurons.

Artificial neurons implemented with oscillators are therefore a promising element for efficient feed-forward artificial neural networks. However, in order to achieve higher computational power, neurons with more advanced transfer functions than a simple threshold are required. **In Section 4.5.1, I define and characterize such a neuron.**

Reservoir computing

Definition of reservoir computing

Similarly to more conventional artificial neural networks, reservoir computing seeks to apply non-linear transformations to data in order to perform a cognitive task, but it offloads non-linearity computation to a "reservoir".

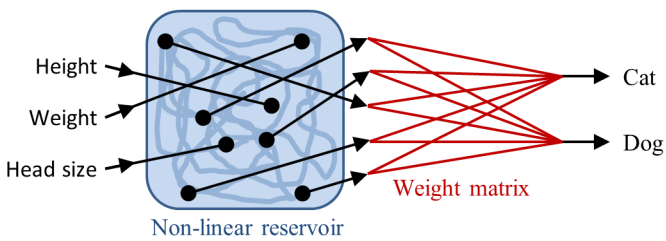


Figure 1.22: A simple reservoir computing approach for classification. Red lines are tunable weight matrices.

Figure 1.22 shows a simple example of reservoir computer. The reservoir (in blue) is a highly non-linear physical system with complex internal dynamics. It is typically made of a set of ran-

domly connected "neurons", or a non-linear continuous medium. The inputs are randomly connected to different "points" of the reservoir, inducing perturbations depending on their values. The response of the reservoir is measured at random points, and each readout value corresponds to a different unknown non-linear transformation of the presented input vector. A simple linear regression between the readout and the expected output for a set of known examples yields the output weight matrix (in red in Figure 1.22). When the dimensionality of the readout vector is high (5 in the Figure 1.22), this approach can solve complex cognitive tasks at low energy cost.

Implementation with nano-oscillators

A time-dependent reservoir computing approach was experimentally implemented by Torrejon *et al.* [163] using a single vortex spin transfer torque nano-oscillator to perform spoken digit classification with state-of-the art recognition rates. A filtered version of a voice signal is applied as an input current to the oscillator, and the envelope of its voltage is read at several points in time. These readout values are then combined by a weight matrix found by linear regression to provide the outputs.

An alternative approach was proposed and simulated by Coulombe *et al.* [164] using spring-coupled micro-electromechanical oscillators as a time-sensitive reservoir, and yields promising results on spoken digit classification.

The complexity of coupled oscillator dynamics prevents a full analytical description of their dynamics. However, reservoir computing can leverage this complexity to achieve computation without requiring a precise description of the network. Reservoir computing with oscillators is therefore a promising approach for efficient computation.

Summary of the chapter

After an intuitive introduction to oscillations their synchronization dynamics, I presented various results from neuroscience suggesting that oscillations may play a role in cognitive computational tasks performed efficiently by the human brain. I observed that such biological oscillatory phenomena could be used as a source of inspiration for efficient oscillator-based computation.

To understand how coupled oscillators could achieve computation, I explained their phase, frequency and synchronization dynamics through the generic linear Kuramoto model, showing that their rich behaviors could be leveraged for computation even with a small number of oscillators. I also extended this study by defining a more advanced model for non-linear oscillators, the Landau-Stuart model.

Implementing compact oscillator-based computing architectures on chip requires using nano-scale oscillators. In order to provide an overview of the capabilities of existing nano-oscillators, I presented the main categories of suitable oscillator technologies, showing that

several could be used in computational architectures.

Finally, I presented various oscillator-based computational architectures that have been proposed, and listed their advantages and limitations.

Chapter 2

Using coupled oscillators for pattern classification

Life is a constant oscillation between the sharp horns of dilemmas.

H. L. MENCKEN

"THE SECOND CHAPTER *of this thesis presents an oscillator-based architecture that performs pattern classification, and evaluates its compatibility with the constraints of nano-technologies.* "

THIS CHAPTER introduces an oscillator-based classification architecture, extends it to function with nano-oscillators and investigates its compatibility with constraints linked to nano-technology. It covers:

1. the presentation of a promising architecture that leverages synchronizations in a network of oscillators to achieve classification and only requires natural frequency control and synchronization readout,
2. adapting this architecture to function with existing nano-oscillator technologies, including the definition of robust synchronization detection circuits,
3. investigating the compatibility of the resulting oscillator-based classification architecture with the constraints of nano-technologies including phase noise, device and coupling variability, phase shifts, geometrical effects, non-linearities as well as scaling capabilities,
4. proposing an alternative readout method inspired by the EEG signals in the brain that uses the mean field signal of the oscillator network as an output.

Introduction

As introduced in Section 1.4.6, most of the proposed oscillator-based pattern classification architectures rely on dynamically tuning couplings, or directly acting on the phases of oscillators, which is hard to implement with current technologies. In 2011, Vassilieva *et al.*[160] proposed a novel classification scheme that only relies on tuning natural frequencies, and reading the synchronization state of a network of weakly, uniformly coupled oscillators. As neither couplings nor individual oscillator phases need to be tuned, this approach appears promising in the context of nano-technologies. However, the seminal work of Vassilieva *et al.* was done in a purely mathematical context and does not cover realistic synchronization detection methods nor the resilience to the imperfections of current nano-oscillator technologies.

In this chapter, I reinterpret and extend the approach of Vassilieva *et al.* into a full-featured nano-oscillator based pattern classification architecture. I first show that this architecture can be implemented by leveraging a general property of nano-oscillators allowing convenient frequency tuning through an external bias (usually current or voltage) and propose robust readout circuits that perform pairwise synchronization detection using elementary logic circuits. I then investigate the robustness of the resulting architecture to the typical constraints of current nano-technologies including phase noise, device and coupling variability, phase shifts, geometrical effects, non-linearities as well as the scaling capabilities of the classifier. I also introduce a modified version of this architecture, which relies on a different readout method inspired by the EEG signals from the brain and uses the mean field signal of the oscillator network as an output for classification.

Defining a nano-oscillator based pattern classifier

The pattern recognition approach at the basis of this study is presented in Figure 2.1(a). It consists of a core network of oscillators that are coupled by fixed bidirectional weak connections. All-to-all coupling is considered in this study unless stated otherwise. Such all-to-all coupling can be realized by electrical coupling of resistive elements [110] as will be presented in Chapter 3, or by summing the signals of all oscillators and re-injecting the total signal by the use of external circuitry [159]. As their natural frequencies are spread, no synchronization between the oscillators happens spontaneously. A set of input oscillators, with stronger couplings to the core oscillators, is used to perturb the core network. The input to the network is encoded as the natural frequency of these input oscillators.

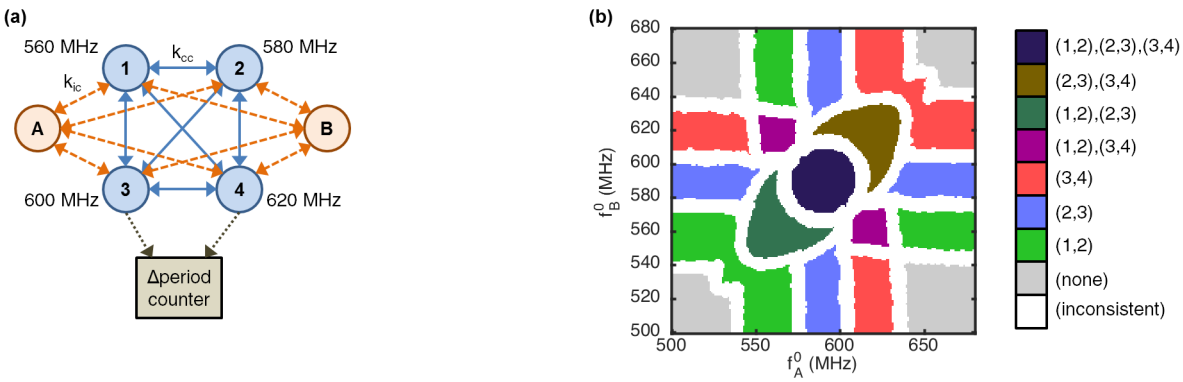


Figure 2.1: (a) Diagram of the oscillator network showing input oscillators A,B and core oscillators 1,2,3,4. (b) The output synchronization readout map of the ideal reference oscillator network. Each color represents a different set of synchronized pairs of core oscillators. The gray regions represent areas where none of the oscillator pairs is synchronized. The white regions correspond to situations where the evaluated synchronization state is sensitive to the initial conditions.

The readout map of Fig. 2.1(b) illustrates the typical response of a system of 4 core and 2 input oscillators. Under the influence of the inputs (or stimuli), synchronizations between oscillators of the core network emerge. Each color in the map represents a unique set of resulting synchronized pairs of core oscillators. The different colored regions correspond to the different output synchronization patterns triggered by the choice of input natural frequencies f_A^0 and f_B^0. On the sides of the map, only one input oscillator interacts with the core network, as the natural frequency of the other input oscillator is very different from the natural frequencies of the core oscillators. This results in the synchronization regions , , involving the synchronization of a single pair of core oscillators. The central regions , , result from the interaction of multiple core oscillators with both input oscillators. They correspond to the synchronization of more than a single pair.

The resulting list of synchronized pairs of core oscillators corresponds to the output synchronization pattern of the system and is strongly dependent on the natural frequencies of the

input oscillators. It is therefore a signature of the input stimuli, and can be used to achieve classification/recognition of the presented input.

By performing an associative operation between a set of analog inputs and a limited set of synchronization patterns, this system behaves as a hetero-associative memory. This scheme can be used for different kinds of multi-class classification problems, such as image classification [160], spoken word classification or decision making. **Shaping the response of such a network so that it outputs the desired class on presented examples from a given data set requires a learning process that is investigated in Chapter 4. This second Chapter only investigates the system at the architectural level.**

Example : A dummy classification task

To provide a better understanding of the classification process using the presented oscillator based classifier, I illustrate it with a dummy task involving the classification of made-up plants into 8 different species depending on 2 of their features: petal length and width.

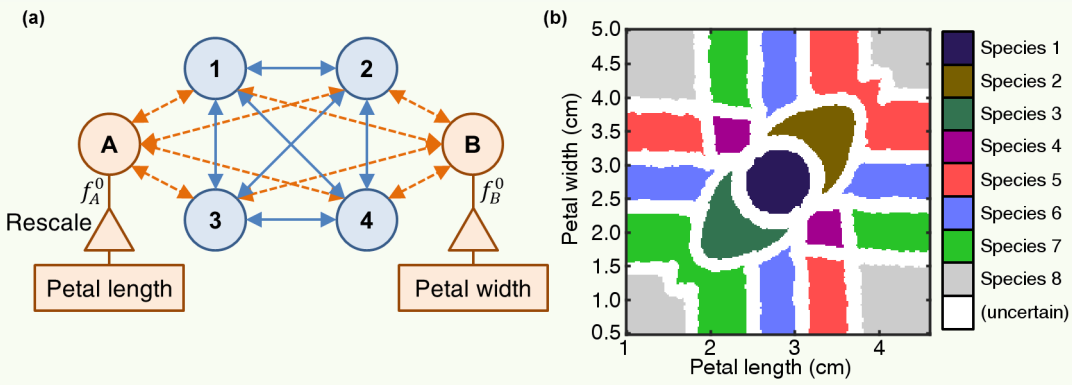


Figure 2.2: Dummy plant species classification task with the oscillator-based classifier. **(a)** Oscillator-based classifier architecture. **(b)** Resulting classification map.

Figure 2.2(a) introduces the architecture where the input features of the presented plant are converted to frequencies by a linear rescaling operation and applied as the natural frequencies of the input oscillators.

Figure 2.2(b) shows the resulting classification map: depending on its features, the presented plant is classified as belonging to a given species. For example a plant with a petal length of 2 cm and a petal width of 1 cm is classified as belonging to “Species 7”.

Note again that this is a dummy task for illustration purposes that was designed to match the existing reference map. On a real task, this response map needs to be tuned to match the task by adjusting the natural frequencies of the core oscillators through a learning algorithm. This is the focus of Chapter 4.

In this work, except when assessing the impact of oscillator non-linearities, I simulate the oscillator-based architecture by using a generalized version of the Kuramoto model defined by equation (2.1) describing the evolution of oscillator i's phase θ_i as a function of its intrinsic frequency f_i^0 and the influence of the other oscillators $\sum_j k_{i,j} \sin(\theta_j - \theta_i + \phi_{i,j})$. The coupling from oscillator j to oscillator i is modeled through the coupling strength $k_{i,j}$ and the coupling phase shift $\phi_{i,j}$. Unless stated otherwise, a phase shift $\phi_{i,j} = 0$ is assumed. When specified, a phase noise term is also included in the simulation:

$$\frac{\dot{\theta}_i}{2\pi} = f_i^0 + \sum_j k_{i,j} \sin(\theta_j - \theta_i + \phi_{i,j}) + \text{Noise}(\text{FWHM}, t)\,. \tag{2.1}$$

The noise is quantified by the FWHM quantity defined to be the Full Width at Half Maximum of the power spectrum density of an isolated oscillator.

Numerical integration

The integration scheme follows:

$$d\theta_i = \left[2\pi \times f_i^0 + \sum_j 2\pi \times k_{i,j} \sin(\theta_j - \theta_i + \phi_{i,j}) + \sqrt{\frac{2\pi \times \text{FWHM}}{dt}} \times \mathcal{N}(0,1) \right] \times dt\,, \tag{2.2}$$

where $\mathcal{N}(0,1)$ is the normal distribution with mean of 0 and a standard deviation of 1.

Simulations not involving noise use fourth order Runge Kutta ODE integration. When noise is included, the specific Euler-Maruyama SDE integration scheme is used instead.

I consider that the core network is initially in a random unsynchronized state before the input stimuli are applied. After applying the stimuli and waiting for the stabilization of the frequencies in the network during a given stabilization time, its state is read by pairwise evaluation of synchronization between core network oscillators. This requires designing a synchronization detection scheme.

A robust synchronization detection scheme

Achieving readout with fast, possibly noisy oscillators in real time is an equally important challenge as the pattern recognition itself. In this section, two simple, easy to implement schemes for the evaluation of the degree of synchronization between pairs of oscillators are proposed and investigated. Then, the quality of the readout of each scheme is evaluated and compared in the case of a network of noiseless oscillators as well as in a more realistic case of noisy oscillators.

Schemes for evaluating synchronization patterns

The readout process of such an architecture involves pairwise synchronization detection between core oscillators, and the output is the resulting list of synchronized pairs. In the literature, noise was observed to not only prevent synchronization [8] in weak coupling conditions but also to induce fluctuations in the synchronization pattern readout during evaluation time, showing transitions in regions where multiple synchronization attractors are available [165]. Synchronization between oscillators in its strictest definition is therefore hardly achieved, as perfect phase-locking seldom occurs.

Thus, a weaker definition of synchronization is needed and should be based on a measure of the degree of synchronization between two signals. For instance, statistical methods such as the variance measure introduced in [160] can be used to define quasi-synchronization between a pair of oscillators {n,m}:

$$\mathrm{Var}_\tau(\sin(\theta_n - \theta_m)) < \epsilon_v\,, \tag{2.3}$$

where the variance is evaluated during a limited time τ. A threshold $\epsilon_v \in [0;0.5]$ is chosen to discriminate quasi-synchronized pairs from non-synchronized pairs, $\epsilon_v = 0$ corresponding to a perfect synchronization requirement. This definition was shown to allow the detection of a rich set of synchronization patterns in weakly coupled networks [160]. However, the hardware implementation of such a detection scheme would involve a combination of complex circuits, or an external computing unit: it is not a reasonable readout technique for an efficient physical implementation.

Here, I propose two other quasi-synchronization detection schemes based on operational principles compatible with an easy CMOS implementation and compare their performances with the variance-based method. The oscillator signals are digitized using Schmitt triggers, intended to increase the noise resilience of the readout, after which a rising edge detection is performed to obtain a single pulse per period of the signals.

A first synchronization evaluation approach that is further called “direct counter” is presented in Fig. 2.3(a). This technique aims at evaluating the difference ΔN_τ between the number of periods of the two signals during a given amount of time τ. The counting is achieved by incrementing or decrementing a counter at each rising edge of the respective signals. The result is then compared to a threshold ϵ_d, and the two oscillators are considered synchronized if $|\Delta N_\tau| < \epsilon_d$.

A second scheme that is further called “flip-flop counter” is presented in Fig. 2.3(b). This technique exploits the fact that if two signals are synchronized, their rising edges should alternate. A counter is then incremented each time two consecutive rising edges of the same signal are not separated by a rising edge of the second. Again, after the evaluation time τ, the two oscillators are considered synchronized if the final value of the counter is strictly lower than a given threshold ϵ_f.

Because they use only limited information from the signals, the two counter-based schemes

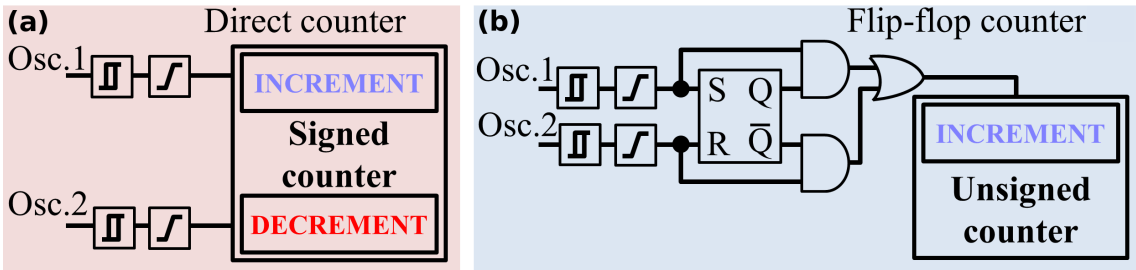


Figure 2.3: Illustration of proposed schemes for evaluating the degree of synchronization between pairs of oscillators: (a) direct counter evaluation technique, (b) flip-flop counter evaluation technique.

differ conceptually from the variance measure which depends on the full time trace of the phase difference between the two oscillators. A major difference exists between the two counter-based schemes: while the direct counter method measures an average frequency difference during the total evaluation time, the flip-flop counter takes into account and sums up every detected local desynchronization event. Yet, the physical implementation of the flip-flop counter would require less components, mainly because of the simpler, unsigned counter it uses.

Equivalence of the detection schemes

To compare the synchronization evaluation schemes, we first investigate the simplified case of a single input oscillator $\{A\}$ and two core oscillators $\{1,2\}$, as illustrated in Fig. 2.4(a). Core oscillators' natural frequencies are set to $\{f_0^{(1)}, f_0^{(2)}\} = \{560, 580\}$ MHz and the input oscillator's natural frequency $f_0^{(A)}$ is swept from 470 to 670 MHz.

Fig. 2.4(a) captures the synchronization phenomenon between the three oscillators by showing the evolution of their average frequencies. While the coupling between core oscillators {1,2} is initially too weak for them to synchronize, they are eventually brought to synchronization when the input oscillator's natural frequency lies in a limited range (552 to 588 MHz).

For each simulation, the oscillator network dynamics are computed during 1 µs. After a 0.5 µs cool-down time to reach the convergence of the network dynamics, the three detection schemes are evaluated during $\tau = 0.5$ µs between core oscillators {1,2}. Their outputs are plotted in Figs. 2.4(b,c,d) before the thresholding operation.

The three curves appear extremely similar. It is surprising to note that, while variance measure and counter approaches use different basic principles, the obtained curves can actually almost be superimposed. All exhibit a very distinct dip to a zero-value when the two core oscillators are synchronized, and a high plateau value when the oscillators are desynchronized. In the intermediate range, where the oscillators are quasi-synchronized, the outputs show a progressive increase, allowing to define thresholds that will discriminate whether the oscillators are quasi-synchronized or not.

From these curves, we choose equivalent thresholds for the three detection schemes: $\epsilon_v =$

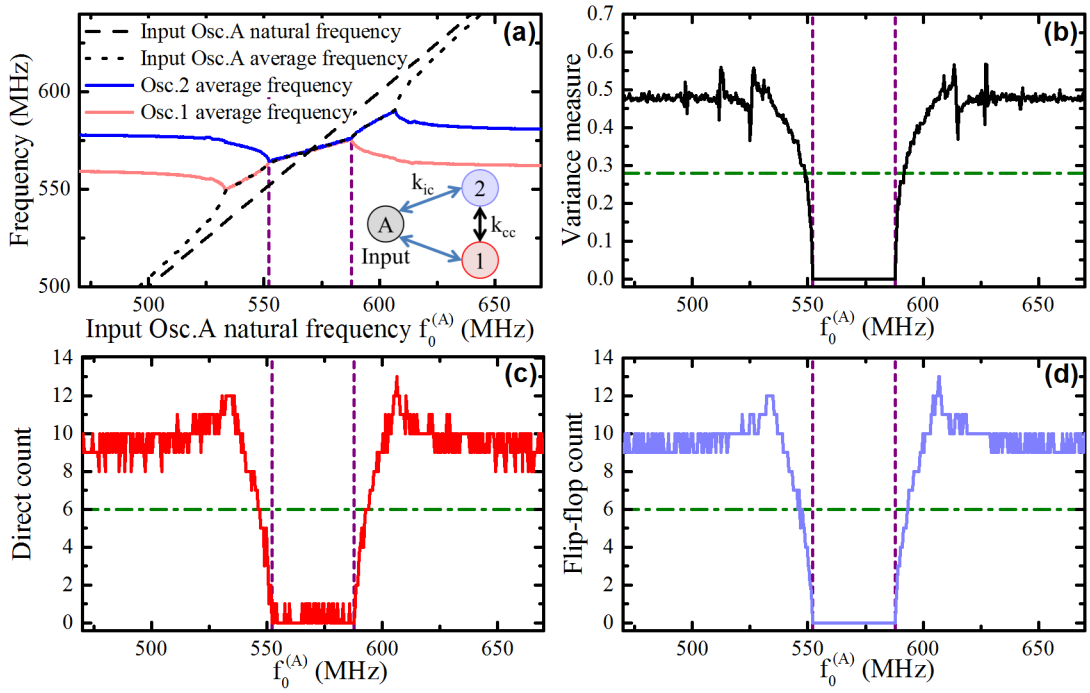


Figure 2.4: Simulations in a simplified, noiseless situation with two core oscillators with fixed natural frequencies $\{f_0^{(1)}, f_0^{(2)}\} = \{560, 580\}$ MHz, and one input oscillator whose natural frequency $f_0^{(A)}$ is varied. (a) Average frequencies of the three oscillators as a function of $f_0^{(A)}$. The quasi-synchronization of the two core oscillators is evaluated using the three readout schemes, and their outputs are plotted as a function of $f_0^{(A)}$ for (b) variance measure, (c) direct counter, and (d) flip-flop counter schemes. Vertical purple dotted lines indicate the range of perfect synchronization between oscillators 1 and 2. Horizontal green dotted lines correspond to selected thresholds under which oscillators are declared quasi-synchronized.

0.28 for the variance measure scheme, $\epsilon_d = 6$ for the direct counter scheme, and $\epsilon_f = 6$ for the flip-flop counter scheme. In the following we use these threshold values if not stated otherwise.

Pattern recognition and comparison of the readout schemes

The three readout schemes can now be evaluated on the full coupled oscillator network of Fig. 2.3(a), introduced in Section 2.2.

Readout maps in the absence of noise

Figs. 2.5(a,b,c) present the readout maps of the synchronization patterns in the core network, as a function of the input oscillators' natural frequencies $\{f_0^{(A)}, f_0^{(B)}\}$ in the case of noiseless oscillators ($\eta = 0$). They are obtained using the variance measure, direct counter and flip-flop counter detection schemes respectively.

As in the previous simulation, the oscillator network dynamics are simulated for 1 µs for

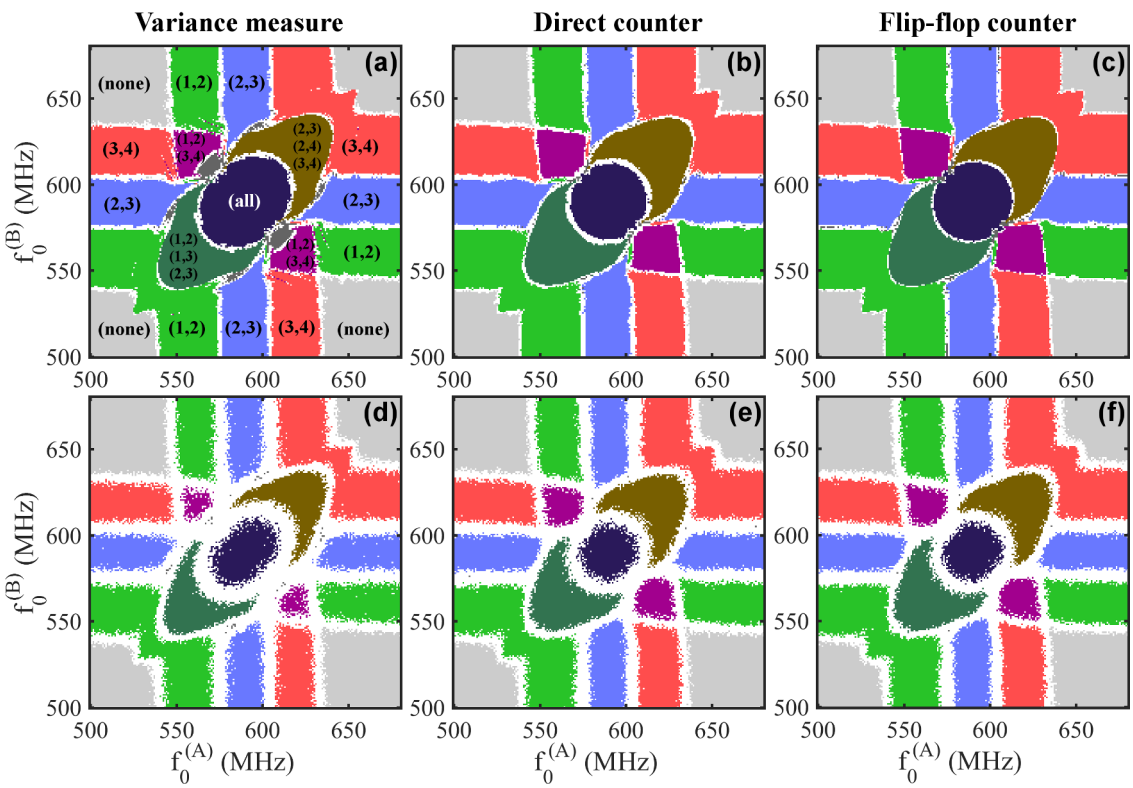


Figure 2.5: Readout maps showing the distribution of synchronization patterns among core oscillators as a function of the two input oscillators' natural frequencies $f_0^{(A)}, f_0^{(B)}$, as detected by the three readout protocols. Each color is associated to a single synchronization pattern, as specified on Fig.(a). The (a,b,c) maps are obtained in a situation with noiseless oscillators, (d,e,f) maps are obtained for oscillators with phase noise corresponding to FWHM=1 MHz. Readout maps are evaluated respectively using: (a,d) the variance measure scheme, (b,e) direct counter scheme, and (c,f) flip-flop counter scheme.

each point in this map. After a $0.5\mu s$ cool-down time to wait for the convergence of the network dynamics, the three detection schemes are performed on each of the six pairs of core oscillators during $\tau = 0.5\mu s$, and the results are compared to their respective thresholds. For each simulation, an output list of synchronized pairs is then given by each readout scheme. To account for the robustness of the readout results to initial conditions, each point on the map is simulated ten times with random initial phases. If the ten simulations do not result in the same output synchronization pattern, the point is discarded as "inconsistent" and left blank on the map. If the ten simulations yield identical results, the point is then colored on the map according to the output pattern.

Note : Computing maps is expensive

Producing these 200 × 200-point maps, to allow precise assessment of the coupled oscillator network behavior, comes at a high computational cost as it requires $200 \times 200 \times 10 = 400,000$ independent simulations per map. For optimal efficiency, the simulations were performed simultaneously on the 2880 cores of a nVidia Tesla K40 GPU, using the Cuda Thrust C++ library.

In this noiseless example, the three evaluation schemes yield rich output maps, with large and well-defined regions associated to different synchronization patterns. The boundary regions (blank) where no repeatable readout is obtained are relatively small. These results show the efficient recognition capability of the oscillator network, as originally presented in Section 2.2. Indeed, it spontaneously discriminates inputs through the establishment of synchronization patterns in its core. Additionally, it is remarkable that all of the introduced synchronization detection schemes are operational and lead to highly similar readout maps.

The capabilities of the oscillator network associated with each readout scheme are evaluated through the number of classes of patterns the architecture is able to discriminate, *i.e.* the number of regions with different readouts that appear on the map. In this counting, we choose to ignore isolated points, as well as porous regions where consecutive points do not consistently yield the same output. To do so, a filter is applied on the readout maps, as illustrated on Fig. 2.6, that only keeps regions which yield identical outputs in an at least 3 MHz radius area. This ensures that the counted classes are represented by large and continuous regions that are tolerant to small input variations.

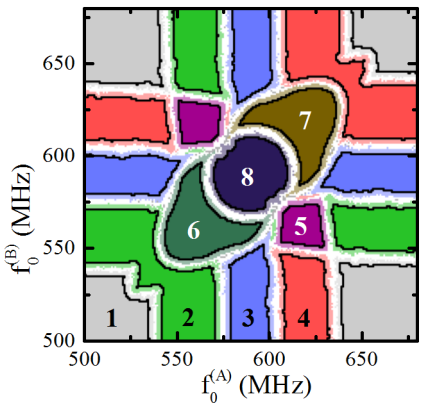


Figure 2.6: Effect of the filter applied to the readout maps for counting discriminated patterns. This is illustrated in the case of the readout map obtained for a noiseless network and the flip-flop counter detection scheme (see Fig. 2.5(c)). Saturated regions delimited by a black line are considered robust and kept by the filter, pale areas are ignored. The numbers on the map index the eight unique discriminated patterns.

In the case of both counter-based readouts, eight patterns are discriminated, each one be-

ing associated to a different synchronization pattern. Meanwhile, a ninth output pattern appears for the variance-based readout (■ in Figure 2.5(a)). This is due to the intrinsic non-transitivity of quasi-synchronization detection methods: for example synchronizations can be detected between oscillators 1 and 2, as well as between 2 and 3, without any synchronization being detected between 1 and 3. Measured non-transitive synchronization patterns are however rare and generally unstable with the low synchronization detection thresholds being used.

Readout maps in the presence of noise

When considering hardware implementations, especially based on nanotechnology such as proposed in Section 1.4, the consequences of the phase noise of the oscillators, which can significantly perturb the network dynamics, needs to be considered. To account for it, we reproduced the simulations of the network dynamics as well as the three readout schemes including a non zero noise corresponding to oscillators' FWHM = 1 MHz.

We show in Fig. 2.5(d,e,f) the readout maps obtained for the noisy oscillator network. Again, for all three detection schemes, the three maps remain very similar. Compared to the noiseless network, the class regions are sensibly reduced, and the blank (inconsistent) regions are getting wider, as could be expected. Indeed, as the noise increases, the repeatability of the readouts becomes an increasing issue. It notably has an impact on the readout map obtained through the variance-based scheme. As one can see in Fig. 2.5(d), the ninth synchronization pattern is no longer observed, and the purple region has become particularly porous.

Testing detection schemes against different network parameters

Modifying the coupling strengths between oscillators in the network changes the distribution of synchronization patterns, and also has an influence on the relative phase dynamics between coupled oscillators, and subsequently on the synchronization readout. To compare the detection schemes in the case of other network configurations, we repeated the simulations of noisy networks with varying input-core coupling strengths k_{ic} and core-core coupling strengths k_{cc}. Fig. 2.7(a) shows the number of discriminated classes of inputs when $k_{cc} = 4$ MHz is kept constant and k_{ic} is varied. Fig. 2.7(b) shows the number of discriminated classes of inputs when $k_{ic} = 12$ MHz is kept constant and k_{cc} is varied.

The two counter-based readout schemes provide quasi-identical results. Both plots show that optimal coupling strengths, for which a maximum number of classes can be discriminated, fall in the same range for the three readout schemes. However, the counter-based schemes lead to the discrimination of a higher number of classes than the variance-based one in large ranges of coupling strengths. These observations suggest that the counter-based definitions of quasi-synchronization might be more robust than the variance-based one. We discuss and interpret this idea in the next section.

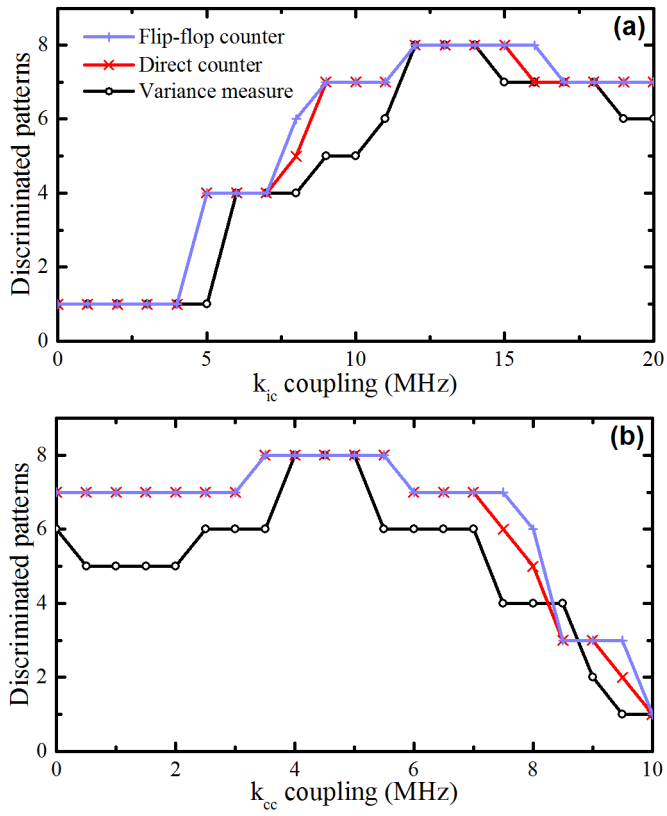


Figure 2.7: Number of discriminated patterns within the core network as a function of (a) input-core k_{ic} coupling and (b) core-core k_{cc} coupling as evaluated using the three detection schemes. Simulations performed with FWHM = 1 MHz.

Sensitivity of the readout schemes to noise and parameters

In this subsection, an in-depth evaluation and comparison of these schemes is performed, and their applicability for a final hardware system is discussed.

Noise resilience of the readout schemes

To assess the relative influence of phase noise on the three readout schemes, we repeat the simulations of the noisy network with increasing noise levels and plot the evolution of the number of classes discriminated by each readout method in Fig. 2.8. The three schemes show different resiliences to noise.

For low noise levels, the variance-based scheme shows the lowest resilience, as it is the first detection scheme that stops being able to discriminate eight classes. Indeed, the variance measure is strongly affected by fluctuations appearing in the phase difference dynamics. Therefore, when looking at the outputs before thresholding, we observe an increasing spreading of the results for the ten repeated simulations with increasing phase noise. This induces spurious detections of synchronization or desynchronization producing many inconsistent points, which

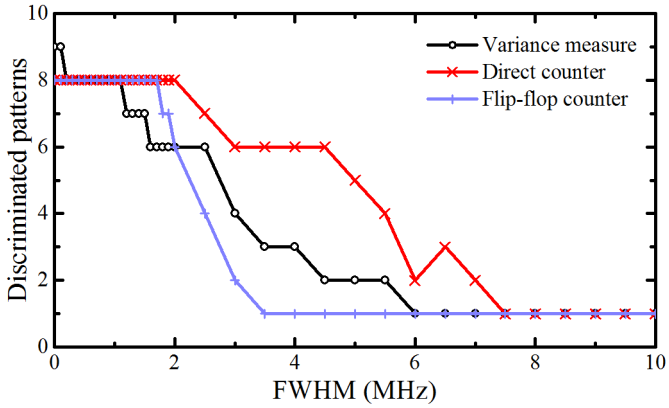


Figure 2.8: Number of discriminated patterns as a function of oscillators' FWHM, as evaluated using the three detection schemes.

leads to the disappearance of some classes in the readout map.

The two counter-based schemes keep their ability to discriminate eight patterns for higher noise levels, even beyond 2 MHz for the direct counter scheme. They evaluate the exact number of desynchronization events and are not sensitive to their dynamics. However, the flip-flop counter scheme appears to fail very rapidly when the FWHM of the oscillators goes above 2 MHz, even faster than the variance-based scheme. In the flip-flop counter detection method, every desynchronization event, *i.e.* phase slip between the two signals, is detected and counted. As the noise level increases, many spurious desynchronization events get detected, which eventually leads the counter to go above the threshold. Only strong synchronizations are then detected. This observation suggests that the threshold level should be raised to adapt to high phase noise oscillator networks.

Because it simply evaluates an average frequency difference between the two noisy signals, the direct counter scheme is the one showing the best resilience to noise. It is still able to discriminate six classes of inputs for oscillators with FWHMs up to 4.5 MHz, when other readout methods only discriminate one or two classes.

Influence of the threshold

We have seen that in the presence of noise, the three readout methods may suffer difficulties to detect quasi-synchronization. In these conditions, the initial choice of the thresholds, in an oversimplified case and for a noiseless network, should be reconsidered. We now analyze the impact of the choice of the threshold on the number of recognized patterns. Fig. 2.9(a,b) shows the total number of discriminated classes for the FWHM = 1 MHz (solid lines) and for the FWHM = 3 MHz (dotted lines) noisy networks as a function of the variance and counter thresholds.

In the case of low phase noise (FWHM = 1 MHz), all three readout techniques identify the maximum number of patterns (eight) for a reasonable range of thresholds. When increasing the

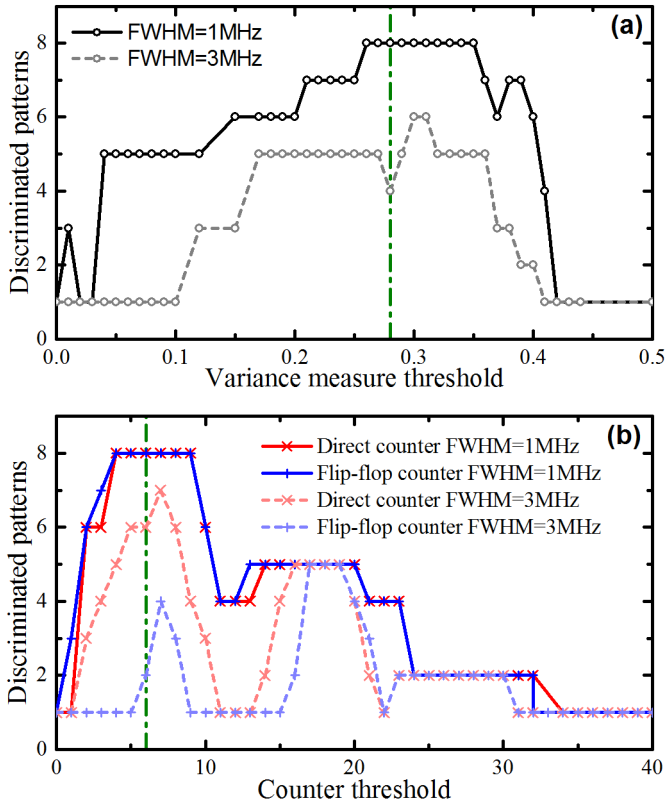


Figure 2.9: Number of discriminated patterns, (a) as a function of the threshold ϵ_v when using the variance measure technique, (b) as a function of the counter threshold when using the direct counter and flip-flop counter techniques. The FWHM of the oscillators is set to 1 MHz (solid lines) and 3 MHz (dotted lines). Vertical green dotted lines indicate the thresholds that were chosen in Section 2.3.3.1 and used in the simulations to obtain readout maps.

thresholds, the variance based detection method rapidly fails when the threshold ϵ_v is chosen above 0.4. On the other hand, the thresholds for counter-based methods can be increased even above 20 without failing (five patterns are still detected), much above the maximum counts observed in 2.3.3.1.

As noticed in Section 2.3.4.1, when the FWHM of oscillators reaches 3 MHz, the three readout methods detect different numbers of patterns using the thresholds chosen initially: six for the direct counter scheme, four for the variance-based scheme, and only two for the flip-flop counter scheme. The presented plots show that these initial choices are not adapted to the higher noise case, and that other optimal thresholds can be found. The direct counter approach still shows the best resilience to noise as up to seven synchronization patterns can be read, while the variance measure approach is limited to detecting up to six patterns. In the counter-based approaches, the optimal thresholds are close to our initial choices, yet the optimal ranges are substantially reduced. For the flip-flop counter scheme, the optimal threshold is found around 18, confirming that high noise induces the spurious detection of many desyn-

chronization events.

Influence of the evaluation time

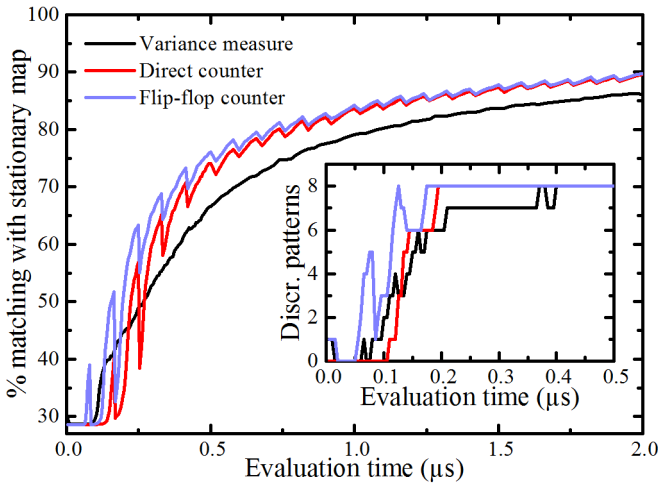


Figure 2.10: Matching percentage between the readout map obtained for a limited integration time τ and the readout map obtained for a long $100\mu s$ integration time, as a function of τ and for the three detection protocols. The inset shows the number of discriminated patterns detected by the three methods as a function of integration time. Simulations are performed with FWHM = 1 MHz.

The synchronization evaluation time τ is an important trade-off for the readout operation, between the speed for recognition and the robustness of the results. Simulations of the recognition networks are performed with varying evaluation time, between 0 and 2μs. The cool-down time is kept to 0.5μs, and the FWHM of the oscillators is 1 MHz. The results are compared to the readout obtained in the case of a long evaluation time $\tau = 100\mu s$ for which the readout is considered stationary and further used as a reference.

In Fig. 2.10, we plot the percentage of matching points between the obtained readout maps and the reference map, as a function of τ and for each readout scheme. The figure inset also shows the evolution of the detected number of patterns as a function of τ. For equivalence in the case of the counter-based detection schemes, the thresholds are adjusted as τ varies so that $\lfloor \epsilon_{\{d,f\}}/\tau \rfloor$ is kept equal to $12\mu s^{-1}$.

We observe a fast convergence of the readout maps for evaluation times up to 0.5μs above which the convergence starts slowing down. After 2μs, the counter-based readout schemes reach 90% matching with the reference, while the variance-based scheme lags behind. For the counter-based methods, the maximum number of patterns (eight) is already detected when $\tau > 0.2\mu$s, while $\tau > 0.4\mu$s is needed for the variance measure scheme. In all case, the choice of $\tau = 0.5\mu$s then offers a reasonable trade-off as the maximum number of patterns is already discriminated.

Choosing the right detector

In situations with oscillators with no or low phase noise, the readouts of all three techniques appear very similar. Although variance measure is the most complex, it is not the most robust to noise, and may actually identify less synchronization patterns in some situations with intermediate noise levels and very weak coupling between core oscillators. The variance measure also appears to converge slower than counter-based approaches, hence requiring longer evaluation time for equivalent precision.

When both counter-based protocols show equivalent results, the flip-flop counter is the best choice for hardware implementation, as it relies simply on an unsigned counter. However the flip-flop counter protocol appears to fail at high noise levels, when it detects a lot of spurious desynchronization events. Nonetheless, the direct counter protocol also shows strong resilience to high noise, again better than the variance measure. The choice between these two techniques should therefore be based on the amount of noise. **For simplicity and consistency of the results across a wide range of noise levels, the direct counter scheme is used in the rest of this thesis.**

This study involved the measure of the synchronization state of all pairs of core oscillators to obtain comprehensive results on the behavior of the detectors. However, from the scalability point of view, using $\frac{1}{2}N \times (N-1)$ detection circuits to read the synchronization state of every pair of N oscillators is a major constraint. In practice, this approach is highly redundant as virtually all stable synchronization states are transitive. Moreover, core oscillators are identical and only defined by their natural frequencies, they can therefore always be sorted according to their natural frequencies. These properties allow retrieving the full synchronization maps by measuring only consecutive pairs of core oscillators: (1,2),(2,3),(3,4) in the case of 4 core oscillators. In that case, the number of detectors scales as $N-1$, which represents an important scalability improvement. This approach is used in the rest of the thesis.

The readout scheme being defined (and published [166]), the robustness and scalability of the whole architecture can now be evaluated within the constraints of nanotechnologies.

Is this architecture nanodevice-ready ?

This section investigates, through extensive numerical simulations and theoretical analysis, the resilience of such a pattern recognition scheme to phase noise, variability, and non-linearities present in nanotechnologies. It then assess the scaling properties of the network by evaluating the evolution of its pattern discrimination capacity with increasing numbers of oscillators. The geometrical effects that can appear in nano-device architectures are also studied.

Behavior under the presence of noise

As observed in the previous section, noise is an important challenge in hardware implementations of oscillator-based computing. To complete the study of the influence of noise in this architecture initiated in Section 2.3.4.1, we simulate the reference architecture for increasing phase noise levels on both input and core oscillators. We then compare the obtained readout map to the ideal map evaluated in the noiseless case (Fig.2.1).

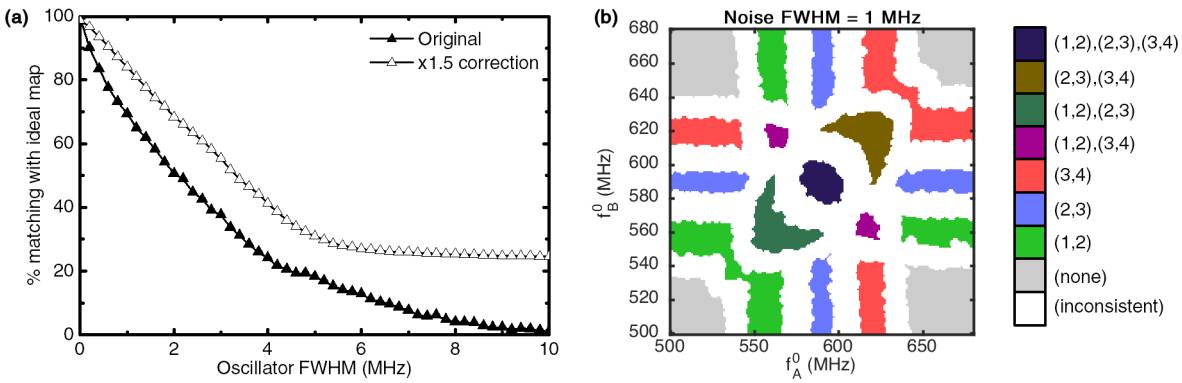


Figure 2.11: (a) Percentage of matching points to the ideal map with respect to noise FWHM for the original (▲) and the adjusted (△) systems. (b) An example of synchronization readout map under oscillator noise FWHM = 1MHz in the original system.

The typical effect of noise is visible on the map of Fig. 2.11(b) obtained in the case of FWHM = 1MHz: it erodes the surface of output synchronization pattern regions. Noise particularly affects the points where synchronization was weak, situated at the boundaries of the regions identified on reference map of Fig. 2.1(b). Figure 2.11(a) shows the percentage of matching points to the noiseless ideal map of Fig. 2.1(b) at different levels of noise. These results show that this architecture is resilient to relatively high noise levels, demonstrating 70% matching at FWHM = 1MHz, corresponding to oscillators with $f/\text{FWHM} \geq 500$. As a comparison, typical auto-oscillating magnetic nano-devices have shown $f/\text{FWHM} \geq 6000$ [65] (FWHM lower than 100kHz at 457MHz, see Section 1.3.5), and mechanical oscillators [87] can achieve $f/\text{FWHM} \approx 50 \times 10^3$ (see [85] and Section 1.3.3), which makes these technologies good candidates for this architecture.

In order to fully mitigate the effects of noise, we found that the distance between the natural frequencies of core oscillators can be increased. Figure 2.11(a) shows results for a case in which the couplings, distances between natural frequencies, and input frequency sweep ranges have been multiplied by a factor of 1.5 (△). This system is notably more robust to noise than the initial system. However, this is a trade-off as it requires accessing a larger range of natural frequencies for the core oscillators, and ensuring stronger couplings.

As a conclusion, the architecture is robust to phase noise when the minimal interdistance between core oscillator natural frequencies is defined according to the noise level.

Numerical integration parameters

The time-step used that ensures convergence of both ODE and SDE integration schemes is dt = 100ps. The total simulation time was 1μs of which the first 0.5μs are the transient stabilization time after which synchronization detection counting starts. This waiting time was chosen to ensure the stabilization of the dynamics on 1,000 simulation runs of the reference system with different random initial conditions. The total simulation time corresponds to approximately 600 periods of the oscillators and is chosen to be a realistic scale for real-life implementations. Two oscillators are considered synchronized when the final absolute value of their counter is strictly less than 6, that is less than about 2% difference in their number of periods.

Effects of natural frequencies variability

As will be studied in Chapter 4, tuning the sizes and positions of the readout synchronization pattern regions requires the ability to set the natural frequency of every core oscillator. This is done either at design time by geometrical or material engineering of each oscillator, or by relying on the knowledge of the natural frequency tuning function f^0(bias) of every oscillator. Both approaches are prone to device variability, which can lead to random shifts in the natural frequencies of the oscillators. As the distances between frequencies in the core network are critical parameters, this variability can induce behavioral changes.

To study the effects of natural frequency variability, the readout map was computed for increasing variability factors and compared to the reference map of Fig. 2.1(b). For increasing Δf^0 values, 100 possible outcomes are computed with core oscillator natural frequencies uniformly drawn in the range $[f_i^0 - \Delta f^0, f_i^0 + \Delta f^0]$ where f_i^0 is the target natural frequency of oscillator i. Fig. 2.12(a) shows the degree of matching with the expected readout map as a function of Δf^0. The solid curve represents the average matching for the 100 simulations, while the blue filled region shows the span between the worst and best matching values reached.

For low variability values, the worst case matching first slowly decreases up to approximately ±3MHz, before dramatically falling below 50% for higher variability values. This worst case corresponds to a situation where the natural frequencies of a pair of core oscillators are brought close enough so that undesired synchronization appears even without input stimuli. Nevertheless, the worst case matching stays over 70% up to a variability of ±2MHz, which corresponds to 10% of the initial difference between two consecutive core network natural frequencies.

A typical example of readout map is shown on Fig. 2.12(b) in a case of 2MHz variability. In this map, f_1^0 and f_2^0 are slightly further apart than expected, which reduces the region ■ in which they can synchronize. On the contrary, f_2^0, f_3^0, f_4^0 are closer than expected and regions

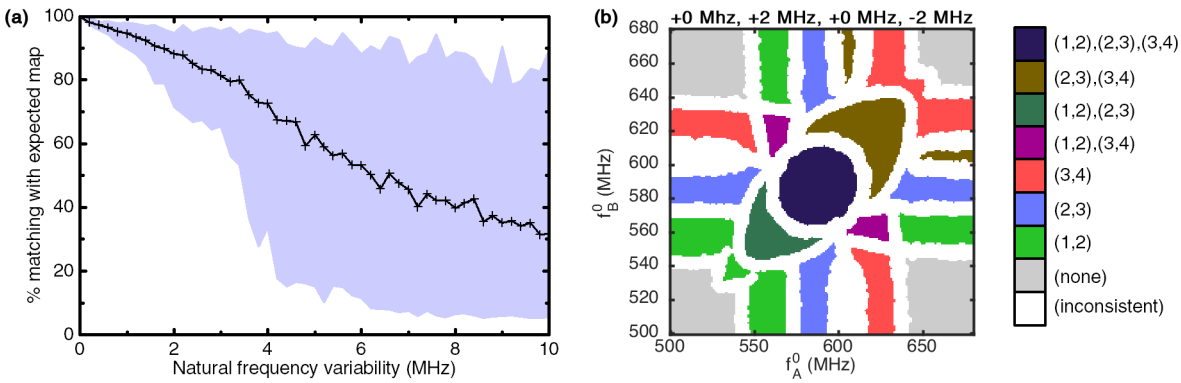


Figure 2.12: (a) Average matching with the expected map for 100 draws on different natural frequency uniform variability ranges, filled between best and worst cases encountered. (b) An example readout map where f_2^0 was shifted by +2MHz and f_4^0 by −2MHz: $\{f_1^0, f_2^0, f_3^0, f_4^0\} = \{560, 582, 600, 618\}$MHz.

where all three are synchronized appear on the sides of the map.

These results show that the system is robust to moderate natural frequency variability, but that high variability on natural frequencies rapidly induces uncertainty on the system response. Again, we conclude that knowledge of the variability level of the system defines the minimal natural frequency difference between consecutive core oscillators.

In the presence of high variations, an effective approach could be to take inspiration from the reinforcement learning algorithms from the machine-learning expertise, and use the readout to implement a feedback of the oscillator biases in order to achieve the correction of the natural frequency values [160, 167], taking advantage from the tunability of the core oscillators.

Effects of coupling constants variability

Variability on the coupling values can arise from variability in inter-oscillator distances (in case of proximity coupling effects), variability in electrical connections, variability in the signal amplitude they emit or variability in their individual response to stimuli [71, 168].

The consequences of such variability were studied by computing the readout map for increasing variability amplitudes in the coupling strengths. For each value of variability amplitude μ ranging between 0 and 100%, 100 simulations were run after randomly drawing individual couplings $k_{a,b}$ in the uniform range $\left[k_{a,b}^0(1-\mu), k_{a,b}^0(1+\mu)\right]$ where $k_{a,b}^0$ is the initial coupling without variability.

A typical map obtained under a coupling variability $\mu = 20\%$ is shown in Fig. 2.13(b). It shows that coupling variability has an influence on the size and shape of the synchronization regions in the readout map. Notably, it has a significant impact on regions corresponding to the synchronization of a single pair of oscillators. Indeed, the smaller the coupling between the two oscillators, the smaller the input frequency range in which they will synchronize is. The readout map also shows that the core network no longer responds symmetrically to the

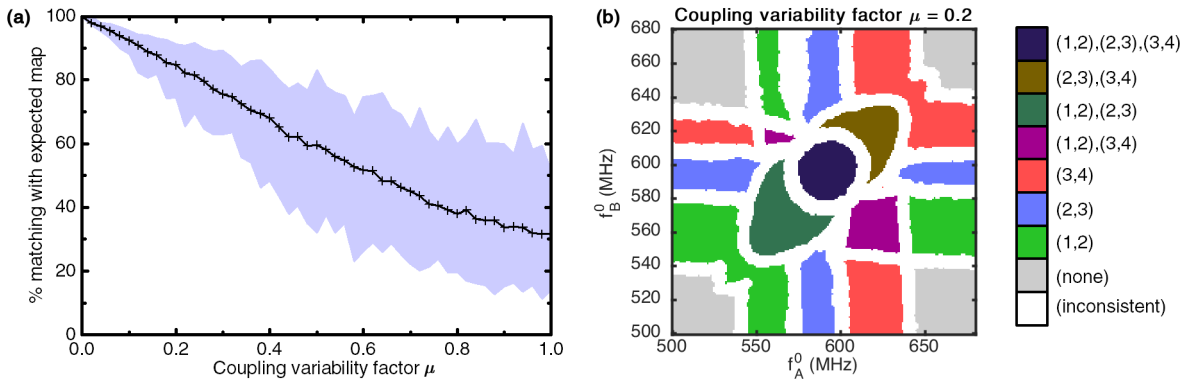


Figure 2.13: (a) Average matching with the expected map for 100 draws with different coupling uniform variability amplitudes μ, filled between best and worst cases encountered. (b) An example readout map for $\mu = 0.2$.

two input stimuli, due to variability in input-core couplings.

The average matching with the expected readout as a function of the coupling strength variability amplitude is shown in Fig. 2.13(a). The filled region represents the span between the worst and best matching rates encountered. These results show that the network is highly robust to coupling variability. Indeed, even 100% variability does not fully hamper the function of the recognition process, as no sudden breakdown is observed. A reasonable 20% variability, even in the worst case scenario that was simulated, leads to more than 70% matching with the expected readout map. This robustness can be attributed to the existence of redundant couplings in the core network, which tends to even out local coupling variations.

These results show that the architecture is robust to coupling variability, and are very encouraging in the context of nanotechnologies, for which precise engineering of individual couplings in a network are complex to achieve.

Effects of coupling phase shifts

As introduced in Section 1.3, coupling between oscillators can arise from many different phenomena including magnetic interaction [111], electrical coupling [169, 170], mechanical coupling [171]. Notably, couplings can have both a conservative and a dissipative component [8, 172], and sometimes involve delays. To fully account for the different types of interactions, a non-zero uniform coupling phase shift term $\phi_{i,j} = \phi$ is added in the solved Kuramoto equation 2.1, and its influence on the system is assessed through the following simulations. The readout map was simulated for different values of the coupling phase-shift term ϕ, in the ideal case of noiseless oscillators and with no variability.

Figure 2.14(a) shows the evolution of the degree of matching of the readout map to the ideal map of Fig. 2.1(b), as well as of the number of different discriminated synchronization patterns, as a function of the coupling phase shift value. Results show that the phase-shift has significant effects on the response of the system, as the matching with the reference map drops

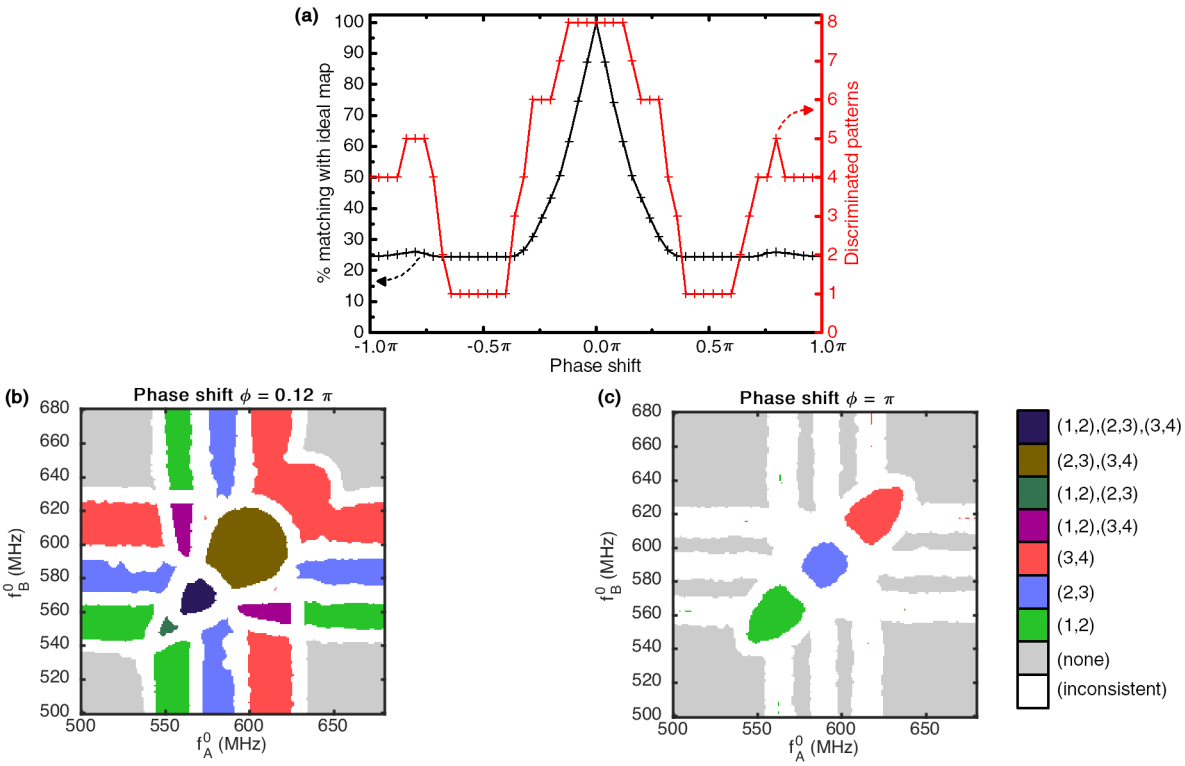


Figure 2.14: (a) Matching with the ideal map (+) and total number of different discriminated patterns (+) in the readout map as a function of phase shift ϕ. (b) An example readout map for a phase shift $\phi = 0.12\pi$. (c) Example readout map for $\phi = \pi$.

with increasing ϕ. Notably, it drops under 50% for $|\phi| > \pi/6$. It then reaches a plateau at 25% of matching for $|\phi| > \pi/3$ where only the areas of the map where no synchronizations are present are consistent with the ideal readout map.

The number of discriminated synchronization patterns is also strongly affected by the appearance of a coupling phase-shift. The maximum number, 8, holds for only small phase-shifts, and then progressively falls as $|\phi|$ increases toward $\frac{\pi}{3}$, for which all the synchronizations break. As shown by the map of Fig. 2.14(c), when $|\phi|$ approaches π, we observe that only synchronization of pairs of oscillators arises, in the regions for which the two inputs have close frequencies.

The map obtained for $\phi = 0.12\pi$ is shown in Figure 2.14(b). We can see that for this phase-shift, the synchronization pattern region shapes are already significantly deformed with regards to Fig. 2.1(b), especially when several pairs of core oscillators are involved. Indeed, while interaction phase-shifts do not prevent synchronization between two oscillators, frustrations arise when several oscillators are involved in the synchronization process as seen in other contexts in [173, 174]. This phenomenon is critical in the context of the stimuli-induced core oscillator synchronizations, as at least three oscillators (2 core and 1 input) are involved.

As a conclusion, as the engineering of the pattern recognition architecture using a network of oscillators relies strongly on the ability to synchronize more than two oscillators, the phase relation between synchronized oscillators should be carefully engineered.

The use of delay lines or reactive components in the network can in particular be integrated in the design process to solve this issue by bringing the coupling phase shift back to zero for optimal performance. Technological solutions allowing this can be found in [169, 173, 175, 176].

Effects of oscillator non-linearity

Non-linear behavior, *i.e.* phase-amplitude coupling, is a common property of nano-oscillators. As explained in Section 1.2.4, it manifests itself through a frequency dependence on the oscillation amplitude. This non-linearity has a strong influence on the synchronization efficiency as it amplifies the effects of interactions [177]. This subsection assesses the consequences of non-linearity on the behavior of the pattern recognition architecture.

I model the oscillators of the system using the Landau-Stuart model with couplings as defined by the following set of coupled equations:

$$\begin{cases} \dot{r}_i &= \gamma(1-r_i^2)r_i + \sum_j k_{i,j} r_j \cos(\theta_j - \theta_i + \phi_0) \\ \dot{\theta}_i &= w_i^0 + \nu\gamma(r_i^2 - 1) + \sum_j k_{i,j} \frac{r_j}{r_i} \sin(\theta_j - \theta_i + \phi_0) \end{cases}, \tag{2.4}$$

where r_i is the amplitude of oscillator i, γ is the damping coefficient for radius deviation, ν is the dimensionless nonlinear frequency shift that quantifies the non-linearity, and ϕ_0 is the coupling phase that depends on the physical nature of the coupling mechanism. Equation 2.4 is a conventional way to model nonlinear oscillators, as described in [177]. The phase equation of this model reduces to the Kuramoto equation 2.1 in the absence of nonlinearity ($\nu = 0$). When phase-amplitude coupling is involved, it can be shown (demonstration in Section 3.4.3) that a pair of oscillators interact with an increased effective coupling term: $k'_{ij} = k_{ij}\frac{r_j}{r_i}\sqrt{1+\nu^2}$. The second effect of the nonlinearity implies a contribution to the effective coupling phase-shift, that then verifies: $\phi' = \phi_0 + \arctan(\nu)$

The choice of a too strong coupling is detrimental for the recognition abilities of the network, this effective increase should be conveniently used to allow the use of weaker physical couplings at design time, or allow a larger spacing between core oscillator natural frequencies. As an illustration, we show in Fig. 2.15 the readout map obtained in a case when the coupling between the oscillators is conservative ($\phi_0 = -\pi/2$) and with a strong non-linearity $\nu = 5$, for which all coupling strengths were reduced by a factor $\sqrt{1+\nu^2} \approx 5$. This rescaling allows to come back to effective coupling strengths close to those of Fig. 2.1(b). However, the map appears deformed as anticipated due to the non-zero effective phase-shift ($\phi' \approx -0.06\pi$ for $\nu = 5$).

We conclude that the nonlinearity effects are interestingly beneficial for the design of the introduced pattern recognition architecture. By artificially increasing the coupling strength, it allows the use of even more weakly coupled networks, but mostly it will allow to increase the spacing between core oscillators' natural frequencies, that has strong interest to mitigate

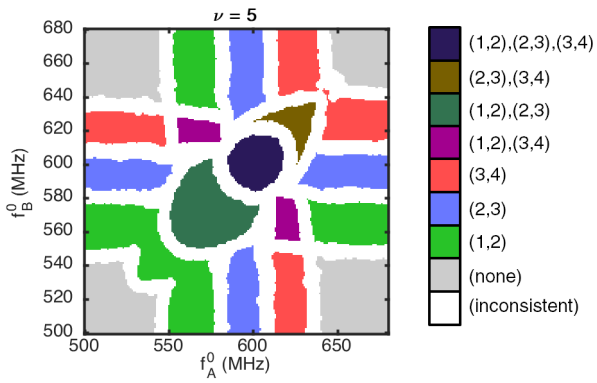


Figure 2.15: Readout map obtained for an oscillator network with conservative coupling ($\phi_0 = -\pi/2$), dimensionless non-linear frequency shift $\nu = 5$ and coupling strengths reduced by a factor $\sqrt{1+\nu^2} \approx 5$ compared to the linear network.

the effects of phase noise and variability.

In case of a conservative coupling, strong non-linearities can also be beneficial to push the effective coupling phase towards zero. In general cases, effects of the non-linear behavior on this phase-shift should be carefully considered as a function of the physical origin of the coupling, and the subsequent ϕ_0 value, aiming for an effective phase-shift as close to 0 as possible.

System scalability

We observed that our reference architecture composed of four core oscillators and two input oscillators already allows discriminating up to eight different classes of stimuli. We now study the evolution of the maximum number of synchronization patterns reached by the proposed architecture as it scales by increasing the number of both core and input oscillators.

During the readout phase, synchronization is evaluated between pairs of core oscillators with consecutive natural frequencies. For N_c core oscillators, each one of the $N_c - 1$ pairwise evaluations can return a positive or a negative answer. The theoretical upper bound on the number of discriminated patterns can then be derived as the number of possible values taken by a binary word of $N_c - 1$ bits, i.e. 2^{N_c-1}.

Nevertheless, depending on the number of input oscillators, different proportions of these synchronization outcomes can be reached. To investigate the actual capacity of the system, we have computed the readout maps of the proposed circuit on GPU for different numbers of core oscillators and different numbers of input oscillators, and have counted the number of different unique available synchronization patterns in each of them. To keep the parameters of the system unchanged, we maintain a constant 20MHz difference between consecutive natural frequencies in the core network. Note that an increase in the number of core oscillators is bounded by the availability of a larger range of natural frequencies. The number of dimensions of the computed maps is equal to the number of inputs. The ideal cases (without noise nor

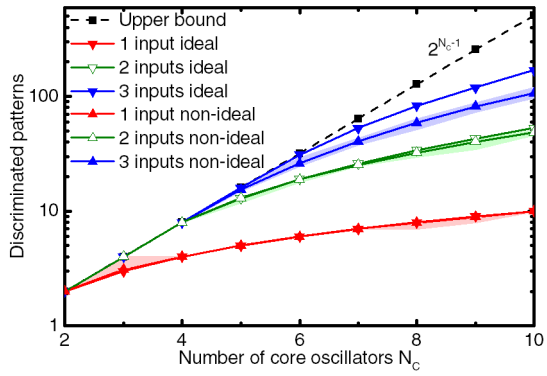


Figure 2.16: Number of different discriminated patterns in the readout map as a function of the number of core oscillators N_c, for $N_i = \{1,2,3\}$ input oscillators, as evaluated in an ideal case and in an average case with high noise and variability. The filled regions represent the span between the best and worst cases encountered during 10 random variability trials. The theoretical upper bound (■) corresponds to 2^{N_c-1}.

variability) in Figure 2.16 show the number of discriminated patterns for the readout maps obtained for different numbers of ideal core and input oscillators, as well as the aforementioned theoretical upper bound.

These results show that the response of the ideal system is very rich as it displays an important number of different synchronization patterns. The number of observed synchronization patterns substantially increases both with the number of core oscillators, and with the number of input oscillators. More specifically, the number of patterns versus number of core oscillators follows the exponential theoretical upper bound curve before its increase starts slowing down. The maximum number of patterns stays equal to the theoretical upper bound longer when more inputs are present. This is due to the fact that a higher number of groups of oscillators can be synchronized independently when more inputs are available, which increases the total number of patterns reachable by the system.

As we have seen that the presence of noise and variability in such oscillator-based computing architectures has an influence on the network's synchronization behavior, it is important to assess the scalability of the system in a non-ideal case. We reproduced simulations including an important level of noise and variability: noise FWHM = 1MHz, ±1MHz variability on natural frequencies and 10% variability on the couplings, as defined in the previous sections. The non-ideal cases in Fig. 2.16 show the average number of discriminated synchronization patterns in the response maps obtained on 10 random trials in these conditions. The filled areas correspond to the span between the best and worst cases encountered during the random trials.

We observe that this level of noise and variability does not or barely affect the number of discriminated patterns in the 1 and 2 input cases, as well as in the 3 inputs case until $N_c = 5$. For higher N_c values, in the three-input case, the number of output synchronization patterns is reduced with regards to the ideal case: in the worst case $N_c = 10$, the number of stable patterns

discriminated drops from 169 in the ideal case, to 107 on average. This loss is mainly due to the noise effects: the tested variability values, although high, have little effect on the number of synchronization patterns. The patterns suppressed by noise tend to be the smallest, therefore less-reliable, patterns.

Nevertheless, we see in Figure 2.16 that the presence of noise and variability does not fundamentally affect the scaling capability of the system, but requires avoiding using unreliable patterns of the response map. We should also remark that, in the presence of noise and variability, the previously mentioned frequency spacing criteria also limit the scalability of the system provided that the accessible natural frequency range for core and input oscillators is limited.

Overall, these results show that the maximum number of synchronization patterns of the proposed system is high and scales well with the number of oscillators, which makes the system attractive for many-class classification and associative memories.

Map resolution and scaling

To keep the the frequency resolution constant, the maps of the scaling study and 10 core oscillator maps were computed using $333 \times 333 \times 10 = 1,108,890$ simulations on GPU.

Effects of geometrical constraints

Uniform all-to-all coupling is a straightforward hypothesis when considering networks of oscillators. Nevertheless, densely packed networks of oscillators also offer the possibility to leverage coupling through proximity effects. In such a situation, the distance between oscillators can affect their coupling strength and/or induce phase shifts in their signals. The geometrical arrangement of devices then appears as a fundamental consideration. It is then interesting, in the nano-device context, to assess the robustness of the studied computing scheme in the case of non-uniform couplings.

Spatially decaying coupling strengths

Proximity couplings usually involve short characteristic interaction distances. Coupling strength then decreases with the physical distance between two oscillators. This applies for example to mechanical couplings through acoustic waves(see [171] and Section 1.3.3), to spin wave coupling of spintronic oscillators [112, 178], to optical couplings [125] or to couplings through dipolar electrical or magnetic fields [111]. See Section 1.3 for more details.

To assess this effect, we simulate our 2-input architecture considering 10 core oscillators arranged along a line, sorted by increasing natural frequency, and with core-core coupling

strengths $k_{i,j}\big|_{(i\neq j)\in\text{core}}$ decreasing exponentially with distance:

$$k_{i,j}\big|_{(i\neq j)\in\text{core}} = k_{cc} \times \exp\left(-\lambda\left(\left|i-j\right|-1\right)\right), \tag{2.5}$$

where λ is the spatial coupling decay factor. For generality, input-core couplings are left unchanged.

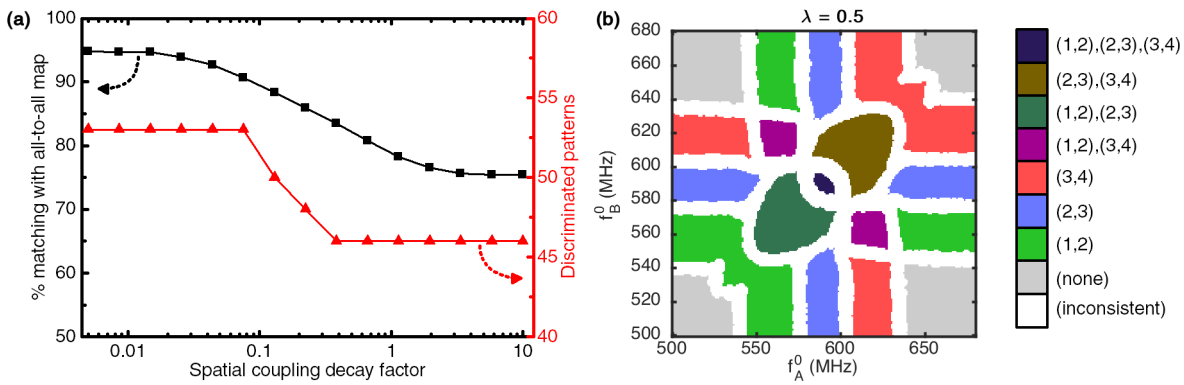


Figure 2.17: (a) Matching with the response map in the uniform all-to-all coupling case (■), and number of discriminated patterns (▲) in the resulting map for $N_c = 10$ core oscillators as a function of the spatial coupling exponential decay factor. The oscillators are assumed to be arranged along a line, spaced by one distance unit, and ordered by increasing natural frequency. (b) Example response map obtained for $N_c = 4$ core oscillators, $N_i = 2$ input oscillators and $\lambda = 0.5$.

Figure 2.17(a) shows the number of discriminated patterns in the response map, as well as its matching to the map with all-to-all uniform couplings ($\lambda = 0$), as functions of the spatial coupling decay factor λ. The graph shows that the system behavior changes minimally up to $\lambda = 0.075$, where it shows 90.6% matching to the all-to-all map with all its 53 patterns still present. The number of discriminated patterns then drops and stabilizes to 46 for $\lambda > 0.4$ and the matching with the all-to-all map drops and stabilizes to 75.4% for $\lambda > 5.8$.

In order to illustrate the effects of such a decay, we also compute an example $N_c = 4$ core oscillator response map under a high decay factor ($\lambda = 0.5$). The corresponding response map Fig.2.17(b) shows that only the 4-oscillator pattern ■ disappears as the coupling decay increases. On the other hand, patterns involving a single input and 2 core oscillators ■,■ and ■ remain unaffected.

Patterns involving at most single pairs of synchronized oscillators remain unaffected because they rely on first-neighbor couplings only. On the other hand, patterns involving at least one group of more than 2 synchronized oscillators are affected because long range interactions between distant neighbors in the group contribute to their stability.

These results show that in a configuration where couplings are decaying in space, or even limited to first neighbors, the system only loses a small portion of its capacity, as most of the synchronization patterns can be stabilized by short-range interactions alone.

Spatially-increasing coupling phase shifts

When coupling in the network involves propagating waves, the distance between two oscillators can also induce non-negligible time delays, and therefore distance-dependent coupling phase shifts. They can also be induced by material non-linearities [179].

To account for distance-dependent coupling phase shifts we simulate the 2-input architecture with 10 core oscillators arranged spatially along a line, ordered by increasing natural frequency, and with a distance-dependent coupling phase shift term between core oscillators $\phi_{i,j}\big|_{(i\neq j)\in\text{core}}$:

$$\phi_{i,j}\big|_{(i\neq j)\in\text{core}} = -\eta \times \left|i-j\right|, \tag{2.6}$$

where η represents the phase shift per unit distance.

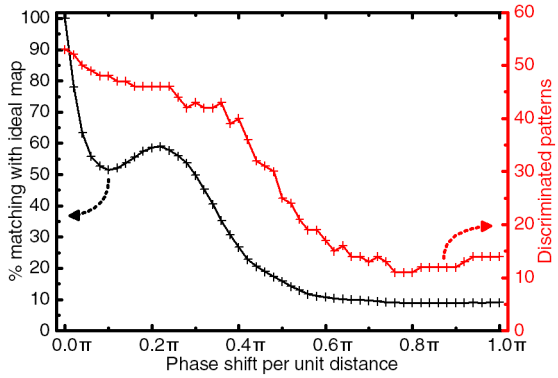


Figure 2.18: Matching with the ideal map (+) and total number of different discriminated patterns (+) in the readout map as a function of the phase shift per distance unit, for $N_c = 10$ core oscillators. Oscillators are assumed to be arranged along a line, spaced by one distance unit, and ordered by increasing natural frequency.

Fig. 2.18 shows the number of discriminated patterns in the response map, as well as its matching to the ideal $\eta = 0$ map, as functions of the phase shift per unit distance η. Similarly to the uniform phase shift case, the response map of the system is heavily altered by distance-related phase shifts. For $\eta = 0.04\pi$, the matching with the ideal map already drops to 63.4%, as the shape of pattern regions is changing, but only 3 patterns are not observed anymore. Generally, while the matching to the ideal $\eta = 0$ map quickly decreases below 60%, the number of stable patterns stays high, over 40, for moderate values of η ($< 0.4\pi$). A locally optimal situation is observed around $\eta = 0.22\pi$, which is related to system symmetries. Patterns involving only 2 core oscillators synchronized with a single input are again the most resilient to these phase shifts, while the apparition of frustrations quickly destabilizes patterns involving multiple synchronized oscillators.

Overall, to keep the maximal capacity of the system, accumulated contributions to the phase-shifts between each pair of oscillators should be brought as close to zero as possible. Considering propagation-related effects, a careful distribution of oscillators in space should be considered to ensure target phase-shift. It is however expected that, in case of decaying

coupling, the influence of distance-dependent phase-shift should be lower.

Advantages and limitations of the architecture

In this section, I showed that a network of nano-oscillators can be used to achieve recognition and classification operations robustly by relying on the rich synchronization dynamics of its oscillators while remaining compatible with their properties. This work was published as [180].

In an ideal situation with noiseless oscillators and in absence of variability issues, this architecture allows the classification of stimuli in a large number of classes. Even if its capacity is diminished as compared to an ideal case, this computing architecture is also fully compatible with noise levels and device variability corresponding to current achievements in nanotechnology. Resilience to noise and device variability is a widespread feature of neural network-inspired architectures [181], and a strong advocate for their use in conjunction with nanotechnologies. In the case of this system, the relaxed nature of synchronization evaluation further helps the system deal with noisy situations.

A challenge for the design of architectures using emerging nano-technologies such as nano-oscillators is the difficulty, with the still limited fabrication capabilities, to integrate high numbers of devices on chip. In this work, I have shown that this architecture can allow for complex classification tasks even with a network of a reduced number of oscillators. It is able to discriminate oscillating stimuli into a number of classes that scales rapidly with the number of core oscillators, even in situations with high phase noise and variability. With this fast increase of the computational complexity with the number of oscillators, relatively complex cognitive tasks can already be achieved with a limited number of oscillators, as illustrated in this study in the case $N_c \leq 10$ oscillators.

Concerning scalability, our study has highlighted network design rules, for which the minimal natural frequency spacing among core oscillators has to be set in agreement with the expected noise and variability amplitudes. As a consequence, the accessible natural frequency range appears as a crucial parameter during the architecture design.

Nanodevice-based oscillators are generally non-linear [177, 182]. I have shown that these non-linearities – *i.e.* phase-amplitude coupling – allow the use of weaker couplings in the core computing network, by increasing the oscillators' synchronization capabilities. They allow to increase the spacing between core oscillators frequencies, but also contribute to the phase-shifts between synchronized pairs of oscillators.

Such phase-shifts, that also arise from the global properties of the coupling, appear to be the most important practical challenge towards achieving oscillator network computation. They are responsible for the appearance of frustrations that restrict synchronization of more than two oscillators, and reduce the global synchronization capabilities of the network. Therefore, this point should receive special attention for the choice of ideal technology as well as ideal coupling type and geometry in the design of the network. Coupling phase shifts can also be adjusted at design time: for instance, in the case of high frequency oscillators, proper trans-

mission line design can introduce phase delays between oscillators so that phase shifts are ideally brought back to zero [169, 173, 175, 176].

The computing scheme described through this work was also shown to be compatible with different oscillator coupling strategies. While all-to-all coupling guarantees the highest classification capacity, coupling through proximity effects, prone to appear in nano-device networks, can also be leveraged with high computing capacity.

Improvements and alternative architectures

Improving the architecture

A proposal to achieve advanced classification could also be to rely on the juxtaposition of several small core networks, trained independently to discriminate complementary subsets of patterns. From a nanotechnology point of view, this also avoids the complex fabrication of large networks of nano-objects and would facilitate the training of the networks.

In the case of magnetic or certain mechanical oscillators, the oscillators also need to be sufficiently spaced apart to prevent interactions through dipolar coupling or local vibrations, which limits the integration density. Embracing this interaction to combine local and global couplings between oscillators could allow the implementation of architectures with rich behaviors and higher density.

An alternative “EEG-like” readout method

This subsection describes a proposal for an alternative readout approach inspired by the EEG signals in the brain.

As introduced in Section 1.1.2.2 and later in Section 1.2.2.3, the mean field activity of an ensemble of neurons in the brain can be obtained as an EEG signal. An analogous quantity can be defined in oscillator networks as the mean of the signals of all the oscillators:

$$\frac{1}{N}\sum_i \sin(\theta_i)\,. \tag{2.7}$$

This mean field signal is also directly proportional to the real part of the Kuramoto order equation (equation 1.4) :

$$\mathrm{Re}\left(re^{i\psi}\right) = \frac{1}{N}\sum_j \sin\left(\theta_j\right). \tag{2.8}$$

When the order parameter r varies slowly compared to the phases $\sin(\theta_i)$, it can be extracted by taking the envelope:

$$S = \text{envelope}\left(\frac{1}{N}\sum_j \sin(\theta_j)\right), \tag{2.9}$$

which is called the "EEG-like" signal in the rest of the manuscript, as it resembles its biological counterpart. Envelope detection itself can be done with simple resistor-capacitor circuits with a suitable decay time constant τ_{EEG}. The "EEG-like" signal is an approximation of the order parameter that is convenient to probe in typical coupled nano-oscillator networks, and provides a measure of the global synchronization state of the network. It could therefore be thresholded in order to perform binary (*yes* or *no*) classification using the same type of architecture as studied throughout this chapter.

Note that S is similar to the degree of matching measures introduced in Section 1.4.4 but is used here in a classification context that does not involve distance measurements.

In order to evaluate this EEG-like readout possibility for the oscillator-based classification architecture, I simulate its response map in terms of the mean value of the EEG-like signal S obtained using different envelope detection time constants τ_{EEG} (Figure 2.19(b,c,d)), and compare it to the corresponding pairwise synchronization map shown in Figure 2.19(a).

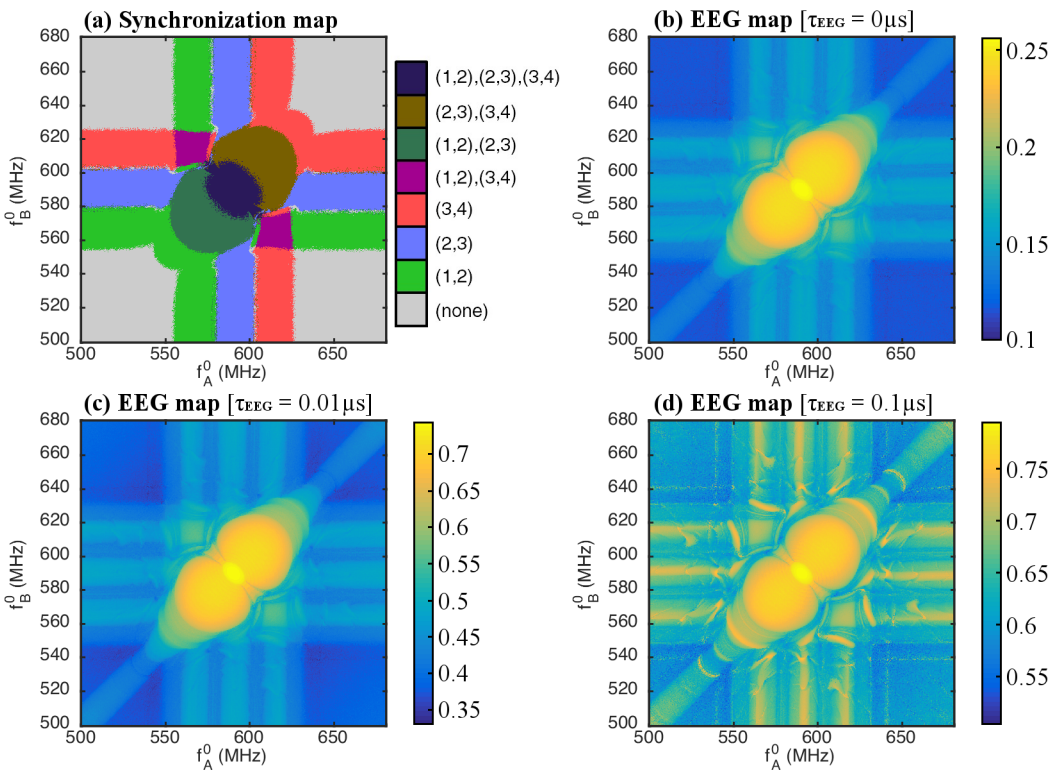


Figure 2.19: **(a)** Synchronization response map of the oscillator-based classification architecture. **(b, c, d)** EEG-like signal maps of the same system with $\tau_{\text{EEG}} = \{0, 0.01, 0.1\}$μs envelope detection time constants respectively.

Figure 2.19(b) shows the EEG-like response map for $\tau_{\text{EEG}} = 0$, which is equivalent to computing the average of the rectified signal $\max(S, 0)$ after waiting for system stabilization. A comparison with the synchronization response map Figure 2.19(a) suggests that the EEG-like signal is linked to the number of synchronized oscillators in the network. It varies between 0.1 when no synchronization occurs and 0.26 when all the oscillators are synchronized.

Figure 2.19(c) shows the EEG response map with an envelope detection time constant $\tau_{\text{EEG}} = 0.01\mu\text{s}$. The map is similar to the case $\tau_{\text{EEG}} = 0\mu\text{s}$ but the contrast has improved: the EEG signal now varies between 0.34 and 0.74. This higher contrast can make the physical readout easier.

Figure 2.19(d) corresponds to the EEG-like response map for $\tau_{\text{EEG}} = 0.1\mu\text{s}$. It shows abrupt transitions of the EEG-like value, as well as a multitude of complex secondary patterns linked to higher-order dynamics which make this response more complex to understand, predict and use in real classification contexts.

These results show that the proposed EEG-like readout method could be used in a binary classification architecture if the envelope detection time constant is chosen properly. Learning algorithms for this alternative architecture are studied in Section 4.5.

Summary of the chapter

In this chapter, I have proposed an oscillator-based pattern classification architecture inspired by the seminal work of Vassilieva *et al.* [160] that consists of a network of coupled "core" oscillators, themselves coupled to a set of "input" oscillators. The pattern to be classified is applied as the set of natural frequencies of the input oscillators, which in turn perturb the core network, and induce synchronizations among core oscillators. The resulting list of synchronized pairs of core oscillators is read using a simple and robust counter-based synchronization scheme. This list corresponds to the class the system has attributed to the presented pattern. The response of the classifier can be tuned to properly solve a given classification problem by adjusting the natural frequencies of the core oscillators.

I have also investigated the compatibility of this architecture with the constraints linked to nano-technologies. In this context, I have shown that the architecture is robust to various phenomena including noise, oscillator variability, coupling variability and oscillator non-linearities. However, this oscillator-based classifier appears to be sensitive to phase shifts which therefore need to be compensated at design time. I have also shown that the maximum number of classes the system can discriminate scales quickly with the number of oscillators.

In the last section, I have proposed a modified version of this architecture where, instead of performing synchronization detection for readout, the envelope of the mean field of the oscillators is measured. This readout method is inspired by the EEG signals observed in the brain, linked to the Kuramoto order parameter, and provides a promising means of binary classification.

Chapter 3

Case of the spintronic vortex nano-oscillator

> Magnetism, as you recall from physics class, is a powerful force that causes certain items to be attracted to refrigerators.
>
> Dave BARRY

"THE THIRD CHAPTER *of this thesis describes how an oscillator-based classifier can be implemented with particular devices: vortex spin transfer nano-oscillators.*"

THIS CHAPTER investigates how the oscillator-based classifier architecture presented in Chapter 2 can be implemented using vortex spin transfer nano-oscillators, an emerging nano-oscillator technology. The chapter covers:

1. A description and SPICE modeling of the vortex spin transfer nano-oscillator
2. The study of an electrical coupling scheme adapted to resistive nano-oscillators that implements the oscillator-based classifier presented in Chapter 2
3. SPICE simulations investigating how well the resulting system matches the expected generic behavior assumed in Chapter 2.

Introduction

As explained in the introductory Section 1.3.5, spin transfer nano-oscillators have multiple advantages including compact size, compatibility with the CMOS fabrication process, wide-range tunable natural frequencies using currents or magnetic fields, and demonstrated synchronization capabilities using electrical or dipolar coupling schemes. A specific type of spin-transfer nano-oscillator involves the rotation of a stable vortex magnetization configuration instead of a full magnetic domain. This type of device, called the vortex spin-transfer nano-oscillator, presents multiple attractive features.

Features of the vortex spin-transfer nano-oscillator

The main highlights of vortex spin transfer nano-oscillators are:

- high frequencies typically of the order of 500 MHz and up to 2 GHz [183],
- high spectral purity, typically FWHM ≈ 100s of kHz at GHz operating frequencies. This spectral purity exceeds that of other spin-transfer nano-oscillators, and represents the main advantage of vortex oscillators [111, 117, 184],
- and wide-range current-based natural frequency tuning, over 100s of MHz [117, 163, 185].
- Experimentally demonstrated electrical and dipolar coupling and synchronization capabilities [111, 169, 186].

Due to their frequencies in the microwave range, low bandwidth and intrinsic broad range current-to-frequency conversion, vortex spin-transfer nano-oscillators are the focus of studies investigating their usage in the telecommunication domain [185, 187, 188].

As part of the MEMOS project, we are in close collaboration with Julie Grollier's team at the CNRS/Thales joint laboratory in Palaiseau. In order to harness the potential of this collaboration, we investigate the implementation possibilities of the oscillator-based architectures introduced in Chapter 2 with the vortex spin-transfer nano-oscillators that our collaborators

are studying experimentally.

This Chapter defines a realistic model of the device, and validates it against the experimental results of our collaborators. A suitable coupling circuit implementing the oscillator-based classifier introduced in Chapter 2 is also defined. Connections to existing oscillator models providing different levels of realism, simulation speed and analytical prediction capabilities are also derived, validated and compared. Finally, the full oscillator-based classifier is simulated in order to investigate the compatibility of the vortex spin transfer nano-oscillator with the classification architecture.

Description of the device

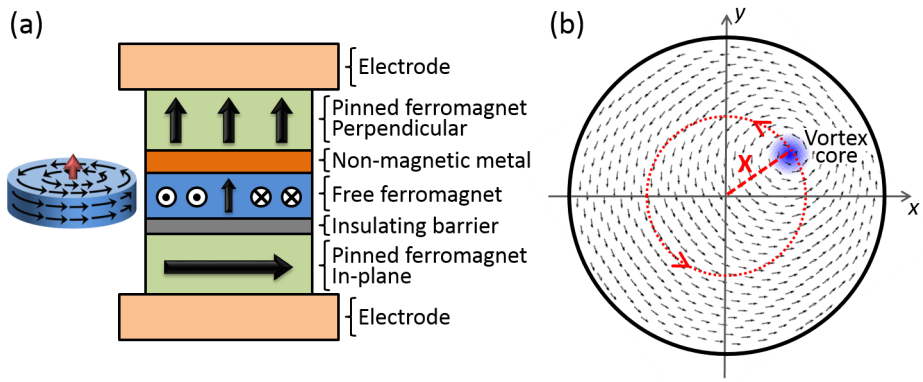


Figure 3.1: **(a)** Sample description. Arrows illustrate the magnetization distribution in ferromagnets. **(b)** Illustration of the vortex magnetization dynamics. Arrows and color scale respectively describe the in-plane and out-of-plane magnetization distribution.

The device under consideration is a 100 nm wide nanopillar hybrid spin transfer vortex based oscillator, as introduced in [189], and presented in Figure 3.1(a). It is composed of a layered stack, including a metallic spin-valve structure with a perpendicularly magnetized ferromagnetic layer, a non-magnetic spacer and a central ferromagnetic layer. The latter is also part of a magnetic tunnel junction, composed of an insulating barrier and an in-plane magnetized ferromagnetic layer. The magnetizations of both perpendicular and in-plane magnetized ferromagnets are considered pinned (*i.e.* constant). The dimensions of the pillar and particularly the aspect ratio of the central ferromagnet are chosen so that its remanent magnetic state is a vortex [190]. Such magnetic state corresponds to the magnetization curling in the layer's plane and popping out-of-plane in the center region called the vortex core (see Figure 3.1). The studied mode of the vortex, called "gyrotropic mode", corresponds to a circular displacement of the vortex core around the center of the magnetic dot [119, 190] as illustrated in Figure 3.1(b).

As current flows through the device, the two pinned layers act as spin-current sources generating a spin-transfer effect on the vortex. As demonstrated in previous works, a source of perpendicular spin polarization is mandatory to start and sustain the vortex precession [191, 192].

On the other hand, the MTJ with in-plane magnetized layer is needed to convert the magnetization oscillations into resistance oscillations [189], and plays a major role in the synchronization process by providing an in-plane component to the current spin polarization [193]. Under sufficient current bias, the device then demonstrates permanent resistance oscillations and a resulting AC voltage appears across the device.

Vortex spin transfer nano-oscillators therefore behave as oscillating resistors, with a frequency increasing with the applied DC current. This allows coupling multiple oscillators electrically.

Defining a model of the device

We propose an implementation of a physical model describing the self-sustained vortex oscillations in a pillar STNO for use in integrated circuit design of hybrid MTJ/CMOS systems. To correctly describe the oscillator behavior at room temperature, our model implementation includes non-linearities [194], as well as phase and amplitude noise effects under thermal fluctuations [66, 183]. These factors can be of major importance to accurately describe the synchronization phenomenon.

We present a VerilogA implementation of the model under the Cadence platform and confront the obtained simulations to previously obtained experimental results. Finally, we demonstrate the compatibility of our model with a CMOS design kit for integrated circuit design by simulating an integrated current source for biasing and synchronization of the STNO.

The vortex spin-transfer oscillator model

Because the vortex gyrotropic mode is isolated from other higher frequency modes, the self-sustained oscillations under spin-transfer effect can be properly described by a collective-variable equation, called the Thiele equation [119, 190], where the global dynamics is described through the evolution of the position of the magnetic vortex core. This equation was recently extended to include the spin-transfer terms [191, 192, 194]. The device is then modeled according to the following equations, where $\boldsymbol{X} = \rho e^{i\chi}$ is the vortex core complex coordinate (see Figure 3.1(b)):

$$iG\dot{\boldsymbol{X}} = -D(\rho)\dot{\boldsymbol{X}} - k(\rho)\boldsymbol{X} + i\kappa_{\perp} I\boldsymbol{X} + (\kappa_{\mathrm{SL}} + i\kappa_{\mathrm{FL}})\, I + \boldsymbol{F}_{noise}(t)\,. \tag{3.1}$$

The device conductance oscillates in time according to:

$$R = R^0 + \tilde{R} = R^0 - R_{amp}\rho \mathrm{Im}(\boldsymbol{X})\,, \quad \text{where} \quad \begin{cases} R^0 = \frac{2R_{\min}}{1+\frac{1}{1+TMR}} \\ R_{amp} = \frac{R^{0^2}\xi}{2R_{\min}}\left(1 - \frac{1}{1+TMR}\right) \end{cases} . \tag{3.2}$$

Equation 3.1 is the modified Thiele equation describing the dynamics of the vortex core, accounting for conservative and dissipative terms. The left hand side gyrotropic term is equal to

Symbol	Expression	Description
r	90 nm	Pillar radius
t	10 nm	Free layer thickness
$\mu_0 M_S$	1.0 T	Free layer saturation magnetization
α	0.01	Free layer Gilbert damping
L_e	6nm	Free layer exchange length
$p_\perp$	60%	Current spin polarization along perpendicular z-axis
$p_\parallel$	100%	Current spin polarization along in-plane x-axis

Table 3.1: Geometrical and magnetic parameters for the free ferromagnetic layer and spin polarized current.

the sum of the effective forces acting on the vortex core: an effective damping force, a spring-like confinement force, and effective spin-transfer forces induced respectively by the perpendicular and the in-plane spin polarizations of the current. Equation 3.2 relates the value of the resistance to the position of the vortex core, and depends on the TMR. All the terms appearing in these equations are derived from geometrical and magnetic parameters. Their expressions are gathered in table 3.2. Additionally, on the right-hand side, a phenomenological stochastic force $F_{noise}(t)$ has been added, describing the action of thermal fluctuations according to fluctuation-dissipation theorem, and derived in the framework of the classical nonlinear oscillators theory [66, 183, 195]. Finally, the amplitude of the oscillations is limited by the existence of a critical velocity v_c for the vortex core that triggers a polarity switching and subsequently stops the self-sustained oscillations [192].

Field-like torque

Note that because at the time of this work the effects of the field-like-torque κ_{FL} was considered negligible in the description of such oscillators, the preliminary version of our model considered until Section 3.2.5 uses a field-like-torque factor $\lambda = 0$ (see Table 3.2).

Model implementation

The described physical model was implemented in VerilogA analog hardware description language. This allows the model to be used as a component in circuits simulated by a broad range of circuit simulation tools, and in particular the industry-standard Cadence Spectre simulator we are using. Simulations were conducted for a $2r = 180$nm diameter device, with a $t = 10$nm thick free $NiFe$ layer whose magnetic parameters are gathered in table 3.1. The minimum resistance was chosen to be $R_{min} = 130\Omega$ with a TMR ratio $\mathrm{TMR}_{\%} = 70\%$ for the tunnel junction, in agreement with previously fabricated devices. A $p_\perp = 60\%$ perpendicular polarization (from

Symbol	Expression	Description	Units
G	$2\pi t M_S/\gamma$	Gyrovector amplitude	$N.s/m$
η	$\frac{1}{2}\ln\left(\frac{r}{4L_e}\right)-\frac{1}{8}$	1st order damping	unitless
η'	$1/12$	2nd order damping	unitless
$D(\rho)$	$\alpha G\left(\eta+\eta'\rho^2\right)$	Non-linear damping	$N.s/m$
k_{ms}	$\frac{10}{9}\mu_0 {M_S}^2 t^2/r$	1st order magneto-static confinement	N/m
k'_{ms}	$k_{ms}/4$	2nd order magneto-static confinement	N/m
k_{Oe}	$0.85\mu_0 M_S t/(\pi r)$	1st order Oersted field confinement	$N.m/A$
k'_{Oe}	$-k_{Oe}/2$	2nd order Oersted field confinement	$N.m/A$
$k(\rho)$	$(k_{ms}+k_{Oe}I)$	Non-linear confinement	N/m
	$+\left(k'_{ms}+k'_{Oe}I\right)\rho^2$	coefficient	
a_J	$g\mu_B/(2teM_S)$	Spin transfer efficiency	$m^2/A/s$
ξ	$2/3$	Average magnetization to vortex displacement ratio	unitless
κ^0_{SL}	$a_J G/\left(2\pi r^2\right)$	Slonczewski torque amplitude	$N/m/A$
$\kappa_\perp$	$p_\perp \kappa^0_{\mathrm{SL}}$	Orthogonal spin transfer	$N/m/A$
κ_{SL}	$2L_e\ln(2)p_\parallel\kappa^0_{\mathrm{SL}}/r$	In-plane Slonczewski spin transfer	$N/m/A$
κ_{FL}	$\lambda\times\kappa^0_{\mathrm{SL}}\xi p_\parallel$	In-plane Field-like torque	$N/m/A$
v_c	$\frac{1}{3}L_e\gamma\mu_0 M_S$	Critical vortex velocity	m/s

Table 3.2: Parameters for the Thiele equation modeling the vortex gyrotropic motion. γ is the electron gyromagnetic ratio, μ_B is the Bohr magneton, μ_0 is the vacuum permeability, $g=2$ is the Landé factor and e is the electron charge.

the the spin-valve) and an ideal $p_{\parallel} = 100\%$ in-plane polarization (from the MTJ) were assumed. These parameters correspond to a critical 220μA current to start the self-sustained oscillations.

The model was tested under constant current bias for transient simulations, in both cases of noise-free and noisy oscillator. The vortex core position was initiated in a random position in the vicinity of the ferromagnet center, allowing the observation of a transitory regime where the oscillations amplitude increases up to its stable value, starting the steady regime of oscillations.

A major criteria for the validation of our model was the stability of the noise-free auto-oscillations frequency and amplitude in the steady-state regime and their independence to the simulation time step. Due to different time-scales for the evolution of the phase and the amplitude of oscillations, we found that the compact model is largely improved by implementing equation (3.1) under polar coordinates instead of Cartesian. We then evaluated that to ensure a steady-state frequency stability, a reasonable criteria was ensuring that the time step for the simulation verifies $\Delta t < T(\rho)/50$ where $T(\rho) = 2\pi/\omega$ is the instantaneous oscillation period of the vortex. We enforce this minimum time step thanks to a *bound_step* VerilogA call within the compact model.

The thermal fluctuation term was implemented through a random Gaussian draw at each integration step of the differential equation. Compatibility of this model with fluctuation-dissipation theory was then verified by evaluating the average fluctuating energy at equilibrium. At zero bias, this energy $\langle \epsilon \rangle = \langle \frac{1}{2} k(\rho) r^2 \rho^2 \rangle$ was verified to be equal to the thermal energy $k_B T$.

10μs simulations of a single oscillator on a 3.20GHz CPU were obtained in respectively 9.43s and 9.81s for the noise-free and room-temperature noisy oscillator. In Figures 3.2(a) and (b), we present the temporal evolution of the device voltage under a constant 260μA current bias in both cases. The observed auto-oscillation frequency is 617.8MHz, in good agreement with the expected frequency for the chosen device dimensions [190]. As it appears on the corresponding spectrum on Figure 3.2(c), fluctuations in the noisy oscillator translate into phase noise that broadens the spectrum peak, accounting for the non-zero peak linewidth observed experimentally [189]. A 14.6MHz linewidth is measured, independent on the time step, in good agreement with the 14.0MHz theoretically predicted linewidth [66].

CMOS-compatible natural frequency tuning

We validated the compatibility of our model with a commercial 28-nm CMOS design kit by designing a MOSFET-based supply for the STNO (see Figure 3.3). The circuit is a voltage controlled current source. A two-transistor architecture is implemented to prevent voltage oscillations across the STNO from disturbing the current bias. With this circuit, the current is kept constant despite the resistance oscillations of the device that can shift the operating point of the transistors. The frequency of the auto-oscillator is then controlled by the gate voltage of the $T1$ transistor, as summarized in Figure 3.3.

The circuit is designed so that a zero command voltage stays below the threshold current

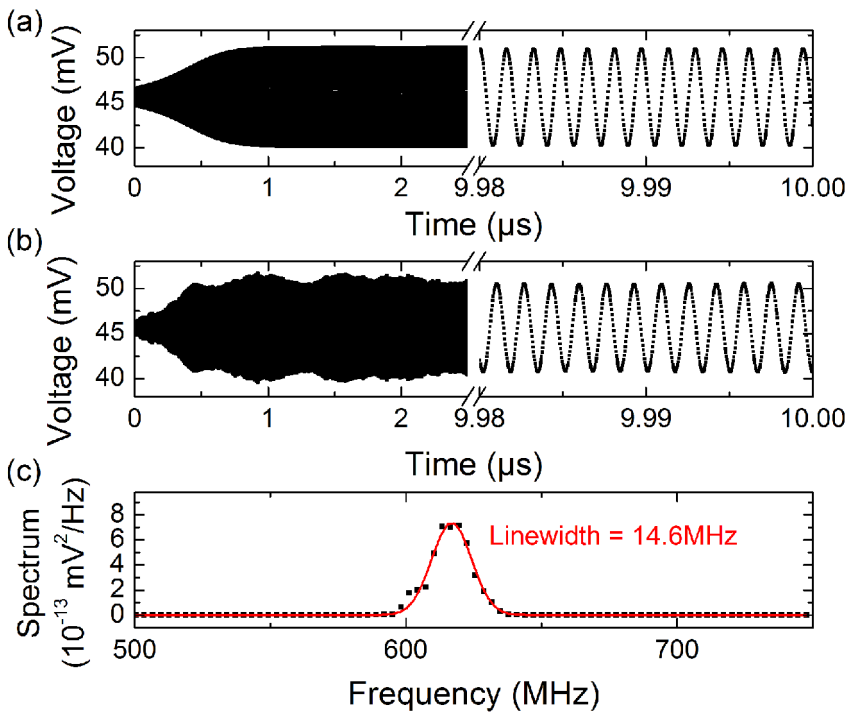


Figure 3.2: **(a)** Simulated voltage versus time for a noise-free oscillator under a constant 260μA bias. **(b)** Simulated voltage versus time and **(c)** associated spectrum for a noisy oscillator under a constant 260μA bias. Full Width at Half-Maximum is determined by fitting the spectrum peak with a Gaussian function.

bias. Self-sustained oscillations appear when a $V_{cde} > 10$mV voltage is applied, and their amplitude keeps increasing as V_{cd} is swept up to 180mV above which the critical vortex speed is reached and oscillations are terminated. Because both current bias and resistance oscillations amplitude are different for each frequency value, it is important to note that the AC voltage amplitude is also varying as the command voltage is swept. As it appears on Figure 3.3, the voltage amplitude varies between 0 and 15mV.

Synchronization to an oscillating signal

Spin transfer nano-oscillators also show the ability to respond to an AC-current excitation by synchronizing to the input signal for sufficient amplitude and if its frequency is close enough to the auto-oscillation frequency [196, 197]. To test the ability of the implemented model to implement this capability, we simulated the response of the STNO when a harmonic AC-voltage is added to the command voltage as shown on Figure 3.4(a). The simulation was run for $V_{cde} =$ 120mV, corresponding to a 261μA current bias and a frequency of 619MHz, to which an AC-signal at a frequency 609MHz was added. The frequency response presented in Figure 3.4(b) shows that a minimum 100mV AC-input amplitude is necessary for the oscillator to shift its frequency and synchronize to the AC signal, which corresponds to the AC-current amplitude

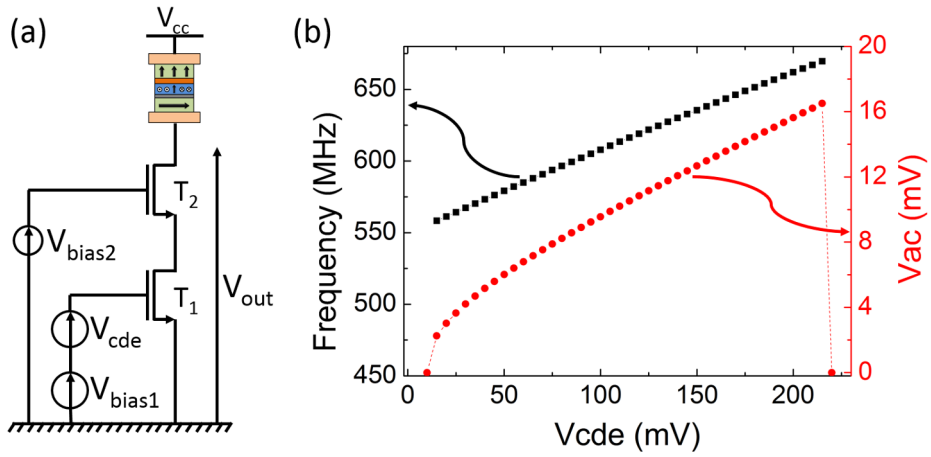


Figure 3.3: **(a)** MOSFET-based control circuit for the spin transfer oscillator, simulated using a commercial 28nm-node design kit. **(b)** Frequency and peak-to-peak amplitude of the output voltage V_{out} versus command voltage.

$i_{AC} = 27.2\mu$A.

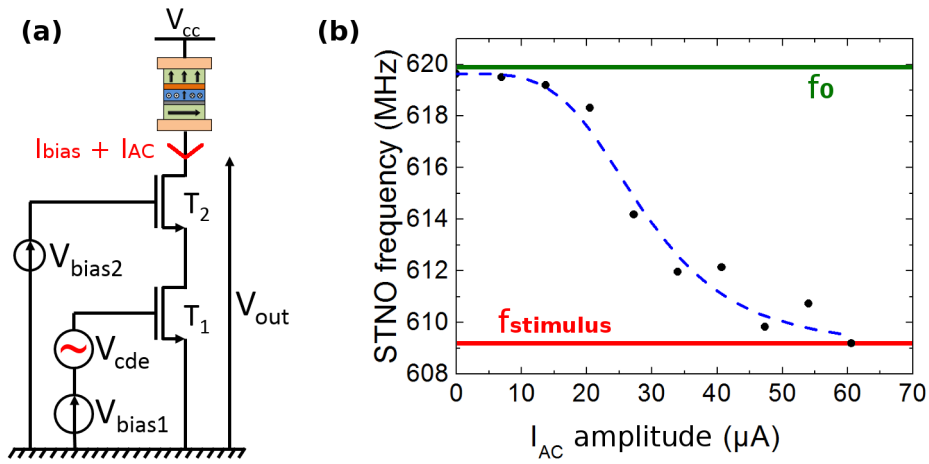


Figure 3.4: **(a)** MOSFET-based control circuit simulated using a commercial 28nm-node design kit. V_{cde} is oscillating at $f_{\text{stimulus}} = 609$MHz to apply an alternating current component I_{AC}, and V_{bias1} sets the natural frequency at about $f_0 = 620$MHz. **(b)** STNO frequency as a function of the amplitude of I_{AC}.

These results show that the model is capable of predicting synchronization to an oscillating current, however the synchronization range initially didn't fully match experimental results [169]. Section 3.2.5 explains the underlying causes of this mismatch and the corrections we applied.

Model parameter	Analytical operating point	CNRS-Thales devices
r	100 nm	100 nm
η	0.588558	1.1372
η'	0.0833333	0.5265
k_{ms}	8.84194×10^{-4} N/m	3.6201×10^{-4} N/m
k'_{ms}	22.1049×10^{-5} N/m	1.086×10^{-5} N/m
k_{Oe}	0.0270563 N.m/A	0.8768 N.m/A
k'_{Oe}	−0.0135282 N.m/A	0 N.m/A
λ	0	5

Table 3.3: Model parameters for the vortex nano-oscillator around the analytical operating point, and around the operating point of the devices studied in collaboration with CNRS-Thales.

Limitations of the model and subsequent improvements

Since our publication of this model in 2015 [70], new experimental results, including results from our collaborators in Thales, have shed light on some of its limitations, after which we have improved the model.

The field-like torque term κ_{FL} was shown to be dominant compared to κ_{SL} by Romera *et al.* [198] to explain the strong synchronization capabilities of these oscillators. We therefore adjusted λ to match experimental values. The underestimated non-linearity of the oscillator also caused noise over-estimation and is corrected by the adjustments.

Adjusting the model to specific devices

The CMOS-compatible Verilog-A model of the vortex-based spin transfer oscillator that we presented in Section 3.2.1 represents the theoretical behavior of a generic vortex nano-oscillator around its typical operation regime. It uses analytically predicted parameters from fundamental physics, and assumes typical material properties. We demonstrated that it successfully accounts for the non-linear behavior of typical vortex nano-oscillators, their current-based natural frequency tuning, as well as their ability to synchronize to external signals.

As real devices have different properties and non-idealities, their operating point is not the same as in the ideal case. Instead, parameters can be chosen to match experimental measurements or micromagnetic simulations around the operating point. As shown by Grimaldi et al [183], this can imply adjustments of multiple orders of magnitude compared to values deduced from the analytical analysis.

In the framework of our cooperation with CNRS-Thales, I used parameters that describe the general behavior of the vortex nano-oscillators studied experimentally by the team of Julie Grollier as given in Table 3.3. Note that the field-like torque is now taken into account with $\lambda = 5$ according to the observations of Lebrun *et al.* [169].

Electrically coupling nano-oscillators in a circuit

Physically coupling STNOs all-to-all can be achieved through electrical coupling which consists in summing the signals of all oscillators and applying the resulting sum signal as an oscillating current to all oscillators. As a result, each oscillator is coupled to each other. Our collaborators studied different circuit implementations of this approach [110]. Julie Grollier's team also implemented one of those coupling schemes [198], proving its operation experimentally. This section focuses on a theoretical description of this approach.

Coupling oscillators in series

Figure 3.5(a) shows the coupling circuit using two oscillators connected in series and capacitively coupled. Two different constant current biases I_1^0 and I_2^0 are applied to oscillators 1 and 2 respectively, in order to define their natural frequencies. As the resistances $R_1(t)$ and $R_2(t)$ of the two oscillators fluctuate, their combined effect induces current oscillations I_{rf} in the whole branch. This current influences the oscillators by accelerating them or slowing them down, which corresponds to non-local, all-to-all coupling. The capacity C must be high enough ($C > 200\mathrm{pF}$) to avoid second-order oscillations, and the load resistor $R_L = 50\,\Omega$ corresponds to the internal resistance of high frequency measurement devices such as spectrum analyzers.

In order to characterize the efficiency of this coupling, I have carried out a noiseless SPICE simulation of this circuit including the Thiele model of the oscillators, at constant oscillator 2 natural frequency defined by $I_2^0 = 706\,\mu\mathrm{A}$ and swept the natural frequency of oscillator 1 by varying I_1^0 from $680\,\mu\mathrm{A}$ to $730\,\mu\mathrm{A}$.

Phase compensation

As explained in Section 2.4.4, the synchronization of oscillators is sensitive to phase shifts. These simulations include a compensation of the phase shift to maximize the synchronization range. Phase compensation has been demonstrated experimentally by our collaborators in the case of vortex spin transfer nano-oscillators using delay lines [169].

Figure 3.5(b) shows the mean frequencies of the two oscillators as functions of I_1^0. Outside of the synchronization range, the mean frequency of Osc. 1 varies quasi-linearly with I_1^0, and shows a natural frequency tuning behavior consistent with experiments [169]. The synchronization range of about 5 MHz also matches its experimentally observed order of magnitude [169].

This validates the behavior of the oscillator compact model as well as the effectiveness of this coupling circuit which allows choosing the natural frequencies of the oscillators by applying bias currents.

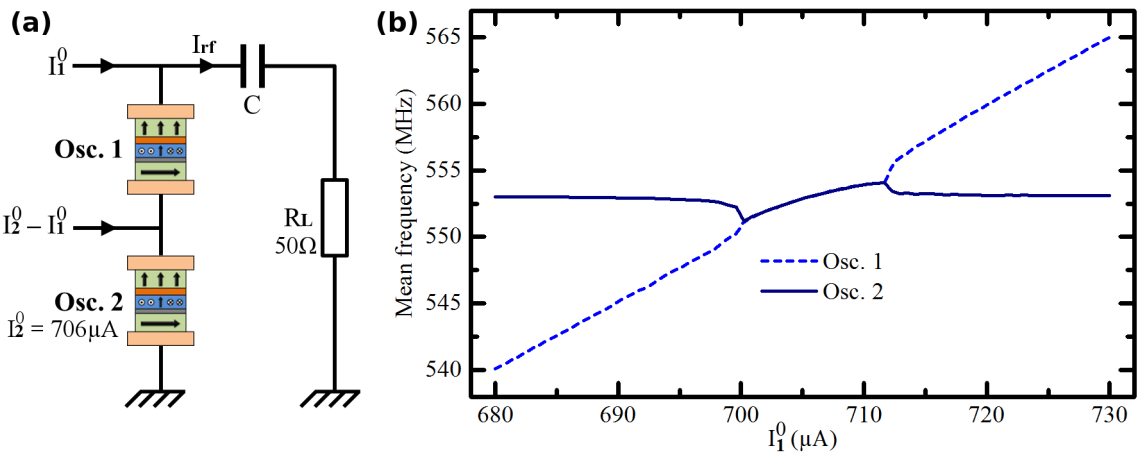


Figure 3.5: SPICE simulation of two coupled vortex nano-oscillators in a circuit. **(a)** Circuit diagram. **(b)** Mean frequency of the two oscillators, as functions of the current bias of Osc. 1 I_1^0. Frequency locking synchronization is achieved for $700\,\mu\text{A} < I_1^0 < 712\,\mu\text{A}$.

Analytical interpretation of the circuit

This approach can be extended to a higher number of oscillators as shown in Figure 3.6. A rigorous analytical analysis provides further predictive power and insight into this scheme.

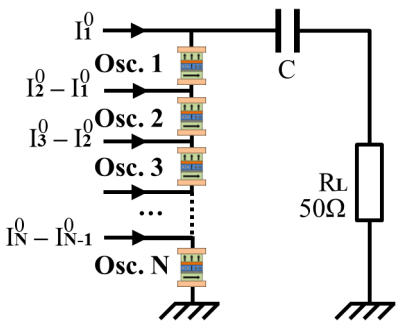


Figure 3.6: Electrical coupling of N vortex spin torque nano-oscillators.

Connecting the current and voltage of capacitor C yields:

$$I_{\text{rf}} = C\frac{d}{dt}\left(\sum_j (I_j^0 - I_{\text{rf}})R_j - I_{\text{rf}}R_L\right). \tag{3.3}$$

As C is large, the capacitor voltage in its stable state can be assumed constant and equal to $\sum_l I_l^0 R_l^0$:

$$\sum_j (I_j^0 - I_{\text{rf}})(R_j^0 + \tilde{R}_j) - I_{\text{rf}}R_L = \sum_j I_j^0 R_j^0. \tag{3.4}$$

Rearranging the terms and using $R_j = R_j^0 + \tilde{R}_j$ provides the expression of the oscillating current

I_{rf}:

$$I_{\mathrm{rf}} = \frac{\sum_l I_l^0 \tilde{R}_l}{R_L + \sum_l R_l^0 + \sum_l \tilde{R}_l}. \tag{3.5}$$

The sum of resistance deviations $\sum_l \tilde{R}_l$ is small compared to the sum of the constant resistances $R_L + \sum_j R_l^0$ not only because of the lower amplitude of its terms, but also because of their mutual compensations. The denominator term can therefore be approximated:

$$I_{\mathrm{rf}} \approx \frac{\sum_l I_l^0 \tilde{R}_l}{R_L + \sum_j R_l^0}. \tag{3.6}$$

The total current through oscillator j is then:

$$I_j^{\mathrm{tot}} = I_j^0 - I_{\mathrm{rf}} = I_j^0 - \sum_m \frac{I_m^0}{R_L + \sum_l R_l^0} \tilde{R}_m, \tag{3.7}$$

and by using the expression of the STNO resistance (equation (3.2)):

$$I_j^{\mathrm{tot}} = I_j^0 + \sum_m \frac{I_m^0}{R_L + \sum_l R_l^0} R_m^{\mathrm{amp}} \rho_m \sin(\theta_m), \tag{3.8}$$

where θ_m is the phase of the oscillator m.

Therefore, the total current through oscillator j is:

$$I_j^{\mathrm{tot}} = I_j^0 + \sum_m A_m \rho_m \sin(\theta_m), \qquad \text{where} \qquad A_m = \frac{I_m^0}{R_L + \sum_l R_l^0} R_m^{\mathrm{amp}}. \tag{3.9}$$

This result confirms that the current through an oscillator j of the circuit is composed of its current bias I_j^0 and of the sum of the signals of all the oscillators in the network. Note that the weight of a single oscillator m in this sum is proportional to its amplitude ρ_m and to its constant current bias I_m^0.

Mean field signal The voltage measured at the terminals of R_L is :

$$U_L = -R_L \sum_m A_m \rho_m \sin(\theta_m). \tag{3.10}$$

This signal is linked to the mean field generated by the combined activity of all the oscillators and acting on the whole network $\sum_i \sin(\theta_i)$. EEG-like architectures such as described in Section 2.5.2 could be readily implemented simply by taking the envelope of this signal with an envelope detection circuit.

This circuit is therefore a valid and realistic approach for STNO coupling, and is used as the reference coupling circuit throughout this chapter. Furthermore, it can also be used for other

two-terminal resistive oscillator devices.

Note : Mean frequency readout

The average frequencies of the oscillators can be measured using a spectrum analyzer in place of R_L as done by our collaborators [169], which allows synchronization detection. However, using a spectrum analyzer is acceptable on a tabletop experiment but not for integrated circuit realization.
This detection can instead be achieved by amplifying the voltage at the terminals of each measured oscillator j:

$$U_j = (R_j^0 + \tilde{R}_j) \times \left(I_j^0 - I_{\mathrm{rf}}\right).$$

The DC contribution $R_j^0 I_j^0$ can be filtered out with a series capacitor, which leaves the oscillating part of the voltage $\tilde{U}_j$. Moreover, simulations show that in this circuit, the amplitude of I_{rf} is low compared to I_j^0, which makes $\tilde{R}_j I_j^0$ the dominant contribution to $\tilde{U}_j$ by an order of magnitude:

$$\tilde{U}_j \approx \tilde{R}_j I_j^0 = R_j^{\mathrm{amp}} \rho_j \sin(\theta_j),$$

which is proportional to the oscillator signal $\sin(\theta_j)$ and can readily be used with the synchronization detection circuits proposed in Section 2.2, providing a more integrable readout approach.

Link with general oscillator models

The Thiele model provides a realistic description of vortex STNOs but can be cumbersome to simulate or not adapted to other similar oscillator technologies. This section investigates how well STNOs are described by a generic non-linear (Landau-Stuart) or even a linear (Kuramoto) model as used in our device-agnostic simulations.

Coupled phase-amplitude equations

The complex Thiele model expresses the evolution of the vortex core position X in the complex plane:

$$iG\dot{X} = -D\dot{X} - kX + i\kappa_\perp IX + (\kappa_{\mathrm{SL}} + i\kappa_{\mathrm{FL}})I, \tag{3.11}$$

where:

$$\begin{cases} D = \alpha G\left(\eta + \eta'\rho^2\right) \\ k = (k_{ms} + k_{Oe}I) + (k'_{ms} + k'_{Oe}I)\rho^2 \end{cases}. \tag{3.12}$$

The expression $\kappa_{\mathrm{SL}} + i\kappa_{\mathrm{FL}}$ is a complex number combining the in-plane effects of two orthogo-

nal torques: the Slonczewski and Field-Like torques. It can be rewritten in polar coordinates:

$$iG\dot{X} = -D\dot{X} - kX + i\kappa_\perp IX + I\kappa_\parallel e^{i\psi_\parallel}, \quad \text{where} \quad \begin{cases} \kappa_\parallel = \sqrt{\kappa_{\mathrm{SL}}^2 + \kappa_{\mathrm{FL}}^2} \\ \psi_\parallel = \tan^{-1}(\kappa_{\mathrm{FL}}/\kappa_{\mathrm{SL}}) \end{cases}. \tag{3.13}$$

This form makes the phase shift $\psi_\parallel$ induced by the in-plane torques explicit. The system parameters used in this study yield $\kappa_\parallel \approx 0.11$ and the phase shift $\psi_\parallel \approx 0.49\pi$ matches the value experimentally obtained by our collaborator R. Lebrun [169]: $\psi_\parallel^{\mathrm{exp}} \approx \pi/2$.

By setting $X = \rho e^{i\theta}$ where ρ is the amplitude and θ the phase of the oscillator, the real and imaginary parts of the resulting equation can be combined, knowing $D > 0$ and $G > 0$, into coupled amplitude and phase equations. Moreover, we split the noise terms into their real and imaginary parts and expand the resulting random walk terms for an integration timestep dt:

$$\begin{cases} \dot{\rho} = \dfrac{G\kappa_\perp I - Dk}{D^2+G^2}\rho - \dfrac{\kappa_\parallel}{\sqrt{D^2+G^2}} I \sin\left(\theta - \psi_\parallel - \psi^0\right) + \sigma\mathcal{N}(0,1) \\ \dot{\theta} = \dfrac{D\kappa_\perp I + Gk}{D^2+G^2} - \dfrac{\kappa_\parallel}{\sqrt{D^2+G^2}} \dfrac{I}{\rho} \cos\left(\theta - \psi_\parallel - \psi^0\right) + \dfrac{\sigma}{\rho}\mathcal{N}(0,1) \end{cases},$$
$$\text{with} \quad \psi^0 = \tan^{-1}(D/G) \quad \text{and} \quad \sigma = \frac{1}{rG}\sqrt{\frac{2k_B TD}{dt}}. \tag{3.14}$$

This convenient form of the Thiele model readily provides the radius stability condition of an isolated oscillator $G\kappa_\perp I = Dk$. In the case of the second order developments of $D(\rho^2)$ and $k(\rho^2)$ used here, this is a second degree polynomial in ρ^2 that can be solved analytically to obtain the **stable amplitude** ρ^0.

Simulation and subsequent spectral analysis show an oscillator linewidth of FWHM = 171.4 kHz which is consistent with experimental observations and solves the noise overestimation problem mentioned in Section 3.2.5. Moreover, when the condition $G\kappa_\perp I = Dk$ is injected as is in the phase equation $\dot{\theta}$ without the coupling term, it yields the natural frequency of the Thiele oscillator:

$$f_{\mathrm{thiele}}^0 = \frac{1}{2\pi}\frac{k}{G}. \tag{3.15}$$

Also note that since $D \ll G$, the phase shift component ψ^0 is considered low ($\psi^0 \approx 0.004\pi$ in my simulations) and $\psi_\parallel$ dominates.

Upper bound on the integration time step

Knowing the expression of the natural frequency f_{thiele}^0 and requiring 50 integration steps per period with a high margin against small frequency variations around this natural frequency, the upper bound on the integration step I used is:

$$\mathrm{dt} < \frac{1}{50} 2\pi \frac{G}{k}.$$

From Thiele to Landau-Stuart

As the frequency of a vortex STNO evolves quadratically with its amplitude, this device is an intrinsically non-linear oscillator. As shown in Section 2.4.5, this non-linearity can have an effect on the system. It is therefore important to evaluate if its description fits the generic Landau-Stuart model widely used in simulation of architectures involving non-linear oscillators. In this subsection, I analytically connect the Thiele model to the Landau-Stuart model using reasonable approximations.

Without couplings, the Thiele amplitude-phase equations (3.14) already resemble the Landau-Stuart model (Section 1.2.4):

$$\begin{array}{ll} \text{Thiele}: & \text{Landau}-\text{Stuart}: \\ \begin{cases} \dot{\rho} = \frac{G\kappa_\perp I - Dk}{D^2+G^2}\rho \\ \dot{\theta} = \frac{D\kappa_\perp I + Gk}{D^2+G^2} \end{cases} & \begin{cases} \dot{\rho} = \gamma({\rho^0}^2 - \rho^2)\rho \\ \dot{\theta} = 2\pi f^0 + \nu\gamma(\rho^2 - {\rho^0}^2) \end{cases} \end{array} .$$

A few approximations are however needed to obtain compatible expressions between the two models.

Developing terms around the equilibrium state

A major complexity of the Thiele model resides in the current or radius dependencies of parameters k and D.

However, the oscillating component I_{rf} of the current is small compared to its constant part I^0 (see Section 3.3). Because of this, the expression of the confinement k can be approximated by the case $I = I^0$:

$$k \approx (k_{ms} + k_{Oe} I^0) + (k'_{ms} + k'_{Oe} I^0)\rho^2 . \tag{3.16}$$

For the same reason:

$$\kappa_\perp I \approx \kappa_\perp I^0 . \tag{3.17}$$

Likewise, simulations show that after the transient period, the amplitude of an oscillator deviates by less than 15% from its stable amplitude ρ^0. The amplitude of an oscillator can therefore be expressed as:

$$\rho = \rho^0 + \delta\rho , \tag{3.18}$$

where $\delta\rho$ is a small deviation.

Using this expression, it is possible to develop the amplitude-dependent terms k and D to the first order in $\delta\rho$:

$$k \approx k^0 + k^1 \rho^2 , \tag{3.19}$$

where $k^0 = (k_{ms} + k_{Oe}I^0) + (k'_{ms} + k'_{Oe}I^0)\rho^{0^2}$ and $k^1 = (k'_{ms} + k'_{Oe}I^0)2\rho^0$.

And for D:

$$D \approx D^0 + D^1\delta\rho\,, \tag{3.20}$$

where $D^0 = \alpha G\left(\eta + \eta'\rho^{0^2}\right)$ and $D^1 = \alpha G\eta' 2\rho^0$.

In the fraction $\frac{1}{D^2+G^2}$, the constant G^2 in the denominator is more than 5000 times larger than D^2. The first order term D^1 can therefore be safely omitted:

$$\frac{1}{D^2+G^2} \approx \frac{1}{D^{0^2}+G^2}\,. \tag{3.21}$$

First order equivalence of the amplitude equations

The Thiele amplitude equation is not directly compatible with the one given by Landau-Stuart, but the two models have similar behaviors around the stable point of operation. The next step is therefore to derive and map the first order expressions of these two models.

Re-injecting the approximated factors back into the Thiele amplitude equation without coupling terms yields, after rearranging the terms:

$$\dot{\rho} = \frac{1}{D^{0^2}+G^2}\left(\left(G\kappa_\perp I^0 - D^0k^0\right) - \left(D^0k^1 + D^1k^0\right)\delta\rho\right)\rho\,. \tag{3.22}$$

At equilibrium, the stable radius ρ^0 verifies $\delta\rho = \dot{\rho} = 0$, which implies:

$$G\kappa_\perp I^0 = D^0k^0\,. \tag{3.23}$$

This is a second degree polynomial in ρ^{0^2} that can be solved exactly to obtain ρ^0 :

$$G\kappa_\perp I^0 = \alpha G\left(\eta + \eta'\rho^{0^2}\right)\left((k_{ms} + k_{Oe}I^0) + (k'_{ms} + k'_{Oe}I^0)\rho^{0^2}\right)\,. \tag{3.24}$$

Taking this into account allows to reduce the amplitude equation to a simple form:

$$\dot{\rho} = \left(-\frac{D^0k^1 + D^1k^0}{D^{0^2}+G^2}\delta\rho\right)\rho\,. \tag{3.25}$$

From this expression, we see that at first order, the Thiele amplitude equation behaves like a restoring force towards $\delta\rho = 0$. To identify this equation to its Landau-Stuart counterpart, we first need to develop the Landau-Stuart equation to the first order with $\rho = \rho^0 + \delta\rho$ in the same way:

$$\dot{\delta\rho} = (-2\gamma\rho^0\delta\rho)\rho\,. \tag{3.26}$$

As the two amplitude equations now have the same first-order form, it is possible to identify

the damping coefficient for radius deviation γ:

$$\gamma = \frac{D^0 k^1 + D^1 k^0}{2\rho^0 (D^{0^2} + G^2)} . \tag{3.27}$$

As a result, the Landau-Stuart amplitude equation equivalent at first order to the Thiele amplitude equation is:

$$\dot{\rho} = \gamma(\rho^{0^2} - \rho^2)\rho , \qquad \text{with} \qquad \gamma = \frac{D^0 k^1 + D^1 k^0}{2\rho^0 (D^{0^2} + G^2)} , \tag{3.28}$$

where the equivalent stable radius ρ^0 is found by solving the polynomial given in equation (3.24).

Equivalence of the phase equations

The Thiele phase equation without couplings directly maps to the Landau-Stuart with only minor approximations (D and k do not need to be linearized in $\delta\rho$):

$$\dot{\theta} = \frac{1}{D^{0^2} + G^2} \left(\kappa_\perp I^0 D + Gk\right) . \tag{3.29}$$

Expanding the terms and rearranging yields:

$$\dot{\theta} = \frac{1}{D^{0^2} + G^2} \left(\kappa_\perp I^0 \alpha G \eta + G(k_{ms} + k_{Oe} I^0)\right) + \frac{1}{D^{0^2} + G^2} \left(\kappa_\perp I^0 \alpha G \eta' + G(k'_{ms} + k'_{Oe} I^0)\right) \rho^2 . \tag{3.30}$$

This expression can be immediately mapped to the Landau-Stuart phase equation:

$$\dot{\theta} = 2\pi f^0 + \nu\gamma(\rho^2 - \rho^{0^2}) . \tag{3.31}$$

Identifying the dimensionless nonlinear frequency shift ν gives:

$$\nu = \frac{1}{\gamma(D^{0^2} + G^2)} \left(\kappa_\perp I^0 \alpha G \eta' + G(k'_{ms} + k'_{Oe} I^0)\right) . \tag{3.32}$$

This expression confirms that this dimensionless parameter that quantifies the non-linearity of the oscillator mainly depends on the second order physical parameters η', k'_{ms}, k'_{Oe}: without them, the oscillator behaves in a linear way. Its value in my simulations is $\nu \approx 2.04$ which is close to the value experimentally predicted by our collaborators $\tan^{-1}(\nu^{\text{experiment}}) \approx 2\pi/5 \implies \nu^{\text{experiment}} \approx 3$. The natural frequency f^0 can be readily found by injecting the amplitude equilibrium condition (3.23) into the phase equation (3.29):

$$f^0 = \frac{1}{2\pi} \frac{k^0}{G} . \tag{3.33}$$

As a result, the Thiele phase equation without couplings is almost perfectly equivalent to the

Landau-Stuart phase equation as defined by:

$$\dot{\theta} = 2\pi f^0 + \nu\gamma(\rho^2 - \rho^{0^2}), \quad \text{where} \quad \begin{cases} f^0 = \frac{1}{2\pi}\frac{k^0}{G} \\ \gamma = \frac{D^0 k^1 + D^1 k^0}{2\rho^0(D^{0^2}+G^2)} \\ \nu = \frac{1}{\gamma(D^{0^2}+G^2)}\left(\kappa_\perp I^0 \alpha G \eta' + G(k'_{ms} + k'_{Oe} I^0)\right) \end{cases} . \tag{3.34}$$

Coupling terms in the amplitude equation

Applying an oscillating current to an STNO acts on its amplitude. Injecting the total current applied to the oscillator j inside the circuit (equation 3.9) into the coupling part of the Thiele amplitude equation (3.14) yields:

$$\dot{\rho} = [\dots] - \frac{\kappa_\parallel}{\sqrt{D^2+G^2}} \sin(\theta - \psi_\parallel - \psi^0)\left(I^0 + \sum_m A_m \rho_m \sin(\theta_m)\right). \tag{3.35}$$

After expansion:

$$\dot{\rho} = [\dots] - \frac{\kappa_\parallel}{\sqrt{D^2+G^2}} I^0 \sin(\theta - \psi_\parallel - \psi^0) - \sum_m \frac{\kappa_\parallel A_m}{\sqrt{D^2+G^2}} \rho_m \sin(\theta_m)\sin(\theta - \psi_\parallel - \psi^0). \tag{3.36}$$

The first term $\sin(\theta - \psi_\parallel - \psi^0)$ is parasitic: it makes the radius oscillate with the phase of the oscillator itself. This term does not provide an explicit contribution to the synchronization dynamics as it does not involve other oscillators at the first order. We chose to approximate it out.

Moreover, $\sin(\theta_m)\sin(\theta - \psi_\parallel - \psi^0) = \frac{1}{2}(\cos(\theta_m - \theta + \psi_\parallel + \psi^0) - \cos(\theta_m + \theta - \psi_\parallel - \psi^0))$. As the absolute frequencies are high (hundreds of MHz) and are tuned in a small range, the term $\cos(\theta_m + \theta - \psi_\parallel - \psi^0)$ oscillates at about twice the frequency of the oscillator. Such high-frequency terms can perturb the oscillator dynamics, but do not induce synchronizations. As a result, we decided to neglect it as well.

These two relatively strong approximations, as well as $D \approx D^0$ in the denominator make the problem more tractable, and yield the following expression for the amplitude coupling term of oscillator i:

$$\dot{\rho}_i = [\dots] + 2\pi \sum_j k_{i,j}\rho_j \cos(\theta_j - \theta_i + \phi_{0i}), \qquad \text{where} \quad \begin{cases} k^{\mathrm{LS}}_{i,j} = \frac{\kappa_{\parallel i} A_j}{4\pi\sqrt{D_i^{0^2}+G_i^2}} \\ \phi_{0i} = \psi_{\parallel i} + \psi_i^0 - \pi \end{cases} . \tag{3.37}$$

This shows an explicit dependence on the phase differences with other oscillators $\theta_j - \theta$, which

is an important factor for entrainment and synchronization.

Note : Self-coupling

Note that the terms $\rho_m \cos(\theta_m - \theta + (\psi_\parallel + \psi^0 - \pi))$ of the coupling sum also include a contribution of the oscillator itself, for which the phase difference is zero. This adds a constant term to the amplitude expression, making the stable amplitude slightly higher, which increases the natural frequency as well, thus deviating from the point of operation f^0 expression derived earlier and around which the system was approximated. However, simulations show that the effects of this self-coupling term are low in practice, that the point of operation deviates only slightly, and that the approximations still hold.

Amplitude coupling allows indirect synchronization effects, as amplitude oscillations translate into frequency oscillations in non-linear oscillators.

Coupling terms in the phase equation

The Thiele phase equation (3.14) also contains a coupling term. The expression of the oscillating current in the circuit can therefore be injected into it to obtain:

$$\dot{\theta} = [\dots] - \frac{\kappa_\parallel}{\sqrt{D^2+G^2}} \frac{1}{\rho} \cos(\theta - \psi_\parallel - \psi^0) \left(I^0 + \sum_m A_m \rho_m \sin(\theta_m) \right). \tag{3.38}$$

Following the same procedure as for the amplitude coupling term, we neglect the parasitic term in $\cos(\theta - \psi_\parallel - \psi^0) I^0$. We also expand $\sin(\theta_m)\cos(\theta - \psi_\parallel - \psi^0) = \frac{1}{2}(\sin(\theta_m + \theta - \psi_\parallel - \psi^0) + \sin(\theta_m - \theta + \psi_\parallel + \psi^0))$ and neglect the double frequency term. These approximations, as well as $D \approx D^0$ in the denominator yield the following phase coupling for oscillator i:

$$\dot{\theta} \approx [\dots] + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}), \qquad \text{where} \quad \begin{cases} k_{i,j}^{\mathrm{LS}} = \frac{\kappa_{\parallel i} A_j}{4\pi\sqrt{D_i^{0^2} + G_i^2}} \\ \phi_{0i} = \psi_{\parallel i} + \psi_i^0 - \pi \end{cases}. \tag{3.39}$$

Complete Landau-Stuart model with couplings

Taking all these considerations into account, the Landau-Stuart model equivalent at first order to the Thiele model is defined by:

$$\begin{cases} \dot{\rho}_i = \gamma_i (\rho_i^{0^2} - \rho_i^2)\rho_i + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \rho_j \cos(\theta_j - \theta_i + \phi_{0i}) \\ \dot{\theta}_i = 2\pi f_i^0 + \nu_i \gamma_i (\rho_i^2 - \rho_i^{0^2}) + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}) \end{cases}. \tag{3.40}$$

In order to test how well this model matches the Thiele model, I simulate it in the same two-oscillator noiseless case as in Figure 3.5(b) with the same phase compensation to maximize synchronization, and compare the results to the Thiele model. Figure 3.7 shows the average frequencies of the two coupled oscillators as functions of the oscillator 1 bias current I_1^0, for the Thiele model (in blue, same values as in Figure 3.5(b)) and for the first order equivalent Landau-Stuart model (in red).

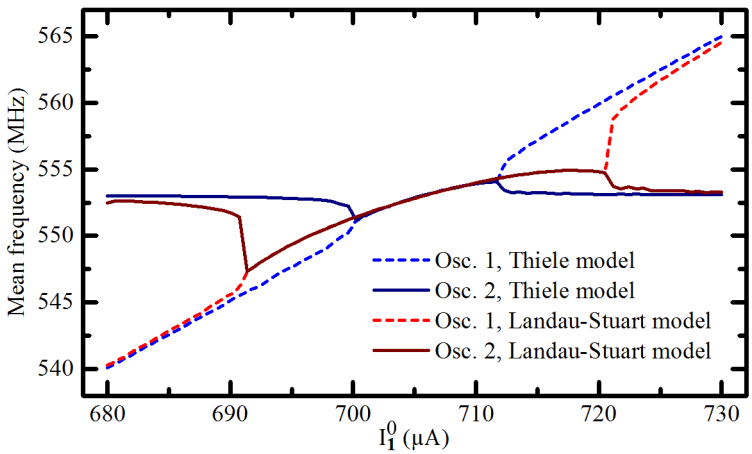


Figure 3.7: Mean frequencies of two electrically coupled nano-oscillators, as functions of the Osc. 1 current bias, as obtained through SPICE simulation with the Thiele model, and using the raw Landau-Stuart model. The raw Landau-Stuart model overestimates the synchronization range.

These results show that this Landau-Stuart model matches the Thiele model in areas far from synchronization which implies that the current-based natural frequency tuning is properly taken into account. The two models also match inside the synchronization range of the Thiele model, thus validating the Landau-Stuart description of this synchronized state. However, this version of the Landau-Stuart model overestimates the synchronization range by more than twice the Thiele prediction.

According to our tests, this inaccuracy lies in the neglected parasitic and double-frequency terms (sections 3.4.2.4 and 3.4.2.5) that can act as noise that reduces the synchronization range, as well and the slight shift in the point of operation due to self-coupling (remark in section 3.4.2.4). It also partly comes from circuit-related approximations neglecting some parasitic oscillations (Section 3.3.2). This results in a large under-estimation of the dynamics perturbing the system and acting against the stability of the synchronization. The net effect of this deviation is a $\approx 2.5\times$ overestimation of the effective coupling strengths $k_{i,j}^{\mathrm{LS}}$.

I therefore corrected the model by dividing $k_{i,j}^{\mathrm{LS}}$ by 2.5 and simulated it again. Figure 3.8 assesses the behavior of the corrected model in the same case as previously (Figure 3.7), comparing it to the Thiele model. This coupling-corrected Landau-Stuart model is in very good agreement, in all regions, with the full Thiele model with SPICE simulation. The coupling-corrected

Landau-Stuart model can therefore successfully approximate the behavior of STNOs:

$$\begin{cases} \dot{\rho}_i = \gamma_i(\rho_i^{0^2} - \rho_i^2)\rho_i + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \rho_j \cos(\theta_j - \theta_i + \phi_{0i}) \\ \dot{\theta}_i = 2\pi f_i^0 + \nu_i \gamma_i(\rho_i^2 - \rho_i^{0^2}) + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}) \end{cases},$$

$$\text{where} \quad \begin{cases} \gamma_i = \dfrac{D_i^0 k_i^1 + D_i^1 k_i^0}{2\rho_i^0 (D_i^{0^2} + G_i^2)} \\ \nu_i = \dfrac{1}{\gamma_i (D^{0^2}_i + G_i^2)} \left(k_{\perp i} I_i^0 \alpha_i G_i \eta_i' + G_i (k'_{\mathrm{ms}\,i} + k'_{\mathrm{Oe}\,i} I_i^0)\right) \\ k_{i,j}^{\mathrm{LS}} = \dfrac{1}{2.5} \dfrac{\kappa_{\parallel i} A_j}{4\pi \sqrt{D_i^{0^2} + G_i^2}} \\ f_i^0 = \dfrac{1}{2\pi} \dfrac{k_i^0}{G_i} \\ \phi_{0i} = \psi_{\parallel i} + \psi_i^0 - \pi \end{cases}. \tag{3.41}$$

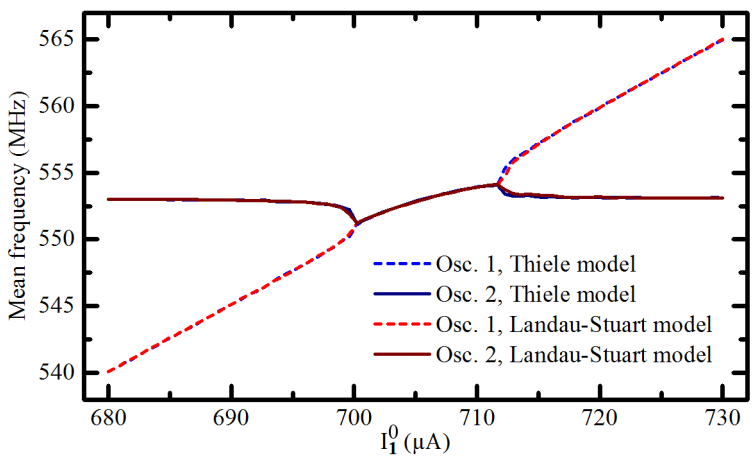


Figure 3.8: Mean frequencies of two electrically coupled nano-oscillators, as functions of the Osc. 1 current bias, as obtained through simulation with the Thiele model, and using a coupling-corrected Landau-Stuart model. Excellent agreement is achieved.

These results show that the behavior of non-linear STNOs is successfully approximated by a Landau-Stuart model, which illustrates the usefulness of this model as a generic description of non-linear oscillators. It also implies that an oscillator-based classification architecture built with vortex spin-transfer nano-oscillators is subject to the effects of non-linearities shown in Section 2.4.5.

Noise terms

The expression of the noise can be approximated by using $\sigma \approx \sigma^0$ around the equilibrium point:

$$\begin{cases} \dot{\rho}_i = \gamma_i(\rho_i^{0^2} - \rho_i^2)\rho_i + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \rho_j \cos(\theta_j - \theta_i + \phi_{0i}) + \sigma_i^0 \mathcal{N}(0,1) \\ \dot{\theta}_i = 2\pi f_i^0 + \nu_i \gamma_i(\rho_i^2 - \rho_i^{0^2}) + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}) + \frac{\sigma_i^0}{\rho_i^0} \mathcal{N}(0,1) \end{cases}, \tag{3.42}$$

where $\sigma_i^0 = \frac{1}{r_i G_i} \sqrt{\frac{2k_B T D_i^0}{dt}}$.

Simulations and subsequent spectral analysis show a linewidth of FWHM = 73.02 kHz, which is lower than the linewidth obtained from the Thiele model (171.4 kHz). This difference is mainly due to the neglected higher order terms that induced linewidth spreading. My simulations have shown that in this case, a correction factor $\sqrt{2}$ can be applied to the noise terms in the Landau-Stuart equations to match the Thiele model:

$$\sigma_i^{0\,(\mathrm{corrected})} = \sqrt{2}\sigma_i^0. \tag{3.43}$$

With this correction, I obtain a consistent linewidth of 186.3 kHz.

Linear approximation: connection with the Kuramoto model

Most simulations in this thesis use the generic linear Kuramoto model. It is therefore important to evaluate how well this model captures the behavior of a typical nano-oscillator. This subsection assesses whether this can be used to describe the complex case of the heavily non-linear vortex STNOs.

The starting point of this analytical study is the coupling-corrected Landau-Stuart model (equation 3.41) that successfully describes the vortex STNO:

$$\begin{cases} \dot{\rho}_i = \gamma_i(\rho_i^{0^2} - \rho_i^2)\rho_i + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \rho_j \cos(\theta_j - \theta_i + \phi_{0i}) \\ \dot{\theta}_i = 2\pi f_i^0 + \nu_i \gamma_i(\rho_i^2 - \rho_i^{0^2}) + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}) \end{cases}. \tag{3.44}$$

As the Kuramoto model assumes a constant amplitude, it demands $\dot{\rho}_i = 0$. This strong approximation transforms the Landau-Stuart amplitude equation into:

$$\gamma_i(\rho_i^2 - \rho_i^{0^2}) = 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \cos(\theta_j - \theta_i + \phi_{0i}). \tag{3.45}$$

This expression can be readily injected in the phase equation in place of $\gamma_i(\rho_i^2 - \rho_i^{0^2})$:

$$\dot{\theta}_i = 2\pi f_i^0 + 2\pi \nu_i \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \cos(\theta_j - \theta_i + \phi_{0i}) + 2\pi \sum_j k_{i,j}^{\mathrm{LS}} \frac{\rho_j}{\rho_i} \sin(\theta_j - \theta_i + \phi_{0i}). \tag{3.46}$$

Combining the trigonometric functions sin and cos, approximating the amplitude ratios $\frac{\rho_j}{\rho_i} \approx \frac{\rho_j^0}{\rho_i^0}$ for each term of the sum and adding the noise term yields the Kuramoto model:

$$\dot{\theta}_i = 2\pi f_i^0 + 2\pi \sum_j k_{i,j} \sin(\theta_j - \theta_i + \phi_i) + \sqrt{\frac{2\pi \mathrm{FWHM}}{dt}} \mathcal{N}(0,1)\,, \text{ where } \begin{cases} k_{i,j} = k_{i,j}^{\mathrm{LS}} \frac{\rho_j^0}{\rho_i^0} \sqrt{1+\nu_i^2} \\ \phi_i = \phi_{0i} + \tan^{-1}(\nu_i) \end{cases} . \tag{3.47}$$

In this expression the non-linearity ν increases the effective Kuramoto coupling strength by a factor $\sqrt{1+\nu_i^2}$, and induces a phase shift $\tan^{-1}(\nu_i)$. The noise term can be freely adjusted to match the desired linewidth FWHM = 171.4 kHz

In order to evaluate this model, I simulate the same noiseless two coupled oscillator case as in Figure 3.8 to compare it to the Landau-Stuart model. The same phase compensation is used as in the previous simulations, and verifies $\phi_i = 0$. Figure 3.9 shows that this Kuramoto model closely matches the Landau-Stuart model both inside and far from the synchronization range, with only minor deviations in the entrainment regions close to synchronization.

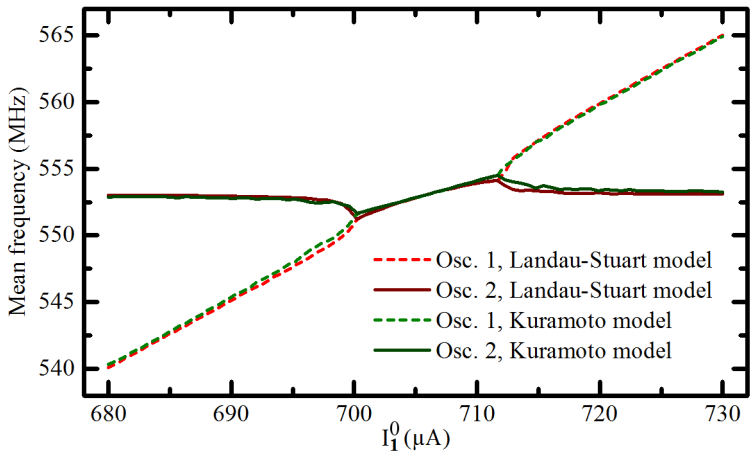


Figure 3.9: Mean frequencies of two electrically coupled nano-oscillators, as functions of the Osc. 1 current bias, as obtained through simulation with the Landau-Stuart model, and using the linear Kuramoto model. Good agreement is obtained between the two models.

Note : Non-linearity and Kuramoto

Our tests show that the Kuramoto approximation of the Landau Stuart model holds well up to reasonable values of the non-linearity ($\nu < 10$), which is the case in this chapter as well as in Section 2.4.5. Only the effective couplings and phase shifts are notably affected. However, the approximation breaks for extremely high values of ν.

The analytical Kuramoto equations of two coupled Vortex STNOs and their validation through

simulation illustrate the capacity of this model to successfully approximate even some non-linear oscillators.

A circuit implementing the pattern recognition architecture

In the previous section, the Thiele model and its Landau-Stuart and Kuramoto approximations were tested in a simple case with only two coupled STNOs. In this section, we design a circuit that implements the full pattern recognition architecture studied in Chapter 2 and simulate it in the noiseless case using the full Thiele model with circuit simulation. We also validate the Landau-Stuart and Kuramoto approximations on this larger system.

Figure 3.10(a) shows a schematic of the circuit that naturally couples 6 STNOs all-to-all with independently tunable natural frequencies, thus implementing the oscillator-based pattern recognition scheme described in Chapter 2. A minor difference is that all the couplings are identical (the input oscillators A ans B do not have stronger couplings) but this simplification due to circuit constraints does not influence the quality nor the fundamental behavior of the architecture. The current biases of the core oscillators are $\{f_1^0, f_2^0, f_3^0, f_4^0\} = \{703, 706, 709, 712\}\mu\text{A}$, the load resistor is $R_L = 50\,\Omega$ and the capacitor $C = 200\,\text{pF}$.

We first run a full circuit simulation using the Thiele STNO model. Figure 3.10(b) shows the synchronization response map of the core oscillators for different values of the current biases of the two input oscillators I_A^0 and I_B^0. The map is close to the reference map studied in Chapter 2 (Figure 2.1(b)) which confirms that this circuit effectively implements the expected pattern recognition behavior.

Some minor differences exist however. One of them is that the synchronization regions are more noisy, mainly due to second order oscillations acting like noise as explained in Section 3.4.2.6, thus making the stable synchronization regions thinner or almost absent such as the ■ region. The second notable difference is that the regions have slightly different shapes than in the reference architecture, and their shapes depend on the frequency ranges in which they appear. For example, ■,■,■ are increasingly thinner at higher frequencies. This is mainly due to the dependency of the system parameters on the value of the bias current as mentionned in Section 3.4.2.1 and to the non-linear dynamics of the oscillators: the stronger the bias current, the larger the amplitude of the oscillator becomes. However, synchronization regions globally behave as expected in the reference architecture.

Figure 3.10(c) shows the resulting synchronization map using the equivalent Landau-Stuart model (equation 3.41). This map is very similar to the Thiele map, and also includes the non-linearity related deformations such as the aforementioned pattern thinning. It is however slightly less noisy as the model does not include some second-order fluctuations (see sections 3.4.2.4 and 3.4.2.5), which makes patterns larger than in the Thiele case. Furthermore, the fully

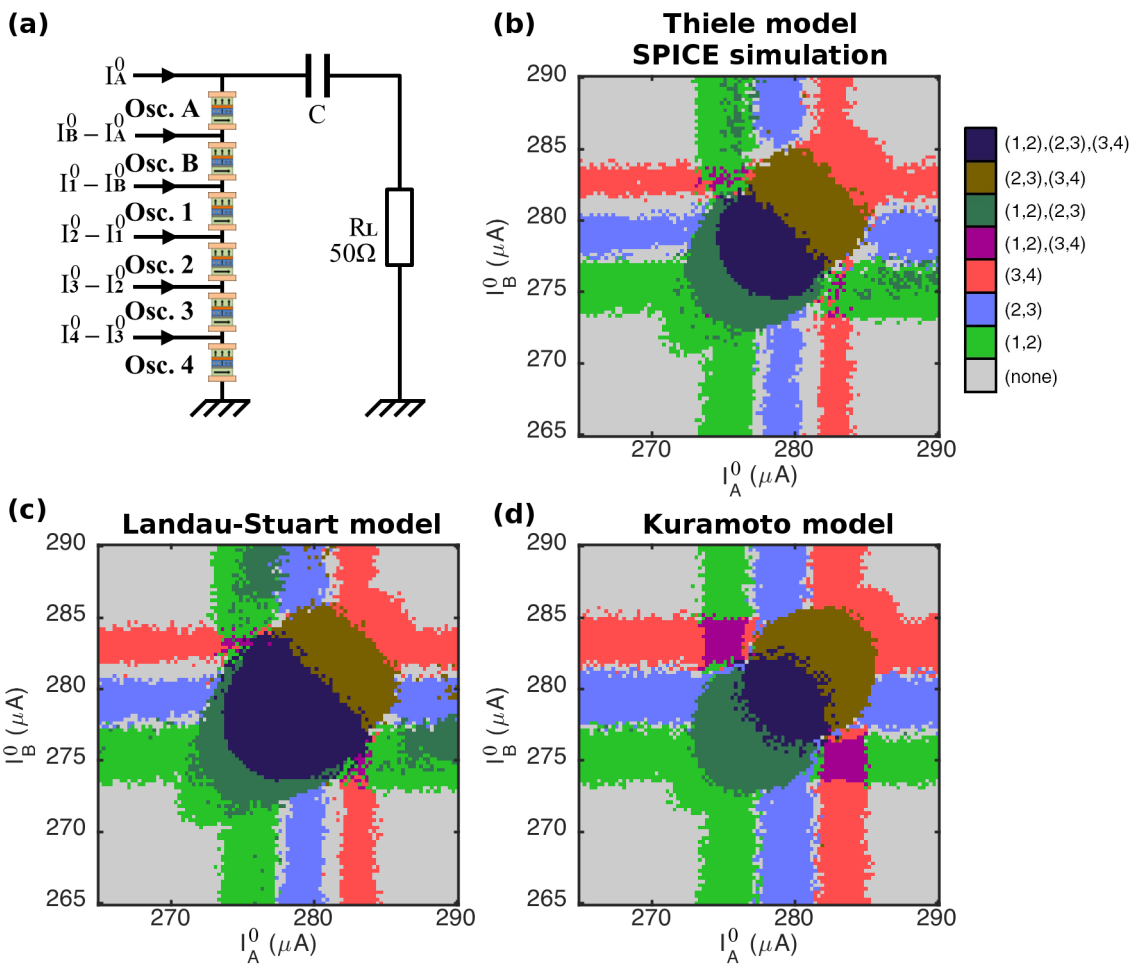


Figure 3.10: Simulation of the pattern recognition architecture with different models. **(a)** Circuit diagram. **(b)** Full circuit simulation with the Thiele compact model. **(c)** Simulation using the Landau-Stuart model. **(d)** Simulation using the Kuramoto model.

synchronous region ■ is notably bigger than in the reference linear map. This is due to the high number of oscillators synchronized through both phase and amplitude, which stabilizes multi-oscillator synchronization states more than in the Kuramoto model. Pattern stabilization also makes an small extra unstable pattern ■ appear on the top and right sides of tha map. Globally, the Landau-Stuart model successfully predicts the behavior of the Thiele model of the pattern recognition circuit, but tends to slightly overestimate synchronization stability.

The last map presented in Figure 3.10(d) shows the result of the same simulation using the equivalent Kuramoto model (equation (3.47). The synchronization regions appear less noisy and more uniform as the Kuramoto model approximates non-linearities away. This response map closely matches the reference map of Chapter 2 (Figure 2.1(b)), and provides a reasonable approximation of the behavior of the circuit of Thiele oscillators.

These results confirm that the proposed circuit conveniently implements the pattern recognition scheme at the heart of this thesis, and that the Landau-Stuart and Kuramoto models

correctly predict its behavior while allowing to generalize our study to many other devices.

Summary of the chapter

In this Chapter, I described the characteristics of vortex spin transfer nano-oscillators that behave like a two-terminal resistor with a resistance value that oscillates in time, and the more current is applied, the faster it oscillates, with the ability to reach GHz frequencies. Using existing theoretical results, we defined and validated a VerilogA SPICE model of the device. The model successfully describes vortex precession, current-based natural frequency tuning, and synchronization to an external oscillating source. However, the underlying theory is derived from fundamental physics, and corresponds to a second-order development of the dynamics of a typical, generic oscillator. As such, it is not tailored to accurately represent the behavior of the particular devices observed by our collaborators from Julie Grollier's group at the CNRS/Thales laboratory. In order to get a successful description of these particular devices, I adjusted the parameters of the model to match experimental results published by our collaborators.

Then, I described a circuit that implements the electrical coupling of multiple resistive nano-oscillators, and simulated it using our adjusted VerilogA device model. Circuit simulation results were in agreement with the experimental observations of our collaborators and successfully described the natural frequency control and synchronization dynamics of electrically coupled oscillators. To complete the understanding of this circuit, I derived the analytical equations of the total current flowing through each oscillator as functions of the oscillator resistance values.

Using the oscillator model equations, the analytical description of the circuit and multiple approximations, I rewrote the description of the oscillator network as a Landau-Stuart model, which is the simplest model describing networks of non-linear oscillator. Simulation results comparing the two models showed that the Landau-Stuart description was accurate but overestimated the synchronization range of a pair of coupled oscillators. After adding a single corrective factor to the coupling expression of the Landau-Stuart model, this mismatch, due to approximations, was successfully compensated. As a result, the corrected Landau-Stuart model was in perfect agreement with the initial circuit simulation results on a two-oscillator circuit.

By applying further approximations, I rewrote the Landau-Stuart model as a linear Kuramoto model. Simulation results comparing the resulting Kuramoto model with the Landau-Stuart model on the two-oscillator case showed good agreement between the two models.

Finally, I described a circuit that implements the oscillator-based classifier presented in Chapter 2 and computed its response map using circuit simulations. I also computed the equivalent response maps using the Landau-Stuart model, and the Kuramoto model. The three response maps were in good agreement, which shows that the proposed circuit successfully implements the oscillator-based classifier, and that the generic linear Kuramoto model used to study the circuit in Chapter 2 is indeed a good description of the system.

These theoretical results suggest that vortex spin transfer nano-oscillators are compatible with the constraints of the proposed architecture. Experiments performed by our collaborators give an oscillator phase noise level FWHM < 1 MHz, which allows the successful operation of the classifier. The oscillator-based classifier can therefore be implemented using vortex spin transfer nano-oscillators, and the robustness results obtained in Chapter 2 using a generic oscillator model fully apply to this case.

Moreover, our collaborators at the CNRS/Thales laboratory have experimentally implemented an architecture similar to the one described in this Chapter. A detailed description of their approach, as well as the associated learning algorithm are presented in Section 4.7.

Chapter 4

Machine learning with oscillator-based architectures

Machines take me by surprise with great frequency.

Alan TURING

"THE FOURTH CHAPTER *of this thesis presents how to train the oscillator-based classifier proposed in Chapter 2 to solve arbitrary classification tasks.* "

THIS CHAPTER focuses on achieving learning with the oscillator-based classifier proposed in Chapter 2 in order to solve arbitrary classification tasks while staying within the constraints of nano-technologies. The goals of this chapter are the following:

1. define learning algorithms allowing to shape the response of the oscillator-based classifier by adjusting its core natural frequencies, in order to solve arbitrary classification tasks given a set of known examples,
2. investigate the learning capabilities of the architecture on basic classification tasks, and deduce architectural improvements allowing to achieve better results,
3. propose and evaluate learning algorithms adapted to the oscillator-based classifier with "EEG-like" readout as well.

Introduction

In order to solve an arbitrary classification task with the oscillator-based classifier proposed in Chapter 2, the response of the system first needs to be shaped accordingly. This can be achieved by supervised learning, which involves presenting known examples to the classifier and adjusting the tunable parameters of the system, here the natural frequencies of the core oscillators, so that the right output is associated to each presented input example. After successful learning, the classifier can perform inference, which means that it can operate on new, unlabeled data.

This Chapter focuses on defining an offline learning algorithm for the oscillator-based classifier. Offline learning implies that the parameters of the network are learned within a simulated computer model of the classifier, and can then be transferred into a physical system for inference. After defining a novel offline learning algorithm dedicated to the oscillator-based classifier proposed in Chapter 2, I perform offline learning and evaluate the capabilities of the architecture on simple classification tasks. In the light of these results, I deduce the limitations of the architecture, and propose architectural changes to achieve better results while keeping the classifier compatible with the constraints of nano-technologies. I then evaluate the resulting extended oscillator-based classifier on typical machine learning tasks. I also propose a different learning approach for the "EEG-like" variant of the classifier introduced in Section 2.5.2, and investigate its learning capabilities. Finally, I introduce online learning possibilities through the experimental work of our collaborators at UMR CNRS/Thales. Contrary to offline learning, online methods carry out the learning process directly within a physical implementation of the classifier, but show other limitations.

An offline learning algorithm for oscillator-based classification

Offline learning algorithms use a numerical model of the oscillator network to perform the learning in simulation, outside of the system. Once the learning is done offline, the obtained parameters are transferred to the real system which can then perform inference. This approach therefore relies on the accuracy of the model, and differences between the real circuit and simulations can cause issues. However, the learning algorithm has access to all the internal state variables of the model, such as the actual phases of the oscillators, allowing advanced model-aware algorithms to be used. Inspiration for such algorithms can be drawn from usual machine learning techniques.

Supervised machine learning principles

The basic principle of supervised machine learning is to present known examples to a tunable computational model (typically a succession of non-linear transformations with tunable parameters $\boldsymbol{\Theta}$ such as the weights of a neural network as introduced in Section 1.4.7), compute a scalar error function E that measures the difference between the vector of outputs $\boldsymbol{h}$ of the model and the vector of expected outputs $\boldsymbol{y}$, adjust the parameters $\boldsymbol{\Theta}$ of the model to minimize this error, and repeat the process until the model correctly performs the desired task on the examples. Once successfully trained, the model can be used on real, unknown data.

Finding the right way to adjust the parameters of the model to minimize the error function E typically involves computing the partial derivatives $\frac{\partial E}{\partial \boldsymbol{\Theta}}$ of the error with respect to all model parameters, which define how the error changes when the corresponding parameters are adjusted. Positive values indicate that the parameter needs to be reduced and negative values indicate that it needs to be increased to minimize the error. The magnitude of the derivative of E with respect to a given parameter measures the sensitivity of the error E to this parameter. A simple optimization method, steepest descent, is readily deduced from this observation and follows:

1. For each example m from a batch of M examples:
   (a) Present the input example vector $\boldsymbol{x}^{(m)}$ to the model
   (b) Compute the output $\boldsymbol{h}^{(m)}$ of the model for example m
   (c) Compute the error function $E^{(m)}\left(\boldsymbol{h}^{(m)}, \boldsymbol{y}^{(m)}\right)$ for example m
   (d) Compute the derivatives $\frac{\partial E^{(m)}}{\partial \boldsymbol{\Theta}}$
2. Compute the mean error $E = \frac{1}{M}\sum_m E^{(m)}$ on the presented examples
3. Compute the mean derivatives $\nabla_{\boldsymbol{\Theta}} E = \frac{1}{M}\sum_m \frac{\partial E^{(m)}}{\partial \boldsymbol{\Theta}}$
4. Apply the parameter update rule: $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \alpha \nabla_{\boldsymbol{\Theta}} E$
   where α is the learning rate that defines the size of the parameter update steps.

5. Repeat the procedure until the mean error E is sufficiently low.

In the case of the oscillator-based classification architecture of Chapter 2, the examples are presented as sets of input oscillator natural frequencies $\boldsymbol{f}^{\mathbf{0}}_{\text{in}}$, and the tunable parameters of the system are the natural frequencies of the core oscillators $\boldsymbol{f}^{\mathbf{0}}_{\text{core}}$. The output of the system is a set of synchronization measures between core oscillators which needs to be properly defined theoretically.

Defining the output of the oscillator-based classifier

The output function for the oscillator-based classifier needs to be differentiable with respect to its parameters $\boldsymbol{f}^{\mathbf{0}}_{\text{core}}$, and to provide a measure of the synchronization quality of each output pair of core oscillators.

A convenient measure of the instability of the synchronization of an oscillator pair (i, j) is the ratio between the absolute value of the mean frequency difference between the two oscillators, and a synchronization threshold frequency under which the oscillators are considered synchronized:

$$D_{i,j} = \frac{|<f_i> - <f_j>|}{\text{thresh}}. \tag{4.1}$$

The mean frequencies $<\boldsymbol{f}>$ are only computed during the last T_{integ} period of the simulation to ensure convergence. If $D_{i,j} < 1$, the oscillators are considered synchronized, and if $D_{i,j} > 1$ they are considered not synchronized. More generally, as this desynchronization measure is continuous, the lower $D_{i,j}$ is, the more consistent the synchronization of oscillators i and j is assumed to be.

The threshold frequency thresh needs to be chosen carefully: too low values would force the learning algorithm to operate inside a region of very stiff synchronization dynamics, while too high values would make it operate in a regime of very low interaction between the oscillators where the outputs are linear transformations of the natural frequencies. For a uniformly coupled network of oscillators, the value of the threshold that provides the richest learning capabilities was found to be approximately half of the coupling strength k between the oscillators: thresh $= k/2$, which is equivalent to the thresholds defined for the synchronization detection schemes presented in Section 2.3.

The ensemble of $D_{i,j}$ values can therefore be used offline as the set of outputs of the classifier. A suitable error function can then be defined using these values.

Choosing an error function for oscillator-based classification

The error function $E^{(m)}$ for presented example m needs to be differentiable as well, and to take high values when expected synchronizations are not met, and low values when they match the expected synchronization pattern for this example.

The desynchronization measure $D_{i,j}^{(m)}$ for each output oscillator pair (i,j) adds a contribution $E_{i,j}^{(m)}$ to this error depending on the expected synchronization state for this pair. One possible expression of this contribution that was shown to work effectively in my simulations is:

if (i, j) synchronization expected :

$$E_{i,j}^{(m)} = \begin{cases} \frac{1}{2}\left(D_{i,j}^{(m)}\right)^2 & \text{if} \quad D_{i,j}^{(m)} \leq 1 \\ \frac{1}{1+e^{-4\left(D_{i,j}^{(m)}-1\right)}} & \text{if} \quad D_{i,j}^{(m)} > 1 \end{cases},$$

if (i, j) synchronization not expected :

$$E_{i,j}^{(m)} = \begin{cases} 1-\frac{1}{2}\left(D_{i,j}^{(m)}\right)^2 & \text{if} \quad D_{i,j}^{(m)} \leq 1 \\ 1-\frac{1}{1+e^{-4\left(D_{i,j}^{(m)}-1\right)}} & \text{if} \quad D_{i,j}^{(m)} > 1 \end{cases}.$$

These equations are plotted in Figure 4.1 as functions of the synchronization measure $D_{i,j}^{(m)}$.

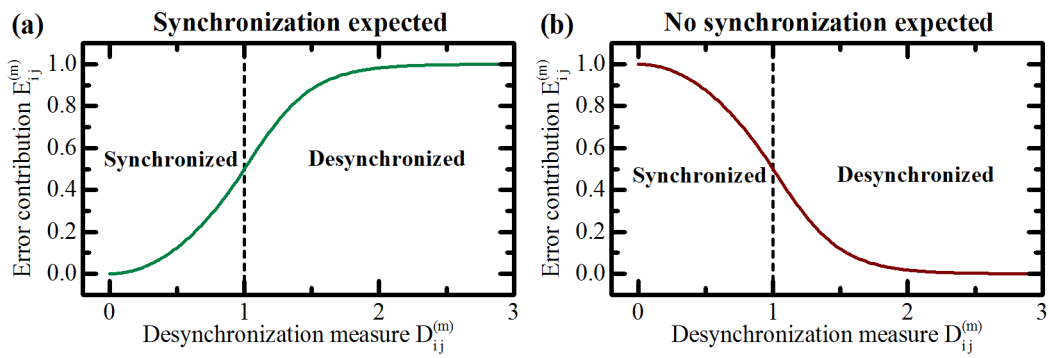


Figure 4.1: Error contribution of a pair of core oscillators (i,j) for example m, as a function of the desynchronization measure $D_{i,j}^{(m)}$. **(a)** When a synchronization of this pair is expected for example m. **(b)** When the pair is expected to be desynchronized for example m. The dashed line represents the limit between synchronization and desynchronization $D_{i,j}^{(m)} = 1$.

I chose these expressions because they are fully continuous, monotonous and differentiable everywhere, and take values within $[0,1)$ with 0 meaning that the synchronization of the pair is exactly as expected for the presented example, while values close to 1 mean that the pair has the opposite of its expected synchronization state. The quadratic left side of this function mainly penalizes expected synchronizations that are about to break ($D_{i,j}^{(m)}$ close to 1). Moreover, the sigmoid right side of this function saturates because the interaction between pairs of oscillators with very distant natural frequencies does not vary substantially as the natural frequencies change.

By combining the contributions of all observed pairs, we get the full expression of the error for example m:

$$E^{(m)} = \sum_{\text{output pairs } (i,j)} E_{i,j}^{(m)}. \tag{4.2}$$

A global view of the architecture including the error functions is presented in Figure 4.2.

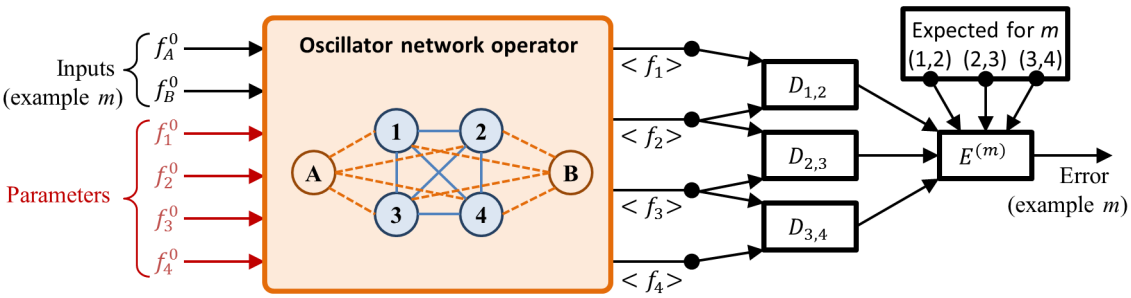


Figure 4.2: Block diagram view of the offline learning model for the oscillator-based classification architecture.

Differentiation of the error

The learning process involves the computation of the derivative of the error with respect to the natural frequencies of the core oscillators $\boldsymbol{f}^{\mathbf{0}}_{\text{core}}$.

$$\frac{\partial E^{(m)}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}} = \sum_{\text{output pairs } (i,j)} \frac{\partial E^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}}. \tag{4.3}$$

It is therefore necessary to find an expression of $\frac{\partial E^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}}$. Using the chain rule of differentiation, the following expression is obtained:

$$\frac{\partial E^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}} = \frac{\partial E^{(m)}_{i,j}}{\partial D^{(m)}_{i,j}} \frac{\partial D^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}}. \tag{4.4}$$

The derivative $\frac{\partial E^{(m)}_{i,j}}{\partial D^{(m)}_{i,j}}$ can be readily computed analytically from the expressions of the differentiable output errors given in Section 4.2.3.

The second factor $\frac{\partial D^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}}$ can also be expanded further from the expression of $D^{(m)}_{i,j}$ given in equation (4.1):

$$\frac{\partial D^{(m)}_{i,j}}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}} = \text{Sign}\left(< f_i > - < f_j >\right) \frac{1}{\text{thresh}} \left(\frac{\partial < f_i >}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}} - \frac{\partial < f_j >}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}} \right). \tag{4.5}$$

The derivatives $\frac{\partial < \boldsymbol{f} >}{\partial \boldsymbol{f}^{\mathbf{0}}_{\text{core}}}$ reflect how the mean frequencies in the network behave when the natural frequencies vary. This corresponds to the Jacobian matrix of the oscillator network transfer function (orange rectangle on Figure 4.2) taking natural frequencies as inputs, and providing mean frequencies as outputs. The Jacobian of the oscillator network does not have a simple expression as it reflects complex relationships resulting from the intricate interaction dynamics of the coupled oscillators, and alternative techniques are required to compute it.

Jacobian computation methods

No analytical transfer function has been found for a finite size oscillator network taking natural frequencies as inputs, and outputting the mean frequencies of the oscillators. This is also true for the matrix of its partial derivatives: the Jacobian of the oscillator network. To compute this Jacobian, I have developed two main approaches: numerical gradient computation, and forward propagation through numerical integration.

Numerical gradient computation

A straightforward way of obtaining an approximation of the Jacobian of the oscillator network transfer function is to perform a numerical differentiation, which is commonly used to verify the derivatives used in neural network libraries as it is model-agnostic. It consists in computing the initial outputs of the system, as well as its outputs after successively shifting each natural frequency by a small value df, one at a time. The derivative of the mean frequency of oscillator i with respect to the natural frequency of oscillator n is then approximately:

$$\frac{\partial < f_i >}{\partial f_n^0} \approx \frac{< f_i > \Big|_{f_n^0 + df} - < f_i > \Big|_{f_n^0}}{df}, \tag{4.6}$$

where $< f_i > \Big|_{f_n^0}$ is the mean frequency of oscillator i obtained without any natural frequency shift, and $< f_i > \Big|_{f_n^0 + df}$ is obtained after adding a small shift df to the natural frequency f_n^0 of oscillator n.

For a network of N_C core oscillators, this method requires simulating it $N_C + 1$ times to compute the Jacobian of the mean core oscillator frequencies as functions of the core natural frequencies. I have performed this computation using an HTML5/Javascript frontend served by a backend C++ server that dispatches all those independent computations, for all the examples, on different GPU-enabled computing servers, and aggregates the results to obtain the Jacobian.

Simulations on the analytically solvable two-oscillator case suggest that this method successfully provides a valid approximation of the Jacobian. However, this approximation of the Jacobian is noisy, and the choice of the df hyperparameter is critical, with an ideal value depending on the system state. This does not allow choosing a unique value for this hyperparameter, and makes this method unpractical.

Note : Advanced numerical differentiation approach

Note that an extended approach for numerical differentiation called "simultaneous perturbation stochastic approximation" may solve this problem as it uses decaying df and learning rate. It also requires only two evaluations per step but may need more steps to converge [199]. Unfortunately, I didn't find a scheduling scheme for this algorithm that improves the results of oscillator-based architectures.

Clamping the derivative

On the verge of synchronizations, oscillator systems follow stiff dynamics (see Figure 1.9), which gives rise to derivatives exceeding 10^{10} inside the Jacobian. When such a derivative appears for one presented example, its value dominates the contributions of all the other examples, and results in erratic optimization. To limit this effect, I clamped the derivative $\frac{\partial <f_i>}{\partial f_n^0}$ for each presented example within defined bounds. This can be achieved by any squashing function. I use tanh as it approximates unity for small values, and progressively clamps more extreme values between -1 and 1.

Forward derivative propagation through numerical integration

An alternative approach I developed computes the Jacobian of the oscillator network as the system is being simulated, with higher precision. For simplicity, the presented method uses basic Euler integration but it could be extended to more advanced integration schemes such as Runge Kutta.

During the numerical integration of the Kuramoto system, the values of all the variables of the system at time $t + \mathrm{dt}$ are given by a differentiable function of their values at the previous integration step t:

$$\theta_i(t+\mathrm{dt}) = \theta_i(t) + 2\pi\mathrm{dt}\left(f_i^0 + \sum_j k_{ij}\sin\left(\theta_j(t) - \theta_i(t)\right)\right). \tag{4.7}$$

By using the chain rule, it is therefore possible to keep track of the derivatives through the integration process, allowing the computation of the Jacobian of the oscillator network with the following algorithm:

1. Apply the natural frequencies $\boldsymbol{f^0}$ and choose random initial phases $\boldsymbol{\theta_0}$.
2. Initialize state variables (the initial state does not depend on the system inputs):

(a) For each oscillator a with respect to each natural frequency f_b^0, initialize the derivatives of the phases: $\frac{\partial \theta_a}{\partial f_b^0} = 0$

(b) Initialize the derivatives of the instantaneous frequencies: $\frac{\partial f_a}{\partial f_b^0} = 0$

(c) Initialize the mean frequency accumulators: $< f_a >= 0$, and their derivatives: $\frac{\partial < f_a >}{\partial f_b^0} = 0$

3. For each time step t until the total simulation time T_{sim}:

   (a) Compute the instantaneous frequencies $f_a = f_a^0 + \sum_j k_{aj} \sin\left(\theta_j - \theta_a\right)$

   (b) Compute their derivatives: $\frac{\partial f_a}{\partial f_b^0} = \delta_{a,b} + \sum_j k_{aj} \left(\frac{\partial \theta_j}{\partial f_b^0} - \frac{\partial \theta_a}{\partial f_b^0}\right) \cos\left(\theta_j - \theta_a\right)$

   (c) If $t \geq T_{\text{sim}} - T_{\text{integ}}$, where T_{integ} is the averaging time, accumulate the means:

      i. Accumulate the mean frequencies: $< f_a > \leftarrow < f_a > + f_a$

      ii. Accumulate their derivatives: $\frac{\partial < f_a >}{\partial f_b^0} \leftarrow \frac{\partial < f_a >}{\partial f_b^0} + \frac{\partial f_a}{\partial f_b^0}$

   (d) Update the phase derivatives: $\frac{\partial \theta_a}{\partial f_b^0} \leftarrow \frac{\partial \theta_a}{\partial f_b^0} + 2\pi \mathrm{dt} \frac{\partial f_a}{\partial f_b^0}$

   (e) Update the phases: $\theta_a \leftarrow \theta_a + 2\pi \mathrm{dt} f_a$

4. Normalize the means: $< f_a > \leftarrow < f_a > / \lfloor T_{\text{integ}}/\mathrm{dt} \rfloor$ and $\frac{\partial < f_a >}{\partial f_b^0} \leftarrow \frac{\partial < f_a >}{\partial f_b^0} / \lfloor T_{\text{integ}}/\mathrm{dt} \rfloor$

This algorithm provides the output of the oscillator network operator $< \boldsymbol{f} >$ as well as its Jacobian $\frac{\partial < f_a >}{\partial f_b^0}$ given the input natural frequencies $\boldsymbol{f^0}$.

As it comes from the analytical differentiation of the numerical update equations, this Jacobian computation approach has excellent precision, which makes it more effective than purely numerical approaches. However, this algorithm is also computationally heavy for large oscillator systems as the number of equations to integrate grows quadratically with the number of oscillators. Due to its precision, it remains however attractive for networks containing few oscillators such as the ones studied in this thesis, as well as in experiments. **From this point and until Section 4.5, Jacobian computations are performed using this method.**

Discussion and potential alternative approaches

Both numerical differentiation and forward-propagation through numerical integration work with systems containing few oscillators but have a limited scalability and become impractical for large networks of oscillators. Alternative differentiation approaches can be imagined, such as training a neural network to represent the transfer function of the oscillator network operator with high fidelity, and then use its Jacobian instead of computing the real one, for better computational efficiency.

Alternatively, other learning algorithms that do not require the derivative could also be used. This includes Genetic Algorithms [200] (discussed in Section 4.5.4) or inferring a learning policy using Reinforcement Learning [201].

Implementation in a standard machine learning framework

Using the same inspiration as for the forward derivative propagation through numerical integration, the numerical solving process of the oscillator network dynamics can be reinterpreted as a recurrent neural network, and therefore implemented as a standard layer in a machine learning framework.

Figure 4.3(a) illustrates the reinterpretation of the oscillator network as a recurrent neural network cell. This elementary cell takes the phases of the oscillators $\boldsymbol{\theta}(t)$ and their natural frequencies $\boldsymbol{f^0}$ as inputs, and outputs the instantaneous frequencies of the oscillators $\boldsymbol{f}(t)$ as well as their new phases for the next time step $\boldsymbol{\theta}(t+\mathrm{dt})$. This elementary cell is fully differentiable.

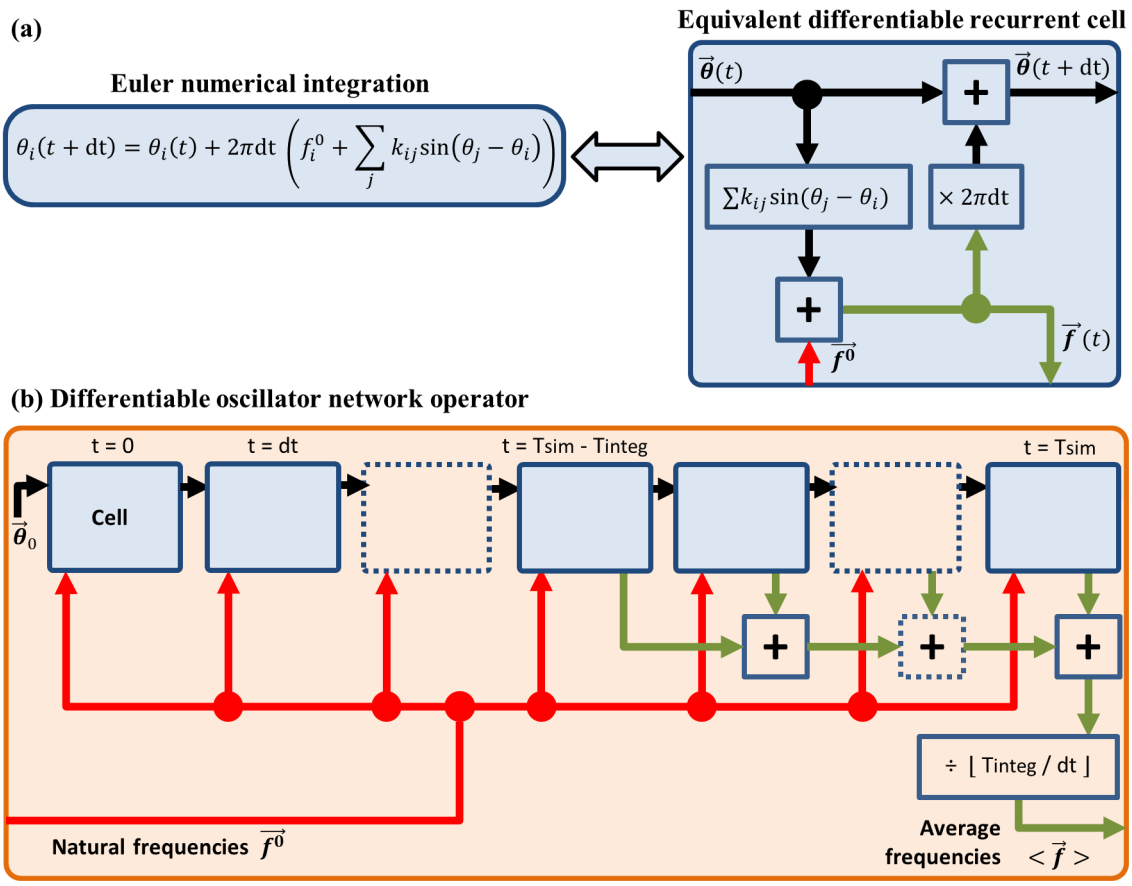


Figure 4.3: **(a)** Diagram of the elementary recurrent cell equivalent to the Euler integration update rule. **(b)** A fully differentiable oscillator network operator using a stack of these elementary cells to represent numerical integration.

In order to represent the full process of numerical integration, these elementary cells are stacked in time and act as a feed-forward differentiable network. Figure 4.3(b) illustrates an oscillator network operator that sets random initial phases $\boldsymbol{\theta_0}$ and runs the integration process for a total time T_{sim}. After waiting for the stabilization of the oscillator network dynamics, the instantaneous frequencies are averaged during T_{integ}. This full oscillator network operator

takes the natural frequencies of the oscillators $\boldsymbol{f}^{\mathbf{0}}$ as inputs, and outputs their mean frequencies $< \boldsymbol{f} >$. And as it is made of a feed-forward composition of differentiable functions, the operator is guaranteed to be fully differentiable.

As a result, this operator can be seamlessly used as a neural network layer inside a standard machine learning framework. To achieve that, I have implemented the oscillator network operator as a custom operator for the Tensorflow machine learning library developed by Google, and used extensively in the machine learning community. Highly optimized machine learning tools provided by the library can then be used to evaluate the oscillator-based classification architecture.

Tensorflow custom operator

A Tensorflow operator is a piece of compiled C++ code imported into the Python front end of the library, which defines two functions. The first reads the input tensor fed to the operator and computes the output tensor. The second reads the input and output tensors and computes the Jacobian.

The custom C++11 operator I wrote implements the forward differentiation algorithm presented in Section 4.2.5.2 using the Eigen linear algebra library, which provides native GPU acceleration allowing to efficiently run the operator on multiple examples in parallel.

Unlike conventional operators where the Jacobian is computed as a separate function of the inputs and outputs of the operator, the oscillator network operator computes its Jacobian at the same as its outputs. As a result, I define a C++ operator with two outputs: the mean frequencies, and the Jacobian. I then use a Python wrapper that has only one output mapped to the mean frequencies output of the underlying C++ operator, and a Jacobian computation function that returns the second output of the underlying C++ operator.

Note : Oscillator recurrent cell

The oscillator recurrent cell described here has an important property: the memory line $\boldsymbol{\theta}(t) \rightarrow \boldsymbol{\theta}(t + \mathrm{dt})$ is only transformed through a linear transformation (one addition) per time step. This property, which also gives their power to Long Short Term Memory recurrent neural networks [202], allows the gradients to be propagated back in time over long durations without vanishing due to squashing non-linear functions. This reduces the vanishing gradient problem and allows the learning to take into account long-term relationships.

In summary, using the proposed expression of the error, and the forward propagation through numerical integration technique for differentiation within a machine learning framework allows training the oscillator-based classification architecture on a large variety of classification tasks.

Learning binary classification

In this section, I investigate the learning capabilities of the oscillator-based classifier on the simple case of binary classification. The task consists in classifying two-dimensional inputs into one of two classes ("class 0" or "class 1") using the general architecture shown in Figure 4.4. The output is obtained by evaluating the synchronization state of only one pair of oscillators. If they are synchronized, the class that the system associates to the presented example is "class 1". If they are not synchronized, the output class is "class 0".

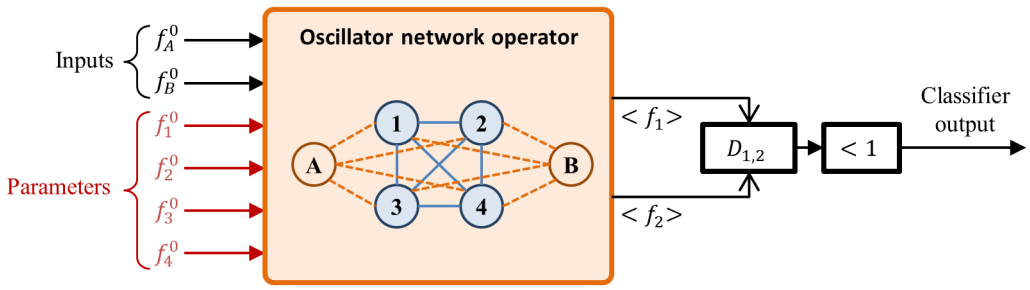


Figure 4.4: Diagram of the oscillator-based binary classifier.

The goal of the learning process is to adjust the natural frequencies of the core oscillators $f_1^0, f_2^0, f_3^0, f_4^0$ so that the influence of the inputs on measured oscillators 1 and 2, as well as their indirect influences through other oscillators, results in the correct classification of the presented examples. Note that the natural frequencies are clamped between $f_{\min}^0 = 450$MHz and $f_{\max}^0 = 750$MHz to avoid non-realistic values.

To provide an intuition of the learning process, I define a simple classification task for which the initial state is shown in Figure 4.5(a). It consists in classifying blue (class 0) and red (class 1) example points. The green background area in the map is where the two output oscillators are synchronized (output class 1), and the target of the learning is to reshape this area so that it covers the red (class 1) examples, and not the blue (class 0) ones. Note that the initial conditions for the core natural frequencies are set to be the same as in Chapter 2.

Figure 4.5(b) shows the response map after 10 learning iterations. The spurious side synchronization lines have almost disappeared, and the central blob has moved towards covering the red (class 1) examples. The classification quality has therefore improved since most red examples are correctly classified as class 1 and most blue examples are correctly classified as class 0.

Figure 4.5(c) shows the map after 20 learning iterations: the classification is now almost perfect. When a red (class 1) example is presented, the output oscillators synchronize to classify it as class 1, and when a blue (class 0) example is presented, they don't synchronize and classify it correctly as class 0.

The classifier also correctly generalizes to inputs it has not been trained with. For example, any new point in the same area as the red ones would be classified as class 1.

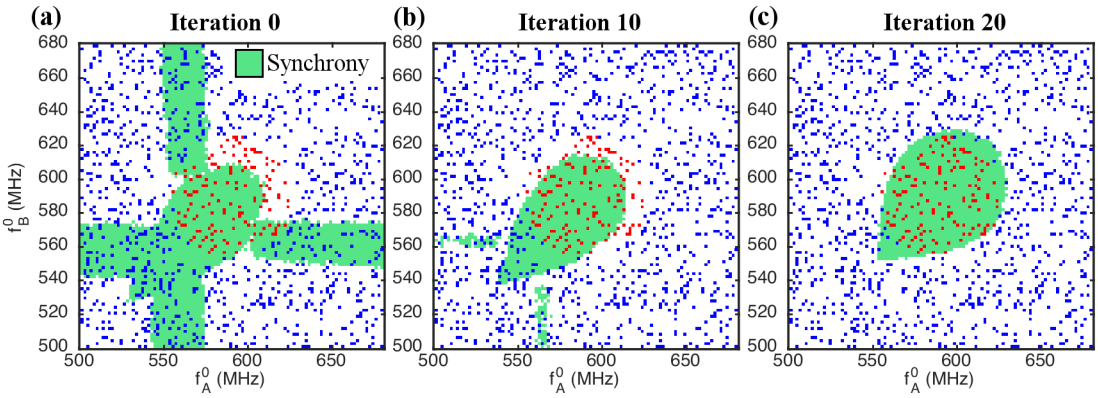


Figure 4.5: **(a)** Initial response map. **(b)** Response map after 10 learning iterations. **(c)** Response map after 20 learning iterations. The green area is where the two output oscillators are synchronized. Synchronization is expected for the class 1 examples (red dots), and no synchronization is expected for the class 0 examples (blue dots).

Inertia for better optimization

As the value of the error inside the parameter space is a complicated function with multiple saddle points and local minima, standard steepest gradient descent can get stuck into one of them, slowing the learning and preventing better solutions from being found. To reduce the impact of these limitations, I have used the ADAM optimizer [203], which performs gradient descent with inertia and second moment prediction, allowing convergence towards better solutions.

These results show that the learning process can work as expected. The classification capabilities of the architecture now need to be evaluated.

Classification capabilities

To get an intuition of the learning capabilities of the oscillator-based classifier, I perform binary classification on two-dimensional inputs.

Performance on a simple task

The classification task previously performed and illustrated in Figure 4.5 consisted in classifying a simple centered circular region of red examples, and was easily solved. In this part, I attempt the same task, but for different positions of the expected synchronization region in order to evaluate the sensitivity of the classifier to translation.

Figure 4.6 shows the response map of the system after 100 learning steps, and for three different positions of the target class 1 region:

- In Figure 4.6(a), the region of expected synchronization is still on the first diagonal of the map but is not centered. Simulation results show that this task is successfully solved by the classifier. More generally, the classifier is not sensitive to translation along the diagonal $f_A^0 = f_B^0$.
- In Figure 4.6(b), the region of expected synchronization is slightly above the center. After learning, the resulting output synchronization region does not fully cover the red examples: it stays on the first diagonal. This shows that the architecture is not capable of classifying this single off-diagonal region properly.
- In Figure 4.6(c), the region of expected synchronization is further away from the diagonal. After learning, the resulting output synchronization region stays stuck on the diagonal and only reaches a few class 1 examples. This confirms that the architecture cannot represent single off-diagonal regions. This is due to the fact that the input oscillators are identical, and only defined by their natural frequencies: when the two inputs are swapped, the same response map is obtained. This causes the response map of the system to be necessarily symmetric along the first diagonal, which represents a major limitation of the architecture.

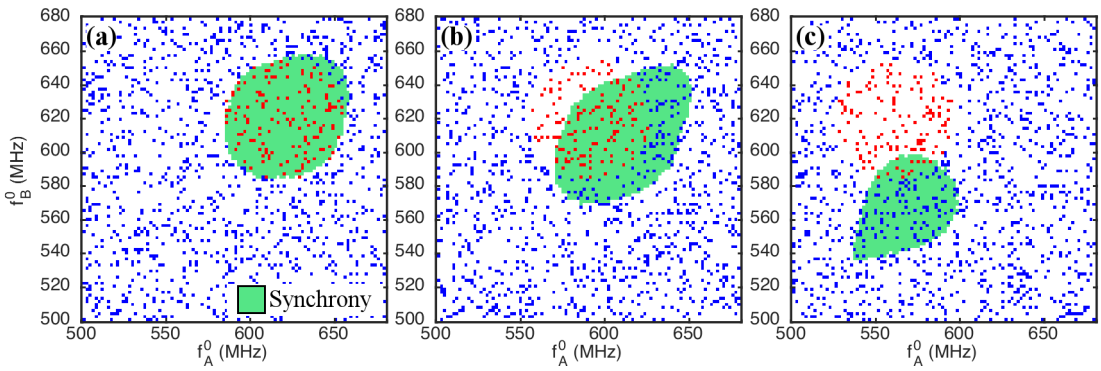


Figure 4.6: Classifying regions at three different positions. On **(a)** the region is off-center but on the first diagonal of the map, and on **(b)** and **(c)** it is off-diagonal.

These results show that the architecture, in its basic form, does not have sufficent classification power due to its intrinsic symmetries. Classifying regions that do not follow these symmetries would require a form of symmetry-breaking mechanism.

Freeing the coupling strengths

One possible approach to break the symmetry would be to allow the individual coupling strengths $k_{i,j}$ to be learned at the same time as the core natural frequencies. For N oscillators, this also provides N^2 more tuning parameters that could allow a finer tuning of the classification response.

Figure 4.7 shows response maps after 100 learning iterations for different regions of expected synchronization:

- The control run on Figure 4.7(a) confirms that allowing the adjustment of couplings does not hamper the classification of symmetric regions along the diagonal.
- Figure 4.7(b) shows the successful classification of a highly off-diagonal region. This shows that tuning the couplings allows the system to represent off-diagonal regions.
- Figure 4.7(c) shows the classification of a more complex, and non-symmetric shape for which more than 90% of the examples are successfully classified. This validates the effectiveness of this architecture.

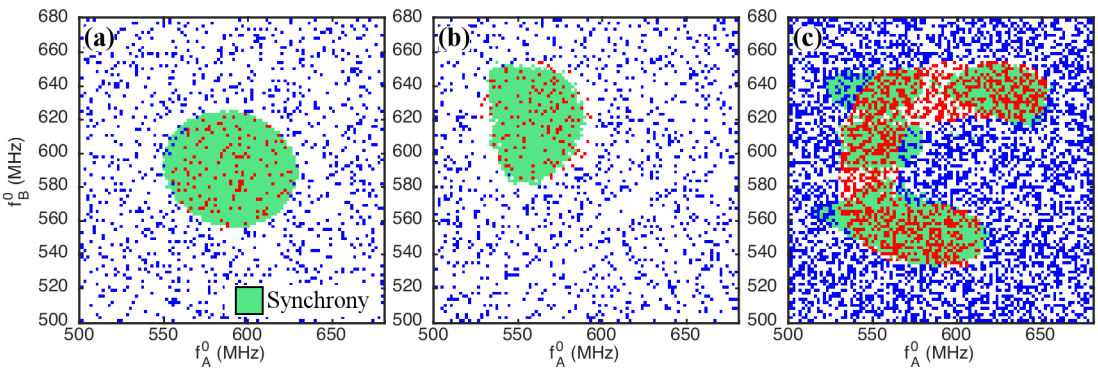


Figure 4.7: Maps after learning by optimizing natural frequencies and couplings for different target shapes.

These results show that allowing the tuning of the couplings during learning successfully breaks the symmetry, provides more tuning parameters, and allows the classifier to represent complex classes. However, this approach is particularly complicated to implement with nanotechnologies as coupling tunability is not physically convenient with most nano-oscillators. An alternative approach therefore needs to be developed to achieve similar symmetry-breaking and parameter space expansion while only acting on natural frequencies.

Making the learning more flexible

In order to achieve symmetry-breaking and increase the number of tunable parameters, I have defined a more general version of the oscillator-based classification architecture that is still compatible with nano-technologies.

Extended architecture

The proposed extended architecture is presented in Figure 4.8. It consists in applying different learned linear combinations of the inputs and a constant bias as the natural frequencies of the oscillator network. This is achieved through a tunable weight matrix W:

$$\boldsymbol{f}^{\mathbf{0}} = \boldsymbol{W} \begin{bmatrix} \mathbf{in} \\ 1 \end{bmatrix}. \tag{4.8}$$

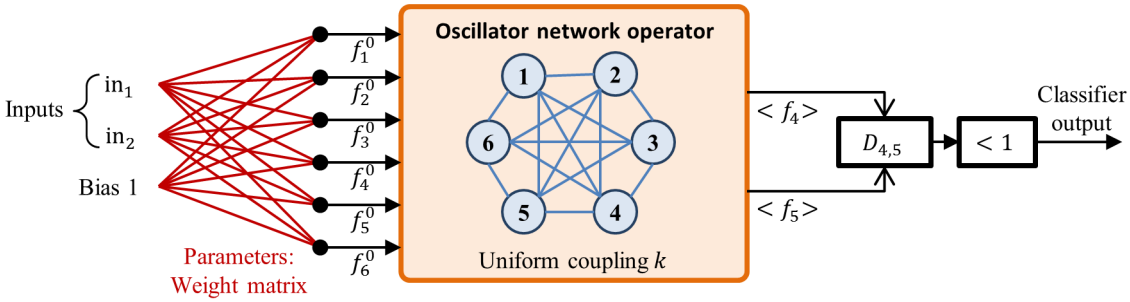


Figure 4.8: Extended version of the binary classification architecture showing the tunable weight matrix as red lines.

It is important to note that the architecture presented previously is a special case of this extended architecture where all the weights are zero except that each input is connected to the corresponding input oscillator $W_{11} = W_{22} = 1$ and the core oscillators are only connected to the constant bias with weights defining their natural frequencies. A specificity of the extended architecture is that presenting an example input can change all the natural frequencies of the network, instead of only acting on the input oscillators.

For N_{in}-dimensional inputs and N oscillators, the weight matrix provides a high number of parameters ($(N_{\text{in}} + 1)N$), and is not constrained by any symmetries which effectively allows symmetry breaking without changing inter-oscillator couplings. The inter-oscillator couplings are therefore constant and uniform, all the oscillators are identical, and only their natural frequencies are adjusted. The architecture is therefore still compatible with many nano-oscillator technologies.

Running inference with this architecture requires computing the matrix-vector multiplications (equation (4.8)). This can be implemented materially by a pre-processing step using classical multiplier chips, Tensor Processing Units [204] or synapse matrix techniques developed for neural network architectures such as SpiNNaker [205] or True North [206]. A more sophisticated approach would be to use a crossbar of resistive memories (or memristive devices) [207] to achieve this computation naturally through Kirchhoff's laws.

This extended oscillator-based pattern classification architecture is therefore a straightforward addition that inherits all the important properties described in the previous chapters, and only requires slight adjustments to the learning algorithm.

Learning capabilities

In order to evaluate the learning capabilities of this extended architecture, the same types of two-dimensional binary classification tasks as in Section 4.2.7 are performed. In this architecture, a coupling constant $k = 10$MHz is used.

Figure 4.9 shows the response maps of the system for different target synchronization regions. In these maps, the class 1 area of expected synchronization is marked in bright red and green colors, and class 0 areas where no synchronization is expected is marked in dark green and red colors: green areas correspond to successful classification, while red areas mark misclassification.

The initial weights before learning are set to match the basic classification architecture behavior presented in the previous chapters, plus uniform random variations in the range $[-10^{-3}; 10^{-3}]$. Figure 4.9(a) shows the response map of the extended architecture before learning for a simple task consisting in classifying a centered circular class 1 region. The response is reminiscent of the one in Figure 4.5(a). After 100 learning steps, Figure 4.9(b) confirms that this simple case is still successfully solved, as well as the off-centered but diagonal case in Figure 4.9(c). More interestingly, Figure 4.9(d) shows that the off-diagonal case is also successfully solved.

These results show that the extended architecture inherits the classification capabilities of the previous one, but is also capable of classifying off-diagonal regions.

I also test the architecture on more complex classification tasks. In Figure 4.10, I show the resulting response maps of the extended classification architecture after learning to classify non diagonal-symmetric stretched regions (Figure 4.10(a and b)), disjoint regions (Figure 4.10(c)) and concave regions (Figure 4.10(d)). All regions are successfully classified by the extended architecture using only 6 oscillators.

This preliminary study of the extended oscillator-based classifier architecture shows promising classification capabilities even with a small number of oscillators.

Performing a canonical machine learning task

In order to evaluate the behavior of the extended oscillator-based classifier on a real machine learning task, I use the simple and popular IRIS dataset which consists in classifying flowers into three different species given four features.

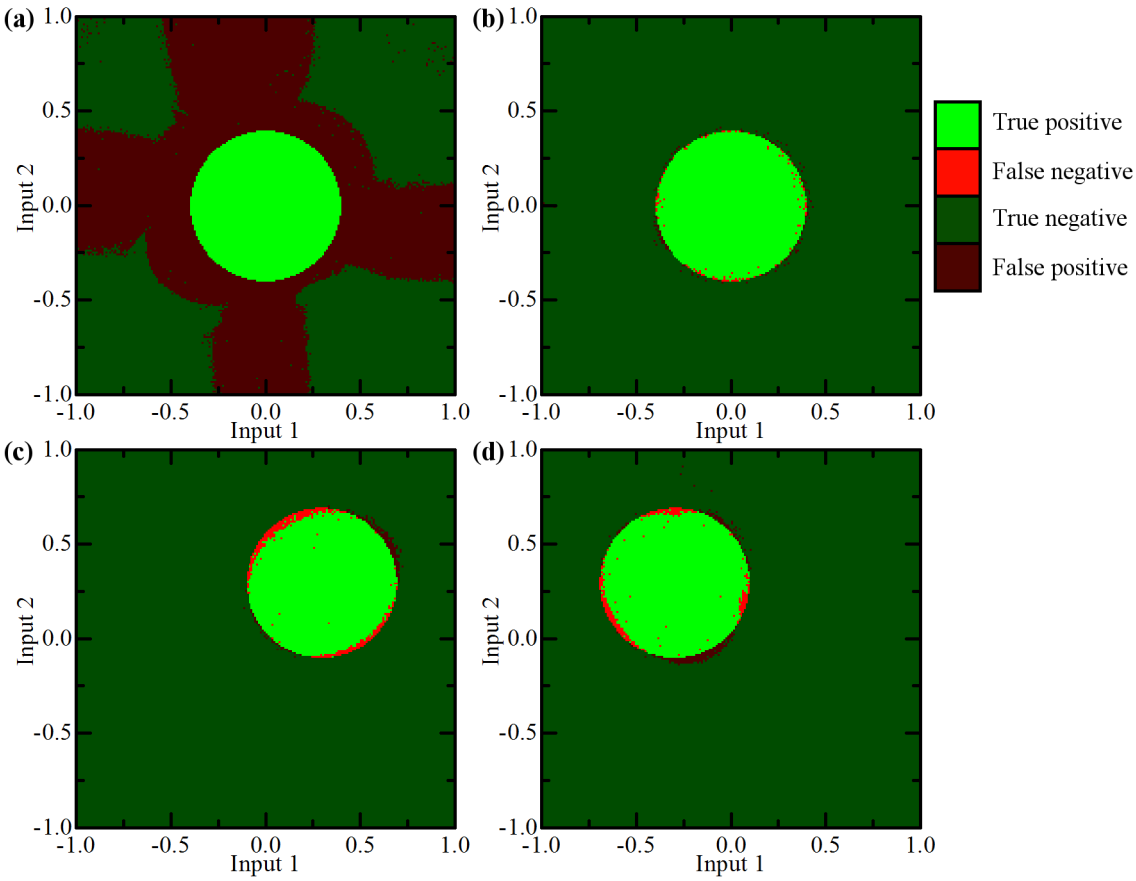


Figure 4.9: Basic classification task with the extended classification architecture. Bright areas represent expected synchronization, dark areas represent expected desynchronization. Green represents successful classification and red highlights misclassification. **(a)** Initial condition before learning on a centered circular pattern. **(b)** Resulting map after 100 learning steps. **(c)** Resulting map for a non-centered diagonal pattern. **(d)** Resulting map for an off-diagonal pattern.

The Iris dataset

The Iris dataset [208] is a historically significant table consisting of measurements performed on the flowers of three different species of the Iris plant genus: *Iris setosa*, *Iris versicolor* and *Iris virginica*. Each row of the table represents one sample flower through four of its features: the length and width of its petals, and the length and width of its sepals. It contains 150 samples, with 50 samples per species.

Given a target species, we train the oscillator-based classifier on the following binary classification task: synchronize the output oscillators if the presented flower is part of the target species, and to desynchronize them if it is part of any other species.

It is important to note that the species *Iris versicolor* is especially difficult to identify as it is not linearly separable from the other species. This means that linear classifiers (like most single-layer neural networks) can not find a hyper-plan in the feature space that separates it

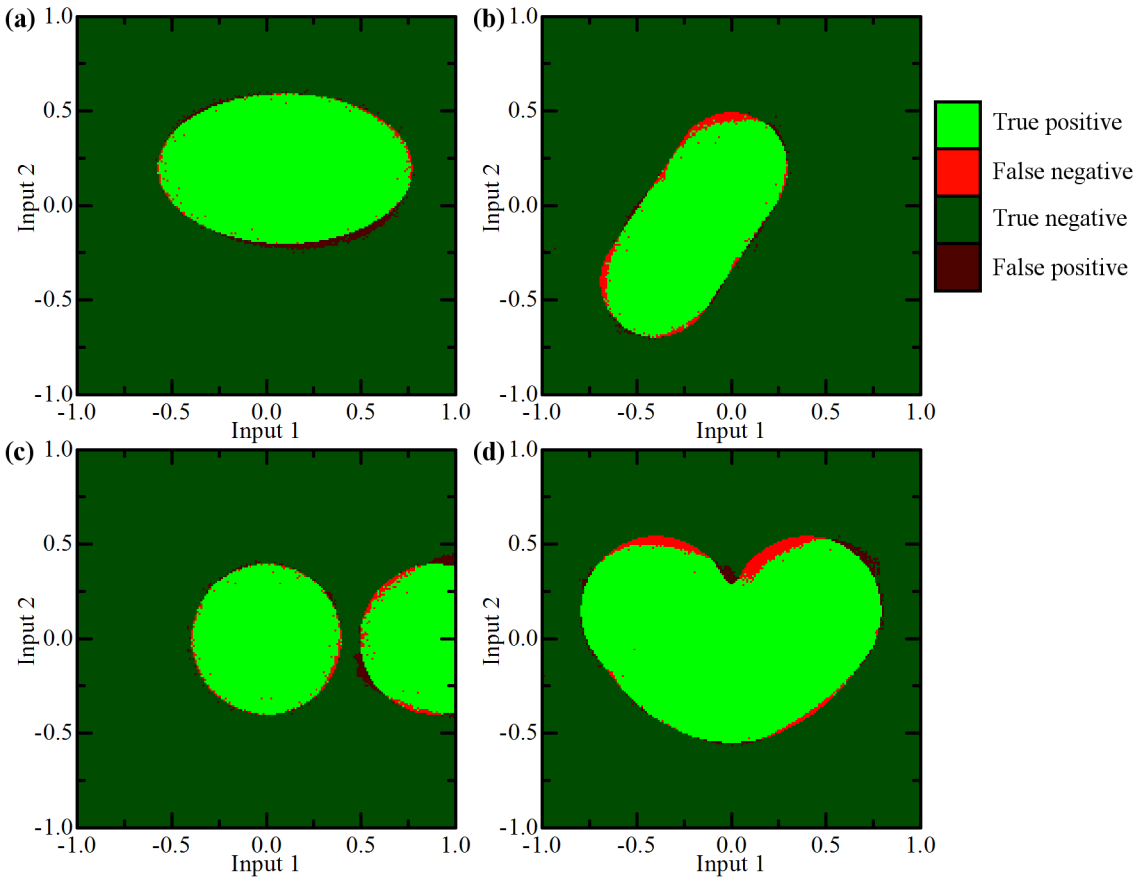


Figure 4.10: Response maps of the extended oscillator-based classification architecture after 100 learning steps on different non diagonal-symmetric stretched regions **(a)** and **(b)**, disjoint regions **(c)** and a concave region **(d)**.

from the other species.

In order to evaluate the generalization power of the architecture, I split the dataset into a training set (30 flowers from each species) and a testing set (the remaining 20 flowers for each species). The learning is performed only using the training set, and the classification rate is evaluated on the testing set examples that the network has not encountered during learning.

Classification quality

The performance of the extended oscillator classifier on the Iris dataset is evaluated by training the network on the binary classification task for different target species.

Figure 4.11 shows the classification rate on the test set as a function of the learning step for each target species. It shows that the learning process successfully increases the classification rates which stabilize after about 120 learning steps. At the end of the learning process, the *Iris setosa* and *Iris virginica* species are perfectly classified for all test examples, while the non-separable class *Iris versicolor* is successfully identified for 98.3% of the test examples.

These results show that the proposed architecture successfully solves a reference classification task. Moreover, the successful identification of the non-linearly separable class *Iris versicolor* suggests that due to its non-linearities, the oscillator-based classifier is capable of performing more advanced tasks than typical single-layer neural networks.

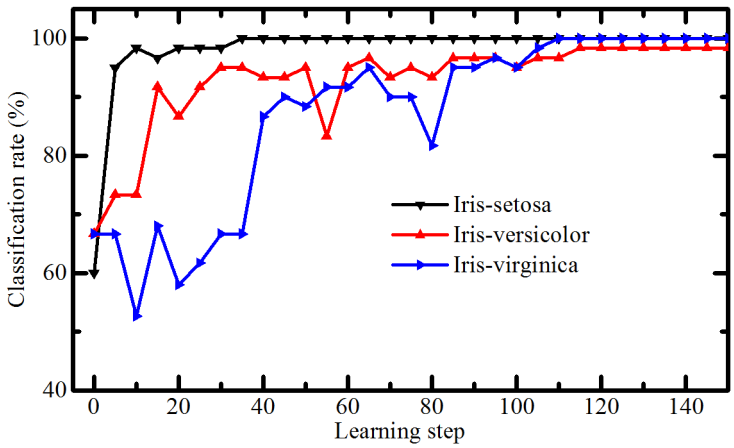


Figure 4.11: Evolution of the test set classification rate of each Iris class by the extended oscillator-based classification architecture, as functions of the learning iteration.

Influence of the number of oscillators

As the dynamics of the oscillator network get more complex with more oscillators, the classification capabilities of the oscillator-based classifier are expected to increase with its size. In order to characterize how these capabilities scale, I perform learning on the classification of the non linearly separable *Iris versicolor* class with different numbers of oscillators in the network. I also compare it to a classical two-layer tanh neural network with different numbers of neurons. Figure 4.12 shows the test set classification rates of the extended oscillator-based classifier and the tanh neural network, as functions of the number of units (oscillators or neurons).

With two oscillators, the oscillator classifier appears to perform substantially better with more than 86.67% test set classification rate versus 73.33% for the two-neuron neural network. With three units, both architectures perform similarly with 95% classification rate. For higher numbers of units, the classification rates stabilize at 98.33% for the oscillator classifier and 96.67% for the neural network.

These results show that the single-layer oscillator-based classifier performs similarly to a two-layer neural network of the same size. These observations confirm that by exploiting its complex dynamics, a single layer using the proposed architecture can perform advanced classification tasks.

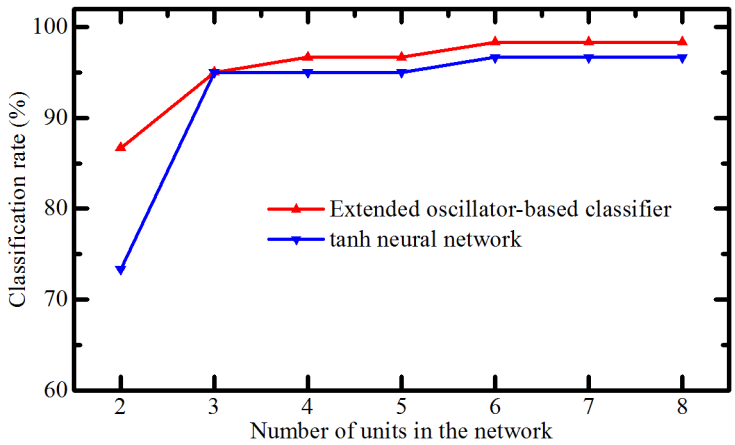


Figure 4.12: (▲) Test set classification rate of the Iris-Versicolor non linearly separable class after learning, as a function of the number of oscillators in the network. (▼) Classification rate on the same task using a 2-layer tanh-based neural network, as a function of the number of neurons.

Impact of phase noise on the classification rate

As the learning is done offline and the weights can then be transferred to a real oscillator system, the impact the phase noise present in the physical system needs to be assessed. In order to characterize the effect of phase noise, I carry out offline learning on the *Iris versicolor* class using a noiseless 6-oscillator extended oscillator classifier and then perform inference on the test set by adding increasing amounts of oscillator phase noise. For each value of oscillator linewidth, I perform 100 inference tasks with different seeds for the noise and the initial phases of the oscillators. Figure 4.13 shows the average test set classification rate obtained on 100 trials as a function of the oscillator linewidth. The blue-filled region represents the span between the worst and best classification rates encountered during the trials.

The data shows that the classification rates decrease slowly with increasing noise. For noise levels typical of spin torque nano-oscillators (FWHM = 1MHz), the classification rates stay around 94.6% on average, with a worst case encountered at 90%. For extremely high noise (FWHM > 5MHz), the worst cases encountered reach 66.67% which is equivalent to the classifier outputting 0 for all the examples (which is right 2/3 of the time). The classifier therefore ceases to be reliable at such noise levels. These results confirm that the extended oscillator-based classifier with offline learning is robust to phase noise, allowing it to function with classical STNOs.

The noise sensitivity of the architecture could in principle be further reduced if noise is included during learning, which forces the learning algorithm to reject solutions giving low classification rates under the presence of noise. Unfortunately, my tests show that the offline learning algorithm I proposed does not function properly when noise is present due to extreme

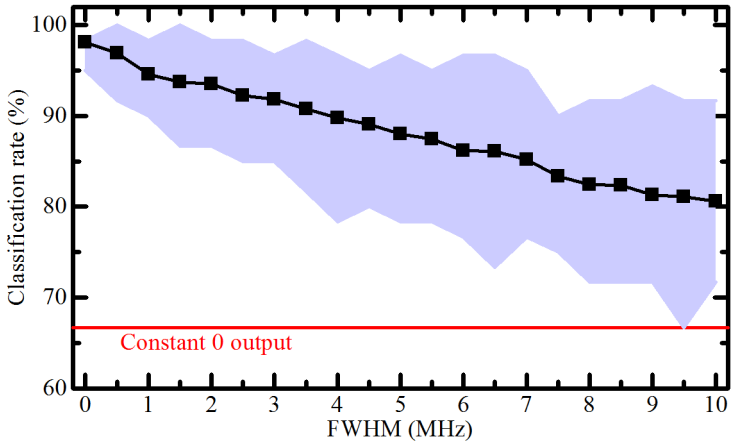


Figure 4.13: Test set classification rate on the Iris-Versicolor class using the extended oscillator-based classification architecture after learning, as a function of the oscillator linewidth (FWHM). The figure shows the average classification rate on 100 trials, and the filled background represents the span between the minimal and maximal classification rates encountered during trials. The horizontal red line is the classification rate obtained if the classifier always outputs 0.

derivative values. Further post-processing techniques for derivative smoothing, or alternative learning algorithms are therefore needed to achieve offline learning with noise.

Discussion on the extended oscillator-based classifier

Simulation results show that the extended oscillator-based classifier can perform binary classification efficiently and scale better than classical neural networks as it exploits to the computational power brought by the complex physics of the oscillator network. However, gradient-based learning methods only work offline and can not take noise into account during learning, which increases the noise sensitivity of the trained system.

The presented approaches could naturally be extended to multi-class classification by reading the synchronization state of as many oscillator pairs as there are classes. Moreover, as the oscillator network operator is fully differentiable, multi-layer architectures composed of stacked oscillator networks could be trained with this method and interfaced with other differentiable operators. For example, in a two-layer architecture, the input natural frequencies of the second layer network would be a learned linear combination of the output average frequencies of the first layer network. This could allow the system to perform more advanced tasks requiring more abstraction.

In summary, this promising architecture exploits the powerful dynamics of nano-oscillator for cognitive tasks, shows good results in simulation, and could be further improved. Possible future directions include studying multi-class and multi-layer classification, signal, sound and

video classification in the time domain by using transient behaviors as well, and implementing the architecture experimentally.

Alternative "EEG-like" readout and adapted learning approaches

The previous sections of this Chapter have focused on architectures where synchronization detection between oscillators is used as the readout method for the classifier output.

An alternative readout method for the oscillator-based binary classification architectures is introduced in Section 2.5.2 and consists in taking the envelope S of the average signal of the oscillators. In Section 3.3.2, I have shown that this signal can be readily obtained from a typical STNO electrical coupling circuit. In this section, I investigate methods to achieve learning using this alternative readout approach.

In the particular case of $N = 2$ synchronized oscillators, the analytical expression of the EEG-like signal S can be derived from the Kuramoto model. In the first subsection I study how a pair of oscillators can be used as a neuron, and how these neurons can be combined to build a neural network and achieve robust learning through standard gradient descent techniques.

In the second subsection, I apply the EEG-like readout method to the extended oscillator-based binary classifier. As gradient-based methods are not satisfactory in this case, I focus on an alternative bio-inspired learning algorithm.

Neural networks based on synchronized pairs of oscillators

The case of $N = 2$ oscillators is well understood in the Kuramoto framework, and analytical expressions for the system variables and their average values can be found. As stated in section 1.4.7, artificial neurons based on pairs of coupled oscillators leverage this knowledge but only implement simple threshold neurons.

In this part, I propose an alternative approach with a neuron design using a pair of coupled oscillators kept in a synchronized state with an analog, differentiable EEG-like signal output. I derive the analytic expression of its activation function, show that this neuron is capable of representing generic logic gates, and assess its capacity to learn these gates through gradient descent. I then simulate the dynamics of artificial neural networks using these neurons on two standard machine learning tasks, and verify their robustness to the intrinsic noise of nano-oscillators.

Oscillator-based neuron

An artificial neuron needs to take a weighted sum of values as input, and output its value after transformation by a non-linear activation function. Such neurons can then be stacked to form

an artificial neural network as shown in Figure 4.14.

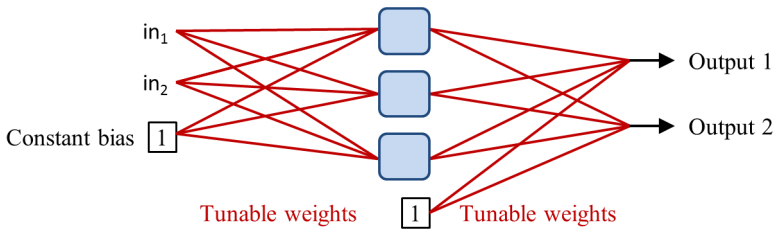


Figure 4.14: A simple feed-forward neural network, where each blue rounded rectangle is an artificial neuron implementing a non-linear activation function.

The neuron circuit I propose is shown in Figure 4.15(a) and relies on a pair of bidirectionally coupled oscillators kept in a synchronized state. In my simulations I use $k = 8$MHz.

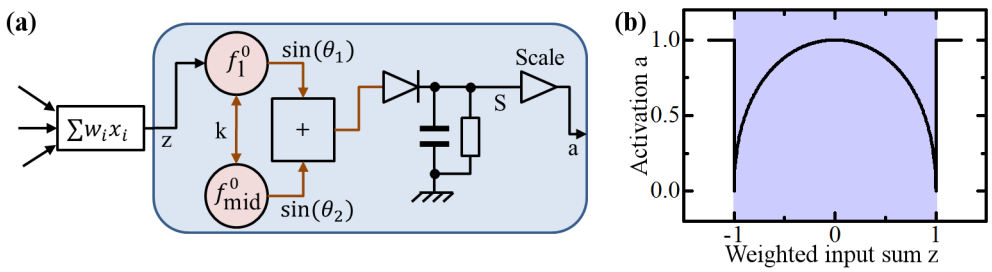


Figure 4.15: **(a)** Simplified schematic of the proposed neuron. Red circles are oscillators. **(b)** Neuron activation function for $f_{amp}^0 = 2k$. The neuron operates in the phase locked (blue filled) region.

The natural frequency of the second oscillator is kept constant ($f_2^0 = f_{mid}^0$), while the natural frequency of the first oscillator is tuned by the weighted sum of the neuron inputs $z = \sum_i w_i x_i$ where x_i is the i-th input of the neuron and w_i the corresponding weight value. Assuming that the natural frequency response of the first oscillator is proportional to this command by a factor f_{amp}^0, its natural frequency is then $f_1^0 = f_{mid}^0 + z f_{amp}^0$. An envelope detector is applied to the sum of the two oscillator signals $\sin(\theta_1) + \sin(\theta_2)$. For simplicity, the resulting EEG-like signal S is rescaled between 0 and 1 which defines the output of the neuron $a = (S - \sqrt{2})/(2 - \sqrt{2})$.

The activation function of the neuron, which corresponds to its output as a function of the weighted input sum, is shown in Figure 4.15(b) and can be expressed analytically. Envelope detection on $\sin(\theta_1) + \sin(\theta_2) = 2\cos\left(\frac{\theta_1-\theta_2}{2}\right)\sin\left(\frac{\theta_1+\theta_2}{2}\right)$ yields $S = \left|2\cos\left(\frac{\theta_1-\theta_2}{2}\right)\right|$. As the two oscillators are synchronized, they run at the same instantaneous frequency $\dot{\theta}_1 = \dot{\theta}_2$. By applying this constraint to the Kuramoto equations, a more useful expression of S is obtained:

$$S = \sqrt{2 + 2\sqrt{1 - \left(\frac{z f_{amp}^0}{2k}\right)^2}} \;\; ; \;\; a = \frac{S - \sqrt{2}}{2 - \sqrt{2}}. \tag{4.9}$$

The paraboloid-shaped activation function (4.9) is fully differentiable in the synchronization region, which allows error derivative computation. However, it is non-monotonous, unlike the sigmoid activation functions used conventionally in artificial neural networks.

Capabilities of the oscillator-based neuron

To assess the capabilities of this unconventional neuron, I perform the learning of two-input logic gates by simulating (4.9) using realistic parameters $f^0_{\text{mid}} = 500\text{MHz}$, $f^0_{\text{amp}} = 10\text{MHz}$, $k = 6\text{MHz}$, typical of spin-torque nano-oscillators, an exponential-decay envelope detector, and the standard steepest gradient descent learning algorithm. The two input weights, and one bias value are learned for $1,000,000$ different sets of initial weight and bias values to assess the robustness of the results.

Table 4.1: Proportion of initial states converging to the correct logic function after learning. Results shown for different logic gates, and for oscillator-based, sigmoid-based, and threshold-based neurons.

Target gate	Oscillator neuron	Sigmoid neuron	Threshold neuron
XOR	15.8	0	0
XNOR	29.8	0	0
Others	100	100	100

Table 4.1 shows the percentage of initial conditions leading to successful logic gate learning, and compares it with a traditional sigmoid neuron, and with a threshold neuron. The results show that the oscillator neuron successfully learns all the gates the threshold or sigmoid neurons can learn, and with favorable initial conditions it also learns to solve XOR and XNOR, which is not achievable by a sigmoid nor a threshold. This is allowed by the non-monotonous activation function of the oscillator pair: each of its lobes ($z < 0$ or $z > 0$) can be used as classical non-linear activation functions while using different lobes for different input examples allows more advanced operations. The energy landscapes of XOR and XNOR exhibit multiple and different local minima which accounts for their non-perfect and different scores. As the amplitude of the gradient with oscillator neurons is higher than with sigmoid neurons, the learning rate parameter of the gradient descent is adjusted to obtain comparable learning iteration numbers.

Application to machine learning tasks

In this section, I simulate more comprehensive neural networks using the oscillator-based neuron. First, a simple single-layer neural network containing three neurons is trained by gradient descent (or equivalently the perceptron learning rule for the threshold neurons) on the standard Iris classification dataset.

The evolutions of the test set classification rates of the three classes during learning are

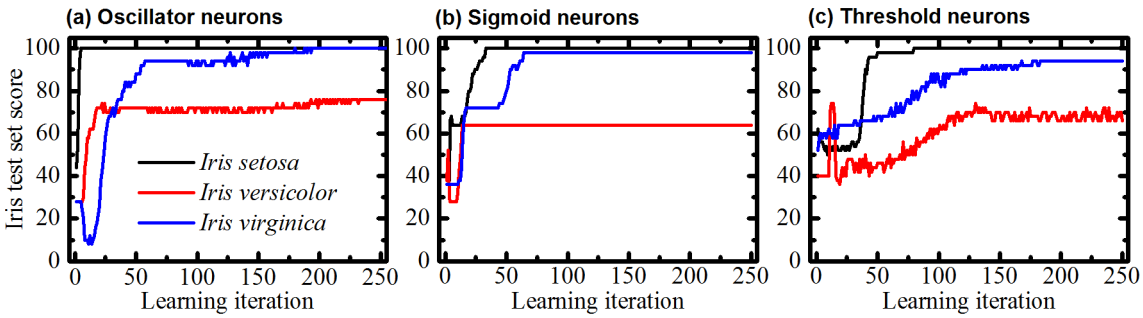


Figure 4.16: Iris test set classification rate evolutions during learning for the three classes using **(a)** oscillator-based neurons, **(b)** classical sigmoid neurons **(c)** threshold neurons.

plotted in Fig. 4.16(a), and compared to a sigmoid layer (Fig. 4.16(b)) and to a threshold layer (Fig. 4.16(c)). The results show that oscillator-based neurons converge to similar recognition rates for classes *Iris setosa* (100%), *Iris virginica* (≥ 95%) and to slightly increased recognition rates for the non-linearly separable class *Iris versicolor* (76% using the oscillator neuron, ≥ 65% using a sigmoid or a threshold neuron). This result confirms that the proposed neuron successfully operates in a neural network.

I then train a more complex two-layer (300 hidden, 10 output) neural network to classify the handwritten digits of the MNIST database, using the back-propagation algorithm. The final test set classification rate of 95.7% is on par with classical sigmoid networks (95.3% [209],), which validates the effectiveness of the proposed neuron in multi-layer neural networks.

Finally, Fig. 4.17 assesses the effects of oscillator noise by showing the MNIST test set classification rate of the fully simulated noisy oscillator system, as a function of the oscillator linewidth. The classification rate stays over 84% for a linewidth of 100kHz, a noise value typical of higher quality spin-torque nano-oscillators [65], which suggests that the proposed approach could be implemented using such nano-devices.

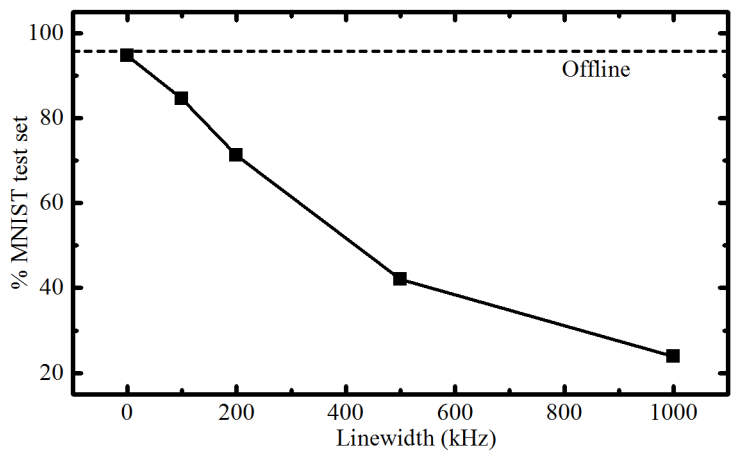


Figure 4.17: MNIST test set classification rate as a function of the oscillator linewidth.

These results, being published as [210], validate through simulation the use of the oscillator-based neuron I proposed for machine learning. It appears to be a valid functional alternative to sigmoid or threshold neurons, and naturally computes its parabola-shaped transfer function through its intrinsic dynamics.

However, the scheme appears to be more sensitive to noise than the approaches based on synchronization detection. This is in part due to the noise sensitivity of envelope detection, but mainly to the gradient-based offline learning method that can not take noise into account to reject noise-sensitive solutions.

EEG-like signal readout in the extended oscillator-based classifier

In this part, I come back to the extended oscillator-based classification architecture. Instead of reading whether or not selected oscillators of the extended classifier are synchronized as done in Section 4.2.6, I study the possibility of using the EEG-like signal S as a readout of the global synchronization state of the network as presented in Section 2.5.2. This signal can then be thresholded to perform binary classification.

Definition of the classification architecture with EEG-like readout

Figure 4.18 shows a schematic of the global classification architecture with EEG-like readout. The underlying architecture is the same as for the extended oscillator-based classifier, with a tunable weight matrix and a uniformly coupled oscillator network (my simulations use k = 7MHz). Only the readout method differs. It consists in taking the envelope S of the mean signal of the oscillators (or any signal proportional to it), rescaling it to obtain the analog output $a = \alpha^a + \beta^a S$, and thresholding it to output a binary value $a > 0.5$.

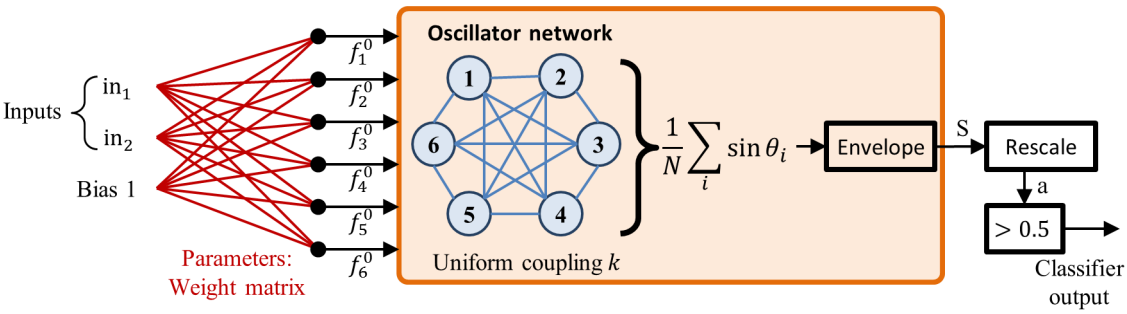


Figure 4.18: Schematic of the extended oscillator-based classifier with EEG-like readout.

The rescaling operation is necessary to bring the output between 0 and 1 independently of the scale of the EEG-like signal S which can change substantially as shown in figures 2.19(b,c,d). This also adds flexibility by allowing the readout of any affine function of S, and not necessary the mean signal itself.

For simplicity, the envelope detection time constant was chosen to be $\tau_{\text{EEG}} = 0$, which is

equivalent to measuring the mean value of S. This choice was made as the study in Section 2.5.2 shows that $\tau_{\text{EEG}} = 0$ provides relatively regular response maps without over-representation of higher order dynamics. Moreover, slightly higher time constants mainly improve the readout range (Figure 2.19(c)), which is already corrected by the rescaling operation.

Just like the two-oscillator neuron, this approach is convenient to implement in hardware with nano-oscillators and standard components. It is therefore an attractive alternative to synchronization detection in the case of binary classification.

Bio-inspired learning using genetic algorithms

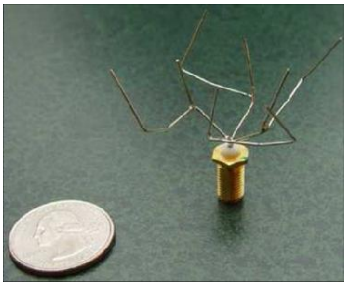
Such an EEG-like readout requires rethinking the learning process of the architecture. My attempts to compute the gradient of this architecture have failed, in part because the rectification operation pulls the derivative to zero when the sum signal is negative, and more importantly because of the the system response itself. As explained in Section 2.5.2, the synchronization regions show a relatively flat response but still contain local bumps due to higher order phenomena, leading to incorrect numerical derivative estimation.

As a result, I focus on "black box" optimization techniques that do not require any information on the model. More specifically, genetic algorithms [200] leverage the evolutionary principles of mutations, sexual reproduction and selection that led to the current efficiency of complex lifeforms on earth in order to perform model-free optimization.

Genetic algorithms

The basic principle of genetic algorithms is to repeatedly apply random changes and combinations to the system parameters and select the best performing solutions. More formally, genetic algorithms seek to minimize the error function of a system defined by a genome, which in the simplest case corresponds to a vector of its parameters, and only require evaluating its performance (fitness) on the target task. If only the outputs of the system are used for evaluation, no information on the internal variables are required and genetic algorithms could also run online.

These algorithms are used whenever gradient and reinforcement based approaches can not be applied and have been used with success to solve complex problems such as defining antenna shapes with special constraints [211], or school timetabling [212]. Genetic algorithms have also recently become an important research topic in neural network optimization [213].



Evolved antenna for space applications. Credit: NASA

In the case of the oscillator-based classifier with EEG-like readout, a genome g corresponds to a given weight matrix, and I define its fitness F_g as an inverse function of its

mean quadratic error E_g on a classification training set.

The genetic algorithm optimization process I used executes the following steps:

1. **Initialization:** create a generation of N_{pop} random genomes (weight matrices) $\{W_g\}_{g\in[1;N_{\text{pop}}]}$.
2. **Evaluation:** run each genome W_g on the classification task to evaluate its fitness F_g.
3. **Selection:**
   (a) Keep N_{survive} good genomes intact for the next generation:
   select N_{survive} genomes at random with probabilities proportional to their fitnesses.
   (b) Select couples for sexual reproduction:
   select $N_{\text{pop}} - N_{\text{survive}}$ pairs of genomes, with probabilities proportional to individual fitnesses.
4. **Sexual reproduction and mutations:** for every selected couple:
   (a) Produce a "child" genome by combining the genomes of the parent couple
   (b) Apply random mutations to the children genomes
5. Replace the current generation with a new one composed of the N_{survive} genomes that survived the previous generation, and the $N_{\text{pop}} - N_{\text{survive}}$ newly produced children genomes.
6. Go back to step 2. to process the new generation.

This process increases the fitness of the population in time through selection, but also explores new solutions with mutations and sexual reproduction. This general approach is relatively standard in genetic algorithms, but the details of each implementation vary substantially. In this work, I have designed a specific implementation that fits the specificities of the oscillator-based classifier. The details of the algorithm are provided in Appendix A.

Evolving binary classification on a simple task

The behavior of the EEG-like readout is linked but nonetheless substantially different from synchronization detection, and their associated learning algorithms are fundamentally different. It is therefore necessary to evaluate the capabilities of this alternative architecture on a standard classification task.

To achieve this comparison, I start by performing a simple two-dimensional binary classification task with 6 oscillators: classify points in a centered circular region as 1, and the other points as 0. Figure 4.19(a) shows the target response map with this region. The training set on which evolution is performed consists of 1000 randomly chosen points different from the map pixels, and the test set consists of the 100×100 pixels of the response map.

Figure 4.19(b) plots the test set classification rate of the best genome (the best-performing genome on the training set), and the mean test set classification rate of the population as functions of the generation number. After about 40 generations, the mean classification rate of the population reaches 95% and its best genome crosses 99%.

The snapshots in figures 4.19(c,d,e) show the rescaled output a map of the best genome of generations 1, 17 and 93 respectively. In the desaturated areas $a < 0.5$ the network classifies the

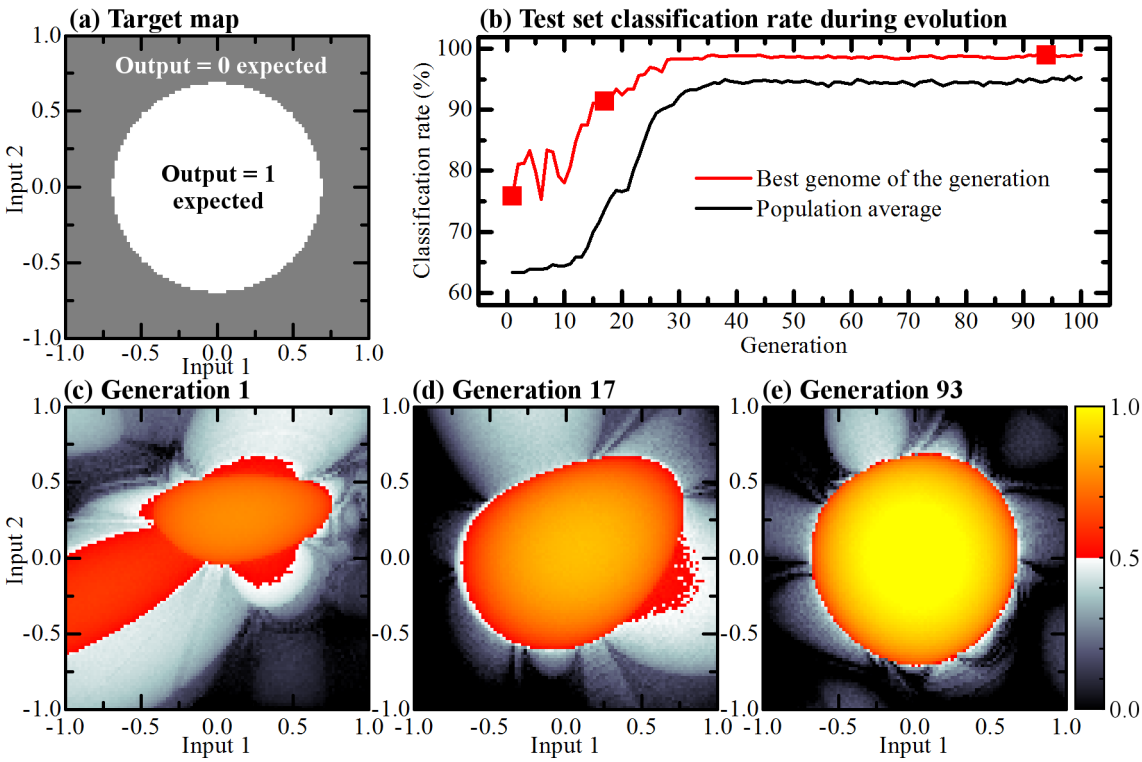


Figure 4.19: Evolving the weight matrix of the extended oscillator-based classifier with EEG-like readout to classify a simple two-dimensional circular region. **(a)** Target response map defining the classification task. **(b)** Test set successful classification rate as a function of the generation number, showing the score of the best genome of the population, as well as the average score of the population. **(c,d,e)** Snapshot maps of the a output of the best genome in the population at generations 1, 17 and 93 respectively. Desaturated areas are classified as 0, and warm-colored areas are classified as 1.

example as class 0, while in warm colored areas $a > 0.5$, the output class is 1. At generation 1, the response is defined by the random initial conditions. After 17 generations, a central blob forms, but the classification is still approximative. At generation 93, the target region is clearly identified with outputs approaching 1.0, while the output is inhibited in class 0 areas and stays close to 0. The classification is therefore successful.

However, contrary to gradient descent which only explores promising directions in the parameter space at each iteration, this genetic algorithm is not model-aware. It relies on random trials, has no memory of previous failures, and therefore mostly explores non-promising directions in the parameter space. This implies a high computational overhead, and the algorithm is very demanding: with a population of N_{pop} genomes and a training set of M_{train} examples, one generation requires $N_{\text{pop}} \times M_{\text{train}}$ offline simulations or online physical executions per generation (here 200,000), most of which do not lead to improvements.

These results show that the genetic algorithm achieves learning and successfully solves this

simple task, at the expense of a high resource overhead.

Parallelization

As they are independent, I run all train and test set simulations for all genomes in parallel on GPU using a custom C++ code, and gather the results at the end of each generation. This represents 2,200,000 parallel simulations per generation.

Evolution on a more complex task, and impact of noise

The next step to characterize the capabilities of the oscillator-based classifier with EEG-like readout is to focus on a more complex task, and evaluate the impact of oscillator phase noise. Note that contrary to gradient-based algorithms that do not support phase noise during learning, genetic algorithms are not sensitive to it, and noise can be applied during the learning process itself.

To perform this analysis, I evolve the classifier for another binary classification task: the two-dimensional target map shown in Figure 4.20(a). The setup is the same as for the previous task but contrary to the simple centered circular region, this target region is concave and not diagonal-symmetric. The learning is performed both without noise, and with an oscillator linewidth of FWHM = 1MHz.

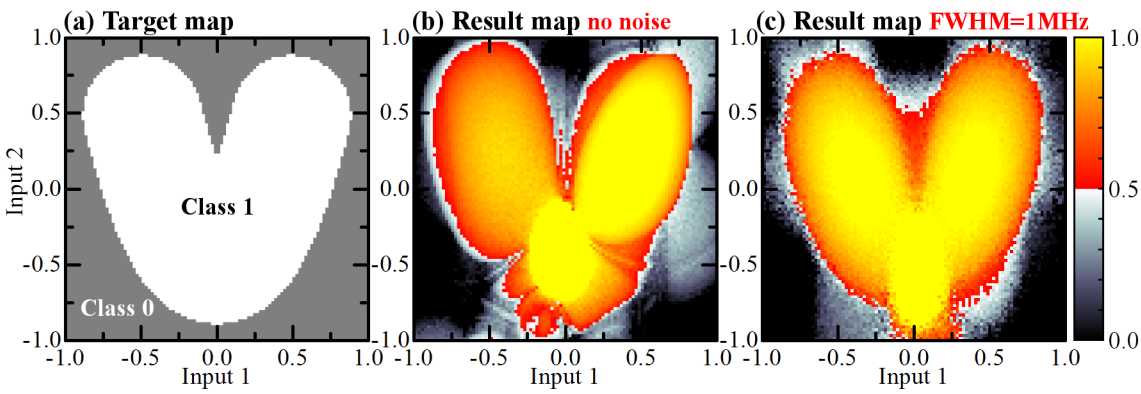


Figure 4.20: Evolved complex two-dimensional classification using the extended oscillator-based classifier with EEG-like readout. **(a)** Target map. **(b)** Map of the rescaled output a of the best genome after 100 generations, without noise. **(c)** Response map of the best genome after evolution during 100 generations with oscillator linedwidths FWHM = 1MHz.

Figure 4.20(b) shows the map of the rescaled output a of the best genome after 100 generations of evolution without phase noise. Its classification rate exceeds 96% which confirms that the classifier quickly solves this task.

Figure 4.20(c) shows the *a* output map of the best genome after 100 generations of evolution in the presence of noise levels consistent with spin torque nano-oscillators. This response map appears more granular and the limits of the classified areas are more blurry. However, the test set classification rate still reaches 95%. This result shows the high resilience of this classifier to realistic phase noise levels. This can be attributed to the presence of noise during the evolution process which eliminates noise-sensitive solutions and favors robust ones, leading to noise robustness being deeply imprinted in the genomes.

Results on the Iris dataset

In the previous subsection, I have shown that the extended oscillator-based classifier with EEG-like readout evolved using a genetic algorithm can solve complex two-dimensional tasks. In order to compare its classification capabilities to the case with synchronization detection and gradient descent, I perform training on the Iris dataset with 6 oscillators for 100 generations.

Target class	**Iris test set classification rates after evolution**		
	No noise	**FWHM = 1MHz**	**FWHM = 10MHz**
Iris setosa	100%	100%	100%
Iris versicolor	98.3%	98.3%	98.3%
Iris virginica	100%	100%	100%

Table 4.2: Iris test set classification rates using the extended oscillator-based classifier with EEG-like outputs for the best genome after 100 generations.

Table 4.2 shows the test set classification rates for the identification of each target class without noise, with FWHM = 1MHz and with FWHM = 10MHz.

Iris setosa and *Iris virginica* are successfully classified 100% of the time, and the non-separable *Iris versicolor* is successfully classified 98.3% of the time. A remarkable observation is that these classification rates stay constant even for extreme noise levels (FWHM = 10MHz). My tests show that the classification rates only start to drop for FWHM > 30MHz.

This result shows that the classification capabilities of this approach are very similar to the case with synchronization detection and gradient descent, but appear more resilient to noise because of the presence of phase noise during learning, which ensures the compatibility of the architecture with a broader range of nano-oscillators. The associated genetic algorithm learning process, however, is substantially slower and computationally demanding.

As a result, the choice between the gradient-optimized classifier and the genetically optimized EEG-like classifier mainly depends on the available learning time and computational power.

Remarks and improvements on the architecture

The simulation results on the extended oscillator-based classifier with EEG-like readout and genetic algorithm learning show promising results on classification tasks. The learning pro-

cess can run both offline and in a hardware implementation but requires a long and resource-hungry learning process.

Moreover, when the genetic algorithm is used in conjunction with a realistic device model, or on a physical implementation of the architecture, it is capable of taking into account every aspect of the system dynamics. As a consequence, variabiliy, noise, non-linearities or higher order dynamics could all be transparently leveraged by the learning algorithm as computational assets.

However, such an architecture can only perform binary classification. A solution to that limitation is to train as many networks as there are classes, each of them identifying one of the classes. If the weight matrix is implemented using a technology that allows rapid rewriting, each of the matrices trained for each target class can also be applied sequentially to discriminate the classes one after the other using the same oscillator network.

Summary of the architectures and offline learning methods

This section summarizes the different architectures and associated offline learning algorithms proposed in this thesis, highlighting their advantages and shortcomings.

Oscillator-based classifier

Readout: **synchronization detection** Learning: **gradient descent**

N_c all-to-all uniformly coupled core oscillators and N_i input oscillators uniformly coupled to all core oscillators. A N_i-dimensional input pattern is presented by setting the input oscillator's natural frequencies, inducing synchronizations in the core network. Synchronization detection is performed on an ensemble of "output" pairs of core oscillators and constitutes the output. The learning process uses iterative gradient-based optimization of all core natural frequencies on a simulated model of the system so that its response closely matches the expected one on the presented examples. The learned natural frequencies can then be transferred into a physical system for inference.

- **+** Uses a standard machine learning framework
- **+** Indirect oscillator interactions fully taken into account
- **−** Offline learning only
- **−** Phase noise not taken into account during learning
- **−** Can only learn simple diagonal-symmetric patterns

Extended oscillator-based classifier

Readout: **synchronization detection** Learning: **gradient descent**

Architecture similar to the previous one but all oscillators are identical (no more input/core difference) and uniformly all-to-all coupled. Their natural frequencies are linear combinations of the components of the presented input and a constant bias. This linear combination is produced by a weight matrix that is tuned by a gradient-based learning algorithm so that the response of the system closely matches the expected one on the presented examples.

- **+** Uses a standard machine learning framework
- **+** Indirect oscillator interactions fully taken into account
- **+** Can discriminate complex arbitrary patterns
- **−** Offline learning only
- **−** Phase noise not taken into account during learning

Oscillator pair neural network

Readout: **EEG-like signal** Learning: **gradient descent**

A neural network where each neuron is a synchronized pair of oscillators, its input shifts the natural frequency of one of its oscillators, and its output is the EEG-like signal rescaled. The activation function of the neuron can be expressed analytically and is differentiable. It is used for offline learning with classical neural network gradient-based techniques optimizing the weight matrices connecting the neurons.

+ Straightforward to implement in existing neural network frameworks
+ Can discriminate complex arbitrary patterns
+ Only requires coupling pairs of oscillators, which is convenient for experiments
− Requires a coupling and a readout circuit for each neuron
− Offline learning only
− Phase noise not taken into account during learning
− High sensitivity to noise

Extended oscillator-based classifier

Readout: **EEG-like signal** Learning: **genetic algorithm**

Architecture similar to the extended oscillator-based classifier with synchronization detection, but the readout consists in measuring the EEG-like signal (the envelope of the mean signal of all the oscillators), then rescaling and thresholding it to perform binary-only classification. As gradient-based learning algorithms do not perform well on this architecture, I use a genetic model-free algorithm to optimize the weight matrix.

+ Online learning capabilities
+ Indirect oscillator interactions fully taken into account
+ All system dynamics implicitly taken into account
+ High noise resilience after learning in the presence of noise
+ Can discriminate complex arbitrary patterns
− Binary classification only
− Long and resource-demanding model-free learning process

Towards full online learning

Online learning algorithms operate directly on the oscillator network. Examples are presented as inputs to the network, its outputs are compared to the expected ones, and its parameters are adjusted accordingly. In the case of the oscillator-based pattern recognition architecture described in Chapter 2, this means that the learning algorithm only has access to measurable quantities such as the synchronization states of the oscillators. Such algorithms would allow these classifiers to adapt to new or changing data without having to go through offline learning each time, which makes them compelling for a wider range of applications. However, designing a fully online learning approach without resorting to time-consuming model-free algorithms is challenging.

A simple online learning method was sketched in the seminal work of Vassilieva *et. al* [160]. It consists in presenting known examples and adjusting core oscillator natural frequencies to promote expected synchronizations or break unexpected ones and obtain the right output for the presented example. This method can be summarized by the following algorithm:

1. Present a known input example to the oscillator network
2. Wait for stabilization
3. For all core oscillator pairs (i, j):
   (a) Read the synchronization state of (i, j)
   (b) If the pair (i, j) is synchronized but is not expected to be for this example:
   $\Rightarrow$ Adjust the natural frequencies f_i^0 and f_j^0 slightly further apart
   (c) If the pair (i, j) is not synchronized but is expected to be for this example:
   $\Rightarrow$ Adjust the natural frequencies f_i^0 and f_j^0 slightly closer to each other
4. Repeat the process with another example

Running this algorithm effectively corresponds to regulating the synchronizations so that they match the expected ones for the training examples. A major advantage of this algorithm is that it is not heavily sensitive to mismatches between the actual system and its model. This makes it robust to variability, extra non-linearities or higher order phenomena. The algorithm operates as long as it can bring natural frequencies closer to induce synchronizations or push them further apart to break synchronizations.

However, this approach only tunes the oscillators used as outputs. Whenever the dimensionality of the classifier output is lower than the number of pairs of core oscillators, this algorithm does not adjust the natural frequencies of the oscillators that are not part of an output pair, as it does not fully evaluate indirect interactions between oscillators. As an example, it can therefore not learn to perform advanced binary classification tasks requiring more than two core oscillators, contrary to the offline learning methods presented in this Chapter. Moreover, such a linear regulation of synchronizations is very sensitive to local minima and the learning process can easily get stuck in suboptimal solutions.

As a result, this algorithm is an important step towards full online learning in oscillator-based classifiers, and is robust to device non-idealities. However, it is still limited to simple classification tasks.

Nevertheless, in the context of the MEMOS project, Julie Grollier's team at CNRS/Thales is working on an experimental implementation of a classifier using spin transfer torque nano-oscillators, similar to the one introduced in Chapter 2, and using an online learning algorithm similar to the one presented in this section, in order to perform spoken vowel classification. Fruitful discussions between our teams have led to a better understanding of the system and its capabilities.

Figure 4.21 shows the architecture used for this project. It consists of a core set of four electrically coupled vortex spin transfer torque nano-oscillators perturbed by antenna-fed radio-frequency signals A and B with respective frequencies f_A and f_B. A spectrum analyzer is used

to measure the average frequencies of the oscillators in the network and spectrograms without and with input signals are also presented in the figure. They show spikes corresponding to oscillators 1,2,3 and 4. After input B is activated, the spectrogram shows that oscillator 4 phase locks to it. The architecture is therefore similar to the one introduced in Chapter 2 but involves only unidirectional coupling from the inputs to the core oscillators.

The experiments have already been performed, and the processing of their results is currently being finalized. They show that the system can be trained to perform a task of spoken vowel classification. These exciting results constitute an important and promising first step towards the implementation of online learning algorithms for oscillator-based classifiers.

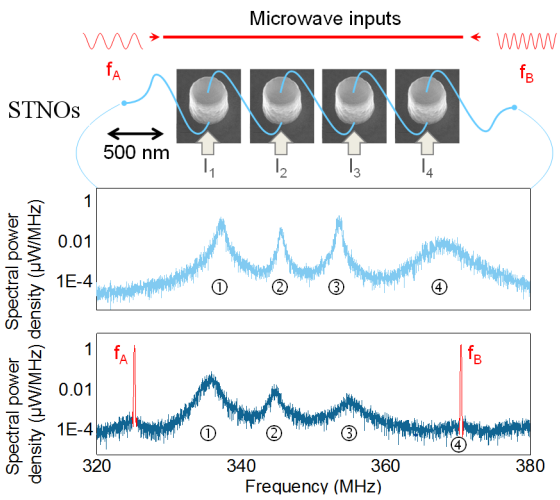


Figure 4.21: Schematic of the experimental classification architecture with STNOs by Julie Grollier's team, showing four electrically coupled vortex STNOs with antenna RF inputs A and B. Mean field signal spectra are shown without input signals, and with input signals respectively.

Summary of the Chapter

In this Chapter, I proposed an offline learning algorithm allowing the oscillator-based classifier architecture defined in Chapter 2 to solve arbitrary classification tasks after training on a set of known examples. This algorithm adjusts the natural frequencies of the core oscillators in order to minimize the classification error on the known examples using gradient descent. After reinterpreting the oscillator-based classifier and its associated learning algorithm in a machine learning context, I was able to perform efficient offline learning of standard classification tasks using the TensorFlow machine learning library. After preliminary results revealed the limitations of the initial architecture, I proposed an extended version of the oscillator-based classifier that corrects these shortcomings. The resulting extended oscillator-based classifier showed competitive classification rates on standard classification tasks.

I also proposed an artificial neuron made of synchronized pairs of oscillators, and derived

its analytical transfer function. This allowed me to assemble these neurons into an artificial feed-forward neural network, and perform offline machine learning using TensorFlow. Depending on the task, the resulting oscillator-based neural networks showed better or similar performance to classical neural networks of the same size. However, this architecture appears to be more noise-sensitive than the oscillator-based classifier.

I proposed a genetic algorithm to perform learning with the alternative "EEG-like" architecture introduced in Section 2.5.2. This model-free, biologically inspired algorithm relies on random mutations, sexual reproduction and fitness-based selection principles in order to optimize the parameters of the classifier and perform successful classification of training examples. On standard classification tasks, this approach achieves classification rates similar to the gradient-optimized extended oscillator-based classifier with synchronization readout while showing higher noise resilience, but requires substantially more computational power.

Finally, I presented leads towards online learning approaches, including a first experimental realization of a learning system using spin torque oscillators by our collaborators of the CNRS/Thales laboratory.

Chapter 5

Random number generation and computing with stochastic oscillators

Everything we care about lies somewhere in the middle, where pattern and randomness interlace.

James GLEICK

"THE FIFTH AND LAST CHAPTER *of this thesis focuses on a fundamentally different approach: using the intrinsically random behavior of superparamagnetic tunnel junctions to achieve efficient random bitstream generation applied to stochastic computing.* "

THIS CHAPTER presents a different kind of device than the ones envisioned in Chapters 2 to 4: intrinsically stochastic spintronic nano-oscillators called superparamagnetic tunnel junctions. As their behavior is purely stochastic, such oscillators can not be used to implement the computing schemes presented in the previous chapters, contrary to harmonic or relaxation oscillators. However, their random behavior can be useful for efficient random number generation, especially for emerging alternative computing schemes that require massive numbers of low energy random numbers [214]. This Chapter studies this approach by focusing on:

1. the experimental measurement of superparamagnetic tunnel junctions for long durations,
2. the design of a whitening scheme that converts the signals obtained from these devices into an unbiased, uncorrelated bitstreams, as well as the validation of the resulting random bitstream quality using a standardized statistical test suite,
3. the design and circuit simulation of a full-featured random number generator using this whitening approach, and characterization of its energy and area requirements, as well as the influence of environmental perturbations,
4. validating the random bitstream generator using the experimental bitstreams on an example probabilistic classification task consisting in classifying e-mail messages as spam or not spam.

These results highlight an alternative path where oscillators can allow novel forms of computation.

Introduction

Many emerging alternative computing approaches, such as stochastic computing [215–219] and some brain-inspired (neuromorphic) schemes [206, 220, 221], require a large quantity of random numbers. However, the circuit area and the energy required to generate these random numbers are major limitations of such computing schemes. For example, in the popular neuromorphic TrueNorth system [206], one third of the neuron area is dedicated to perform random number generation. Indeed, one million random bits are required, at each integration step of the system. More concerning, in stochastic computing architectures, which consist in representing and manipulating values as probabilistic bitstreams, random number generation is typically the dominant source of energy consumption, as the logic performed using the random bits is generally quite simple and efficient by principle. Many practical stochastic computing schemes therefore try to limit their reliance on expensive independent random bits using various techniques, including the sharing or reuse of random bits [217, 222, 223]. However, such tricks limit the capabilities of stochastic computing to small tasks, as they introduce

correlations between signals.

Most of the aforementioned unconventional computing circuits use pseudo-random number generators, which generate bitstreams that appear random using fully deterministic iterative bit scrambling operations. But these either lead to low quality random numbers or are highly energy and area-consuming. A preferable solution would be to rely on “true” random number generators that generate random bits based on physical phenomena that are intrinsically random. However, this is also difficult to realize with minimal energy consumption. This difficulty is due to the fact that most true random number generators function by triggering events whose outcome is intrinsically random. Triggering these events comes with a non-negligible energy cost. The most energy-efficient example uses a bistable CMOS circuit forced into in a meta-stable state which then randomly falls into one of the two stable states, generating one random bit [224]. It consumes 3pJ/bit and a circuit area of $4000\mu m^2$.

In order to reduce this large area footprint, recent proposals suggest to leverage the inherent stochastic programming properties that arise in many of the bi-stable nano-devices developed for memory applications [225]. This approach was investigated with oxide-based resistive memory devices [226–229], phase-change memory devices [230, 231], magnetic memory devices [232–234], as well as with straintronic memory devices [235]. However, these approaches are based on repeated, energy-intensive programming operations, and still require high energy for random bit generation. For instance, it requires dozens of pJ/bit to induce a stochastic switch of magnetization in magnetic tunnel junctions with two stable states, as proposed in the "Spin-Dice" concept [232], due to the high energy barrier between the magnetic states. Optimized schemes have been proposed [236–238], predicting further reduction of the energy cost per bit, but are still bounded by the need of a costly perturb operation. While proposing high quality random number with high throughput, such strategies are not fit for emerging neuro-inspired computing applications like stochastic computing architectures.

A more natural approach would be to extract random numbers directly from thermal noise, as it provides randomness at no energy cost. Unfortunately, this approach requires large circuits to amplify thermal noise into a large signal of random bits, and has never been shown to be more energy efficient than the first approach until now. The lowest energy solution today is to use jitter as a way to efficiently amplify the noise present in CMOS ring oscillators (described in Section 1.3.2). The most energy efficient implementation [239] requires $23pJ/bit$ and $375\mu m^2$.

In the present work, we propose the use of a stochastic oscillating device that intrinsically amplifies thermal noise without external energy supply: superparamagnetic tunnel junctions. These bi-stable magnetic tunnel junctions rely on a magnetic stack similar to the magnetic nano-oscillators introduced in Section 1.3.5, and are reminiscent of the junctions used for Magnetic Random Access Memories (MRAMs) [104]. However, contrarily to MRAM cells, the energy barrier between the two magnetic states is very low, and thermal fluctuations induce repeated and stochastic magnetization switching between the two states at room temperature. There-

fore, no write operations are required and a low-energy readout of the device state naturally produces random bits. In this Chapter, we show that these devices permit the generation of high quality random numbers at 20fJ/bit using less than $2\mu m^2$, which is orders of magnitude more efficient in terms of energy and area than current solutions.

We first show experimentally that superparamagnetic tunnel junctions allow the generation of high-quality random bits with minimal readout circuitry and that their behavior can be predicted by existing physical models. We then use the model to investigate the influence of device scaling and environmental factors on random bit quality and speed. Circuit simulation enables an estimation of the energy efficiency of random bit generation. Finally, we demonstrate the potential of these devices for unconventional computing through the example task of email messages classification using random bits extracted from the experimental data, and show that they are particularly adapted to computing schemes trading off speed for ultra low energy consumption.

Exploiting superparamagnetic tunnel junctions for random bitstream generation

The stochastic behavior of superparamagnetic tunnel junctions

Superparamagnetic tunnel junctions are bistable spintronic nanodevices composed of a high stability pinned nanomagnet and a low-stability "free" nanomagnet, separated by a tunnel oxide layer (Fig. 5.1(a)). Their structure is highly similar to the magnetic tunnel junctions used as the basic cells of MRAMs. The devices I measured experimentally, as part of our collaboration with Julie Grollier's group at CNRS/Thales, were fabricated by A. Fukushima, K. Yakushiji, H. Kubota and S. Yuasa at AIST Japan using sputtering, following a standard magnetic tunnel junction process, and with the CMOS-compatible stack detailed in Fig. 5.1(b). E-beam lithography patterning was then performed to produce $50 \times 150 nm^2$ elliptic pillars.

The free magnet has two stable states, parallel (P) and antiparallel (AP) relatively to the pinned layer (Fig. 5.1(c)). Through the tunnel magneto-resistance effect [106], the electrical resistance of the junction in the AP state R_{AP} is higher than the resistance in the P state R_P. This effect is traditionally measured through the TMR coefficient defined by $R_{AP}/R_P = 1 + \text{TMR}$.

The lateral dimensions of the device are chosen so that the effective energy barrier between the two stable states is not very high compared to $k_B T$. Unlike the case of MRAMs, for which the magnetization direction of the free magnet is highly stable and can only be switched by proper external action, the magnetization direction of the superparamagnetic free magnet spontaneously switches between its two stable states, due to low stability relative to thermal fluctuations (Fig. 5.1(c)) [240, 241]. Here, no bias or perturb scheme is required to provoke these random fluctuations, but only temperature.

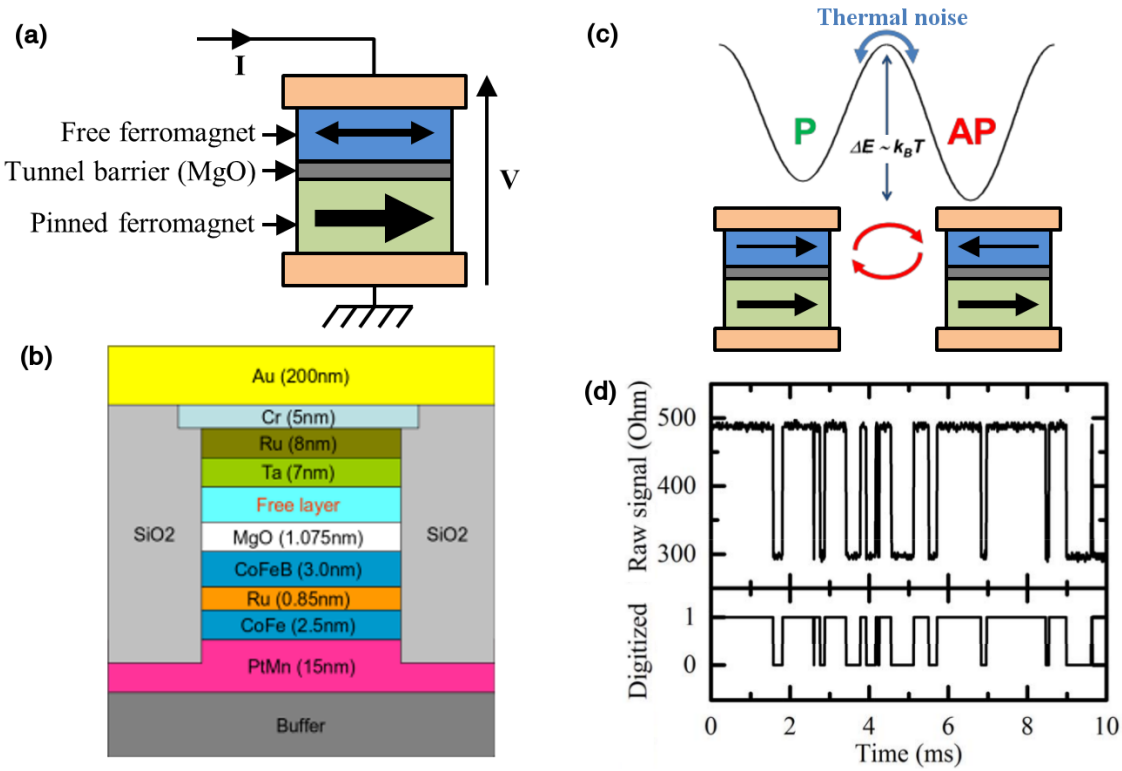


Figure 5.1: Structure and behavior of superparamagnetic tunnel junctions. **(a)** Basic structure of the measured superparamagnetic tunnel junctions and readout setup. **(b)** Detailed stack of the junctions. **(c)** Representation of the two stable magnetic states, and the associated energy barrier. **(d)** Experimental resistance trace and thresholding operation.

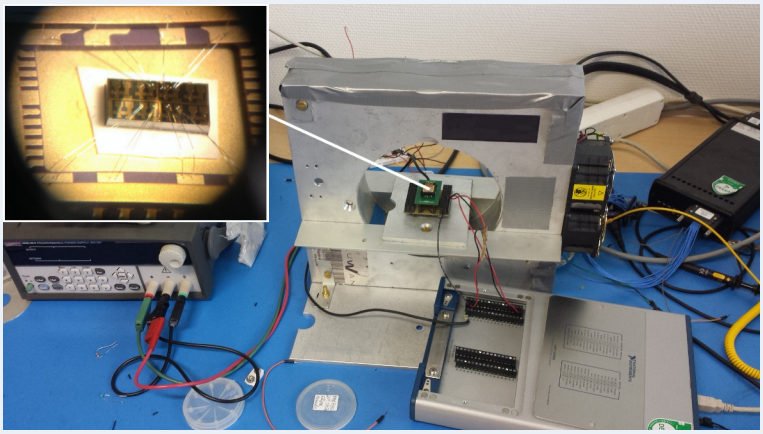
Resistance versus time measurements were done on junctions by applying a low 10 µA constant current through the junction. Such a small current amplitude was chosen to have negligible influence on the magnetic behavior of the device [242] and to maximize its lifetime while providing a clear signal. Fig. 5.1(d) shows a sample from the time evolution of the electrical resistance of a junction measured at room temperature, as well as a binarized version, obtained by thresholding. We see that the resistance follows two-state fluctuations analogous to a random telegraph signal. The mean frequency of fluctuations is strongly related to the shape and material properties of the junction [243].

Experimental setup

The current is applied, and the voltage measured using a NI USB 6351 source and measurement unit connected to a PC running a custom C++ multi-thread high speed data acquisition, buffering, processing and storage software I wrote for this purpose.

The dies containing the superparamagnetic tunnel junctions were glued an wirebonded by Miguel Romera, Alice Mizrahi and myself to PLCC 32-pin chip carriers. The carriers are held by a PLCC socket on a sample holder circuit board I built to provide 4-terminal sensing that minimizes noise and improves measurement precision when the applied currents are low, as well as over-voltage protection. The junctions are highly sensitive and voltages over about 200 mV can break their tunnel barrier. This has motivated the choice of MMBD4148SE fast small signal diodes for protection, as well as proper grounding against electrostatic discharges.

In order to apply magnetic fields, I built a fan-cooled 10 mT electromagnet that carries the sample holder board at the center of its coil. The electromagnet is powered by a computer-controllable current source also driven by the C++ software running on the PC. The current-to-field calibration was done manually using a teslameter.



Sample holder board inside the electromagnet connected to the NI USB 6351 source and measurement unit (bottom right). Zoom on the chip with 4 wire-bonded superparamagnetic tunnel junctions.

Fig. 5.2(a) shows the histograms of the dwell times in the '1' (AP) and '0' (P) states, obtained through measurement of a superparamagnetic tunnel junction over a 10 second period. We see that these histograms can be fitted by an exponential law, which is characteristic of a Poisson process. Fig. 5.2(b) presents the power spectrum density of the same signal, superimposed with the expected power spectrum density of a random telegraph signal based on a Poisson process. Excellent agreement between the measured results and the hypothesis of a Poisson process is seen.

Random bits can be extracted by sampling the voltage across the device at a constant fre-

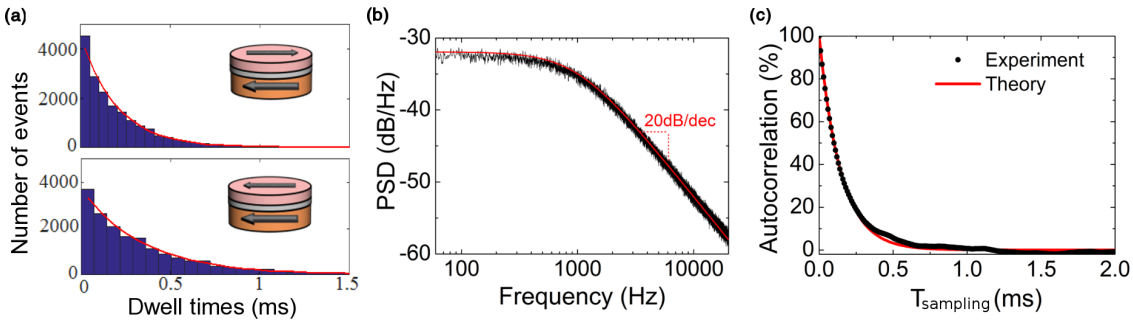


Figure 5.2: Statistics of the experimental superparamagnetic tunnel junction signal. **(a)** Experimental histograms of the dwell times in Anti-parallel (AP, top Figure, high resistance) and Parallel (P, bottom Figure, low resistance) states, for a superparamagnetic magnetic tunnel junction measured during $10s$. **(b)** Experimental power spectrum density of the resistance signal. **(c)** Autocorrelation of the experimental resistance signal as a function of signal sampling period.

quency. The voltage was initially sampled at 100kHz, and bitstreams with slower sampling rates were obtained by subsampling the initial bitstream. To evaluate the quality of the obtained random bits, the device was measured for over 2.5 days, producing 21.2 gigabits. No external magnetic field was applied during the measurement.

Optimizing the quality of random bits

The sampling frequency needs to be chosen carefully relative to the mean switching frequency of the junction, defined as $F_{\mathrm{MTJ}} = 1/(\tau_1 + \tau_0)$, where τ_1 and τ_0 are the mean dwell times in state 1 and 0, respectively. F_{MTJ} was measured to be 1.66kHz ($\tau_1 + \tau_0 \approx 604\mu s$). Fig. 5.2(c) presents the correlation of consecutive bits extracted at different sampling rates. This result is superimposed with the one theoretically expected from a Poisson process. At high sampling frequency, subsequent bits are naturally autocorrelated (at $F_{\mathrm{sampling}} = 100$kHz, correlation reaches 92.8%), and can therefore not be used for applications. This correlation decreases exponentially with the sampling period, which can therefore be chosen based on the correlation requirements on the random numbers.

As observed, in Fig. 5.2(a), the AP and P states possess an asymmetric stability: the device spends more time on average in the P state than in the AP state, which corresponds to a mean state (mean of the binarized signal) of 60.5%. This asymmetry can be connected to the stray field induced by the pinned magnet layer structure, which is present in all magnetic tunnel junctions [244]. This biasing field offsets the junction mean state from the ideal 50% value required for most applications, and is subject to device-to-device variations.

In order to eliminate this bias and any residual bit correlation, a “whitening” of the random bits is therefore required. To achieve this operation, we make use of a standard technique: combining several bitstreams into a single one using XOR gates. It can be shown (mathematical derivation available in Appendix B) that the auto-correlation after XOR whitening is the

product of the individual auto-correlations of the combined signals. It therefore decreases exponentially with the number of combined MTJ bitstreams, and is always lower than the auto-correlation of any of the combined signals. In the same way, the mean state of the whitened bitstream gets exponentially closer to 50% with the number of XOR-combined bitstreams, and stays always closer to perfect balance than any of the bitstreams being combined. As a reference, a more advanced but heavy stateful whitening technique ("Blum" [245]) was also applied to the raw measurements.

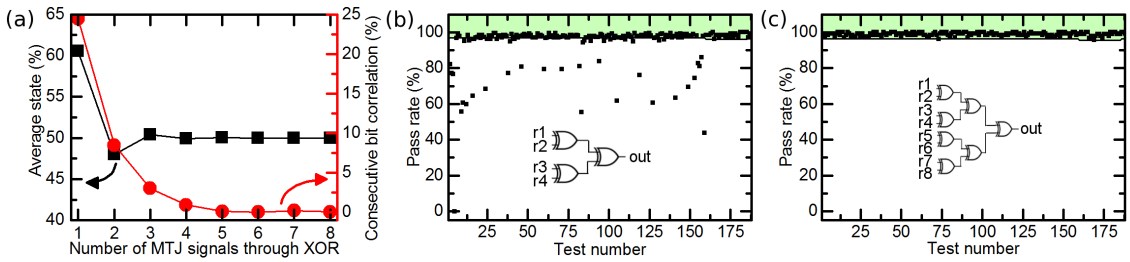


Figure 5.3: Whitened experimental random bitstream quality assessment. **(a)** Mean state and consecutive bit autocorrelation as functions of the number of independent superparamagnetic tunnel junction signals combined by XOR. NIST STS randomness quality test results on experimental data whitened by XOR4 **(b)** and XOR8 **(c)** at a F_{sampling} = 5kHz sampling frequency. When all test results are in the green area, the bitstream is consistent with cryptographic quality.

As an illustration, we consider bits extracted at a frequency of 5kHz. The bitstream was then divided into chunks of equal length which were used as independent signals and XOR-combined bit per bit for the XOR whitening process. We plot in Fig. 5.3(a) the consecutive bit correlation and the mean state of the whitened bitstream as functions of the number of signals combined by XOR. The correlation and the mean value bias decrease with the number of XOR-combined signals. With 4 bitstreams (XOR4), the resulting consecutive bit correlation drops under 1% and the mean value reaches 49.9%. For 8 bitstreams (XOR8), the auto-correlation is below 0.06% and the mean state reaches 50% with a standard deviation of 0.5%. These results suggest that XOR whitening can correct correlation and mean value issues.

However, in order to fully evaluate the quality of a whitened bitstream, signal autocorrelation and mean state are not sufficient metrics. We therefore used the standardized National Institute of Standards and Technology Statistical Test Suite (NIST STS) [246], which evaluates the quality of the random bitstream against 188 tests. The NIST STS computes the statistics of bitstreams, such as mean value, auto-correlation, standard deviation, estimated entropy or pattern occurrence frequencies, and checks weather they are consistent with perfect randomness. It also looks for the presence of repeated structures, linear dependencies, and other behaviors unexpected in a perfectly random bitstream.

To perform the NIST STS tests, the bitstream to be tested, measured during 2.5 days, is divided into 1 Mbits sequences. Each chunk is then tested independently, and the pass rate (percentage of one million bits sequences passing the test) was computed for each of the 188

F_{sampling}	$F_{\text{sampling}}/F_{\text{MTJ}}$	**Raw**	**XOR2**	**XOR4**	**XOR8**
100 kHz	60.4	0	10.1	10.1	10.1
20 kHz	12.1	0.5	0.5	10.6	12.2
9.1 kHz	5.5	1.1	10.6	10.6	88.3
5.9 kHz	3.6	1.1	1.1	16.5	100
5 kHz	3.0	1.1	1.1	72.9	100
1.9 kHz	1.1	1.1	14.4	97.9	100
0.9 kHz	0.54	1.1	14.4	98.4	100
0.7 kHz	0.42	1.1	16.0	97.9	100
0.5 kHz	0.30	1.1	16.0	98.4	100

Table 5.1: NIST Statistical Test Suite results for the whitened experimental random bitstream. Percentage of NIST STS tests satisfying cryptographic quality requirements for different numbers of combined bitstreams, and different sampling frequencies.

tests. Fig. 5.3(b) and (c) show the results for XOR4 and XOR8 whitened bitstreams respectively. For a bitstream to be consistent with cryptographic quality, the pass rates of all tests should lie in the green region [246], corresponding to the expected minimal pass rate provided by the NIST STS, dependent on the number of tested chunks. We can see that bits extracted by XOR8 whitening pass this requirement (this was also the case with the reference Blum technique), while with XOR4 whitening only a fraction of the tests are consistent with cryptographic quality of the random bits [1].

Table 5.1 presents more comprehensive results: the proportion of tests in the green area is given for XOR-whitened bitstreams at different sampling frequencies and numbers of XOR-combined bitstreams. The results confirm that the quality of the whitened bitstream increases for lower sampling frequencies (less correlation) and higher numbers of XOR-combined bitstreams (less correlation and lower bias). Higher numbers of XOR-combined bitstreams therefore allow further increasing the sampling rate while still passing all the NIST statistical tests, at the expense of more circuit area and energy consumption. XOR8 at $F_{\text{sampling}}/F_{\text{MTJ}} = 3.0$ appears to be an optimal choice, with 100% of the tests consistent with cryptographic quality and the highest sampling frequency.

Consistent results were observed on a second sample, measured during 1.5 days, producing 8.96 gigabits.

[1] The NIST tests also include a uniformity condition on the distribution of P-Values among tested sequences [246]. This condition was passed for all tests for sequences processed by Blum and XOR8.

Scaling capabilities in terms of speed and energy consumption

Device model and properties

Further studying the potential of superparamagnetic tunnel junctions for random number generation requires a realistic model of the device. In the literature, at low electric current, magnetic tunnel junctions switching is usually described by an Arrhenius-Néel two-states analysis, modeling a thermally activated magnetic switching [247]. The mean switching rates in each state are then described by:

$$\begin{cases} r_{0\to1} = 1/\tau_0 = f_0 \exp\left(-\frac{\Delta E_{0\to1}}{k_B T}\right) \\ r_{1\to0} = 1/\tau_1 = f_0 \exp\left(-\frac{\Delta E_{1\to0}}{k_B T}\right) \end{cases}, \tag{5.1}$$

where $f_0 = 1$GHz is the magnetic attempt frequency, $\Delta E_{0\to1}$ and $\Delta E_{1\to0}$ are the energy barriers associated with each transition (see Fig. 5.1(c)). Our experimental results suggesting that superparamagnetic tunnel junction switching is a Poisson process are consistent with this model.

The superparamagnetic tunnel junctions that we characterized experimentally in this study are slow devices. They can be used to generate random bits at kHz frequencies, sufficient for real-time brain-inspired systems like [206], but not for high performance applications. In our 50 × 150nm superparamagnetic tunnel junctions, we identified that the switching occurs through nucleation and propagation of a magnetic domain, probably seeded by fluctuations in a subset of grains within it [240] (by opposition to single domain magnetic reversal). The small effective switching volume of the free ferromagnet involved in the nucleation of the switching process explains the low energy barriers observed between the stable states, as compared to the full volume reversal energy barrier expected for those dimensions (evaluated to be approximately 126kT).

By contrast, recent experiments on perpendicular magnetic anisotropy (PMA) magnetic tunnel junctions have shown that aggressively scaled devices (diameters smaller than 35nm) switch at the scale of the whole volume [243]. Therefore, in the context of random number generators, extreme scaling of the nanodevices appears as providential, as smaller volumes and areas are directly linked to a lower magnetization stability of the free magnet [248], increasing random bit generation speed exponentially. This is in sharp contrast with MRAMs, where conservation of stability with extreme scaling is an important challenge [249].

Minimizing auto-correlation

From the study described previously, we observe that a 26% correlation between consecutive bits can be efficiently whitened out by XOR8 and allow generated random numbers to pass all the NIST STS tests.

This consideration, together with the model, allows us to evaluate quantitatively the speed of scaled random bit generators based on superparamagnetic tunnel junctions, by evaluating

the maximum sampling frequency to keep the correlation $\rho^c_{X,X+1} \lesssim 26\%$.
The auto-correlation function of a telegraphic signal can be derived simply from its escape rates, allowing to express the average correlation between consecutive samples as :

$$\rho_{X,X+1}(T_{\text{sampling}}) = e^{-(r_{0\to1}+r_{1\to0})\,T_{\text{sampling}}}, \tag{5.2}$$

where $r_{0\to1}$ and $r_{1\to0}$ are the escape rates from 0 and 1 states respectively, and $T_{sampling}$ is the sampling period. In case of a symmetrical system:

$$r_{0\to1} = r_{1\to0} = f_0 \exp\left(-\frac{\Delta E}{k_B T}\right) = 2F_{\text{MTJ}}, \tag{5.3}$$

where ΔE is the energy barrier separating the two states. $\Delta E = K_{\text{eff}}(D)\pi\frac{D^2}{4}t$ is derived as a function of the device diameter D, where $t = 1.6$nm is the free magnet thickness and the effective anisotropy $K_{eff}(D)$ is derived considering interfacial anisotropy and bulk anisotropies, using experimental values from [243].
The maximal sampling frequency is then obtained knowing the critical correlation $\rho^c_{X,X+1}$ that can be mitigated by the whitening operation :

$$F^{\max}_{\text{sampling}} = \frac{4F_{\text{MTJ}}}{\ln(1/\rho^c_{X,X+1})} = \frac{2f_0}{\ln(1/\rho^c_{X,X+1})}\exp\left(-\frac{\Delta E}{k_B T}\right). \tag{5.4}$$

For the a critical correlation $\rho^c_{X,X+1} \approx 26\%$, the maximum sampling frequency is:

$$F^{\max}_{\text{sampling}} \approx 3F_{\text{MTJ}} = \frac{3}{2}f_0 \exp\left(-\frac{\Delta E}{k_B T}\right). \tag{5.5}$$

Fig. 5.4(a), based on this derivation, shows that random bits could be generated at up to tens of MHz for energy barriers below $5k_BT$, corresponding to a diameter of 8nm.

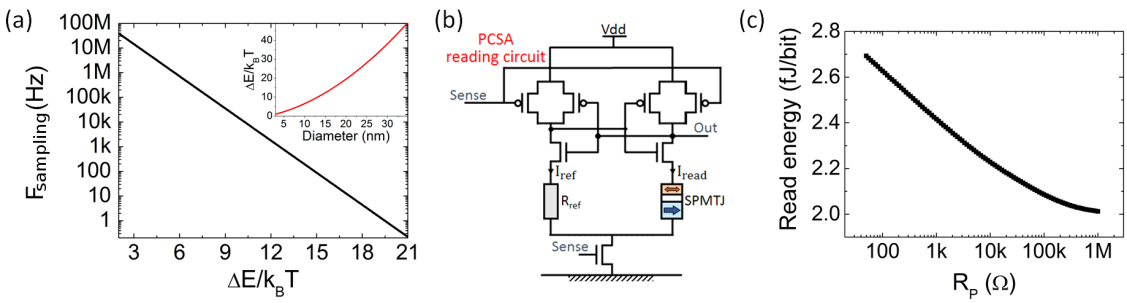


Figure 5.4: **(a)** Effect of scaling the energy barrier on the ideal sampling frequency, based on the device model. The inset shows the energy barrier as a function of the junction diameter for PMA-MTJs. **(b)** Precharge sense amplifier (PCSA) circuit for reading the state of a superparamagnetic tunnel junction. **(c)** PCSA reading energy as a function of the superparamagnetic tunnel junction P state resistance R_P.

Readout circuit

In addition, in a final system, specialized transistor-based electronics needs to be associated to the superparamagnetic tunnel junctions to read their states without interfering with the random bit generation quality. Here, we considered a precharge sense amplifier circuit (PCSA, Fig. 5.4(b)), a CMOS circuit originally proposed as an MRAM read circuit [250]. We simulated this circuit using standard integrated circuit design software (Cadence tools) and the transistor models of a 28nm commercial technology. The superparamagnetic tunnel junctions were modeled using a compact (VerilogA-based) model implementing the Arrhenius-Néel model. The results of circuit simulation (Fig. 5.4(c)) show that the read energy is relatively independent from superparamagnetic tunnel junction resistance, and very low ($\approx$ 2fJ/bit).

We also evaluated the read disturb effect of the PCSA. Reading the state of a junction can potentially affect random bit generation through the spin torque effect. Based on the spin torque model of [247], its impact on the mean state is around 10^{-6} for junctions such as the one we characterized experimentally. Ultrascaled junctions are expected to be more sensitive to this effect. The SPICE simulation results presented in Figure 5.5 show the mean state of a perfectly balanced ultrascaled stochastic magnetic tunnel junction as a function of the reading frequency. These results show that the read disturb effect stays below 0.1% for ultrascaled junctions functioning at high frequencies. This small effect would therefore be corrected by whitening.

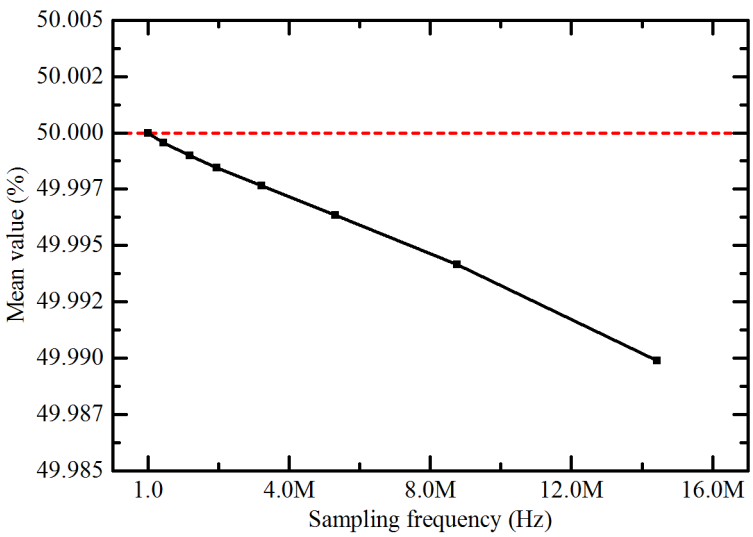


Figure 5.5: Mean value of a perfectly balanced ultrascaled stochastic MTJ as a function of the sampling frequency, obtained through SPICE simulation of the PCSA circuit and using our device model. The dashed red line represents the ideal 50% mean value.

Evaluating the energy consumption of random bit generation requires taking into account the whitening process. As XOR whitening combines multiple junction states per generated bit, it requires multiple read operations per generated bit. XOR8 reads 8 junctions to generate a

bit, and requires 20fJ/bit in average (including the XOR gate operation). In terms of area, in a 28 nm technology, the layout of a full XOR8 random bit generator takes less than $2\mu m^2$. XOR4 whitening would require 9.8fJ/bit and a $1\mu m^2$ area.

These results show the potential of superparamagnetic tunnel junctions for state of the art low-energy random number generation.

Sensitivity of the random bitstream generator to perturbations

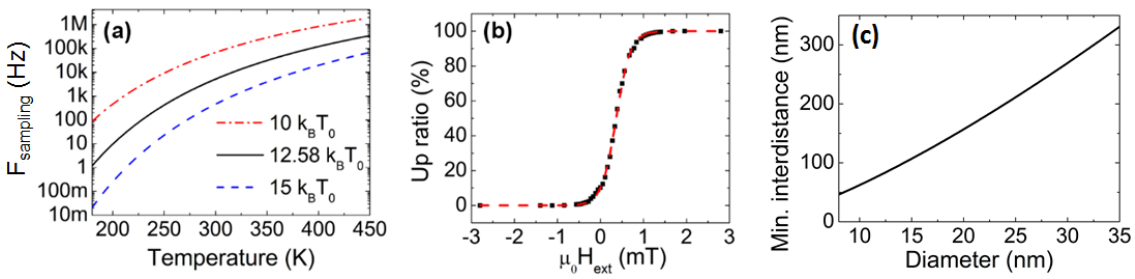


Figure 5.6: External perturbations and crosstalk effects. **(a)** Theoretical curve of the maximum sampling frequency for high quality random bit generation, as a function of temperature, for different junction stabilities (the black curve corresponds to the junction characterized in Figs. 5.2 and 5.3). **(b)** Black symbols: experimental mean state of the junction (up ratio) as a function of the applied magnetic field (red dotted line: theory). **(c)** Theoretical minimal distance between superparamagnetic tunnel junctions allowed to prevent crosstalk, as a function of the superparamagnetic tunnel junction diameter.

Although superparamagnetic tunnel junctions allow random number generation with minimal energy, their sensitivity to external perturbations must be carefully evaluated.

Influence of temperature and magnetic fields

First, as the stochastic switching of superparamagnetic tunnel junctions is thermally activated, temperature directly affects their switching rates. Fig. 5.6(a), based on the model introduced in the previous section, shows the temperature dependence of the maximum sampling frequency for several values of the effective barrier. Higher temperatures produce better random numbers: as temperature increases, the superparamagnetic tunnel junction switching rates increase accordingly, thus allowing faster sampling frequencies. Devices should therefore be sized based on their lowest operation temperature.

Superparamagnetic tunnel junctions are also sensitive to magnetic fields. Fig. 5.6(b) shows the experimental mean state of a superparamagnetic junction as the function of external magnetic field. Fields of a few Oe shift the mean state to a level that cannot be corrected by XOR8 whitening. Magnetic shielding is therefore necessary for applications. Such technology (based on mu metals) has already been developed for MRAM.

Crosstalk effects

Finally, a challenge regarding scalability and integration is that closely packed superparamagnetic tunnel junctions can interact by dipolar interaction, which could lead to correlations in random numbers.

To evaluate this effect, we model the free magnetic layers of two identical side by side superparamagnetic tunnel junctions as two dipoles interacting through dipolar coupling (figure 5.7(a)).

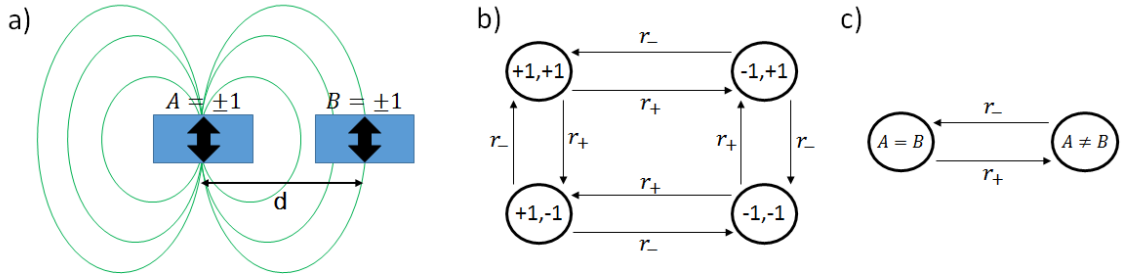


Figure 5.7: **(a)** Representation of the free magnetic layers of two identical side by side superparamagnetic tunnel junctions modeled as two dipoles. Their states are associated respectively to the variables A and B, each taking two possible values $\{+1,-1\}$. **(b)** 4-states Markov chain showing the possible transitions between all {A,B} system states, and associated transition rates. **(c)** Simplified Markov chain.

The two superparamagnetic tunnel junctions constitute a stochastic system described as a 4-states markov process [251] illustrated in figure 5.7(b). Their states are modeled by two variables $\{A,B\}$ that can take two values $\{-1,+1\}$. In the absence of interaction, the transition rates are strictly identical: $r = f_0 \exp\left(-\frac{\Delta E}{k_B T}\right)$. However, dipolar interaction induce an asymmetry so that two different rates have to be considered [240]:

$$\begin{cases} r_+ = f_0 \exp\left(-\dfrac{\Delta E}{k_B T}\left(1+\dfrac{H_{dip}}{H_k}\right)^2\right) \\ r_- = f_0 \exp\left(-\dfrac{\Delta E}{k_B T}\left(1-\dfrac{H_{dip}}{H_k}\right)^2\right) \end{cases}, \tag{5.6}$$

where $H_k = \frac{2\Delta E}{k_B T}$ is the magnetic layer's coercive field and H_{dip} is the amplitude of the dipolar field radiated by one junction onto the other and is given by $H_{dip} = \frac{M_S V}{4\pi d^3}$
This system can be simplified as the two-state Markov chain shown in figure 5.7(c) and following:

$$\begin{cases} \partial_t P_{A=B} = +r_- P_{A\neq B} - r_+ P_{A=B} \\ \partial_t P_{A\neq B} = -r_- P_{A\neq B} + r_+ P_{A=B} \end{cases}, \tag{5.7}$$

with stationary solutions : $P_{A=B} = \frac{r_+}{r_+ + r_-}$ and $P_{A\neq B} = \frac{r_-}{r_+ + r_-}$.

The cross-correlation between the two junctions states is then expressed as:

$$\rho = \frac{|\langle AB \rangle|}{\sqrt{\langle A^2 \rangle \langle B^2 \rangle}} = |\langle AB \rangle| = P_{A \neq B} - P_{A=B} \,. \tag{5.8}$$

Using the expressions from equation 5.6, we obtain :

$$\rho = \tanh\left(2 \frac{\Delta E}{k_B T} \frac{H_{dip}}{H_k}\right). \tag{5.9}$$

Replacing the expressions of the dipolar and coercive fields:

$$\rho = \tanh\left(\frac{\mu_0 (M_S V)^2}{4\pi k_B T d^3}\right). \tag{5.10}$$

We evaluate that in the case of perpendicularly magnetized superparamagnetic tunnel junctions, the center-to-center distance between two devices guaranteeing negligible crosstalk, corresponding to less than $\rho_c = 0.1\%$ cross-correlation, is given by:

$$d_c = \left(\frac{\mu_0 (M_S V)^2}{4\pi k_B T \tanh^{-1}(\rho_c)}\right)^{1/3}. \tag{5.11}$$

Fig. 5.6(c) shows the evolution of this critical distance at room temperature as the diameter of the junctions is scaled down. It falls below 100nm for ultimately scaled 10nm diameter devices, which constitutes a layout design rule, and which would be naturally respected if the junctions are associated with PCSA circuits.

Comparison table with other hardware random bitstream generators

In this section, the properties of the proposed superparamagnetic tunnel junction random number generator are compared to other proposed approaches from the litterature.

Table 5.2 compares the functioning principle, the circuit area usage, the energy per bit, the bit generation rate and the bitstream quality for 17 different low-energy random number generation approaches at different stages of development.

Note that some authors present statistical test results on bitstreams generated by simulated architectures that use pseudo-random number generators to emulate the underlying random physical phenomena of the system (for example noise or metastability). These results must be with be taken with caution: with a good underlying pseudo-RNG, perfect statistical test results on the system output show at best that the simulation doesn't degrade the quality of the pseudo-RNG.

This table shows that the XOR8 random bitstream generator we proposed in this chapter reaches state-of-the-art energy/bit and area efficiencies. Depending on the application, random number quality can be slightly reduced by switching to XOR4 in order to divide by two the energy and area requirements.

Ref.	Description	Area	Energy per bit	Bit rate	Quality
This work	Experimenal measurement of superparamagnetic MTJs. XOR8 whitening.	*n/a*	*n/a*	5 kHz	100% NIST STS
This work	PCSA sampling of superparamagnetic MTJs. XOR8 whitening. Theory & simulations.	2 μm^2	20 fJ	10 MHz	*n/a*
[252] [253]	Ring oscillator jitter. Xilinx XCV300E-8 FPGA.	15 slices, 1% of FPGA	1.8 µJ	29 kHz	100% NIST STS
[254]	Ring oscillator jitter. Built in 0.18 µm CMOS.	0.0016 mm^2	230 pJ	10 MHz	100% FIPS 140-1 [255]
[256]	Single oxide trap telegraph noise amplification. Von Neumann whitener [257]. Built in 0.12 µm CMOS.	0.009 mm^2	250 pJ	200 kHz	*No info.*
[253]	Metastable latch. Running estimate filter. Built in 0.35 µm CMOS.	0.031 mm^2	1.88 nJ	5 kHz	100% NIST STS
[239]	Ring oscillator jitter. Built in 28 nm CMOS.	375 μm^2	23 pJ	23 MHz	100% NIST STS
[224]	Meta-stable CMOS circuit. Built in 45 nm CMOS.	4000 μm^2	2.9 pJ	2.4 GHz	100% NIST STS
Spin dice [232]	Probabilistic MTJ writing. XOR3 whitening. Experimental sampling.	*No info.*	*No info.*	600 kHz	Most NIST STS See [232]
[233]	Probabilistic MTJ writing. Probability tracking. Experimentally verified.	*No info.*	*No info.*	57 MHz	100% NIST STS
[258]	Probabilistic MTJ writing. Probability tracking. 28 nm FDSOI simulation.	5.88 μm^2	1.25 pJ	66.6 MHz	12/15 NIST STS (see [258])
[236]	Probabilistic MTJ writing. Simulation.	*No info.*	57.1 fJ	*No info.*	*No info.*
[238]	Probabilistic MTJ writing. Simulation.	139.96 μm^2	311 fJ	29.6 GHz	*n/a*
[259]	Single-electron transistor seeding a LFSR pseudo-RNG. Simulations.	*No info.*	*No info.*	See [259].	100% NIST STS
[260]	Differential ReRAM telegraph noise extractor feeding a LFSR pseudo-RNG. Experimental results.	*No info.*	*No info.*	*No info.*	100% NIST STS LSTM-tested
[261]	Non-deterministic OxRAM reset operation. Simulations.	*No info.*	*No info.*	25 kHz	See [261].
[235]	Straintronics metastability. 65 nm CMOS simulations.	0.001 mm^2	19 fJ	110 MHz	100% NIST STS

Table 5.2: Comparison of different proposed low power hardware random number generators.

Using superparamagnetic tunnel junctions for unconventional computing

This section focuses on applying the superparamagnetic tunnel junction random number generator to perform Bayesian inference with stochastic computing using C-elements on a toy problem illustrating its effectiveness for low energy stochastic computing.

Introduction to stochastic computing

Stochastic computing is an alternative computational approach for efficiently performing mathematical operations on data encoded as probabilistic bitstreams (infinite time sequences of 0s and 1s with different occurrence probabilities) instead of binary number representations. A given random bitstream A can encode an analog value an one of its statistical properties, typically the proportion of 1s, also called the "mean value" or "probability $P(A)$" of the bitstream.

The principles, advantages and some important constraints of stochastic computing are well illustrated by the stochastic multiplier circuit.

Figure 5.8(a) shows a standard 4-bit digital multiplier circuit that computes the product of two numbers in binary representation. It consists of 24 logic gates (134 transistors) and performs computations with a precision specified at design time.

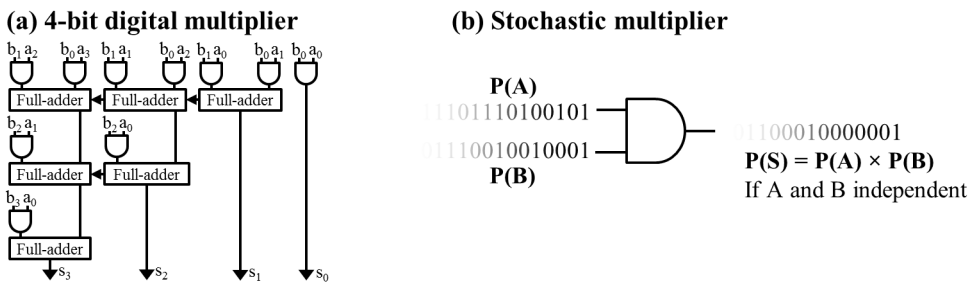


Figure 5.8: **(a)** 4-bit digital multiplier. **(b)** Stochastic multiplier computing a bitstream with a probability $P(S)$ equal to the product of the input bitstream probabilities $P(A) \times P(B)$.

On the other hand, figure 5.8(b) shows a stochastic multiplier circuit: a single AND logic gate (4 transistors). It takes as inputs two independent probabilistic bitstreams A and B with probabilities $P(A)$ and $P(B)$, performs bit-wise AND operations, and outputs a bitstream C with probability $P(C) = P(A) \times P(B)$, therefore naturally implementing multiplication of probabilities. In practice, the precision of the computation only depends on the averaging time of the signal C for the measurement of $P(C)$.

The stochastic multiplier circuit therefore performs a low energy (one logic gate) product operation, and its precision can be freely chosen through the averaging time.

However, this approach requires the two input bitstreams to be independent ($P(B|A) = P(B)$). If it is not the case, the output becomes $P(S) = P(A)P(B|A)$.

This simple example illustrates the power of stochastic computing which allows it to perform certain operations on decimal numbers with finite arbitrary precision at low energy and circuit area, which matches the requirements of Internet of Things devices, for example.

This example also underlines the requirement of the bitstreams to behave in a purely stochastic way (i.e. like independent biased coin flips), which requires high quality random bit generation compatible with low power and area requirements.

Performing Bayesian inference with stochastic computing using probabilistic bitstream generators

Bayesian inference is a technique that infers the probability of a given event occurring knowing a set of clues. It is used for example for illness diagnosis given a set of symptoms [262]. In this subsection, I present the principles of Bayesian inference, and how to implement it in a stochastic computing context to achieve simple binary classification.

Principles of Bayesian inference

Bayesian inference is an important approach in statistics and stochastic computing for classification and decision making [262]. It consists in predicting the "posterior" probability $P(X|E_1, E_2, \cdots, E_N)$ of an event X occurring for a given observed set of conditions ("evidence") $\{E_1, E_2, \cdots, E_N\}$. For example, it can represent the probability of a patient having a given illness ($X = 1$) knowing the presence or absence of symptoms $E_1 = 1, 0, \cdots, E_N = 1, 0$. This prediction relies on the Bayes' Theorem:

$$P(X|E_1, E_2, \cdots, E_N) = \frac{P(E_1, E_2, \cdots, E_N|X)P(X)}{P(E_1, E_2, \cdots, E_N)}. \tag{5.12}$$

In this equation, $P(E_1, E_2, \cdots, E_N|X)$ represents the "likelihood" gathered from statistics on known data. For example, it can represent the probability of a given set of symptoms occurring in ill $X = 1$ or healthy $X = 0$ patients.

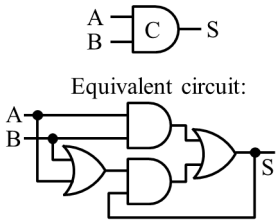
$P(X)$ is the "prior" which characterizes knowledge of the global occurrence probabilities of the outcomes of X. It could represent the known occurrence rates of the illness among patients. When no prior knowledge is available, this value can be set to 0.5.

$P(E_1, E_2, \cdots, E_N)$ is the "marginal probability" of the evidence. It characterizes the global occurrence probability of a given evidence set in the data: for example the probability of a given set of symptoms being observed on a randomly chosen patient.

Bayesian inference using C-elements

The Muller C-element, also known as the C-gate, is a simple sequential logic element with two inputs A and B, and one output S. It outputs the value of its inputs when they are identical, or keeps its previous output value S_{prev} when the inputs are different. Its symbol and equivalent circuit are shown in Figure 5.9(a), and its truth table is given in Figure 5.9(b).

(a) C-element



(b) Truth table

A	B	S
0	0	0
0	1	S_{prev}
1	0	S_{prev}
1	1	1

Figure 5.9: **(a)** C-element symbol and equivalent circuit. **(b)** C-element truth table.

Given the truth table in Figure 5.9(b), the output S of the C-element is 1 when:

- $A = 1$ and $B = 1$
- $\big((A = 1 \text{ and } B = 0) \text{ or } (A = 0 \text{ and } B = 1)\big)$ and $S_{\text{prev}} = 1$

This translates into the following equivalent probabilistic expression:

$$P(S) = P\left((A \cap B) \cup \left(\left(\left(A \cap \bar{B}\right) \cup \left(\bar{A} \cap B\right)\right) \cap S_{\text{prev}}\right)\right). \tag{5.13}$$

When bitstreams A and B are independent, not auto-correlated in time, and that the output probability is stable ($P(S) = P(S_{\text{prev}})$), the output probability of the C-element becomes:

$$P(S) = \frac{P(A)P(B)}{P(A)P(B) + (1 - P(A))(1 - P(B))}. \tag{5.14}$$

Bayes' theorem (equation 5.12) in the case of a single evidence term E_1 can also be reformulated into a similar form:

$$P(X|E_1) = \frac{P^*(E_1)P(X)}{P^*(E_1)P(X) + (1 - P^*(E_1))(1 - P(X))}, \quad \text{where} \quad P^*(E_1) = \frac{P(E_1|X)}{P(E_1|X) + P(E_1|\bar{X})}. \tag{5.15}$$

The left hand equation corresponds to the C-element output probability equation 5.14 with the following two inputs :

- A probabilistic bitstream with the prior probability $P(X)$
- A probabilistic bitstream with the normalized evidence probability $P^*(E_1)$

This result shows that a single C-element naturally computes $P(X|E_1)$ provided two independent and non-autocorrelated bitstreams with probabilities computed from the prior knowledge and known data statistics.

It can also be shown that multiple C-elements can be stacked in a tree-like structure to take into account multiple evidence terms [216], given the naive hypothesis that the occurrences of different words are conditionally independent. This hypothesis is relatively strong but was shown to provide good results in practice, and is at the basis of naive Bayes classifiers that have proven effective in various practical applications such as text classification or medical diagnosis [263].

Figure 5.10(c) shows the general structure of a cascaded C-element Bayesian inference architecture with 7 input bitstreams representing the normalized evidence terms with probabilities $\{P^*(E_1), \cdots, P^*(E_7)\}$, and a bitstream with the prior probability $P(X)$. The architecture has one output that provides the probability $P(X|E_1, \cdots, E_7)$. Thresholding this output allows simple binary classification.

Note : Successive operations and correlation

A common problem in stochastic computing is that many binary operations on bitstreams introduces correlations. In this example, each layer of C-elements creates larger "domains" of successive 0s and 1s in its outputs. The assumption of non-autocorrelation of the C-element inputs is therefore less and less satisfied starting from the second layer C-elements, and the quality of the inference falls when many layers are present. Decorrelation techniques exist but can be expensive in terms of energy or circuit area. This limitation is discussed in detail in [216].

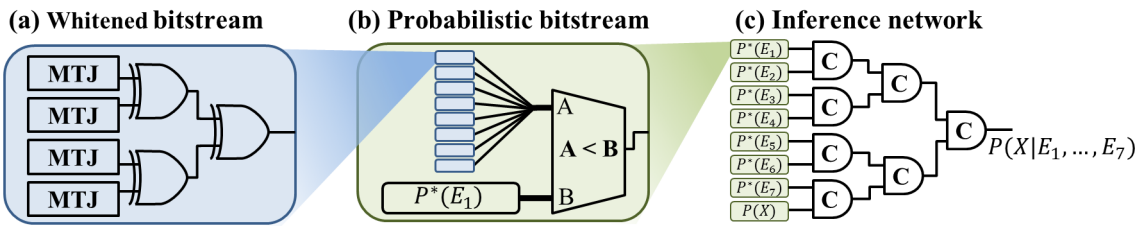


Figure 5.10: Schematic of a full C-element based Bayesian inference architecture using superparamagnetic magnetic junctions for probabilistic bitstream generation. **(a)** XOR-whitened stochastic magnetic tunnel junction random bitstream generator. **(b)** Probabilistic bitstream generation from whitened bitstreams using a comparator. **(c)** C-element Bayesian inference network.

Implementation using stochastic nano-oscillators

This classifier requires one probabilistic bitstream per evidence term, and one for the prior. These bitstreams can be generated at low energy cost with the proposed superparamagnetic tunnel junction random bit generator.

Figure 5.10(a) shows the XOR-whitened generator that produces independent, non-autocorrelated and unbiased "coin flip" bits. Figure 5.10(b) shows how those unbiased bitstreams are turned into probabilistic bitstreams. Random integers A are generated with each bit of their K-bit binary representation coming from an unbiased bitstream. These numbers uniformly span integers in the range $[0 \,..\, 2^K - 1]$. They are then compared, using a digital comparator, to a stored number $B = \text{Round}\left(P \times 2^K\right)$ with the strict comparison $A < B$. The output of the comparator is then a probabilistic bitstream with a probability of approximately P. The precision of this approximation, due to the integer rounding operation, increases with the number of bits K that encode the integers. However, higher K bit depths require proportionally higher numbers of unbiased random bit generators, which defines a trade-off between the precision of probability encoding, and circuit area and energy consumption.

Using this approach, the full architecture presented in Figure 5.10 can be implemented to perform low energy Bayesian inference.

Note : Direct probabilistic bitstream generation

As the probability of the non-whitened bitstream generated by a superparamagnetic tunnel junctions is tunable using magnetic fields (figure 5.6(b)) or currents [247], it could be possible to directly use it as a tunable probabilistic bitstream source. This would reduce the number of junctions to one per probabilistic bitstream and avoid the comparison operation, but would require lower sampling rates to erase auto-correlations previously corrected by the whitening operation.

In this work, I focus on illustrating the effectiveness of the superparamagnetic tunnel junction random number generator using this architecture because it constitutes a simple concrete example that requires important quantities of high quality random bits.

Implementing Bayesian inference for message classification

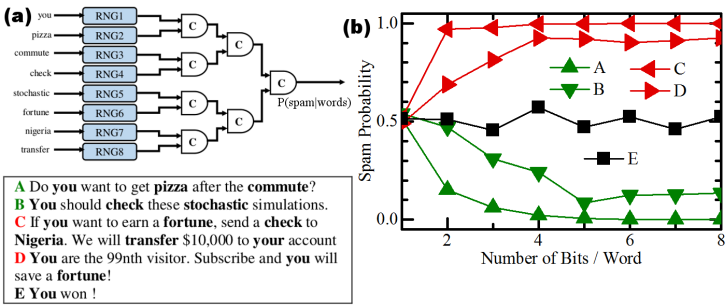
To illustrate the potential of superparamagnetic tunnel junctions for unconventional computing, we use the experimental whitened random bitstreams as inputs for a C-element based Bayesian inference scheme to perform message classification. In this pedagogical task, we use this circuit to classify email messages as spam or not spam (sample messages are presented in Fig. 5.11(a)), as recently introduced in [216].

The approach uses a dictionary of known words with their associated occurrence rates in spam and non-spam messages. Each word i of the dictionary constitutes an evidence term E_i and has an associated probabilistic bitstream generator whose probability $P^*(E_i)$ of drawing a 1 is set to a different value depending on the presence (or absence) of the word in the presented sentence. No prior is assumed ($P(\text{spam}) = 0.5$), which allows omitting this term from the C-element tree. The outputs of the bitstream generators are then combined using C-

elements to perform an approximate Bayesian inference. The time average of the output gives the probability of the presented message being spam.

Fig. 5.11(b) gives the spam probability inferred using XOR4-whitened bitstreams and shows that the more random bit generators are used per word, the more precisely the probabilistic binary generator can be tuned and the better the prediction is. Also, the longer the output averaging time, the more accurate the answer of the system is. A trade-off to keep a low energy consumption is found for 8 random bit generators / word and averaging over 2000 samples.

Because of its reliance on multiple stages of binary bitstream combination, and fine generator probability tuning, this circuit is sensitive to the quality of the underlying random number generator. We tested the circuit using raw 5kHz-sampled experimental bitstreams, as well as its XOR4 and XOR8 whitened versions. When the bits are not whitened, the circuit does not perform satisfactorily (Fig. 5.11(c)). Using bits whitened with XOR8, the circuit performs as well as the reference Blum whitener, successfully classifying all messages. Furthermore, XOR4, which does not pass all NIST STS tests, also provides perfect classification while requiring less energy.



(c)

Method	Messages correctly classified
No Whitening	60%
XOR4	100%
XOR8	100%
Blum	100%

Figure 5.11: Email classification with stochastic computing using whitened experimental random bitstreams. **(a)** Stochastic email classification circuit, and email messages to classify. One "RNG" block includes several random bit generators to provide bits with controllable probability. **(b)** Resulting spam probability as a function of the number of random bits per word using XOR4-whitened experimental 5kHz data over 2000 iterations. **(c)** Spam classification success rates for different whitening techniques for 5kHz sampling, using 8bits/word and 2000 iterations.

These results highlight the potential of the approach for low-energy applications. Using the results of the previous section, circuit simulation with 8 random bit generators / word and 2,000 clock cycles shows that a message can be classified using only nJ energy. The exact energy depends on the number of words in the dictionary, and is shown in figure 5.12.

This simple study shows that superparamagnetic tunnel junctions can be used to achieve efficient random number generation for low-power probabilistic computing.

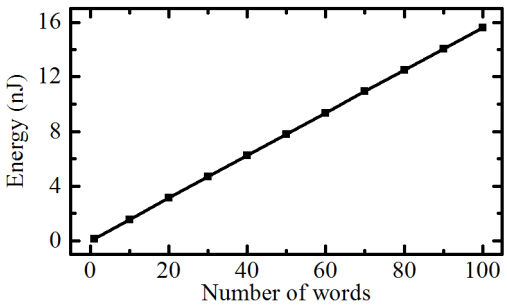


Figure 5.12: Total energy required to classify e-mail messages, as a function of the number of words in the dictionary.

Conclusion

In this work published as [264], we have shown that the natural dynamics of superparamagnetic tunnel junctions produces random telegraph signals that can be read and turned into high quality random bitstreams with minimal energy and circuit overhead, while staying fully compatible with standard CMOS fabrication processes.

The whitening process turning these measurements into usable random bitstreams implies energy and area overhead. But while reference Blum whitening would add important CMOS overhead, whitening based on XOR gates adds very little. XOR8 and Blum both provide high random bit quality consistent with cryptographic requirements, but XOR8 fits better to low energy applications, as it typically requires only 20fJ/bit and $2\mu m^2$, orders of magnitudes less than the demonstrated state of the art. This efficiency comes at the cost of speed. Scaled superparamagnetic tunnel junctions could generate random bits at speeds of dozens of MHz, which is slower than higher energy random bit generators, but sufficient for many unconventional computing schemes in very low power consumption contexts such as the Internet of Things. This efficiency also comes at the cost of a certain sensitivity of random bit generation to the environment, making it prone to attacks. Random bit generation based on superparamagnetic tunnel junctions is therefore much better suited for unconventional computing than for cryptographic applications.

The evaluation of the probabilistic email classifier circuit also suggests that in many alternative computing schemes, lower-quality whitening can be used successfully to achieve extreme energy efficiency without degrading performance. At design time, a balance between random number quality, generation speed, and energy consumption can be freely chosen to suit the target application. This is especially important in the context of modern Bayesian inference systems [265, 266], but also for embedded circuits and Internet of Things applications that are designed to work at low frequencies and low energies.

This study shows, through the example of superparamagnetic tunnel junctions acting as natural noise amplifiers, that emerging nanodevices, and stochastic nano-oscillators in particular, can be used as highly efficient true randomness sources for a wide range of applications.

Conclusions and future work

> "L'homme ne peut découvrir de nouveaux océans tant qu'il n'a pas le courage de perdre de vue la côte."
>
> André GIDE

"THIS FINAL PART *summarizes the work done in this thesis, highlighting its contributions. In the light of those conclusions, ideas for future research directions are suggested.* "

THIS part summarizes the thesis, highlights its main contributions and draws conclusions, as well as promising ideas for future research on the topic.

Summary

After listing existing nano-oscillator technologies, the first Chapter of this thesis shows that oxide-based relaxation oscillators, CMOS ring oscillators, MEMS oscillators, Josephson junctions and spintronic nano-oscillators have great potential for applications. The Chapter then summarizes existing oscillator-based computing approaches, and concludes that they perform well on specific tasks, but that pattern classifiers, currently in high demand, do not have satisfying oscillator-based implementations.

As a starting point to implement such a classifier, I study a promising oscillator-based pattern classification idea formulated in a purely mathematical context by Vassilieva *et al.* which uses synchronization readout and natural frequency control in an oscillator network to achieve classification. After proposing simple and robust synchronization detection schemes, a full-featured oscillator-based classifier architecture is defined in the second Chapter. Through extensive simulations, I show that this architecture scales well and is compatible with the constraints and non-idealities of nano-technologies provided certain design rules.

The third Chapter focuses on a specific oscillator type, the vortex spin transfer nano-oscillator, studied experimentally by Julie Grollier's group. The Chapter defines an accurate VerilogA model of the device, compatible with industrial circuit simulation tools, which show that the device matches the requirements of the classifier architecture. The third Chapter then describes a circuit using these devices to implement the classifier. Thorough mathematical analysis and simulations show that this circuit behaves as expected. A variant of the architecture with EEG-like readout is also proposed.

To perform useful classification tasks, a learning method is required. In the fourth Chapter, I propose a novel offline learning algorithm, with extensions to the classifier architecture, allowing it to perform more advanced tasks. A learning algorithm for the EEG-like version is also proposed in the Chapter. Both approaches show near-perfect results on standard machine learning benchmarks, implying that these architectures, with their learning algorithms, are good oscillator-based classifiers.

The last chapter focuses on a fundamentally different approach using intrinsically stochastic magnetic tunnel junction (MTJ) oscillators for random bitstream generation. I define a XOR-based whitening scheme producing random bits from MTJ signals, and validate it with experimental measurements I performed on devices provided by Julie Grollier. The resulting bitstream passes all NIST tests for randomness quality. Circuit simulations show that this generator has better energy and circuit area than the state of the art, and represents an important asset for low power probabilistic computing. This observation was validated by using experimental bitstreams to feed a probabilistic classifier circuit that successfully classified a set of e-mail massages as spam or not.

Conclusion and perspectives

As traditional computing technologies reach scaling limits, and appear fundamentally ill-suited for increasingly ubiquitous cognitive processing, various alternative computing schemes are being explored. Among them, multiple pattern classification approaches leveraging the complex dynamics of coupled oscillators have been proposed, but suffer from incompatibilities with current technologies. This thesis introduced realistic oscillator-based pattern classification schemes leveraging current nano-oscillator technologies, and showing competitive performance on standard classification tasks. Moreover, stochastic computing architectures using logic operations on probabilistic bitstreams for efficient computing show great potential for low-energy applications, but their power consumption is currently dominated by the heavy cost of random bit generation. This thesis also proposed a solution to this bottleneck by leveraging intrinsically stochastic nano-oscillators for energy-efficient, high quality random bit generation.

This thesis confirmed that nano-oscillators offer attractive properties for implementing advanced computational architectures. The case of the vortex spin transfer nano-oscillator was investigated in detail, showing that this type of device provides rich dynamics and high integration capabilities, which makes it a good candidate for computational architectures. Moreover, vortex spin transfer nano-oscillators offer a wide variety of coupling approaches, including both local and global electric and magnetic couplings. A possible development to this thesis could be to combine multiple such coupling schemes in a single oscillator network, which could give rise to extremely rich dynamics, potentially allowing more computational power to be extracted from a limited number of oscillators. Further experimental work is however required to increase the number of vortex spin transfer nano-oscillators that can be coupled on chip, and reduce their power consumption in order to pave the way towards higher scale experimental realizations of oscillator-based computing architectures. Similar studies should also be carried out with other types of nano-oscillators, including popular relaxation nano-oscillators or Josephson junctions.

Through the design of machine learning approaches for the oscillator-based classifier, this thesis also suggests that achieving competitive results with trainable nano-device based architectures requires rethinking machine learning principles to take device physics into account at their core. Research efforts should also be invested into online learning in order to provide more effective and flexible learning capabilities to nanodevice-based architectures. Despite recent advances, achieving effective online learning with nano-devices remains highly challenging, probably even more than offline learning.

Moreover, our realization of a high quality, low power true random number generator and probabilistic message classifier using stochastic nano-oscillators shows that undesirable phenomena, such as the noise that plagues devices when they are scaled down, could also be em-

braced and leveraged to achieve new types of computation.

In summary, this thesis contributes to a recent vision of nano-electronics, in which nano-devices act as more than simple switches, and their rich intrinsic physics are fully leveraged to achieve advanced computational tasks with a minimal number of devices. In this quest, inspiration can be drawn from biological systems, which have been optimized over billions of years following similar principles, in order to create a new generation of highly scalable, energy efficient and cognitive-ready computing systems.

List of publications and contributions

Peer-Reviewed Journal Articles

✠ YP LIN, CH BENNETT, T CABARET, **D VODENICAREVIC**, D CHABI, D QUERLIOZ, B JOUSSELME, V DERYCKE and JO KLEIN, "Physical Realization of a Supervised Learning System Built with Organic Memristive Synapses" , *Scientific Reports*, vol. 6, p. 31932, 2016.

✠ **D VODENICAREVIC**, N LOCATELLI, FA ARAUJO, J GROLLIER and D QUERLIOZ, "A Nanotechnology-Ready Computing Scheme based on a Weakly Coupled Oscillator Network" , *Scientific Reports*, vol. 7, p. 44772, 2017.

✠ **D VODENICAREVIC**, N LOCATELLI, A MIZRAHI, JS FRIEDMAN, AF VINCENT, M ROMERA, A FUKUSHIMA, K YAKUSHIJI, H KUBOTA, S YUASA, S TIWARI, J GROLLIER and D QUERLIOZ, "Low-Energy Truly Random Number Generation with Superparamagnetic Tunnel Junctions for Unconventional Computing" , *Physical Review Applied (PRA)*, vol. 8, p. 054045, 2017.

✠ M ROMERA, P TALATCHIAN, S TSUNEGI, FA ARAUJO, V CROS, P BORTOLOTTI, K YAKUSHIJI, A FUKUSHIMA, H KUBOTA, S YUASA, **D VODENICAREVIC**, N LOCATELLI, D QUERLIOZ and J GROLLIER, "Training an Array of Four Coupled Spin-torque Nano-oscillators to Classify Spoken Vowels" , 2018. *In preparation.*

Peer-Reviewed International Conference Proceedings

✠ N LOCATELLI, **D VODENICAREVIC**, W ZHAO, JO KLEIN, J GROLLIER and D QUERLIOZ, "Vortex-based Spin Transfer Oscillator Compact Model for IC Design" , *IEEE International Symposium on Circuits and Systems (ISCAS)*, p. 589-592, 2015.

✠ N. LOCATELLI, AF VINCENT, JS FRIEDMAN, **D VODENICAREVIC**, JV KIM, JO KLEIN, W ZHAO, J GROLLIER and D QUERLIOZ, "Spintronic Devices as Key Elements for Energy-efficient Neuroinspired Architectures" , *IEEE Design, Automation and Test in Europe (DATE)*, p. 994-999, 2015.

✠ **D VODENICAREVIC**, N LOCATELLI, J GROLLIER and D QUERLIOZ, "Synchronization Detection in Networks of Coupled Oscillators for Pattern Recognition" , *IEEE International Joint Con-*

ference on Neural Networks (IJCNN), p. 2015-2022, 2016.

✠ **D VODENICAREVIC**, N LOCATELLI and D QUERLIOZ, "A Neural Network Based on Synchronized Pairs of Nano-Oscillators" , *IEEE International Conference on Nanotechnology*, 2017.

International Conferences Without Proceedings

✠ **D VODENICAREVIC**, N LOCATELLI, JO KLEIN, J GROLLIER and D QUERLIOZ, "Non-linear Spin Transfer Oscillator Model for Neuro-inspired Architecture Design" , *Nonlinear Dynamics in Computational Neuroscience: from Physics and Biology to ICT*, 2015. *Poster*.

✠ *Capo-Caccia Cognitive Neuromorphic Engineering Workshop*, 2015. *Workgroups and oral presentations*.

✠ *Capo-Caccia Cognitive Neuromorphic Engineering Workshop*, 2016. *Workgroups and oral presentations*.

National Conferences

✠ **D VODENICAREVIC**, N LOCATELLI, JO KLEIN, J GROLLIER and D QUERLIOZ, "Computing with Oscillators" , *GDR Biocomp*, 2015. *Poster*.

✠ **D VODENICAREVIC**, N LOCATELLI and D QUERLIOZ, "Comparaison de la Consommation Énergétique entre les Systèmes Biologiques et les Systèmes Artificiels." , *Journée conférence et débat: atteindre une efficacité énergétique extrême dans les systèmes de calcul avec la bio-inspiration*, 2016. *Oral presentation*.

✠ **D VODENICAREVIC**, N LOCATELLI, J GROLLIER and D QUERLIOZ, "Efficient Classification with Coupled Oscillators." , *GDR Biocomp*, 2016. *Oral presentation*.

Other Conferences

✠ **D VODENICAREVIC**, N LOCATELLI, JO KLEIN, J GROLLIER and D QUERLIOZ, "Computing with Oscillators" , *Rencontres des Jeunes Physiciens*, 2015. *Poster*.

✠ **D VODENICAREVIC**, N LOCATELLI, J GROLLIER and D QUERLIOZ, "Associative Memory with Nanoscale Oscillators" , *Les Trimestres du LMB*, 2015. *Oral presentation*.

✠ **D VODENICAREVIC**, N LOCATELLI, JO KLEIN, J GROLLIER and D QUERLIOZ, "Associative Memory Leveraging the Quasi-Synchronization of Kuramoto Oscillators - Principle of Operation and Detection Circuits" , *MEMOS project summit*, 2016. *Oral presentation*.

✠ **D VODENICAREVIC**, N LOCATELLI, J GROLLIER and D QUERLIOZ, "Nanocomposants pour

le Calcul Bioinspiré" , *Institut des Systèmes Intelligents et de Robotique*, 2016. *Oral presentation.*

✠ **D VODENICAREVIC**, N LOCATELLI, J GROLLIER and D QUERLIOZ, "Leveraging the Dynamics of Nano-oscillators for Efficient Computation" , *Invited talk at NIST, Washington DC*, 2017. *Oral presentation.*

Appendix A

Genetic algorithm for learning with the "EEG-like" architecture

"THIS APPENDIX *details the implementation of the genetic algorithm summarized in Section 4.5.4.1.* "

This Appendix details the implementation of each step of the genetic algorithm summarized in Section 4.5.4.1, and uses the same notations. It details initialization, evaluation, selection, sexual reproduction and mutations, while providing the values chosen for all the constants involved in the algorithm.

Initialization

Initialization consists in generating an initial, random population of genomes. Contrary to the previously presented architectures that are initialized before learning to a given state with only slight variations, genetic algorithms require a heavily random and diverse initial population [267]. This allows distant solutions in the parameter space to be evaluated, combined and optimized. High population diversity improves parallel solution search, reduces the sensitivity to local fitness maxima, and enhances the capabilities of sexual reproduction.

In my simulations, the population size is $N_{\text{pop}} = 200$. I randomly selected the weights coming from the inputs in the uniform range $\pm\left(f^0_{\text{max}} - f^0_{\text{min}}\right)$ where $f^0_{\text{max}} = 750\text{MHz}$ and $f^0_{\text{min}} = 450\text{MHz}$ are the maximal and minimal natural frequencies allowed (values outside this range are clamped). Biases are selected so that the natural frequencies without inputs are uniformly randomly distributed in $[f^0_{\text{min}}; f^0_{\text{max}}]$.

Evaluation

Evaluation involves running the architecture using each genome g on the training set to obtain the fitnesses F_g. I use the following algorithm:

1. For each genome W_g, and every training set example m:
   simulate or physically run the architecture using the corresponding weight matrix to obtain the EEG-like signals $S_g^{(m)}$.
2. For every genome W_g:
   (a) Perform simple linear regression on the rescaling parameters α_g^a and β_g^a to minimize the quadratic norm between the vector of its rescaled outputs on the training examples $\boldsymbol{a_g}$ and the expected outputs vector $\boldsymbol{y_g}$.
   Keep the optimally rescaled outputs $\boldsymbol{a_g}$.
   (b) Compute the average quadratic error on the N_{train} training examples:
   $E_g = \frac{1}{N_{\text{train}}} \sum_m \left(a_g^{(m)} - y_g^{(m)}\right)^2$
   (c) Deduce the fitness of the genome: $F_g = \dfrac{1}{10^{-10} + E_g}$
   (10^{-10} prevents infinite values if $E_g = 0$)

The fitness F_g rates how well a genome W_g performs and dictates its probability to be selected for.

Note : Loss function choice

The quadratic error loss function was chosen instead of the cross-entropy loss usually applied to classification tasks mainly because cross-entropy doesn't behave properly with non-clamped rescaled functions, and would add complexity for regression and online learning.

Selection

Selection steers the evolution process towards maximizing fitness. It involves choosing the best genomes and allowing them to survive the generation and/or reproduce sexually. I use the following selection process:

1. Define a discrete probability distribution Q_{sel} of the genomes in the population where the probability of drawing a given genome g is weighted by its fitness F_g.
2. Select good genomes to be kept intact for the next generation:
   Draw N_{survive} from Q_{sel}. Add them to a "survivors" list. The same genome can be chosen multiple times.
3. Select $N_{\text{pop}} - N_{\text{survive}}$ pairs of genomes for sexual reproduction. Each genome is independently drawn from Q_{sel}. A genome can be paired with itself.

Allowing individuals with high fitness to survive reduces the probability of losing well performing solutions from the gene pool, and allows the next generation to reuse their parameters. In my simulations, I allow a survival rate of 20% which corresponds to $N_{\text{survive}} = 40$.

Moreover, the selection process uses probabilistic selection instead of directly ranking and selecting the best individuals for survival and sexual reproduction. This gives a chance to every genome to explore the parameter space before being selected away, which improves the diversity of the population and the resilience to local fitness maxima.

Sexual reproduction and mutations

Sexual reproduction and mutations drive the exploration of potential new solutions in the parameter space.

Sexual reproduction consists in combining two parent genomes in order to produce offspring that inherit and combine parental characters. It was shown to accelerate the evolution of both living beings [268], and simulated systems [200]. It is now heavily used in genetic algorithms.

Random mutations involve randomly shifting some parameters of a genome to make it explore nearby solutions in the parameter space. Throughout generations, it brings genomes towards local fitness maxima by trial and error.

To implement sexual reproduction and mutations, I used the following algorithm:

1. For each selected pair of parent genomes $P = \{W_a, W_b\}$, generate a new child genome W_c in which each matrix element W_{cij} is defined according to the following procedure:
   (a) Draw a random number p in the uniform range $[0;1)$.
   (b) If $p < P_{\text{combine}}$: set W_{cij} to a random value between W_{aij} and W_{bij}.
   else: set W_{cij} to either W_{aij} or W_{bij} with 50% chance.
   (c) Draw a new random number p in the uniform range $[0;1)$.
   (d) If $p < P_{\text{mutate}}$: add a random value to W_{aij} drawn from a 0-centered normal distribution with standard deviation σ_{mutate}.

During the sexual reproduction process, each matrix element is either inherited from a random parent, or as a combination of the two parent values with a probability $P_{\text{combine}} = 0.1$. Inheriting and combining parental traits provides an efficient way to accelerate the exploration of the parameter space while keeping previously found beneficial innovations.

The mutation probability in my simulations is $P_{\text{mutate}} = 0.1$ and the standard deviation is $\sigma_{\text{mutate}} = 0.1\left(f^0_{\text{max}} - f^0_{\text{min}}\right)$.

Appendix B

Effects of XOR whitening on bitstream probability and auto-correlation

"IN THIS APPENDIX, *we mathematically demonstrate the effects of XOR whitening on the bitstream mean value and auto-correlation.* "

Our results show that applying XOR whitening allows to pass all the NIST tests at higher sampling rates (see tables in article and S2). This is due to XOR combining the randomness of independent, individually lower quality bitstreams into a high quality one, with lower auto-correlation, and a mean value closer to 0.5.

Here we demonstrate mathematically that a XOR gate combines two independent bitstreams A and B into a whitened bitstream $S = \mathrm{XOR}(A, B)$ that has lower auto-correlation and a mean value closer to 0.5 than both A and B. We assume $P(A) \in (0;1)$ and $P(B) \in (0;1)$.

Auto-correlation

Auto-correlation can arise in consecutive samples of the bitstreams at t_1 and $t_2 = t_1 + \mathrm{dt}$. We define A_1 and A_2 as the two samples of A at t_1 and t_2 respectively. The same goes with B_1 and B_2 for bitstream B, and S_1, S_2 for the output bitstream S. We assume that all the probabilities are stable in time (for example $P(A_1) = P(A_2) = P(A)$). The conditional probability $P(A_1|A_2)$ can be reversed using the Bayes theorem, and shows the following symmetry:

$$P(A_1|A_2) = \frac{P(A_1)}{P(A_2)} P(A_2|A_1) = P(A_2|A_1)\,. \tag{B.1}$$

This also applies to $P(B_1|B_2)$ and $P(S_1|S_2)$.
We also have:

$$\begin{aligned} P(A_2 \cap \bar{A}_1) &= (1 - P(A))P(A_2|\bar{A}_1) = (1 - P(A))\frac{P(A)}{1 - P(A)} P(\bar{A}_1|A_2) \\ &= P(A)\big(1 - P(A_2|A_1)\big)\,. \end{aligned} \tag{B.2}$$

and

$$P(\bar{A}_2 \cap A_1) = P(A)P(\bar{A}_2|A_1) = P(A)\big(1 - P(A_2|A_1)\big)\,. \tag{B.3}$$

This implies the symmetry $P(\bar{A}_2 \cap A_1) = P(A_2 \cap \bar{A}_1)$.
We define the probability of the two samples A_1 and A_2 being equal as:

$$P(A_2 = A_1) = P\left((A_2 \cap A_1) \cup \left(\bar{A}_2 \cap \bar{A}_1\right)\right) = P\left(A_2 \cap A_1\right) + P\left(\bar{A}_2 \cap \bar{A}_1\right)\,. \tag{B.4}$$

Moreover, the sum of the probabilities of all the outcomes on A_1 and A_2 is one:

$$P(\bar{A}_2 \cap \bar{A}_1) + P(A_2 \cap A_1) + P(\bar{A}_2 \cap A_1) + P(A_2 \cap \bar{A}_1) = 1\,. \tag{B.5}$$

By identifying $P(A_2 = A_1)$ and using the symmetry $P(\bar{A}_2 \cap A_1) = P(A_2 \cap \bar{A}_1)$ in this equation we get:

$$P(\bar{A}_2 \cap A_1) = P(A_2 \cap \bar{A}_1) = \frac{1 - P(A_2 = A_1)}{2}\,. \tag{B.6}$$

Note that these results also apply to B and S.

The output of the XOR gate is equal for the two consecutive samples ($S_2 = S_1$) in the following cases:

- $A_1 = 0$, $B_1 = 0$, $A_2 = 0$, $B_2 = 0$; with probability $P(\bar{A}_2 \cap \bar{A}_1)P(\bar{B}_2 \cap \bar{B}_1)$,
- $A_1 = 0$, $B_1 = 0$, $A_2 = 1$, $B_2 = 1$; with probability $P(A_2 \cap \bar{A}_1)P(B_2 \cap \bar{B}_1)$,
- $A_1 = 1$, $B_1 = 1$, $A_2 = 0$, $B_2 = 0$; with probability $P(\bar{A}_2 \cap A_1)P(\bar{B}_2 \cap B_1)$,
- $A_1 = 1$, $B_1 = 1$, $A_2 = 1$, $B_2 = 1$; with probability $P(A_2 \cap A_1)P(B_2 \cap B_1)$,
- $A_1 = 0$, $B_1 = 1$, $A_2 = 0$, $B_2 = 1$; with probability $P(\bar{A}_2 \cap \bar{A}_1)P(B_2 \cap B_1)$,
- $A_1 = 0$, $B_1 = 1$, $A_2 = 1$, $B_2 = 0$; with probability $P(A_2 \cap \bar{A}_1)P(\bar{B}_2 \cap B_1)$,
- $A_1 = 1$, $B_1 = 0$, $A_2 = 0$, $B_2 = 1$; with probability $P(\bar{A}_2 \cap A_1)P(B_2 \cap \bar{B}_1)$,
- $A_1 = 1$, $B_1 = 0$, $A_2 = 1$, $B_2 = 0$; with probability $P(A_2 \cap A_1)P(\bar{B}_2 \cap \bar{B}_1)$.

The probability $P(S_2 = S_1)$ is obtained by summing those probabilities. After summing, factoring and using the $P(\bar{A}_2 \cap A_1) = P(A_2 \cap \bar{A}_1)$ symmetry we get:

$$\begin{aligned} P(S_2 = S_1) =& 4P(A_2 \cap \bar{A}_1)P(B_2 \cap \bar{B}_1) \\ &+ \big(P(A_2 \cap A_1) + P(\bar{A}_2 \cap \bar{A}_1)\big)\big(P(B_2 \cap B_1) + P(\bar{B}_2 \cap \bar{B}_1)\big). \end{aligned} \tag{B.7}$$

By identifying $P(A_2 = A_1)$ and $P(B_2 = B_1)$ we obtain the formula:

$$P(S_2 = S_1) = 1 - P(A_2 = A_1) - P(B_2 = B_1) + 2P(A_2 = A_1)P(B_2 = B_1), \tag{B.8}$$

from which the auto-correlation $\rho^S_{t_1,t_2}$ of the output bitstream S is deduced using $\rho^S_{t_1,t_2} = P(S_2 = S_1) - P(S_2 \neq S_1)$, and similar expressions for A and B (called $\rho^A_{t_1,t_2}$ and $\rho^B_{t_1,t_2}$):

$$\begin{aligned} \rho^S_{t_1,t_2} &= P(S_2 = S_1) - P(S_2 \neq S_1) = 2P(S_2 = S_1) - 1 \\ &= 1 - 2P(A_2 = A_1) - 2P(B_2 = B_1) + 4P(A_2 = A_1)P(B_2 = B_1) \\ &= \rho^A_{t_1,t_2} \times \rho^B_{t_1,t_2}. \end{aligned} \tag{B.9}$$

This result shows that the auto-correlation of the XOR output is always lower in magnitude than the auto-correlation of any of its inputs.

Moreover, the output of a XOR gate can be safely used as an input of another one for further whitening. Since the XOR operation is associative, XOR whitening can both be organized as a tree or a chain. The auto-correlation obtained after whitening N independent MTJ signals $\{A_1, A_2, \cdots, A_N\}$ is therefore the product of all the individual auto-correlations of the MTJ signals:

$$\rho^S_{t_1,t_2} = \prod_{i=1}^{N} \rho^{A_i}_{t_1,t_2}. \tag{B.10}$$

As a result, the magnitude of the auto-correlation after XOR whitening decreases exponentially on average with the number of combined MTJ signals. Highly correlated MTJ signals still contribute to this reduction, but less than the ones with low auto-correlation.

Mean value

The mean value $P(S)$ after XOR(A,B) is :

$$\begin{aligned} P(S) &= P\big((A \cap \bar{B}) \cup (\bar{A} \cap B)\big) = P(A)\big(1 - P(B)\big) + \big(1 - P(A)\big)P(B) \\ &= P(A) + P(B) - 2P(A)P(B)\,. \end{aligned} \tag{B.11}$$

The normalized difference between the ideal 0.5 mean value and $P(S)$ is $P(\bar{S}) - P(S) = 2\big(0.5 - P(S)\big)$. The same is defined for input A with $2\big(0.5 - P(A)\big)$ and input B with $2\big(0.5 - P(B)\big)$. The previous expression of $P(S)$ can be rearranged to obtain:

$$2\big(0.5 - P(S)\big) = 2\big(0.5 - P(A)\big) \times 2\big(0.5 - P(B)\big)\,. \tag{B.12}$$

When $P(A) = 0.5$ or $P(B) = 0.5$, we get $P(S) = 0.5$. Otherwise, we get the equivalence:

$$2|0.5 - P(S)| < 2|0.5 - P(A)| \Leftrightarrow 2|0.5 - P(B)| < 1\,, \tag{B.13}$$

which is always true.
These results show that the mean value at the output of a XOR gate is always closer to 0.5 than the mean value of any of its inputs, or equal to 0.5 if any of the inputs has already a mean value of exactly 0.5. In the case of N combined signals $\{A_1, A_2, \cdots, A_N\}$, the mean value of the whitened signal follows:

$$2\big(0.5 - P(S)\big) = \prod_{i=1}^{N} 2\big(0.5 - P(A_i)\big)\,. \tag{B.14}$$

This shows that the mean value of the whitened signal gets exponentially closer on average to 50% as the number of combined signals increases. Again, better-balanced signals contribute more to balancing the whitened bitstream than highly unbalanced ones.

As a conclusion, this study shows that XOR whitening reduces auto-correlations exponentially with the number of combined signals and brings the mean value exponentially closer to 50% with the number of combined signals.

Synthèse en français

“Ici nous présentons une synthèse en français des points importants de cette thèse. A summary in French of the main points of this work is presented here. ”

Introduction et contexte

De la reconnaissance automatique des visages et de la parole aux suggestions de films, de la traduction automatique aux véhicules autonomes, de la publicité ciblée à la détection de fraude, l'"intelligence artificielle" fait l'objet d'une demande sans précédent. Les ordinateurs personnels ou institutionnels, les appareils portables, et autres objets connectés se doivent de dépasser leurs fonctions traditionnelles de calcul logique et arithmétique pour effectuer des tâches "cognitives". Les progrès récents en apprentissage automatique ont ouvert un large spectre d'applications jusque là réservées aux êtres humains, et considérées il y a peu comme inaccessibles aux ordinateurs. Ainsi, les algorithmes de reconnaissance d'images affichent-ils des taux de reconnaissance supérieurs aux humains, et l'année 2016 a vu un ordinateur battre un champion humain au très stratégique jeu de Go.

Les architectures des ordinateurs et la technologie sous-jacente des transistors CMOS ont été optimisées sur des décennies pour effectuer des successions de milliards d'opérations logiques et arithmétiques par seconde, à haute précision, et sans aucune erreur. Au niveau architectural, les ordinateurs utilisent des unités séparées pour la mémoire et le calcul, séparation qui induit un sur-coût de communication appelé goulot d'étranglement de Von Neumann.

Cependant, la plupart des algorithmes cognitifs sont massivement parallélisables, fonctionnent de manière distribuée, et sont tolérants au bruit et aux erreurs. De ce fait, les ordinateurs classiques ne représentent pas la plateforme idéale pour ces nouveaux algorithmes. Des efforts ont été investis pour y adapter les ordinateurs, notamment avec l'utilisation de cartes graphiques contenant des milliers de processeurs, la réduction de la précision du calcul à virgule flottante, et l'apparition d'accélérateurs matériels (TPU) pour le calcul tensoriel au cœur de l'apprentissage automatique. Malgré ces efforts, les super-calculateurs consomment toujours des méga-watts pour effectuer des tâches que le cerveau humain traite avec 20 watt et des neurones bruités et peu fiables mais massivement parallèles qui combinent calcul et mémoire.

Les ordinateurs consommant plus de 10% de la puissance électrique mondiale en 2013, la technologie CMOS atteignant des limites fondamentales, et avec une demande croissante pour le calcul cognitif à basse énergie dans les systèmes portables et l'internet des objets, ces constatations ont ravivé l'intérêt pour le développement de systèmes de calcul alternatifs, souvent inspirés du cerveau.

Plus particulièrement, certaines approches visent à utiliser la dynamique des réseaux d'oscillateurs pour le calcul cognitif. Les phénomènes oscillatoires sont très communs dans la nature, et ont été observés à différentes échelles dans le cerveau humain, où ils semblent jouer un rôle dans le traitement de données. Des études mathématiques ont également démontré que même de petits réseaux de seulement quelques oscillateurs couplés suivaient déjà une physique très complexe exhibant chaos, transitions de phase et synchronisations. De plus, des avancées récentes en nano-technologie ont ouvert la voie à la fabrication de nano-oscillateurs à basse consom-

mation et massivement intégrables dans les systèmes actuels. Toutes ces découvertes suscitent un fort engouement pour la recherche et le développement de systèmes de calcul utilisant la physique très riche des nano-oscillateurs pour effectuer des tâches cognitives efficacement.

Diverses technologies de nano-oscillateurs existent, en particulier les oscillateurs à changement de phase, les nano-oscillateurs mécaniques, les nano-oscillateurs magnétiques, ou encore les nano-oscillateurs supraconducteurs. Leur fréquence de fonctionnement dans un circuit isolé, appelée “fréquence naturelle”, est aisément contrôlable par application de courants ou de tensions. De plus, de multiples nano-oscillateurs peuvent être couplés les uns aux autres, et des oscillateurs couplés se synchronisent lorsque leurs fréquences naturelles sont suffisamment proches.

Parmi les tâches cognitives, la classification de motifs est l’une des plus demandées. Elle consiste à attribuer automatiquement une étiquette à un motif présenté, indiquant la classe du motif. Par exemple, la classification de visages permet d’identifier une personne à partir d’une photographie de son visage. De même, la classification de fleurs permet d’identifier l’espèce à laquelle appartient une fleur à partir de certaines de ses propriétés (taille des pétales, sépales, etc...). Des architectures de classification à l’aide d’oscillateurs ont été proposées par le passé, mais présentent des problèmes de passage à l’échelle, ou sont incompatibles avec les nanotechnologies actuelles.

Cette thèse se fonde sur une architecture de classification à l’aide d’oscillateurs initialement proposée dans un cadre mathématique par Vassilieva *et al.* et qui requiert uniquement le contrôle des fréquences naturelles d’un réseau uniforme d’oscillateurs couplés, et la lecture de son état de synchronisation. Ceci la rend très prometteuse car compatible avec les contraintes technologiques actuelles. Cette architecture est composée d’un réseau d’oscillateurs couplés dits de “cœur”, eux-mêmes couplés à un ensemble d’oscillateurs d’entrée. Un motif est présenté en ajustant les fréquences naturelles des oscillateurs d’entrée, ce qui induit des perturbations du réseau de cœur, et y fait apparaître des synchronisations. La liste résultante des paires synchronisées d’oscillateurs de cœur est lue et représente l’étiquette attribuée par le réseau au motif présenté. La réponse du système doit être adaptée au problème à résoudre en ajustant les fréquences naturelles des oscillateurs de cœur.

Durant ma thèse, j’ai étendu cette architecture, l’ai adaptée au contexte des nano-oscillateurs, évalué sa robustesse, et défini un algorithme d’apprentissage adapté pour effectuer des tâches classiques d’apprentissage automatique.

Dans le dernier chapitre, j’ai effectué des mesures expérimentales sur une technologie fondamentalement différente: les nano-oscillateurs stochastiques, qui oscillent entre deux états de résistance de manière entièrement stochastique. À l’aide de ces données, j’ai pu définir et évaluer un circuit de génération de nombres aléatoires, que j’ai ensuite intégré au sein d’une architecture de classification probabiliste.

Résultats

Afin de pouvoir lire l'état de synchronisation du réseau de cœur, j'ai défini deux circuits de détection simples. Ces deux circuits évaluent tous deux l'état de synchronisation d'une paire d'oscillateurs en utilisant une bascule de Schmitt pour numériser les signaux de chaque oscillateur. L'un des circuits (le "compteur direct") compte la différence absolue du nombre de fronts montants entre les deux oscillateurs, et nécessite un compteur binaire signé pouvant être incrémenté et décrémenté. Le second circuit utilise une bascule RS pour compter les défauts d'alternance des fronts montants entre les deux oscillateurs, et ne nécessite qu'un compteur non signé pouvant être incrémenté. Un seuil peut ensuite être appliqué à ces compteurs, en dessous duquel la paire d'oscillateurs est considérée synchronisée. Afin de valider l'usage de ces circuits, j'en ai évalué, par simulation, la robustesse au bruit, la rapidité de convergence du résultat, la sensibilité au choix du seuil, et le comportement au sein de l'architecture de classification. Les deux circuits affichent des performances similaires, fournissent le résultat attendu sur le classificateur à oscillateurs et sont robustes à des niveaux de bruit réalistes. Pour de très hauts niveaux de bruit, le compteur direct offre une meilleure robustesse, au prix d'un circuit légèrement plus lourd. Le choix entre les deux détecteurs doit donc être fait en fonction du niveau de bruit attendu.

Ayant défini des circuits de détection, j'a ensuite effectué un ensemble de simulations sur l'architecture de classification complète avec un modèle générique d'oscillateurs (modèle de Kuramoto) afin d'évaluer la robustesse et la compatibilité de l'architecture avec les nano-technologies actuelles. J'ai ainsi simulé le classificateur en présence de différents niveaux de bruit, montrant que l'architecture est en effet robuste aux niveaux de bruit observés dans les principales technologies actuelles de nano-oscillateurs. Par ailleurs, les imperfections de fabrication induisent souvent une variabilité sur les composants et leurs connexions. J'ai donc évalué la sensibilité de l'architecture à la variabilité des fréquences naturelles des oscillateurs, ainsi qu'à celle des couplages entre oscillateurs, montrant que le classificateur est robuste à ces variabilités. En outre, en fonction de la méthode de couplage entre oscillateurs, des déphasages peuvent apparaître. Pour en mesurer les conséquences, j'ai simulé le classificateur à plusieurs déphasages induits par les couplages, montrant que l'architecture est sensible à ce phénomène. Celui-ci peut cependant être corrigé de différentes manières lors de la conception du système. Certains nano-oscillateurs peuvent également se comporter de façon non-linéaire. De ce fait, j'ai simulé l'architecture avec un modèle d'oscillateur non-linéaire, et ce à différents niveaux de non-linéarité, montrant que celle-ci augmente l'intensité du couplage, et induit des déphasages pouvant dans certains cas être bénéfiques à l'architecture. Les méthodes physiques de couplage entre oscillateurs peuvent aussi présenter une décroissance spatiale ou des déphasages liés à la distance entre oscillateurs. En simulant ces conditions, j'ai pu montrer que l'architecture est robuste à celles-ci à condition de les prendre en compte durant la phase de conception. Enfin, j'ai évalué le passage à l'échelle du classificateur, montrant que le nombre de classes

pouvant être discriminées augmente très rapidement avec le nombre d'oscillateurs de cœur et d'entrée.

Tous ces résultats tendent à confirmer que l'architecture ainsi définie est compatible avec les technologies actuelles.

J'ai également proposé une version alternative ("EEG-inspirée") de l'architecture utilisant le signal moyen des oscillateurs, semblable au signal électroencéphalographique du cerveau, pour lire l'état de synchronisation du réseau. Mes simulations ont montré que cette approche permettrait également de classifier des motifs, mais avec deux classes uniquement.

Le troisième chapitre se concentre sur une technologie particulière d'oscillateur: le nano-oscillateur vortex à transfert de spin. Ces nano-oscillateurs magnétiques présentent de multiples avantages parmi lesquels de faibles niveaux de bruit, des fréquences élevées (jusqu'au gigahertz), des fréquences naturelles ajustables sur une large gamme par application de courants, et des possibilités de couplage et synchronisation démontrées expérimentalement. De plus, nos collaborateurs du groupe de Julie Grollier à l'UMR CNRS/Thalès étudient ces dispositifs expérimentalement.

J'ai ainsi pu définir un modèle VerilogA de cette technologie de nano-oscillateur, compatible avec les outils industriels de simulation de circuits, notamment Cadence Spectre. À l'aide de simulations de circuits, j'ai pu montrer que ce modèle, une fois ajusté aux données expérimentales disponibles, prédit correctement la dépendance en courant de la fréquence naturelle, mais aussi la synchronisation à un signal oscillant.

J'ai ensuite défini un circuit de couplage entre ces oscillateurs qui implémente naturellement l'architecture de classification. Des simulations et un traitement mathématique m'ont permis de montrer que la réponse de ce système est proche de la réponse idéale du classificateur à oscillateurs génériques étudié précédemment.

Par ailleurs, un développement mathématique modélisant les oscillateurs au sein du circuit de couplage m'a permis d'approximer leur comportement selon un modèle d'oscillateur non-linéaire générique. Des simulations ont confirmé la validité de cette ré-écriture. Des approximations supplémentaires m'ont permis de formuler le comportement de ce circuit d'oscillateurs selon le modèle linéaire de Kuramoto. Cette formulation linéaire s'est également montrée semblable au modèle d'oscillateur complet dans plusieurs simulations, montrant que la description du système par le modèle de Kuramoto est valide, et suggérant que les résultats de robustesse obtenus avec le modèle de Kuramoto s'appliquent à cette technologie d'oscillateurs.

J'ai également pu montrer que la variante EEG-inspirée de l'architecture est aisément implémentable avec ce circuit.

Ces résultats et développements m'ont permis de montrer que les architectures proposées dans cette thèse sont compatibles avec la technologie des oscillateurs vortex à transfert de spin.

L'application du classificateur proposé dans cette thèse à une tâche particulière nécessite au préalable une étape d'apprentissage durant laquelle des exemples connus sont présentés, et les paramètres du système ajustés pour produire l'étiquette attendue.

Dans le quatrième chapitre, j'ai présenté un algorithme d'apprentissage hors ligne spécifique à cette architecture. J'ai tout d'abord défini une fonction d'erreur continue mesurant l'erreur moyenne commise par le classificateur sur un ensemble d'exemples connus. J'ai ensuite proposé une nouvelle approche de calcul du Jacobien d'un réseau d'oscillateurs permettant le calcul des dérivées partielles de la fonction d'erreur par rapport aux fréquences naturelles de cœur. Un algorithme d'optimisation par descente de gradient a ensuite permis d'utiliser cette dérivée pour ajuster les fréquences naturelles de cœur afin de minimiser l'erreur, ce qui correspond à un apprentissage de la classification souhaitée.

Des simulations d'apprentissage sur des problèmes de classification simples ont cependant montré que les capacités de classification de l'architecture se limitaient à des motifs particuliers du fait d'un faible nombre de paramètres et de la présence de symétries. À la lumière de ces résultats, j'ai proposé une version étendue, mais non moins réaliste, de l'architecture où la fréquence naturelle de chaque oscillateur est une combinaison linéaire apprise des composantes du motif d'entrée et d'une constante. Mes simulations sur cette nouvelle variante ont montré qu'elle était capable d'effectuer des tâches de classification de référence avec succès, et ce même en présence de bruit.

J'ai également étendu de la même façon la variante EEG-inspirée de l'architecture, et j'y ai joint un algorithme d'apprentissage génétique fondé sur les mutations, la reproduction sexuée et la sélection naturelle. Cette approche a fourni des résultats compétitifs sur des tâches classiques d'apprentissage automatique, tout en étant très robuste au bruit. Cependant, l'algorithme génétique s'est avéré beaucoup plus lourd en temps de calcul que l'algorithme de descente de gradient.

Ce travail m'a ainsi permis de proposer des algorithmes d'apprentissage pour classificateurs à oscillateurs, et de démontrer leur efficacité sur des tâches classiques de classification.

Dans le dernier chapitre de cette thèse, j'ai étudié une technologie fondamentalement différente: les oscillateurs magnétiques stochastiques. Ces oscillateurs ne suivent pas un comportement harmonique, mais oscillent de façon aléatoire, sous l'effet du bruit thermique, entre deux états différents de résistance électrique. À partir d'échantillons fournis par l'équipe de Julie Grollier, j'ai pu effectuer des mesures expérimentales de résistance sur ces dispositifs sur des temps longs. Après seuillage, ces mesures ont produit des signaux binaires télégraphiques de plusieurs gigabits.

Ayant à disposition des oscillateurs stochastiques produisant naturellement des signaux télégraphiques sans apport énergétique, je me suis concentré sur la réalisation d'un générateur de nombres aléatoires à basse énergie. Cependant, ces signaux télégraphiques étaient autocorrélés du fait de la fréquence d'échantillonnage élevée, et leur état moyen éloigné de 50% du

fait d'imperfections sur les dispositifs.

J'ai donc évalué la fréquence d'échantillonnage idéale, et combiné 8 signaux télégraphiques par des portes XOR afin de produire des bits indépendants et non biaisés. La séquence résultante a réussi tous les tests NIST STS, validant ainsi la qualité des bits générés.

À l'aide d'un modèle des oscillateurs stochastiques, des simulations circuit ont montré que 8 oscillateurs miniaturisés au maximum, lus à l'aide d'un circuit PCSA habituellement utilisé pour les mémoires magnétiques MRAM, et combinés par XOR permettaient de produire des nombres aléatoires de haute qualité à plusieurs mégahertz, 20 fJ/bit et une surface circuit de seulement $2\,\mu m^2$. Ces caractéristiques en font l'état de l'art de la génération de nombres aléatoires à basse énergie.

J'ai également évalué l'influence de perturbations environnementales sur la qualité des bits générés en considérant les variations de température, l'interaction dipolaire entre les oscillateurs magnétiques proches voisins, et la présence de champs magnétiques globaux. L'approche apparaît peu sensible aux variations de température, et l'éloignement entre oscillateurs induit par la présence des circuits de lecture PCSA rend l'interaction dipolaire négligeable. En revanche, le générateur est très sensible à l'application de champs magnétiques extérieurs, ce que j'ai pu valider expérimentalement. Par conséquent, un blindage magnétique tel que le Mu Metal utilisé dans les MRAM est nécessaire.

Enfin, j'ai utilisé les signaux expérimentaux combinés par XOR pour alimenter en nombres aléatoires un circuit simulé de classification probabiliste utilisant des portes C. Ce circuit a classifié avec succès des messages électroniques en tant que SPAM et NON SPAM avec des énergies de l'ordre du nanojoule, validant ainsi l'utilisation de ce générateur de nombres aléatoires dans le cadre du calcul probabiliste à basse énergie.

En résumé, cette thèse propose donc plusieurs nouvelles approches de calcul à l'aide de nano-oscillateurs et valide leur fonctionnement dans des cas réalistes.

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, December 2015.

[2] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[3] J. Hasler. Special report : Can we copy the brain? - A road map for the artificial brain. *IEEE Spectrum*, 54(6):46–50, June 2017.

[4] Mark P Mills. The cloud begins with coal: Big data, big networks, big infrastructure, and big power. *Digital Power Group*, 2013.

[5] Rachel Courtland. Transistors Could Stop Shrinking in 2021. *IEEE Spectrum: Technology, Engineering, and Science News*, July 2016.

[6] Steven H. Strogatz, Daniel M. Abrams, Allan McRobie, Bruno Eckhardt, and Edward Ott. Theoretical mechanics: Crowd synchrony on the Millennium Bridge. *Nature*, 438(7064):43–44, November 2005.

[7] Z. Néda, E. Ravasz, Y. Brechet, T. Vicsek, and A.-L. Barabási. Self-organizing processes: The sound of many hands clapping. *Nature*, 403(6772):849–850, February 2000.

[8] Arkady Pikovsky, Jürgen Kurths, and Michael Rosenblum. *Synchronization : a universal concept in nonlinear sciences*. Cambridge Univ. Press, Cambridge [u.a.], 2003.

[9] Renato E. Mirollo and Steven H. Strogatz. Synchronization of Pulse-Coupled Biological Oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, December 1990.

[10] Adilson E. Motter, Seth A. Myers, Marian Anghel, and Takashi Nishikawa. Spontaneous synchrony in power-grid networks. *Nature Physics*, 9(3):191–197, March 2013.

[11] Bo Li and K. Y. Michael Wong. Optimizing synchronization stability of the Kuramoto model in complex networks and power grids. *Physical Review E*, 95(1):012207, January 2017.

[12] Bernd Blasius, Amit Huppert, and Lewi Stone. Complex dynamics and phase synchronization in spatially extended ecological systems. *Nature*, 399(6734):354–359, May 1999.

[13] Andrew Moiseff and Jonathan Copeland. Firefly Synchrony: A Behavioral Strategy to Minimize Visual Clutter. *Science*, 329(5988):181–181, July 2010.

[14] John Buck and Elisabeth Buck. Mechanism of Rhythmic Synchronous Flashing of Fireflies. *Science*, 159(3821):1319–1327, March 1968.

[15] Herbert Spencer Barber. *North American Fireflies of the Genus Photuris*. Smithsonian Institution, 1951. Google-Books-ID: fd5YAQAACAAJ.

[16] Y. L. Maistrenko, O. V. Popovych, and P. A. Tass. Desynchronization and Chaos in the Kuramoto Model. In *Dynamics of Coupled Map Lattices and of Related Spatially Extended Systems*, number 671 in Lecture Notes in Physics, pages 285–306. Springer Berlin Heidelberg, 2005. DOI: 10.1007/11360810_12.

[17] Vegard Flovik, Ferran Macià, and Erik Wahlström. Describing synchronization and topological excitations in arrays of magnetic spin torque oscillators through the Kuramoto model. *Scientific Reports*, 6:32528, September 2016.

[18] Frederico A.C. Azevedo, Ludmila R.B. Carvalho, Lea T. Grinberg, José Marcelo Farfel, Renata E.L. Ferretti, Renata E.P. Leite, Wilson Jacob Filho, Roberto Lent, and Suzana Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541, April 2009.

[19] Thomas Trappenberg. *Fundamentals of Computational Neuroscience*. OUP Oxford, 2010. Google-Books-ID: 1xSLktiRAX4C.

[20] William E. Skaggs, Bruce L. McNaughton, Matthew A. Wilson, and Carol A. Barnes. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6(2):149–172, January 1996.

[21] Natalia Caporale and Yang Dan. Spike Timing–Dependent Plasticity: A Hebbian Learning Rule. *Annual Review of Neuroscience*, 31(1):25–46, 2008.

[22] H. M. Pinsker. Aplysia bursting neurons as endogenous oscillators. I. Phase-response curves for pulsed inhibitory synaptic input. *Journal of Neurophysiology*, 40(3):527–543, May 1977.

[23] Daniel K. Hartline and David F. Russell. Endogenous burst capability in a neuron of the gastric mill pattern generator of the spiny lobster Panulirus interruptus. *Journal of Neurobiology*, 15(5):345–364, September 1984.

[24] Dirk Bucher, Gal Haspel, Jorge Golowasch, and Farzan Nadim. Central Pattern Generators. In *eLS*. John Wiley & Sons, Ltd, 2001. DOI: 10.1002/9780470015902.a0000032.pub2.

[25] Eve Marder and Dirk Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986–R996, November 2001.

[26] Anca Doloc-Mihu and Ronald L. Calabrese. A database of computational models of a half-center oscillator for analyzing how neuronal parameters influence network activity. *Journal of Biological Physics*, 37(3):263–283, June 2011.

[27] Tamara J Schlichter, Brian Mulloney, Carmen Smarandache, and Timothy J Lewis. Phase response properties of an idealized half-center oscillator. *BMC Neuroscience*, 11(Suppl 1):P3, July 2010.

[28] Michael D. Fox, Abraham Z. Snyder, Justin L. Vincent, Maurizio Corbetta, David C. Van Essen, and Marcus E. Raichle. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9673–9678, July 2005.

[29] J. Craig Henry. Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, Fifth Edition. *Neurology*, 67(11):2092–2092–a, December 2006.

[30] J. R. Millan, F. Renkens, J. Mourino, and W. Gerstner. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6):1026–1033, June 2004.

[31] Jennifer A. Mohawk, Carla B. Green, and Joseph S. Takahashi. CENTRAL AND PERIPHERAL CIRCADIAN CLOCKS IN MAMMALS. *Annual review of neuroscience*, 35:445–462, 2012.

[32] Ole Jensen, Jochen Kaiser, and Jean-Philippe Lachaux. Human gamma-frequency oscillations associated with attention and memory. *Trends in Neurosciences*, 30(7):317–324, July 2007.

[33] Wolfgang H. R. Miltner, Christoph Braun, Matthias Arnold, Herbert Witte, and Edward Taub. Coherence of gamma-band EEG activity as a basis for associative learning. *Nature*, 397(6718):434–436, February 1999.

[34] Xiao-Jing Wang. Neurophysiological and computational principles of cortical rhythms in cognition. *Physiological Reviews*, 90(3):1195–1268, July 2010.

[35] C. Allen and C. F. Stevens. An evaluation of causes for unreliability of synaptic transmission. *Proceedings of the National Academy of Sciences*, 91(22):10380–10383, October 1994.

[36] A. Aldo Faisal, Luc P. J. Selen, and Daniel M. Wolpert. Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303, April 2008.

[37] E. D. Adrian and Yngve Zotterman. The impulses produced by sensory nerve-endings. *The Journal of Physiology*, 61(2):151–171, April 1926.

[38] M. B. Sachs, H. F. Voigt, and E. D. Young. Auditory nerve representation of vowels in background noise. *Journal of Neurophysiology*, 50(1):27–45, July 1983.

[39] John Huxter, Neil Burgess, and John O'Keefe. Independent rate and temporal coding in hippocampal pyramidal cells. *Nature*, 425(6960):828–832, October 2003.

[40] A. S. Landsman, E. Neftci, and D. R. Muir. Noise robustness and spatially patterned synchronization of cortical oscillators. *New Journal of Physics*, 14(12):123031, 2012.

[41] Mark S. Goldman. Enhancement of Information Transmission Efficiency by Synaptic Failures. *Neural Computation*, 16(6):1137–1162, June 2004.

[42] Mark S. Goldman, Pedro Maldonado, and L. F. Abbott. Redundancy Reduction and Sustained Firing with Stochastic Depressing Synapses. *Journal of Neuroscience*, 22(2):584–591, January 2002.

[43] G. Bard Ermentrout, Roberto F. Galán, and Nathaniel N. Urban. Reliability, synchrony and noise. *Trends in Neurosciences*, 31(8):428–434, August 2008.

[44] Pengcheng Zhou, Shawn D. Burton, Nathaniel N. Urban, and G. Bard Ermentrout. Impact of neuronal heterogeneity on correlated colored noise-induced synchronization. *Frontiers in Computational Neuroscience*, 7, August 2013.

[45] Daqing Guo and Chunguang Li. Stochastic and coherence resonance in feed-forward-loop neuronal network motifs. *Physical Review E*, 79(5):051921, May 2009.

[46] Paul C. Bressloff and Yi Ming Lai. Stochastic synchronization of neuronal populations with intrinsic and extrinsic noise. *The Journal of Mathematical Neuroscience*, 1(1):2, May 2011.

[47] Changgui Gu, Jinshan Xu, Jos Rohling, Huijie Yang, and Zonghua Liu. Noise Induces Oscillation and Synchronization of the Circadian Neurons. *PLOS ONE*, 10(12):e0145360, December 2015.

[48] Balth van der Pol Jun D.Sc. LXXXVIII. On “relaxation-oscillations”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):978–992, November 1926.

[49] A. B. Pippard. *The Physics of Vibration*. Cambridge University Press, 1989. DOI: 10.1017/CBO9780511622908.

[50] Jean-Marc Ginoux and Christophe Letellier. Van der Pol and the history of relaxation oscillations: Toward the emergence of a concept. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(2):023120, April 2012.

[51] D. Bhowmik and M. Shanahan. How well do oscillator models capture the behaviour of biological neurons? In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, June 2012. 00005.

[52] L. Schimansky-Geier. Kuramoto, Y., Chemical Oscillations, Waves, and Turbulence. Berlin-Heidelberg-New York-Tokyo, Springer-Verlag 1984. VIII, 156 S., 41 Abb., DM 79,—. US $ 28.80. ISBN 3-540-13322-4 (Springer Series in Synergetics 19). *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 66(7):296–296, 1986.

[53] Steven H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1–4):1–20, September 2000. 01501.

[54] Derek Michael Forrester. Arrays of coupled chemical oscillators. *Scientific Reports*, 5:srep16994, November 2015.

[55] K. Vasudevan, M. Cavers, and A. Ware. Earthquake sequencing: chimera states with Kuramoto model dynamics on directed graphs. *Nonlin. Processes Geophys.*, 22(5):499–512, September 2015.

[56] Juan A. Acebrón, L. L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The Kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1):137–185, 2005.

[57] Florian Dörfler and Francesco Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014.

[58] Joana Cabral, Etienne Hugues, Olaf Sporns, and Gustavo Deco. Role of local network oscillations in resting-state functional connectivity. *NeuroImage*, 57(1):130–139, July 2011.

[59] D. Cumin and C. P. Unsworth. Generalising the Kuramoto model for the study of neuronal synchronisation in the brain. *Physica D: Nonlinear Phenomena*, 226(2):181–196, February 2007.

[60] Michael Breakspear, Stewart Heitmann, and Andreas Daffertshofer. Generative Models of Cortical Oscillations: Neurobiological Implications of the Kuramoto Model. *Frontiers in Human Neuroscience*, 4, 2010.

[61] Joana Cabral, Henry Luckhoo, Mark Woolrich, Morten Joensson, Hamid Mohseni, Adam Baker, Morten L. Kringelbach, and Gustavo Deco. Exploring mechanisms of spontaneous functional connectivity in MEG: How delayed network interactions lead to structured amplitude envelopes of band-pass filtered oscillations. *NeuroImage*, 90:423–435, April 2014.

[62] L. Q. English, Zhuwei Zeng, and David Mertens. Experimental study of synchronization of coupled electrical self-oscillators and comparison to the Sakaguchi-Kuramoto model. *Physical Review E*, 92(5):052912, November 2015.

[63] X. Xin, T. Kikkawa, and Y. Liu. Analytical solutions of equilibrium points of the standard Kuramoto model: 3 and 4 oscillators. In *2016 American Control Conference (ACC)*, pages 2447–2452, July 2016.

[64] Chengwei Wang, Nicolás Rubido, Celso Grebogi, and Murilo S. Baptista. Approximate solution for frequency synchronization in a finite-size Kuramoto model. *Physical Review E*, 92(6):062808, December 2015.

[65] Sumito Tsunegi, Hitoshi Kubota, Kay Yakushiji, Makoto Konoto, Shingo Tamaru, Akio Fukushima, Hiroko Arai, Hiroshi Imamura, Eva Grimaldi, Romain Lebrun, Julie Grollier, Vincent Cros, and Shinji Yuasa. High emission power and Q factor in spin torque vortex oscillator consisting of FeB free layer. *Applied Physics Express*, 7(6):063009, June 2014.

[66] V. S. Tiberkevich, A. N. Slavin, and Joo-Von Kim. Temperature dependence of nonlinear auto-oscillator linewidths: Application to spin-torque nano-oscillators. *Phys. Rev. B*, 78(9):092401, 2008.

[67] Reihaneh Kouhi Esfahani, Farhad Shahbazi, and Keivan Aghababaei Samani. Noise-induced synchronization in small world networks of phase oscillators. *Physical Review E*, 86(3):036204, September 2012.

[68] Vasil S. Tiberkevich, Roman S. Khymyn, Hong X. Tang, and Andrei N. Slavin. Sensitivity to external signals and synchronization properties of a non-isochronous auto-oscillator with delayed feedback. *Scientific Reports*, 4:srep03873, January 2014.

[69] Barry E. Willner, Willard L. Miranker, and Chien-Ping Lu. Neural Organization of the Locomotive Oscillator. *Biol. Cybern.*, 68(4):307–320, February 1993.

[70] N. Locatelli, D. Vodenicarevic, W. Zhao, J. O. Klein, J. Grollier, and D. Querlioz. Vortex-based spin transfer oscillator compact model for IC design. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 589–592, May 2015.

[71] A. Sharma, J.A. Bain, and J.A. Weldon. Phase Coupling and Control of Oxide-based Oscillators for Neuromorphic Computing. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, PP(99):1–1, 2015.

[72] Matthew D. Pickett and R. Stanley Williams. Sub-100 fJ and sub-nanosecond thermally driven threshold switching in niobium oxide crosspoint nanodevices. *Nanotechnology*, 23(21):215202, 2012.

[73] Matthew D. Pickett, Gilberto Medeiros-Ribeiro, and R. Stanley Williams. A scalable neuristor built with Mott memristors. *Nature Materials*, 12(2):114–117, February 2013.

[74] Pablo Stoliar, Julien Tranchant, Benoit Corraze, Etienne Janod, Marie-Paule Besland, Federico Tesler, Marcelo Rozenberg, and Laurent Cario. A Leaky-Integrate-and-Fire Neuron Analog Realized with a Mott Insulator. *Advanced Functional Materials*, 27(11):n/a–n/a, March 2017.

[75] A. Beaumont, J. Leroy, J.-C. Orlianges, and A. Crunteanu. Current-induced electrical self-oscillations across out-of-plane threshold switches based on VO2 layers integrated in crossbars geometry. *Journal of Applied Physics*, 115(15):154502, April 2014.

[76] A. Parihar, N. Shukla, S. Datta, and A. Raychowdhury. Computing with dynamical systems in the post-CMOS era. In *2016 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, pages 110–111, July 2016.

[77] X. Zhao, R. Chebli, and M. Sawan. A wide tuning range voltage-controlled ring oscillator dedicated to ultrasound transmitter. In *Proceedings. The 16th International Conference on Microelectronics, 2004. ICM 2004.*, pages 313–316, December 2004.

[78] M. Grozing, B. Phillip, and M. Berroth. CMOS ring oscillator with quadrature outputs and 100 MHz to 3.5 GHz tuning range. In *ESSCIRC 2004 - 29th European Solid-State Circuits Conference (IEEE Cat. No.03EX705)*, pages 679–682, September 2003.

[79] A. A. A. Ghani and A. Saparon. A 1.4ghz CMOS Low-Phase Noise Voltage-Controlled Ring Oscillator. In *2007 5th Student Conference on Research and Development*, pages 1–5, December 2007.

[80] Bhavana Goyal, Shruti Suman, and P. K. Ghosh. Design of Ultra Low Power Voltage Controlled Ring Oscillator. In Nilesh Modi, Pramode Verma, and Bhushan Trivedi, editors, *Proceedings of International Conference on Communication and Networks: ComNet 2016*, pages 513–521. Springer Singapore, Singapore, 2017. DOI: 10.1007/978-981-10-2750-5_53.

[81] A. S. Elwakil and K. N. Salama. Nonlinear models of some ring oscillators. In *2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, pages 217–220, June 2008.

[82] Xiaoqing Ge, Murat Arcak, and Khaled N. Salama. Nonlinear Analysis of Ring Oscillator and Cross-Coupled Oscillator Circuits. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*.

[83] J. A. Walraven. Introduction to applications and industries for microelectromechanical systems (MEMS). In *International Test Conference, 2003. Proceedings. ITC 2003.*, volume 1, pages 674–680, September 2003.

[84] J. Philippe, G. Arndt, E. Colinet, M. Savoye, T. Ernst, E. Ollier, and J. Arcamone. Fully monolithic and ultra-compact NEMS-CMOS self-oscillator based-on single-crystal silicon resonators and low-cost CMOS circuitry. In *2014 IEEE 27th International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 1071–1074, January 2014.

[85] X. L. Feng, C. J. White, A. Hajimiri, and M. L. Roukes. A self-sustaining ultrahigh-frequency nanoelectromechanical oscillator. *Nature Nanotechnology*, 3(6):342–346, June 2008.

[86] Changyao Chen, Damian Zanette, David Czaplewski, Jeffrey Guest, and Daniel Lopez. A self-saturating mechanical oscillator with linear feedback. In *Bulletin of the American Physical Society*, volume Volume 61, Number 2. American Physical Society, 2016.

[87] M. Zalalutdinov, K. L. Aubin, M. Pandey, A. T. Zehnder, R. H. Rand, H. G. Craighead, J. M. Parpia, and B. H. Houston. Frequency entrainment for micromechanical oscillator. *Applied Physics Letters*, 83(16):3281–3283, October 2003.

[88] E. Colinet and C. Kharrat. Modal Control of Mechanically Coupled NEMS Array for Tunable Oscillators. In *2010 First Workshop on Hardware and Software Implementation and Control of Distributed MEMS*, pages 16–19, June 2010.

[89] Mian Zhang, Gustavo S. Wiederhecker, Sasikanth Manipatruni, Arthur Barnard, Paul McEuen, and Michal Lipson. Synchronization of Micromechanical Oscillators Using Light. *Physical Review Letters*, 109(23):233906, December 2012.

[90] Matthew H. Matheny, Matt Grau, Luis G. Villanueva, Rassul B. Karabalin, M. C. Cross, and Michael L. Roukes. Phase Synchronization of Two Anharmonic Nanomechanical Oscillators. *Physical Review Letters*, 112(1):014101, January 2014.

[91] Georg Heinrich, Max Ludwig, Jiang Qian, Björn Kubala, and Florian Marquardt. Collective Dynamics in Optomechanical Arrays. *Physical Review Letters*, 107(4):043603, July 2011.

[92] C. A. Holmes, C. P. Meaney, and G. J. Milburn. Synchronization of many nanomechanical resonators coupled via a common cavity field. *Physical Review E*, 85(6):066203, June 2012.

[93] S. P. Benz and C. J. Burroughs. Coherent emission from two-dimensional Josephson junction arrays. *Applied Physics Letters*, 58(19):2162–2164, May 1991.

[94] S. P. Benz and C. J. Burroughs. Two-dimensional arrays of Josephson junctions as voltage-tunable oscillators. *Superconductor Science and Technology*, 4(11):561, 1991.

[95] R. P. Robertazzi and R. A. Buhrman. Josephson terahertz local oscillator. *IEEE Transactions on Magnetics*, 25(2):1384–1387, March 1989.

[96] L. Ozyuzer, A. E. Koshelev, C. Kurter, N. Gopalsami, Q. Li, M. Tachiki, K. Kadowaki, T. Yamamoto, H. Minami, H. Yamaguchi, T. Tachiki, K. E. Gray, W.-K. Kwok, and U. Welp. Emission of Coherent THz Radiation from Superconductors. *Science*, 318(5854):1291–1293, November 2007.

[97] S. P. Benz and P. A. A. Booi. High-frequency oscillators using phase-locked arrays of Josephson junctions. In *Proceedings of IEEE 48th Annual Symposium on Frequency Control*, pages 666–669, June 1994.

[98] Shane A. Cybart, E. Y. Cho, T. J. Wong, Björn H. Wehlin, Meng K. Ma, Chuong Huynh, and R. C. Dynes. Nano Josephson superconducting tunnel junctions in YBa2cu3o7–δ directly patterned with a focused helium ion beam. *Nature Nanotechnology*, 10(7):598–602, July 2015.

[99] A. B. Cawthorne, P. Barbara, S. V. Shitov, C. J. Lobb, K. Wiesenfeld, and A. Zangwill. Synchronized oscillations in Josephson junction arrays: The role of distributed coupling. *Physical Review B*, 60(10):7575–7578, September 1999.

[100] A. K. Jain, K. K. Likharev, J. E. Lukens, and J. E. Sauvageau. Mutual phase-locking in Josephson junction arrays. *Physics Reports*, 109(6):309–426, July 1984.

[101] Yurii N. Ovchinnikov and Vladimir Z. Kresin. Networks of Josephson junctions and their synchronization. *Physical Review B*, 88(21):214504, December 2013.

[102] Kurt Wiesenfeld, Pere Colet, and Steven H. Strogatz. Frequency locking in Josephson arrays: Connection with the Kuramoto model. *Physical Review E*, 57(2):1563–1569, February 1998.

[103] T. Kashiwagi, H. Kubo, K. Sakamoto, T. Yuasa, Y. Tanabe, C. Watanabe, T. Tanaka, Y. Komori, R. Ota, G. Kuwano, K Nakamura, T. Katsuragawa, M. Tsujimoto, T. Yamamoto, R. Yoshizaki, H. Minami, K. Kadowaki, and R. A. Klemm. The present status of high- T c superconducting terahertz emitters. *Superconductor Science and Technology*, 30(7):074008, 2017.

[104] Dmytro Apalkov, Alexey Khvalkovskiy, Steven Watts, Vladimir Nikitin, Xueti Tang, Daniel Lottis, Kiseok Moon, Xiao Luo, Eugene Chen, Adrian Ong, Alexander Driskill-Smith,

and Mohamad Krounbi. Spin-transfer Torque Magnetic Random Access Memory (STT-MRAM). *J. Emerg. Technol. Comput. Syst.*, 9(2):13:1–13:35, May 2013.

[105] D. C. Ralph and M. D. Stiles. Spin transfer torques. *Journal of Magnetism and Magnetic Materials*, 320(7):1190–1216, April 2008.

[106] J. Z. Sun and D. C. Ralph. Magnetoresistance and spin-transfer torque in magnetic tunnel junctions. *Journal of Magnetism and Magnetic Materials*, 320(7):1227 – 1237, 2008.

[107] Adrien F. Vincent. *Toward using innovative memory devices as artificial synapses in neuro-inspired electronics*. Theses, Université Paris-Saclay, February 2017.

[108] Hyun Seok Choi, Sun Yool Kang, Seong Jun Cho, Inn-Yeal Oh, Mincheol Shin, Hyuncheol Park, Chaun Jang, Byoung-Chul Min, Sang-Il Kim, Seung-Young Park, and Chul Soon Park. Spin nano-oscillator-based wireless communication. *Scientific Reports*, 4, 2014.

[109] Stefano Bonetti, Pranaba Muduli, Fred Mancoff, and Johan Åkerman. Spin torque oscillator frequency versus magnetic field angle: The prospect of operation beyond 65 GHz. *Applied Physics Letters*, 94(10):102507, March 2009.

[110] B. Georges, J. Grollier, V. Cros, and A. Fert. Impact of the electrical connection of spin transfer nano-oscillators on their synchronization: an analytical study. *Applied Physics Letters*, 92(23):232504, June 2008.

[111] Nicolas Locatelli, Abbass Hamadeh, Flavio Abreu Araujo, Anatoly D. Belanovsky, Petr N. Skirdkov, Romain Lebrun, Vladimir V. Naletov, Konstantin A. Zvezdin, Manuel Muñoz, Julie Grollier, Olivier Klein, Vincent Cros, and Grégoire de Loubens. Efficient Synchronization of Dipolarly Coupled Vortex-Based Spin Transfer Nano-Oscillators. *Scientific Reports*, 5:17039, November 2015.

[112] A. A. Awad, P. Dürrenfeld, A. Houshang, M. Dvornik, E. Iacocca, R. K. Dumas, and J. Åkerman. Long-range mutual synchronization of spin Hall nano-oscillators. *Nature Physics*, 13(3):292–299, March 2017.

[113] Shehzaad Kaka, Matthew R. Pufall, William H. Rippard, Thomas J. Silva, Stephen E. Russek, and Jordan A. Katine. Mutual phase-locking of microwave spin torque nano-oscillators. *Nature*, 437(7057):389–392, September 2005.

[114] F. B. Mancoff, N. D. Rizzo, B. N. Engel, and S. Tehrani. Phase-locking in double-point-contact spin-transfer devices. *Nature*, 437(7057):393–395, September 2005.

[115] J. Grollier, V. Cros, and A. Fert. Synchronization of spin-transfer oscillators driven by stimulated microwave currents. *Phys. Rev. B*, 73(6):060409, 2006.

[116] A. D. Belanovsky, N. Locatelli, P. N. Skirdkov, F. Abreu Araujo, K. A. Zvezdin, J. Grollier, V. Cros, and A. K. Zvezdin. Numerical and analytical investigation of the synchronization of dipolarly coupled vortex spin-torque nano-oscillators. *Appl. Phys. Lett.*, 103(12):122405, September 2013.

[117] A. Dussaux, B. Georges, J. Grollier, V. Cros, A. V. Khvalkovskiy, A. Fukushima, M. Konoto, H. Kubota, K. Yakushiji, S. Yuasa, K. A. Zvezdin, K. Ando, and A. Fert. Large microwave generation from current-driven magnetic vortex oscillators in magnetic tunnel junctions. *Nature Communications*, 1:8, April 2010.

[118] Shingo Tamaru, Hitoshi Kubota, Kay Yakushiji, Shinji Yuasa, and Akio Fukushima. Extremely Coherent Microwave Emission from Spin Torque Oscillator Stabilized by Phase Locked Loop. *Scientific Reports*, 5, December 2015.

[119] A. A. Thiele. Steady-State Motion of Magnetic Domains. *Phys. Rev. Lett.*, 30(6):230–233, 1973.

[120] Patrick Villard, Ursula Ebels, Dimitri Houssameddine, Jordan Katine, Daniele Mauri, Bertrand Delaet, Pierre Vincent, Marie-Claire Cyrille, Bernard Viala, Jean-Philippe Michel, JÉrÔme Prouvee, and Frank Badets. A GHz Spintronic-Based RF Oscillator. *IEEE Journal of Solid-State Circuits*, 45(1):214–223, January 2010.

[121] Oleksandr Prokopenko, Elena Bankowski, Thomas Meitzler, Vasil Tiberkevich, and Andrei Slavin. Spin-Torque Nano-Oscillator as a Microwave Signal Source. *IEEE Magnetics Letters*, 2:3000104–3000104, 2011.

[122] H. Eisele. State of the art and future of electronic sources at terahertz frequencies. *Electronics Letters*, 46(26):s8–s11, December 2010.

[123] J. D. Deschênes, L. C. Sinclair, F. R. Giorgetta, W. C. Swann, E. Baumann, I. Coddington, and N. R. Newbury. Synchronization of optical oscillators over a free-space link at the femtosecond level. In *2015 Conference on Lasers and Electro-Optics (CLEO)*, pages 1–2, May 2015.

[124] Roman Lavrov, Michael Peil, Maxime Jacquot, Laurent Larger, Vladimir Udaltsov, and John Dudley. Electro-optic delay oscillator with nonlocal nonlinearity: Optical phase dynamics, chaos, and synchronization. *Physical Review E*, 80(2):026207, August 2009.

[125] Frank C. Hoppensteadt and Eugene M. Izhikevich. Synchronization of laser oscillators, associative memory, and optical neurocomputing. *Physical Review E*, 62(3):4010–4013, September 2000.

[126] Annette F. Taylor, Mark R. Tinsley, Fang Wang, Zhaoyang Huang, and Kenneth Showalter. Dynamical Quorum Sensing and Synchronization in Large Populations of Chemical Oscillators. *Science*, 323(5914):614–617, January 2009.

[127] T. B. DeMarse and K. P. Dockendorf. Adaptive flight control with living neuronal networks on microelectrode arrays. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 3, pages 1548–1551 vol. 3, July 2005.

[128] Shin'ichi Ishiwata, Yuta Shimamoto, and Madoka Suzuki. Molecular motors as an auto-oscillator. *HFSP Journal*, 4(3-4):100–104, June 2010.

[129] R. Rojas and U. Hashagen. *The First Computers: History and Architectures*. History of computing. MIT Press, 2002.

[130] E. Goto. The Parametron, a Digital Computing Element Which Utilizes Parametric Oscillation. *Proceedings of the IRE*, 47(8):1304–1316, 1959.

[131] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Reversible logic gate using adiabatic superconducting devices. *Scientific Reports*, 4:srep06354, September 2014.

[132] Jaijeet Roychowdhury. Boolean Computation Using Self-Sustaining Nonlinear Oscillators. *arXiv:1410.5016 [cs]*, October 2014. 00000 arXiv: 1410.5016.

[133] Yunus Kesim, Darshil Gala, James Bain, and Jeffrey Weldon. Phase Based Boolean Computation Using GeTe6 Oscillators. Pittsburgh, 2017.

[134] I. Mahboob and H. Yamaguchi. Bit storage and bit flip operations in an electromechanical oscillator. *Nature Nanotechnology*, 3(5):275–279, May 2008.

[135] Neeraj Sharma and Lalit M. Aggarwal. Automated medical image segmentation techniques. *Journal of Medical Physics / Association of Medical Physicists of India*, 35(1):3–14, 2010.

[136] Dmitri E. Nikonov, Ian A. Young, and George I. Bourianoff. Convolutional Networks for Image Processing by Coupled Oscillator Arrays. *arXiv:1409.4469 [cond-mat, physics:nlin]*, September 2014. arXiv: 1409.4469.

[137] Y. Fang, M. J. Cotter, D. M. Chiarulli, and S. P. Levitan. Image segmentation using frequency locking of coupled oscillators. In *2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, pages 1–2, July 2014.

[138] K. Yogendra, D. Fan, Y. Shim, M. Koo, and K. Roy. Computing with coupled Spin Torque Nano Oscillators. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 312–317, January 2016.

[139] Dániel Marx. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica Electrical Engineering*, 48(1-2):11–16, 2004.

[140] Abhinav Parihar, Nikhil Shukla, Matthew Jerry, Suman Datta, and Arijit Raychowdhury. Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. *Scientific Reports*, 7(1):911, April 2017.

[141] Chai Wah Wu. Graph coloring via synchronization of coupled oscillators. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 45(9):974–978, September 1998.

[142] Sof Anthony Lee. k-Phase Oscillator Synchronization for Graph Coloring. *Mathematics in Computer Science*, 3(1):61–72, March 2010.

[143] D.E. Nikonov, G. Csaba, W. Porod, T. Shibata, D. Voils, D. Hammerstrom, I.A. Young, and G.I. Bourianoff. Coupled-Oscillator Associative Memory Array Operation for Pattern Recognition. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 1:85–93, 2015.

[144] Dmitri E. Nikonov, Gyorgy Csaba, Wolfgang Porod, Tadashi Shibata, Danny Voils, Dan Hammerstrom, Ian A. Young, and George I. Bourianoff. Coupled-Oscillator Associative Memory Array Operation. *arXiv:1304.6125 [cond-mat]*, April 2013.

[145] N. Shukla, W. Y. Tsai, M. Jerry, M. Barth, V. Narayanan, and S. Datta. Ultra low power coupled oscillator arrays for computer vision applications. In *2016 IEEE Symposium on VLSI Technology*, pages 1–2, June 2016.

[146] P. Maffezzoni, L. Daniel, N. Shukla, S. Datta, and A. Raychowdhury. Modeling and Simulation of Vanadium Dioxide Relaxation Oscillators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(9):2207–2215, September 2015.

[147] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, April 1982.

[148] F.C. Hoppensteadt and E.M. Izhikevich. Associative memory of weakly connected oscillators. In *, International Conference on Neural Networks,1997*, volume 2, pages 1135–1138 vol.2, 1997.

[149] P. Maffezzoni, B. Bahr, Z. Zhang, and L. Daniel. Analysis and Design of Boolean Associative Memories Made of Resonant Oscillator Arrays. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(11):1964–1973, November 2016.

[150] F. C. Hoppensteadt and E. M. Izhikevich. Synchronization of MEMS resonators and mechanical neurocomputing. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(2):133–138, 2001.

[151] R. W. Hölzel and K. Krischer. Pattern recognition with simple oscillating circuits. *New Journal of Physics*, 13(7):073031, July 2011.

[152] Takashi Nishikawa, Ying-Cheng Lai, and Frank C. Hoppensteadt. Capacity of Oscillatory Associative-Memory Networks with Error-Free Retrieval. *Physical Review Letters*, 92(10):108101, March 2004.

[153] Anne Elayne Bernstein. The contributions of Marcel Proust to psychoanalysis. *The Journal of the American Academy of Psychoanalysis and Dynamic Psychiatry*, 33(1):137–148, 2005.

[154] Federico Stella and Alessandro Treves. Associative Memory Storage and Retrieval: Involvement of Theta Oscillations in Hippocampal Information Processing. *Neural Plasticity*, 2011, 2011.

[155] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, June 2011.

[156] R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos. Biometric identification through hand geometry measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1168–1171, October 2000.

[157] Bart Baesens, Rudy Setiono, Christophe Mues, and Jan Vanthienen. Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. *Management Science*, 49(3):312–329, March 2003.

[158] Yan Fang, C.N. Gnegy, T. Shibata, D. Dash, D.M. Chiarulli, and S.P. Levitan. Non-Boolean Associative Processing: Circuits, System Architecture, and Algorithms. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 1:95–103, 2015.

[159] G. Csaba and W. Porod. Computational Study of Spin-Torque Oscillator Interactions for Non-Boolean Computing Applications. *IEEE Transactions on Magnetics*, 49(7):4447–4451, July 2013.

[160] E. Vassilieva, G. Pinto, J. Acacio de Barros, and P. Suppes. Learning Pattern Recognition Through Quasi-Synchronization of Phase Oscillators. *IEEE Transactions on Neural Networks*, 22(1):84–95, January 2011.

[161] Pablo Kaluza. Computation with phase oscillators: An oscillatory perceptron model. 2013.

[162] K. Yogendra, D. Fan, and K. Roy. Coupled Spin Torque Nano Oscillators for Low Power Neural Computation. *IEEE Transactions on Magnetics*, PP(99):1–1, 2015. 00000.

[163] Jacob Torrejon, Mathieu Riou, Flavio Abreu Araujo, Sumito Tsunegi, Guru Khalsa, Damien Querlioz, Paolo Bortolotti, Vincent Cros, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D. Stiles, and Julie Grollier. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664):428–431, July 2017.

[164] Jean C. Coulombe, Mark C. A. York, and Julien Sylvestre. Computing with networks of nonlinear mechanical oscillators. *PLoS ONE*, 12(6), June 2017.

[165] Talitha Weiss, Andreas Kronwald, and Florian Marquardt. Noise-induced transitions in optomechanical synchronization. *New Journal of Physics*, 18(1):013043, 2016.

[166] Damir Vodenicarevic, Nicolas Locatelli, Julie Grollier, and Damien Querlioz. Synchronization Detection in Networks of Coupled Oscillators for Pattern Recognition. Vancouver, 2016.

[167] Roman Borisyuk, Mike Denham, Frank Hoppensteadt, Yakov Kazanovich, and Olga Vinogradova. An oscillatory neural network model of sparse distributed memory and novelty detection. *Biosystems*, 58(1–3):265–272, December 2000.

[168] J. Persson, Yan Zhou, and Johan Akerman. Phase-locked spin torque oscillators: Impact of device variability and time delay. *Journal of Applied Physics*, 101(9):09A503, May 2007.

[169] R. Lebrun, S. Tsunegi, P. Bortolotti, H. Kubota, A. S. Jenkins, M. Romera, K. Yakushiji, A. Fukushima, J. Grollier, S. Yuasa, and V. Cros. Mutual synchronization of spin torque nano-oscillators through a non-local and tunable electrical coupling. *arXiv:1601.01247 [cond-mat]*, January 2016. arXiv: 1601.01247.

[170] D. K. Agrawal, P. Thiruvenkatanathan, J. Yan, and A. A. Seshia. Electrically coupled MEMS oscillators. In *2011 Joint Conference of the IEEE International Frequency Control and the European Frequency and Time Forum (FCS) Proceedings*, pages 1–5, May 2011.

[171] Seung-Bo Shim, Matthias Imboden, and Pritiraj Mohanty. Synchronized Oscillation in Coupled Nanomechanical Oscillators. *Science*, 316(5821):95–99, April 2007.

[172] Grigory V. Osipov, Jürgen Kurths, and Changsong Zhou. *Synchronization in Oscillatory Networks*. Springer Science & Business Media, August 2007.

[173] Vasil Tiberkevich, Andrei Slavin, Elena Bankowski, and Grant Gerhart. Phase-locking and frustration in an array of nonlinear spin-torque nano-oscillators. *Applied Physics Letters*, 95(26):262505, December 2009.

[174] Xia Huang, Jian Gao, Yu-Ting Sun, Zhi-Gang Zheng, and Can Xu. Effects of frustration on explosive synchronization. *Frontiers of Physics*, 11(6):110504, June 2016.

[175] S. Datta, N. Shukla, M. Cotter, A. Parihar, and A. Raychowdhury. Neuro inspired computing with coupled relaxation oscillators. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2014.

[176] Y. Zhou, S. Bonetti, J. Persson, and J. Akerman. Capacitance Enhanced Synchronization of Pairs of Spin-Transfer Oscillators. *IEEE Transactions on Magnetics*, 45(6):2421–2423, June 2009.

[177] A. Slavin and V. Tiberkevich. Nonlinear Auto-Oscillator Theory of Microwave Generation by Spin-Polarized Current. *IEEE Transactions on Magnetics*, 45(4):1875–1918, April 2009.

[178] A. Houshang, E. Iacocca, P. Du?rrenfeld, S. R. Sani, J. Åkerman, and R. K. Dumas. Spin-wave-beam driven synchronization of nanocontact spin-torque oscillators. *Nature Nanotechnology*, advance online publication, December 2015.

[179] Sunil Pratap Singh and Nar Singh. Nonlinear Effects in Optical Fibers: Origin, Management and Applications. *Progress In Electromagnetics Research*, 73:249–275, 2007.

[180] Damir Vodenicarevic, Nicolas Locatelli, Flavio Abreu Araujo, Julie Grollier, and Damien Querlioz. A Nanotechnology-Ready Computing Scheme based on a Weakly Coupled Oscillator Network. *Scientific Reports*, 2017.

[181] D. Querlioz, O. Bichler, A. F. Vincent, and C. Gamrat. Bioinspired Programming of Memory Devices for Implementing an Inference Engine. *Proceedings of the IEEE*, 103(8):1398–1416, August 2015.

[182] R. M. C. Mestrom, R. H. B. Fey, J. T. M. van Beek, K. L. Phan, and H. Nijmeijer. Modelling the dynamics of a MEMS resonator: Simulations and experiments. *Sensors and Actuators A: Physical*, 142(1):306–315, March 2008.

[183] Eva Grimaldi, Antoine Dussaux, Paolo Bortolotti, Julie Grollier, Grégoire Pillet, Akio Fukushima, Hitoshi Kubota, Kay Yakushiji, Shinji Yuasa, and Vincent Cros. Response to noise of a vortex based spin transfer nano-oscillator. *Phys. Rev. B*, 89(10):104404, March 2014.

[184] V. S. Pribiag, I. N. Krivorotov, G. D. Fuchs, P. M. Braganca, O. Ozatay, J. C. Sankey, D. C. Ralph, and R. A. Buhrman. Magnetic vortex oscillator driven by d.c. spin-polarized current. *Nature Physics*, 3(7):498–503, May 2007.

[185] M. Manfrini, T. Devolder, Joo-Von Kim, P. Crozat, C. Chappert, W. Van Roy, and L. Lagae. Frequency shift keying in vortex-based spin torque oscillators. *Journal of Applied Physics*, 109(8):083940, April 2011.

[186] Sumito Tsunegi, Romain Lebrun, Eva Grimaldi, Alex S. Jenkins, Hitoshi Kubota, Kay Yakushiji, Paolo Bortolotti, Julie Grollier, Akio Fukushima, Shinji Yuasa, and Vincent

Cros. Electrical synchronization of spin-torque oscillators driven by self-emitted high frequency current (Conference Presentation). volume 9931, page 99310O, October 2016.

[187] Martin Kreissig, R. Lebrun, F. Protze, K. J. Merazzo, J. Hem, L. Vila, R. Ferreira, M. C. Cyrille, F. Ellinger, V. Cros, U. Ebels, and P. Bortolotti. Vortex spin-torque oscillator stabilized by phase locked loop using integrated circuits. *AIP Advances*, 7(5):056653, February 2017.

[188] E. Grimaldi, R. Lebrun, A Jenkins, A Dussaux, J. Grollier, V. Cros, A Fert, H. Kubota, K. Yakushiji, A Fukushima, R. Matsumoto, S. Yuasa, G. Cibiel, P. Bortolotti, and G. Pillet. Spintronic nano-oscillators: Towards nanoscale and tunable frequency devices. In *2014 IEEE International Frequency Control Symposium*, pages 1–6.

[189] A. Dussaux, E. Grimaldi, B. Rache Salles, A. S. Jenkins, A. V. Khvalkovskiy, P. Bortolotti, J. Grollier, H. Kubota, A. Fukushima, K. Yakushiji, S. Yuasa, V. Cros, and A. Fert. Large amplitude spin torque vortex oscillations at zero external field using a perpendicular spin polarizer. *Appl. Phys. Lett.*, 105(2):022404, July 2014.

[190] K. Yu Guslienko. Magnetic vortex state stability, reversal and dynamics in restricted geometries. *Journal of Nanoscience and Nanotechnology*, 8(6):2745–2760, June 2008.

[191] Q. Mistral, M. van Kampen, G. Hrkac, Joo-Von Kim, T. Devolder, P. Crozat, C. Chappert, L. Lagae, and T. Schrefl. Current-Driven Vortex Oscillations in Metallic Nanocontacts. *Phys. Rev. Lett.*, 100(25):257201, 2008.

[192] A. V. Khvalkovskiy, J. Grollier, A. Dussaux, Konstantin A. Zvezdin, and V. Cros. Vortex oscillations induced by spin-polarized current in a magnetic nanopillar: Analytical versus micromagnetic calculations. *Phys. Rev. B*, 80(14):140401, October 2009.

[193] A. V. Khvalkovskiy, J. Grollier, N. Locatelli, Ya V. Gorbunov, K. A. Zvezdin, and V. Cros. Nonuniformity of a planar polarizer for spin-transfer-induced vortex oscillations at zero field. *Appl. Phys. Lett.*, 96(21):212507–212507–3, May 2010.

[194] A. Dussaux, A. V. Khvalkovskiy, P. Bortolotti, J. Grollier, V. Cros, and A. Fert. Field dependence of spin-transfer-induced vortex dynamics in the nonlinear regime. *Phys. Rev. B*, 86(1):014402, 2012.

[195] T. Kamppeter, F.G. Mertens, A. Sánchez, A.R. Bishop, F. Domínguez-Adame, and N. Grønbech-Jensen. Finite temperature dynamics of vortices in the two dimensional anisotropic Heisenberg model. *Eur. Phys. J. B*, 7(4):607–618, 1999.

[196] A. Dussaux, A. V. Khvalkovskiy, J. Grollier, V. Cros, A. Fukushima, M. Konoto, H. Kubota, K. Yakushiji, S. Yuasa, K. Ando, and A. Fert. Phase locking of vortex based spin transfer oscillators to a microwave current. *Appl. Phys. Lett.*, 98(13):132506–132506–3, March 2011.

[197] R. Lehndorff, D. E. Bürgler, C. M. Schneider, and Z. Celinski. Injection locking of the gyrotropic vortex motion in a nanopillar. *Appl. Phys. Lett.*, 97(14):142503–142503–3, October 2010.

[198] M. Romera, P. Talatchian, R. Lebrun, K. J. Merazzo, P. Bortolotti, L. Vila, J. D. Costa, R. Ferreira, P. P. Freitas, M.-C. Cyrille, U. Ebels, V. Cros, and J. Grollier. Enhancing the injection locking range of spin torque oscillators through mutual coupling. *Applied Physics Letters*, 109(25):252404, December 2016.

[199] James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

[200] SN Sivanandam and SN Deepa. *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.

[201] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[202] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[203] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[204] David Schneider. Deeper and cheaper machine learning [top tech 2017]. *IEEE Spectrum*, 54(1):42–43, 2017.

[205] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown. Overview of the SpiNNaker System Architecture. *IEEE Transactions on Computers*, 62(12):2454–2467, December 2013.

[206] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, August 2014.

[207] Yu-Pu Lin, Christopher H. Bennett, Théo Cabaret, Damir Vodenicarevic, Djaafar Chabi, Damien Querlioz, Bruno Jousselme, Vincent Derycke, and Jacques-Olivier Klein. Physical Realization of a Supervised Learning System Built with Organic Memristive Synapses. *Scientific Reports*, 6:31932, September 2016.

[208] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188, September 1936.

[209] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[210] D. Vodenicarevic, N. Locatelli, and D. Querlioz. A neural network based on synchronized pairs of nano-oscillators. In *2017 IEEE 17th International Conference on Nanotechnology (IEEE-NANO)*, pages 512–514, July 2017.

[211] Gregory S Hornby, Al Globus, Derek S Linden, and Jason D Lohn. Automated antenna design with evolutionary algorithms. In *AIAA Space*, pages 19–21, 2006.

[212] David Abramson and J Abela. A parallel genetic algorithm for solving the school timetabling problem. 1991.

[213] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[214] N. Locatelli, A. F. Vincent, A. Mizrahi, J. S. Friedman, D. Vodenicarevic, J. V. Kim, J. O. Klein, W. Zhao, J. Grollier, and D. Querlioz. Spintronic devices as key elements for energy-efficient neuroinspired architectures. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 994–999, March 2015.

[215] Armin Alaghi and John P. Hayes. Survey of Stochastic Computing. *ACM Trans. Embed. Comput. Syst.*, 12(2s):92:1–92:19, May 2013.

[216] Joseph S. Friedman, Laurie E. Calvet, Pierre Bessiere, Jacques Droulez, and Damien Querlioz. Bayesian Inference With Muller C-Elements. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(6):895–904, June 2016.

[217] Antoni Morro, Vincent Canals, Antoni Oliver, Miquel L. Alomar, and Josep L. Rossello. Ultra-Fast Data-Mining Hardware Architecture Based on Stochastic Computing. *PLOS ONE*, 10(5):e0124176, May 2015.

[218] Tara Julia Hamilton, Saeed Afshar, Andre van Schaik, and Jonathan Tapson. Stochastic electronics: a neuro-inspired design paradigm for integrated circuits. *Proceedings of the IEEE*, 102(5):843–859, 2014.

[219] Chris Winstead, Vincent C Gaudet, Anthony Rapley, and Christian Schlegel. Stochastic iterative decoders. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 1116–1120. IEEE, 2005.

[220] Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

[221] Manan Suri, Damien Querlioz, Olivier Bichler, Giorgio Palma, Elisa Vianello, Dominique Vuillaume, Christian Gamrat, and Barbara DeSalvo. Bio-inspired stochastic computing

using binary CBRAM synapses. *IEEE Transactions on Electron Devices*, 60(7):2402–2409, 2013.

[222] B. D. Brown and H. C. Card. Stochastic neural computation. I. Computational elements. *IEEE Transactions on Computers*, 50(9):891–905, September 2001.

[223] S. Sharifi Tehrani, W. J. Gross, and S. Mannor. Stochastic decoding of LDPC codes. *IEEE Communications Letters*, 10(10):716–718, October 2006.

[224] Sanu K. Mathew, Suresh Srinivasan, Mark A. Anders, Himanshu Kaul, Steven K. Hsu, Farhana Sheikh, Amit Agarwal, Sudhir Satpathy, and Ram K. Krishnamurthy. 2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors. *IEEE Journal of Solid-State Circuits*, 47(11):2807–2821, November 2012.

[225] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki. Nano Meets Security: Exploring Nanoelectronic Devices for Security Applications. *Proceedings of the IEEE*, 103(5):829–849, 2015.

[226] C. Y. Huang, W. C. Shen, Y. H. Tseng, Y. C. King, and C. J. Lin. A Contact-Resistive Random-Access-Memory-Based True Random Number Generator. *IEEE Electron Device Letters*, 33(8):1108–1110, 2012.

[227] Simone Balatti, Stefano Ambrogio, Zhongqiang Wang, and Daniele Ielmini. True random number generation by variability of resistive switching in oxide-based devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 5(2):214–221, 2015.

[228] Yandan Wang, Wei Wen, Hai Li, and Miao Hu. A Novel True Random Number Generator Design Leveraging Emerging Memristor Technology. In *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '15, pages 271–276, New York, NY, USA, 2015. ACM.

[229] M. Hu, Y. Wang, W. Wen, Y. Wang, and H. Li. Leveraging Stochastic Memristor Devices in Neuromorphic Hardware Systems. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(2):235–246, June 2016.

[230] E. Piccinini, R. Brunetti, and M. Rudan. Self-Heating Phase-Change Memory-Array Demonstrator for True Random Number Generation. *IEEE Transactions on Electron Devices*, PP(99):1–8, 2017.

[231] X. Fong, M. C. Chen, and K. Roy. Generating true random numbers using on-chip complementary polarizer spin-transfer torque magnetic tunnel junctions. In *72nd Device Research Conference*, pages 103–104, June 2014.

[232] Akio Fukushima, Takayuki Seki, Kay Yakushiji, Hitoshi Kubota, Hiroshi Imamura, Shinji Yuasa, and Koji Ando. Spin dice: A scalable truly random number generator based on spintronics. *Applied Physics Express*, 7(8):083001, August 2014.

[233] Won Ho Choi, Yang Lv, Jongyeon Kim, A. Deshpande, Gyuseong Kang, Jian-Ping Wang, and C. H. Kim. A Magnetic Tunnel Junction based True Random Number Generator with conditional perturb and real-time output probability tracking. In *2014 IEEE International Electron Devices Meeting*, pages 12.5.1–12.5.4, December 2014.

[234] S. Oosawa, T. Konishi, N. Onizawa, and T. Hanyu. Design of an STT-MTJ based true random number generator using digitally controlled probability-locked loop. In *2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS)*, pages 1–4, June 2015.

[235] M. Barangi, J. S. Chang, and P. Mazumder. Straintronics-Based True Random Number Generator for High-Speed and Energy-Limited Applications. *IEEE Transactions on Magnetics*, 52(1):1–9, January 2016.

[236] Y. Kim, X. Fong, and K. Roy. Spin-Orbit-Torque-Based Spin-Dice: A True Random-Number Generator. *IEEE Magnetics Letters*, 6:1–4, 2015.

[237] A. Sengupta, A. Jaiswal, and K. Roy. True random number generation using voltage controlled spin-dice. In *2016 74th Annual Device Research Conference (DRC)*, pages 1–2, June 2016.

[238] Hochul Lee, Farbod Ebrahimi, Pedram Khalili Amiri, and Kang L. Wang. Design of high-throughput and low-power true random number generator utilizing perpendicularly magnetized voltage-controlled magnetic tunnel junction. *AIP Advances*, 7(5):055934, May 2017.

[239] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester. 16.3 A 23mb/s 23pj/b fully synthesized true-random-number generator in 28nm and 65nm CMOS. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 280–281, February 2014.

[240] William Rippard, Ranko Heindl, Matthew Pufall, Stephen Russek, and Anthony Kos. Thermal relaxation rates of magnetic nanoparticles in the presence of magnetic fields and spin-transfer effects. *Physical Review B*, 84(6):064439, August 2011.

[241] Alice Mizrahi, Nicolas Locatelli, Romain Lebrun, Vincent Cros, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Damien Querlioz, and Julie Grollier. Controlling the phase locking of stochastic magnetic bits for ultra-low power computation. *Scientific Reports*, 6, 2016.

[242] Adrien F. Vincent, Nicolas Locatelli, Jacques-Olivier Klein, Weisheng S. Zhao, Sylvie Galdin-Retailleau, and Damien Querlioz. Analytical Macrospin Modeling of the Stochastic Switching Time of Spin-Transfer Torque Devices. *IEEE Transactions on Electron Devices*, 62(1):164–170, January 2015.

[243] H. Sato, E. C. I. Enobio, M. Yamanouchi, S. Ikeda, S. Fukami, S. Kanai, F. Matsukura, and H. Ohno. Properties of magnetic tunnel junctions with a MgO/CoFeB/Ta/CoFeB/MgO recording structure down to junction diameter of 11 nm. *Applied Physics Letters*, 105(6):062403, August 2014.

[244] Jun Hayakawa, Shoji Ikeda, Young Min Lee, Ryutaro Sasaki, Toshiyasu Meguro, Fumihiro Matsukura, Hiromasa Takahashi, and Hideo Ohno. Current-Induced Magnetization Switching in MgO Barrier Based Magnetic Tunnel Junctions with CoFeB/Ru/CoFeB Synthetic Ferrimagnetic Free Layer. *Japanese Journal of Applied Physics*, 45(10L):L1057, October 2006.

[245] Manuel Blum. Independent unbiased coin flips from a correlated biased source—A finite state markov chain. *Combinatorica*, 6(2):97–108, 1986.

[246] Juan Soto. Statistical testing of random number generators. In *Proceedings of the 22nd National Information Systems Security Conference*, volume 10, page 12. NIST Gaithersburg, MD, 1999.

[247] A. Mizrahi, N. Locatelli, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, J. Kim, J. Grollier, and D. Querlioz. Magnetic stochastic oscillators: Noise-induced synchronization to under-threshold excitation and comprehensive compact model. In *2015 IEEE Magnetics Conference (INTERMAG)*, pages 1–1, May 2015.

[248] H. Sato, T. Yamamoto, M. Yamanouchi, S. Ikeda, S. Fukami, K. Kinoshita, F. Matsukura, N. Kasai, and H. Ohno. Comprehensive study of CoFeB-MgO magnetic tunnel junction characteristics with single- and double-interface scaling down to 1x nm. In *2013 IEEE International Electron Devices Meeting*, pages 3.2.1–3.2.4, December 2013.

[249] Thomas M Maffitt, John K DeBrosse, JA Gabric, Earl T Gow, Mark C Lamorey, John S Parenteau, Dennis R Willmott, Mark A Wood, and William J Gallagher. Design considerations for MRAM. *IBM Journal of Research and Development*, 50(1):25–39, 2006.

[250] W. Zhao, C. Chappert, V. Javerliac, and J. P. Noziere. High Speed, High Stability and Low Power Sensing Amplifier for MTJ/CMOS Hybrid Logic Circuits. *IEEE Transactions on Magnetics*, 45(10):3784–3787, October 2009.

[251] Alexander Neiman, Lutz Schimansky-Geier, Frank Moss, Boris Shulgin, and James J. Collins. Synchronization of noisy systems by stochastic signals. *Physical Review E*, 60(1):284–292, 1999.

[252] K. H. Tsoi, K. H. Leung, and P. H. W. Leong. Compact FPGA-based true and pseudo random number generators. In *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2003. FCCM 2003.*, pages 51–61, April 2003.

[253] J. Holleman, B. Otis, S. Bridges, A. Mitros, and C. Diorio. A 2.92 μW Hardware Random Number Generator. In *2006 Proceedings of the 32nd European Solid-State Circuits Conference*, pages 134–137, September 2006.

[254] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo. A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. *IEEE Transactions on Computers*, 52(4):403–409, April 2003.

[255] Security Requirements for Cryptographic Modules (Std. FIPS 140-1). Technical report, National Institute of Standards and Technology, 1994.

[256] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes. A low-power true random number generator using random telegraph noise of single oxide-traps. In *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, pages 1666–1675, February 2006.

[257] John Von Neumann and Abraham Haskel Taub. *Various Techniques Used in Connection With Random Digits, John von Neumann collected works*, volume 5. Pergamon Press, Oxford, 1963.

[258] You Wang, Hao Cai, Lirida A. B. Naviner, Jacques-Olivier Klein, Jianlei Yang, and Weisheng Zhao. A novel circuit design of true random number generator using magnetic tunnel junction. In *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 123–128, Los Alamitos, CA, USA, 2016. IEEE Computer Society.

[259] S. Tannu and A. Sharma. Low power random number generator using single electron transistor. In *2012 International Conference on Communication, Information Computing Technology (ICCICT)*, pages 1–4, October 2012.

[260] Jeeson Kim, Taimur Ahmed, Hussein Nili, Nhan Duy Truong, Jiawei Yang, Doo Seok Jeong, Sharath Sriram, Damith C. Ranasinghe, and Omid Kavehei. Nano-Intrinsic True Random Number Generation. *arXiv:1701.06020 [cs]*, January 2017. arXiv: 1701.06020.

[261] S. Sahay, A. Kumar, V. Parmar, and M. Suri. OxRAM RNG Circuits Exploiting Multiple Undesirable Nanoscale Phenomena. *IEEE Transactions on Nanotechnology*, PP(99):1–1, 2017.

[262] Igor Kononenko. Inductive and Bayesian Learning in Medical Diagnosis. *Applied Artificial Intelligence*, 7(4):317–337, October 1993.

[263] Irina Rish. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM, 2001.

[264] D. Vodenicarevic, N. Locatelli, A. Mizrahi, J. S. Friedman, A. F. Vincent, M. Romera, A. Fukushima, K. Yakushiji, H. Kubota, S. Yuasa, S. Tiwari, J. Grollier, and D. Querlioz. Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing. *Phys. Rev. Applied*, 8:054045, Nov 2017.

[265] Alexandre Coninx, Pierre Bessière, Emmanuel Mazer, Jacques Droulez, Raphaël Laurent, M Awais Aslam, and Jorge Lobo. Bayesian sensor fusion with fast and low power stochastic circuits. In *Rebooting Computing (ICRC), IEEE International Conference on*, pages 1–8. IEEE, 2016.

[266] Marvin Faix, Raphael Laurent, Pierre Bessiere, Emmanuel Mazer, and Jacques Droulez. Design of Stochastic Machines Dedicated to Approximate Bayesian inferences. *IEEE Transactions on Emerging Topics in Computing*, 2016.

[267] Heikki Maaranen, Kaisa Miettinen, and Antti Penttinen. On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, 37(3):405, March 2007.

[268] George Christopher Williams. *Sex and evolution*. Number 8. Princeton University Press, 1975.

ÉCOLE DOCTORALE
Physique et ingénierie : électrons, photons, sciences du vivant (EOBE)

Titre : Rythmes et Oscillations : une Vision pour la Nanoélectronique

Mots clés : apprentissage automatique, oscillateurs, nanotechnologie, spintronique, classification

Résumé : Avec l'avènement de l'intelligence artificielle, les ordinateurs, appareils mobiles et objets connectés sont amenés à dépasser les calculs arithmétiques et logiques pour lesquels ils ont été optimisés durant des décennies, afin d'effectuer des tâches “cognitives” telles que la traduction automatique ou la reconnaissance d'images et de voix, et pour lesquelles ils ne sont pas adaptés. Ainsi, un supercalculateur peut-il consommer des mégawatts pour effectuer des tâches que le cerveau humain traite avec 20 watt. Par conséquent, des systèmes de calcul alternatifs inspirés du cerveau font l'objet de recherches importantes. En particulier, les oscillations neurales semblant être liées à certains traitements de données dans le cerveau ont inspiré des approches détournant la physique complexe des réseaux d'oscillateurs couplés pour effectuer des tâches cognitives efficacement. Cette thèse se fonde sur les avancées récentes en nano-technologies permettant la fabrication de nano-oscillateurs hautement intégrables pour proposer et étudier de nouvelles architectures neuro-inspirées de classification de motifs exploitant la dynamique des oscillateurs couplés et pouvant être implémentées sur puce.

Title: Rhythms and Oscillations: a Vision for Nanoelectronics

Keywords: machine learning, oscillators, nanotechnology, spintronics, classification

Abstract: With the advent of artificial intelligence, computers, mobile devices and other connected objects are being pushed beyond the realm of arithmetic and logic operations, for which they have been optimized over decades, in order to process "cognitive" tasks such as automatic translation and image or voice recognition, for which they are not the ideal substrate. As a result, supercomputers may require megawatts to process tasks for which the human brain only needs 20 watts. This has revived interest into the design of alternative computing schemes inspired by the brain. In particular, neural oscillations that appear to be linked to computational activity in the brain have inspired approaches leveraging the complex physics of networks of coupled oscillators in order to process cognitive tasks efficiently. In the light of recent advances in nano-technology allowing the fabrication of highly integrable nano-oscillators, this thesis proposes and studies novel neuro-inspired oscillator-based pattern classification architectures that could be implemented on chip.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l’Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France