Analysis and applications of spintronic oscillators in artificial intelligence and combinatorial optimization problems

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-IND/31–ELECTRICAL ENGINEERING

Final Dissertation

Analysis and applications of spintronic oscillators in artificial intelligence and combinatorial optimization problems

---

by Luciano Mazza

Supervisors:

Prof. Mario Carpentieri

Prof. Vito Puliafito

Coordinator of Ph.D. Program:
Prof. Mario Carpentieri

Course n°37, 01/11/2021-31/10/2024

List of abbreviations

ac - Alternating Current
AI - Artificial Intelligence
ANN - Artificial Neural Network
BNN - Binary Neural Network
CNN - Convolutional Neural Network
COP - Combinatorial Optimization Problem
CUDA - Compute Unified Device Architecture
dc - Direct Current
DOM - Degree of Match
DOR - Degree of Rectification
FC - Fully Connected (layer)
FL - Free Layer
GB - Gigabyte
GMR - Giant Magnetoresistance
GPU - Graphics Processing Unit
IM - Ising Machine
IP - In Plane
LLG - Landau-Lifshitz-Gilbert (equation)
LLGS - Landau-Lifshitz-Gilbert-Slonczewski (equation)
LLM - Large Language Model
MAC - Multiply-And-Accumulate
ML - Machine Learning
MOSFET - Metal Oxide Semiconductor Field-Effect Transistor
MRAM - Magnetic Random-Access Memory
MTJ - Magnetic Tunnel Junction
NN - Neural Network
NP - Non-Polynomial
OIM - Oscillator-based Ising Machine
OOP - Out Of Plane
PL - Pinned Layer
RAM - Random-Access Memory
ReLU - Rectified Linear Unit

SAF - Synthetic Antiferromagnet
SHIL - Sub-Harmonic Injection Locking
STD - Spin-Torque Diode
STO - Spin-Torque Oscillator
STT - Spin-Transfer Torque
TB - Terabyte
TMR - Tunnel Magnetoresistance
TSP - Travelling Salesman Problem

Abstract

This thesis investigates the use of spintronic oscillators for artificial intelligence (AI) and combinatorial optimization applications, effectively using their inherent physical nonlinearities to perform complex computational tasks efficiently. Spintronic devices, specifically magnetic tunnel junctions (MTJs), have gained attention due to their low power consumption, compact form and compatibility with silicon substrates making them ideal candidates for devices for analog computing. In this work, MTJ-based oscillators are also analyzed as potential computational units for solving optimization problems implementing an Ising machine.

The study covers the theoretical and practical aspects of using MTJs in AI, particularly focusing on implementing analog multiplication through spin-torque oscillators to reduce computational overhead in neural networks. Through micromagnetic simulations, we show that MTJs can reliably perform analog multiplication. We tested this implementation in a convolutional neural network achieving high accuracies even with device variability, which holds potential for power efficient AI applications.

Furthermore, the thesis explores how these oscillators can solve the Max-Cut problem and similar NP-hard combinatorial optimization challenges by simulating phase dynamics in Ising models. We propose the use of an efficient algorithm that allows for finding good solutions for sparse problems with extremely large sizes. This helps us approaching a problem with 20 million nodes, the largest in literature. The accuracy of the system is tested comparing the performance obtained solving benchmark problems with other reference state-of-the-art solutions.

Finally, this work introduces the use of vortex MTJs for the implementation of memory devices whose polarity can be deterministically written and read with the use of frequency inputs, and can be selectively controlled in a chains without individual access. These devices enable the multiplication between an analog signal, encoded in the power of the input alternated current, and the stored binary value, effectively implementing a building block of a binary neural network. We present experimental results of a chain with two and three cascaded devices.

Contents

Chapter 1

Oscillators, magnetic models and artificial intelligence

The aim of this work is the analysis and application of spintronic oscillators for the acceleration of the computation in the fields of artificial intelligence (AI) and combinatorial optimization problems (COPs).

Regarding AI, we will present a novel implementation corroborated by theoretical analyses and simulations of an analog multiplier that can be obtained simply using an oscillator injected with an alternated current close to its resonance frequency. We used this multiplier in the training and test of a convolutional neural network (CNN) for the recognition of handwritten digits without loosing accuracy.

We will present experimental proof of the use of magnetic tunnel junctions (MTJs) with magnetic vortices for storing and reading binary weights, as well as performing analog multiplication with input signals, all controlled by alternated currents. We realized a prototype of a chain with two and three devices that do not require individual device access, and can be controlled using alternated currents (ac) with specific frequencies.

We will show how the simulation of networks of oscillators and their interactions can be used to find accurate solutions for COPs. We optimized our system for the solution of large and sparse graph problems and we approached a problem with 20 million of nodes, the largest in literature.

The outcomes of this work are derived from a combination of theoretical analyses and experimental results. The theoretical analyses were primarily conducted using micromagnetic models and oscillator models. Micromagnetic models provide insights into the time evolution of the magnetization patterns of a device with a specific magnetic structure and account for various factors, including material properties, shape, applied currents, and magnetic fields. Due to the complexity of these simulations, the behavior can be studied for few hundreds of nanoseconds.

Oscillator models, which require less material-specific information, are much

less computationally expensive and are particularly useful for studying longer interactions between a single device and external signals and the collective behavior of interconnected devices. These models were employed to simulate networks of devices and their interactions.

By integrating these two approaches we gain a comprehensive understanding of the potential capabilities of the analyzed spintronic devices. "All models are wrong, some are useful" [1], is the iconic phrase attributed to the statistician George Box that well describes the necessity of using a model in the right scenario and knowing its limitations, and by integrating the two models we gain a comprehensive understanding of the behavior of spintronic oscillators for this specific use.

1.1 Micromagnetic modeling

Magnetism is a phenomenon that arises from the interactions of electric charges, particularly through the spin and orbital motion of electrons around the atomic nucleus [2]. These interactions generate magnetic moments, which collectively determine the magnetic properties of a material.

The micromagnetic model divides the device into small cells, where it is assumed that all magnetic moments, μ_i, are parallel and aligned in the same direction, allowing them to be represented by a single vector [3, 4]. This assumption holds for cells with a size smaller than the exchange length, defined by:

$$L_{\text{ex}} = \sqrt{\frac{2A}{\mu_0 M_s^2}} \tag{1.1}$$

where A is the exchange stiffness constant (J/m), μ_0 is the permeability of free space ($4\pi \times 10^{-7}$ H/m), and M_s is the saturation magnetization of the material (A/m).

In this work, we consider micromagnetic simulations performed on thin films, corresponding with the (FL) of an MTJ, as with a single layer along the z-axis, and a varying number of horizontal cells, ranging from 50×50 to 200×200.

Micromagnetic simulations are particularly suitable for analyzing devices whose largest dimensions typically range from nanometers to micrometers, extending up to the millimeter scale.

The local magnetization within each cell can be expressed as a continuous vector, dependent on space and time:

$$\mathbf{M}(\mathbf{r}, t) = \frac{1}{dV} \sum_{i=1}^{N} \mu_{\mathbf{i}} \tag{1.2}$$

which represents the density of magnetic moments in a ferromagnetic volume.

The direction of this vector is described by the unit vector:

$$\mathbf{m}(\mathbf{r}, t) = \frac{\mathbf{M}(\mathbf{r}, t)}{M_s} \tag{1.3}$$

1.2 Energy contributions

The micromagnetic model is based on minimizing the total energy of the system, which is a combination of several components. The components relevant to this work will be described in the following sections.

1.2.1 Exchange energy

The exchange energy arises from the quantum mechanical interaction between neighboring spins, which favors parallel alignment to minimize energy. This energy penalizes deviations from uniform magnetization, leading to smoother magnetization configurations over short distances. The exchange energy density is given by:

$$E_{\text{ex}} = A \left(\nabla \mathbf{m}\right)^2 . \tag{1.4}$$

This term plays a crucial role in stabilizing the magnetic structure at the nanoscale ensuring no abrupt changes between the magnetization of adjacent cells.

1.2.2 Anisotropy energy

Anisotropy energy accounts for the preference of magnetic moments to align along specific crystallographic directions, known as easy axes. This energy arises due to spin-orbit coupling, which makes certain orientations of the magnetization energetically favorable. For uniaxial anisotropy, the energy density is given by:

$$E_{\text{ani}} = -K_u \left(\mathbf{m} \cdot \mathbf{e}_u\right)^2 \tag{1.5}$$

where K_u is the anisotropy constant, and $\mathbf{e}_u$ is the unit vector along the easy axis. The anisotropy energy helps to define the preferred magnetization directions within a material. For thin films, where the dimension on the z-axis is much smaller than the other two, we can define xy-plane as the easy plane since the magnetization is not favored along the z direction. This term is minimized when the magnetization vector is parallel to the easy axis, and it is minimized for small variations between adjacent cells.

1.2.3 Magnetostatic energy

Magnetostatic energy, also known as demagnetizing energy, is associated with the self-interaction of the magnetic stray field generated by the magnetization distribution. It tends to oppose the formation of magnetic poles at the surface, encouraging

magnetic moments to align in such a way that reduces the external field. The magnetostatic energy density can be written as:

$$E_{\text{mag}} = -\frac{1}{2}\mathbf{M} \cdot \mathbf{H}_{\text{d}} \tag{1.6}$$

where $\mathbf{H}_{\text{d}}$ is the demagnetizing field, which represents the internal magnetic field produced by the magnetic material itself due to its magnetization $\mathbf{M}$.

The demagnetizing field $\mathbf{H}_{\text{d}}$ acts to reduce the magnetostatic energy by counteracting the magnetization within the material. This field arises because the magnetization $\mathbf{M}$ generates magnetic poles at the surface and edges of the material, creating a stray field outside and within the material. The demagnetizing field thus opposes the magnetization and seeks to minimize surface poles, effectively encouraging configurations that reduce the overall external field.

In practical terms, the demagnetizing field often leads to the formation of complex magnetization patterns, particularly in structures with confined geometries, such as nanostructures. These patterns, including domain formation and vortex states, help minimize the magnetostatic energy by locally aligning magnetic moments in a way that cancels out or significantly reduces stray fields.

1.2.4 Zeeman energy

The Zeeman energy describes the interaction between the magnetization and an externally applied magnetic field. It favors alignment of the magnetic moments with the external field, and its energy density is given by:

$$E_{\text{Z}} = -\mu_0 \mathbf{M} \cdot \mathbf{H}_{\text{ext}} \tag{1.7}$$

where $\mathbf{H}_{\text{ext}}$ is the external magnetic field, and this energy is minimized when the two vectors have the same direction and verse. This term scales with the intensity of the external field.

1.2.5 Oersted field

The Oersted field refers to the magnetic field generated by an electric current passing through or near the magnetic material. This field can interact with the magnetization, influencing the overall energy of the system. The Oersted field $\mathbf{H}_{\text{Oe}}$ is given by Ampere's law:

$$\nabla \times \mathbf{H}_{\text{Oe}} = \mathbf{J} \tag{1.8}$$

where $\mathbf{J}$ is the current density. The Oersted field becomes particularly significant in devices involving current-induced magnetization dynamics.

1.2.6 Thermal field

In real magnetic systems, thermal fluctuations play a significant role especially at the nanoscale, where thermal energy can influence the magnetization dynamics. The thermal field accounts for the random fluctuations in magnetization caused by temperature, and it is particularly important when studying thermally activated processes, such as magnetization reversal and switching.

The thermal field $\mathbf{H}_{\text{th}}$ is typically modeled as a stochastic term in micromagnetic simulations, adding a random perturbation to the effective magnetic field. This random thermal field is incorporated into the LLG equation to simulate the impact of temperature on the magnetization dynamics. The thermal field can be expressed as

$$\mathbf{H}_{\text{th}}(t) = \sqrt{\frac{2\alpha k_B T}{\gamma \mu_0 M_s V dt}}\, \mathbf{G}(t), \tag{1.9}$$

where α is the damping factor, k_B is Boltzmann's constant, T is the temperature in Kelvin, V is the volume of the simulation cell, dt is the time step, $\mathbf{G}(t)$ is a vector of Gaussian-distributed random numbers with zero mean and unit variance.

The thermal field introduces randomness into the micromagnetic model, simulating the effect of thermal noise [5]. This field affects the magnetization over time and is essential for studying phenomena such as thermal stability, and the behavior of magnetic systems at room temperatures.

1.2.7 Effective field

The total energy of the micromagnetic system is obtained by summing all the individual contributions from the different energy terms described previously. The total energy E_{total} can be expressed as:

$$E_{\text{total}} = E_{\text{ex}} + E_{\text{ani}} + E_{\text{mag}} + E_{\text{Z}}. \tag{1.10}$$

Other terms can be included for simulating specific configurations, like the Dzyaloshinskii-Moriya interaction.

The minimization of this total energy determines the equilibrium configuration of the magnetization in the material. The combined effects of these energy terms lead to complex and rich magnetic phenomena, especially in nanoscale devices. Each term contributes to the stability, domain structure, and dynamic behavior of the magnetic system. The interaction between these energy contributions can lead to a variety of magnetization states, such as domains, domain walls, vortices, and other topological features.

In practice, the effective magnetic field $\mathbf{H}_{\text{eff}}$ used in the Landau-Lifshitz-Gilbert (LLG) equation is derived from the total energy by:

$$\mathbf{H}_{\text{eff}} = -\frac{1}{\mu_0}\frac{\delta E_{\text{total}}}{\delta \mathbf{M}} + \mathbf{H}_{\text{Oe}} + \mathbf{H}_{\text{th}}, \tag{1.11}$$

and since the Oersted and thermal components are easier to describe as fields, we can simply add them to the $\mathbf{H}_{\text{eff}}$ evaluation.

This effective field, incorporating all the energy contributions, governs the magnetization dynamics according to the LLG equation. As such, understanding the balance of these energy components is crucial for predicting and controlling the magnetic behavior in simulations and experiments.

1.2.8 Landau-Lifshitz-Gilbert equation

The time evolution of the magnetization is governed by the LLG equation, which describes how the magnetization responds to the effective field derived from the various energy contributions. The LLG equation is expressed as:

$$\frac{d\mathbf{M}}{dt} = -\gamma \mathbf{M} \times \mathbf{H}_{\text{eff}} + \frac{\alpha}{M_s} \mathbf{M} \times \frac{d\mathbf{M}}{dt}. \tag{1.12}$$

The first term describes the precession of the magnetization around the effective field, while the second term represents the damping, which drives the magnetization toward equilibrium. The LLG equation is fundamental in simulating the dynamic behavior of magnetic systems.

1.3 The Spin-Transfer Torque

The Spin-transfer torque (STT) is a critical mechanism in the operation of spintronic devices as it allows the control of the magnetization of the FL with an input current, making the system highly integrable.

STT relies on a spin-polarized current, which is generated by passing an unpolarized current through a hard magnetic material known as a polarizer. The polarizer filters the current based on the electron spin: electrons with spins aligned to the polarizer's magnetization pass through, while those with opposite spins are reflected. This selective filtering of spins results in a spin-polarized current.

As illustrated in Fig. 1.1 (a), this process is essential for transferring angular momentum from the current to the magnetization of the free layer. When the spin-polarized current reaches the FL, typically made of a softer magnetic material, it exerts torque on the magnetization. This torque can rotate the magnetization of the FL, aligning it with the direction of the polarizer's magnetization.

To reverse the magnetization of the FL, we just need to apply a current in the opposite direction and the back-scattering effect will induce a spin-polarized current with opposite polarity. The electrons will transfer their torque to the FL, inducing an anti-parallel alignment compared with the polarizer, as shown in Fig. 1.1 (b).

In Fig. 1.1, the spins are considered to be only polarized up and down for simplicity, but the same concept can be extended to a realistic case, where the spin

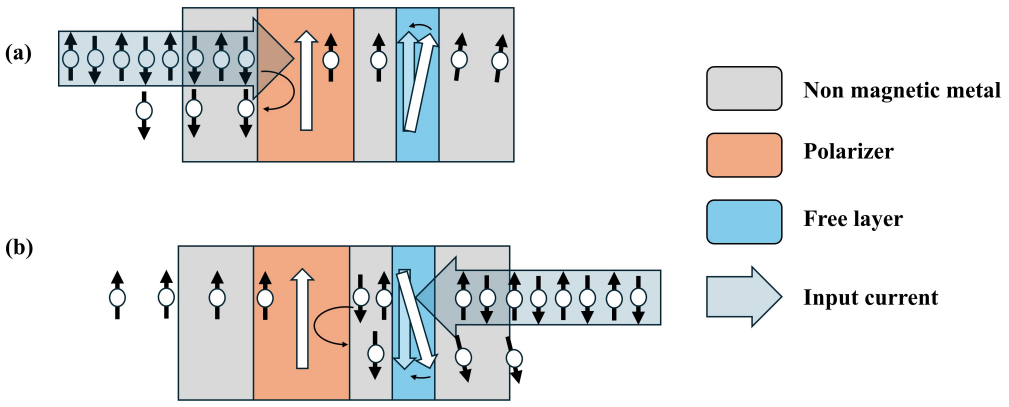

Figure 1.1: (a) Sketch of the spin-polarization of the current polarization and STT aligning the FL with the polarizer. (b) Sketch of the current polarization via back-scattering and anti-aligning the FL with the polarizer.

of the input current has a random orientation in space. The theoretical foundation of STT is described by the Landau-Lifshitz-Gilbert-Slonczewski (LLGS) equation, which extends the LLG equation by incorporating a torque term due to spin-transfer effects. The LLGS equation can be written as

$$\frac{d\mathbf{m}}{dt} = -\gamma \mathbf{m} \times \mathbf{H}_{\text{eff}} + \alpha \mathbf{m} \times \frac{d\mathbf{m}}{dt} + \sigma \mathbf{I}_{\text{dc}} \left[\mathbf{m} \times (\mathbf{m} \times \mathbf{m}_p) - q(\mathbf{m} \times \mathbf{m}_p) \right], \quad (1.13)$$

where σ represents the spin polarization efficiency $\mathbf{m}_p$ is the unit vector of the polarization direction of the spin current. q represents the ratio between the field-like torque and the Slonczewski torque and $\mathbf{I}_{\text{dc}}$ is the current magnitude [6, 7, 8]. This equation is referred to as the Landau-Lifshitz-Gilbert-Slonczewski (LLGS) equation.

In summary, STT allows for the control of the magnetization vector of a magnetic layer with the use of spin-polarized current. This effect is of critical importance for the integrability of spintronic devices, as current integrated chips make use of currents and voltages.

If the damping term is compensated by the input STT component, the system starts oscillating as the LLGS is dominated by the precessing component. The precessional motion is maintained as long as the spin-polarized current is applied, leading to a steady-state oscillation frequency determined by the properties of the magnetic layers and the current magnitude, with frequencies between hundreds of MHz and hundreds of GHz.

1.4 The resistive effects on ferromagnets

In 1988, Albert Fert and Peter Grünberg independently discovered the phenomenon of Giant Magnetoresistance (GMR), which revealed that the electrical resistance of magnetic materials in multi-layered structures depends on the relative orientation of their magnetization [9, 10].

This discovery has been key for the realization and miniaturization of the hard-disk technology, where the digital bits are stored in small magnetic cells, and generally this effect allows for the detection of the magnetization of the free layer with a simple resistance measurement.

1.4.1 Giant Magnetoresistance

In the experiments carried on by Fert and Grünberg the samples were composed of alternating thin film ferromagnetic and non-magnetic metal layers. The resistance of these multilayers changes dramatically with the relative alignment of the magnetizations in adjacent ferromagnetic layers [11].

In a simplified GMR stack with two magnetic layers, the resistance is low when the magnetizations are parallel and high when they are antiparallel. This is due to

the spin-dependent scattering of electrons at the interfaces between ferromagnetic and non-magnetic layers. When the magnetizations are parallel, electrons with spins aligned to the magnetization encounter less scattering, resulting in lower resistance. In the antiparallel alignment, electrons face higher scattering, increasing the resistance.

In a simplified model where the input current is considered as the sum of two spin-polarized currents (spin-up current and spin-down current), the behavior of electron scattering varies depending on the magnetization alignment. In the parallel state, only one of the spin-polarized components is scattered, resulting in lower resistance, as shown in Fig. 1.2 (a). However, in the anti-parallel state, both spin-up and spin-down components experience significant scattering, which increases the overall resistance of the system, Fig. 1.2 (b). In this figure for the purpose of simplification it is assumed that when a polarized current passes through a magnetized material with opposite direction, half of the electrons are scattered, otherwise no scattering happens. The total number of electrons found in the right side of the scheme is inversely proportional to the detected resistance.

The GMR ratio is a useful metric for the evaluation of these devices and it is given by:

$$\mathrm{GMR} = \frac{R_{\mathrm{AP}} - R_{\mathrm{P}}}{R_{\mathrm{P}}}, \tag{1.14}$$

where R_{AP} and R_{P} are the resistances in the anti.parallel and parallel states, respectively. Usual values of GMR are found between 10% and 20% at room temperature [12].

If the magnetization of one of the two layers, typically called the reference layer, is known, determining the magnetization orientation of the other becomes straightforward with a simple resistance measurement. This is a key finding as reading the magnetization of the free layer becomes easy also in integrated implementations.

1.4.2 Tunnel Magnetoresistance

A similar phenomenon to GMR occurs when the two magnetic layers are separated by a thin insulating barrier; this is known as Tunneling Magnetoresistance (TMR). In this case, when the magnetization vectors of the ferromagnetic layers are aligned parallel, the density of states for spin-up and spin-down electrons aligns in both layers. This alignment increases the probability of electrons tunneling through the insulator due to the spin-polarized conduction, resulting in lower resistance.

In the anti-parallel configuration, the density of states for spin-up and spin-down electrons in the two ferromagnetic layers is misaligned. As a result, the tunneling probability decreases, leading to an increase in the overall resistance.

The TMR ratio, the key metric for this phenomenon, is expressed as:

$$\mathrm{TMR} = \frac{R_{\mathrm{AP}} - R_{\mathrm{P}}}{R_{\mathrm{P}}}, \tag{1.15}$$

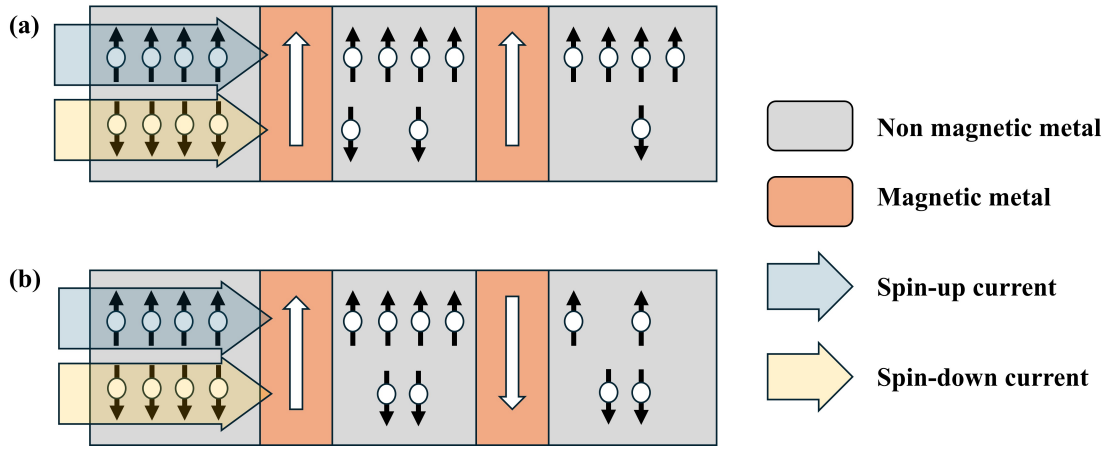

Figure 1.2: Sketch of the GMR phenomenon considering the case of current flowing from the polarizer to the FL (a), and vice versa (b). In this example, when a spin-polarized current passes through a magnetized material with opposite direction, half of the electrons are scattered.

and modern MTJs achieve TMR ratios of over 600%, making them highly effective for applications in magnetic random-access memory (MRAM) and read heads in hard disk drives [13, 14, 15].

We can now define the magnetic tunnel junction (MTJ) structure, a key component in spintronic devices, consisting of a free layer (FL) and a reference layer, separated by a thin insulating barrier, as illustrated in Fig. 1.3 reported from [16]. This image shows also a sketch of the realization of a pinning layer with the use of a synthetic antiferromagnet (SAF), which is composed of two ferromagnetic layers coupled through an antiferromagnetic interaction [17]. The SAF offers several advantages, including reduced magnetic interference and greater stability, making it an ideal substitute for the pinned layer in many applications.

In conclusion, we have demonstrated how the MTJ structure enables the efficient writing and reading of the magnetization state of the FL using only current. This is essential for advancing spintronic oscillators, as we have observed that a dc input can offset the intrinsic damping, inducing sustained magnetization oscillations in the FL. Through the GMR or TMR effect, these oscillations manifest as variations in the device's resistance as when a dc current is applied, the oscillation of the magnetization leads to a variation in the voltage across the device terminals, generating a measurable frequency signal. This variation makes the system easy to control in integrated solutions, making it one of the smallest controllable oscillators in nature [18, 19].

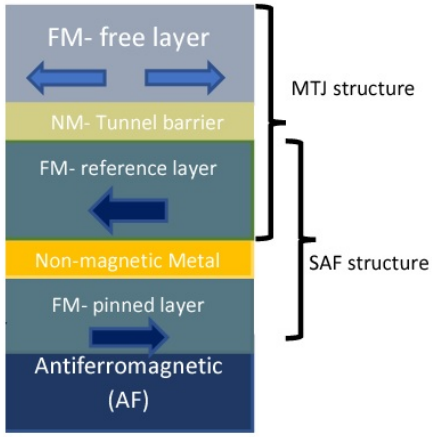

Figure 1.3: (a) Sketch of the structure of the MTJ with a SAF as pinnning layer. Image reported from [16].

1.5 Artificial Intelligence

1.5.1 The advantages of analog computing in AI

With the advent of ChatGPT in late 2022, the world witnessed a profound transformation as artificial intelligence, particularly large language models (LLMs) [20, 21, 22], became part of the daily life of millions of users. These models have quickly established themselves as indispensable tools across various domains, from enhancing customer service interactions to assisting in the generation of scientific papers and creative content [23, 24].

This rapid adoption of LLMs has started a competitive race among leading technology companies to develop increasingly sophisticated models. As the scale and complexity of these models grow, so does the demand for computational power, leading to significant energy consumption and resource requirements. The focus on developing more powerful models, however, has highlighted the inefficiencies inherent in current computing architectures. Considering the frequency of use, the most time-consuming and energy-intensive operations in AI implementations are often associated with memory read and write processes, as well as the multiplication of numerical values.

In response to these challenges, academic and industrial research are trying to identify and implement more efficient computational solutions [19, 25]. In this work we will analyze two main areas of exploration for the optimization of neural networks (NNs): the realization of analog multiplication and in-memory computing. The first has a critical impact on the carbon footprint, and the second one could also drastically reduce the operation times as devices can be used both as memory and computing units, enabling operations like the multiply-and-accumulate (MAC) directly within the memory. The added cost is the increased complexity of manufacturing and control of analog signals.

1.5.2 Machine Learning and Deep Learning

Machine learning (ML) is a subset of AI that describes the systems that learn from data and make predictions or decisions without being explicitly programmed for every task. At its core, ML revolves around training models on datasets to recognize patterns and relationships, enabling these models to apply generalized patterns to unseen data. The field is commonly divided into three main types: supervised learning, where the model learns from labeled data [26]; unsupervised learning, which relies on unlabeled data to identify structures and patterns [27]; and reinforcement learning, where agents learn by interacting with an environment and receiving feedback in the form of rewards or penalties [28, 29].

Deep learning, a specialized area within machine learning, represents a significant advancement, driven by the development of artificial neural networks (ANNs).

Deep learning models automatically learn features through multiple layers of neurons (hence the word deep), enabling them to excel in complex tasks like image recognition and natural language processing. The advent of deep learning required the realization of large datasets and the availability of extended computational power, and this is the reason why, even though this field was theorized in late 1950s [30], practical implementations have been developed in recent years such the development of autonomous driving cars, medical images interpreters, text and image generation and many others [31, 32]. Central to deep learning are multilayered networks where each layer extracts increasingly abstract features from the data, leading to more accurate predictions.

In the following, we will introduce linear layers, useful for extracting general information over large inputs, and CNNs [33], suitable for analyzing data structures organized in grids, like images.

1.5.3 Fully Connected layers

Fully Connected (FC) layers, also known as linear or dense layers, are fundamental components of NNs. A FC layer performs a linear transformation by multiplying the input vector by a weight matrix and adding a bias term. Mathematically, this is represented as:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{1.16}$$

where $\mathbf{x}$ is the input, $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, and $\mathbf{y}$ is the output. This transformation is crucial because it maps the input data into a new space, allowing the network to learn different representations and extract important features from the data.

In most cases, except for the input and output layers, the output passes through an activation function that introduces nonlinearities in the network. The introduced nonlinearities is the key difference between NNs and linear transformations and those are necessary for the learning of complex patterns, like image and language features, and for strengthening the connections between related neurons. The connections implement the interactions between neurons and their impact on the processing of the information.

A simple and effective activation function is the ReLU (rectified linear unit), that returns the input value for positive inputs and zero for negative inputs, or in simpler terms:

$$\mathrm{ReLU}(x) = \max(0, x). \tag{1.17}$$

Figure 1.4 shows a sketch of a deep fully connected neural network composed only of linear layers. The network is divided into input, hidden and output layers.

Each connection represents a weight and each circle in the hidden layers represents the application of the activation function on the sum of all the outputs from

the previous layer multiplied by a specific weight. On the right is presented a zoom of the framed portion representing how the weights link neurons in adjacent layers.

The output of the neuron with position x of the hidden layer l can be defined as

$$\mathbf{L}_l(x) = \text{ReLU}\left(\mathbf{b}_l(x) + \sum_{i=1}^{N} \mathbf{L}_{l-1}(i) \cdot \mathbf{W}_{l-1}(i,x)\right). \tag{1.18}$$

where $\mathbf{W}_{l-1}$ is the matrix containing all the weights that connect the layer l with the previous one, $\mathbf{L}_{l-1}$ is the vector with the outputs of the previous layer and $\mathbf{b}_l$ is the vector containing the biases of nodes of the specific layer.

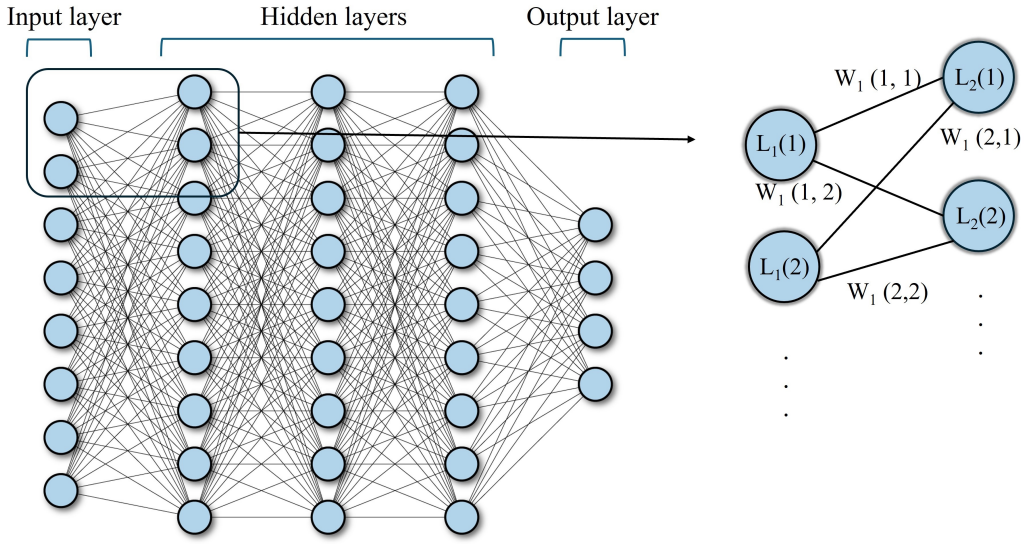

Figure 1.4: Sketch of a NN composed of an input layer, three hidden layers and an output layer. Each connection represents a weight and each circle in the hidden layers represents the application of the activation function. On the right is represented a zoom of the framed portion with the explicit evaluation of the values of L_{21} and L_{22}, with W_{11} and W_{12} being the weights and B_{21} and B_{22} being the biases. Image source (modified) [34].

Linear layers play a key role in neural networks by connecting all the neurons from one layer to those in the next, making them particularly powerful for combining information learned from different parts of the input. They enable the network to make complex predictions by aggregating and synthesizing features learned by previous layers [35].

In deep learning models, FC layers are often found at the end of the network, following convolutional or recurrent layers.

The importance of FC layers lies in their capacity to consolidate learned features into meaningful outputs. While other types of layers (such as convolutional layers)

are specialized for processing specific data types (e.g., spatial data in CNNs), linear layers are versatile and can be used for a wide range of tasks, such as classification, regression, and sequence generation. Their simplicity and efficiency make them flexible and essential for the decision-making process in neural networks.

Despite their straightforward operation, FC layers are computationally intensive when dealing with high-dimensional data, as they require a large number of parameters. However, they remain a crucial part of most deep learning models, balancing complexity and generalization, and ensuring that the network can interpret the information extracted by specialized layers, and for this reason in the following example we will observe a FC layer analyzes the output of two convolutional layers.

1.5.4 Convolutional Neural Networks

CNNs are specialized NNs designed to process grid-like data, such as images. They use layers that focus on detecting spatial hierarchies in data, making them highly effective for tasks like image recognition and object detection [36].

At the heart of a CNN are its convolutional layers, which apply filters (also called kernels) to input data to detect features like edges, textures, or patterns. These filters move across the input data, performing element-wise operations that produce feature maps. A major advantage of convolutional layers is weight sharing, where the same filter is applied across different parts of the input. This makes CNNs more efficient than fully connected layers and gives them the ability to detect features regardless of their position within the image, providing translation invariance.

Figure 1.5 describes and example of convolution of a 3×3 filter applied to a 5×5 matrix. We can observe that each element of the final matrix is higher in value if the analyzed window matches the filter. In this way, the network can study the images and recognize different patterns. The size of each dimension of the output matrix in this case is $5 - (3 - 1)/2$, and in general $S_i - (S_w - 1)/2$, where S_i is the size of the input matrix and S_w the size of the weight matrix. A zero-padding frame is usually added to small images to avoid this reduction of dimensionality.

The values contained in each filter are chosen by the network during training.

Following the convolutional layers, pooling layers (like max pooling) reduce the spatial dimensions of the feature maps, retaining the most important information while making the network more computationally efficient and reducing overfitting. Pooling helps CNNs focus on high-level abstract features while preserving the essential spatial relationships of the data.

Activation functions are applied after each convolution to introduce non-linearity into the model, essential to model non-linear relationships in data, further enhancing their capacity to identify intricate features.

The final layers of a CNN often include FC layers, which take the high-level features extracted by the convolutional and pooling layers and map them to output categories, such as finding the right class for an input image. These layers are

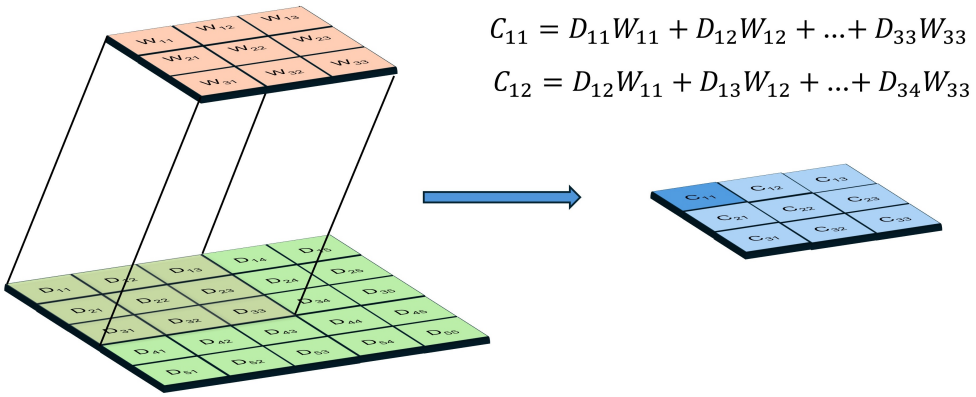

Figure 1.5: Sketch of the first step of the convolution between a 3×3 filter W (orange) by a 5×5 data matrix D (green) resulting in a result 3×3 matrix C (blue). The following values are obtained shifting the filter horizontally and vertically through the whole data matrix D.

critical for consolidating the learned features and producing the final decision or prediction.

The importance of CNN layers lies in their ability to progressively learn features at different levels of abstraction, from simple edges in the early layers to more complex shapes and objects in deeper layers. This hierarchical structure has made CNNs the backbone of many modern AI applications, including computer vision, video analysis, and even tasks outside of visual data, such as text analysis and game AI.

By allowing the network to focus on spatial hierarchies and reduce computational complexity, CNN layers enable models to efficiently handle high-dimensional data, contributing significantly to the success of AI in areas such as medical imaging, autonomous driving, and facial recognition.

1.5.5 An example of a CNN applied for the recognition of handwritten images

We present a practical example of a CNN applied to the problem of handwritten digit recognition. The network is trained on the MNIST dataset [37], which consists of 60,000 images of handwritten digits from 0 to 9, each represented as a grayscale image of size 28×28 pixels. The goal of the CNN is to classify each image into one of the 10 possible digit classes (from 0 to 9).

The architecture can be described as a sequence of transformations applied to the input image and is composed of the following layers with the size of the output data in each step represented in square brackets:

- **Input Layer:** The input to the network is a 28×28 grayscale image ì$[28 \times 28]$.
- **Convolutional Layer:** The first convolutional layer applies sixteen 3×3 filters to the input image. The images are zero-padded in order to compensate for the reduction of dimensionality after the convolution. Each filter scans the input image and detects low-level features, such as edges or textures. The result is a set of feature maps, where each feature map corresponds to a filter. The idea behind this layer is that the network is able to adapt the filters during the training phase such that the system is able to recognize the local features of handwritten numbers, like the diagonal line for the 7, or the roundness of the 8. $[28 \times 28 \times 16$, one image per filter$]$.
- **ReLU Activation:** A ReLU activation function is applied element-wise to the feature maps. ReLU introduces non-linearity to the model, allowing the CNN to capture complex patterns $[28 \times 28 \times 16]$.
- **Pooling Layer:** After applying ReLU, a max-pooling operation is used to reduce the dimensionality of the feature maps, making the network more

computationally efficient and reducing overfitting. The pooling layer reduces the size of the feature maps by selecting the maximum value in each region of a fixed size (in this case, 2×2) [$14 \times 14 \times 16$].

- **Second Convolutional Layer:** The images are zero-padded again and a second convolutional layer with 16 filters, each of size $14 \times 14 \times 16$ is applied to the pooled feature maps, this layer extracts higher-level features from the image, such as the fact that the 8 is composed of two circles, or that the zero is a large circle [$14 \times 14 \times 16$].
- **ReLU Activation:** Another ReLU activation function is applied to the output of the second convolutional layer, introducing further non-linearity [$14 \times 14 \times 16$].
- **Pooling Layer:** A second max-pooling layer is applied [$7 \times 7 \times 16$].
- **Flatten Layer:** The output of the set of 16 two-dimensional feature maps is flattened into a one-dimensional vector to serve as input for the fully connected layer [784].
- **Fully Connected Layer:** The flattened feature vector is passed through a fully connected (linear) layer. This layer connects every neuron to every neuron in the previous layer. The fully connected layer has 10 outputs, corresponding to the 10 digit classes (0 to 9) [10].

The CNN is trained using the backpropagation algorithm, which calculates the gradient of the loss function with respect to each weight by applying the chain rule through the network. During backpropagation, the error is propagated from the output layer back to the input layer, updating the weights of the network to minimize the overall loss.

This process allows the model to learn from its mistakes by adjusting the weights to reduce the difference between the predicted output and the true labels.

The output of the model is a score (also named logit) for each class representing if the input image has specific features associated with that class. This output can be translated to the confidence of the network that each specific class is the correct output. This is evaluated as

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \tag{1.19}$$

where $z = [z_1, z_2, \ldots, z_n]$ is the score vector and C is the number of classes.

The optimization of the model is done using the Adam optimizer, a variant of gradient descent that combines the advantages of two popular algorithms: AdaGrad and RMSProp [38]. After backpropagating what is the influence of each weight

of the network on the studied output, the Adam optimizer adapts the learning rate (a parameter that describes the magnitude of the variation applied to each weight) based on both the first moment (mean) and the second moment (uncentered variance) of the gradient. This process allows the model to learn from its mistakes by adjusting the weights to reduce the difference between the predicted output and the true labels and the adaptive learning rate leads to faster and more stable convergence during training, especially in large datasets or complex models. The final aim is always to minimize the loss.

The loss function used is categorical cross-entropy, which measures the difference between the predicted probability distribution and the true distribution of the target labels. The cross-entropy loss for a given sample is defined as:

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{1.20}$$

y_i is the true label (represented as a one-hot encoded vector), and p_i is the predicted probability (the softmax output) for class i. The goal of training is to minimize this loss, reducing the difference between the predicted and true distributions.

The dataset is usually divided into three parts: training, validation, and test sets. The training set is used to train the model by feeding it into the network over multiple iterations (epochs) to adjust the model's parameters. The validation set is used during training to monitor the model's performance and tune hyperparameters, such as the learning rate or number of layers, without influencing the model's parameters directly. Finally, the test set is reserved for the final evaluation of the model, providing an unbiased estimate of its performance on unseen data, ensuring the model's ability to generalize.

This example, illustrated in Fig. 1.6, shows the steps to implement a classification task using a CNN; this process results being very computationally expensive as during the training all the images of the dataset are passing through the whole network for multiple epochs. Except from some logarithmic and exponential evaluations, the whole training and test of the network is formed by sums and multiplications. The analog implementation of the MAC would provide a significant improvement in the optimization power and chip area of AI-specific devices. In this work we propose two solutions for the improvement of the efficiency of NNs: one for the implementation of the analog multiplication obtained simulating spin-torque oscillators tested with the presented CNN structure, and an experimental prototype for the implementation of the MAC between analog inputs and binary weights for applications in binary neural networks (BNN).

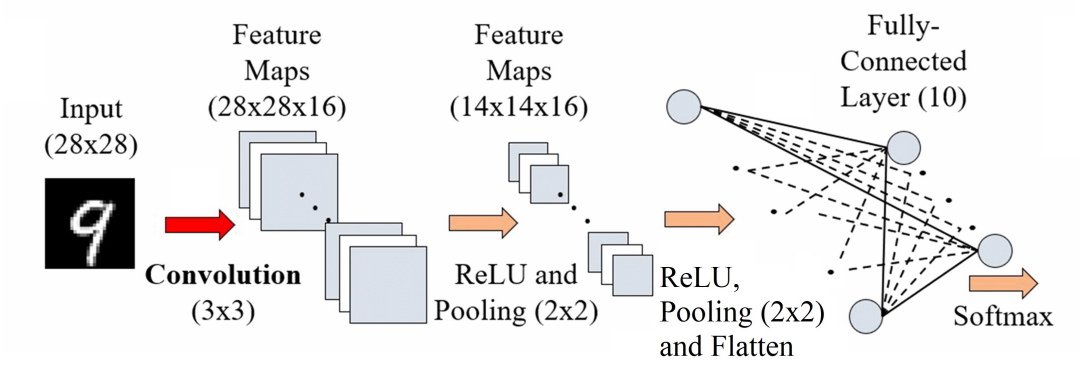

Figure 1.6: Sketch of an exemplary CNN.

Chapter 2

Spintronic Oscillators for Analog Multiplication

This chapter describes the application of spintronic oscillators for implementing analog multiplication between two numbers and shows an application within the machine learning domain.

As later described in detail, given any parabolic phenomenon, with few measurements and a constant scaling operation it is possible to multiply any two numbers. In this chapter we observe two ways of obtaining a parabolic phenomenon using the interaction between two oscillators and between an oscillator and an external current source.

The aim of applying these mechanisms is to achieve a rapid and low-power consumption implementation, leveraged by the implementation of nanometric-scale spintronic oscillators that would act as co-processing units in conventional digital devices.

To validate the effectiveness of this approach, the system has been analyzed through a combination of micromagnetic simulations and experimental data. These simulations provide a detailed understanding of the spintronic oscillator's behavior under various conditions, while experimental data offer real-world insights into the system's performance. Furthermore, the practical application of this technology has been demonstrated through its use in a handwritten digit recognition task, which is a standard benchmark in machine learning.

2.1 Analog multiplication with a parabola

The aim of this paragraph is the derivation of the multiplication of any two numbers F and G given an ideal parabolic phenomenon $P(X) = aX^2 + bX + c$. When a signal composed by the difference of two F and G is applied to the parabolic

function $P(X)$, we obtain

$$P(F-G) = aF^2 + aG^2 - 2aFG + bF - bG + c. \tag{2.1}$$

We can substitute $P(F)$ and $P(G)$

$$P(F-G) = P(F) + P(-G) - 2aFG - c, \tag{2.2}$$

and from this equation it is easy to extract the FG product

$$FG = \frac{P(F-G) - P(F) - P(-G) + c}{-2a}. \tag{2.3}$$

In summary, from three measurements of a parabolic phenomenon, $P(F-G)$, $P(F)$ and $P(-G)$, it is possible to obtain the analog multiplication of any two numbers rescaling the output with constant values.

This is the key concept that will be used in the following part of the chapter, where the parabolic function will be implemented using the difference between two dc currents injected into spintronic devices, and the output will be a dc voltage measured at the ends of the devices.

2.2 Modeling the devices

We will model the device using a theoretic framework that comprehensively takes in consideration the nonlinear features of spintronic devices. This model will be referred in the rest of the work as Slavin model for the scientist who devised it [39].

2.2.1 Slavin model of a single oscillator

The model defines the behavior of a single oscillator with the complex variable $c(t)$ characterized by an amplitude $p(t)$ and a phase $\phi(t)$ which defines the oscillation as:

$$c(t) = \sqrt{p(t)}e^{j\phi(t)}. \tag{2.4}$$

The oscillator system is analyzed as a reactive component, with a positive and a negative damping. In a circuital analogy, these correspond to a inductor-capacitor couple, a positive and a negative resistance. The oscillation is observed when the positive and negative component compensate each other and the circuit is completely reactive, as represented in Fig. 2.1. In magnetic terms, this can be achieved compensating the damping of the magnetization with spin-transfer torque such that the system can precess indefinitely. Considering Eq. 1.13, this condition is obtained when

$$\alpha(\mathbf{m} \times (\mathbf{m} \times \mathbf{h}_{\text{eff}})) = \sigma \mathbf{I}_{\text{dc}} \left[\mathbf{m} \times (\mathbf{m} \times \mathbf{m}_{\text{p}}) - q(\mathbf{m} \times \mathbf{m}_{\text{p}})\right]. \tag{2.5}$$

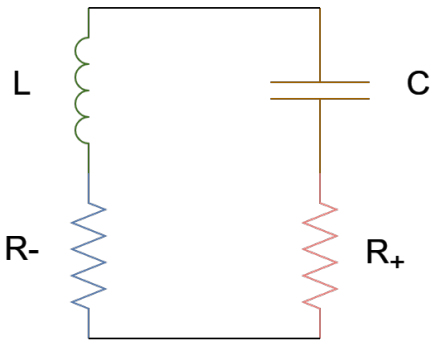

Figure 2.1: Equivalent circuit of the tunnel-diode oscillator. When the device is self oscillating, the resistive components compensate each other. Figure adapted from [39].

There is usually not a single value of I_{dc} that satisfies this condition, but a range of currents [40, 41].

The oscillation is described by the variation of the $c(t)$ variable through time, and it is defined as:

$$\frac{\partial c}{\partial t} + j\omega(|c|^2)c + \Gamma_+(|c|^2)c - \Gamma_-(|c|^2)c = f(t). \tag{2.6}$$

In this equation ω represents the natural frequency of the device, and Γ_+ and Γ_- correspond with the positive and negative damping. The term $f(t)$ represents an external stimulus such the application of an input current or the influence of a coupled oscillator. In absence of an external stimulus this term is null.

The dependence of the frequency ω on the power of the oscillations is modulated through the nonlinear frequency shift parameter N

$$\omega(|c|^2) = \omega_0 + N|c|^2, \tag{2.7}$$

which can be calculated as $N = 8\pi\gamma M_0$ [39].

2.2.2 Slavin model of two interacting oscillators

Considering a system of two spintronic oscillators (namely 1 and 2) in presence of a communication channel, each device exerts a slight influence on the other, which can be characterized with $f(t)$ in 2.6. The influence from one device on the other is analyzed as an external force with a coupling parameter Ω determining the amplitude, and a phase shift parameter β that takes into account the reciprocal latency needed for the signal generated in one device to reach the other.

In this case, the expression that determines the oscillation of a system with two devices is:

$$\frac{\partial c_1}{\partial t} + j\omega_1(|c_1|^2)c_1 + \Gamma_{+,1}(|c_1|^2)c_1 - \Gamma_{-,1}(|c_1|^2)c_1 = \Omega_{1,2}e^{j\beta_{1,2}}c_2. \tag{2.8}$$

The equation for the second oscillator is obtained by substituting each subscript 1 with 2 and vice versa.

In our following analyses we will consider $\Omega_{i,j} = \Omega_{j,i} = \Omega$ and $\beta_{i,j} = \beta_{j,i} = \beta$ representing a symmetric system. An active version of this system has been recently realized experimentally with signal amplifiers and controlling the phase shift [42].

2.2.3 From complex amplitude to power and phase

The variables c_1 and c_2 that determine the oscillations are a complex variables and can be decomposed in oscillation powers and phases, as in Eq. 2.4. This exemplification makes the system easier to analyze with ordinary differential equation

(ODE) solvers, and helps studying the phase-locking behavior between two or more devices.

In the following part this derivation is presented.

Considering Eq. 2.4, the time derivative of c_1 can be defined as

$$\frac{\partial c_1}{\partial t} = \frac{\partial p_1}{\partial t}\frac{e^{j\phi_1(t)}}{2\sqrt{p_1(t)}} + j\sqrt{p_1(t)}e^{j\phi_1(t)}\frac{\partial \phi_1}{\partial t}, \tag{2.9}$$

hence this can be substituted, together with Eq. 2.4, in Eq. 2.8 obtaining:

$$\begin{aligned}\frac{\partial p_1}{\partial t}\frac{e^{j\phi_1(t)}}{2\sqrt{p_1(t)}} + j\sqrt{p_1(t)}e^{j\phi_1(t)}\frac{\partial \phi_1}{\partial t} + j\omega_1(p_1)\sqrt{p_1(t)}e^{j\phi_1(t)} + \Gamma_{+,1}(p_1)\sqrt{p_1(t)}e^{j\phi_1(t)} \\ -\Gamma_{-,1}(p_1)\sqrt{p_1(t)}e^{j\phi_1(t)} = \Omega\sqrt{p_2(t)}e^{j(\phi_2(t)+\beta)}\end{aligned} \tag{2.10}$$

and the whole equation is multiplied by $\frac{2\sqrt{p_1(t)}}{e^{j\phi_1(t)}}$, obtaining

$$\begin{aligned}\frac{\partial p_1}{\partial t} + 2jp_1(t)\frac{\partial \phi_1}{\partial t} + 2j\omega_1(p_1)p_1(t) + 2\Gamma_{+,1}(p_1)p_1(t) - 2\Gamma_{-,1}(p_1)p_1(t) = \\ 2\Omega\sqrt{p_1(t)p_2(t)}e^{j(\phi_2(t)-\phi_1(t)+\beta)}.\end{aligned} \tag{2.11}$$

The exponential components can be simplified using Euler's formula

$$e^{j\alpha} = \cos\alpha + j\sin\alpha \tag{2.12}$$

which is applied to Eq. 2.11 such that the two complex components can be extracted and evaluated independently. The real part represents the oscillation power

$$\frac{\partial p_1}{\partial t} = -2p_1(t)(\Gamma_{+,1}(p_1) - 2\Gamma_{-,1}(p_1)) + 2\Omega\sqrt{p_1(t)p_2(t)}\cos(\phi_2(t) - \phi_1(t) + \beta), \tag{2.13}$$

and the imaginary part the phase

$$\frac{\partial \phi_1}{\partial t} = -\omega_1(p_1) + \Omega\sqrt{\frac{p_2(t)}{p_1(t)}}\sin(\phi_2(t) - \phi_1(t) + \beta). \tag{2.14}$$

In summary, for a system with two oscillators, the evolution of the powers and phases will be dictated by the following equations:

$$\frac{\partial p_1}{\partial t} = 2\Omega\sqrt{p_1(t)p_2(t)}\cos(\phi_1(t) - \phi_2(t) - \beta) - 2(\Gamma_{+,1} - \Gamma_{-,1})p_1(t), \tag{2.15}$$

$$\frac{\partial \phi_1}{\partial t} = -\omega_1(p_1) - \Omega\sqrt{\frac{p_2(t)}{p_1(t)}}\sin(\phi_1(t) - \phi_2(t) - \beta), \tag{2.16}$$

$$\frac{\partial p_2}{\partial t} = 2\Omega\sqrt{p_1(t)p_2(t)}\cos\left(\phi_1(t) - \phi_2(t) - \beta\right) - 2(\Gamma_{+,2} - \Gamma_{-,2})p_2(t), \tag{2.17}$$

$$\frac{\partial \phi_2}{\partial t} = -\omega_2(p_2) - \Omega\sqrt{\frac{p_1(t)}{p_2(t)}}\sin\left(\phi_2(t) - \phi_1(t) - \beta\right). \tag{2.18}$$

These equations can be extended for the case of n oscillators influencing each other that will be useful in the following chapter, and the i^{th} element is represented as

$$\frac{\partial p_i}{\partial t} = -2(\Gamma_{+,i} - \Gamma_{-,i})p_i(t) + \sum_{j=1, j\neq i}^{n} 2\Omega\sqrt{p_i(t)p_j(t)}\cos\left(\phi_i(t) - \phi_j(t) - \beta\right), \tag{2.19}$$

$$\frac{\partial \phi_i}{\partial t} = -\omega_i(p_i) - \sum_{j=1, j\neq i}^{n} \Omega\sqrt{\frac{p_j(t)}{p_i(t)}}\sin\left(\phi_i(t) - \phi_j(t) - \beta\right). \tag{2.20}$$

2.2.4 An external signal injected to an oscillator

When an external signal with amplitude f_e and frequency ω_e is applied to the device, whether through a superimposed ac current or a magnetic field, if its frequency falls within a specific range close to the natural resonance frequency of the oscillator, the device becomes influenced by this external signal. As a result, the oscillator's frequency locks onto the external frequency, a phenomenon known as injection locking, where the device synchronizes to the injected signal.

This effect is widely utilized in modern technological applications [43, 44], as it enables a frequency-stable output, enhances output power, and reduces phase noise. These nonlinear properties make injection locking particularly valuable from a computing standpoint. In this work, we use this effect to implement the analog multiplication between two values.

Starting from Eq. 2.6, the application of an external signal with amplitude f_e and angular frequency ω_e to a spintronic oscillator can be modeled as

$$\frac{\partial c}{\partial t} + j\omega(|c|^2)c + \Gamma_+(|c|^2)c - \Gamma_-(|c|^2)c = f_e e^{-j\omega_e t}. \tag{2.21}$$

Following the previous derivation, we can decompose the power and phase of the oscillations as:

$$\frac{\partial p}{\partial t} = -2(\Gamma_+ - \Gamma_-)p(t) + 2\sqrt{p(t)}f_e\cos(\phi + \omega_e t), \tag{2.22}$$

$$\frac{\partial \phi}{\partial t} = -\omega(p) - \frac{f_e}{\sqrt{p(t)}}\sin(\phi + \omega_e t). \tag{2.23}$$

These equations can be easily extended to the case of having an injected signal in a system with n interacting devices, and will be useful in the following chapter.

In such case we have:

$$\frac{\partial p_i}{\partial t} = -2(\Gamma_{+,i} - \Gamma_{-,i})p_i(t) + \sum_{j=1,j\neq i}^{n} 2\Omega\sqrt{p_i(t)p_j(t)}\cos\left(\phi_i(t) - \phi_j(t) - \beta\right) + 2\sqrt{p_i(t)}f_e\cos(\phi_i + \omega_e t), \tag{2.24}$$

$$\frac{\partial \phi_i}{\partial t} = -\omega_i(p_i) - \sum_{j=1,j\neq i}^{n} \Omega\sqrt{\frac{p_j(t)}{p_i(t)}}\sin\left(\phi_i(t) - \phi_j(t) - \beta\right) - \frac{f_e}{\sqrt{p_i(t)}}\sin(\phi_i + \omega_e t). \tag{2.25}$$

2.3 The degree of match

Having established that any parabolic phenomenon can facilitate analog multiplication, this section presents an initial analysis of how to achieve such parabolic behavior using spin-torque oscillators (STOs) using the degree of match (DOM)[45, 46, 47].

The DOM is a valuable mathematical tool used to assess the degree of frequency locking between two complex oscillating variables and it is defined as

$$DOM(t) = \frac{1}{2}|c_1(t) + c_2(t)| \tag{2.26}$$

and in this paragraph it is observed that for STOs it is parabolic in a frequency region named locking bandwidth.

2.3.1 Simulation parameters

The parameters of the two simulated oscillators have been chosen to represent devices used in current state-of-the-art implementations [48], and are listed in Table 2.1.

Considering two coupled STOs with similar fabrication characteristics, when the same input current is applied, they will begin oscillating at similar frequencies [49, 50]. If the coupling is sufficiently strong, the two devices will synchronize, meaning they will oscillate at exactly the same frequency.

To achieve this locking, the natural frequencies of the devices must lie within the locking bandwidth which is a specific range of frequencies [39, 51].

Since the oscillation frequency is dependent on the input dc current, for two identical devices with parameters listed in Table 2.1, changing the input current will result in a shift in the operating frequency, and the locking range can be observed by applying different currents.

Parameter	Value	Parameter	Value
Ω	10^7	$N/2\pi$	-3.44GHz
$\omega_0/2\pi$	4.2GHz	V	$85 * 140 * 1.8\text{nm}^3$
β	-1.63π	M_s	9500Oe
Q	2.66	γ	$2.21 \times 10^5\text{m/C}$
$\Gamma_g/2\pi$	252MHz	σ	$2.5 \times 10^{12}\text{S/m}$

Table 2.1: Parameters used in the simulations.

2.3.2 Multiplication and locking bandwidth

The input currents for the following analyses are centered around $2I_{\text{th}}$. Specifically, to evaluate how the two devices interact under different input currents, the input of one device will be kept fixed while the input of the other device will vary. To simplify this evaluation, we will analyze the difference between supercriticalities as input, defined as $\Delta\xi = (I_1 - I_2)/I_{\text{th}}$.

Figure 2.2 (a) shows the final values of several DOM analyses, computed as a function of $\Delta\xi$, where a clearly regular behavior defines the locking region. Outside this region, the behavior is random. This distinction is further illustrated in Figure 2.2 (b), which shows the time evolution of three DOM curves for synchronized oscillators (solid lines) and unsynchronized oscillators (dashed lines). Each line represents a single case, and the final values from these analyses are used in Figure 2.2 (a). When the oscillators are synchronized, the DOM converges and the converging values are higher when the natural frequencies of the devices are matching, for $\Delta\xi = 0$. Instead, outside the locking bandwidth, the DOM does not converge and the final value is random.

Figure 2.2 (c) provides a close-up view of the DOM peak (for $0.01 < \Delta\xi < 0.0078$) shown in (a), comparing the numerical DOM with an ideal parabola. The two curves closely overlap, yielding a correlation coefficient of $r = 99.95\%$. This region of the DOM curve is highly effective for performing multiplication, as demonstrated in Figure 2.2 (d), which compares 10^4 examples of random multiplications computed using the DOM (blue dots) with ideal multiplication (red line), showing excellent agreement. The inset reveals a slightly asymmetric error distribution, with a root mean square error of $e_{\text{RMS}} = 0.003$.

2.3.3 The phase shift between oscillators

One of the most influential parameters in the analysis of Eqs. 2.15-2.18 is the phase shift β. This phase shift primarily depends on the type of interaction and the delay

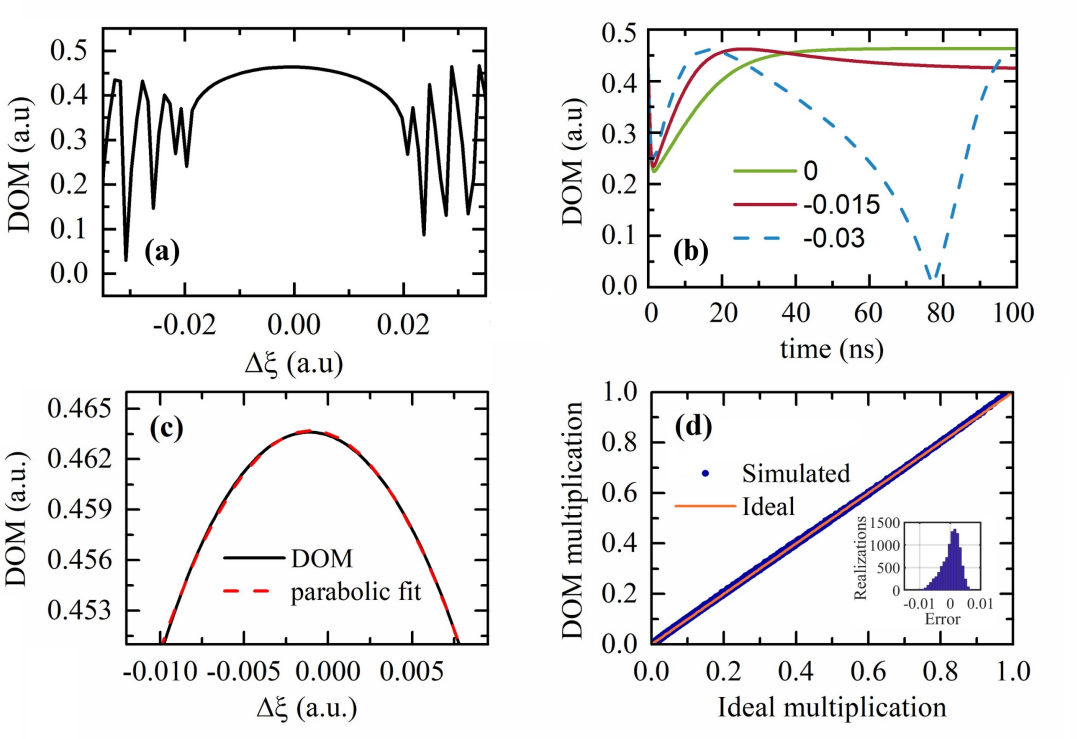

Figure 2.2: Analysis of the DOM for two coupled STOs. (a) DOM vs. $\Delta\xi$ exhibiting a parabolic behavior within the locking range. (b) Representation of multiple DOM analyses over time, where each line corresponds to a different $\Delta\xi$ input, represented in the legend. Continuous lines indicate cases of frequency locking between the oscillators. (c) Close-up view of (a), showing a clear second-order behavior in the DOM (solid black curve) compared to an ideal parabolic fit (dashed red curve). (d) Representation of 10^4 multiplications obtained using the simulated DOM (blue dots), with the ideal result shown as the bisector of the first quadrant (orange line). The inset displays the error distribution. Figure adapted from [48].

of the coupling signal [39]. In the case of devices sharing the same substrate it can be adjusted by altering the distance between the oscillators or by incorporating a delay line between them, making it an additional degree of freedom during the design process.

The DOM described in Eq. 2.2 (a) exhibits a distinct parabolic behavior when the oscillators are locked in-phase. However, for opposite-phase locking, a similar pattern is observed subtracting the two complex variables in Eq. 2.26. Figure 2.3 shows how the locking bandwidth, defined as difference in supercriticality, varies with respect to the parameter β for both in-phase and opposite-phase DOM analyses. The figure indicates that for $\beta = -1.64\pi$ and $\beta = -0.63\pi$ there is a pronounced peak, corresponding with a large bandwidth in terms of differential input supercriticality, making these values optimal. The first value, in particular, has been selected for Fig. 2.2. Figures 2.3 (b) and (c) present two different DOM evaluations for $\beta = -1.43\pi$, a suboptimal value, and $\beta = -1.13\pi$, the worst-case scenario observed. The supercriticality bandwidth was calculated by determining the maximum distance between relative minima in the DOM plot. The optimal value $\beta = -1.64\pi$ is used for the analysis in the subsequent sections.

2.3.4 Device mismatching

In former analyses, it was assumed that the two oscillators were identical. However, it is crucial to examine the DOM when dealing with two different interacting oscillators to test if the system is robust to device mismatching, a common problem in the manufacturing of spintronic devices. Specifically, we focused on the nonlinear frequency shift coefficient N.

Figure 2.4 (a) compares the DOM for two cases: $N_1 = N_2$ shown in orange, these are the values used in Fig. 2.2 (a), and $N_1 = 1.05N_2$ shown in blue. The two curves share similar characteristics, with the primary difference being a shift in the input current required to achieve the locking range.

Figure 2.4 (b) illustrates the frequency curves for these two configurations, considering both equal and differing values of N_1, as a function of $\Delta\xi$. With N_2 fixed, it is evident that outside the locking region, the oscillating frequency of this device remains roughly constant. The frequency of the first device increases linearly outside the locking range as expected since $\Delta\xi$ is proportional to its input current. In this context, a variation on the nonlinear frequency shift N for an oscillator results in a shift in its frequency curve, which in turn causes a shift in the locking region. Similar results were observed when considering larger differences in the N coefficients.

This analysis confirms the validity of the DOM method for devices with varying characteristics.

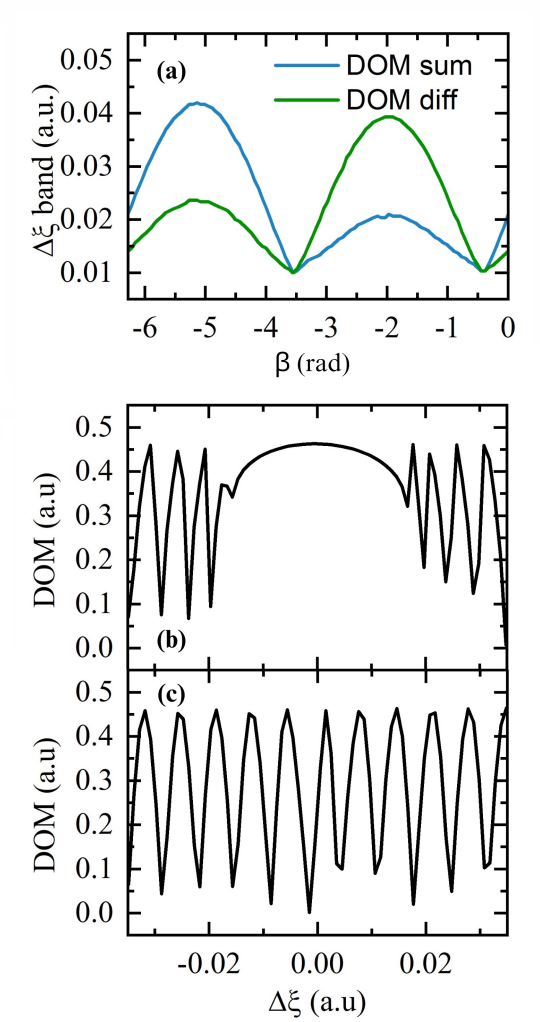

Figure 2.3: (a) Analysis of the supercriticality bandwidth as a function of the phase shift β.(b) DOM observed for the suboptimal value of $\beta = -1.43\pi$. (c) DOM observed for the worst-case scenario with $\beta = -1.13\pi$. Figure adapted from [48].

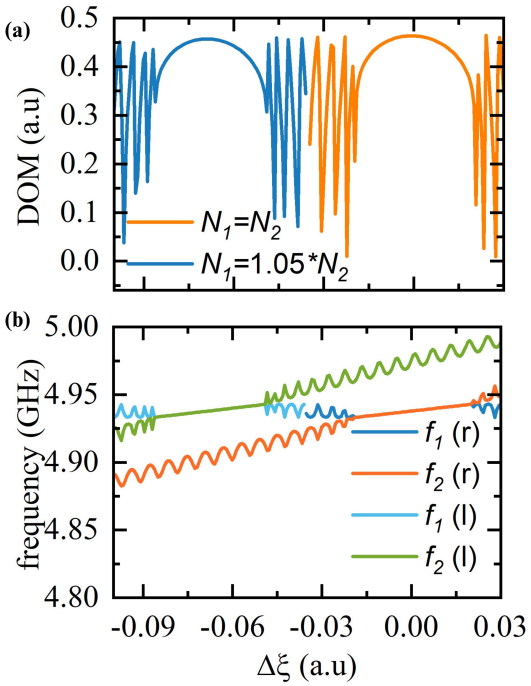

Figure 2.4: (a) A comparison of the DOM for $N_1 = N_2$ (as shown in Fig. 2.2 (a)) in orange, and for $N_1 = 1.05N_2$ in blue. (b) Frequencies of the two oscillators for $N_1 = N_2$ (left) and $N_1 = 1.05N_2$ (right). The light and dark blue curves represent the first oscillator, which exhibits a constant oscillating frequency outside the locking region. The orange and green curves represent the frequency of the second oscillator, whose frequency changes together with the input $\Delta\xi$. Figure adapted from [48].

2.3.5 Application of thermal noise

The DOM has also been computed with thermal noise included in the model as an additional Gaussian stochastic term with zero mean and unit variance. The amplitude of this noise is given by

$$D_n = \sqrt{\frac{2\Gamma_+(p)k_BT}{L_{ex}\omega(p)}} \tag{2.27}$$

where k_B is Boltzmann's constant, T is the temperature, and L_{ex} is the exchange length [52]. We analyzed temperature values up to 400 K and found that, within this parameter space, the thermal noise does not influence the parabolic trend of the DOM. The correlation coefficient remains $r_{\text{noise}} = 99.95\%$ at 400 K.

In conclusion, the DOM is a useful tool for evaluating the degree of synchronization of two oscillators, which resulted being much resistant to device variation and the application of thermal noise. However, even though it has a clear parabolic behavior, it is not practical for the computation of the analog multiplication as it requires the detection of both the amplitude and phase from the two oscillators, which is a particularly challenging task for integrated devices.

2.4 The Degree of Rectification

The Degree of Rectification (DOR) characterizes how effectively a single oscillator synchronizes with an external ac input. When a dc current is applied to a Spin-Torque Oscillator (STO), it induces an oscillatory behavior. If an alternating input signal is introduced and its frequency falls within the oscillator's locking range, the STO will lock its frequency to the external input, due to the injection locking [53, 54].

Similar to the Degree of Match (DOM), the locking bandwidth can also be observed in relation to the applied dc input. Within this range, the oscillator's frequency remains constant and matches the external signal, although the phase difference between the two signals varies for different inputs. There are two main differences with the previous case:

- Only one device and an external signal are required, halving the device requirements;

- The phase variation influences the rectification properties of the device, resulting in an output dc voltage at the oscillator terminals that depends on the input current in a parabolic manner [55]. This process simplifies the reading of the parabolic signal in an integrated circuit.

As this phenomenon describes quantitatively how much the two signals are matched and this results in a rectification voltage, we named it "Degree of Rectification".

This section presents an analysis of the DOR based on micromagnetic simulations, which are then compared with experimental data and applied to a practical implementation in a CNN for the recognition of handwritten digits [56, 57].

2.4.1 Micromagnetic analyses

In this section we present the frequency and phase behavior of the free layer of an MTJ when an alternating current is applied.

The analyzed device is a hybrid MTJ, illustrated in Fig. 2.5 (a), composed of an out of plane (OOP) FL (1.63-nm-thick $Co_{20}Fe_{60}B_{20}$) and an in-plane (IP) polarizer (synthetic antiferromagnet $Co_{70}Fe_{30}$ (2.3 nm)/Ru (0.85 nm)/$Co_{40}Fe_{40}B_{20}$ (2.4 nm)) exchange biased by a PtMn (15 nm) layer. The device is patterned with an elliptical cross-section (150×60 nm^2) and its resistances in the parallel and antiparallel states are $R_P = 640\ \Omega$ and $R_{AP} = 1200\ \Omega$, respectively. An additional advantage of this device is its zero-field operation [53].

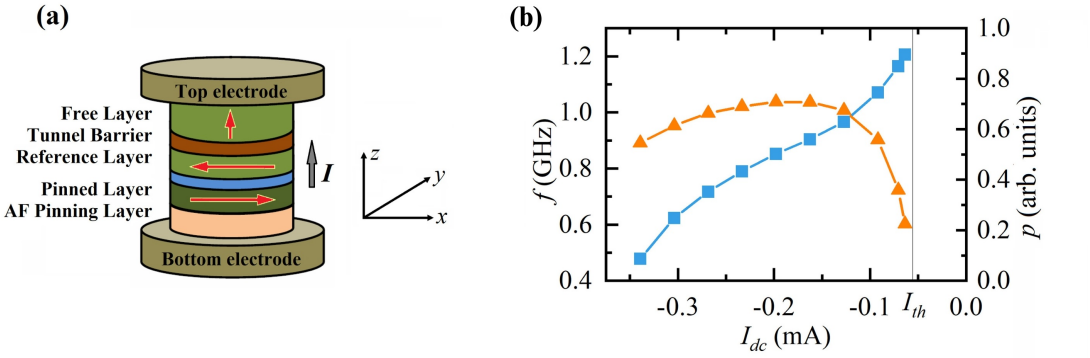

Figure 2.5: (a) Sketch of the MTJ. Simulated frequency (blue squares) and powers (orange triangles) of the oscillations for various dc input currents. Figure adapted from [55].

To explain the concept of the DOR, we perform micromagnetic simulations of the MTJ's FL magnetization by numerically solving an adaptation of Eq. 1.13 [53, 6, 58], and the spin polarization efficiency is defined as

$$\sigma = \frac{g|\mu_B|}{|e|\gamma_0 M_S^2 V_{\mathrm{FL}}}, \tag{2.28}$$

where g is the gyromagnetic splitting factor, μ_B is the Bohr magneton, e is the electron charge, and V_{FL} is the volume of the free layer. The total current flowing through the MTJ is given by

$$I = I_{\mathrm{dc}} + I_{\mathrm{ac,max}} \sin(2\pi f_{\mathrm{ac}} + \phi_{\mathrm{ac}}), \tag{2.29}$$

and we consider a null ϕ_{ac}.

2.4.2 Simulation results

Fig 2.5 (b) shows the oscillation frequencies (orange triangles) and powers of the magnetization (blue squares) of self-oscillations when an above-threshold dc input is applied. This step is key to determine the threshold current $|I_{\text{th}}| = 0.056\,\text{mA}$ and the working frequencies for the subsequent application of an ac current. The negative currents are due to the simulation of the polarizer aligned over the $-x$ axis.

The nonlinear frequency shift parameter N is evaluated as $\frac{df_0}{dp_0} \approx -411\,\text{MHz}$. In the oscillation regime, as the magnetoresistance varies at the same frequency as the ac current, we observe a rectification voltage [59, 60].

Within the specific locking range, variations in the input dc current do not change the frequency but modify the amplitude of the oscillating magnetization $dm_X(I_{\text{dc}})$, which is related to the oscillator power, p, by $dm_X = \sqrt{p}$. The intrinsic phase shift, $\phi(I_{\text{dc}})$, between the ac current and the oscillating signal is also influenced by the dc current [61]. The output voltage can be calculated using [53]:

$$V_{\text{dc}} = \frac{(R_{\text{AP}} - R_{\text{P}})\sqrt{p} I_{\text{ac,max}}}{4} \cos[\phi(I_{\text{dc}})]. \tag{2.30}$$

Considering Eq. 2.30 we can notice that the relation between the dc voltage and the phase is cosinusoidal and not parabolic. There is a close relation between the cosine and the parabola for angles close to zero, π or multiples of π, and to apply this transformation we can use the Taylor-Mc Laurin expansion of the cosine which, truncated to the second term, is

$$\cos(\phi) \approx 1 - \frac{\phi^2}{2}. \tag{2.31}$$

This is valid for angles close to zero or even multiples of π, and a similar formulation can be found for odd multiples of π. Fig. 2.6 shows a comparison of the cosine (blue), parabola (orange) and their difference (green), which is close to zero for angles between -0.5 and 0.5 radians. This effectively means that for small angles, all the considerations previously made about computing with parabolic terms can be extended to cosinusoidal (and sinusoidal) phenomena.

We can now approximate the dc voltage as a function of the angle

$$V_{dc} \approx K(1 - \frac{\phi^2}{2}), \tag{2.32}$$

where K is the multiplying factor of Eq. 2.30, and we want to define a relation between the dc output voltage and the the dc input current.

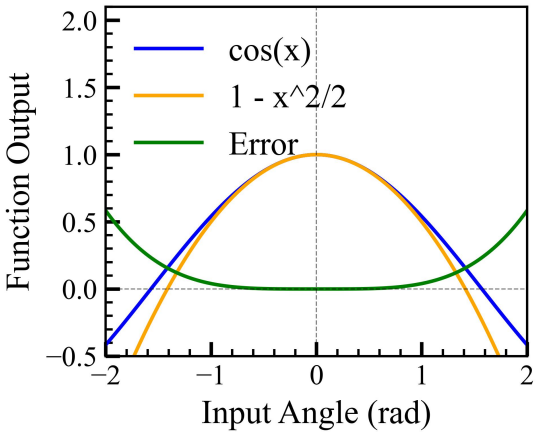

Figure 2.6: Plot of the cosine (blue) and parabolic function (orange) $1 - \frac{\phi^2}{2}$. Their difference is reported with a green line.

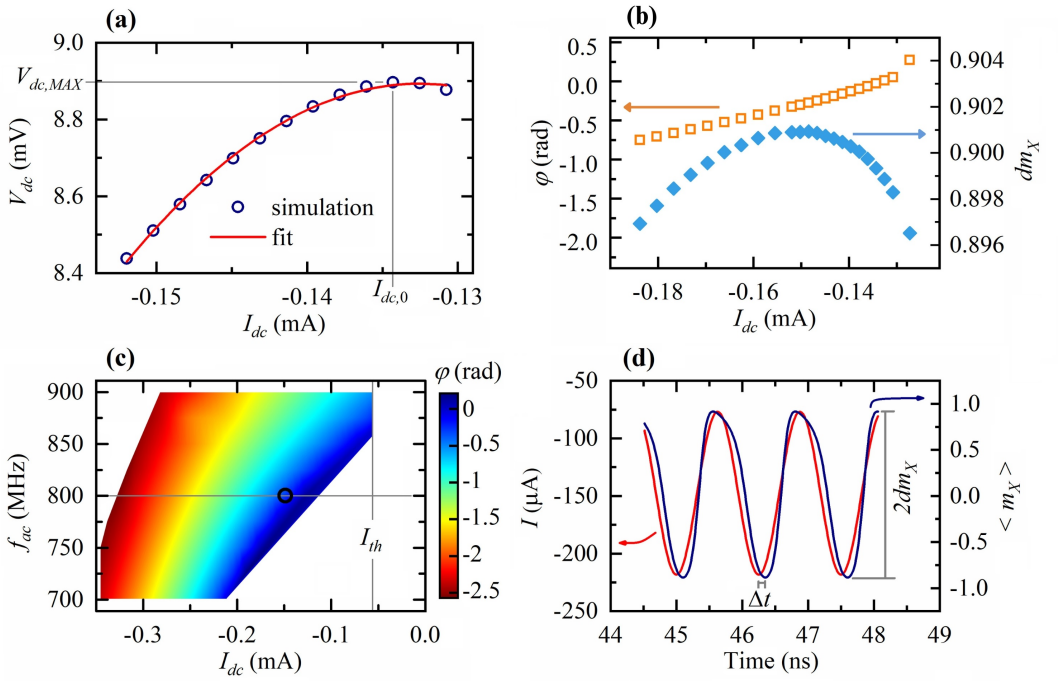

Figure 2.7: (a) Rectified dc voltage as a function of the applied dc current for the spin-torque diode, with $I_{\mathrm{ac,MAX}} = 70.7\,\mu\mathrm{A}$ and $f_{\mathrm{ac}} = 800\,\mathrm{MHz}$. The circles represent the results from micromagnetic simulations, while the solid line shows the corresponding parabolic fit. (b) The intrinsic phase shift (empty squares) and the amplitude of the magnetization along the x-axis (filled diamonds) are shown as a function of the dc current for the same $I_{\mathrm{ac,MAX}}$ and f_{ac} as in panel (a). (c) Phase diagram of the intrinsic phase shift plotted as a function of both the microwave frequency and the dc current, with $I_{\mathrm{ac,MAX}} = 70.7\,\mu\mathrm{A}$. The vertical line indicates the threshold current for auto-oscillation, $|I_{\mathrm{th}}| = 0.056\,\mathrm{mA}$, while the horizontal line marks the microwave frequency used in panels (a) and (b). (d) Time-domain traces of the applied current (left y-axis) and the spatially averaged x-component of the magnetization $\langle m_X \rangle$ (right y-axis), corresponding to the working point marked by the circle in panel (c). The time delay Δt between the two traces is also indicated in the figure. Figure adapted from [55].

Figure 2.7 (a) provides an example of rectification voltage obtained for $I_{\mathrm{ac,max}} = 70.7\,\mu\mathrm{A}$ and $f_{\mathrm{ac}} = 800\,\mathrm{MHz}$. The maximum voltage is achieved for $I_{\mathrm{dc},0} = -0.134\,\mathrm{mA}$, corresponding to a phase shift close to zero (see Fig. 2.7 (b)), where the additional phase shift due to the polarizer orientation is not considered). Figure 2.7 (b) shows dm_X (filled diamonds) and ϕ (empty squares) for the simulations in Fig. 2.7 (a).

The intrinsic phase shift, obtained for different input currents when applying an external ac signal, exhibits a quasi-linear dependence on the dc current, with minor deviations near the edge of the locking region, similar to what is observed in [61, 62]. The amplitude of magnetization shows a weak dependence on the current, which is expected for an oscillator with a large nonlinear frequency shift. The power p of the injection-locked oscillator is described by

$$\frac{p}{p_0} = 1 + \frac{\sqrt{\sigma} I_{\mathrm{ac,max}}}{1 + (N/P\xi)^2}, \tag{2.33}$$

where $\xi = I_{\mathrm{dc}}/I_{\mathrm{th}}$ is the supercriticality of the dc bias current and P is the effective damping rate [63]. For the studied device, N/P exceeds 15, resulting in a reduced dependence of the oscillator power on I_{dc}, as shown in Fig. 2.7 (b) (blue diamonds), where a variation of less than 3% is observed in dm_X. We can conclude that the variation of the phase shift is the dominant effect in Eq. 2.30.

The quasi-linear trend of the intrinsic phase shift shown in Fig. 2.7(b) can be approximated by $\phi(I_{\mathrm{dc}}) = mI_{\mathrm{dc}} + n$. The fitting parameters are identified from the rectified voltage as follows. The maximum rectified voltage, $V_{\mathrm{dc,max}} \approx K$, is achieved at $I_{\mathrm{dc},0}$ (see Fig. 2.7 (a) and (b)), where ϕ is zero, leading to $n = -mI_{\mathrm{dc},0}$.

We can now substitute the linear phase-current relation in Eq. 2.32 obtaining

$$V_{dc} \approx K(1 - \frac{mI_{dc}^2 + 2mnI_{dc} + n^2}{2}). \tag{2.34}$$

We can use this relation to calculate m, since

$$\left.\frac{d^2 V_{\mathrm{dc}}}{dI_{\mathrm{dc}}^2}\right|_{I_{\mathrm{dc}}=I_{\mathrm{dc},0}} = -m^2 V_{\mathrm{dc,max}}, \tag{2.35}$$

and the other terms of this equation are easy to extract from experimental data.

Finally, knowing m and n, we obtain the relation between the output dc voltage and the input dc current

$$V_{\mathrm{dc}}(I_{\mathrm{dc}}) = aI_{\mathrm{dc}}^2 + bI_{\mathrm{dc}} + c, \tag{2.36}$$

where the coefficients are given by:

$$a = -\frac{1}{2} V_{\mathrm{dc,max}} m^2, \quad b = V_{\mathrm{dc,max}} m^2 I_{\mathrm{dc},0}, \quad c = V_{\mathrm{dc,max}} \left\{ 1 - \frac{(mI_{\mathrm{dc},0})^2}{2} \right\}. \tag{2.37}$$

This parabolic relation, can be utilized in future works to easily estimate the DOR behavior and yields results that closely align with experimental results, as can be seen in Figure 2.8, that illustrates a comparison between the parabolic fit of the micromagnetic data shown in Fig. 2.7 (a) and the parabola derived from analytically evaluated parameters based on the experimental data presented in [53], demonstrating excellent agreement. The parameters are reported in Tab. 2.2.

In order to consider the resistive effects of the device on the rectified signal, an additional term proportional to the dc current is included in the rectified voltage, leading to a linear shift in the parabolic equation:

$$V_{\mathrm{dc}}(I_{\mathrm{dc}}) = aI_{\mathrm{dc}}^2 + (b + R_{\mathrm{dc}})I_{\mathrm{dc}} + c, \tag{2.38}$$

where R_{dc} represents the variation in dc resistance induced by the microwave input.

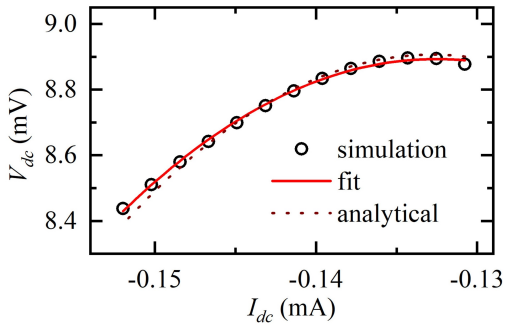

Figure 2.8: Comparison of V_{dc} values derived from micromagnetic simulations (circles), the parabolic fit (solid line), and the parabola obtained using analytical data (dashed line). The parameters of the fit and the analytical analysis are reported in Tab. 2.1. Figure reported from [55].

Parabolic DOR: $V_{dc}(I_{dc}) = aI_{dc}^2 + bI_{dc} + c$		
Parameters	**Fit**	**Analytical**
$\mathbf{a(mV/mA)^2}$	-1.225×10^3	-1.397×10^3
b(mV/mA)	-325	-371
c (mV)	-12.6	-15.7

Table 2.2: Parameters obtained fitting experimental result and analytically using Eq. 2.37, and applied in the analytical curve represented in Fig. 2.8.

In order to implement a multiplier using spintronic diodes, it is essential for the devices to operate with currents and microwave input frequencies that bring

the intrinsic phase shift ϕ close to zero or π, as this is the closest portion to an ideal parabola (see Eq. 2.31). Figure 2.7 (c) summarizes the results of a systematic investigation of ϕ as a function of I_{dc} and f_{ac} for $I_{\mathrm{ac,max}} = 70.7\,\mu\mathrm{A}$. The vertical line indicates the threshold current I_{th}, and the horizontal one represents the working point for the data in Fig. 2.7 (a) and (b). For this particular device geometry, $\phi = 0$ is achieved near the boundary of the locking range.

Figure 2.7 (d) provides an example of the time-domain evolution of the spatially averaged x-component of the magnetization $\langle m_X \rangle$, obtained for $I_{\mathrm{dc}} = -0.148\,\mathrm{mA}$ and $f_{\mathrm{ac}} = 800\,\mathrm{MHz}$ (see the circle in Fig. 2.7 (c)), along with the ac current and the indication of dm_X. A constant time shift can be observed when comparing the time traces. The magnetization dynamics are primarily driven by a first harmonic containing approximately 76% of the total energy, while higher-order harmonics account for the remaining 24%, as illustrated in Fig. 2.10. These higher-order harmonics may have a direct impact on the measurement of the intrinsic phase shift in time-domain traces. Therefore, the intrinsic phase shift is calculated in the Fourier space.

Figure 2.9 shows an example of the evolution of the magnetization when a dc step is applied to induce the injection locking regime. The transient time is approximately 10 ns, which provides a good estimate of the speed of the multiplication operation.

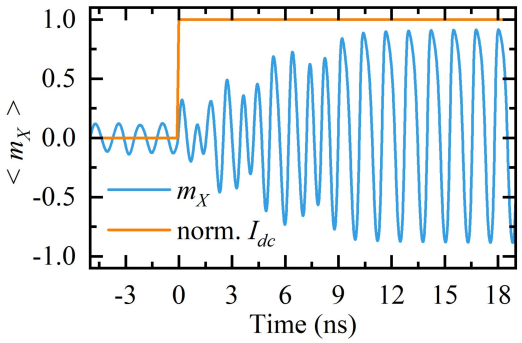

Figure 2.9: Time-domain trace illustrating the injection locking of the x-component of the magnetization (blue solid line) obtained with the application of a dc current step from 0 to $-0.148\,\mathrm{mA}$. The corresponding normalized dc current is depicted by the orange solid line. The applied ac current has an amplitude of $I_{\mathrm{ac,MAX}} = 70.7\,\mu\mathrm{A}$ and a frequency of $f_{\mathrm{ac}} = 800\,\mathrm{MHz}$. Figure adapted from [55].

2.4.3 DOR-based analog multiplication

From the device's input-output relationship, the parameters a, b, and c can be identified, satisfying the relation $V_{\mathrm{dc}}(I_{\mathrm{dc}}) = aI_{\mathrm{dc}}^2 + bI_{\mathrm{dc}} + c$, which links the bias current I_{dc} and the rectified dc voltage V_{dc}. The input current range is then scaled

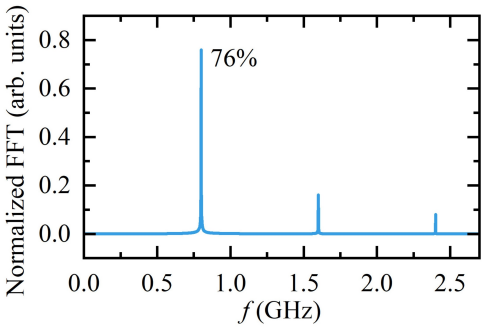

Figure 2.10: FFT of the x-component of the magnetization obtained for $I_{\mathrm{dc}} = -0.148\,\mathrm{mA}$, $I_{\mathrm{ac}} = 70.7\,\mu\mathrm{A}$, and $f_{\mathrm{ac}} = 800\,\mathrm{MHz}$ depicted by the circle in Fig. 2.7 (c), normalized by the sum of the three dominant peaks. The first harmonic contributes approximately 76% of the total. Adapted from [55].

to the desired input range x (for simplicity, we will consider the range $[-1, 0]$) using the following linear transformation:

$$I_{\mathrm{dc}} = |I_{\mathrm{dc},0} - I_{\mathrm{dc},-1}|x + I_{\mathrm{dc},0},$$

where $I_{\mathrm{dc},0}$ and $I_{\mathrm{dc},-1}$ are the current values corresponding to the numeric inputs 0 and -1. This results in an even parabolic equation where $V_{\mathrm{dc}}(x) = V_{\mathrm{dc}}(-x)$. The new parabolic relationship is given by $V_{\mathrm{dc}}(x) = a'x^2 + c'$, where $a' = a|I_{\mathrm{dc},0} - I_{\mathrm{dc},-1}|^2$ and $c' = V_{\mathrm{dc,MAX}}$ (see Fig. 2.11). The final calculation to compute FG depends on evaluating the voltages for $x = F$, G, and $(F - G)$ combined as shown in Eq. 2.3.

For example, consider Fig. 2.11, which shows experimental values of the rectified V_{dc} plotted against both I_{dc} and the input x. If we take $F = -0.62$, $G = -0.44$, and $F - G = -0.18$, the corresponding V_{dc} values, $V_{\mathrm{dc},F} = 17.85\,\mathrm{mV}$, $V_{\mathrm{dc},G} = 18.30\,\mathrm{mV}$, and $V_{\mathrm{dc},F-G} = 18.57\,\mathrm{mV}$, can be used in Eq. 2.3, considering the parameters $a' = -2.0476\,\mathrm{mV}$ and $c' = 18.565\,\mathrm{mV}$. In this manner, the product obtained is 0.241, which is very close to the desired result $FG = 0.273$.

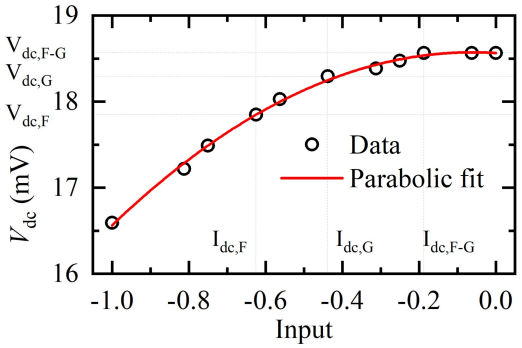

Figure 2.11: Experimental data (circles) presented in the article [53] and parabolic fit (solid line), showing the rectified voltage as a function of the numerical input encoded with the dc current for the parabolic equation. Figure adapted from [55].

We propose two scenarios for the implementation of the analog multiplication in hardware with the DOR:

1) Maximum speed: This is achieved by using three diodes for each multiplication and a CMOS circuit to perform the addition. The required time is the sum of the time needed to achieve locking and the time to perform the addition (division is handled simultaneously with an appropriate gain for the analog adder).

2) Minimal area occupancy: In this case, the three DOR operations are performed using the same diode. The time required is at least three times longer, and additional memory elements are necessary to store the data before the summation.

Considering the low area occupancy of the simulated devices, the first scenario is the most advantageous for current technological needs.

2.4.4 Application in computer vision

As an initial step, we evaluate the micromagnetic and experimental multiplications based on DOR comparing it with the ideal case. Figures 2.12 (a) and (b) present 200 multiplications obtained using DOR, derived using numerical data in Fig. 2.7 (a) (circles) and experimental results from Fig. 2.11, compared to the ideal multiplication output (solid line). The findings indicate that the correlation between the ideal case and the micromagnetic (experimental) DOR multiplication is 99.93% (99.83%).

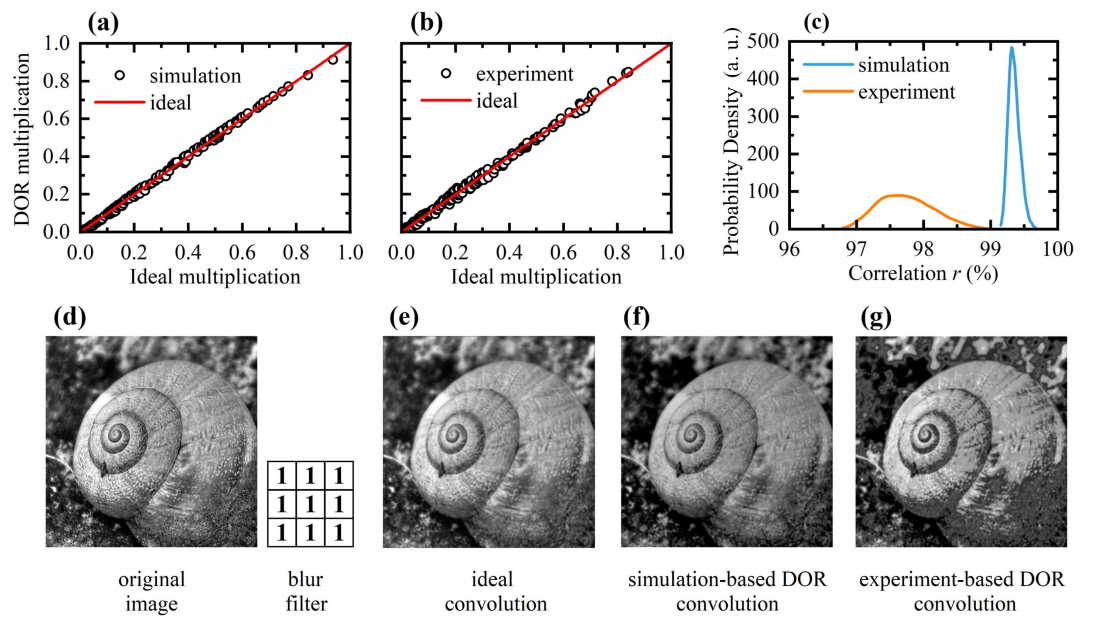

Figure 2.12: (a) Comparison between DOR-based multiplications derived from micromagnetic simulations (circles) and the ideal multiplication (solid line, bisector of the first quadrant). (b) Similar comparison as in (a), but using the experimental curve for DOR multiplication presented in [53]. (c) Probability density functions of the correlation for the convolution of 10^4 random filters, considering DOR-based multiplication via simulation (blue curve) and experimental data (orange line). (d) Image of a snail, extracted from the ImageNet dataset [33]; the inset shows the 3×3 blur filter used for the convolution. (e) Convolution result using ideal multiplication. (f) DOR-based convolution result obtained using micromagnetic data. (g) DOR-based convolution result obtained using experimental data. Figure adapted from [55].

The second evaluation involves the convolution of a snail image (extracted from the ImageNet dataset [33]) with 3×3 filters. Figure 2.12 (c) depicts the probability density functions (PDFs) of the correlation coefficients, r, computed from 10^4 random filter instances. The mean correlation coefficients are $\bar{r}_{\text{sim}} = 99.41\%$ and $\bar{r}_{\text{expt.}} = 97.87\%$ for the simulated and experimental data, respectively. The lower

average correlation and greater variability in the experimental data arise from less-precise parabolic behavior.

As an illustration, we display the convolution of the snail image in Fig. 2.12 (d), using a 3 × 3 blurring filter with uniform weights. Although this filter is not included in the random statistical analysis shown in Fig. 2.12 (c), it represents an edge case where multiplication errors are comparable, and these errors accumulate during convolution making it a worst-case scenario. Figures 2.12(e, f, g) show the convolution results for the ideal case (e) and DOR-based ((f) simulated and (g) experimental) multiplications. The correlation coefficients are $r_{\text{sim}} = 99.07\%$ and $r_{\text{expt.}} = 96.64\%$, which, as expected, fall outside the lower tails of the PDFs in Fig. 2.12 (c). Similar results are found with other images from the same dataset.

Currently, as the development and application of LLMs we are witnessing a race between big tech companies for the realization of the most performing model [64], new academic analysis are demonstrating that the precision in the MAC can be sacrificed for reducing the memory impact of AI models without significant impact on performance [65, 66]. In this context, we test the impact of DOR-based multiplication in a simple CNN for the recognition of handwritten images. Our analysis demonstrates that the impact of having a less accurate precision in the multiplication operation is minimal on the global accuracy of the network. Specifically, we consider a basic CNN with the architecture depicted in Fig. 2.13 (a), which is the one described in detail in the previous chapter.

The CNN is trained using Python and TensorFlow on the MNIST dataset [56], with a training set of 48000 images and a validation set of 12000 images. Testing is carried out on a test set of 10000 images. To avoid overfitting, dropout layers [67] and early stopping are applied. The recognition accuracy achieved is 98.64% on the training set and 98.57% on the test set.

Next, the trained weights are used to evaluate the accuracy on the same test set, taking into account DOR-based multiplication in the convolutional layers (ConvDOR). The recognition accuracy in this case is 96.83%, and when DOR-based multiplication is applied to both the convolutional and fully connected layers (ConvDOR+FCDOR), the accuracy drops to 94.72%, as summarized in Table 2.3 (row a). Since this test simulates the potential hardware effects of spin-torque diodes (STDs) in DOR-based multiplication, the accuracy reduction (less than 4%) can be mitigated with a few additional training iterations of the FC layer. In this case, we apply DOR-based multiplication to the convolutional layer and ideal multiplication to the FC layer (ConvDOR+trainFC), resulting in an improved accuracy of 98.40%, which is comparable to the original accuracy of 98.57%.

2.4.5 Robustness analysis

To study how this system is robust to device-to-device variations, we introduce a random variation of ±2.5% to the parameters of the parabola used in DOR-based

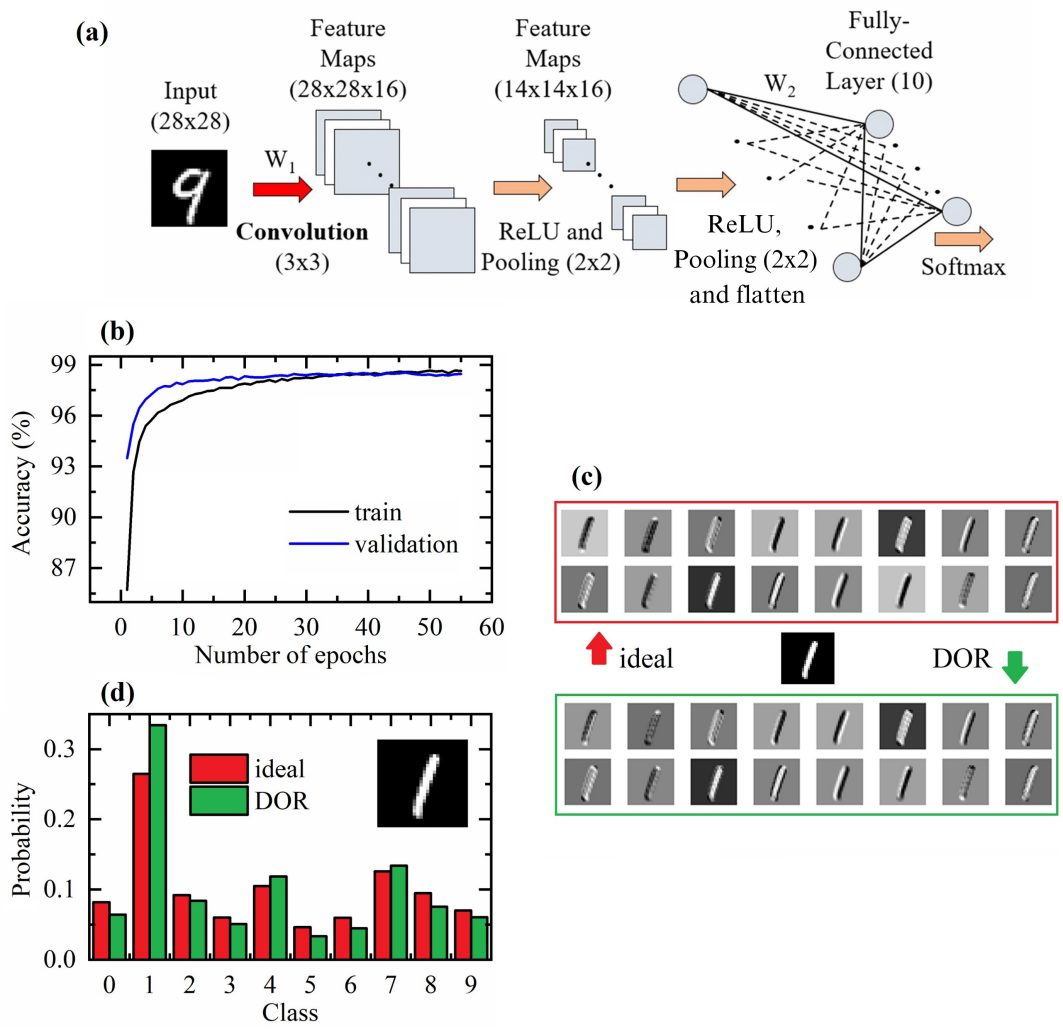

Figure 2.13: (a) Structure of the CNN. (b) Recognition accuracy percentage as a function of the number of epochs. The black (blue) line represents results for the training (validation) dataset. (c) Feature maps of a test image obtained using ideal multiplication (top) and DOR-based multiplication (bottom). (d) Probability distribution of all classes for the image of the handwritten digit one in the inset, obtained from the CNN using ideal multiplication (red) and the CNN using DOR-based multiplication applied to the convolutional layers with additional training of the FC layer (green). Figure adapted from [55].

multiplication. Fig. 2.14 shows how this variation has an impact on 200 computed multiplications. Table 2.3 (row b) shows that introducing this variation in the convolutional and FC layers, the accuracy of the network is dramatically reduced; nonetheless, adding a subsequent training phase recovers almost fully the ideal accuracy. This means that, with the training, the network is adapting to the added nonlinearities of the system.

As previously discussed, the cosine function can be approximated to a parabola for values close to multiples of π, and considering a large input portion of the considered experimental curve, as represented in Fig. 2.15 leads to less precise multiplications; row (c) in Table 2.3 shows that also in this case, the accuracy after the training of the FC layer is comparable with the benchmark.

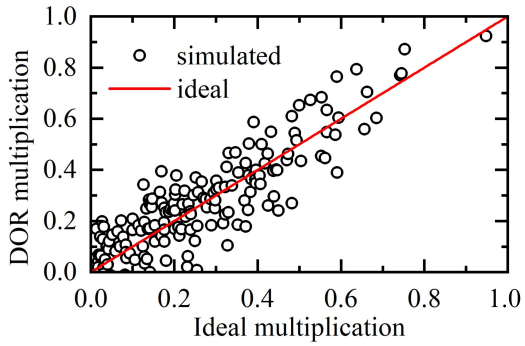

Figure 2.14: (a) 200 random multiplications (circles) performed using the DOR-based method, incorporating a 2.5% random variation in the parameters, compared with the ideal result (solid line). Figure adapted from [55].

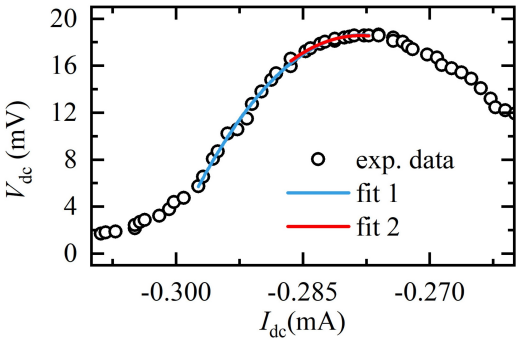

Figure 2.15: Experimental data showing the output dc voltage as a function of the input dc current in an injection-locked STD (circles), reported in Ref. [53]. The data are fitted with ideal parabolas, considering both a narrower range (red curve) and a wider range (blue curve). Near the peak, the red curve overlaps with the blue one. Figure adapted from [55].

	Test accuracy (%)			
	Ideal	**Conv**$_{DOR}$	**Conv**$_{DOR}$ + **FC**$_{DOR}$	**Conv**$_{DOR}$ + **trainFC**
a	98.57	96.83	94.72	98.40
b		85.51	51.18	98.33
c		97.07	93.11	98.35

Table 2.3: Accuracies obtained applying the ideal multiplication (**Ideal**), the DOR-based multiplication applied in the convolution layers (**Conv**$_{DOR}$), the DOR-based multiplication applied in the convolution layers and in the FC layer (**Conv**$_{DOR}$ + **FC**$_{DOR}$), the DOR-based multiplication applied in the convolutional layers training the FC layers after the substitution (**Conv**$_{DOR}$ + **trainFC**). (a) Results obtained using the curve shown in 2.11. (b) Results obtained simulating device to device variations. (c) Results obtained using a larger input-current range (the blue curve in Fig. 2.15).

Figure 2.16 presents a simulation of the studied device with $I_{\mathrm{ac,max}} = 70.7\,\mu$A and $f = 543\,$MHz at room temperature. In the presence of a thermal field, the frequency of self-oscillation is reduced, as expected due to the decrease in saturation magnetization. It's interesting to notice that the transient time is reduced to few nanoseconds.

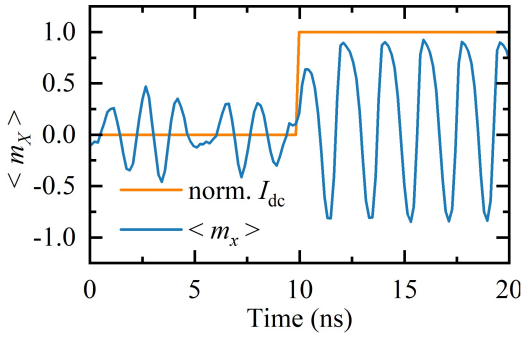

Figure 2.16: An example illustrating the application of a dc current step from 0 to -0.177 mA (with the normalized dc current shown in orange), alongside a plot of the magnetization transient along the x-axis (blue). This is observed when an alternating current with amplitude $I_{\mathrm{ac,MAX}} = 70.7\mu$A and frequency $f = 543$MHz is applied at room temperature. Figure adapted from [55].

2.5 Conclusion

In this chapter, we investigated the application of spintronic oscillators for analog multiplication, with a particular emphasis on their potential in reducing computational complexity and energy consumption in artificial intelligence tasks. We defined and implemented through micromagnetic simulations and analyses performed using experimental data, a method for the realization of the analog multiplication using the DOM and DOR characteristics of spin-torque oscillators.

We demonstrated that STOs can achieve analog multiplications under a variety of conditions, including device mismatches and thermal noise. Furthermore, the application of DOR-based multiplication in computer vision tasks, such as image recognition through convolutional neural networks, highlighted the practical potential of these devices, even if the individual multiplications are not precise.

In conclusion, this chapter described how spintronic oscillators can be used for designing an analog accelerator for artificial intelligence implementations, proposing a low-size and low-power solution.

Chapter 3

Oscillators applied to Combinatorial Optimization Problems

This chapter explores how oscillators can be employed to implement solvers for combinatorial problems.

We focus on problems characterized by non-polynomial (NP) complexity [68], where the time required to find the optimal solution grows more than linearly with the problem size, making them highly inefficient to solve using conventional algorithms.

One of the most well-known problems in this category is the travelling salesman problem (TSP), which involves finding the shortest possible route to visit n locations, given the distances between each pair of points. The time complexity for evaluating all possible routes and identifying the optimal one is $O(n!)$, making it computationally prohibitive for large n. However, in many cases, it is acceptable to find a good solution rather than the absolute best. For these situations, heuristic methods can be highly effective [69, 70, 71, 72, 73, 74, 75, 76].

In practical scenarios, for example, finding a route that is 10.1 km long when the optimal path is 10 km may be preferable if it significantly reduces the time needed to reach a solution. Heuristics can be useful in such cases because they rely on a degree of randomness in their approach. This stochastic feature allows for the exploration of multiple potential solutions without exhaustively evaluating every possibility, thus speeding up the process.

3.1 The Max-Cut Problem

The Max-Cut problem [74] is used as a reference for this work as it is a well-known NP-hard problem [77] that requires a limited amount of parameters. Given an undirected weighted graph (meaning that the connections don't have a direction and have an associated value) $G = (V, E)$, where V is the set of vertices (or nodes)

and E is the set of edges (or connections), the objective of the Max-Cut problem is to partition the vertices into two disjoint sets such that the accumulated amount of weights of edges "cut" between the sets is maximized. Figure 3.1 shows an exemplary graph composed of four nodes with weighted connections and its corresponding Max-Cut, represented by the black dashed line. The nodes are labelled with ones and zeros associated to the two disjoint sets. The maximum combination of cut edges is trivial in this example and the Max-Cut is 3.

Despite its NP-hardness, heuristic and approximation algorithms have been developed to tackle the Max-Cut problem. The approach that will be analyzed in this work utilizes Ising machines (IMs), which associates the problem to a Hemiltonian energy function such that its minimization will also be a solution of the starting problem.

IMs originated from the Ising model, initially developed as a method to describe ferromagnetism in statistical physics [78]. The Ising model, introduced by Wilhelm Lenz and later studied in detail by his student Ernst Ising, represents spins on a lattice that can interact with their neighbors. Over time, researchers realized that the energy minimization process inherent in the Ising model could be repurposed to solve combinatorial optimization problems (COPs).

One of the critical challenges in the performance analysis of combinatorial problems, such as the Max-Cut problem, lies in the large number of possible combinations to check for an exact solution as it is required to evaluate approximately 2^N combinations, as each node can have two states. For example, in a problem with 100 nodes ($N = 100$), there are approximately 1.26×10^{30} combinations. Due to the computational infeasibility of this approach, solver accuracy is typically assessed using well-established problem sets comparing the results obtained with other solvers in literature. In this work, we consider the G-set [79], and the accuracy is defined as the ratio between the obtained Max-Cut and the reference value, which corresponds to the results published in [80].

Another essential metric for evaluating combinatorial solvers is scalability, i.e., the ability to handle problems of varying sizes, with the goal of addressing the largest possible instances. In this work, we study scalability based on d-regular graphs, where each node is connected to exactly d other nodes. When $d = 3$, the problem instance is referred to as cubic.

In this work, we explore different models of oscillators applied for the implementations of IMs using the Max-Cut as a benchmark problem.

3.2 How Ising Machines solve combinatorial problems

The Ising model consists of the following components:

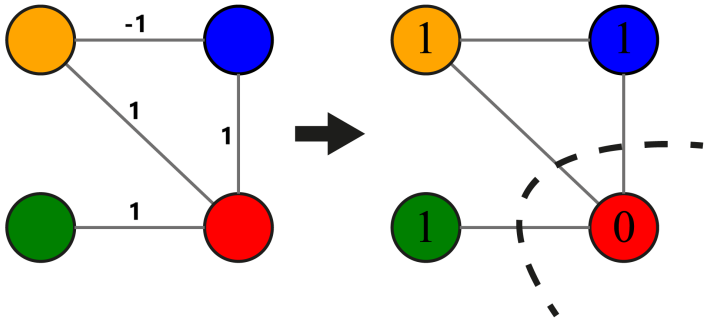

Figure 3.1: A schematic representation of a 4-node Max-Cut problem and its solution, corresponding to MC= 3. Figure adapted from [81].

- **Spins:** The system consists of a network of spins, where each spin can be in one of two states: $+1$ (up) or -1 (down). These spins represent the binary state of the nodes, commonly found in combinatorial problems.
- **Interactions:** Each pair of spins interacts via a coupling strength, J_{ij}. This interaction determines whether two spins prefer to align or anti-align.
- **External Fields:** Each spin can experience an external bias field, denoted as h_i, which influences the state of each spin individually.

The system's total energy, or Hamiltonian, is given by:

$$H = -\sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i \tag{3.1}$$

where $s_i \in \{-1, 1\}$ represents the state of the i-th spin, and J_{ij} and h_i are parameters that define the problem.

The Max-Cut has been chosen because by definition all the bias terms h_i are null, reducing the controlling parameters of the solver.

Many combinatorial optimization problems, such as the TSP, max-cut, or graph partitioning, can be formulated as minimizing an objective function. These objective functions are often quadratic, which can be mapped to an Ising Hamiltonian. The objective then becomes finding the spin configuration that minimizes the total energy, which corresponds to the optimal solution of the combinatorial problem.

The article [82] presents the mapping of many NP problems to Ising-compatible formulations.

The goal of an Ising machine is to find the ground state, which is the configuration of spins that minimizes the energy. Since the energy landscape contains many local minima that correspond to suboptimal solutions, finding the global minimum is akin to solving the combinatorial optimization problem.

By adjusting the coupling strengths J_{ij} and biases h_i according to the problem's formulation, the energetic landscape changes and the Ising machine explores different spin configurations. The machine evolves toward low-energy states and, if the mapping has been done correctly, these usually correspond to good solutions of the starting problem.

A the state-of-the-art, there is not a common solution for solving combinatorial optimization problems. In the last century many algorithmic implementations have been proposed, but usually these are not suitable for approaching problems with large sizes, achieve suboptimal accuracies, are very problem-specific, or a combination of the three [83, 84, 85].

Hardware solutions are very promising as, even if the solving times grow more than linearly with the size of the problem, having analog computation often means dividing the computing times by 3-4 orders of magnitude comparing with software solutions, shifting it for small problems from milliseconds to microseconds or lower,

depending on the used devices. This means that for larger problems, solutions in the order of seconds-minutes are still acceptable, while algorithmic approaches might take days or months. In this category we find quantum annealers, optical solvers, devices implemented with LC oscillators and spintronic ones.

The key idea behind the implementation of the hardware Ising machine is that the Ising Hamiltonian is associated to the energy functions that govern the devices' behavior. Due to the inherent properties of the system, which tends to favor a low energy state, the machine naturally evolves toward a local minimum effectively finding a good solution of the starting problem.

Quantum Annealers, as for the D-Wave products [86, 87], use quantum effects to explore the energetic landscape and find the global minimum by slowly "annealing" the system, meaning that the input amplitude is gradually increased (or decreased) and this helps the solver escape local minima. Quantum solvers use superconducting circuits and quantum mechanics for the computation. In this way, very fast solutions can be achieved (in the order of microseconds) at the cost of bulky systems that require entire lab rooms and tens of kW of power for the cryogenic cooling, necessary for keeping the qubits at temperatures of the order of 15 mK [88].

Optical solutions, where the coupling is implemented with delay lines and phase modulation, can be useful for approaching large problems (the largest hardware Ising machine is optical with 100k spins), but kilometers of optic fiber are required making the system bulky and not suitable for integration [69, 89, 90].

Electrical solutions use inductors and capacitors to implement oscillators, and as well are not suitable for realizing highly integrated implementations [91, 92], nevertheless these allow to realize a very educating environment to test how things work in a macro scale. Spintronic solutions are very promising as the devices can be manufactured with sizes in the order of tens to hundreds of nanometers [93] and are compatible with the silicon technology. However, as in most hardware solutions, the connectivity is a critical challenge, and as for current technological solutions, the connections are only local between physically adjacent nodes and not reprogrammable. Nonetheless many progress have been done in recent years [94, 95, 96, 97] and the field is growing rapidly.

The main alternative to hardware implementations is software simulation, where the machine is simulated using classical computational resources. This approach offers flexibility, allowing researchers to study in detail the behavior of oscillators and, as demonstrated in this work, to realize networks with tens of millions of nodes [98, 99, 100, 101, 102, 103, 104].

The focus of this chapter will be the analysis of simulated oscillator Ising machines (OIMs) [105] for two main reasons: to model possible spintronic-compatible solutions and to test the performance in terms of solving times, size and accuracy of the digital implementation itself.

3.3 Modeling OIMs

To implement an IM using oscillators, certain requirements of the Ising model must be fulfilled to realize a system that minimizes an Hamiltonian energy function. Specifically, we need a system that represents a network of nodes interacting with each other and characterized by binary outputs. This binarization is essential for problems like the Max-Cut, where each node must be labeled with a binary value corresponding to one of the two disjoint groups into which the system is divided.

Additionally, we need to incorporate a biasing term, which, while not necessary for solving the Max-Cut problem, is crucial for other applications.

The adjacency matrix J_{ij} can be realized by exploiting the interaction properties of the oscillators, as described, for example, by Eqs. 2.15-2.18, which illustrate how two oscillators influence each other.

To achieve the binarization of the oscillator phases, a suitable method is the application of a signal at twice the oscillation frequency. This induces a phenomenon known as sub-harmonic injection locking (SHIL) [106, 107], where the system locks onto the external ac signal.

Figure 3.2 illustrates an example of an oscillatory signal locked to an injected signal with the same frequency (a) and with twice the frequency (b). In the case of injection locking, the oscillator locks to the external signal both in frequency and phase. In the case of SHIL, while the oscillator does not lock in frequency, only two distinct phases are possible, separated by π and this can be utilized to binarize the system. Hence, in a system of two interacting oscillators, depending on the sign of the interaction, the devices will oscillate either in phase or out of phase.

In an IM, the biasing term can be understood as a mechanism that influences a spin to settle into the $+1$ state when a positive bias is applied, and into the -1 state when a negative bias is applied. In our system, all oscillators operate at the same frequency, but the injection locking mechanism can still be useful as it affects the phase of each individual node and its application makes one of the two possible phases more favorable, effectively guiding the oscillator to adopt the phase that corresponds to either the $+1$ or -1 state, effectively implementing the bias.

In summary, starting from the example illustrated in Fig. 3.1, we introduce a second layer representing the natural evolution of the phases in the network of oscillators. This is shown in Fig. 3.3, where the randomly initialized phases evolve and converge towards the binary states -1 ($\phi = 0$) and 1 ($\phi = \pi$).

3.3.1 Kuramoto Model

The Kuramoto model of oscillators is developed for the analysis of group behavior of biological systems and later found to be well representative of physical systems [108, 109]. It is also suitable for the implementation of an OIM, where each phase of the

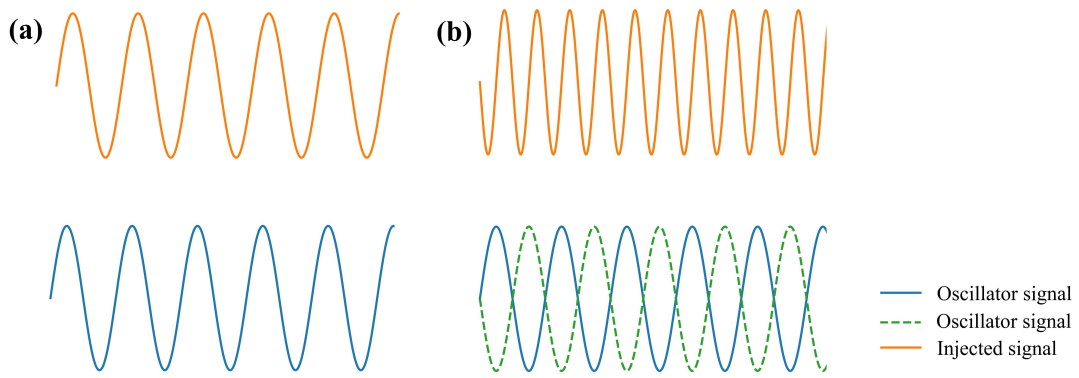

Figure 3.2: Illustration of injection locking (a) and SHIL (b), where the oscillator signal is depicted by the blue and green dashed lines, and the injected signal is represented by the solid orange line.

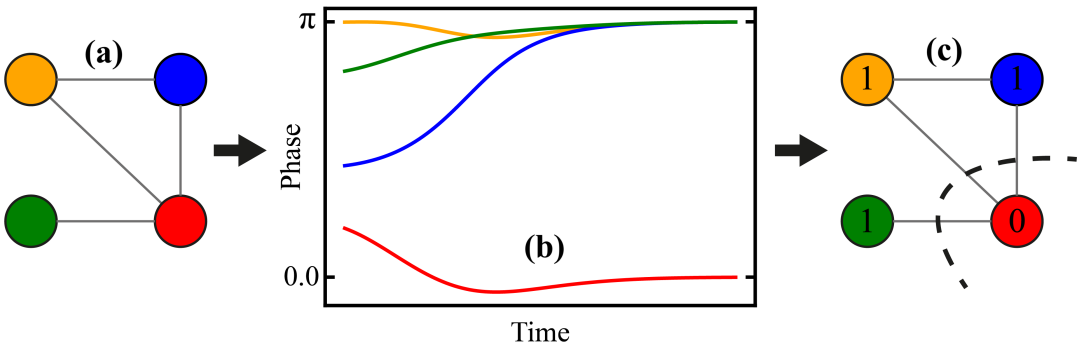

Figure 3.3: (a) Illustration of an exemplary graph. (b) Time evolution of the OIM where each phase converges to the output state, represented in (c). Figure adapted from [81].

network is defined as

$$\frac{d\phi_i}{dt} = -K\sum_{j=1}^{N} J_{i,j}\sin(\phi_i - \phi_j) - S\sin(2\phi_i) + A\langle\xi\rangle \tag{3.2}$$

and the time dependence of ϕ_i, ϕ_j, and ξ has been omitted for simplicity.

The first term describes the interaction between the phase ϕ_i and the other phases in the system, determined by the non-zero elements in the i^{th} row of the adjacency matrix J (or equivalently, the i^{th} column, since the matrix is symmetric). When $J_{i,j}$ is positive, the j^{th} phase attracts the i^{th} one, and the system tends to minimize the difference $(\phi_i - \phi_j)$; if $J_{i,j}$ is negative, the j^{th} phase acts as a repeller, maximizing the difference $(\phi_i - \phi_j)$ in the phase space.

The second term represents the SHIL signal that is necessary for the binarization of the energetic landscape introducing minima for phases equal to 0 and π.

The final term, $\langle\xi\rangle$, represents Gaussian white noise with zero mean and unit variance. This noise helps the system explore the energy landscape by providing the necessary energy to escape local minima, a technique commonly employed in similar solvers [73, 102, 110, 111, 112].

The tree terms are modulated by the amplitude factors K, S and A.

Since the systems studied in this work consist of identical oscillators, we employ a frequency-normalized version of the Kuramoto model, as the frequency term only adds a constant shift to all phases without altering the system's dynamics.

3.3.2 Slavin Model

Analogously, we adapted the Slavin model of oscillators to test if this model, which well represents realistic devices, can provide insights about future spintronic implementations.

Adapting the model to the Ising case, the evolution of each oscillator can be described by a set of two coupled differential equations [63]:

$$\begin{aligned}\frac{dp_i}{dt} = &-2p_i\left[\Gamma_{+,i}(p_i) - \Gamma_{-,i}(p_i)\right] + 2F_e\sqrt{p_i}\cos(2\omega_i t + 2\phi_i)\\ &+ 2\Omega\sum_{j=1}^{N} J_{ij}\sqrt{p_i p_j}\cos(\phi_i - \phi_j - \beta) + \xi_p(t),\end{aligned} \tag{3.3}$$

$$\begin{aligned}\frac{d\phi_i}{dt} = &-\omega_i(p_i) - \frac{F_e}{\sqrt{p_i}}\sin(2\omega_i t + 2\phi_i)\\ &+ \Omega\sum_{j=1}^{N} J_{ij}\sqrt{\frac{p_j}{p_i}}\sin(\phi_i - \phi_j + \beta) + \xi_\phi(t),\end{aligned} \tag{3.4}$$

where ϕ_i and p_i represent the time evolution of the oscillator's phase and power, respectively.

In both equations, the first term on the right-hand side describes the behavior of an isolated oscillator. The functions Γ_+ and Γ_- account for the positive and negative damping effects. By expanding these functions to the first order, we obtain $\Gamma_+(p_i) = \Gamma_0(1 + Qp_i)$ and $\Gamma_-(p_i) = \Gamma_0 I_{\text{ratio}}(1 - p_i)$, where Q is the nonlinear damping coefficient, and I_{ratio} is the ratio of applied current to the threshold current required for self-oscillation. These equations have been validated by experimental data [63, 113, 114].

As previously mentioned, the oscillator's frequency, ω_i, depends on its power, p_i, through the relationship $\omega_i = \omega_0 + N_0 p_i$, where ω_0 is the resonance frequency and N_0 is the nonlinear frequency shift. Also in this case we are studying a system with identical oscillators, and due to the relation between frequency and oscillatory power, the final frequency can vary slightly from device to device for different power values.

Noise is introduced following the approach described in [63], and the results, with or without thermal noise at room temperature, are qualitatively similar.

The term with amplitude F_e represents the external signal used for SHIL. The third terms model the interaction between oscillators, with the coupling strength Ω and the network topology determining their interaction. The parameter β is the phase delay between the coupled signals, which depends primarily on the coupling mechanism and the spatial separation of the oscillators and its effect has been studied in the previous chapter.

The parameters used for the simulations are based on experimental data from MTJ-based spintronic oscillators [115] with CoFeB as the free layer (see Tab. 3.1 for the complete parameter set).

The Q and N parameters have been chosen from experimental measurements[116], and we conducted a systematic analysis for understanding how they influence the overall accuracy to improve future hardware design and choose the right technology of oscillators.

Fig. 3.4 shows the average Max-Cut obtained simulating 100 iterations per cell. The results show that, in general, the parameter Q has a lesser impact on the Max-Cut score compared to the parameter N, whose optimal value is approximately $N \approx 10N_0$, considering N_0 as a reference value [97, 116]. These findings suggest that the nonlinear behavior of spintronic oscillators could be advantageous for implementing an IM hardware system.

3.3.3 Comparison between Kuramoto and Slavin Models

Figure 3.5 (a) and (b) show the time variation of the oscillator phases (and the powers for Slavin model) obtained inputting the same problem to the two solvers. In both cases an annealing of the parameters controlling the injection locking and

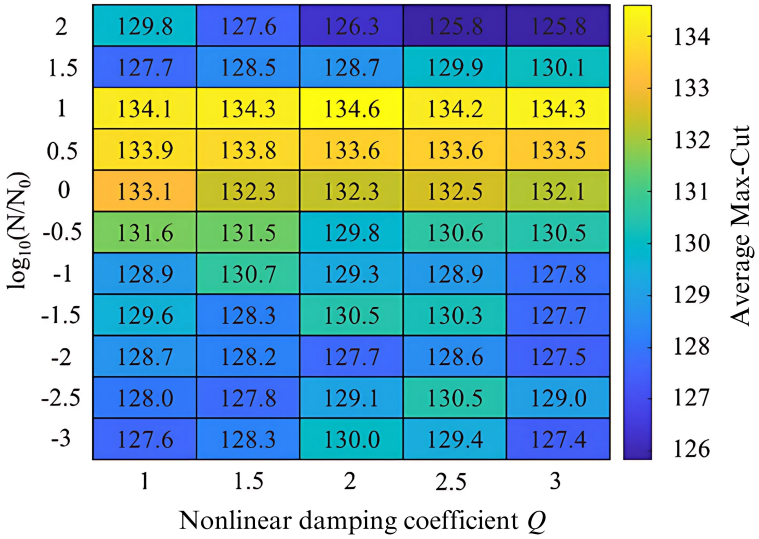

Figure 3.4: A grid search was conducted to find the optimal values for the nonlinear frequency shift (N) and the nonlinear damping coefficient (Q). The Max-Cut performance was evaluated by averaging the results from inputting 100 randomly generated cubic graphs, each with 100 oscillators, into the model for various combinations of N and Q values. Figure adapted from[117].

Parameter	Value
ω_0 (GHz)	4.2
$\frac{N}{2\pi}$ (GHz)	-3.44
Q	2
I_{ratio}	2
Γ_G (MHz)	252
β (rad)	-0.64π
F_e	3×10^9
Ω	1×10^9
dt (ps)	5

Table 3.1: Slavin Model Parameters.

SHIL is applied in an increasing manner until a treshold is reached and the values are resetted to zero. The annealing cycle is shown in green in Fig. 3.5 (c) and the amplitude of the noise it kept constant. Figure 3.5 (a) presents the phases of the oscillators modelled with Kuramoto's theory. We can distinguish two main behaviors in this plot as the phases either align or diverge, and this depends on the amplitude of the coupling coefficient, when it is too high, the phases start mixing and this helps getting out of local minima looking for a better configuration. Analogously, Fig. 3.5 (b) shows the phases and the powers simulated using Slavin model. In this case the mixing happens when the interactions are low, and increasing the amplitude of the control parameters makes the system stabilize to the found solution. Also the powers are represented in this plot, and it is clear how there is a binarization also regarding the powers.

Figure 3.5 (c) shows the cut evaluated through time for the two models in red and blue, which in this case reach the same maximum value of 136.

From this analysis we can conclude that both Kuramoto and Slavin models are capable of achieving good results of Max-Cut instances meaning that this technology is promising for the approach of similar problems and that spintronic oscillators might be useful for the development of an hardware solver. From a computational point of view, the Slavin model results more expensive than the Kuarmoto one in terms of time and memory complexity, as two coupled ordinary differential equations (ODEs) must be evaluated instead of one.

In summary, this analysis demonstrates that both the Slavin and Kuramoto models of oscillators are effective for implementing an IM to tackle Max-Cut problems, and show a comparable behavior. However, the Slavin model proves to be

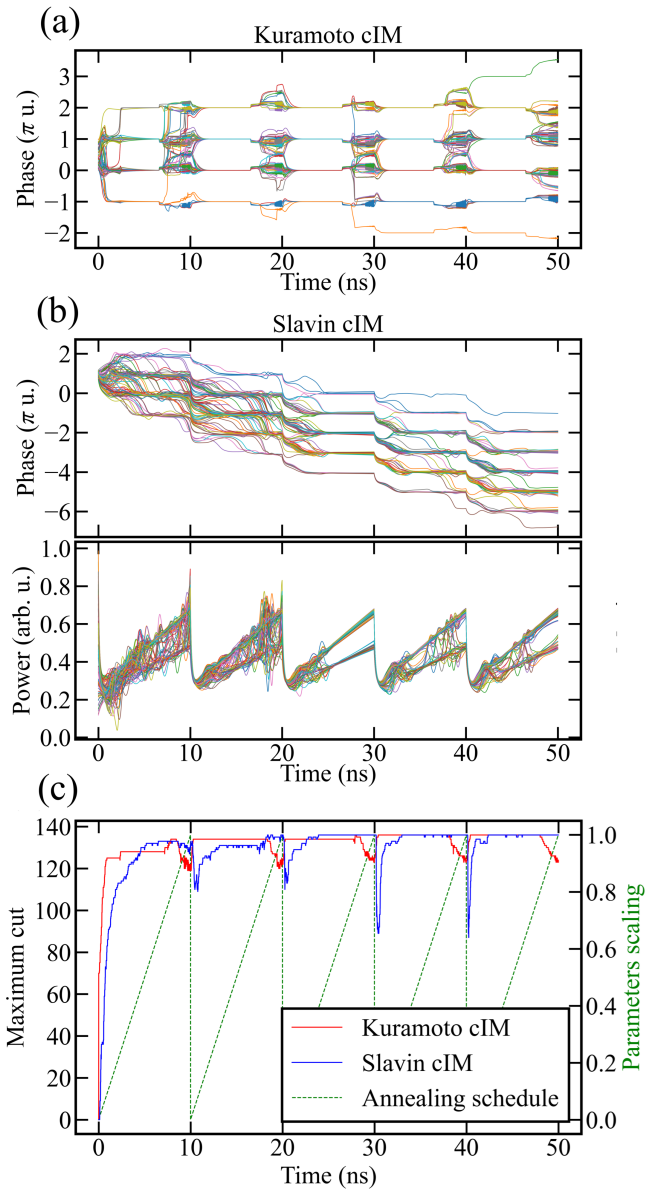

Figure 3.5: Sample executions of a maximum cut search on the same randomly generated cubic graph with 100 oscillators, simulated using the Kuramoto model (a) and the Slavin model (b). The graphs illustrate the oscillator phases, which are used to determine the cut value, as well as the power of the oscillators in the Slavin model. The cut values for both models, calculated at each time step (solid lines), are displayed in (c) along with the linear annealing schedule for the amplitude parameters (dashed line). Both models employ a sawtooth-shaped annealing schedule. Figure adapted from [117].

more computationally expensive. Consequently, in the following analyses, we will focus solely on the Kuramoto model. Nevertheless, it is reasonable to expect that the results presented here could be extended to the Slavin model and, more broadly, to potential hardware implementations.

3.4 Optimizing and Scaling Up OIMs

In this section we will present some modifications to the Kuramoto model to improve its efficiency and scalability for a GPU implementation, approaching d-regular problems with up to 20 million nodes, one order of magnitude higher than the largest we could find in literature.

We will observe more in detail the effect of noise annealing and how this can significantly improve the accuracies. Furthermore, we introduce an annealing approach that divides the analysis into checkpoints, enabling the system to resume from a prior state after each iteration. This method enhances the exploration of the solution space in scenarios with limited time. The system consistently attains accuracies averaging over 99.5% (up to 99.9%) on G-set problems, with computation times ranging from under 5 minutes to 1 hour. This makes it highly applicable to large-scale and time-sensitive tasks.

3.4.1 Problem Generation

When addressing extremely large problems ($N > 1M$), the generation of problem instances must also be optimized. In this section, we present a simple and efficient code developed for generating d-regular graphs with unitary weights, specifically for sparse cases ($d < 50$).

In a d-regular graph, each node is connected to exactly d other nodes. Consequently, the code must adhere to three key constraints:

- Each node must be connected to precisely d other nodes.
- Self-connections are not allowed.
- Duplicate connections between the same nodes are not allowed.

The core idea involves initializing a vector composed of integer values from 1 to N, denoted as $\mathbf{v} = [1, 2, \ldots, N]$. This vector is randomly shuffled, and adjacent values in the sequence are used to establish new connections. As an example, after shuffling v[1] is connected with v[2], then v[3] is connected with v[4], etc. In this way we are sure that all the nodes will be connected in couples.

After this step, it's necessary to check that the new connections are all unique and not duplicate of those from previous steps. If this is not the case, the previous

step is repeated. Otherwise, the degree is increased by one. The generation is stopped when the degree reaches the input value d. The pseudocode of the graph generation is presented below, where the weights are considered to be all unitary.

We chose the Fisher-Yates algorithm for shuffling the main vector due to its linear computational complexity with respect to the number of nodes, similar to other operations in the algorithm.

Unlike the `edge_list` output, the `check_vector` also includes duplicate connections. This is essential for the `check_regularity` function, which verifies that no duplicate values exist in each row for columns up to the current degree. If duplicates are found, the function returns 0, causing the `while` loop to restart.

The primary limitation of this algorithm is its time complexity dependency on the degree of the problem, which is more than linear. This impacts the algorithm's performance, already impacting for degrees higher than 10, making this solution suitable only for sparse regular instances.

3.4.2 Noise Annealing

To effectively approach COPs, solvers must possess two key abilities:

- They must be able to explore the energy landscape broadly to identify promising regions.
- They need to focus on local energy minima to find accurate solutions.

To perform a wide exploration of the energy landscape, solvers rely on momentum, which is provided by injected noise. This allows them to overcome large energy barriers. The key advantage of using stochastic methods over deterministic algorithms lies in this ability of locating the optimal region of the landscape where a good local minimum resides, which is computationally expensive for deterministic approaches.

Once a promising region of the energy landscape is identified, the solver must search for the local energy minimum to provide a precise solution. In this phase, algorithmic methods have proven effective, as the nearest local minimum can be deterministically evaluated [80, 118]. The benefit of using an Ising Machine is the smooth transition between these two processes, which can be modulated by annealing the noise amplitude.

In an OIM, when noise is absent, the phases either attract or repel each other, and the SHIL enforces two stable phase states: 0 or π.

When noise of sufficient amplitude is applied, the system is continuously pushed out of its local energy minimum, allowing it to explore different configurations. Unlike changing the initial conditions, which occurs only once, applying noise causes continuous exploration of the landscape.

Codice 3.1: Pseudocode for d-regular graph generation

```
# Initialization
SET d as degree of the problem
SET N as number of oscillators

SET shuffling_vector to increasing integer values [1, 2, ..., N]
INITIALIZE edge_list with zeros and size (d * N / 2, 3)
INITIALIZE check_vector with zeros and size (N, d)

# Start of the cycle
FOR each deg from 1 to d:
    SET flag = 0
    WHILE flag == 0:
        CALL Fisher-Yates_shuffling(shuffling_vector)
        FOR each index j from 1 to N / 2:
            SET connection1[deg, j] = shuffling_vector[j]
            SET connection2[deg, j] = shuffling_vector[N / 2 + j]
            CALL update_check_vector(check_vector, connection1[deg, j],
            ↪  connection2[deg, j], deg, j)

        # The function update_check_vector updates the current check_vector with
        ↪  the new connections before checking if all constraints are satisfied
        END FOR

        # While loop iterates until check_regularity confirms the graph is
        ↪  regular up to the current degree
        SET flag = check_regularity(check_vector, deg)

    END WHILE

    # Update the edge_list with the generated and verified edge vector
    SET edge_list[deg * N / 2 : (deg + 1) * N / 2, 1] = connection1
    SET edge_list[deg * N / 2 : (deg + 1) * N / 2, 2] = connection2
    SET edge_list[deg * N / 2 : (deg + 1) * N / 2, 3] = ones(N / 2)

END FOR
```

The most effective way to find good energy minima involves starting with a high level of noise and gradually reducing it over time. This process is commonly known as annealing [119, 120].

Figure 3.6 (a) provides a 2-dimensional representation of the energy landscape. The system is capable of exploring states between the noise level and the black landscape. When reducing linearly the noise, the system gets stuck in region of the landscape progressively smaller until being trapped in a local minimum. Figure 3.6 (b) shows the cuts evaluated during an exemplary iteration, where the noise is

applied in a linearly decreasing fashion until the system achieves an optimal and stable solution.

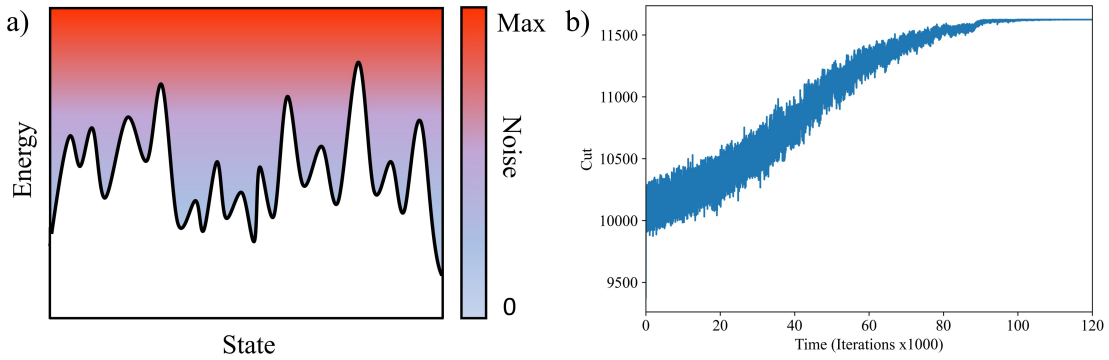

Figure 3.6: (a) Sketch of a simplified energy landscape. When substantial noise is introduced, the system moves freely across the landscape, exploring multiple states. As the noise gradually decreases, the system tends to stabilize in one of the lower-energy states, which are separated by barriers that cannot be crossed without noise. (b) Exemplary analysis of the cut throughout an entire iteration, during which a linearly decreasing noise is applied until achieving an optimal and stable solution.

3.4.3 Algorithmic Implementation

This section describes the implementation of a GPU-accelerated OIM that uses Heun's method for the integration of ODEs, developed in native C++/CUDA. The solver is specifically optimized for large, sparse, d-regular problems. As we now shift our focus to the Kuramoto model of oscillators from a computational rather than physical perspective, a hyperbolic tangent term is introduced within the sinusoidal component of the coupling term of the phase dynamics, resulting in:

$$\frac{d\phi_i}{dt} = -K \sum_{j=1}^{N} J_{i,j} \tanh(Q \sin(\phi_i - \phi_j)) - S \sin(2\phi_i) + A\langle\xi\rangle, \tag{3.5}$$

where this tanh term amplifies the sinusoidal function's influence. Figure 3.7 shows a plot of $\sin(x)$, $\tanh(2\sin(x))$, and $\tanh(10\sin(x))$. Considering x as the phase difference between two oscillators, we observe that the coupling contribution becomes zero only when the two phases are either in phase ($x = 0$) or in phase opposition ($x = \pi$). In all other cases, the coupling contribution is non-zero, and the addition of the tanh function ensures a significant interaction even with minimal phase difference, thus accelerating convergence.

The parameter Q is set to 10, as described in [105].

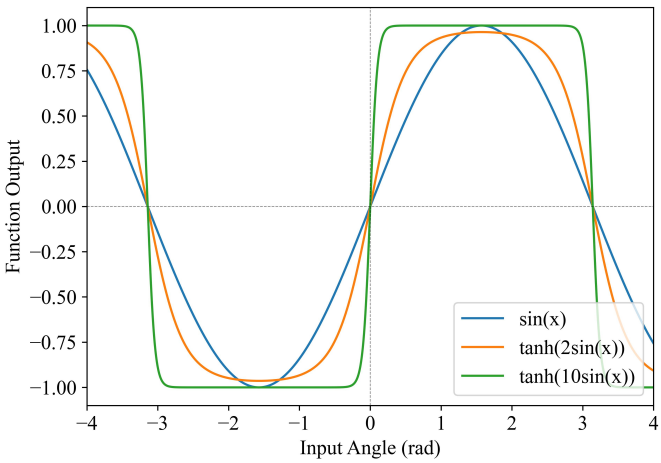

Figure 3.7: Plot of $\sin(x)$, $\tanh(2\sin(x))$ and $\tanh(10\sin(x))$.

The main scalability limitation arises from the adjacency matrix in Eq. 3.2, which governs the coupling between oscillators and has a size of N^2 [72]. For d-regular problems, each row (or column, since the matrix is symmetric) contains only d non-zero values. To exploit this sparsity, we used a weighted edge list representation for the graphs. This data structure is a matrix with three columns: two store the indices of the connected nodes, and the third contains the edge weights. The number of rows corresponds to the total number of connections, which for d-regular problems is $d \cdot N/2$.

The goal of this approach is to convert the adjacency matrix into a structure of size $3 \cdot d \cdot N/2$, maintaining all the necessary information without explicitly storing the null connections. This is particularly advantageous for sparse problems, where the degree is limited.

The second key step involves computing the first term in Eq. 3.2, and the coupling information is processed and stored in a support vector.

By the end of the loop, the support vector holds all the coupling data, allowing the phase variation for each oscillator to be expressed as:

$$\frac{d\phi_i}{dt} = \text{SupportVector}_i - S\sin(2\phi_i) + \xi. \tag{3.6}$$

In other words, we proposed the implementation of an edge list and a support vector with a total size of $2 \cdot d \cdot N$ substituting the adjacency matrix. With this modification, the phase variations can be calculated using two sequential (non-nested) loops: one with $d \cdot N/2$ iterations and another with N iterations, this results being an improvement in the time and memory complexity for sparse problems comparing with the original formulation of Eq. 3.2 that requires an adjacency matrix of size N^2, and consequently N^2 steps are necessary in each interaction. In essence, this optimization reduces the time and memory complexities from $\mathrm{O}(N^2)$ to $\mathrm{O}(d \cdot N)$, making this method excellent for sparse (low d) instances.

A figure of merit can be defined by dividing the memory requirements of both methods, yielding $4 \cdot d/N$. When this figure of merit is less than 1, the vector-based representation is preferable for both memory efficiency and computation time. In practice, since the dot product (present in the starting formulation) is efficiently implemented with GPUs, the proposed method is preferable when $4 \cdot d/N \ll 1$.

Figure 3.8 illustrates a 2-regular graph with 5 nodes and unitary weights (a), along with the corresponding representations of the adjacency matrix J, the equivalent edge list, and the support vector that is updated at each iteration (b).

The pseudocode of the support vector implementation is reported below.

3.4.4 Scalability

Figure 3.9 (a) shows the Max-Cut values obtained for d-regular problems with degrees ranging from 3 to 25 and sizes up to 20 million nodes, overcoming the

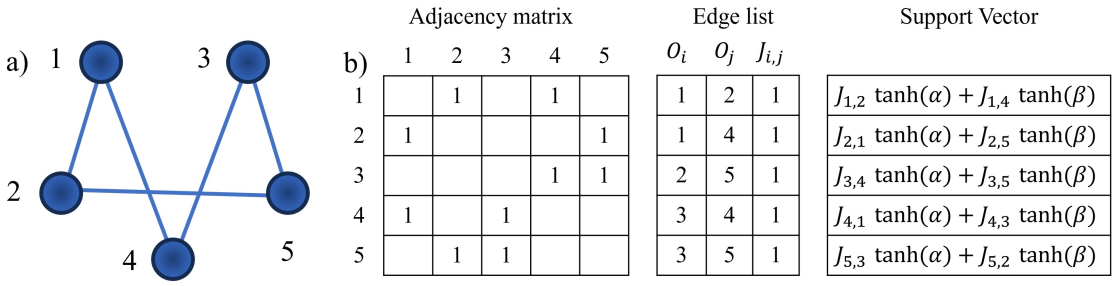

Figure 3.8: (a) Diagram of a 2-regular graph consisting of 5 nodes with uniform connections. (b) Representation, from left to right, of the adjacency matrix (zeros shown as blank spaces), the corresponding edge list, and the support vector. Here, O_i represents the index of the i^{th} oscillator. α and β are placeholders for the terms $Q\sin(\varphi_i - \varphi_j)$ and $Q\sin(\varphi_j - \varphi_i)$, respectively, where Q is a constant and φ_i, φ_j are the phases of oscillators i and j.

Codice 3.2: Pseudocode for the update of the Support Vector

```
// Initialization
INITIALIZE SupportVector with zeros and size N * d / 2

// Start of the cycle
FOR each k from 1 to N * d / 2:
    SET Osc1 = EdgeList[k, 1]
    SET Osc2 = EdgeList[k, 2]
    SET Coupling = EdgeList[k, 3]

    SET SupportVector[Osc1] += K * Coupling * tanh(Q * sin(phase[Osc1] -
    ↪  phase[Osc2]))
    SET SupportVector[Osc2] += K * Coupling * tanh(Q * sin(phase[Osc2] -
    ↪  phase[Osc1]))
END FOR
```

largest problem size approached by an IM found in the literature by an order of magnitude. For problems with sizes up to 1 million nodes, the results were averaged over 20 instances, while a single instance was considered for larger problems. The article [121] shows a linear relation between the theoretical upper bound of the Max-Cut and the size of the problem for a fixed degree; the same is shown in Fig. 3.9 (a) demonstrating the solver's ability to find high-quality solutions across small and large problems without significant accuracy loss. It is important to note that Fig. 3.9 (a) provides a broad view of the solver's accuracy, though subtle variations may not be easily noticeable due to the logarithmic scale. The detailed accuracy of the solver is discussed in the next chapter and evaluated using a benchmark set of smaller problems.

The upper part of Fig. 3.9 (b) shows the solving times for the instances in Fig. 3.9 (a). For smaller problems, the execution time is dominated by initialization processes, such as memory allocation. For problems with more than 1 million nodes, the solving time scales linearly with the graph size. This linearity is more clearly observed in the lower part of the figure, which shows solving times normalized by the number of nodes.

In summary, the runtime and memory requirements scale linearly with the number of simulated nodes, the degree of the problem (or equivalently, with the total number of connections for non-regular instances), and the number of time steps.

The analyses were conducted with one thousand time steps using an Nvidia T1000 GPU with 8 GB of RAM.

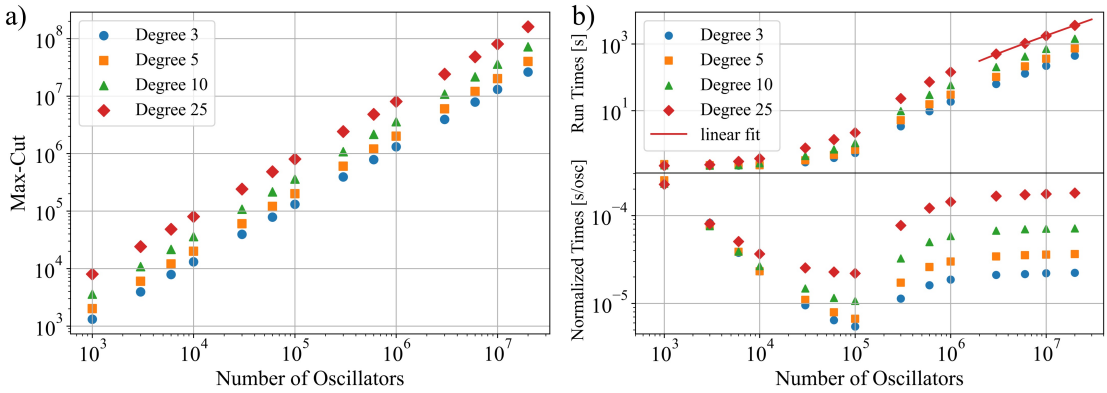

Figure 3.9: (a) Max-Cut values obtained for problems of various sizes and densities. (b) The upper part of the figure shows the runtimes of the problems mentioned in (a). The red curve, given by the equation $t = 1.8 \times 10^{-4} \times N - 58.1$, illustrates the interpolation of the solution times for the last 4 data points with a degree of 25, indicating a linear relationship between the number of oscillators and the runtime. The lower section of the figure presents the solution times normalized by the number of oscillators. For linear solution times, this graph should remain constant, which is observed for problems with sizes exceeding 1 million.

3.4.5 Accuracy

Evaluating the accuracy of d-regular problems with varying sizes is challenging because the exact solution for random instances may differ in each case. While upper bounds have been studied, such analyses are primarily applicable to dense problems [121, 122].

To facilitate comparison with other state-of-the-art methods, we benchmarked the OIM using the G-set [79], a well-known collection of problems with sizes ranging from 800 to 20000 nodes and diverse connection structures.

Figure 3.10 (a) shows a comparative analysis of the accuracy for 100 instances, with the number of time steps per iteration varying from 100 to 10^5. The mean values are indicated by the dotted lines. The plot illustrates that, as the number of time steps increases, the average accuracy improves, and the variance decreases, signifying that the probability of finding a good solution in a single run increases with the simulation time, as expected. However, when looking for the best solution, it is necessary to repeat the analysis multiple times.

To determine the optimal balance between the duration of each iteration and the number of repetitions, we fixed the total number of time steps and varied the iteration length, ensuring that the product of the number of iterations and the size of each iteration remained constant at 10^7 steps.

Figure 3.10 (b) shows that for problem G25, the best configuration is achieved by performing an analysis with 10^5 time steps, repeated 100 times.

These analyses were conducted with linearly decreasing noise, and fixed values for the K and S parameters.

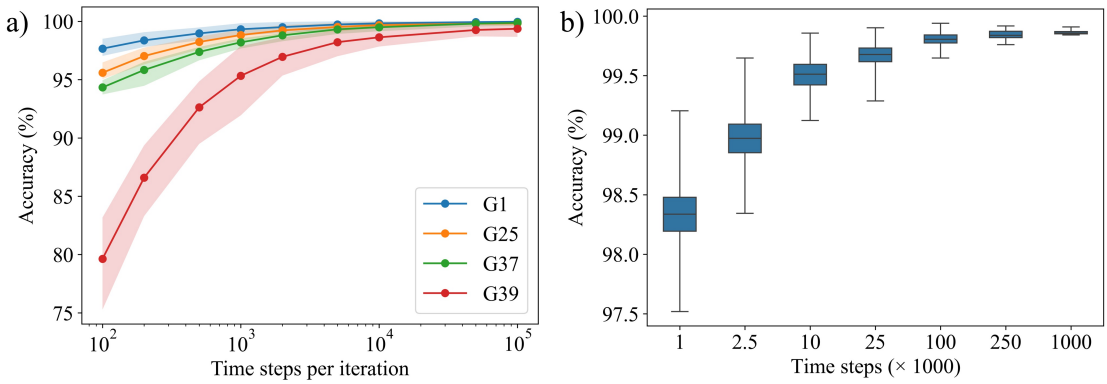

Figure 3.10: (a) Plot of the accuracies achieved by varying the iteration durations for different problems in the G-set. Each data point represents the average of 100 iterations, with the accuracy values indicated by the colored shading. (b) A boxplot generated by varying the number of steps used in each iteration, with the iterations repeated until a total of 10^7 time steps is reached.

3.4.6 Segmented analysis

In this section, we present an alternative approach to enhance system accuracy within a constrained timeframe by segmenting OIM operations across a limited number of total time steps.

During the annealing process, the OIM begins from a random configuration and, as noise is gradually reduced, it becomes trapped in various energy regions until a local minimum is reached. To explore multiple local minima, this process must be repeated several times.

We introduced checkpoints in the analysis where the phase state is saved and used as the starting point for future runs of the OIM. In this way, after completing a full iteration, instead of restarting from a random initial state, the search resumes from a checkpoint, with the previously saved state and the corresponding noise value. This enables the exploration of several local minima in a shorter time. This concept can be interpreted as a bifurcation analysis of the energy landscape. An example is shown in Fig. 3.11 (a), where the system converges to different local minima, improving the exploration of a specific region of the energy landscape. The inset of Fig. 3.11 (a) provides a zoomed view of the last segment repeated five times, with the dashed lines indicating the reference Max-Cut value (green) [80] and the one obtained in the study that introduced OIMs (red) [105] for this problem. The vertical lines represent the checkpoints.

After repeating the analysis from a saved state a chosen number of times, the system returns to a previous checkpoint and the process is repeated. In every case, the noise values are also restored to those saved at the checkpoint.

Using this strategy, numerous configurations can be examined with different segment durations and repetition schedules. Figure 3.11 (b) shows a boxplot of four different trials. T1 represents the best result from Fig. 3.10 (b), achieved with 100 iterations, each consisting of 10^5 time steps. The other trials represent example runs, keeping the total number of time steps limited to 10^7 while testing different segmentation routines. T2, T3, and T4 use $94 \cdot 10^3$, $200 \cdot 10^3$, and $840 \cdot 10^3$ time steps for a complete iteration, as reported in Tab. 3.2, with varying checkpoint repetitions to reach the target of 10^7 total steps. T1 and T2 have similar durations and comparable means, but segmenting the analysis (T2) allowed for the exploration of a greater number of local minima, resulting in solutions with lower energy and higher cuts. T3 performs the best, resulting in a good balance between longer segments and a higher number of repetitions, allowing for broader exploration of the landscape. This configuration is used in the following analysis. The dashed lines serve as reference points, representing the results from the original OIM study [105] and the reference values [80], which are used as benchmarks for subsequent accuracy evaluations. Figure 3.11 shows that during this analysis the proposed method overcame multiple times previous OIM results (red dashed line), however the reference value (green dashed line) has not been reached.

Although the presented analysis is conducted for only one exemplary problem, similar results have been obtained for different problems of the G-set.

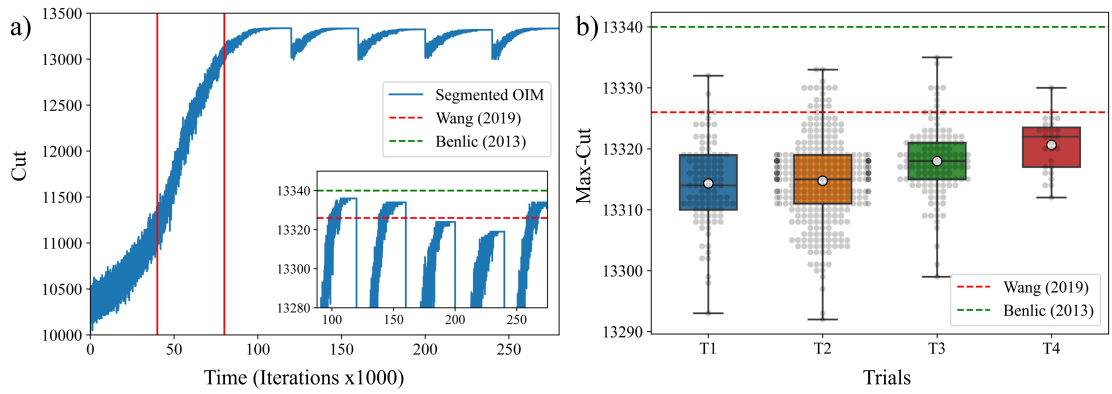

Figure 3.11: (a) An illustrative representation of the evaluated cut during the segmented analysis, with different checkpoints marked by vertical red lines. Five repetitions are shown, each starting from the last checkpoint. The inset provides a close-up view of the cuts. The dashed lines represent the maximum values obtained by the works introducing OIM [105] (in red) and the reference values [80] (in green). (b) A boxplot showing the results from four different trials: without segmentation (T1) and with varying segmentation approaches (T2, T3, and T4). The whiskers represent the full range of observations, with the median marked by a horizontal line and the mean indicated by a white dot.

Trial	Duration S1	Duration S2	Duration S3	Repetitions S1	Repetitions S2	Repetitions S3
T2	32k	32k	30k	5	6	10
T3	70k	70k	60k	5	5	5
T4	300k	300k	240k	3	3	3

Table 3.2: Durations in time steps and the number of repetitions for each checkpoint of the trials shown in Figure 3.11 (b). The analysis was carried out with a fixed total of 10^7 steps.

3.4.7 G set evaluation

Based on the previous considerations, the G-set problems were approached using the T3 segmented analysis with two minor variations, focusing primarily on the size of the analysis, fixed to 10^6, 10^7, and $4 \cdot 10^7$ time steps. These correspond to total computation times of approximately 2-5, 20-50, and 80-200 minutes per problem, respectively. These durations include the reading and writing processes that occur multiple times during each iteration.

The results of this analysis are presented in Table 3.3, alongside the results reported in [80] and [105].

The average accuracies computed for the G1-G54 problems demonstrate that the OIM achieves an average accuracy exceeding 99.5%, with the highest accuracies obtained from the longest runs, as expected.

When considering the G55-G81 problems, which are typically excluded from benchmark analyses due to their size, the accuracies slightly decrease but still surpass the 99% threshold on average.

In some instances, highlighted in bold in Table 3.3, the Max-Cut found was higher than the reference values.

In conclusion, the proposed method is effective for obtaining both fast, accurate results within minutes and highly accurate results within hours.

Problem	Benlic et al.[80]	Wang et al.[105]	This work (10^6 steps)	This work (10^7 steps)	This work ($4 \cdot 10^7$ steps)
G1	11624	11624	11624	11624	11624
G2	11620	11620	11615	11617	11617
G3	11622	11622	11615	11622	11622
G4	11646	11646	11640	11641	11644
G5	11631	11631	11631	11627	11631
G6	2178	2178	2176	2178	2178
G7	2006	2000	1997	1998	1998
G8	2005	2004	1992	2005	2004
G9	2054	2054	2043	2046	2048
G10	2000	2000	1997	1998	1999
G11	564	564	554	564	564
G12	556	556	552	556	556
G13	582	582	574	582	582
G14	3064	3061	3060	3062	3063
G15	3050	3049	3040	3050	3050
G16	3052	3052	3041	3052	3052
G17	3047	3046	3037	3045	3047

Continued on next page

Continued from previous page

Problem	Benlic et al.[80]	Wang et al.[105]	This work (10^6 steps)	This work (10^7 steps)	This work ($4 \cdot 10^7$ steps)
G18	992	990	988	991	991
G19	906	906	903	906	906
G20	941	941	941	941	941
G21	931	931	930	930	931
G22	13359	13356	13348	13357	13358
G23	13344	13333	13325	13336	13336
G24	13337	13329	13303	13335	13335
G25	13340	13326	13319	13326	13333
G26	13328	13313	13299	13324	13322
G27	3341	3323	3318	3341	3341
G28	3298	3285	3270	3297	3298
G29	3405	3396	3371	3396	3391
G30	3412	3402	3380	3412	3412
G31	3309	3296	3286	3306	3306
G32	1410	1402	1378	1402	1404
G33	1382	1374	1356	1374	1376
G34	1384	1374	1362	1380	1380
G35	7684	7675	7645	7684	7684
G36	7678	7663	7635	7673	7674
G37	7689	7679	7643	7680	7686
G38	7687	7679	7642	7685	**7688**
G39	2408	2404	2385	2408	2408
G40	2400	2389	2385	2395	2397

Continued on next page

Continued from previous page

Problem	Benlic et al.[80]	Wang et al.[105]	This work (10^6 steps)	This work (10^7 steps)	This work ($4 \cdot 10^7$ steps)
G41	2405	2401	2400	2404	2405
G42	2481	2469	2459	2472	2474
G43	6660	6660	6656	6656	6657
G44	6650	6648	6648	6649	6649
G45	6654	6653	6642	6653	6654
G46	6649	6649	6643	6646	6646
G47	6657	6656	6650	6656	6656
G48	6000	6000	6000	6000	6000
G49	6000	6000	6000	6000	6000
G50	5880	5874	5846	5880	5880
G51	3848	3846	3829	3847	3848
G52	3851	3848	3835	3847	3850
G53	3850	3846	3835	3847	3850
G54	3852	3850	3840	3851	3851
G55	10294	-	10201	10283	10289
G56	4012	-	3919	4004	4009
G57	3492	-	3406	3462	3470
G58	19263	-	19160	19263	**19271**
G59	6078	-	6001	6070	6069
G60	14176	-	14071	14169	14172
G61	5789	-	5661	5782	5788
G62	4868	-	4742	4830	4826
G63	26997	-	26870	26996	**27003**

Continued on next page

Continued from previous page

Problem	Benlic et al.[80]	Wang et al.[105]	This work (10^6 steps)	This work (10^7 steps)	This work ($4 \cdot 10^7$ steps)
G64	8735	-	8630	8715	8723
G65	5558	-	5404	5504	5510
G66	6360	-	6178	6292	6296
G67	6940	-	6780	6886	6890
G70	9541	-	9416	**9565**	**9562**
G72	6998	-	6804	6928	6934
G77	9926	-	9652	9842	9844
G81	14030	-	13638	13892	13910
Avg. Accuracy 1-54	-	99.87%	99.51%	99.92%	99.94%
Avg. Accuracy 1-81	-	-	99.18%	99.82%	99.85%

Table 3.3: Max-Cut values of the G-set problems achieved in [80] using BLS, in [105] using OIMs, and by the OIM implementation presented in this work under three configurations: 10^6, 10^7, and $4 \cdot 10^7$ total steps per problem.

3.5 Conclusion

In this chapter, we have demonstrated the applicability of oscillator-based Ising Machines to combinatorial optimization problems, specifically focusing on the Max-Cut problem. Through modeling approaches based on the Kuramoto and Slavin models, we observed that both approach the problems in a similar manner. However, the Kuramoto model proved to be more computationally efficient, making it preferable for large-scale implementations.

We developed an algorithmic solution that optimized both time and space complexities, achieving linear scalability with respect to the number of connections. Furthermore, the combination of noise annealing techniques and efficient GPU-accelerated implementations allowed us to solve problems with up to 20 million

nodes, surpassing the current state-of-the-art by an order of magnitude.

The segmented analysis approach further improved the accuracy and efficiency of the OIMs, ensuring that near-optimal solutions could be obtained within minutes for smaller accuracies, while more accurate solutions could be obtained over hours.

In conclusion, the results presented in this chapter establish OIMs as a viable and scalable solution for combinatorial optimization tasks, offering significant improvements in both computational speed (reduced to a linear dependency on the number of connections) and solution accuracy.

Chapter 4

Controlling vortex oscillators with an ac current input

This chapter explores the use of magnetic vortex oscillators to perform analog multiplication between an analog signal and a binary weight, a crucial operation in BNNs. This approach holds significant potential for developing fast and energy-efficient accelerators in artificial intelligence applications.

We demonstrate how a vortex oscillator, implemented using a MTJ stack with a diameter of less than 1 µm, can implement the writing and reading functionalities of binary weights solely through the application of an alternating current. These devices enable the multiplication of an analog input signal encoded in the amplitude of the ac current by a binary weight. The key advantage lies in the use of frequency-based current to write and read the weights, which allows for cascading multiple devices with different resonance frequencies. This eliminates the need for individual access to each device, enabling independent writing and reading in a chain configuration. This is particularly advantageous in potential 3D structures, where direct access to each device is impractical or inefficient.

The concept is simple and rooted in theoretical analysis, and we present experimental results from the implementation of a prototype consisting of a chain of three devices. These findings underscore the feasibility of using vortex oscillators to enhance AI hardware accelerators by simplifying architecture and reducing power consumption.

4.1 Vortex oscillators

Vortex oscillators are based on the dynamic behavior of magnetic vortices in ferromagnetic materials, typically found in thin films or nanodots. A magnetic vortex is a particular spin configuration where the magnetic moments in a thin, disk-shaped or ellyptical-shaped ferromagnet align in a circular fashion around a central core.

At the center of this structure, known as the vortex core, the magnetization points perpendicular to the plane, either upwards or downwards. Figure 4.1 (a) presents a sketch of the magnetization along the z-axis for the two polarity states, and (b) shows a top view of two micromagnetic analyses where the arrows represent the magnetization in local areas and the blue and red colors represent the positive and negative OOP component, respectively. The two examples presented also show a different chirality, meaning that the circular orientation of the IP components can be either clockwise or counterclockwise; this is another key property of magnetic vortices, which however, will not be analyzed in the work.

The vortex core can move within the plane of the magnetic material in response to external stimuli, such as spin-polarized currents or magnetic fields [123, 124]. This movement is called the gyrotropic motion, where the vortex core precesses around its equilibrium position [125]. The dynamics of this motion can be affected by external interactions such as magnetic fields, spin-transfer torque or spin-orbit torque[126, 127, 128].

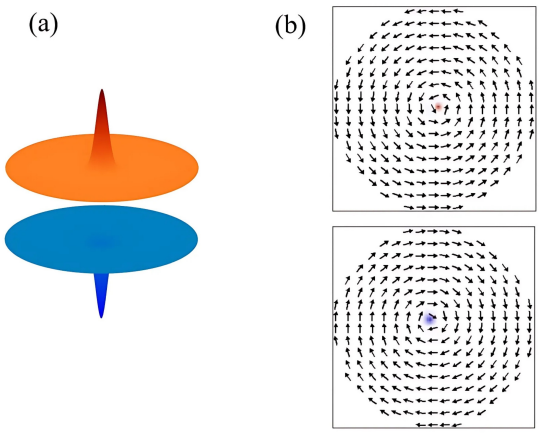

Figure 4.1: (a) Three dimensional sketch of the z-axis component of the magnetization for the two polarities. (b) Top view of the magnetization obtained with two micromangetic simulations of a circular FL with a vortex. The IP magnetization is represented in white, while the vortex cores are represented in red and blue for their OOP up and down components.

4.1.1 Gyrotropic Motion and Frequency

The gyrotropic motion of the vortex exhibits a circular trajectory on the magnetic disk. This motion occurs at a characteristic frequency, typically in the range of hundreds of MHz to several GHz, depending on the size and material properties of

the ferromagnetic disk. The frequency of vortex oscillators is determined by several factors:

- **Material properties**: The magnetic properties of the material, the fabrication process and imperfections influence the gyrotropic frequency.
- **Size of the device**: The core is subject to the boundary effects present at the edges of the devices and the size of the magnetic disk influences the range of motion and the frequency [126].
- **External biasing fields**: The application of an external magnetic field can tune the frequency by either pinning the vortex core or altering its equilibrium position [129, 130]. This effect will be studied more in detail in this work.
- **Applied currents**: Spin-polarized currents can induce sustained motion of the vortex core by exerting a torque on the spins, leading to continuous oscillations with the applied frequency. The amplitude of the oscillations increases for applied currents with frequencies close to the natural resonance frequency of the oscillator[123, 124, 128, 131, 132].

4.1.2 Spin-Transfer Torque and diode effect

The use of magnetic vortices as oscillators primarily relies on the STT effect. When a dc spin-polarized current passes through the magnetic material, the angular momentum of the electrons is transferred to the magnetic structure, causing the vortex to move. If this dc current is maintained, it sustains the motion of the vortex core, leading to continuous oscillations.

The oscillations generated by the vortex motion can be detected as a time-varying resistance in the device due to the TMR effect. This oscillating resistance produces an ac voltage signal, making vortex oscillators useful as high-frequency microwave signal generators.

In electronics, the term oscillator typically refers to a device that converts a dc input into an ac output. However, vortex oscillators can also function as diodes. Specifically, when provided with an ac input, vortex oscillators can generate a measurable dc voltage, and this phenomenon is known as spin-diode effect [60, 133]. Given a device with specific characteristics, the amplitude of the dc voltage generated depends on the natural frequency of oscillation of the device, and the input frequency and power [134]. The plot of several dc voltages measured for different values of input frequency is named spin-diode curve.

Figure 4.2 shows an example of an experimental spin-diode curve obtained for an input with -30dBm (or 1μW) of power, where the resonance frequency is observed at about 400 MHz.

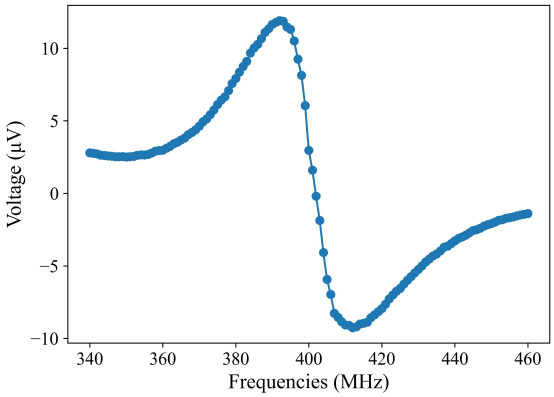

Figure 4.2: Plot of the experimental points of a spin-diode curve, obtained for an input with -30dBm (or 1μW) of power.

4.2 The influence of a dc magnetic field on the resonance frequency

A key aspect of this analysis is the interaction between the magnetic vortex and an OOP external magnetic field. As experimentally demonstrated in [129], the resonance frequencies of magnetic vortices in nanodisks subjected to a constant perpendicular magnetic field increases or decreases, depending on the polarity of the vortex core, and the magnitude of the shift depends on the applied field.

Figure 4.3, reported from the work [129], shows experimental proof of the resonance frequency splitting for different fields applied (a), and an example of the absorption signals obtained with the magnetic resonance force microscopy (b).

In essence, the resonance frequency of the gyrotropic motion is directly influenced by the applied perpendicular magnetic field, with two distinct behaviors observed for the different polarity states. In this work, we analyze MTJ vortex structures and observe a slight frequency shift in the resonance without the application of an external field, indicating the presence of a small intrinsic field component within the MTJ stack. Although this component is not included by design, it can still be useful for reading and writing the vortex core polarization, as will be demonstrated in the following sections.

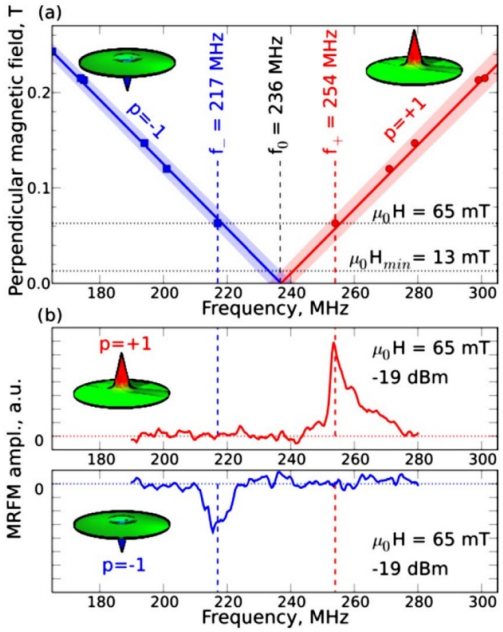

Figure 4.3: (a) Frequency splitting observed in experimental analyses applying an OOP magnetic field with different intensities. (b) Response obtained with the magnetic resonance force microscopy. Figure reported from [129].

4.3 The MTJ devices

Although the fabrication process was not part of this work, for a better understanding and reproducibility of the results, it is useful to provide some information about the key features of the measured devices.

For a device to be considered a suitable candidate for this analysis, it must meet the following key requirements:

- The design and structure should ensure that the vortex is the ground state of the magnetization at room temperature in the absence of an external magnetic field;
- It should exhibit high TMR to facilitate spin-diode conversion and enable reliable dc output measurements;
- The free layer should be as homogeneous as possible to minimize pinning effects that would alter the motion of the vortex core.

These specifications were achieved by the authors of the study [135], who also supplied the devices used in this work. They fabricated a circular MTJ structure featuring a CoFe(2.0 nm)/Ru(0.7 nm)/CoFeB(2.6 nm) synthetic antiferromagnet (SAF) as the pinned layer, and a CoFeSiB free layer, separated by a thin MgO(1.0 nm) layer. The amorphous nature of the free layer helps to reduce crystalline defects. An annealing treatment at 330°C for 2 hours in a 1 T magnetic field was applied to align the pinning layer and crystallize the MgO oxide barrier.

In summary, the measured device is a circular-section MTJ stack with various diameters, all below 1 µm, and a highly optimized free layer designed to minimize imperfections.

4.4 The measurement setup and routine

The measurement experiments consisted mainly in measuring the dc voltage obtained when an ac input is passed through the device.

The measurement setup consists of the following components:

- An AC current generator;
- A bias tee to separate the ac and dc components;
- A nanovoltmeter for dc voltage detection;
- A wire-bonded device.

This setup is intentionally simple, as no amplification stages or external magnetic fields are required, and one of the key advantages of this work is that the results can be replicated in an integrated solution.

Each device is individually wire-bonded to a gold-plated substrate, with connections of minimal length to reduce the absorption of interference signals in the frequency range of interest of 0.1 to 1 GHz. The gold plate is then connected to the instruments via RF cables.

The basic measurement routine involves a few straightforward steps:

- Injecting a current with a specific power and frequency into the sample;
- Waiting for a few milliseconds to avoid measuring transient phenomena;
- Detecting the resulting DC voltage.

This simple routine is used to obtain each data point of a spin-diode analysis, where the input frequency is swept, and forms the basis for more complex analyses discussed later in this work.

Some other configurations have been tested with the use of external antennas, high intensity magnetic fields and dc currents, but this work focuses on the application of only an ac current.

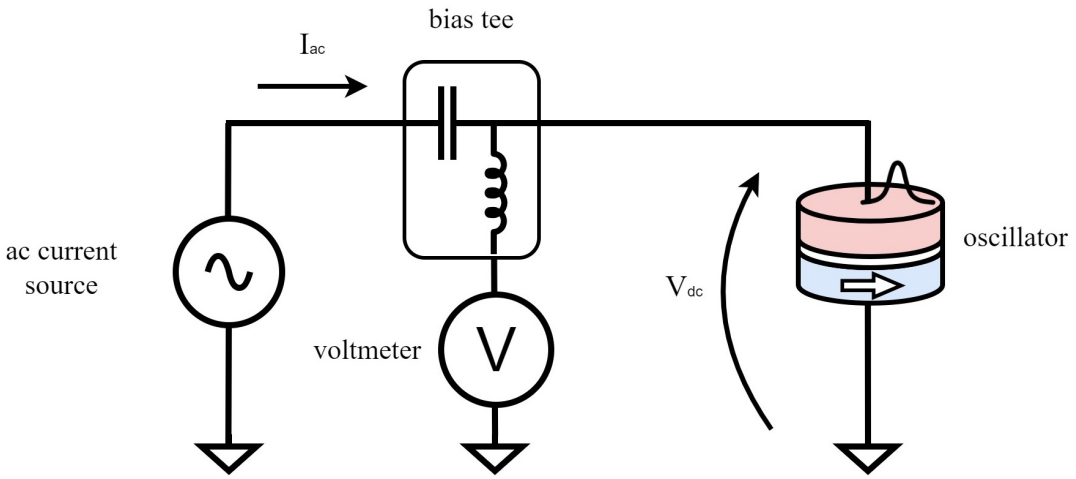

Figure 4.4: Schematic of the measurement setup.

4.4.1 Reading the core state

The spin-diode curve is a key component of this project, as it provides significant information about the devices in a very limited amount of time, as with just a few seconds of measurements, we can immediately determine the following:

- **Resonance frequency:** The curve directly reveals the resonance frequency of the device.

- **Gyrotropic motion as the main signal:** When a peak is observed between 50 MHz and 1000 MHz, we can safely assume that gyrotropic motion is being measured. If the resonance frequency largely shifts (>100MHz) when applying inputs with different powers, it may indicate the influence of pinning phenomena on the motion. A small shift of few MHz for different input powers is expected due to the nonlinear frequency shift described in previous chapters (see Eq.2.7).

- **Integrity of the MgO barrier:** This can be assessed by examining the amplitude of the output. If the amplitude falls within a specific range, the MgO barrier is considered intact; otherwise, the measured output is significantly lower. The MgO barrier is the most delicate part of the measured devices, and can be broken with a few volts[136]. Therefore, protection from static discharge is essential when handling MTJs.

- **Expulsion of the vortex core:** If the vortex core is expelled during the measurement, the spin-diode curve becomes flat near for applied frequencies close to the resonance, as illustrated in [137].

- **Core polarity:** The resonance frequency shift between the two polarity states leads to small but measurable changes in the spin-diode curve for the same applied powers, as it is shown in Fig. 4.5. This is a key result of this work.

Figure 4.5 shows an example of spin-diode curves evaluated on the same device initialized with different vortex polarities, where it's clearly visible the frequency shift. If one of the curves and its associated polarity state is known, it can be used as a reference to identify the polarity of the vortex core.

In practical implementations, when the frequency response of the device is known, only one dc voltage measure is necessary. The vertical line in Fig. 4.5 shows that for input currents with frequency of 394 MHz, the voltage measured is either positive or negative due to the core polarity, making the detection extremely simple.

Although magnetic devices are usually characterized by hysteretical behavior, when evaluating the spin-diode curve, the few milliseconds of delay between analyzing different points are orders of magnitude longer than the relaxation time of the structure, which returns to its ground state after each measurement. This ensures that if the spin-diode curve is measured multiple times under identical conditions, the output remains consistent, apart from minor measurement errors.

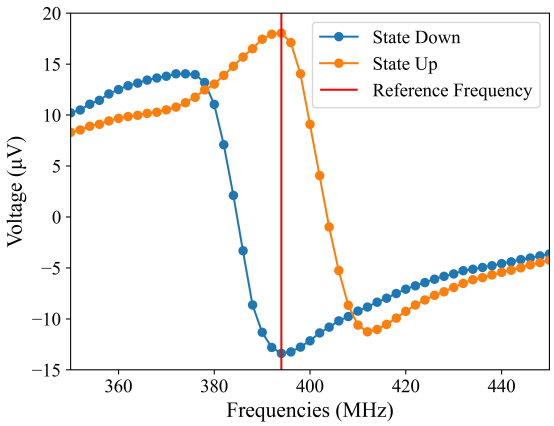

Figure 4.5: Plot of the experimental points of two spin-diode curves measured on the same device initialized with two distinct polarity states. For each point, the power of the ac current input is -30dBm.

4.5 Multiplication of the input value for a binary signal

The output dc voltage measured in the spin-diode curve depends linearly on the input power, and this phenomenon has been previously addressed in the context of analog multiply-and-accumulate (MAC) operations in neuromorphic implementations [138, 139]. In these implementations, the factors to be multiplied are encoded in the power of the injected signal and in the amplitude of a dc current passing through an antenna deposited over the devices, which is linearly shifting the resonance frequency of the devices.

We can apply the input/output linearity and the observed frequency shift to multiply a value encoded in the input power with the binary weight encoded in the vortex core polarity.

Figure 4.6 (a) shows the spin-diode curves obtained for various input amplitudes corresponding to the up (orange) and down (blue) core polarity states. In this plot, the amplitude of the response increases linearly with the applied input.

Figure 4.6 (b) shows the voltages measured when an input with variable power is applied with a fixed frequency, indicated by the vertical line in Figure 4.6 (a). In other words, Figure (b) presents only the measured points of (a) aligned over the vertical line. This Figure highlights the clear linear relationship between the input power and the output dc voltage, multiplied by the binary state encoded in the core.

From these analyses, we can conclude that a single device can not only store a binary weight, but also multiply an analog value by that weight. This is a key result in the context of current technology where the memory transfer is orders of magnitude larger than the computation times [140, 141], and having devices that are capable to both store information and manipulate them, like for the binary MAC, will significantly reduce this gap, commonly known as the Von Neumann bottleneck.

To provide context, current spintronic neuromorphic solutions for in-memory computation rely on controlling the movement of domain walls [142, 143, 144, 145], skyrmions [146, 147], the dynamics of superparamagnetic devices [148] and other effects [149, 150]. All of these implementations require precise fine-tuning of the devices' operational points through additional currents, magnetic fields, non-miniaturizable state detectors, or a combination of the three. These constraints diminish the advantages of leveraging physical phenomena at the nanoscale.

In contrast, the solution proposed in this work requires only an alternating current with a specific frequency to store the weight in the devices. This approach enables both the storage of binary weights and their use in multiplying input signals, significantly enhancing the potential for miniaturization in future neuromorphic devices.

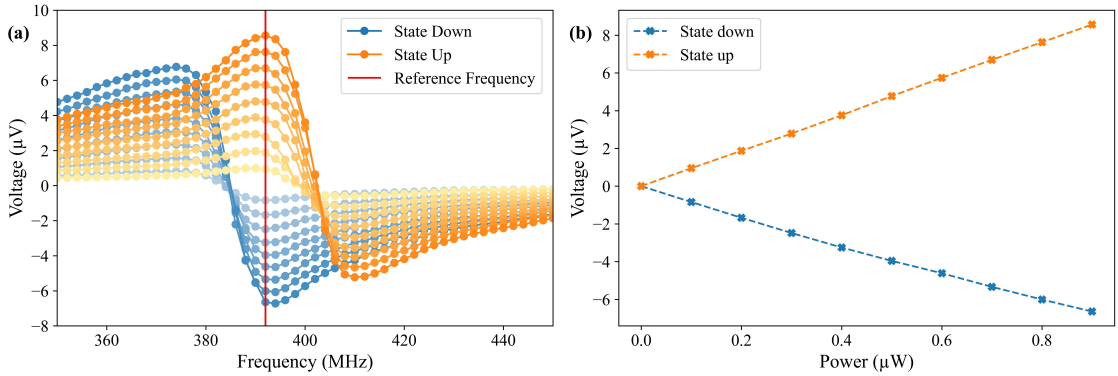

Figure 4.6: (a) Plot of experimental points of many spin-diode curves obtained linearly increasing powers from 0.1 μW to 1 μW for the two polarity states. The output voltage linearly increases with the injected power. The red line represents the frequency with maximum output difference between the states. (b) Plot of the dc voltages detected for different powers when the input signal has the frequency depicted by the red line in (a).

4.6 Writing the core state

In the state of the art, numerous studies have demonstrated how STT can be effectively used to switch the vortex core [124, 132, 151]. The application of a spin-polarized alternated current induces the gyrotropic motion of the vortex, which begins to oscillate with a frequency f corresponding to the one of the applied signal. The radius r of this oscillation depends on the applied power and the proximity to the resonance frequency. The tangential velocity of the vortex core can be expressed as $v = 2\pi fr$. As the velocity increases, simulations and theoretical analysis show that the variation in magnetization due to the core's motion induces a local out-of-plane (OOP) magnetization with an orientation opposite to that of the core [127]. At high velocities, this local field generates a vortex-antivortex pair with opposite polarities. As a result, the observed antivortex and core compensate each other, and a new vortex with opposite polarity remains. In essence, the polarity of the core switches when the tangential velocity exceeds a critical threshold. This phenomenon is well described in Fig.4.7 reported from [124], which shows a few timeframes of micromagnetic analyses showing the inversion of polarization of the core.

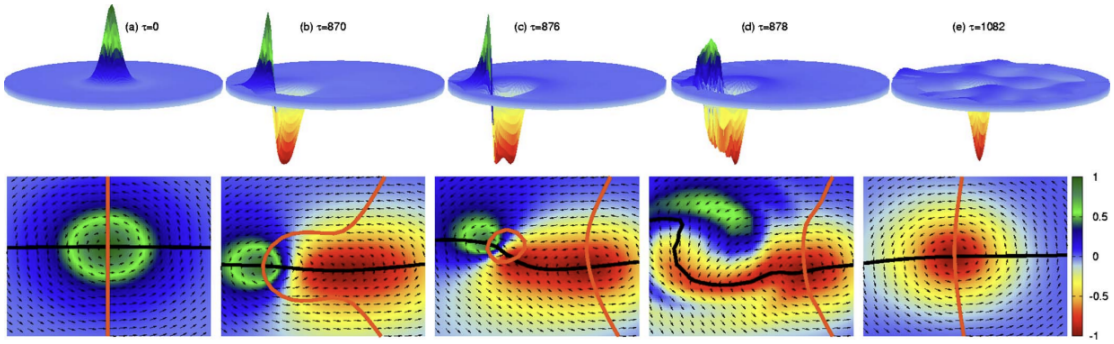

Figure 4.7: Simulation of the time evolution of the core switching process, 3D (upper diagrams) and top view (lower diagrams). Figure reported from [124].

In summary, by applying ac inputs with frequencies near resonance and sufficient power, the vortex core's polarity can be reversed. This phenomenon, combined with the resonance frequency shift, allows for selective manipulation of the devices.

Figure 4.8 (a) illustrates an hypotesis of a phase diagram depicting the switching mechanism as a function of input frequency and power in the absence of an applied OOP magnetic field. For input powers below the threshold, the tangential velocity does not surpass the critical value, and no switching occurs. In the colored region, the core switches continuously for the entire duration of the ac input. After the input is turned off, the core's final state is found randomly in either polarity.

Figure 4.8 (b) presents the hypotesis of a phase diagram when a small OOP magnetic field of the order of few mT is applied, depending on the initial vortex core

polarization. The switching regions are slightly shifted, and new regions emerge. If the vortex is initialized with a down polarization and an input with power and frequency within the orange region of the diagram is applied, no switching occurs, and the core retains its down polarization. However, if the initial polarization is up and the same input is applied, the core switches polarity. Once the core switches to down, the resonance frequency shifts, meaning that further application of the same frequency and power will no longer induce switching, leaving the final core state with down polarization. Thus, after applying inputs from the orange region, the final core polarization is always down, regardless of the initial state. The opposite holds true for the blue region, where the final polarization is always up.

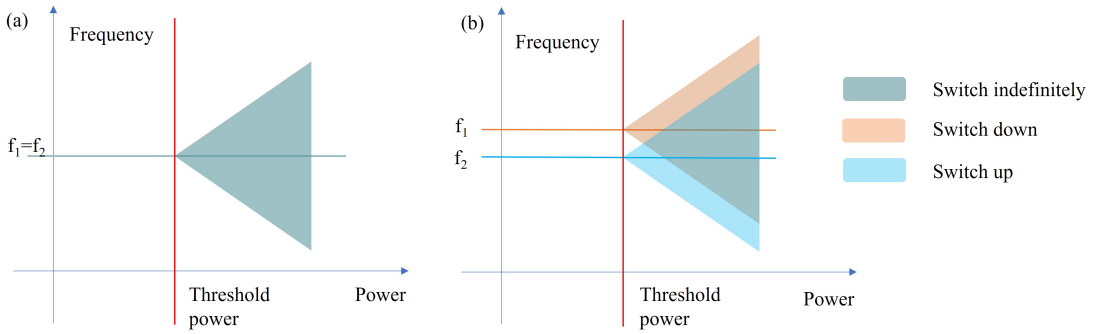

Figure 4.8: Hypotesis of phase diagrams describing the switching pheonomenon depending on the applied frequency and power in absence of magnetic field applied (a) and with a small OOP magnetic field (b), as in the observed devices. In the latter case, two new regions appear in which the system always switches from up to down or vice versa.

In conclusion, we successfully developed a method to control the vortex core polarity using an ac current with specific input power and frequency. This method is also useful for initializing the devices.

The detection of this phenomenon relies on recognizing the polarization state through the spin-diode effect described in the previous chapter.

For subsequent analyses, the following measurement routine was employed:

- Initialization of the core polarization;
- Detection of the current polarization state to verify correct initialization;
- Application of an input with specific frequency and power;
- Detection of the final polarization state.

This routine requires reliable initialization of the polarization states. If the blue and orange regions of the devices are unknown, two approaches can be employed:

- Find a point within the grey region of Fig. 4.8 (b) and initialize the core with a random polarization. During post-processing, divide the analyses based on

their initialized polarization state. This method is effective if a large number of measurements are made per point (e.g., 100 analyses per point, divided by the initialization of the core).

- Chose a power value above the threshold and apply varying frequencies to locate one point in the blue region and one in the orange region. This method requires some trial and error, and could be useful if a quick characterization is needed.

Figure 4.9 (a) shows an experimentally obtained switching diagram. The blue crosses represent points where, over 100 analyses, the switching from state down to up was achieved 100% of the time. The opposite case is represented by the orange crosses. The dots and crosses represent the bands of Fig 4.8 (b), confirming the presented hypothesis.

Figure 4.9 (b) presents the probability of switching between the two states as a function of frequency, with input power held constant at 1mW. The blue curve represents analyses where the initial polarization was set to down, while the orange curve corresponds to analyses where the initial polarization was up. Every point of these curves is the result of 100 iterations, and we can clearly se that there are two zones where the switching happens deterministically. These are the frequencies of interest for the writing of the core. In between these curves, the final state observed is random. Outside the presented frequencies, the switching never happens.

During both the writing and reading of the states, the ac current input is used near the resonance frequency; the key difference between the two analyses is that reading is performed using low power signals (on the order of -30 dBm, or 1 μW), while writing requires high power signals (on the order of 0 dBm, or 1 mW).

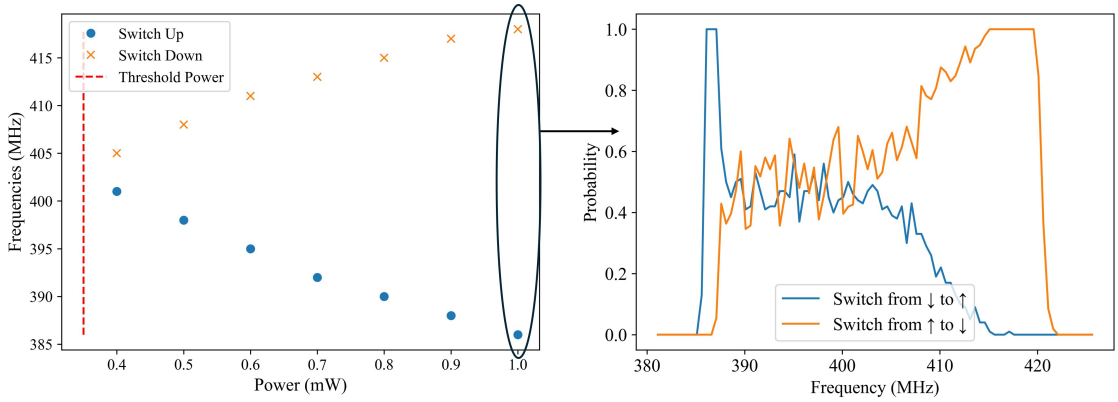

Figure 4.9: (a) Plot of the experimentally observed points with specific power and frequency for which the switching has been observed the 100% of times over 100 instances. The red dashed line represents the threshold power. (b) Plot of the probability of switching from down to up (blue) and from up to down (orange) when the power of the input signal is fixed at 1mW.

4.7 A chain of multiple devices

In the previous sections, we demonstrated how to read and write the core of a vortex oscillator, establishing that it is possible to successfully implement a memory block that operates solely on ac frequency currents.

When an input with a frequency outside the resonance range is applied, the system exhibits two key behaviors: in the reading case, the detected dc voltage is nearly zero, and in the writing case, the device does not switch. This indicates that the system is insensitive to inputs with frequencies that deviate from those near the device's resonance.

As a consequence, we can cascade multiple devices, each with a different resonance frequency, eliminating the need for individual access. This configuration allows us to read and write to each device independently by selecting the frequency and power of the input signals. In essence, this describes a multi-bit memory block that does not require direct access to each bit.

Figure 4.10 (a) and (b) illustrate two possible configurations. The first is the cascade setup, whose prototype results are presented in the following section. In the second configuration the devices are stacked vertically, creating a 3D structure. This design enhances space efficiency by significantly reducing the system's footprint. In both configurations, input signals are used to select the target device by tuning to the corresponding resonance frequency. In the figure different colors in the signals represent different frequencies, each corresponding to a specific device.

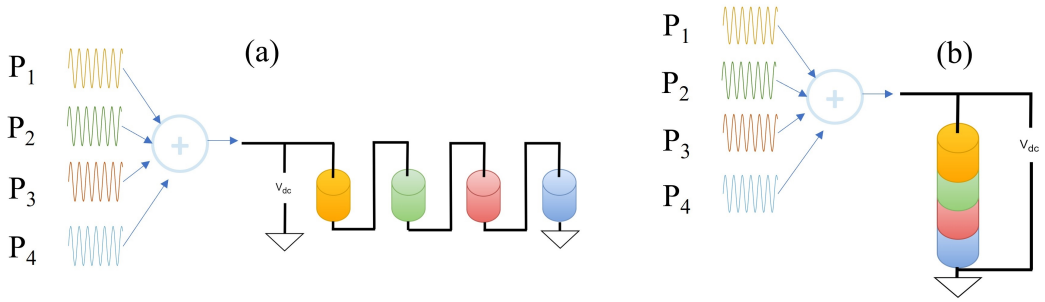

Figure 4.10: (a) Cascade configuration, every oscillator is connected with the other without individual access. (b) Stack configuration, the oscillators are deposited one over the other. In both cases, the frequency applied matches the resonance frequency of the target device.

4.7.1 Two-device chain

To demonstrate the system's ability to control multiple devices using only frequency signals, we implemented a chain of two devices, carefully selected for their distinct resonance frequencies. Figure 4.11 (a) presents four experimental spin-diode responses, obtained by initializing the oscillators in the four possible combinations

of states. The input signals have variable frequencies, with the power fixed at -30 dBm (1 μW). The resonance frequencies of the devices are approximately 370 MHz and 460 MHz.

In the green region of the plot the green and blue curves (both characterized by having the first oscillator in the down state) are nearly identical, as are the orange and red curves (with the first oscillator in the up state). This indicates that, regardless of the state of the second oscillator, the dc output voltage in this frequency range is primarily influenced by the state of the first oscillator. Thus, the core polarity of the first oscillator can be easily distinguished from a single measurement with an input frequency of around 370 MHz.

A similar pattern is observed in the blue part of the plot, where the orange and blue curves (second oscillator down) are closely aligned, as are the red and green curves (second oscillator up). Although the shift in resonance frequency between the two states is smaller for the second oscillator, a dc voltage measurement at an input frequency of 465 MHz can still reliably provide information about the oscillator's core state.

Figure 4.11 (b) shows the probability of switching from one state to another when applying inputs with a power of 0 dBm (1 mW) at various frequencies, where each point of the curves is the result of 100 trials. As previously mentioned, the state of each oscillator is detected both before and after each pulse. The legend indicates that the switching of one device occurs independently of the state of the other (denoted by an "X"). Of particular interest are the regions of the plot that show a 100% probability of switching, which demonstrates that we can deterministically control the core polarization of the device.

We can notice that, especially for the device with the lowest resonance frequency, the frequency windows of interest in (a) and (b) are slightly shifted. This is due to the nonlinear frequency shift, an effect studied in detail in the previous chapter, that introduces a dependence of the resonance frequency on the amplitude of the oscillations, which is closely related with the power of the ac input. This effect is not desirable for this specific implementation as, in future integrations, the windows of frequencies for the reading and writing functionalities must be juxtaposed to fit as many devices as possible in a chain.

The realization of this prototype required overcoming several challenges:

- Finding devices with the right resonance frequency. While knowing the size gives an indication of the resonance frequency range, each device has slightly different values of resonance frequencies, caused by device-to-device variations, leading to variations in their exact resonance frequency. For this application we need devices with resonance frequencies separated by more than 100 MHz, at least during the prototype phase.

- Not all devices successfully switch states. We observed that the ratio of useful devices increases with a larger device diameter, though the underlying

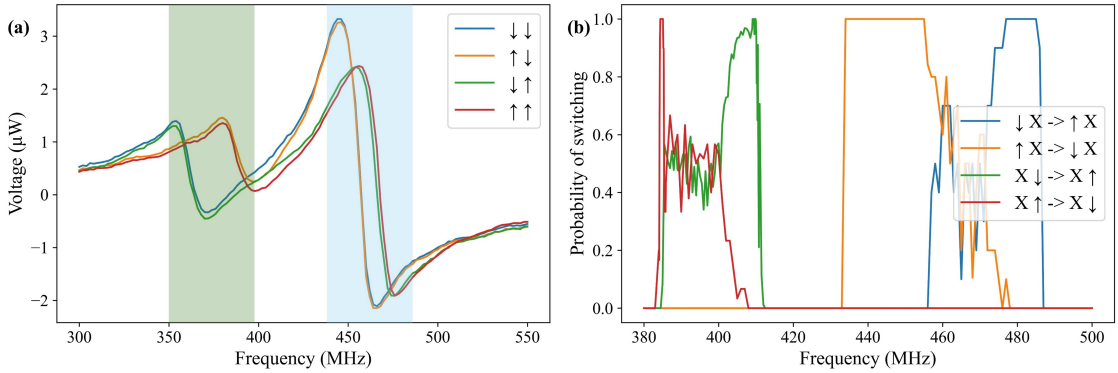

Figure 4.11: (a) Spin-diode experimental curve obtained initializing each oscillator in the four possible states. For each resonance frequency, the curves divide in two groups, depending on the state of a specific oscillator, and this can be used to detect easily the state of that oscillator. The green and blue areas represent the frequencies at which the two devices are susceptible to external inputs. (b) Probabilities of switching a specific oscillator from an initialized state. The windows of 100% probability show how we can control (write) deterministically the core polarity states. Each point is the result of 100 measurements.

cause remains unclear. Further analysis may provide insights to improve the likelihood of manufacturing devices that consistently switch.

- Among the devices that do switch, not all exhibit a frequency window with a 100% switching probability.

- The devices had to be wire-bonded, and constructing a circuit with multiple wire-bonded chiplets (each of them containing a target oscillator) increases the risk of breakage during the process (several devices have been broken in the process) and introduces additional noise due to resistance mismatches and power losses.

The next generation of prototypes should aim for an integrated design to minimize losses, caused by multiple wire bonds, chaining devices with increasing sizes, characterized by decreasing resonance frequencies.

4.7.2 Three-device chain

We present the results obtained by adding an additional device with a resonance frequency of approximately 220 MHz to the previously reported chain, thereby realizing a system with three cascaded devices.

Figure 4.12 (a) shows the spin-diode curves obtained after initializing the system in all eight possible configurations, as indicated in the legend, with an input power of -10 dBm (100 μW). As observed before, near the natural resonance frequency of each device, represented by the yellow, green and blue areas, the curves form two distinct groups that merge depending on the state of the respective oscillator, making the detection of the state straightforward at least for the yellow and green areas. The spin-diode curves associated with the blue area exhibit different trends for the two core polarities, however the resonance frequencies are very close, making the reading of the polarity harder with a one-shot voltage measurement. This device is the same analyzed in Fig. 4.11 (a) resonating in the blue window of frequencies, but the higher input power applied in this case causes the resonance frequencies of the two polarity states to overlap. The enhanced input power is necessary due to the cascading of three devices, which increases both the input resistance and the number of bonded wires, leading to losses caused by reflections resulting from impedance mismatches with the current source. These reflections are visible as small oscillations in the curves.

Figure 4.12 (b) illustrates the probability of switching the core polarity of each device following the application of an input signal with a power of 9 dBm (about 8 mW). In every case, a frequency window is observed where the probability of switching is high, indicating that we can write the polarity of each device's core individually and deterministically.

In summary, these early prototypes demonstrate the potential of vortex MTJ oscillators to be effectively controlled using only an ac frequency input.

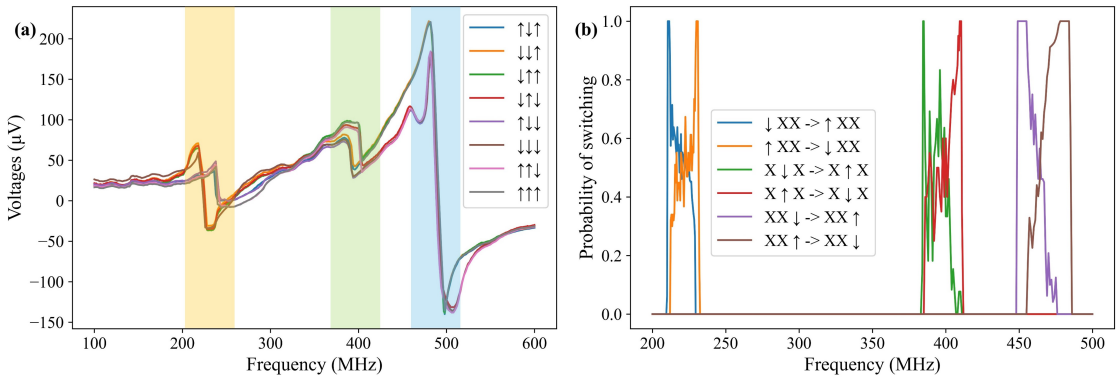

Figure 4.12: (a) Spin-diode curve obtained initializing each oscillator in the eight possible states, as in 4.11 (a). The yellow, green and blue areas represent the frequencies at which the three devices are susceptible to external inputs for the specific input power. (b) Probabilities of switching a specific oscillator from an initialized state, as in 4.11 (b). The windows of 100% probability show how we can control (write) deterministically the core polarity states. Each point is the result of 100 measurements.

4.8 Conclusion

We present a novel experimental observation of the intrinsic effect of an OOP magnetic field in a MTJ stack, designed for use in a memory device and as a neuromorphic node for multiplying binary weights and analog inputs.

Our experiments demonstrate how the vortex core polarization within an MTJ can be used to store binary weights. We introduce a method for reading and writing the vortex core using low-power and high-power ac current inputs, respectively, by tuning the input frequency, eliminating the need for external antennas or applied magnetic fields. This method enables multiplication between a binary weight and a continuous value, which is encoded in the power input to the system.

We successfully demonstrate this with prototypes of two- and three-device chains, achieving complete control without direct access to individual elements.

Future prototypes should prioritize integrating these devices into a single chip, reducing the need for wire bonds and addressing impedance mismatch challenges. Further refinement in controlling each device's switching dynamics will optimize performance and scalability of this memory architecture for practical applications. The incorporation of these devices into AI accelerators and systems demanding high-performance, and low-power computation will mark a significant advancement in the fields of spintronics and neuromorphic computing.

Conclusion

This work studied the use of spintronic oscillators as accelerators in computer vision, for efficiently finding high-quality solutions to combinatorial optimization problems, and for the implementation of in-memory computing devices.

We observed that using parabolic (or cosinusoidal) phenomena, while less precise than conventional implementations, proves highly effective in artificial intelligence applications, where the network compensate for any introduced imprecision. Extending these analyses to other devices in the future would be valuable; for example, exploring the quadratic relationship between drain current and gate voltage in MOSFET transistors could reveal a very size- and power-efficient implementation.

Our approach successfully tackled Max-Cut problems with up to 20 million nodes using an architecture optimized for large, sparse problems and oscillator simulations based on the Kuramoto model. This method also achieved high accuracy ($>$99.5%) compared to a reference solver. Given this optimization, it would be interesting to apply the system to practical problems requiring a solver capable of computing extremely large graphs. From a technical perspective, it would be interesting to test this architecture in a multi-GPU environment, and with an optimized C++ code on a system with terabytes of RAM.

Finally, we experimentally demonstrated that vortex MTJs can be used to implement an effective multi-bit memory device controllable by frequency inputs, enabling multiplication of an analog input and a binary weight. We developed an initial prototype with three chained devices, achieving control without individual device access. Future work could test the scalability of this architecture to determine the maximum number of devices that can be controlled simultaneously, necessitating the design of an integrated prototype.

In conclusion, this thesis has explored the potential of spintronic oscillators for diverse applications, highlighting the advantages of analog computing for achieving compact, low-power, and high-speed implementations.

Bibliography

[1] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[2] Giorgio Bertotti. *Hysteresis in Magnetism: For Physicists, Materials Scientists, and Engineers*. Academic Press, San Diego, CA, 1998.

[3] W. F. Brown. Micromagnetics, domains, and resonance. *Journal of Applied Physics*, 30:S62–S69, apr 1959.

[4] W. F. Brown. *Micromagnetics*. Interscience Tracts on Physics and Astronomy. Interscience Publishers, New York, 1963.

[5] G. Finocchio, B. Azzerboni, G. D. Fuchs, R. A. Buhrman, and L. Torres. Micromagnetic modeling of magnetization switching driven by spin-polarized current in magnetic tunnel junctions. *J. Appl. Phys.*, 101:063914, 2007.

[6] J. C. Slonczewski. Current-driven excitation of magnetic multilayers. *J. Magn. Magn. Mater.*, 159:L1, 1996.

[7] L. Berger. Emission of spin waves by a magnetic multilayer traversed by a current. *Phys. Rev. B*, 54:9353, 1996.

[8] A. Meo et al. Spin-transfer and spin-orbit torques in the landau–lifshitz–gilbert equation. *Journal of Physics: Condensed Matter*, 35:025801, jan 2023.

[9] M. N. Baibich, J. M. Broto, A. Fert, F. Nguyen Van Dau, F. Petroff, P. Etienne, G. Creuzet, A. Friederich, and J. Chazelas. Giant magnetoresistance of (001)fe/(001)cr magnetic superlattices. *Physical Review Letters*, 61(21):2472–2475, 1988.

[10] S. M. Thompson. The discovery, development and future of gmr: The nobel prize 2007. *Journal of Physics D: Applied Physics*, 41(9):093001, mar 2008.

[11] J. M. D. Coey. *Magnetism and Magnetic Materials*. Cambridge University Press, 2010.

[12] E. Y. Tsymbal and D. G. Pettifor. Perspectives of giant magnetoresistance. *Solid State Physics*, 56:113–237, 2001.

[13] S. Mao, Yonghua Chen, Feng Liu, Xingfu Chen, Bin Xu, P. Lu, M. Patwari, Haiwen Xi, Clifton H. Chang, B. Miller, Dave Menard, B. Pant, Jay Loven, K. Duxstad, Shaoping Li, Zheng-Jun Zhang, A. Johnston, R. Lamberton, M. Gubbins, T. Mclaughlin, J. Gadbois, Juren Ding, Bill Cross, S. Xue, and

P. Ryan. Commercial tmr heads for hard disk drives: characterization and extendibility at 300 gbit/in. *IEEE Transactions on Magnetics*, 42:97–102, 2006.

[14] S. Ikeda, J. Hayakawa, Y. Ashizawa, Y. M. Lee, K. Miura, H. Hasegawa, M. Tsunoda, F. Matsukura, and H. Ohno. Tunnel magnetoresistance of 604 *Applied Physics Letters*, 93:082508, 2008.

[15] T. Scheike, Z. Wen, H. Sukegawa, and S. Mitani. 631% room temperature tunnel magnetoresistance with large oscillation effect in CoFe/MgO/CoFe(001) junctions. *Applied Physics Letters*, 122(11):112404, 03 2023.

[16] Abdelrahman G. Qoutb and Eby G. Friedman. Mtj magnetization switching mechanisms for iot applications. *Proceedings of the 2018 Great Lakes Symposium on VLSI*, 2018.

[17] T. Nakano, M. Oogane, T. Furuichi, and Y. Ando. Magnetic tunnel junctions using perpendicularly magnetized synthetic antiferromagnetic reference layer for wide-dynamic-range magnetic sensors. *Applied Physics Letters*, 110:012401, 2017.

[18] B. Fang and Z. Zeng. Spin transfer nano-oscillators. *Chinese Science Bulletin*, 2014.

[19] J. Torrejon, M. Riou, F. Abreu Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664):428–431, 2017.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[23] Brady D. Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, S. Shimray, and Ziang Wang. Chatgpt and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74:570 – 581, 2023.

[24] Viriya Taecharungroj. "what can chatgpt do?" analyzing early reactions to the innovative ai chatbot on twitter. *Big Data Cogn. Comput.*, 7:35, 2023.

[25] D. Marković, F. A. Mizrahi, D. Querlioz, and J. Grollier. Physics for neuromorphic computing. *Nature Reviews Physics*, 2(9):499–510, 2020.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[27] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. NIPS Deep Learning Workshop.

[29] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

[30] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 2672–2680, 2014.

[32] Ahmet İlker Tekkeşin. Artificial intelligence in healthcare: Past, present and future. *The Anatolian Journal of Cardiology*, 22:8–9, 2019.

[33] L. Fei-Fei, J. Deng, and K. Li. Imagenet: Constructing a large-scale image database. *J. Vis.*, 9:1037, 2010.

[34] Y. S. Resheff Witten, T. Hope and I. Lieder. *TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning*. O'Reilly Media, Sebastopol, CA, 2017. Chapter 4: Building Deep Learning Models in TensorFlow.

[35] S. H. Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119, 2020.

[36] K. O'Shea and R. Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[37] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[38] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] Andrei Slavin and Vasil Tiberkevich. Nonlinear auto-oscillator theory of microwave generation by spin-polarized current. *IEEE Transactions on Magnetics*, 45(4):1875–1918, 2009.

[40] K. Y. Guslienko, G. R. Aranda, and J. Gonzalez. Spin torque and critical currents for magnetic vortex nano-oscillator in nanopillars. *Journal of Physics:*

Conference Series, 292(1):012006, apr 2011.

[41] A. Manchon, J. Železný, I. M. Miron, T. Jungwirth, J. Sinova, A. Thiaville, K. Garello, and P. Gambardella. Current-induced spin-orbit torques in ferromagnetic and antiferromagnetic systems. *Rev. Mod. Phys.*, 91:035004, Sep 2019.

[42] S. Wittrock, S. Perna, R. Lebrun, et al. Non-hermiticity in spintronics: oscillation death in coupled spintronic nano-oscillators through emerging exceptional points. *Nature Communications*, 15:971, 2024.

[43] T. Gabara. An 0.25 /spl mu/m cmos injection locked 5.6 gb/s clock and data recovery cell. pages 84–87, 1999.

[44] Z. Liu and R. Slavík. Optical injection locking: From principle to applications. *Journal of Lightwave Technology*, 38(1):43–59, 2020.

[45] D. M. Chiarulli, B. Jennings, Y. Fang, A. Seel, and S. P. Levitan. A computational primitive for convolution based on coupled oscillator arrays. 07-10-July:125–130, 2015.

[46] I. A. Young D. E. Nikonov and G. I. Bourianoff. Convolutional networks for image processing by coupled oscillator arrays. *arXiv preprint arXiv:1409.4469*, 2014.

[47] 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.

[48] L. Mazza, V. Puliafito, M. Carpentieri, and G. Finocchio. Robustness of using degree of match in performing analog multiplication with spin-torque oscillators. *Solid-State Electronics*, 183:108045, 2021.

[49] S. Kaka, M. R. Pufall, W. H. Rippard, T. J. Silva, S. E. Russek, and J. A. Katine. Mutual phase-locking of microwave spin torque nano-oscillators. *Nature*, 437(7057):389–392, 2005.

[50] Z. Zeng, G. Finocchio, B. Zhang, P. K. Amiri, J. A. Katine, I. N. Krivorotov, Y. Huai, J. Langer, B. Azzerboni, K. L. Wang, and H. Jiang. Ultralow-current-density and bias-field-free spin-transfer nano-oscillator. *Sci. Rep.*, 3:1426, 2013.

[51] K. Yogendra, D. Fan, Y. Shim, M. Koo, and K. Roy. Computing with coupled spin torque nano oscillators. 25–28-Janu:312–317, 2016.

[52] 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. *Appl. Phys. Lett.*, 92(23):232504, 2008.

[53] B. Fang, M. Carpentieri, X. Hao, H. Jiang, J. A. Katine, I. N. Krivorotov, B. Ocker, J. Langer, K. L. Wang, B. Zhang, B. Azzerboni, P. K. Amiri, G. Finocchio, and Z. Zeng. Giant spin-torque diode sensitivity in the absence of bias magnetic field. *Nat. Commun.*, 7:11259, 2016.

[54] V. Puliafito, L. Sanchez-Tejerina, M. Carpentieri, B. Azzerboni, and

G. Finocchio. Modulation, injection locking, and pulling in an antiferromagnetic spin-orbit torque oscillator. *IEEE Trans. Magn.*, 57:4100106, 2021.

[55] E. Raimondo Eleonora A. Giordano Z. Zeng L. Mazza, V. Puliafito, M. Carpentieri, and G. Finocchio. Computing with injection-locked spintronic diodes. *Phys. Rev. Appl.*, 17:014045, Jan 2022.

[56] E. Kussul and T. Baidyk. Improved method of handwritten digit recognition tested on mnist database. *Image Vis. Comput.*, 22:971, 2004.

[57] E. Raimondo, A. Giordano, A. Grimaldi, V. Puliafito, M. Carpentieri, Z. Zeng, R. Tomasello, and G. Finocchio. Reliability of neural networks based on spintronic neurons. *IEEE Magn. Lett.*, 12:6102805, 2021.

[58] J. C. Slonczewski. Currents, torques, and polarization factors in magnetic tunnel junctions. *Phys. Rev. B*, 71:024411, 2005.

[59] S. Louis, V. Tyberkevych, J. Li, I. Lisenkov, R. Khymyn, E. Bankowski, T. Meitzler, I. Krivorotov, and A. Slavin. Low power microwave signal detection with a spin-torque nano-oscillator in the active self-oscillating regime. *IEEE Trans. Magn.*, 53:1400804, 2017.

[60] A. A. Tulapurkar, Y. Suzuki, A. Fukushima, H. Kubota, H. Maehara, K. Tsunekawa, D. D. Djayaprawira, N. Watanabe, and S. Yuasa. Spin-torque diode effect in magnetic tunnel junctions. *Nature*, 438:339, 2005.

[61] Y. Zhou, J. Persson, and J. Åkerman. Intrinsic phase shift between a spin torque oscillator and an alternating current. *J. Appl. Phys.*, 101:09A510, 2007.

[62] Y. Zhou, V. Tiberkevich, G. Consolo, E. Iacocca, B. Azzerboni, A. Slavin, and J. Åkerman. Oscillatory transient regime in the forced dynamics of a nonlinear auto oscillator. *Phys. Rev. B*, 82:012408, 2010.

[63] A. Slavin and V. Tiberkevich. Nonlinear auto-oscillator theory of microwave generation by spin-polarized current. *IEEE Trans. Magn.*, 45:1875, 2009.

[64] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, Virtual Event, Canada, 2021. ACM.

[65] P. Wallis Z. Allen-Zhu Y. Li S. Wang L. Wang Lu E. J. Hu, Y. Shen and W. Chen. Lora: Low-rank adaptation of large language models, 2021.

[66] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929, 2014.

[68] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness.* 1979.

[69] T. et al. Inagaki. A coherent ising machine for 2000-node optimization problems. *Science (80-.)*, 354:603–606, 2016.

[70] Y. Yamamoto, K. Aihara, T. Leleu, K. Kawarabayashi, S. Kako, M. Fejer, K. Inoue, and H. Takesue. Coherent ising machines—optical neural networks operating at the quantum limit. *Npj Quantum Inf.*, 3:49, 2017.

[71] H. Goto, K. Tatsumura, and A. R. Dixon. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems. *Sci. Adv.*, 5:1–9, 2019.

[72] N. A. et al. Aadit. Massively parallel probabilistic computing with sparse ising machines. *Nat. Electron.*, 5:460–468, 2022.

[73] A. et al. Grimaldi. Spintronics-compatible approach to solving maximum-satisfiability problems with probabilistic computing, invertible logic, and parallel tempering. *Phys. Rev. Appl.*, 17:024052, 2022.

[74] W. Ben-Ameur, A. R. Mahjoub, and J. Neto. The maximum cut problem. 9781848216:131–172, 2014.

[75] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of ising spin glasses: New experimental results with a branch-and-cut algorithm. *J. Stat. Phys.*, 80:487, 1995.

[76] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta. Stochastic p-bits for invertible logic. *Phys. Rev. X*, 7:031014, 2017.

[77] R. M. Karp. Reducibility among combinatorial problems. page 85–103, 1972.

[78] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[79] G-set problems, `https://web.stanford.edu/~yyye/yyye/Gset/`,.

[80] U. Benlic and J.-K. Hao. Breakout local search for the max-cut problem. *Eng. Appl. Artif. Intell.*, 26:1162–1173, 2013.

[81] Luciano Mazza, Eleonora Raimondo, Andrea Grimaldi, and Vito Puliafito. Simulated oscillator-based ising machine for two million nodes max-cut problems. pages 1037–1041, 2023.

[82] A. Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014.

[83] D. P. Williamson and M. Goemans. Improved maximum approximation algorithms for using cut and satisfiability programming problems semidefinite. *Science (80-.)*, 42:1115–1145, 1994.

[84] P. Berman and M. Karpinski. On some tighter inapproximability results (extended abstract). pages 200–209, 1999.

[85] E. Halperin, D. Livnat, and U. Zwick. Max cut in cubic graphs. *Journal of Algorithms*, 53(2):169–185, 2004.

[86] R. et al. Hamerly. Experimental investigation of performance differences between coherent ising machines and a quantum annealer. *Sci. Adv.*, 5:eaau0823, 2019.

[87] J. Laydevant, D. Marković, and J. Grollier. Training an ising machine with equilibrium propagation. *ArXiv*, abs/2305.18321, 2023.

[88] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, 339(6124):1169–1174, 2013.

[89] T. et al. Honjo. 100,000-spin coherent ising machine. *Sci. Adv.*, 7, 2021.

[90] P. L. et al. McMahon. A fully programmable 100-spin coherent ising machine with all-to-all connections. *Science (80-.)*, 354:614–617, 2016.

[91] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog. Analog coupled oscillator based weighted ising machine. *Sci. Rep.*, 9:14786, 2019.

[92] K. Ochs, B. Al Beattie, and S. Jenderny. An ising machine solving max-cut problems based on the circuit synthesis of the phase dynamics of a modified kuramoto model. pages 982–985, 2021.

[93] Nilamani Behera, A. Chaurasiya, Victor H. Gonz'alez, A. Litvinenko, L. Bainsla, Akash Kumar, A. Awad, H. Fulara, and J. Aakerman. Ultra-low current 10 nm spin hall nano-oscillators. *Advanced materials*, page e2305002, 2023.

[94] A. Litvinenko, R. Khymyn, V. H. González, A. A. Awad, V. Tyberkevych, A. Slavin, and J. Åkerman. A spinwave ising machine. *ArXiv*, 2209.04291, 2022.

[95] A. Litvinenko, R. Khymyn, R. Ovcharov, and J. Åkerman. A 50-spin surface acoustic wave ising machine. page 1–15, 2023.

[96] B. C. McGoldrick, J. Z. Sun, and L. Liu. Ising machine based on electrically coupled spin hall nano-oscillators. *Phys. Rev. Appl.*, 17:14006, 2022.

[97] D. I. Albertsson, M. Zahedinejad, A. Houshang, R. Khymyn, J. Åkerman, and A. Rusu. Ultrafast ising machines using spin torque nano-oscillators. *Appl. Phys. Lett.*, 118(11):112404, 2021.

[98] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno. A 20k-spin ising chip to solve combinatorial optimization problems with cmos annealing. *IEEE Journal of Solid-State Circuits*, 51(1):303–309, Jan 2016.

[99] T. G. J. Myklebust. Solving maximum cut problems by simulated annealing, May 2015. Available online at http://arxiv.org/abs/1505.03068.

[100] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics*, 7(APR), Apr 2019.

[101] H. Goto. Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network. *Scientific Reports*, 6(1):21686, Feb 2016.

[102] T. Kanao and H. Goto. Simulated bifurcation assisted by thermal fluctuation. *Commun. Phys.*, 5:153, 2022.

[103] T. Wang, L. Wu, P. Nobel, and J. Roychowdhury. Solving combinatorial optimisation problems using oscillator based ising machines. *Nat. Comput.*, 20:287, 2021.

[104] I. Bello, H. Pham, Q. V. Le, M. Norouzi, S. Bengio, and G. Brain. Neural combinatorial optimization with reinforcement learning, 2017. Workshop track at ICLR 2017.

[105] T. Wang and J. Roychowdhury. Oim: Oscillator-based ising machines for solving combinatorial optimisation problems. 11493 LNCS:232–256, 2019.

[106] R. Sharma, N. Sisodia, P. Dürrenfeld, J. Åkerman, and P. K. Muduli. Time-domain stability of parametric synchronization in a spin-torque nano-oscillator based on a magnetic tunnel junction. *Phys. Rev. B*, 96:024427, 2017.

[107] A. Houshang, M. Zahedinejad, S. Muralidhar, R. Khymyn, M. Rajabali, H. Fulara, A. A. Awad, J. Åkerman, J. Chęcinski, and M. Dvornik. Phase-binarized spin hall nano-oscillator arrays: Towards spin hall ising machines. *Phys. Rev. Appl.*, 17:014003, 2022.

[108] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, pages 420–422. Springer, 1975.

[109] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, and R. Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.*, 77:137, 2005.

[110] W. A. et al. Borders. Integer factorization using stochastic magnetic tunnel junctions. *Nature*, 573:390–393, 2019.

[111] S. et al. Chowdhury. A full-stack view of probabilistic computing with p-bits: Devices, architectures, and algorithms. *IEEE J. Explor. Solid-State Comput. Devices Circuits*, 9:1–11, 2023.

[112] L. Mazza, E. Raimondo, A. Grimaldi, and V. Puliafito. Simulated oscillator-based ising machine for two million nodes max-cut problems. page 1037–1041, 2023.

[113] V. E. Demidov, S. Urazhdin, and S. O. Demokritov. Direct observation and mapping of spin waves emitted by spin-torque nano-oscillators. *Nat. Mater.*, 9:984, 2010.

[114] B. Georges, J. Grollier, M. Darques, V. Cros, C. Deranlot, B. Marcilhac, G. Faini, and A. Fert. Coupling efficiency for phase locking of a spin transfer nano-oscillator to a microwave current. *Phys. Rev. Lett.*, 101:017201, 2008.

[115] T. Chen, A. Eklund, E. Iacocca, S. Rodriguez, B. G. Malm, J. Akerman, and A. Rusu. Comprehensive and macrospin-based magnetic tunnel junction spin torque oscillator model-part ii: Verilog-a model implementation. *IEEE Trans. Electron Devices*, 62:1045, 2015.

[116] P. A. Khalili K. H. Cheung J. A. Katine J. Langer K. L. Wang Z. M. Zeng, P. Upadhyaya and H. W. Jiang. Enhancement of microwave emission in magnetic tunnel junction oscillators through in-plane field orientation. *Applied Physics Letters*, 99(3):032503, 2011.

[117] E. Raimondo P. Tullo D. Rodrigues K. Y. Camsari Kerem V. Crupi M. Carpentieri V. Puliafito A. Grimaldi, L. Mazza and G. Finocchio. Evaluating spintronics-compatible implementations of ising machines. *Phys. Rev. Appl.*, 20:024005, Aug 2023.

[118] M. Etscheid and H. Röglin. Smoothed analysis of local search for the maximum-cut problem. *ACM Trans. Algorithms*, 13:1–12, 2017.

[119] Ibrahim H. Osman and Gilbert Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5):513–628, 1996.

[120] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[121] W. Duckworth and M. Zito. Large independent sets in random regular graphs. *Theor. Comput. Sci.*, 410:5236–5243, 2009.

[122] M. J. A. Schuetz, J. K. Brubaker, and H. G. Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nat. Mach. Intell.*, 4:367–377, 2022.

[123] H. Stoll K. W. Chou T. Tyliszczak R. Hertel M. Fähnle B. Van Waeyenberge, A. Puzic et al. Magnetic vortex core reversal by excitation with short bursts of an alternating field. *Nature*, 444(7118):461–464, 2006.

[124] Y. Gaididei D. D. Sheka and F. G. Mertens. Current induced switching of vortex polarity in magnetic nanodisks. *Applied Physics Letters*, 91(8):082509, August 2007. Research Article.

[125] M. T. Bryan G. Hrkac, P. S. Keatley and K. Butler. Magnetic vortex oscillators. *Journal of Physics D: Applied Physics*, 48(45):453001, 2015. Published 6 October 2015.

[126] V. Novosad Y. Otani H. Shima K. Y. Guslienko, B. A. Ivanov and K. Fukamichi. Eigenfrequencies of vortex state excitations in magnetic submicron-size disks. *Journal of Applied Physics*, 91(10):8037–8042, 2001.

[127] K. Lee K. Y.. Guslienko and S. Kim. Dynamic origin of vortex core switching in soft magnetic nanodots. *Physical Review Letters*, 100(2):027203, 2008.

[128] Y. Nakatani K. Kobayashi H. Kohno K. Yamada, S. Kasai and T. Ono. Switching magnetic vortex core by a single nanosecond current pulse. *Applied Physics Letters*, 91:062507, 2007.

[129] O. Klein A. Riegler F. Lochner G. Schmidt L. W. Molenkamp V. S. Tiberkevich B. Pigeau, G. de Loubens and A. N. Slavin. A frequency-controlled magnetic vortex memory. *Applied Physics Letters*, 96(13):132506, 2010.

[130] M. Fähnle R. Hertel, S. Gliga and C. M. Schneider. Current-induced magnetic vortex core switching in a permalloy nanodisk. *Applied Physics Letters*, 90(14):142512, 2007.

[131] E. Girgis J. Kolthammer Y.K. Hong A. Lyle B.C. Choi, J. Rudge. Spin-current pulse induced switching of vortex chirality in permalloy/cu/co nanopillars. *Applied Physics Letters*, 91(2):022501, 2007.

[132] F. G. Mertens J. Caputo, Y. Gaididei and D. D. Sheka. Vortex polarity switching by a spin-polarized current. *Physical Review Letters*, 98(5):056604, 2007.

[133] J. Grollier, V. Cros, and A. Fert. Synchronization of spin-transfer oscillators driven by stimulated microwave currents. *Physical Review B*, 73(6):060409, 2006.

[134] 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, 2010.

[135] L. C. Benetti A. Schulman P. Anacleto M. S. Claro I. Caha F. L. Deepak E. P. A. S. Jenkins, L. Martins and R. Ferreira. The impact of local pinning sites in magnetic tunnel junctions with non-homogeneous free layers. *Communications Materials*, 5:7, 2024.

[136] A. Thomas O. Schebaum A. A. Khan, J. Schmalhorst and G. Reiss. Analysis of dielectric breakdown in cofeb/mgo/cofeb magnetic tunnel junction. *Journal of Applied Physics*, 103(12):123705, 2008.

[137] A. S. Jenkins, R. Lebrun, E. Grimaldi, S. Tsunegi, P. Bortolotti, H. Kubota, K. Yakushiji, A. Fukushima, G. De Loubens, O. Klein, S. Yuasa, and V. Cros. Spin-torque resonant expulsion of the vortex core for an efficient radiofrequency detection scheme. *Nat. Nanotechnol.*, 11:360, 2016.

[138] N. Leroux, D. Markovic, E. Martin, T. Petrisor, D. Querlioz, A. Mizrahi, and J. Grollier. Radio-frequency multiply-and-accumulate operations with spintronic synapses. *Phys. Rev. Appl.*, 15:034067, 2021.

[139] D. Marković D. Sanz-Hernández J. Trastoy P. Bortolotti L. Martins A. Jenkins R. Ferreira N. Leroux, A. Mizrahi and J. Grollier. Hardware realization of the multiply and accumulate operation on radio-frequency signals with magnetic tunnel junctions. *Neuromorphic Computing and Engineering*, 1(1):011001, jul 2021.

[140] R. Khaddam-Aljameh A. Sebastian, M. Le Gallo and E. Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7):529–544, 2020.

[141] O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun. Processing data where it makes sense: Enabling in-memory computation. *Microprocessors and Microsystems*, 67:28–41, 2019.

[142] J. Grollier, D. Querlioz, K. Y. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles. Neuromorphic spintronics. *Nature Electronics*, 3(7):360–370, 2020.

[143] X. Wang, Y. Chen, H. Xi, H. Li, and D. Dimitrov. Spintronic memristor through spin-torque-induced magnetization motion. *IEEE Electron Device Letters*, 30(3):294–297, 2009.

[144] A. Yamaguchi, T. Ono, S. Nasu, K. Miyake, K. Mibu, and T. Shinjo. Real-space observation of current-driven domain wall motion in submicron magnetic wires. *Physical Review Letters*, 92(7):077205, 2004.

[145] T. Leonard, S. Liu, H. Jin, and J. A. C. Incorvia. Stochastic domain wall-magnetic tunnel junction artificial neurons for noise-resilient spiking neural networks. *Applied Physics Letters*, 122(26):262406, 2023.

[146] Y. Huang, W. Kang, X. Zhang, Y. Zhou, and W. Zhao. Magnetic skyrmion-based synaptic devices. *Nanotechnology*, 28(8):08LT02, 2017.

[147] X. Chen, W. Kang, D. Zhu, X. Zhang, N. Lei, Y. Zhang, Y. Zhou, and W. Zhao. A compact skyrmionic leaky–integrate–fire spiking neuron device. *Nanoscale*, 10(13):6139–6146, 2018.

[148] A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollier, and D. Querlioz. Neural-like computing with populations of superparamagnetic basis functions. *Nature Communications*, 9(1):1533, 2018.

[149] P. Wadley, B. Howells, J. Železný, C. Andrews, V. Hills, R. P. Campion, V. Novák, K. Olejník, F. Maccherozzi, S. S. Dhesi, S. Y. Martin, T. Wagner, J. Wunderlich, F. Freimuth, Y. Mokrousov, J. Kuneš, J. S. Chauhan, M. J. Grzybowski, A. W. Rushforth, K. W. Edmonds, B. L. Gallagher, and T. Jungwirth. Electrical switching of an antiferromagnet. *Science*, 351(6273):587–590, 2016.

[150] M. J. Grzybowski, P. Wadley, K. W. Edmonds, R. Beardsley, V. Hills, R. P. Campion, B. L. Gallagher, J. S. Chauhan, V. Novak, T. Jungwirth, F. Maccherozzi, and S. S. Dhesi. Imaging current-induced switching of antiferromagnetic domains in cumnas. *Physical Review Letters*, 118(5):057701, 2017.

[151] A. Drews M. Bolte-G. Meier S. Bohlens, B. Krüger and D. Pfannkuche. Current controlled random-access memory based on magnetic vortex handedness. *Applied Physics Letters*, 93(14):142508, 2008.