UCLouvain

École polytechnique de Louvain

Hardware neural networks for energy efficient machine learning tasks

Author: **Anatole Moureaux**
Supervisors: **Dr. Flavio Abreu Araujo, Pr. Luc Piraux**
Readers: **Pr. Denis Flandre, Pr. Laurent Jacques, Pr. Gian-Marco Rignanese**
Academic year 2021–2022
Master [120] in Chemical and Materials Engineering

Abstract

Reservoir computing is a machine learning model based on the use of a recurrent, black-box, artificial neural network called a *reservoir*, where the input is processed by being projected into a higher dimension space. This setup can be used to perform interesting and concrete tasks, such as speech and pattern recognition. Usually, the neurons constituting the reservoir are modeled as a software using classic CMOS units. Spintronics, which describes the spin-dependent behavior of conduction electrons in metallic materials and the influence of magnetic fields on the latter, can be used to implement a hardware and eco-energetic version of the artificial neurons (and hence, the reservoir) needed for the solving of a problem in the framework of reservoir computing. This also allows to avoid the limitations inherent to the von Neumann architecture, on which the classical software neural networks are usually implemented. To do so, the non-linear dynamics of the magnetization of a specific type of magnetic nanostructures is used. These nanostructures are called *spin-torque vortex nano-oscillators* (STVOs). In this work, the dynamics of a STVO is modeled using the Thiele equation approach (TEA), which prevents the use of expensive and time-consuming micromagnetic simulations. Different original models (numerical and analytical) are then used to do so, leading to the simulation of an entire neural network. These models are tested using a custom-made benchmarking pattern recognition task in order to assess their characteristics. Parametric studies are also performed to test the impact of different parameters on the efficiency of the simulated neural network.

Acknowlegdements

Throughout this academic year, I felt blessed to be allowed to work on this thesis. Since last year when I contacted Flavio Abreu Araujo for the first time, until the end of the redaction of this work, I've never lost the passion and interest about the topic I was working on. All the advancements and discoveries made during the year have punctuated my daily life, so that it was always a pleasure to come to the lab to work on the thesis.

I would like to deeply thank Dr. Flavio Abreu Araujo for the unlimited help and dedication he has shown since last September. From the explanations of the different concepts and models, to the establishment of an optimal and stimulating working environment in the lab, but also the numerous advices he gave me outside the working hours. I couldn't quantify what I owe him at this stage.

I also would like to thank Simon de Wergifosse and Chloé Chopin, who actively participated in the good working atmosphere in the lab, for the interesting discussions we had together, and for pushing me to consider a future in research. I also would like to thank Pr. Luc Piraux and Jimmy Weber for generating my interest in the wonderful world of spintronics and in the object of this thesis.

I'm deeply grateful to my family for pushing me since my earliest years to follow my dreams and ambitions in STEM, and for their unlimited support and motivation. I wouldn't certainly be here without them.

I would like to thank my roommates, for being by my side for two years now and my friends, who I was able to relate to all along my studies. They surely helped making my journey at EPL much more enjoyable than what I could ever hope for.

I would like to thank Arnaud for his unconditional love, support and patience.

Contents

List of Figures

List of Tables

List of acronyms and variables

Acronyms

AWGN	Additive White Gaussian Noise
CPU	Central Processing Unit
DOS	Density Of States
FM	FerroMagnetic
MMS	MicroMagnetic Simulation
MTJ	Magnetic Tunnel Junction
NM	Non-Magnetic
NN	Neural Network
RC	Reservoir Computing
RNN	Recurrent Neural Network
SNR	Signal-to-Noise Ratio
STNO	Spin Torque Nano-Oscillator
STT	Spin Transfer Torque
STVO	Spin Torque Vortex (Nano-)Oscillator
TEA	Thiele Equation Approach
TMR	Tunneling MagnetoResistance
TW	Tau-Wise
WTA	Winner-Takes-All

Variables

A	Exchange stiffness coefficient
a_J	Spin-transfer efficiency
b_i	Bias of neuron i
C	Chirality of a magnetic vortex
D	Damping constant
E	Energy
E_{F}	Fermi energy
e	Charge of the electron
$\mathbf{F}$	Acoustic filter
$\mathbf{F}^{\mathrm{ST}}$	Forces related to the perpendicular STT effect
f	Activation function (of a neuron)
f_{NL}	Non-linear function implemented by a STVO
G	Gyrotropic constant
g	g-factor
$\mathbf{H}^{\mathrm{eff}}$	Effective (external) magnetic field
$\mathbf{H}^{\mathrm{Oe}}$	Ampère-Oersted field
h	Thickness of the free FM layer
$\hbar$	Planck's constant
J	Current density
J_{cr1}	First critical current density
J_{cr2}	Second critical current density
k	Spring-like restoring force constant
k^{ex}	Exchange spring-like restoring force constant

k^{ms}	Magnetostatic spring-like restoring force constant
k^{Oe}	Ampère-Oersted field spring-like restoring force constant
l_{ex}	Exchange length
$\mathbf{M}$	Magnetization of the fixed FM layer OR mask
$\hat{\mathbf{M}}$	Unit vector in the direction of $\mathbf{M}$
M_{s}	Saturation magnetization (of the free layer)
$M_{\mathrm{s}}^{\mathrm{ref}}$	Saturation magnetization of the fixed FM layer
$\mathbf{m}$	Magnetization of the free FM layer
m_{e}	Mass of the electron
N_f	Number of frequency channels
N_θ	Number of virtual neurons
N_τ	Number of time steps in a audio file
P	Polarity of a magnetic vortex
P_{cond}	Probability of conduction of an electron through a tunnel barrier
p_J	Polarization of the spin-polarized current
R	Radius of the free FM layer
s	In-plane reduced position of the vortex core
$\mathbf{T}_\sigma$	(Target) classification matrix of a data sample/audio file σ
$\hat{\mathbf{T}}_\sigma$	Estimated classification matrix of a data sample/audio file σ
$\mathbf{t}_\sigma$	(Target) classification vector of a data sample/audio file σ (= 1 column from $\mathbf{T}_\sigma$)
$\hat{\mathbf{t}}_{\sigma,\mathrm{av}}$	τ-averaged estimated classification vector of a data sample/audio file σ
$\hat{\mathbf{t}}_{\sigma,\tau}$	Estimated classification vector of a time step τ in data sample/audio file σ
$\mathbf{V}_\sigma$	Output matrix of the reservoir for input σ
$\mathbf{v}_\sigma$	Flattened output matrix of the reservoir for input audio file σ

$\mathbf{W}$	Weights matrix
W	Potential magnetic energy
W^{ex}	Exchange potential magnetic energy
W^{ms}	Magnetostatic potential magnetic energy
W^{Oe}	Ampère-Oersted field potential magnetic energy
$w_{i,j}$	weight of the synapse between neurons i and j
$\mathbf{X}$	In-plane position of the vortex core
$\dot{\mathbf{X}}$	In-plane velocity of the vortex core
$\mathbf{X}_\sigma$	Audio file σ
$\mathbf{X}'_\sigma$	Acoustically filtered audio file σ
$\mathbf{X}''_\sigma$	Masked and acoustically filtered audio file σ
x	Number of correctly recognized time steps from the same audio file
$\hat{\mathbf{T}}_\sigma$	Estimated classification matrix of a data sample σ
x_j	Input from neuron j
$\mathbf{x}''_\sigma$	Flattened, masked and acoustically filtered audio file σ
y	Expected output of a NN (= target)
$\hat{y}$	Actual output of a NN (= estimated target)
$y_{i,\mathrm{out}}$	Output of neuron i
α	Gilbert damping constant
Γ	Driving parameter
γ	Gyromagnetic ratio
γ_{G}	Gilbert gyromagnetic ratio
$\eta(\theta)$	Slonczewski STT parametric function
θ	Angle between $\mathbf{M}$ and $\mathbf{m}$

μ_{B} Bohr magneton

σ Index representing a sample of data to classify

ω Angular frequency of the gyrotropic motion of the vortex core

Introduction

The present work consists in the presentation and the assessment of the performances of different original models developed to describe the non-linear dynamics of the magnetization of a specific type of magnetic nanostructures called spin torque vortex nano-oscillators (STVOs). These nanoscale oscillators can mimic the behavior of biological neurons. Hence, they can be used to implement hardware neuronal units and hardware neural networks. Eventually, one can use them to perform eco-energetic machine learning tasks in the framework of reservoir computing, such as pattern and speech recognition.

In order to design efficient hardware neural networks based on STVOs, the complex dynamics of their magnetization must be accurately and efficiently simulated. Micromagnetic simulations can be used to do so, but these are extremely time and energy consuming. On the other hand, the phenomenological model that was used until now to describe the behavior of STVOs was proven to be inaccurate when compared to experimental data. More efficient and accurate state-of-the-art models, numerical or analytical, have been developed from the Thiele equation approach. To answer to the different open questions raised by the results of the previously published works on the subject, the assessment of the ability of these unpublished models to simulate STVOs efficiently and accurately is of greater importance.

First, the basics of reservoir computing will be presented, followed by a presentation of the spintronics concepts that will be involved in this work. Spin torque vortex nano-oscillators will then be described. Afterwards, several state-of-the-art methods and examples of hardware artificial intelligence based on STVOs in the framework of reservoir computing will be presented, in order to demonstrate the diversified range of existing implementations.

Then, original models of the dynamics of STVOs in the scope of the Thiele equation approach will be presented. The mathematical foundations and the characteristics of these singularly different models will be explained, as well as their resulting strengths and weaknesses.

To assess the performances of the developed models, a benchmarking pattern recognition task was designed. It will be presented, from the database to the different recognition methods. Each of the previously presented models will then be used to simulate a STVO and emulate a hardware neural network. The link will be made between the different implementations of the neural network and the different models, allowing to compare their performances in an applicative context. The presentation of two parametric studies will also be made, in order to highlight the ability of the developed models to allow efficient and accurate simulations of STVO-based neural networks. The results will be analyzed and interpreted, thus allowing to draw conclusions on the ability of the models, and on the optimal range of operating conditions of the STVOs thanks to the parametric studies.

Chapter 1

Using magnetic nanostructures in the framework of reservoir computing

This chapter aims to describe how certain types of magnetic nanostructures can be used to implement neurons in order to perform machine learning tasks in the framework of reservoir computing. First, the basics of reservoir computing are explained, as well as the qualities required by a system to design neurons and reservoirs. The limitations of the current software implementations and the motivations to design hardware neural networks are also presented. Then, the different spintronics concepts and notions involved in these hardware implementations are described. The characteristics of two types of functional magnetic nanostructures of interest (STNOs and STVOs) are explained. Finally, the two fields previously presented are gathered, and different designs and implementations of hardware neural networks from the literature are exposed, showing the large variety of uses of STVOs in the field of machine learning.

1.1 Machine learning tasks in the framework of reservoir computing

1.1.1 Machine learning and neural networks

In the past few decades, artificial intelligence has gained a lot of interest from the scientific community due to the numerous applications in which it can be used. Nowadays, artificial intelligence is employed almost everywhere, from social networks to self-driving cars. Machine learning is an artificial intelligence technique that allows a computer to process data in a way that is far more valuable than with any other handwritten program. Among the different tasks that can be performed with this technology, data classification stands as one of the most common and typical proofs of its ability. The so-called intelligence lies in the fact that the technique allows to effectively classify data that was never seen before by the program, as long as the program has been properly trained with sufficient amounts of data [1].

Machine learning algorithms operate using an artificial structure called a neural network (Fig.1.1). This structure is composed of a set of neurons, linked together by synapses. The neurons are units able to process the information by transmitting to other neurons

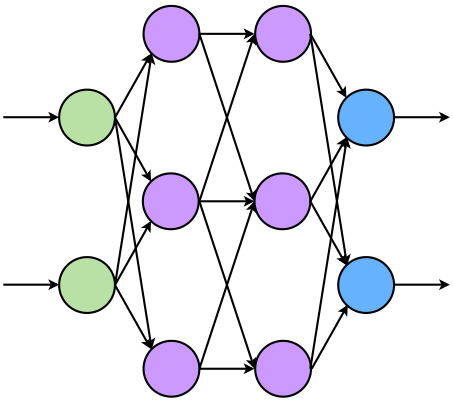

Fig. 1.1: Example of (feed-forward) neural network.

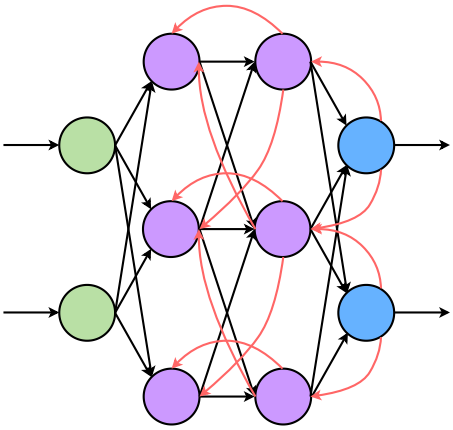

Fig. 1.2: Example of recurrent neural network. The recurrent connections are shown in red.

Fig. 1.3: Examples of classical neural network and recurrent neural network. The input layer is shown in green, the hidden layers are shown in purple and the readout (output) layer is shown in blue.

a linear combination of the different streams of input x_j they receive. Each stream of input x_j comes from a different adjacent neuron j, and is multiplied in the neuron i by a certain weight $w_{i,j}$, defined by the synapse located between neurons i and j (Eq.1.1). After having added a bias b_i to the linear combination in the neuron i, the result is fed to an activation function f (modeling the non-linear behavior of the neurons) to produce the output $y_{i,\text{out}}$ [2]. Several layers of neurons can be distinguished. The input layer (in green in Fig.1.1), is the first layer of neurons in the network and receives the input. The last layer of the network is the readout layer (in blue in Fig.1.1) and emits the output of the network. Between these two layers lies a certain number of additional neurons organized in several layers. As these neurons are often not accessible, these layers are called hidden layers. Basic artificial neural networks are qualified of feed-forward. Indeed, information is transmitted from the input layer to the readout layer in one direction.

$$y_{i,\text{out}} = f\left(\sum_j w_{i,j}x_j + b_i\right) \tag{1.1}$$

In the case of temporal problems, such as speech recognition, language translation or image labeling, recurrent neural networks (Fig.1.2) are used. They consist of neural networks where the output of the neurons from a certain layer is re-injected back into the network, allowing what could be considered as memory or feedback. This allows to process data while keeping a certain context : the data flow is no longer feed-forward and the different parts of the input are no longer independent in time, as they are linked by the recurrent connections. Recurrent connections can also be found between a neuron and the same neuron.

1.1.2 Limitations

Data classification is only possible thanks to the tuning of a lot of parameters : the weight of each synapse. The determination of the different weights requires a heavy procedure

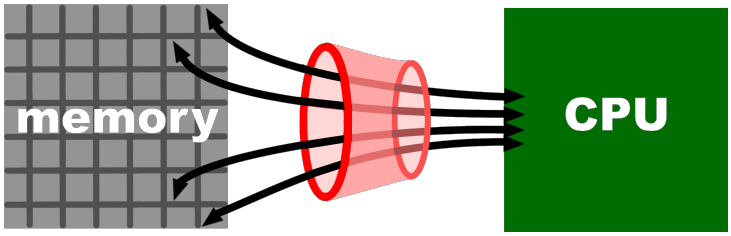

Fig. 1.4: Representation of the von Neumann bottleneck (in red) between the memory and the central processing unit (CPU)

consisting in the minimization of a loss function representing the distance between the expected output y and the actual output $\hat{y}$ of the NN for m training samples of data, with respect to all the weights $w_{i,j}$ of the network (Eq.1.2).

$$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{1.2}$$

As the neurons are modeled using software functions in most implementations, this minimization procedure itself requires a lot of labeled data, and a lot of computational power. This corresponds to the training phase of the NN. Moreover, classical RNN don't always allow to find the best minimization of the loss function : often, a local minimum is found, not a global one. As the demand for artificial intelligence keeps growing all around the world, those training procedures become somewhat problematic in terms of energy and time consumption. Indeed, the training of a NN involves transiting data back and forth between the memory unit of the computer (where the value of all the weights are stored), and the central processing unit (CPU), where the computation of the minimization of the loss function takes place. As the amount of data to be processed by the neural networks of nowadays also becomes more and more important, the performances of NN get close to an upper limit defined by the von Neumann bottleneck : a limitation of the throughput of data between the memory and the CPU due to the architecture of standard computers (represented in red in Fig.1.4). Hence, the time and the energy needed to train neural networks are expected to increase as the amount of training data increases as well [3].

1.1.3 Reservoir computing

Reservoir computing overcomes those limitations by reducing the amount of training needed by the NN. The so called reservoir is composed of a classical recurrent neural network of neurons showing a non-linear behavior, but the connections between the neurons are random and fixed (see Fig.1.5). That means that the weights corresponding to the synapses inside the reservoir are not updated during the training phase. In fact, only the weights of the readout layer have to be trained, and this training is trivial and can be carried out linearly. Furthermore, there is no need to know the weight of the connections or any information about the internal state of the reservoir to perform data classification ; the reservoir can be considered as a black box. However, reservoir computing still allows state-of-the-art data classification.

The reservoir thus allows to treat the input data non-linearly, thanks to the non-linear behavior of the neurons. The high number of neurons allows to project the input into

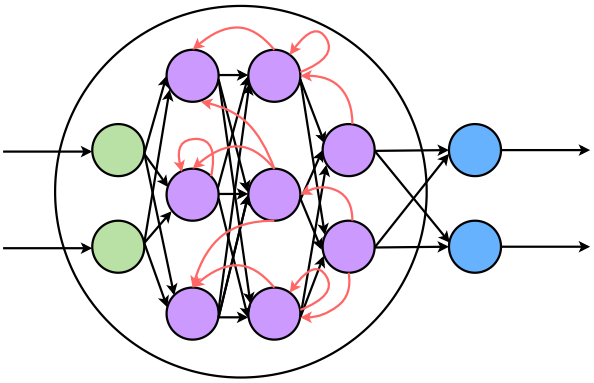

Fig. 1.5: Representation of a neural reservoir. The recurrent connections are shown in gray. For a sample σ fed into the input layer (in green), the output of the readout layer (in blue) corresponds to the matrix $\mathbf{V}_\sigma$.

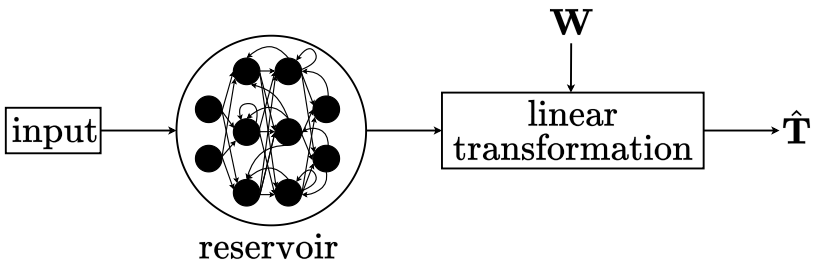

Fig. 1.6: Data classification process using reservoir computing

a space of higher dimension, leading to an efficient classification of the latter. Only one linear transformation is needed to obtain the results of the classification : the estimation of the classification target $\hat{\mathbf{T}}_\sigma$ of a data sample σ is obtained by multiplying the output of the reservoir $\mathbf{V}_\sigma$ by the weights of the reservoir $\mathbf{W}$ (see Eq.1.3 and Fig.1.6). To train the RNN to classify data into several categories, several samples σ of labeled training data (which means that the exact classification matrices $\mathbf{T}_\sigma$, or targets, are known) are fed to the reservoir. Then, the corresponding outputs $\mathbf{V}_\sigma$ of the reservoir are recorded. The weights matrix $\mathbf{W}$ can then be obtained by linear regression using a simple matrix inversion (see Eq.1.4 and Fig.1.7) [4]. In some case-specific applications like speech recognition, additional non-linearity can be added in the treatment process of the input data before it enters the reservoir, as it will be seen later on.

$$\hat{\mathbf{T}}_\sigma = \mathbf{W}\mathbf{V}_\sigma \tag{1.3}$$

$$\mathbf{W} = \mathbf{T}_\sigma \mathbf{V}_\sigma^{-1} \tag{1.4}$$

The easy training of the reservoir is the most advantageous feature of reservoir computing, but it is not the only one. Reservoir computing is indeed well suited for hardware implementation. Hardware implementations allow to intertwine the memory and the processing of the information as in the human brain, which then allows to circumvent the von Neumann bottleneck problem [3, 5, 6, 7, 8].

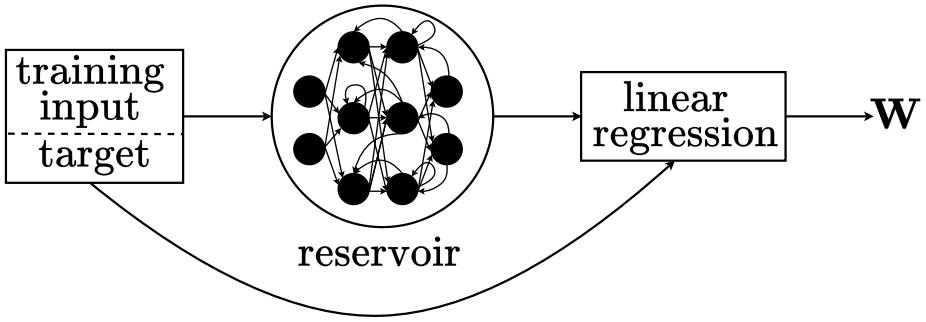

Fig. 1.7: Training process of the reservoir

1.2 Spin torque nano-oscillators and spin torque vortex nano-oscillators

1.2.1 Spintronics

Spintronics, also called magnetoelectronics, is the branch of physics that describes the spin-dependent transport of conduction electrons in magnetic structures. Since its discovery by Albert Fert et al. in 1988 [9], a lot of related applications have been developed, allowing considerable technological breakthroughs.

In ferromagnetic materials (such as $3d$ transition metals like iron, cobalt and nickel), electrons are separated into two distinct populations : electrons whose spin is parallel to the direction of the magnetization of the material (spin up electrons), and electrons whose spin is anti-parallel to the direction of the magnetization (spin down electrons). This naturally also holds for conduction electrons, belonging to the $3d$ and $4s$ bands.

Furthermore, the density of states of $3d$ transition metals is not symmetrical for the two populations of conduction electrons, as it can be seen in Fig.1.8. Indeed, a shift in energy can be seen between the two sub-bands $3d_\uparrow$ and $3d_\downarrow$, the bands corresponding to the two spin populations of the band $3d$. As the states are filled until $E = E_\mathrm{F}$, more states are filled in the $3d_\uparrow$ sub-band than in the $3d_\downarrow$. Hence, spin up electrons are called *majority spin electrons*, while spin down electrons are called *minority spin electrons*. Finally, it can also be seen in Fig.1.8 that the densities of states of the $3d_\uparrow$ and $3d_\downarrow$ are not the identical at the Fermi level ; it is higher for minority spin electrons than for majority spin electrons [10].

1.2.2 Tunnel magnetoresistance

Tunnel magnetoresistance allows to tune the electrical resistance of a magnetic tunnel junction by modifying its magnetic configuration. It takes place in magnetic tunnel junctions, nanostructures made of a stack of ferromagnetic materials spearated by a layer of non-magnetic insulating material such as Al_2O_3 or MgO (Fig.1.9).

The probability for an electron to be conducted through the tunnel barrier made of the non-magnetic spacer is proportional to the density of states at E_F before the tunnel barrier, and the density of states at E_F after the tunnel barrier (Eq.1.5).

$$P_\mathrm{conduction} \propto \mathrm{DOS}_\mathrm{left}(E_\mathrm{F}) \times \mathrm{DOS}_\mathrm{right}(E_\mathrm{F}) \tag{1.5}$$

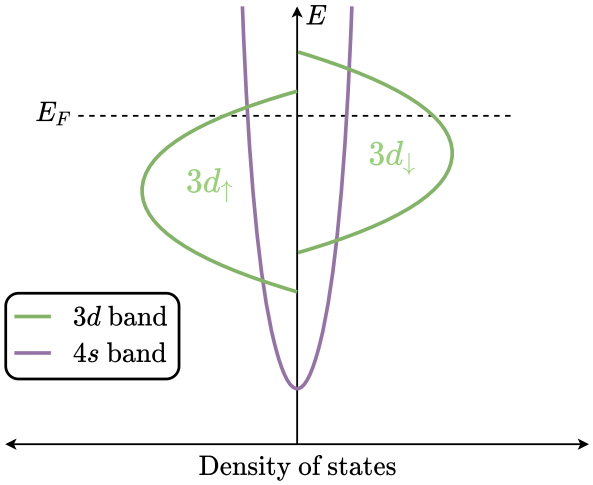

Fig. 1.8: Spin-dependent density of states of $3d$ electrons in a ferromagnetic material

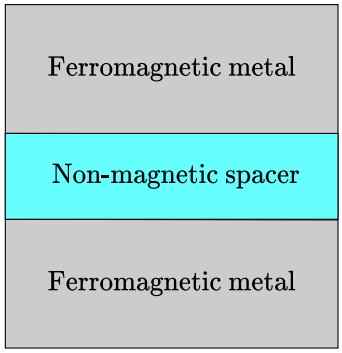

Fig. 1.9: Magnetic tunnel junction

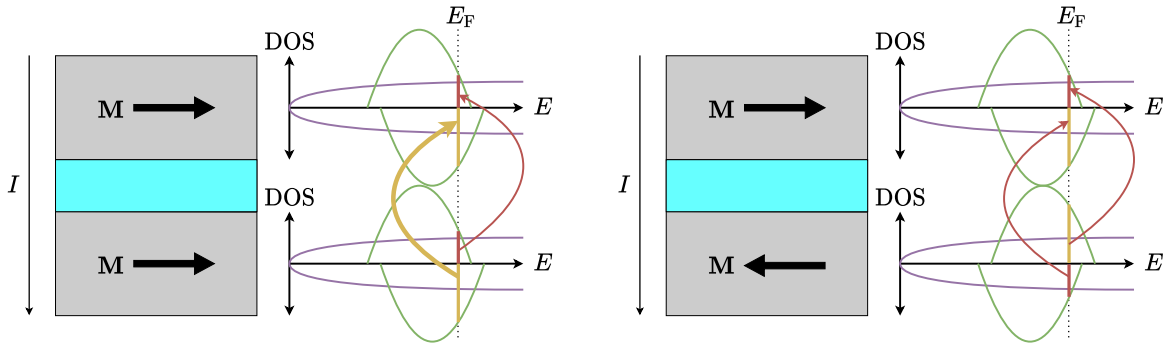

Fig. 1.10: Tunneling magnetoresistance. **Left** : Parallel configuration. The probability of electron conduction is high due to the high density of states for minority electrons on the two sides of the barrier (yellow arrow), leading to a low resistance state. **Right** : Antiparallel configuration. The probability of electron conduction is lower than in the parallel configuration due to the inversion of the densities of states across the barrier, leading to a state of higher resistance.

By tuning the direction of the magnetization of the layers across the tunnel barrier independently, it is possible to tune the corresponding densities of states and vary the overall probability of electron conduction, hence influencing the resistance (Fig.1.10) [11].

1.2.3 Spin transfer torque

Spin transfer torque is a phenomenon occurring in multilayered magnetic nanostructures such as the one represented in Fig.1.9. However, it is supposed here that the first ferromagnetic layer (where the electrons are injected) is much thicker than the second one. As a result, the magnetization of the first FM layer is fixed, while the magnetization of the second layer is free to be redirected into another direction. These layers are thus respectively called the *fixed FM layer* and the *free FM layer*. Initially, it can be considered that the entire nanostructure is submitted to the same external magnetic field $\mathbf{H}_{\text{ext}}$, and that the magnetizations of the two FM layers are slightly misaligned. When a high current density is injected into this kind of nanostructures, two phenomena occur successively.

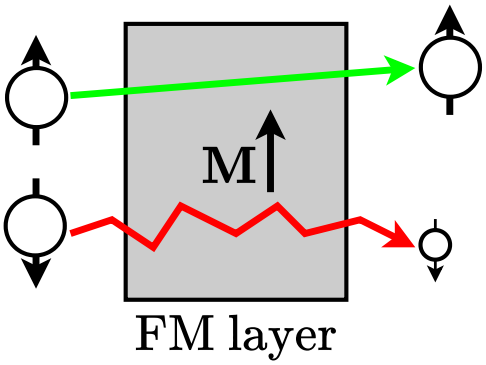

Fig. 1.11: Spin-filter effect. The electrons whose spin is aligned with the magnetization **M** of the FM material (spin up electrons) are more easily transmitted than spin down electrons.

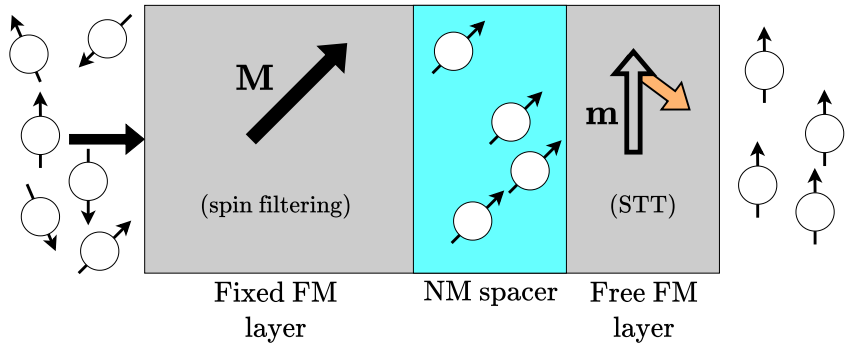

Fig. 1.12: Spin transfer torque in a FM/NM/FM nanostructure

First, the fixed FM layer will spin-polarize the electron flow. Indeed, as it can be seen in Fig.1.8, the density of states at the Fermi level is higher for minority spin electrons than for majority spin electrons. Hence, the probability of scattering is higher for minority spin electrons, as the electrons have more states to scatter to, than for majority spin electrons. As a result, the majority spin electrons (i.e electrons whose spin is directed in the same direction as the magnetization **M**) will be preferentially transmitted. This is called the *spin filtering effect* (see Fig.1.11 and Fig.1.12) [10, 12].

Next, as the spin-polarized electrons hit the free FM layer, a spin transfer torque effect will be observed. This phenomenon consists in the transfer of angular momentum from the spin of the spin-polarized electrons to the magnetization of a ferromagnetic material. This is due to the spin-polarized electron flow being spin-filtered by the free FM layer, whose magnetization is directed in another direction than the spin of the incoming electrons. As the spin of the transmitted electrons is directed into the same direction as the magnetization, the free FM layer absorbs a part of the spin angular momentum of the electrons, as if it had exerted a torque on the spin of the electrons. Due to the conservation of angular momentum during the process, the spin-polarized current exerts a torque of the same amplitude but in the opposite direction, on the magnetization of the free FM layer (see Fig.1.12) [13].

The dynamics of the magnetization **m** of the free layer, which corresponds to the sum of all the magnetic moments per unit volume, is described by the Landau-Lifshitz-Gilbert-Slonczewski equation (Eq.1.6) when it is subjected to an external magnetic field $\mathbf{H}^{\text{eff}}$ and

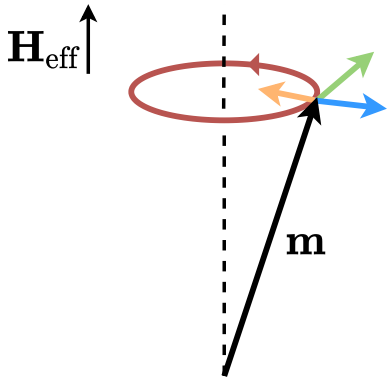

Fig. 1.13: The different contributions to the dynamics of a magnetic moment $\mathbf{m}$ submitted to an external magnetic field $\mathbf{H}^{\text{eff}}$ and a spin-polarized current. The first term of Eq.1.6 is represented in green, and the Larmor precession in red. The Gilbert damping term is represented in orange and the contribution of the Slonczewski STT and field-like torque term is represented in blue.

a spin-polarized current [10, 14].

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = -\gamma\mathbf{m}\times\mathbf{H}^{\text{eff}} + \frac{\alpha}{M_{\text{s}}}\mathbf{m}\times\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} - \frac{v}{M_{\text{s}}^2}\mathbf{m}\times(\mathbf{m}\times(\mathbf{J}\cdot\nabla)\,\mathbf{m}) - \xi\frac{v}{M_{\text{s}}}\mathbf{m}\times(\mathbf{J}\cdot\nabla)\,\mathbf{m} \quad (1.6)$$

1. The first term $-\gamma\mathbf{m}\times\mathbf{H}^{\text{eff}}$ represents the precession that the magnetization $\mathbf{m}$ will spontaneously follow when submitted to $\mathbf{H}^{\text{eff}}$, with $\gamma = \dfrac{g|\mathrm{e}|}{2m_{\mathrm{e}}}$ the gyromagnetic ratio, and g the g-factor equal to -2 for free electrons. This is called the Larmor precession.

2. The second term $\dfrac{\alpha}{M_{\text{s}}}\mathbf{m}\times\dfrac{\mathrm{d}\mathbf{m}}{\mathrm{d}t}$ is a damping term called the Gilbert term, which tends to align the magnetic moment $\mathbf{m}$ with $\mathbf{H}^{\text{eff}}$. Here, α is the Gilbert damping constant, and M_s is the saturation magnetization of the free FM layer (the norm of $\mathbf{m}$).

3. The third term $\dfrac{v}{M_{\text{s}}^2}\mathbf{m}\times(\mathbf{m}\times(\mathbf{J}\cdot\nabla)\,\mathbf{m})$ is called the Slonczewski STT term and corresponds to the contribution of the spin transfer torque, with $v = \dfrac{P\mu_{\text{B}}}{\mathrm{e}M_{\text{s}}\,(1+\xi^2)}$ the coupling constant between the spin polarized current $\mathbf{J}$ and the magnetization of the free layer $\mathbf{m}$ (P being the polarization of the current $\mathbf{J}$, μ_{B} the Bohr magneton, M_{s} the saturation magnetization of the free FM layer and ξ a degree of non-adiabaticity). The injection of spin-polarized electrons into the free layer will counteract the Gilbert damping term and sustain the Larmor precession.

4. The last term $\xi\dfrac{v}{M_{\text{s}}}\mathbf{m}\times(\mathbf{J}\cdot\nabla)\,\mathbf{m}$ is the field-like torque term, and corresponds to an additional STT term that is specific to magnetic tunnel junctions [14].

The different components of the dynamics of $\mathbf{m}$ are represented in Fig.1.13.

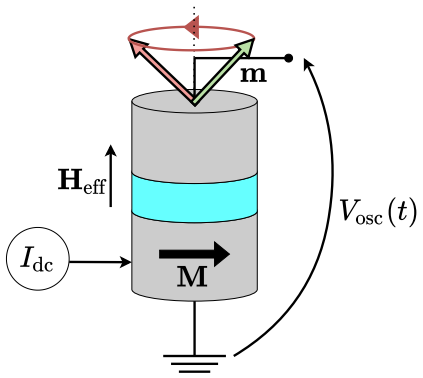

Fig. 1.14: Representation of a STNO. The green arrow corresponds to the MTJ being in the low resistance state, while the red arrow corresponds to the MTJ being in the high resistance state.

1.2.4 Spin torque nano-oscillators

Spin torque nano-oscillators are functional devices relying on both magnetoresistance and spin transfer torque phenomena to emit an oscillating signal in the microwave range. STNOs consist of multilayered magnetic nanostructures with an uniform magnetization, such as the one depicted in Fig.1.12. Usually, it is built as a nanopillar, whose diameter can range from 10 to 50 nm, and height of about 50 nm [15]. These ranges of dimensions have been motivated by the high current densities (of the order of MA/cm^2) needed for the STT effect to be noticeable over noise. Thanks to the developments of the past few decades in micro- and nano-fabrication and characterization techniques, smaller and smaller structures have been made accessible, considerably lowering the required current intensities.

By injecting a proper (dc) current intensity into a STNO, it is possible to sustain a stable precession of the uniform magnetization of the free FM layer thanks to the STT effect counteracting the Gilbert damping. Hence, due to the presence of the fixed FM layer, the magnetic configuration of the MTJ can steadily oscillate between parallel and antiparallel states. By using the TMR effect presented in subsection 1.2.2, the overall resistance of the MTJ will oscillate between states of high, low, and intermediate electrical resistance (Fig.1.14). As the injected current is dc, the voltage across the MTJ will oscillate with frequencies located in the microwave range.

To fabricate the fixed FM layer, materials with a strong and hardly redirectable magnetic moment are wanted, such as CoFeB [15]. The fixing of the magnetization can also be improved by increasing the thickness of the layer (it is usually 1 to 20 nm thick [15, 16]), or by coupling the fixed layer to an antiferromagnetic layer, which makes it difficult to excite spin torque on the magnetization when it spin-filters the injected current. For the free FM layer on the other hand, a softer FM material such as FeB will be chosen [15, 17]. To facilitate the STT-induced dynamics of the free magnetization, the thickness of this layer is often smaller than the thickness of the fixed FM layer (usually between 3 and 6 nm [13, 15]. Finally, the NM layer, which plays the role of tunnel barrier in the MTJ, is often made of cristalline MgO, due to the high TMR ratio it allows to obtain. The thickness is usually of the order of 1 nm [11, 15, 18].

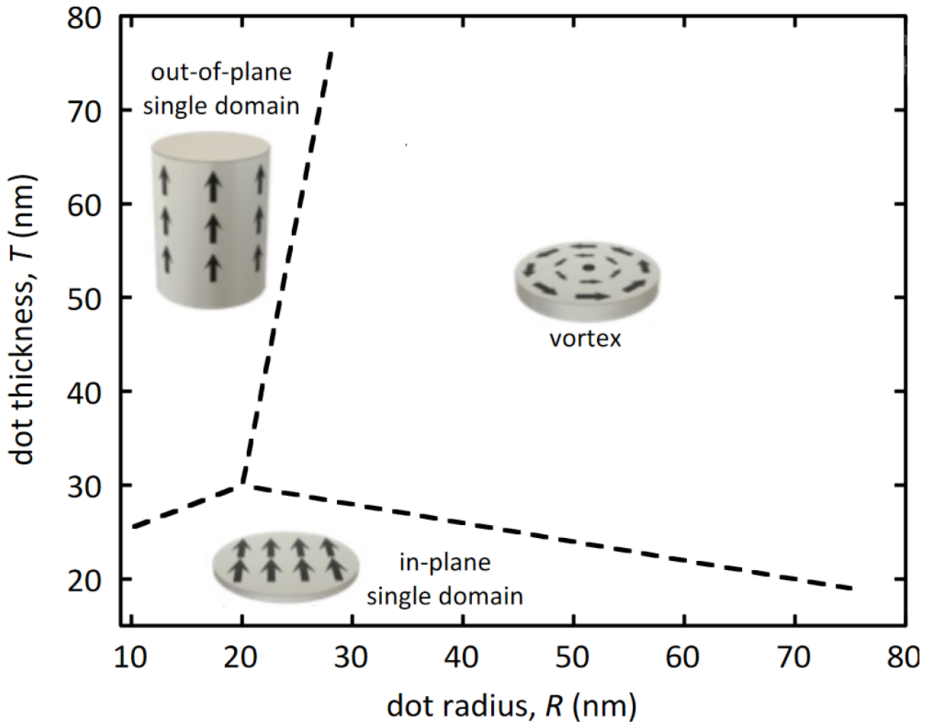

Fig. 1.15: Schematic phase diagram of the magnetic ground state of the free FM layer, depending on its aspect ratio. The dashed lines separating the three different states were obtained based on theoretical and experimental results. Version from [21], adapted from [19]. This phase diagram is universal for all soft FM materials if the axes are normalized to the exchange length l_{ex} of the material [22].

1.2.5 Spin torque vortex nano-oscillators

When the aspect ratio $\xi = h/2R$ (with h the thickness of the free FM layer and R its radius) lies into a certain range, the ground state of the magnetization turns into a vortex (Fig.1.16), and the STNO hence becomes a spin torque vortex nano-oscillator (STVO). A magnetic vortex is a magnetic ground state where the magnetization is not uniform, originating from the interplay between the magneto-static energy and the exchange energy. One part of the magnetization is curling in the plane defined by the free layer. The other part of the free magnetization lies at the center of the vortex, where it either points in or out of the surface of the free layer, due to the minimization of the high energetic configuration generated by the adjacent anti-aligned magnetic moments [14].

If the aspect ratio ξ of the free FM layer is too high, the shape anisotropy will tend to align the magnetization along the axis of the nanopillar. If it is too low, the same shape anisotropy will tend to confine the magnetization into the plane defined by the free layer. The presence of a magnetic vortex thus require the aspect ratio ξ to lie in a specific phase (see Fig.1.15) [19].

A magnetic vortex is defined by its chirality C (± 1), representing the curling direction of the magnetization, and its polarity P (± 1), representing the direction in which the vortex core is pointing (see Fig.1.16) [20].

As with STNOs, when a magnetic vortex is submitted to a spin-polarized current, the STT interaction induces gyrotropic oscillations of the vortex core on the surface of the free layer (Fig.1.17). The radius of this precessional motion and the frequency

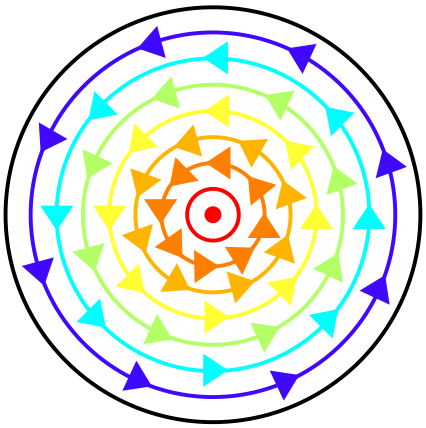

Fig. 1.16: Representation of a magnetic vortex ($C = +1$, $P = +1$)

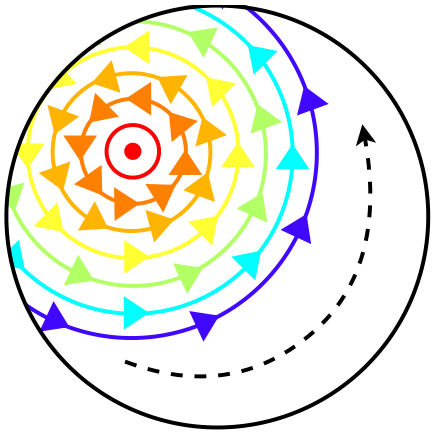

Fig. 1.17: Gyrotropic motion of the vortex core in a STVO submitted to spin polarized current

of the oscillations are non-linearly dependent on the density of spin-polarized current that is injected in the oscillator. This gyrotropic movement of the vortex core also has an influence on the electrical resistance of the structure through the TMR effect : the differential resistance $\mathrm{d}V/\mathrm{d}I$ is indeed expected to decrease as the vortex core approaches the nanopillar boundaries. When no spin-polarized current is injected into the oscillator, the vortex core goes relaxes to the center of the free layer, and the electrical resistance becomes maximal. However if the vortex core gets too close to the boundaries due to the STT effect, it can be expelled, hence restoring a state of quasi-uniform magnetization (c-state) in the free layer [23].

1.3 Reservoir compting using STVOs

The non-linear dynamics of the vortex core of STVOs can be used to implement a hardware neuron to be used in a reservoir (see subsection 1.1.3). Indeed, their extremely small size suggests a possible integration of a high number of neurons on the same chip, with the idea to make concrete eco-energetic artificial intelligence. Furthermore, the intrinsic relaxation time of the physical phenomena ruling the gyrotropic motion of the core can be taken advantage of to implement fading memory, which is a highly desirable feature

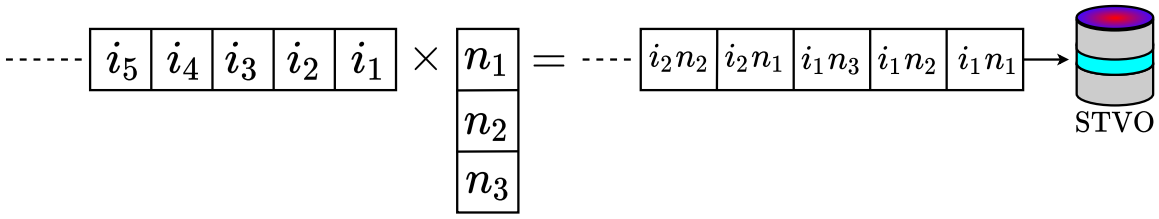

Fig. 1.18: Time multiplexing for single node RC. The input stream is multiplied by a mask of 3 values, allowing to emulate a reservoir of 3 neurons connected through space.

when performing temporal machine learning tasks.

1.3.1 Single node reservoir computing

Single node reservoir computing is a technique consisting in time-multiplexing the input data in order to emulate a network of N recurrently interconnected neurons using only one oscillator. This allows to study the performances of the resulting virtual reservoir without undergo the high complexity of an actual neural network of N oscillators [5, 6, 4, 15, 18]. To do so, the input signal that has to be treated by the reservoir is multiplied by a vector of N fixed and random values called a mask. Each data sample from the input signal is thus mapped onto a vector of N distinct values that is injected in the neuron (Fig.1.18). By ensuring that the time between the injection of two masked values of the same data sample into the oscillator is lower than the relaxation time of the oscillator, it is possible to connect the N neurons through time, and compensate the loss of spatial parallelism due to the presence of only one oscillator. The state of the oscillator in the future hence depends on the state of the oscillator in the past, the transient state of the oscillator being used as carrying medium for the data [15].

1.3.2 Pattern recognition

Pattern recognition consists in classifying pieces of input data into several distinct pattern categories. Despite looking fairly simple to perform with a standard algorithm, this task is actually non-linearly separable and constitutes an good benchmarking task for a hardware artificial intelligence. Several works have proven the ability of hardware neural networks to perform pattern recognition efficiently in the framework of reservoir computing [5, 8, 15].

In [18], a STVO was used as a neuron to classify waveforms between sine and square patterns. To implement the reservoir, a single STVO made of CoFeB, MgO and FeB was used and 25 neurons were emulated using time-multiplexing. The input signal was composed of randomly arranged sine and square periods, each of them sampled 8 times. The amplitude of the signals was encoded in a frequency-modulated signal injected into a metallic strip, generating a microwave magnetic field above the oscillator. The interaction between the "input" magnetic field and the vortex of the free layer was then recorded by analyzing the output signal of the oscillator. The output y of the neural network for each data sample is a weighted sum of the output of the i virtual neurons, with W_i and x_i the weight corresponding to the neuron i and its input, and f_{NL} the non-linear function implemented by the dynamics of the STVO (Eq.1.7) [18]. The weight vector $\mathbf{W}$ is linearly

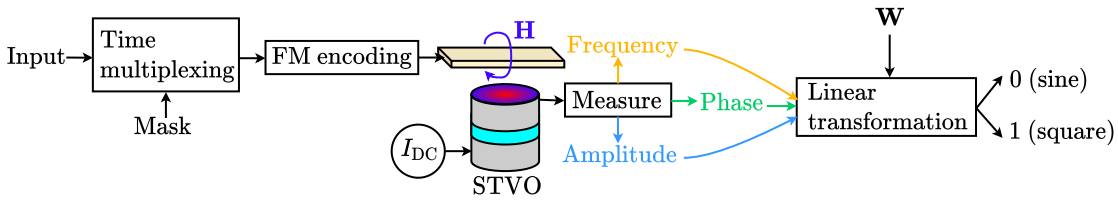

Fig. 1.19: Simplified schematic of a setup of hardware RC using the phase, the frequency or the amplitude of a STVO.

computed as explained in Eq.1.4.

$$y = \sum_{i=1}^{N=25} W_i f_{\mathrm{NL}}(x_i) \tag{1.7}$$

The function f_{NL} can either be the frequency, the phase or the amplitude of the output signal, as all those quantities vary non-linearly with the input frequency (see Fig.1.19).

As a result, state-of-the-art recognition rates were obtained (Fig.1.20). The frequency range in which the input signal was encoded has been shown to have a non-negligible influence on these results. Indeed, the oscillator tended to synchronize with the frequency input. In the locking range (in yellow in Fig.1.20), which is the frequency range where the oscillator and the input signal are perfectly synchronized, the noise is greatly reduced, which is desirable as the nanoscale oscillators tend to be very noisy. However f_{NL} becomes a linear function in the locking range, leading to a hardly achievable learning for the NN. It is hence optimal to center the input frequency range at the borders between the locking range and the frequency-pulling range (in green in Fig.1.20), to take advantage of the decrease of the noise (in the locking range), and the non-linear f_{NL} (in the frequency-pulling range).

This work highlights the fact that the phase, frequency and amplitude of the output signal of the oscillator can be used to perform the classification, as long as it is strongly non-linearly dependent on the input signal. Moreover, time multiplexing allows to efficiently emulate a large spatial NN using a single STVO.

1.3.3 Speech recognition

Speech recognition is a very common machine learning task nowadays due to the numerous applications based on this technique. Hence, demonstrating the ability of a STVO-based hardware neural network to perform such task is of great importance. Speech recognition of digits from 0 to 9 has been demonstrated in [5, 6, 8, 15] and the performances were assessed in [4].

Database

The database that was used comes from the TI-46 database, and is composed of 10 utterances of the 10 digits from 0 to 9, pronounced by 5 different female speakers (a total of 500 audio files). The goal here is to recognize the digit that is pronounced in each audio file for all the speakers. Half of the database is used to train the NN, while the other half is used to test its performances.

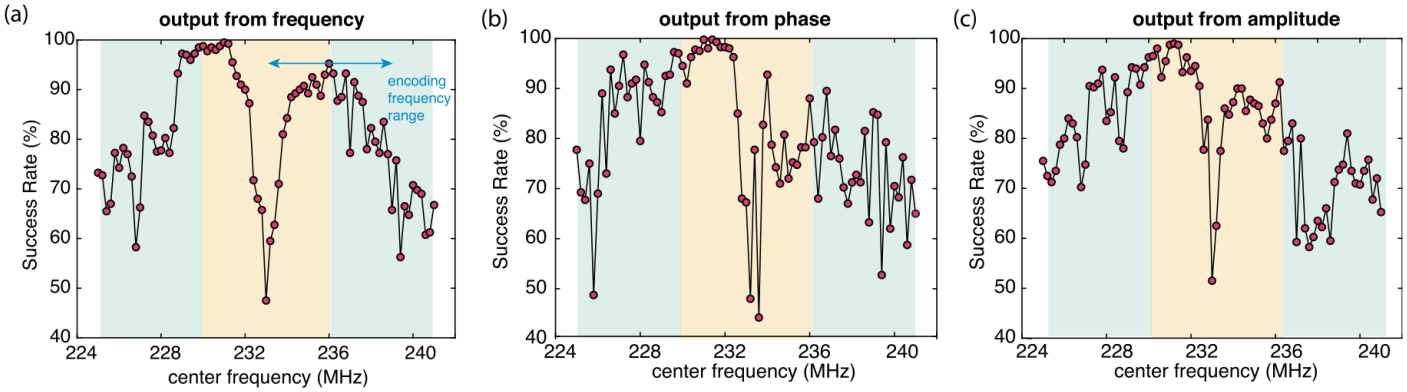

Fig. 1.20: Success rates obtained when decoding from the frequency, phase, and amplitude of the oscillator, as a function of the center of the frequency range chosen for encoding the input data. The frequency range used for encoding is indicated by a blue double arrow for two measurement points. Yellow and green shaded areas designate the locking range and the frequency pulling range, respectively. From [18].

Pre-treatment of the input

As the acoustic features of natural speech are mainly encoded in the frequency domain, each audio file $\mathbf{X}_\sigma$ is acoustically filtered into N_f frequency channels (Eq.1.8). This can be done using different acoustic filters $\mathbf{F}$ (Lyon's ear cochleagram, Mel-frequency cepstral coefficients (MFCC) filter, linear spectrogram and Spectro HP), whose main distinctive features are the number of produced frequency channels (between 13 for the MFCC filter and 78 for the cochleagram) and the level of non-linearity of the filter. Indeed, the feature extraction can be performed linearly (for example by using the linear spectrogram model) or strongly non-linearly, as it is the case for the cochleagram and the Spectro HP model. The higher the level of non-linearity of the filter is, the higher the speech recognition rate *without any NN* is (up to 95.8% for the cochleagram and 89% for the Spectro HP model). On the other hand, the linear spectrogram only achieves a score of 10%, which can be understood as a random choice, as there are 10 categories to choose between in this case (the 10 digits). This underlines the necessity of a certain level of non-linearity in the treatment of the data to obtain an efficient classification [4].

$$\mathbf{X}_\sigma \mapsto \mathbf{F}\mathbf{X}_\sigma = \mathbf{X}'_\sigma \tag{1.8}$$

Afterwards, the input is multiplied with a random binary mask $\mathbf{M}$ (filled only with 1s and -1s) in order to operate the time-multiplexing (Eq.1.9). Indeed, in this case a NN of 400 neurons is emulated using only one STVO (see subsection 1.3.1).

$$\mathbf{X}'_\sigma \mapsto \mathbf{M}\mathbf{X}'_\sigma = \mathbf{X}''_\sigma \tag{1.9}$$

Finally, $\mathbf{X}''_\sigma$ is flattened into a 1D vector, which is subsequently converted into an AM signal, which will be injected into the oscillator. The whole process is represented in Fig.1.21. $\mathbf{X}_\sigma$ and $\mathbf{X}'_\sigma$ are represented in Fig.1.22 and Fig.1.23.

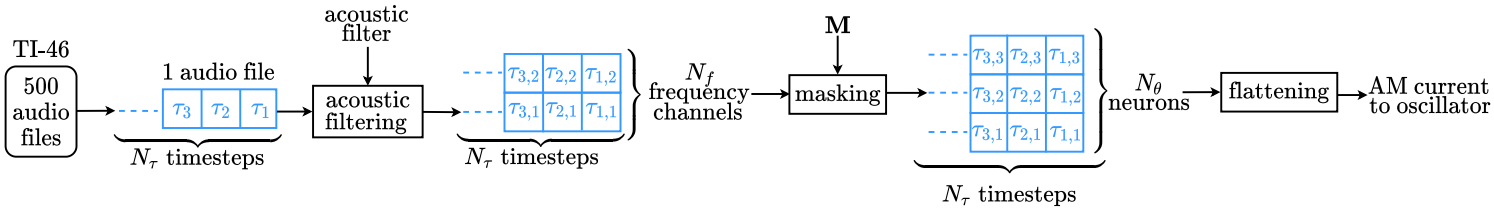

Fig. 1.21: Pre-treatment of the data from the database to the STVO. Each input audio file $\mathbf{X}_\sigma$ $(1 \times N_\tau)$ is composed of N_τ timesteps. The acoustic filter $\mathbf{F}$ $(N_f \times 1)$ converts $\mathbf{X}_\sigma$ into a $(N_f \times N_\tau)$ matrix $\mathbf{X}'_\sigma$, N_f being the number of frequency channels produced by $\mathbf{F}$. Then, the mask $\mathbf{M}$ $((N_\theta \times N_f)$ with N_θ the number of virtual neurons) is applied on the input to perform time multiplexing. Finally, the resulting $(N_\theta \times N_\tau)$ matrix $\mathbf{X}''_\sigma$ is flattened into a 1D vector $\mathbf{x}''_\sigma$ and is converted into an AM signal which will be injected into the STVO.

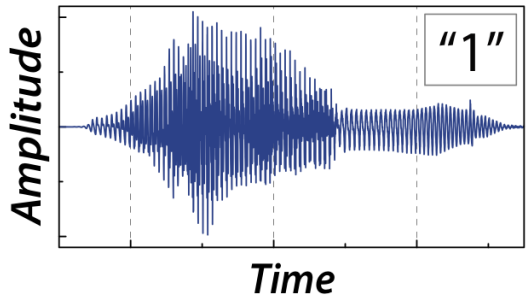

Fig. 1.22: Audio waveform corresponding to the digit 1 pronounced ($\mathbf{X}_\sigma$). From [4].

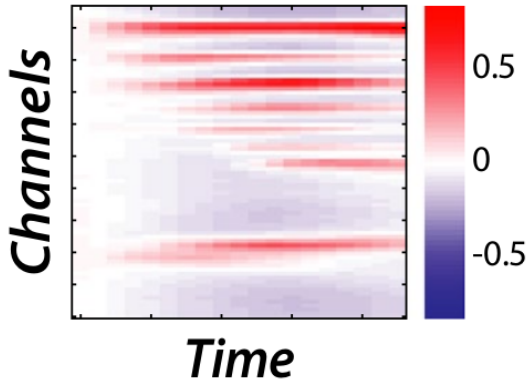

Fig. 1.23: Filtering to frequency channels for acoustic feature extraction. The signal during each time interval τ is decomposed into N_f frequency channels, resulting in the acoustically filtered input ($\mathbf{X}'_\sigma$). From [4].

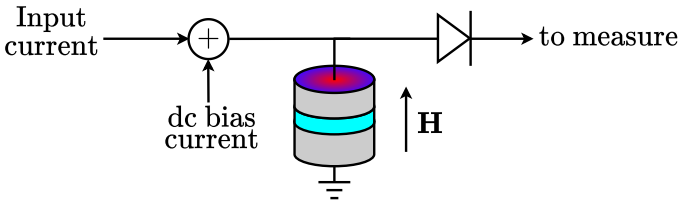

Fig. 1.24: STVO-treatment of the data

Reservoir

The reservoir is implemented by a single 375 nm diameter STVO. A DC bias current of 7 mA is injected into the STVO to trigger the dynamics of the vortex core. The MTJ is also submitted to an external magnetic field of 430 mT (see Methods in [15]). The AM signal generated by the input is added to the DC bias current in order to induce transient states in the dynamics of the vortex core. The amplitude of the output voltage $\mathbf{v}_\sigma$ is then recorded in real time using an oscilloscope and a diode (Fig.1.24). This step can be modeled with Eq.1.10, where the STVO() function represents the non-linear response of the STVO. The final output of the reservoir is $\mathbf{V}_\sigma$, which is obtained by reshaping $\mathbf{v}_\sigma$ into the same shape as the initial shape of $\mathbf{X}''_\sigma$, i.e $(N_\theta \times N_\tau)$ (Eq.1.11).

$$\mathbf{v}_\sigma = \text{STNO}(\mathbf{x}''_\sigma) \tag{1.10}$$

$$\mathbf{V}_\sigma = \text{reshape}(\mathbf{v}_\sigma) \tag{1.11}$$

Training phase

The weights matrix $\mathbf{W}$ is retrieved using Eq.1.4. As $\mathbf{V}_\sigma$ is rarely square and need to be inverted, the Moore-Penrose inversion is used [4]. Actually, the database is subdivided into 10 subsets of 50 audio files, with the same number of each digit. Then, n of those subsets are used for the training phase (i.e $50n$ audio files), and the $50(10-n)$ signals left are used for testing. The computation of $\mathbf{W}$ is done for all the training audio files at the same time (Eq.1.12). Here, the target matrices $\mathbf{T}_\sigma$ (of dimension $(d \times N_\tau)$, with $d = 10$ the number of categories into which the data has to be classified and N_τ the number of time steps into which the file is decomposed) are filled with 0s except on the line whose index corresponds to the digit that is pronounced in the audio file σ. For example, if the digit "5" was pronounced, the target matrix is composed of 0s everywhere except on the sixth line, which is filled with 1s.

$$\mathbf{W} = [\mathbf{T}_{\sigma_1} \quad \mathbf{T}_{\sigma_2} \quad ... \quad \mathbf{T}_{\sigma_{50n}}] \, [\mathbf{V}_{\sigma_1} \quad \mathbf{V}_{\sigma_2} \quad ... \quad \mathbf{V}_{\sigma_{50n}}]^{-1} \tag{1.12}$$

The ability of the NN to *classify* the data can be evaluated by checking the recognition rate for the data that was used for the training (called seen data). To do so, Eq.1.3 is also generalized for all the $50n$ audio samples (Eq.1.13). The resulting matrices $\hat{\mathbf{T}}_\sigma$ are estimators of the digit that was pronounced in the audio file σ. Each column in $\hat{\mathbf{T}}_\sigma$ corresponds to a time step τ in the signal, and each line corresponds to a digit (the first line corresponds to 0, the second to 1 and so on). Hence, a value close to one at the nth

line in a given column implies that there is a high probability that the digit $n-1$ was pronounced at this specific time step. The estimation of the pronounced digit can thus be retrieved for each time step as the index of the maximum value in each column of $\hat{\mathbf{T}}_\sigma$ using the argmax() function.

$$\left[\hat{\mathbf{T}}_{\sigma_1} \quad \hat{\mathbf{T}}_{\sigma_2} \quad ... \quad \hat{\mathbf{T}}_{\sigma_{50n}}\right] = \mathbf{W}\left[\mathbf{V}_{\sigma_1} \quad \mathbf{V}_{\sigma_2} \quad ... \quad \mathbf{V}_{\sigma_{50n}}\right] \tag{1.13}$$

The recognition rate can be evaluated with two approaches :

1. **Tau-Wise (TW)** approach : Each audio file σ is composed of a specific number N_τ of time steps, which depends on the length of the file. With this approach, the recognition of each time step is considered individually (see Eq.1.14, with $\hat{\mathbf{T}}_{\sigma,\tau}$ a given column of $\hat{\mathbf{T}}_\sigma$ and $\hat{d}_{\sigma,\tau}$ the corresponding estimation of the pronounced digit). The score for the recognition of a single audio file σ is given by Eq.1.15, with x the number of time steps in the signal that are correctly recognized and N_τ the total number of time steps in this signal/audio file.

$$\hat{d}_{\sigma,\tau} = \text{argmax}\left(\hat{\mathbf{T}}_{\sigma,\tau}\right) \tag{1.14}$$

$$\text{score}_\sigma = \frac{x}{N_\tau} \in [0,1] \tag{1.15}$$

   In some cases, there can be more than one maximal value in a given column of $\hat{\mathbf{T}}_\sigma$. This corresponds to a situation where the neural network was not able to treat the data efficiently. Hence, the pronunciation of all the corresponding digits is equiprobable. Therefore, the estimation $\hat{d}_{\sigma,\tau}$ is chosen randomly between the different maximums.

2. **Winner-Takes-All (WTA)** approach : Here, the mean is computed on each time step of a file to obtain an average of the estimations for the whole audio file. Then, the estimated digit is obtained with the argmax() function, and finally compared with the true target (Eq.1.16). The score is thus either 0 or 1 for the whole signal. This method has the advantage to absorb any potential inaccuracies that may occur with the TW approach, thanks to the average.

$$\hat{d}_{\sigma,\text{av}} = \text{argmax}\left(\frac{\sum_\tau \hat{\mathbf{T}}_{\sigma,\tau}}{N_\tau}\right) \tag{1.16}$$

   Indeed, if less than half the number of samples in a signal is recognized incorrectly, the WTA approach will absorb the errors thanks to the averaging, and the whole signal will still be recognized correctly. This is not the case with the TW approach, as each sample is treated independently in each signal. This difference is illustrated in Fig.1.25.

At first sight, it may look like the TW approach is not efficient compared to the WTA approach, as the recognition rate of the latter is generally higher (see Fig.1.25). However, one must remember than the TW approach can be used to recognize partial data inputs with a relatively high accuracy, while the WTA approach must rely on the whole signal. In the applicative context, the TW approach hence finds its utility in the recognition of vocal snippets for example.

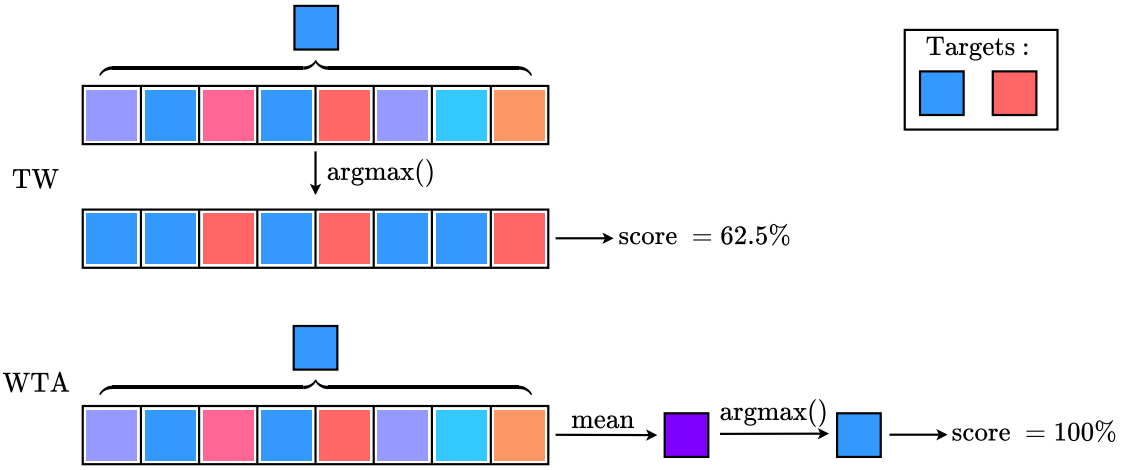

Fig. 1.25: Distinction between the TW and WTA approaches. The WTA approach absorbs the errors occurring in the recognition of the whole signal, leading to a score of 100%.

Testing phase

Now that **W** has been computed, the recognition can be tested with unseen data ($50(10-n)$ signals). This is done by using Eq.1.13, Eq.1.14, Eq.1.15 and Eq.1.16. Again, both TW and WTA approaches can be used.

Note that to ensure the cross-validation of the results, the study is performed multiple times with the different possible combinations of training and testing subsets. As there are n training subsets, the final results are averaged over the

$$\frac{10!}{n!(10-n)!}$$

possible ways to choose the training and the testing subsets [4].

Results

The results were studied accordingly to the acoustic filter that was used in [4]. It was shown that the contribution of the NN depends on the acoustic filter used. Indeed, if the filter is strongly non-linear (like the cochleagram or the Spectro HP model), the contribution of the reservoir is far lower than with a more linear filter, or no filter at all. This is due to the fact that the role of the reservoir is to treat the data non-linearly. If this treatment is already done by the acoustic filter, the contribution of the NN is reduced (Fig.1.26, [4]). Besides highlighting the role of non-linearity, which is one of the three key features of a RNN (see subsection 1.3.4), these results show the ability of a hardware RNN to treat data efficiently while requiring a minimal amount of pre-treatment (it can be seen that without any acoustic filtering, the WSR can get up to 80%).

Another interesting thing to note is that the performances of the simulated STVO (in purple in Fig.1.26) are slightly under-evaluated compared to the the experimental STVO. This difference is probably due to an inaccuracy of the model used for the simulations. In [4], the transient state of the dynamics of the vortex core was simulated using a approximation based on exponential decays. It is however suspected that the dynamics of the core is far more complex than that. An accurate description of the non-linear dynamics

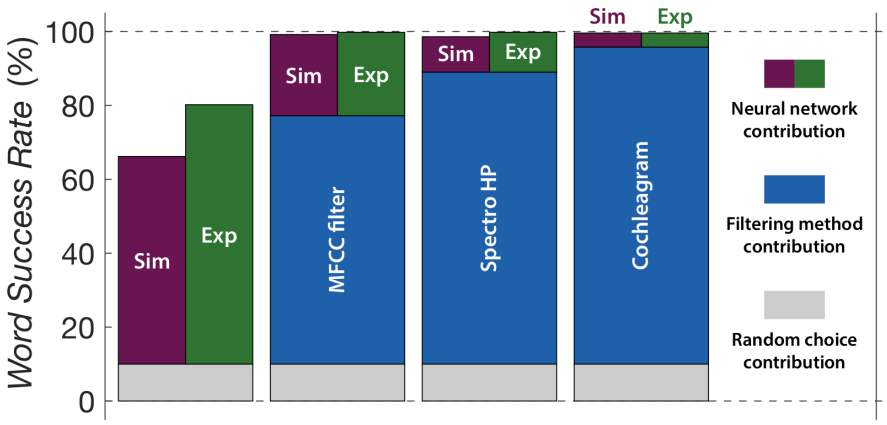

Fig. 1.26: Contributions to the spoken digit cross-validated test recognition rate. Random choice level is shown in grey, the filtering methods in blue, and the neural network under the reservoir computing approach in purple and green for the simulations and experiments, respectively. Here, 9 data subsets (90% of the database) are used for training our reservoir computing model and the remaining subset (10% of the database) is used to perform the recognition task. Extracted from [4].

of the vortex core is undeniably required in order to properly simulate its behavior. Proper simulations can then lead to a better understanding of the influence of various parameters on the performances of the NN, which is itself important to design the best physical sample and setup. **This conclusion, as well as the open questions raised by the various results obtained in [4] and [15], is actually at the origin of this work.**

1.3.4 Important features

Before going further, let's remind the important features required by a hardware NN. The performances of a hardware NN are based on 3 characteristics. Each of them influences in its own way the ability of the NN to classify and recognize data.

Non-linearity

The non-linearity of the neurons is the key for an efficient transformation of the inputs. Non-linearity can be added thanks to the acoustic filtering of the input, which will extract the useful features from the signal before it even enters the reservoir. It can also be added thanks to the non-linear response of the STVO. The conditions under which the oscillator is used (like the bias current I_{dc} and the amplitude of the signal) also have an influence on the dynamical regime that is involved into the treatment of the data, hence influencing the performances of the NN.

Stability

The performances of the NN are also highly impacted by external perturbations, like noise. It is important to test and quantify the ability of the NN to classify and recognize noisy data, as nanostructures are greatly impacted by noise.

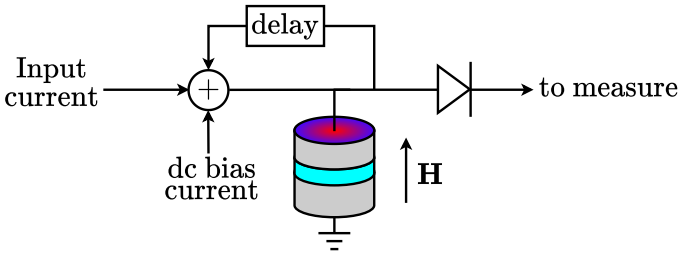

Fig. 1.27: Setup of delayed feedback loop STVO

Memory

Memory is also very important to handle temporal problems such as pattern and speech recognition. It can be tuned by varying the rate at which the input samples enter the reservoir, the goal being to maintain a transient state in the dynamics of the vortex core. Fast input rates will indeed outpace the transient state of the dynamics, while slow rates will saturate the dynamics into the steady state. A compromise should thus be found. More studies have proved the feasibility of adding artificial extrinsic memory into the NN by connecting a delayed feedback loop to the STVO emulating the reservoir (Fig.1.27). The input signals thus enter the NN multiple times due to the feedback, allowing the oscillator to treat the current data along with ancient data. As only the recent history is important in the treatment of temporal problems, the signal is attenuated every time it re-enters the oscillator, thus implementing *fading memory*. This extrinsic memory has been shown to allow the STVO to remember past inputs tens of times longer than with only intrinsic memory, hence greatly improving the classification efficiency [24].

Summary

Magnetic nano-oscillators such as STVOs can be used to design hardware nano-devices imitating the behavior of biological neurons. This is possible thanks to the non-linear dynamics of their magnetization, which is itself due to the spin transfer torque, an interaction whose foundations can be explained using spintronics. If used in the framework computing, a technique of machine learning consisting in designing a fixed and random neural network called a reservoir, they can allow to perform concrete machine learning tasks like pattern and speech recognition. Such hardware neural networks, in the framework of reservoir computing, are much more energy efficient than classic CMOS implementations. They can also be used to circumvent the limitations inherent to the von Neumann architecture, on which the majority of current computers are based.

Chapter 2

The Thiele equation approach (TEA)

Using STVOs as neurons requires the ability to understand and predict the dynamics of the vortex core accurately. Hence, Eq.1.6 needs to be solved over a sufficiently large domain and integrated over a sufficiently large simulation time. Moreover, it has been shown in [4] that the model that was used until now to simulate the behavior of the vortex core was not accurate enough to match the physical reality. Accurate and efficient simulations of STVOs are thus the key to enhance the understanding and the development of hardware neural networks based on spintronics.

In this work, the Thiele equation approach (TEA) was used to efficiently simulate the dynamics of the vortex core with increased accuracy, and to answer to the open questions raised by the results of the previous works ([4], [15]). Furthermore, several implementations were explored in order to speed up the simulations as much as possible.

Contributions

The results exposed in the following chapter were obtained by the team composed of Flavio ABREU ARAUJO (F.A.A.), Chloé CHOPIN (C.C.), and Simon de WERGIFOSSE (S.d.W.).

The work performed in this master thesis consisted in the understanding, appropriation and description of the different models and their mathematical foundations in the more general scope of the present work.

2.1 Micromagnetic simulations

Micromagnetic simulations (MMS) consist in simulating the magnetic behavior of a material at the sub-micrometer scale. To do so, a magnetic sample (the spatial domain) is discretized into a grid, and so is the time over which the simulation is carried out. This kind of simulations allows to obtain quantitative results of high quality, matching the experiment with a high precision. However, MMS in spintronics are based on the solving and integration of Eq.1.6 over the discretization grid, which is highly energy and time consuming [25]. Indeed, this grid is required to have a cell length smaller than the exchange length of the material composing the free FM layer to accurately model the interactions between the magnetic moments in the material [21]. The exchange length,

which measures the relative strength of exchange and self-magnetostatic energies [26], is typically between 1 and 100 nm [27]. Furthermore, the effective magnetic field that is used to compute the dynamics of the vortex core is composed of numerous different contributions, like the magnetostatic, magnetocrystalline and exchange energies. The sum of those contributions needs to be minimized over the whole system using numerical methods in order to simulate the magnetic behavior as accurately as possible. Finally, long range interactions can lead to high spatial complexities ($\mathcal{O}(N^2)$), therefore requiring compromises between the quality of the results and the required resources [25]. Hence, other ways to simulate the dynamics of a STVO more efficiently have been looked for, for example by deriving analytical expressions. However, micromagnetic simulations remain of great interest, for example for the verification of said analytical models, or the correction of the latter using simulated values.

2.2 Thiele equation approach (TEA)

This section features developments and results obtained by the team of Dr. Flavio ABREU ARAUJO. This work is still under progress, but part of the results will soon be the object of a publication in Scientific Reports.

The Thiele equation describes the dynamics of the vortex core in a magnetic dot (here, the magnetic dot is actually the free layer of the STVO). By considering the core as a quasiparticle, one can write Eq.2.1 where $\mathbf{X}$ represents the position of the vortex core in the $(\mathbf{e}_x\mathbf{e}_y)$ plane.

$$G\left(\mathbf{e}_z \times \dot{\mathbf{X}}\right) + D\dot{\mathbf{X}} = \frac{\partial W}{\partial \mathbf{X}} + \mathbf{F}^{\mathrm{ST}} \tag{2.1}$$

G and D are the gyrotropic and damping constants (Eq.2.2 and Eq.2.3, with P the polarity of the magnetic vortex, M_s the saturation magnetization of the free layer, h its thickness, γ_G the Gilbert gyromagnetic ratio, α_G the Gilbert damping parameter and $\eta = \frac{1}{2}\ln(R/2l_\mathrm{ex}) + \frac{3}{8}$, where $l_\mathrm{ex} = \sqrt{A/(2\pi M_\mathrm{s}^2)}$ is the exchange length, A being the exchange stiffness coefficient).

$$G = -2\pi P M_\mathrm{s} h/\gamma_\mathrm{G} \tag{2.2}$$

$$D = -\alpha_\mathrm{G}\eta|G| \tag{2.3}$$

W is the potential magnetic energy related to a displacement of the vortex core, expressed using different contributions and their corresponding spring-like restoring force constants in Eq.2.4, Eq.2.5, Eq.2.6 (where $\Lambda_{0,\xi}$, a_ξ, b_ξ and c_ξ are parameters depending on the aspect ratio of the free layer ξ, which can be determined using numerical methods [28]), Eq.2.7 and Eq.2.8.

$$\frac{\partial W}{\partial \mathbf{X}} = \frac{\partial\left(W^{\mathrm{ex}} + W^{\mathrm{ms}} + W^{\mathrm{Oe}}\right)}{\partial \mathbf{X}} = \left(k^{\mathrm{ex}} + k^{\mathrm{ms}} + k^{\mathrm{Oe}}\right)\mathbf{X} = k\mathbf{X} \tag{2.4}$$

$$k^{\mathrm{ex}} = (2\pi)^2 h M_\mathrm{s}^2 \frac{(l_\mathrm{ex}/R)^2}{(1-s^2)} \tag{2.5}$$

$$k_\xi^{\mathrm{ms}} = \frac{8M_\mathrm{s}^2 h^2}{R}\Lambda_{0,\xi}\left(1 + a_\xi s^2 + b_\xi s^4 + c_\xi s^6\right) \tag{2.6}$$

$$k^{\mathrm{Oe}} = C J_{\mathrm{dc}} \kappa^{\mathrm{Oe}} \tag{2.7}$$

$$\kappa^{\mathrm{Oe}} = \frac{8\pi^2}{75} M_s R h \left(1 - \frac{4}{7}s^2 - \frac{1}{7}s^4 - \frac{16}{231}s^6 - \frac{125}{3003}s^8\right) \tag{2.8}$$

$\mathbf{F}^{\mathrm{ST}}$ represents the forces related to the perpendicular component of the STT effect (Eq.2.9) [28, 29], with $\kappa_{\perp}^{\mathrm{ST}}$ expressed in Eq.2.10, $a_J = p_J \hbar/(2|e|M_{\mathrm{s}}^{\mathrm{ref}} h)$ being the spin-transfer efficiency, with p_J the polarization of the spin-polarized current and $M_{\mathrm{s}}^{\mathrm{ref}}$ the saturation polarization of the fixed layer.

$$\mathbf{F}^{\mathrm{ST}} = \mathbf{F}_{\perp}^{\mathrm{ST}} = \kappa_{\perp}^{\mathrm{ST}} J_{\mathrm{dc}} \left(\mathbf{e}_z \times \mathbf{X}\right) \tag{2.9}$$

$$\kappa_{\perp}^{\mathrm{ST}} = \pi a_J M_s h p_z \tag{2.10}$$

Under specific conditions (when the current is spin-polarized perpendicularly to the plane of the magnetic vortex of the free layer), Eq.2.1 can be rewritten as Eq.2.11, then leading to a set of linear first order differential equations (Eq.2.12), with Γ a driving parameter ($\Gamma > 0$ if the orbit of the core is increasing, $\Gamma < 0$ if it is decreasing, and $\Gamma = 0$ if steady-state has been reached), and ω the angular frequency of the precessional movement [28]. Both of these variables depend on the injected DC current density J_{dc}, as well as the D and G constants, and the spring-like restoring force constants k^{ex}, k^{ms} and k^{Oe} respectively related to the exchange energy, the magnetostatic energy and the Ampère-Oersted field generated by the injected current (Eq.2.13 and Eq.2.14, [28]) .

$$\begin{bmatrix} D & -G \\ G & D \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} k & \kappa_{\perp}^{\mathrm{ST}} J_{\mathrm{dc}} \\ -\kappa_{\perp}^{\mathrm{ST}} J_{\mathrm{dc}} & k \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \underbrace{\begin{bmatrix} \Gamma & -\omega \\ \omega & \Gamma \end{bmatrix}}_{\overline{\overline{\Omega}}} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.12}$$

$$\Gamma = \frac{D\left(k^{\mathrm{ex}} + k^{\mathrm{ms}} + C J_{\mathrm{dc}} k^{\mathrm{Oe}}\right) + G J_{\mathrm{dc}} \kappa_{\perp}^{ST}}{D^2 + G^2} \tag{2.13}$$

$$\omega = \frac{D J_{\mathrm{dc}} \kappa_{\perp}^{\mathrm{ST}} - G\left(k^{\mathrm{ex}} + k^{\mathrm{ms}} + C J_{\mathrm{dc}} k^{\mathrm{Oe}}\right)}{D^2 + G^2} \tag{2.14}$$

The solution of Eq.2.12 is given by Eq.2.15 [28].

$$\begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} = ||\mathbf{X}|| \exp(\Gamma t) \begin{bmatrix} \sin(\omega t) \\ -\cos(\omega t) \end{bmatrix} \tag{2.15}$$

Solving Eq.2.12 thus allows to retrieve the position of the vortex core analytically and its reduced position $s(t)$ (see Eq.2.16, with R the radius of the dot) for any current density J_{dc} between J_{cr1} and J_{cr2}.

$$s(t) = \frac{||\mathbf{X}(t)||}{R} = \frac{\sqrt{X^2(t) + Y^2(t)}}{R} \tag{2.16}$$

J_{cr1} is defined as the current density needed to trigger the gyrotropic motion of the vortex core. This is the current density required to reach steady state (when the STT

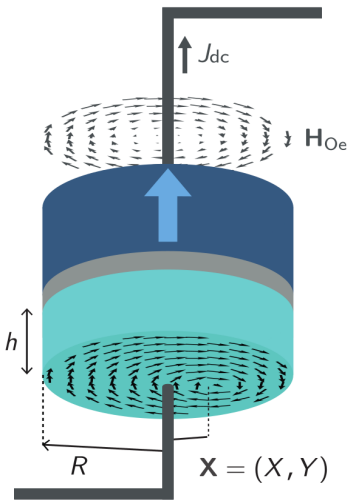

Fig. 2.1: Illustration of the Ampère-Oersted field due to the injection of current in a STVO. The chirality is C^+ as $\mathbf{H}_{\text{Oe}}$ is curling in the same direction as the in-plane magnetization of the vortex. Extracted from [28].

interaction perfectly counteracts the magnetic damping) when the core is still at the center of the vortex (Eq.2.17). It can be finally expressed as Eq.2.18. On the other hand, J_{cr2} is the current density needed by the core to reach its maximum orbit (at $s = 0.8R$). Indeed, the vortex becomes unstable if its reduced position s gets any further, and might eventually be expelled out of the oscillator.

$$\Gamma(s=0) = \frac{D_0\left(k_0^{\text{ex}} + k_0^{\text{ms}} + CJ_{\text{dc}}k_0^{\text{Oe}}\right) + G_0 J_{\text{cr1}}\kappa_\perp^{ST}}{D_0^2 + G_0^2} = 0 \tag{2.17}$$

$$J_{\text{cr1}} = \frac{-D_0\left(k_0^{\text{ex}} + k_0^{\text{ms}}\right)}{D_0 C k_0^{\text{Oe}} + G_0\kappa_\perp^{\text{ST}}} \tag{2.18}$$

A splitting phenomenon was also highlighted in [28]. Indeed, it was shown that Eq.2.18 and micromagnetic simulations yielded different results for the value of J_{cr1} and J_{cr2} depending on the consideration of the Ampère-Oersted field $\mathbf{H}_{\text{Oe}}$ generated by the current injected into the oscillator (see Fig.2.1), and depending on the chirality C ($C = +1$ when $\mathbf{H}_{\text{Oe}}$ is curling into the same direction as the vortex, $C = -1$ otherwise), as it can be seen in Tab.2.1. This is an important observation for the future studies as the Ampère-Oersted field, which is usually neglected, is proved to have a non-negligible influence on the value of the parameters ruling the dynamics of the vortex core, and its behavior in general, leading to three distinct regimes of operation (see Fig.2.2 and Fig.2.3).

In practice, the time-dependent position of the vortex core $\mathbf{X}(t)$ can be obtained by solving Eq.2.12 numerically. To do so, the value of Γ and ω must be accurately computed. In the steady-state regime, both of those quantities are constant but in the transient state (which is actually of interest for neuromorphic applications), they are strongly non-linearly dependent on the reduced position s of the core, due to a dependence of G and D on s. No convincing expressions for $G(s)$ and $D(s)$ exist in the literature, but satisfying expressions can be found using even power expansions of s (see Eq.2.19 and Eq.2.20). This is exactly what is performed in an undergoing work of the team composed of Dr.

	C^+	noOe	C^-
$J_{\mathrm{cr1}}^{\mathrm{TEA}}$ [MA/cm^2]	6.27	5.88	5.54
$J_{\mathrm{cr1}}^{\mathrm{num}}$ [MA/cm^2]	6.47	6.11	5.77
$J_{\mathrm{cr2}}^{\mathrm{num}}$ [MA/cm^2]	9.83	9.10	8.43
$J_{\mathrm{cr1}}^{\mathrm{num}}/J_{\mathrm{cr2}}^{\mathrm{num}}$	1.52	1.50	1.46

Tab. 2.1: Comparison of J_{cr1} given by Eq.2.18 under TEA and micromagnetic results for the different configurations of the Ampère-Oersted field and the in-plane magnetization. noOe corresponds to neglecting $\mathbf{H}_{\mathrm{Oe}}$. Extracted from [28].

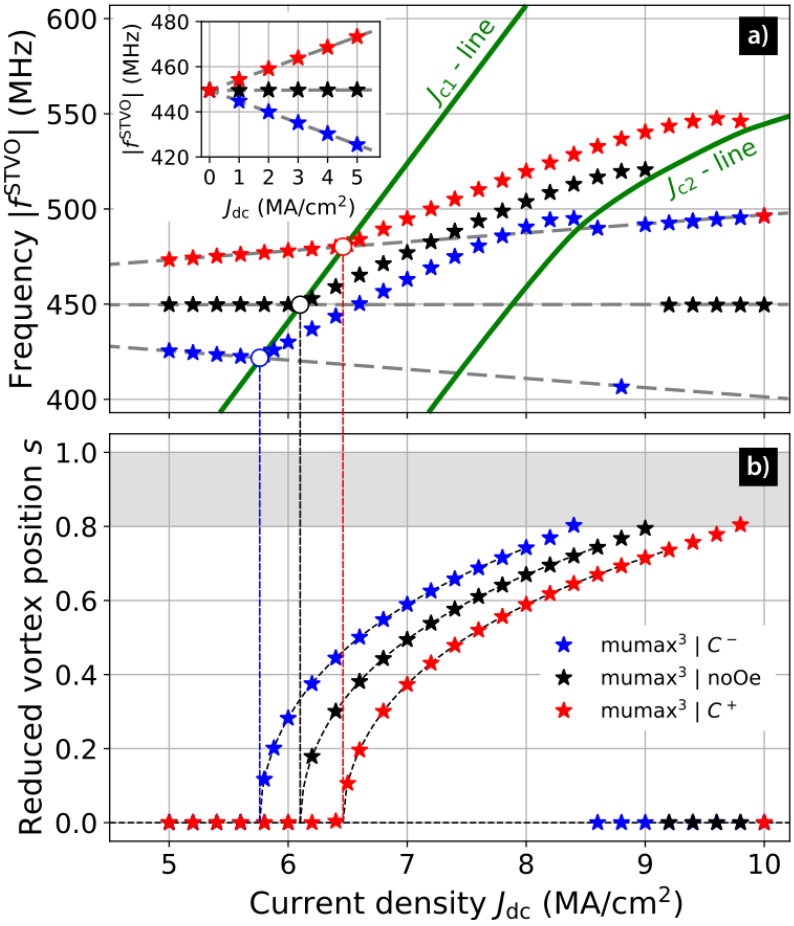

Fig. 2.2: Vortex dynamical properties vs DC current excitation. **a)** Absolute values of the vortex gyrotropic frequency f^{STVO} and **b)** vortex reduced position s as a function of the DC current density J_{dc}. The colours black, red and blue correspond to simulations without Ampère-Oersted field (noOe), with with Ampère-Oersted field and $C = +1$ (C^+) and with Ampère-Oersted field and $C = -1$ (C^-), respectively. The critical current J_{cr1} gives rise to the first green J_{cr1}-line. The J_{cr1}-line represents the transition between the first resonant regime and the steady-state gyroscopic regime. The J_{cr1} values are plotted across both sub-figures by corresponding dashed lines. Extracted from [28].

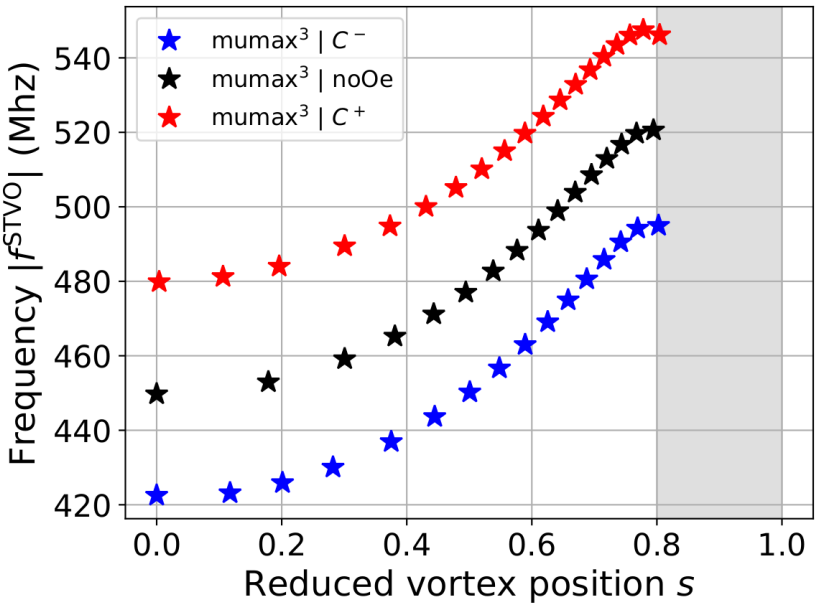

Fig. 2.3: Gyroscopic frequency $\left|f^{\mathrm{STVO}}\right|$ as a function of the reduced vortex core position s. Only data in the steady-state oscillating regime is shown here. Extracted from [28].

Flavio ABREU ARAUJO, Chloé CHOPIN and Simon de WERGIFOSSE. In their work, the different coefficients are obtained by fitting micromagnetic simulations results to the resulting expressions. This approach was hence defined as *Data-driven Thiele equation appproach* (DD-TEA), and a paper on this subject is currently pending publication.

$$G(s) = G_0\left(1 + a_G s^2 + b_G s^4 + c_G s^6 + d_G s^8\right) \tag{2.19}$$

$$D(s) = D_0\left(1 + a_D s^2 + b_D s^4 + c_D s^6 + d_D s^8\right) \tag{2.20}$$

Despite being quantitatively accurate, solving Eq.2.12 still suffers from the drawbacks related to the rapidly varying solution of Eq.2.15, which is not ideal in the case of a numerical resolution.

Therefore, in order to describe the dynamics of the vortex core much more efficiently than with Eq.2.12, the focus can be made on the value of the envelope $s(t)$ that represents the reduced position of the vortex core (Eq.2.16) rather than on its position $\mathbf{X}(t)$. Indeed, the envelope varies far more slowly than $\mathbf{X}(t)$ (Fig.2.4). The ODE to be solved thus becomes

$$\dot{s} = \overline{\overline{\Omega}} s \tag{2.21}$$

and the solution writes

$$s(t) = \frac{||\mathbf{X}(t)||}{R} \tag{2.22}$$

$$= \left|\left| \frac{||\mathbf{X}||}{R} \exp(\Gamma t) \begin{bmatrix} \sin(\omega t) \\ -\cos(\omega t) \end{bmatrix} \right|\right| \tag{2.23}$$

$$= s_0 \exp(\Gamma t) \tag{2.24}$$

with s_0 the steady-state reduced position of the vortex core when only the bias current density is injected on the STVO (i.e in the absence of input signal).

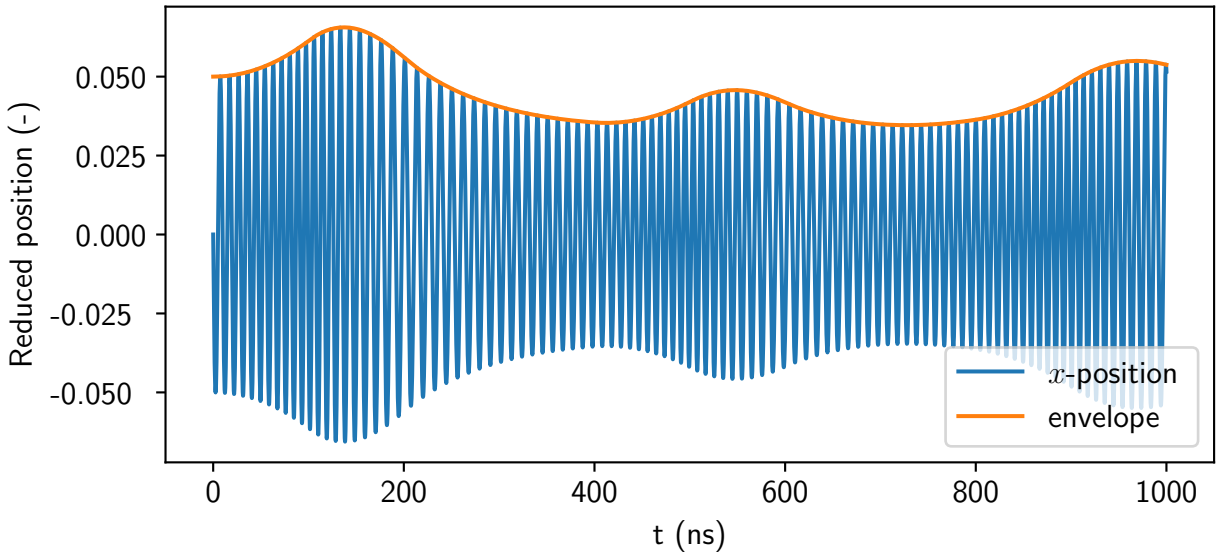

Fig. 2.4: Comparison between the reduced time-dependent x-position of the vortex core and the envelope $s(t)$

Note that Eq.2.24 only holds for a constant Γ. However, it is known that Γ is actually strongly non-linearly dependent on s in the transient state. As s is itself dependent on t, one can finally write the solution as Eq.2.25.

$$s(t) = s_0 \exp\left(\int_0^t \Gamma(s(t'))\mathrm{d}t'\right) \tag{2.25}$$

As $\Gamma(s(t))$ is actually very difficult to integrate (see Eq.2.13), the relation of Eq.2.25 is re-expressed as follows :

$$\frac{s(t)}{s_0} = \exp\left(\int_0^t \Gamma(s(t'))\mathrm{d}t'\right) \tag{2.26}$$

$$\Leftrightarrow \ln\left(\frac{s(t)}{s_0}\right) = \int_0^t \Gamma(s(t'))\mathrm{d}t' = \ln(s(t)) - \ln(s_0) \tag{2.27}$$

$$\Leftrightarrow \ln(s(t)) = \int_0^t \Gamma(s(t'))\mathrm{d}t' + \ln(s_0) \tag{2.28}$$

By integrating both sides of Eq.2.28, and thanks to the fundamental theorem of calculus, one finally obtains

$$\frac{1}{s(t)}\dot{s}(t) = \Gamma(s(t)) \tag{2.29}$$

$$\Leftrightarrow \dot{s}(t) = \Gamma(s(t))s(t) \tag{2.30}$$

2.3 Low-order Thiele equation approach (LO-TEA)

The main feature of LO-TEA is the truncation of the power expansion of $\Gamma(s)$ to the second order. Indeed, $\Gamma(s)$ can also be expressed as an even power expansion of s. LO-TEA thus considers $\Gamma(s)$ as in Eq.2.31.

$$\Gamma(s(t)) = \alpha + \beta s^2(t) \tag{2.31}$$

The α and β coefficients can themselves be expressed as a polynomial of the current density $J = J_{\mathrm{dc}} + J_{\mathrm{in}}$, with J_{dc} the DC bias current density and J_{in} the current density related to the input signal (Eq.2.32, Eq.2.34), so that the final expression for $\Gamma(s(t))$ is Eq.2.35.
Finally, Eq.2.31 and Eq.2.35 allow to rewrite Eq.2.30 as Eq.2.36 and Eq.2.37, which are in fact Bernoulli differential equations (i.e of the form $y' + P(x)y = Q(x)y^n$).

$$\alpha(J) = a_J J + a \tag{2.32}$$
$$= a_J \left(J - J_{\mathrm{cr1}}\right) \tag{2.33}$$

$$\beta(J) = b_J J + b \tag{2.34}$$

$$\Gamma(s(t)) = J\left(a_J + b_J s^2(t)\right) + \left(a + b s^2(t)\right) \tag{2.35}$$

$$\dot{s}(t) = \alpha s(t) + \beta s^3(t) \tag{2.36}$$
$$= \left(a_J J + a\right) s(t) + \left(b_J J + b\right) s^3(t) \tag{2.37}$$

The 4 parameters involved in the dynamics of the vortex core under the LO-TEA are expressed in Eq.2.38, Eq.2.39, Eq.2.40 and Eq.2.41. Depending on the chirality of the magnetic vortex, and the consideration -or not- of the Ampère-Oersted field, three distinct sets of parameters $\{a_J, b_J, a, b\}$ can be derived, due to the splitting phenomenon introduced in [28]. Note that the first critical current density can be expressed using these parameters as in Eq.2.42.

$$a_J = \frac{C D_0 \kappa_0^{\mathrm{Oe}} + G_0 \kappa^{\mathrm{ST}}}{D_0^2 + G_0^2} \tag{2.38}$$

$$b_J = \frac{-C D_0^3 \kappa_0^{\mathrm{Oe}} a^{\mathrm{D}} + C D_0^3 \kappa_0^{\mathrm{Oe}} a^{\mathrm{Oe}} + C D_0 G_0^2 \kappa_0^{\mathrm{Oe}} a^{\mathrm{D}} - 2 C D_0 G_0^2 \kappa_0^{\mathrm{Oe}} a^{\mathrm{G}} + C D_0 G_0^2 \kappa_0^{\mathrm{Oe}} a^{\mathrm{Oe}} - 2 D_0^2 G_0 \kappa^{\mathrm{ST}} a^{\mathrm{D}} + D_0^2 G_0 \kappa^{\mathrm{ST}} a^{\mathrm{G}} - G_0^3 \kappa^{\mathrm{ST}} a^{\mathrm{G}}}{D_0^4 + 2 D_0^2 G_0^2 + G_0^4} \tag{2.39}$$

$$a = \frac{D_0 k_0^{\mathrm{ex}} + D_0 k_0^{\mathrm{ms}}}{D_0^2 + G_0^2} \tag{2.40}$$

$$b = \frac{-D_0^3 a^{\mathrm{D}} k_0^{\mathrm{ex}} - D_0^3 a^{\mathrm{D}} k_0^{\mathrm{ms}} + D_0^3 a^{\mathrm{ms}} k_0^{\mathrm{ms}} + D_0^3 k_0^{\mathrm{ex}} + D_0 G_0^2 a^{\mathrm{D}} k_0^{\mathrm{ex}} + D_0 G_0^2 a^{\mathrm{D}} k_0^{\mathrm{ms}} - 2 D_0 G_0^2 a^{\mathrm{G}} k_0^{\mathrm{ex}} - 2 D_0 G_0^2 a^{\mathrm{G}} k_0^{\mathrm{ms}} + D_0 G_0^2 a^{\mathrm{ms}} k_0^{\mathrm{ms}} + D_0 G_0^2 k_0^{\mathrm{ex}}}{D_0^4 + 2 D_0^2 G_0^2 + G_0^4} \tag{2.41}$$

$$J_{\mathrm{cr1}} = \frac{-a}{a_J} \tag{2.42}$$

Solving Eq.2.21 using Eq.2.31 is far more stable on the numerical point of view than solving Eq.2.12, as the envelope varies far more slowly than the position of the core (Fig.2.4). The corresponding less steep solution allows the numerical solver to reach convergence much more quickly than with the xy solution of Eq.2.15. Combined with the simplified expression for the derivative of the envelope $\dot{s}(t)$ at Eq.2.30, this allows to solve

Eq.2.21 about 10 times faster than Eq.2.12. This way of modeling the dynamics of the vortex core will be referred to as the LO-TEA s-ODE model.

Using Eq.2.25 and Eq.2.31, it is eventually possible to derive a fully analytical expression for $s(t)$ (Eq.2.43). This analytical expression constitutes the most advantageous feature of the LO-TEA. Indeed, no more numerical resolution is needed in order to solve the dynamics of the vortex core. This allows to significantly decrease the time needed to simulate a speech recognition task with the emulated STVO.

$$s(t) = \frac{s_0}{\sqrt{\left(1 + \frac{s_0^2}{\alpha/\beta}\right)\exp(-2\alpha t) - \frac{s_0^2}{\alpha/\beta}}} \tag{2.43}$$

Furthermore, the reduced position $s_\mathrm{f} = s(t \to \infty)$ of the vortex core when steady-state is reached under a constant current density (i.e when $\Gamma = 0$) is given by Eq.2.46.

$$s_\mathrm{f}(J_\mathrm{dc}) = s(t \to \infty) \tag{2.44}$$

$$= \frac{s_0}{\sqrt{-\frac{s_0^2}{\alpha/\beta}}} = \frac{s_0}{\left(\frac{s_0}{\sqrt{-\alpha/\beta}}\right)} = s_0 \frac{\sqrt{-\alpha/\beta}}{s_0} \tag{2.45}$$

$$= \sqrt{\frac{-\alpha(J_\mathrm{dc})}{\beta(J_\mathrm{dc})}} \tag{2.46}$$

LO-TEA thus allows to build two distinct models for the simulation of the dynamics of the vortex core. The first one is based on the numerical solving of an ODE, where the required expression for the derivative $\dot{s}(t)$ is simple and quickly computable. The second model is an analytical expression for $s(t)$, allowing explicit and trivial computation of $s(t)$ by truncating the expansion of the driving parameter $\Gamma(s(t))$ to the second order.

2.4 High precision Thiele equation approach (HP-TEA)

As the expression at Eq.2.43 is a truncation of the true dynamics of the vortex core to the second order, one can suspect that a part of the complexity of the actual physical phenomenon is neglected. Fortunately, as Eq.2.36 and Eq.2.37 are Bernoulli differential equations, the power n that is involved can be any real number. In other words, it is possible to adapt Eq.2.36 and Eq.2.37 using a continuous range of power values (see Eq.2.47). Hence, the suspected lack of complexity of Eq.2.43 can be compensated using the homologous higer-order solution (Eq.2.48). To do so, a convenient expression for n must be found.

$$\dot{s}(t) = \alpha s(t) + \beta s^n(t) \tag{2.47}$$

$$s(t) = \frac{s_0}{\sqrt[n]{\left(1 - \frac{s_0^n}{\alpha/\beta}\right)\exp(-n\alpha t) + \frac{s_0^n}{\alpha/\beta}}} \tag{2.48}$$

b_J	-95.968883
c_J	23.535504
d_J	-2.871111
e_J	0.175043
f_J	-0.004282
b	157.142220

Tab. 2.2: Coefficients of Eq.2.49

b_J	-3032.315785
c_J	741.379846
d_J	-89.896255
e_J	5.425502
f_J	-0.130746
b	4930.675312

Tab. 2.3: Coefficients of Eq.2.51

It is possible to derive a polynomial expression $n(J)$ (see Eq.2.49, with the coefficients given in Tab.2.2) that gives for any current density between J_{cr1} and J_{cr2} the order n that allows to compensate the remaining complexity from the Thiele equation when plugged into Eq.2.48. The coefficients of Tab.2.2 were obtained by fitting results obtained by micromagnetic simulations to the polynomial. Finally, an expression homologous to Eq.2.46 can also be deduced, accounting for the higher order terms in the expression of $\Gamma(s)$ (Eq.2.50).

$$n(J) = b_J J + c_J J^2 + d_J J^3 + e_J J^4 + f_J J^5 + b \tag{2.49}$$

$$s_{\mathrm{f}}(J_{\mathrm{dc}}) = \left(\frac{\alpha(J_{\mathrm{dc}})}{\beta(J_{\mathrm{dc}})}\right)^{1/n(J_{\mathrm{dc}})} \tag{2.50}$$

Note that β was also expressed using a higher-order polynomial expression, by fitting the values obtained with MMS to the polynomial (Eq.2.51 and Tab.2.3).

$$\beta(J) = b_J J + c_J J^2 + d_J J^3 + e_J J^4 + f_J J^5 + b \tag{2.51}$$

Summary

The development of hardware neural networks using STVOs requires to accurately simulate their behavior. To do so, several techniques can be used, with different levels of accuracy and efficiency. Micromagnetic simulations allow to obtain quantitative results of high quality and accuracy. However, they are highly time consuming, which makes them undesirable for extensive simulations. The Thiele equation approach consists in using the Thiele equation to simulate the dynamics of the vortex core of a STVO. However, this approach suffers from the steepness of the rapidly varying coordinates of the vortex core in the plane of the free FM layer. By considering the envelope $s(t)$ of the gyrotropic motion of the vortex core rather than its $x(t)$ and $y(t)$ coordinates, the problem can be solved 10 times faster. To do so, the power expansion of the driving parameter $\Gamma(s(t))$ is truncated to the second order. This model is called the LO-TEA model. It is also possible to describe the reduced position of the vortex core using a purely analytical expression based on this truncation. This allows to considerably speed up the simulations compared to the numerical methods used previously. Finally, the loss of complexity due to the truncation in the LO-TEA analytical model can be compensated by generalizing the expression to the order n, which can be determined as a function of the current density using a polynomial fit of the results obtained by MMS. This constitutes the HP-TEA model.

Chapter 3

The benchmarking task

The developments made in chapter 2 allow to know the dynamics of the vortex core quickly and accurately, leading to an efficient ways to simulate the behavior of a STVO. It is hence possible to perform a recognition task in the framework of RC by simulating a virtual oscillator. To do so, a task of pattern recognition was designed. This was used to benchmark the different models presented earlier. Moreover, the rapidity at which the simulations can be carried out allows to study the influence of several operating parameters on the recognition rate of the simulated NN while performing the benchmarking task.

Contributions

The code implied in the next chapter was adapted from the original version written by Flavio ABREU ARAUJO (F.A.A). The implementation of the different models was also carried out by F.A.A.

The work performed in this master thesis consisted in the understanding and appropriation of the code, and its adaption into the benchmarking recognition task. It also includes the overall design of the benchmarking task, from the different databases to the recognition process, as well as the conduction of the parametric studies and the collection of the different results.

3.1 Database

The task consisted in the recognition of sine and square waveforms. A database containing 80 sine periods and 80 square periods was thus created. The database was split into 2 data subsets : one for training and one for testing. The targets corresponding to the database were also included.

Actually, 8 variants of the database were designed, based on the combination of 3 criteria :

- **Number of samples per period** : By default, each period was composed of 8 evenly spaced samples (Fig.3.1 and Fig.3.2). Another variant of the database contains periods spanning over 10 samples (Fig.3.3 and Fig.3.4). The two additional samples are 0s that are used to pad the periods. One of them is placed at each border of the signal. This was meant to simulate "blank" time steps between the

different periods. Note that the sine periods were slightly set out of phase so that the presence of two samples on $y = 0$ was avoided (see the first and the fifth dots on Fig.3.1). Hence, in the case of 8 samples, the sine periods are represented by the following

$$[0.050, 0.742, 1.0, 0.672, -0.050, -0.742, -1.0, -0.672]$$

while the square periods are represented by

$$[1, 1, 1, 1, -1, -1, -1, -1]$$

- **Format of the targets** : The targets, which are composed of the $\mathbf{T}$ matrix for each period, were implemented using scalar values and vectors. In case of a sine period, the target is either the value 1, or a vector similar to

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & ... & 1 & 1 & 1 \\ 0 & 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix}}_{\text{8 or 10 samples}}$$

  In the case of a square period, the target is either the value -1, or a vector similar to

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 1 & 1 & 1 & ... & 1 & 1 & 1 \end{bmatrix}}_{\text{8 or 10 samples}}$$

  The vector format is interesting as it is more quickly scalable to another number of categories of classification than with scalar targets.

  From now on, any time step whose estimated target $\hat{\mathbf{T}}$ is positive (in the scalar case) or $\begin{bmatrix} \sim 1 \\ \sim 0 \end{bmatrix}$ (in the vector case) is considered as being part of a sine period. For square time steps, the estimated target must be negative or $\begin{bmatrix} \sim 0 \\ \sim 1 \end{bmatrix}$ depending on the format of the targets.

- **Arrangement of the signals in each data subset** : Two variants were also designed based on the arrangement of the signals in the data subsets. In the first case, the signals were arranged in an alternate fashion in each data subset. In the other case, the signals were arranged randomly in the subsets. Note that as there are 80 sine periods and 80 square periods in the database, each data subset was composed of 40 sine periods and 40 square periods, no matter the arrangement of the latter in the subsets.

The database that was used in most of the studies in this work is composed of 8-sampled periods. Indeed, no significant improvement was observed with the padded signals. As the simulations are meant to match the reality as close as possible, and as both padded and non-padded waveforms are achievable in practice, the non-padded signals were chose to ensure consistency in the results.

Again, no significant difference was observed between the use of scalar and vector targets. Despite the easier scalability of vector targets for higher numbers of categories,

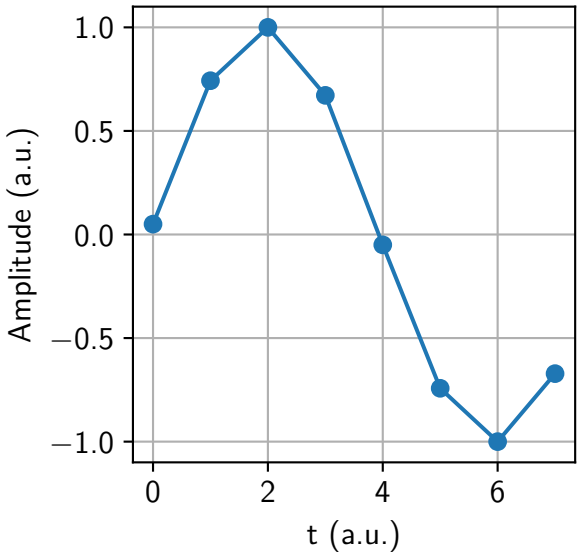

Fig. 3.1: 8-sampled sine period

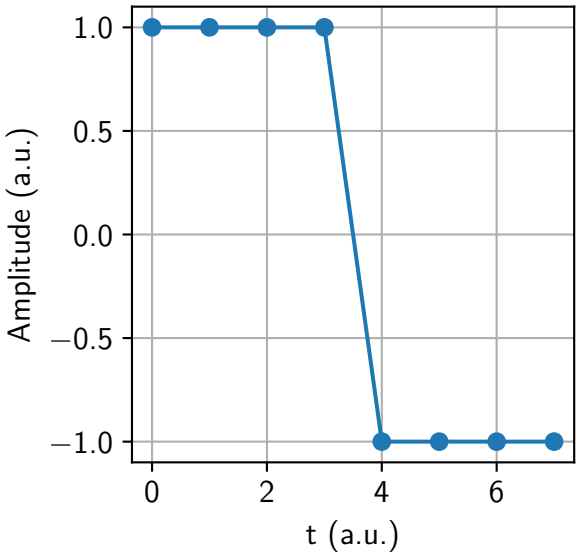

Fig. 3.2: 8-sampled square period

the use of scalar targets stays suitable for two categories of interest in this work. All of the studies were thus conducted with scalar targets for the sake of simplicity and consistency.

Finally, the database was arranged in both ways. Indeed, a subset of 80 alternate signals (40 sines + 40 squares) was used to train the NN. In practice, the data involved in training phase (classification) of a NN is well controlled, and hence the training data is allowed to be relatively clean. For the testing phase (recognition), a random subset of 40 sine and 40 square signals was used, as in most of the applicative situations, the input data is supposed to be more random than during the training phase.

3.2 Oscillator

Several types of simulated oscillators (i.e models) have been used during this thesis. Each of them has its pros and cons. During the work, these unpublished models were constantly improved in accurateness and efficiency, allowing simulations of increasing quality and a better understanding of the emulated neural network. Each of the approaches developed in chapter 2 is the base of a different model, allowing to simulate a STVO using `Python` accordingly.

3.2.1 Phenomenological oscillator

The phenomenological oscillator was the first model to have been developed, and the first to have been used. It consists in defining a set of parameters which, combined, allow to simulate the overall behavior of the vortex core, without necessarily understand the actual underlying mechanisms accurately. These parameters are listed in Tab.3.1 (and are encapsulated into the structure `OSC` in the code), and the code is available in section A.1. The input vector `Input` is composed of scalar values between 0 and 1, that will be multiplied by $\mathrm{Amp_V}$ to simulate an input voltage between 0 and $\mathrm{Amp_V}$. The

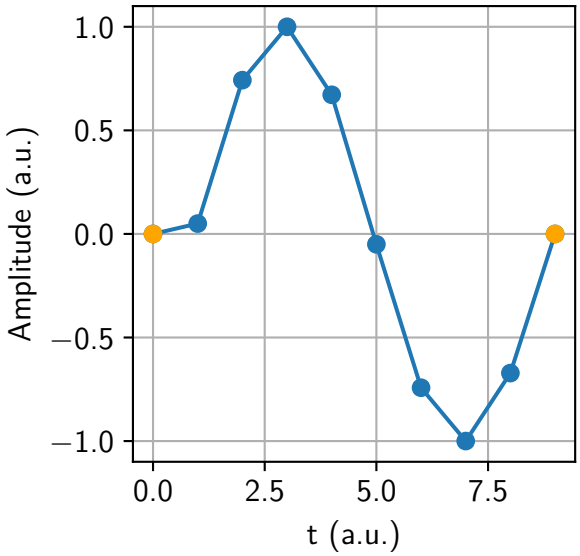

Fig. 3.3: 10-sampled sine period. The padding 0s are displayed in orange.

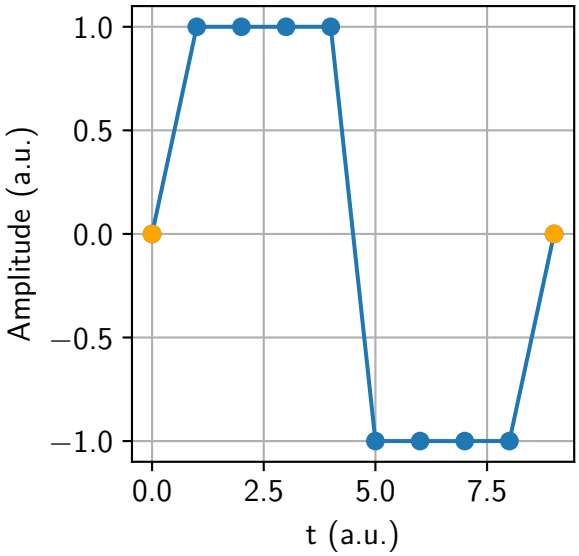

Fig. 3.4: 10-sampled square period. The padding 0s are displayed in orange.

$\mathrm{d}t$ (ns)	Time step (between two input samples)
T_{relax} (ns)	Relaxation time of the oscillator
R_{osc} (Ohm)	DC resistance of the oscillator
I_0 (A)	DC bias current
$\mathrm{Amp_V}$ (V)	Maximum amplitude of the input signal
p_{A} (-)	Offset constant
I_{c} (A)	First critical current

Tab. 3.1: Parameters for the phenomenological oscillator

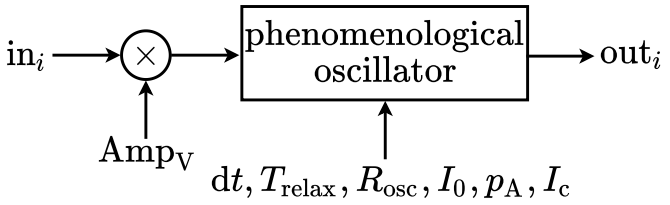

Fig. 3.5: Schematic of the phenomenological oscillator

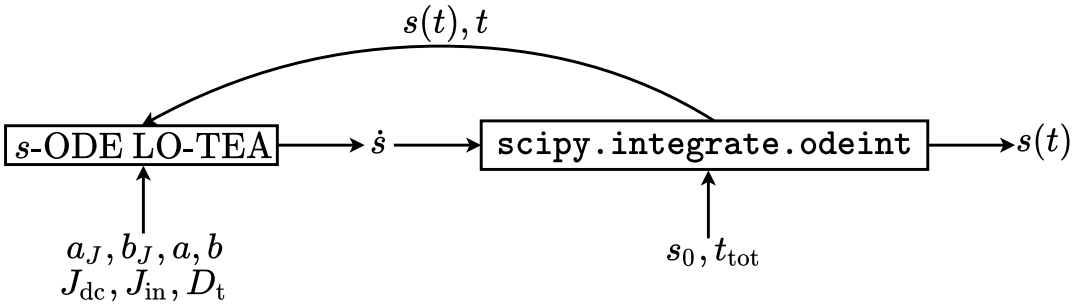

Fig. 3.6: Schematic of the s-ODE LO-TEA oscillator

output out$_i$ of the oscillator for an input sample in$_i$ is computed as in Eq.3.1, and the first value of the output is given by Eq.3.2 (Fig.3.5) [4]. The values of the input signal `Input` are supposed to be spaced in time by dt seconds.

$$\mathrm{out}_i = p_\mathrm{A}\sqrt{\left|(I_0 - I_\mathrm{c}) - \frac{\mathrm{Amp_V in}_i}{R_\mathrm{osc}}\right|}\left(1 - \exp\left(-\frac{\mathrm{d}t}{T_\mathrm{relax}}\right)\right) + \mathrm{out}_{i-1}\exp\left(-\frac{\mathrm{d}t}{T_\mathrm{relax}}\right) \quad (3.1)$$

$$\mathrm{out}_0 = p_\mathrm{A}\sqrt{\left|(I_0 - I_\mathrm{c}) - \frac{\mathrm{Amp_V in}_0}{R_\mathrm{osc}}\right|} \quad (3.2)$$

The relatively simple description of the phenomenological oscillator is also its main disadvantage. Indeed, the actual dynamics of the vortex core is far more complex, which is why the simulations do not match the physical sample in [4] (see Fig.1.26).

3.2.2 s-ODE LO-TEA oscillator

This model is the first to have been developed in the scope of the TEA. It involves the numerical resolution Eq.2.21, which can be re-expressed as Eq.2.30. The convergence of the solution is reached relatively quickly by the numerical solver thanks to the smoothness of the solution, which describes the envelope of the reduced position of the vortex core $s(t)$. The expression for $\Gamma(s(t))$ is the truncated power expansion of Eq.2.31, computed using the parameters a_J, b_J, a and b. The model is represented in Fig.3.6 and the code is available in section A.2. It allows to know the reduced position of the vortex core $s(t)$ from $t = 0$ to $t = t_\mathrm{tot}$ for a signal encoded in the input current density J_in. Note that the parameters a_J, b_J, a and b are encapsulated in the array `LOM`, and the D_t constant, representing the time separating two samples in the input signal, is used to obtain the last signal sample injected into the model at time t.

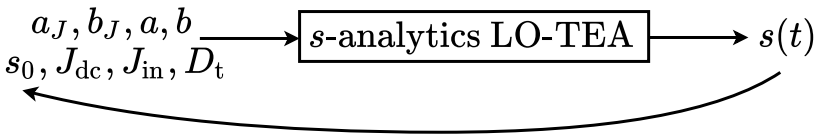

Fig. 3.7: Schematic of the s-analytics LO-TEA oscillator

3.2.3 s-analytics LO-TEA oscillator

The s-analytics LO-TEA oscillator takes advantage of the fully analytical expression for $s(t)$ derived in Eq.2.43. From now on, no more numerical solver is needed, and the dynamics of the vortex core can be obtained quickly and trivially (Fig.3.7). The simulation time t_{tot} is also directly deducted from the length of the input signal J_{in} and the time constant D_{t} separating two consecutive input samples. The model (see section A.3) hence computes $s(t = D_{\mathrm{t}})$ for each sample of the input signal. The LO-TEA parameters are also given in the `LOM` array. Note that recurrence is still needed as $s(t_i)$ is used as initial value s_0 for the computation of the next time step $s(t_{i+1})$.

3.2.4 s-analytics HP-TEA oscillator

Finally, it is possible to account for all the possibly neglected complexity due to the truncation in Eq.2.31 by using the generalized relation of Eq.2.48 and the expressions of Eq.2.49 and Eq.2.51. The model is thus similar to that of the s-analytics LO-TEA oscillator represented in Fig.3.7, and the code is available in section A.4. Note that here again, recurrence is needed to propagate the computed reduced position at a certain time $s(t_i)$ as the initial reduced position for the next time step t_{i+1}.

3.3 Operating parameters

To complete the simulations of the STVO, several parameters have to be fixed accordingly to the physical reality. These parameters are not meant to be changed often, as they correspond to a specific device.

3.3.1 DC resistance of the oscillator

The DC resistance of the oscillator corresponds to the electrical resistance of the STVO without considering the resistance variations due to the TMR effect. It is defined in Eq.3.3, with ρ the overall resistivity of the structure, L its length (here, the height H of the STVO) and d its diameter. The DC resistance is fixed to $R_{\mathrm{osc}} = 140.6\Omega$ accordingly to the literature and the diameter of the oscillator (subsection 3.3.3) [4, 15, 17].

$$R_{\mathrm{osc}} = \rho \frac{L}{\pi \left(\frac{d}{2}\right)^2} \tag{3.3}$$

3.3.2 Chirality and Ampère-Oersted field

When the DC bias current is injected into the STVO, a curling Ampère-Oersted magnetic field will be created. This Ampère-Oersted field can interact with the magnetic vortex of the free FM layer, resulting into a splitting of the dynamics of the vortex core [28]. Indeed, depending on the configuration of the setup (which depends on the chirality C of the vortex and the curling direction of the Ampère-Oersted field), the value of critical parameters such as J_{cr1} can vary. Furthermore, the consideration of the Ampère-Oersted field itself has an influence of said parameters. Hence, the 3 different cases (C^+ when the Ampère-Oersted field and the magnetic vortex are curling in the same direction, C^- when they are curling in opposite directions and noOe when the Ampère-Oersted field is neglected) lead to 3 different sets of parameters a_J, b_J, a, b. These 3 sets of parameters induce 3 distinctive dynamical regimes, so that the results yielded by the simulations can be analyzed under the prism of the splitting phenomenon at their origin. In this work, this configuration (C^+) was kept constant during all the simulations.

3.3.3 Geometry of the oscillator

The STVO is composed of a nanopillar of height H and diameter d. Only the diameter of the STVO is relevant in the scope of our simulations, as the current density is often converted into current intensity and conversely. The diameter d is hence defined accordingly to the literature and its value is fixed to 200 nm. The diameter of the STVO plays a critical role in its energy consumption, as lower current intensities are needed to sustain the transient state in the dynamics of the vortex core in a STVO with a smaller diameter.

3.3.4 Sampling rate

Finally, the sampling rate corresponds to the number of input samples injected into the STVO in 1 second. It is hence linked to the D_{t} parameter representing the elapsed time between two different input samples through Eq.3.4. This parameter may seem of lower importance, but actually its value has to be chosen properly. Indeed, if the sampling rate is too high, the oscillator won't be allowed to relax sufficiently between the consecutive inputs, which is strongly not wanted as the relaxation of the dynamics of the vortex core plays a crucial role in the non-linear treatment of the data. If the sampling rate is too low, the oscillator will reach steady-state between the consecutive inputs. This is also not wanted as the dynamics of the vortex core must stay in the transient state to allow an efficient data treatment. In our simulations, the D_{t} parameter is set to 50 nanoseconds, which corresponds to a sampling rate of 20 MSamples.

$$D_{\mathrm{t}} = \frac{1}{\text{sampling rate}} \tag{3.4}$$

3.3.5 Base working current I_{w}

The working current I_{w} is the DC bias current that is constantly injected into the oscillator to trigger the motion of the vortex core. The input signal is added to I_{w}, and the result is finally injected into the STVO. The value of I_{w} is important as it defines the range

of the oscillator dynamics that will be covered by the signal, and hence will be used to non-linearly process the data. Overall, I_{w} (or V_{w}, see Eq.3.5) can be seen as a value of current (or voltage) at the center of the range defined by the input data and the noise (black vertical line in Fig.3.8).

$$V_{\mathrm{w}} = R_{\mathrm{osc}} I_{\mathrm{w}} \tag{3.5}$$

3.3.6 Amplitude of the input signal

The signal (here, the waveforms) can be mapped on a defined voltage range around the operating voltage of Eq.3.5. For a given voltage range ΔV, the amplitude of the sine and square periods is mapped from $[-1, +1]$ in the database to $[-\Delta V/2, +\Delta V/2]$ before being processed further.

The input current I_{input} is defined by the voltage amplitude ΔV of the signal through Eq.3.6. It is represented by the red interval around the working current I_{w} in Fig.3.8.

$$I_{\mathrm{input}} = \frac{\Delta V}{2R_{\mathrm{osc}}} \tag{3.6}$$

3.3.7 Base noise level

To assess the performances of the recognition of "dirtier" inputs, additive white Gaussian noise can be added to the periods of the database. The power of the signal without noise is given by Eq.3.7 and Eq.3.8. The average power of white noise is given by Eq.3.9 and Eq.3.10, with σ the standard deviation, which is related to the quadratic mean of the amplitude of the noise in volts. Finally, the SNR is defined as Eq.3.11 and Eq.3.12. Hence, for a given $\mathrm{SNR_{dB}}$, a vector of random values normally distributed around $\mu = 0$ with a standard deviation of

$$\sigma = \sqrt{\frac{R_{\mathrm{osc}} I_{\mathrm{dc}}^2}{10^{(\mathrm{SNR_{dB}}/10)}}}$$

is generated and added to the clean signal. The voltage range $\Delta V_{\mathrm{noise}}$ related to the noise can be approximated to Eq.3.13 thanks to the 6σ rule [30]. The corresponding current I_{noise} is thus given by Eq.3.14, and is represented by the blue ranges around the red input ranges in Fig.3.8.

$$P_{\mathrm{s}} = R_{\mathrm{osc}} I_{\mathrm{dc}}^2 \quad (\mathrm{W}) \tag{3.7}$$

$$P_{\mathrm{s}} = 10 \log_{10} \left(R_{\mathrm{osc}} I_{\mathrm{dc}}^2 \right) \quad (\mathrm{dB}) \tag{3.8}$$

$$P_{\mathrm{n}} = \sigma^2 \quad (\mathrm{W}) \tag{3.9}$$

$$P_{\mathrm{n}} = 10 \log_{10} \left(\sigma^2 \right) \quad (\mathrm{dB}) \tag{3.10}$$

$$\mathrm{SNR} = \frac{R_{\mathrm{osc}} I_{\mathrm{dc}}^2}{\sigma^2} \tag{3.11}$$

$$\mathrm{SNR_{dB}} = 10 \log_{10} \left(\frac{R_{\mathrm{osc}} I_{\mathrm{dc}}^2}{\sigma^2} \right) \quad (\mathrm{dB}) \tag{3.12}$$

$$\Delta V_{\mathrm{noise}} = 6\sigma = 6 \sqrt{\frac{R_{\mathrm{osc}} I_{\mathrm{dc}}^2}{10^{(\mathrm{SNR_{dB}}/10)}}} \tag{3.13}$$

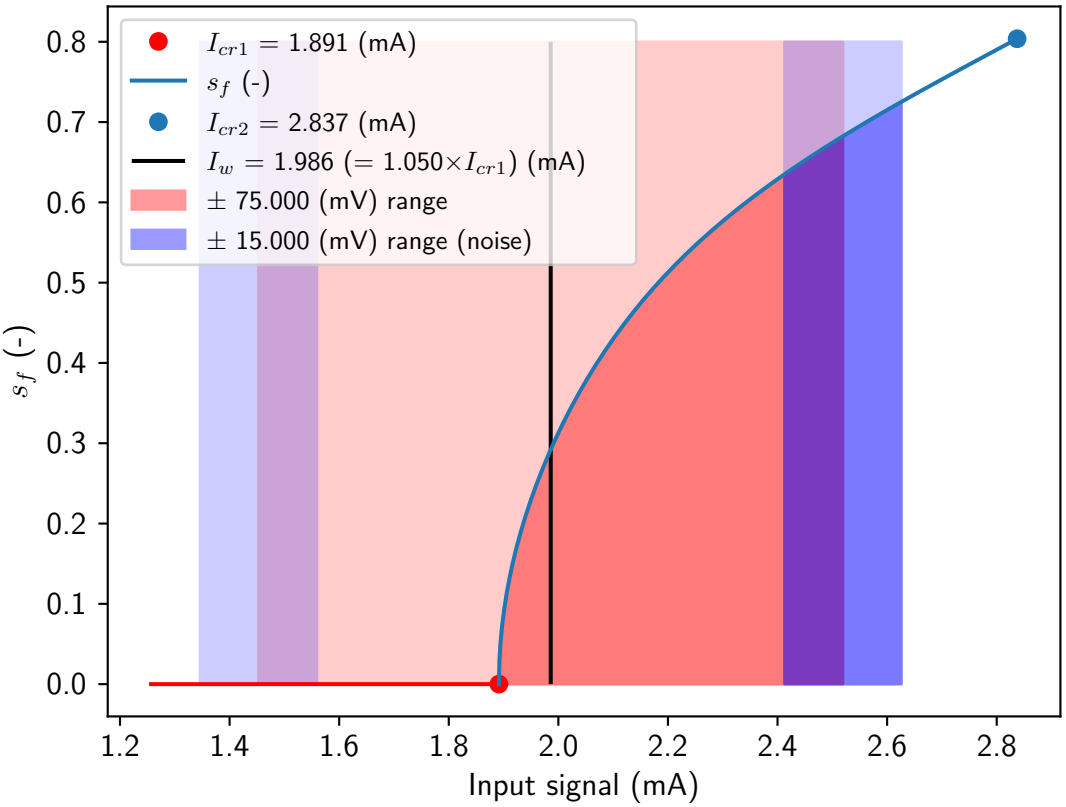

Fig. 3.8: Example of the current map and the range sounded by the dynamics of the oscillator. The black line corresponds to the working current I_{W}, the red range corresponds to a input voltage range of $\Delta V = 150$ mV, and the blue ranges correspond to an additional noise voltage range of $\Delta V_{\mathrm{noise}} = 30$ mV. The sum $V_{\mathrm{w}} + \Delta V/2 + \Delta V_{\mathrm{noise}}/2$, once converted into current, cannot exceed the upper limit defined by I_{cr2}. The curve is the steady-state reduced position of the vortex core s_{f} for each input current intensity.

$$I_{\mathrm{noise}} = \frac{\Delta V_{\mathrm{noise}}}{2R_{\mathrm{osc}}} \tag{3.14}$$

To conclude, the current related to the overall noisy signal spans on the whole colored range around the black vertical line in Fig.3.8. One must pay attention to the fact that the sum of the working current I_{W}, the current due to the input signal I_{input} (Eq.3.6) and the current due to the noise (Eq.3.14) does not exceed the limit set by I_{cr2}, which is obviously not wanted as the gyrotropic motion vortex core might be expelled under such conditions.

3.4 Parametric studies

The quick and accurate simulation of the dynamics of the STVO allows to emulate an entire NN through the single node reservoir computing approach. This way, the performances of the NN can be assessed for the resolution of a simple benchmarking classification task. Moreover, the influence of specific operating parameters on the success rate of the procedure can be observed. In this section, the *s*-ODE LO-TEA, *s*-analytics LO-TEA and *s*-analytics HP-TEA models are used to perform parametric studies concerning two important parameters : the working (bias) DC current I_{w} and the noise level of the input signal.

3.4.1 Working current I_{w}

The working current I_{w} was swept from $(1.05 \times I_{\mathrm{cr1}})$ to I_{cr2}. After having added the input signal and the base noise level, the performances of the NN were recorded. Note that during the I_{w} parametric sweep, the SNR isn't kept constant as the noise originates from external factors that are independent of the bias current. The average noise voltage range ($= 3\sigma$ following the 6 sigma rule) is rather kept constant, leading to a decrease of the SNR as the bias current increases.

Thus, as the quality of the signal is improving, it could be expected that the performances of the NN increase as the DC bias current intensity increases. However, this increase may also be at the expense of the performances in the classification task, as the range of the non-linearity sounded by the signal may be less favorable to an efficient non-linear treatment with higher I_{w} intensities.

3.4.2 Noise level

Parametric sweeps of the signal-to-noise ratio were also carried out, using the expression of σ developed in subsection 3.3.7. The peak-to-peak ratio between the signal and the noise is given by Eq.3.15. Several levels of noise (in blue) and the resulting signals (in orange) are displayed from Fig.3.9 to Fig.3.14. It can be seen that for positive SNRs (from Fig.3.9 to Fig.3.11), the difference between the sine and the square waveforms remains possible for the human eye. At 0 dB (Fig.3.12), the power of the noise is equal to that of the signal, making it impossible to make the difference between the two periods. With decreasing SNRs (from Fig.3.13 to Fig.3.14), the noise eventually saturates the original signal, turning it into a random resulting signal of high amplitude. The peak-to-peak ratio, given by Eq.3.15, indicates the ratio between the maximum amplitude of the noise and the maximum amplitude of the signal.

$$\text{p-2-p ratio} = \frac{\max(\text{noise}) - \min(\text{noise})}{\max(\text{signal}) - \min(\text{signal})} \tag{3.15}$$

3.5 Recognition

The recognition task is represented in Fig.3.16. After the sweep range of the variable of interest has been defined, the corresponding input signals are generated using the alternate database. Then, each input signal (corresponding to one distinct value of the sweep range) is randomly masked to simulate the single node reservoir computing approach, and is injected in one of the three most efficient models defined in section 3.2 (s-ODE LO-TEA, s-analytics LO-TEA or s-analytics HP-TEA). Afterwards, the performances of the NN are recorded. This is done 200 times for every single value of the sweep range to ensure the canceling of random irregularities due to the noise added to the signal. Finally, the mean of performances is performed over the 200 results, allows to eventually generate a plot of the average performances of the NN all along the sweep.

In the present case, a sine period corresponds to the scalar output 0 and a square period corresponds to the scalar output 1. Thus, each estimation of the NN that is < 0.5

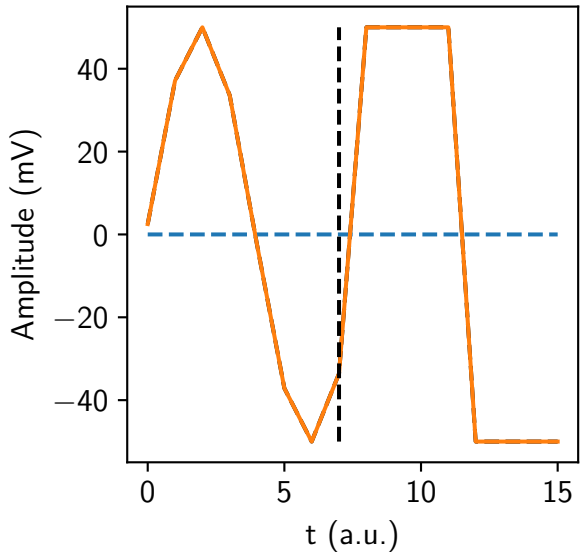

Fig. 3.9: $\mathrm{SNR_{dB}} = 100$ dB, p-2-p ratio $= 1.79 \times 10^{-5}$

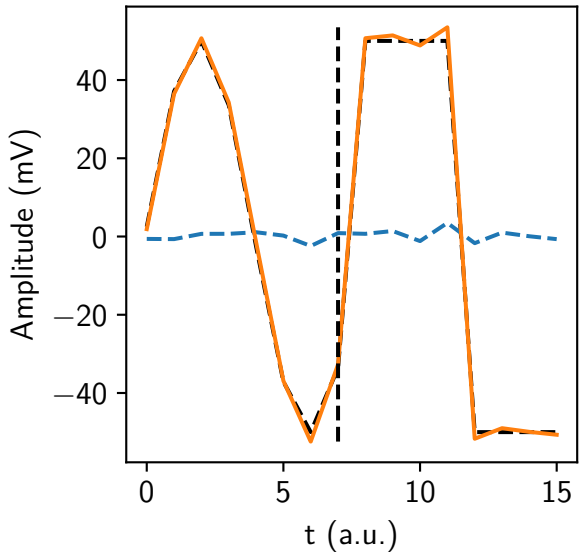

Fig. 3.10: $\mathrm{SNR_{dB}} = 30$ dB, p-2-p ratio $= 0.059$

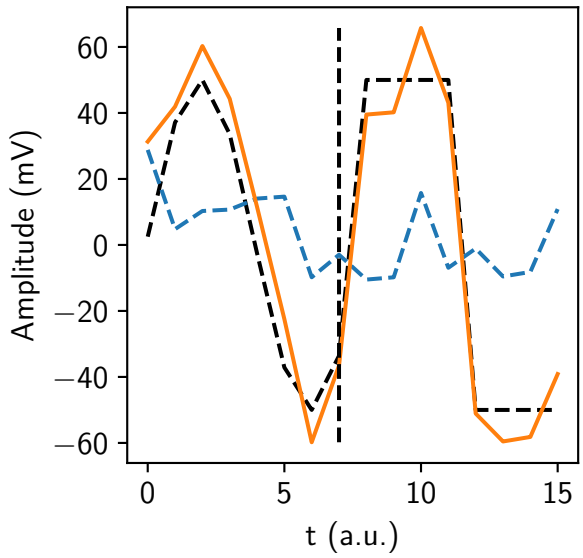

Fig. 3.11: $\mathrm{SNR_{dB}} = 10$ dB, p-2-p ratio $= 0.39$

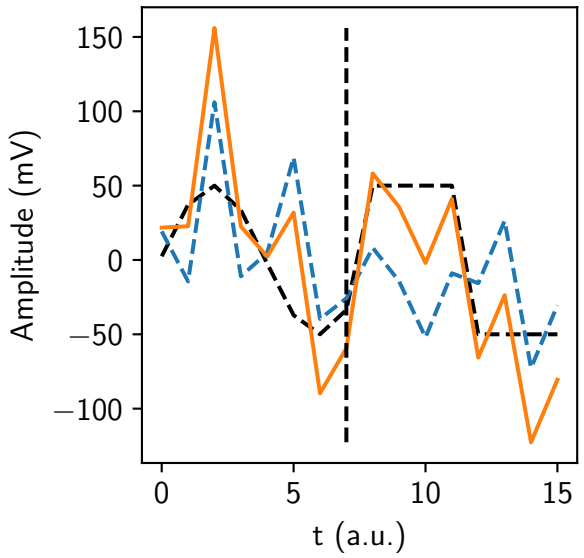

Fig. 3.12: $\mathrm{SNR_{dB}} = 0$ dB, p-2-p ratio $= 1.77$

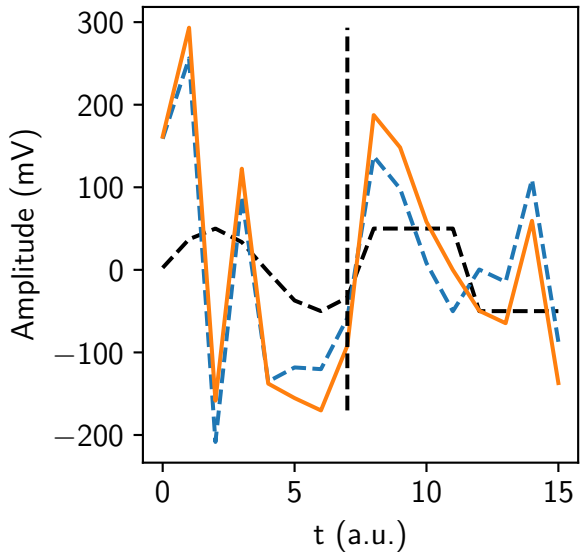

Fig. 3.13: $\mathrm{SNR_{dB}} = -10$ dB, p-2-p ratio $= 4.65$

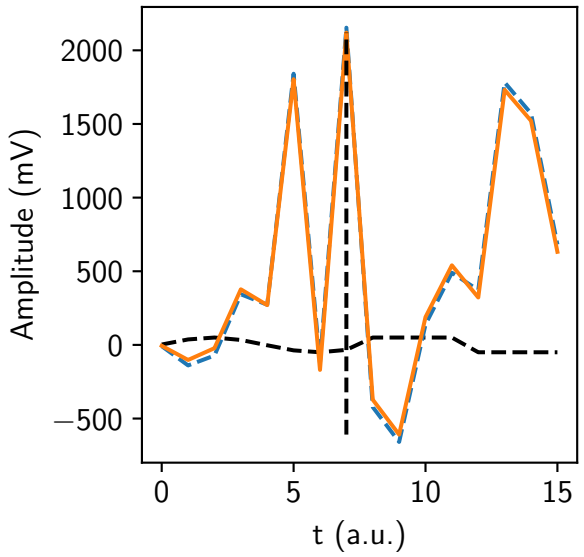

Fig. 3.14: $\mathrm{SNR_{dB}} = -30$ dB, p-2-p ratio $= 28.13$

Fig. 3.15

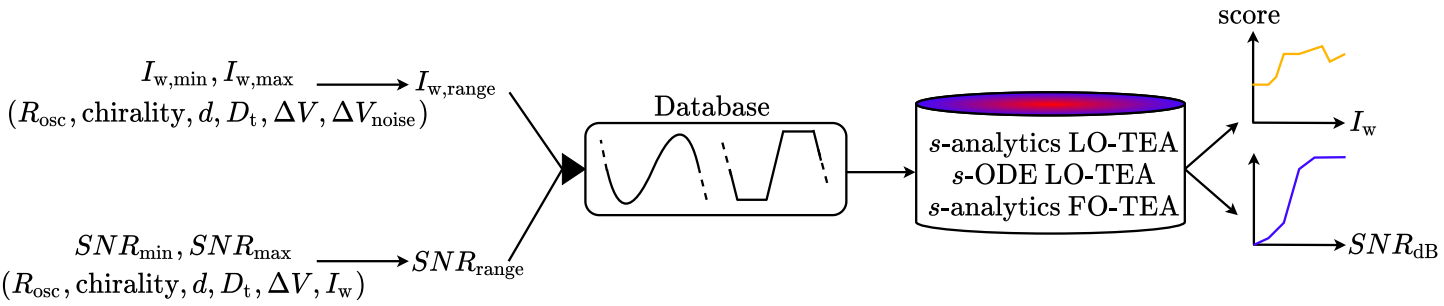

Fig. 3.16: Schematic of the parametric sweeps in DC bias current intensity and noise level

is considered as a sine, and each estimation that is > 0.5 is considered as a square. In the extremely rare (yet probable) case where the estimated output is perfectly 0.5, the corresponding result is chosen randomly between 0 and 1 (this case corresponds to a perfect indecision of the NN, a situation that is probable to arise at high noise levels).

Speaking of performances, two quantities can actually be studied to understand the ability of the NN to classify and recognize the different waveforms :

1. **Success rate (score %)** : This corresponds to the ratio between the number of signals that have been correctly recognized by the neural network and the total number of signals. Here again, two categories can be distinguished : the τ-wise approach, where each signal is split into 8 independent samples and the score for the whole signal is computed using the result of the recognition for each of these samples (Eq.1.14 and Eq.1.15), and the WTA approach, where the estimation for a whole signal is obtained from the average of the estimation for each of the samples of the signal (Eq.1.16).

2. **Root-Mean-Square (-)** : The RMS is a well-known quantitative indicator corresponding to the RMS of the distance between the estimated target (before applying the argmax() function) and the expected target. It offers the advantage to understand how "sure" the NN is about its predictions. Indeed, under given operating conditions, a sine period can be correctly recognized to 100% using the TW or WTA approach, but the RMS of the estimation can be higher than in other conditions. This can be understood as the fact that in this specific case, the NN is somewhat less confident about its prediction. The RMS value for a signal σ is computed as in Eq.3.16 in the TW approach and as in Eq.3.17 in the WTA approach. Thanks to the average introduced in Eq.3.17, the RMS is expected to be smaller for the WTA approach than for the TW approach.

$$\mathrm{RMS_{TW}} = \sqrt{\frac{\sum_{\tau} \left(\mathbf{t}_{\sigma} - \hat{\mathbf{t}}_{\sigma,\tau}\right)^2}{N_{\tau}}} \tag{3.16}$$

$$\mathrm{RMS_{WTA}} = \sqrt{\left(\mathbf{t}_{\sigma} - \frac{\sum_{\tau} \hat{\mathbf{t}}_{\sigma,\tau}}{N_{\tau}}\right)^2} \tag{3.17}$$

Each of these quantitative quality estimators is hence used when performing the parametric sweeps in order to have the best insight as possible about the influence of said parameters on the performances of the neural network.

Summary

Using the different models developed in the previous chapter, different approaches can be used to simulate a STVO, accurately and efficiently. To assess the ability of each of these models, but also to get a better insight of how the dynamics of a STVO can be influenced, a benchmarking task consisting in classifying sine and square waveforms is developed. Different types of databases were generated. Then, a STVO was modeled using specific operating parameters in accordance with the literature and the experimental reality. The dynamics of the vortex core was simulated using some of the previously developed models, at choice. Two kinds of parametric studies were designed : one corresponding to a sweep of the DC bias current injected into the STVO, and another corresponding to the level of noise to be added on the signal. The resulting performances of the neural network were assessed using two quantitative indicators : the success rate and the RMS.

Chapter 4

Comparison between the different models of oscillators

The three most developed models describing the dynamics of the vortex core of a STVO (s-ODE LO-TEA, s-analytics LO-TEA and s-analytics HP-TEA) are compared in the light of their average performances in the benchmarking task.

4.1 s-ODE LO-TEA vs. s-analytics LO-TEA

The s-ODE LO-TEA and the s-analytics LO-TEA are first compared. To do so, the results yielded by the two models for the computation of keys quantities are compared. Then, the time required to perform the benchmark task are compared.

4.1.1 Computation of the position of the vortex core $s(t)$

First, the envelope of the reduced position of the vortex core s is computed for the same input signal, using the s-ODE LO-TEA model (see Fig.3.6) and the s-analytics LO-TEA model (see Fig.3.7). The injected signal input signal J_{in} is a linear range from I_{cr1} to $I_{\text{cr2}} = 1.5I_{\text{cr1}}$ (see Fig.4.1). The corresponding reduced position is displayed in Fig.4.2. It can be directly seen that the computed reduced position is exactly the same over the whole length of the signal, for the two models. This is also true for any random input signal, meaning that the results yielded by the two models are perfectly consistent. As the reduced position is the quantity of interest that is used to perform the recognition, one can expect that the simulated artificial intelligence will yield the same results with both models.

4.1.2 Required simulation time

Despite yielding identical results, the two models differ by the time required to simulate the dynamics of the vortex core. The benchmarking task is thus performed using the two models, and the elapsed time is recorded. To ensure consistent results for the required simulation time, this procedure is repeated 200 times and the mean of the required time is performed for both two models. It also ensures that the other results (score and RMS) are averaged over a large number of simulations to ensure relevant comparison between both

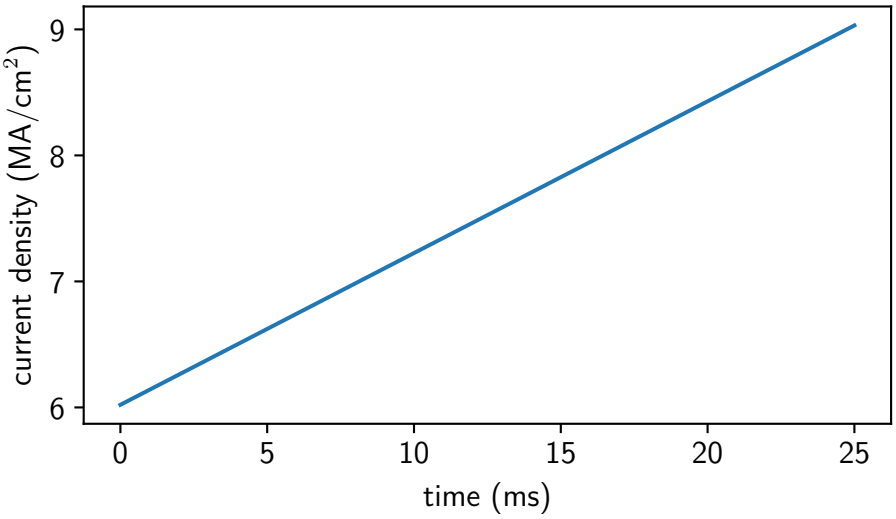

Fig. 4.1: Test input signal injected into the s-ODE LO-TEA model and into the s-analytics LO-TEA model. The sample are spaced in time by 25 ns, and the signal lasts 25 ms.

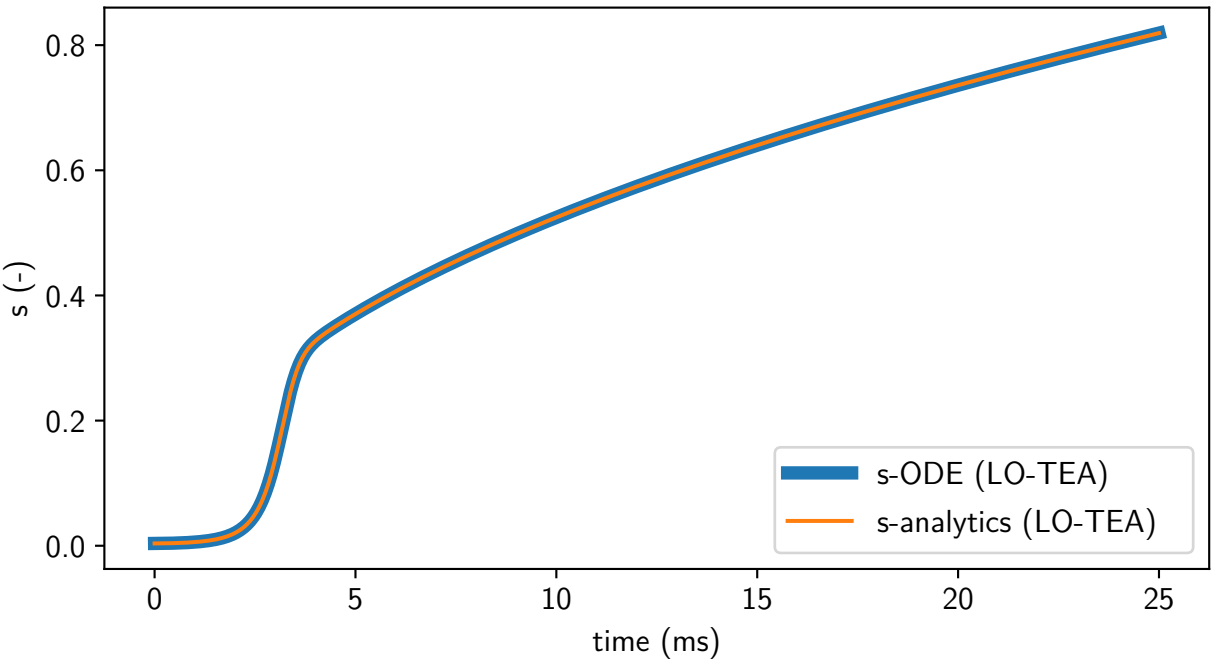

Fig. 4.2: Simulated reduced position s of the vortex core computed using the s-ODE LO-TEA and s-analytics LO-TEA models

R_{osc}	140.6 Ω
Chirality	C^+
D_{t}	50 ns
ΔV	150 mV
d	200 nm
I_{w}	1.986 mA
ΔV_{noise}	50 mV

Tab. 4.1: Input parameters for the assessment of the required computation time of the *s*-ODE LO-TEA and the *s*-analytics LO-TEA models

		***s*-ODE LO-TEA**	***s*-analytics LO-TEA**
Training	Score (WTA)	100%	100%
	Score (TW)	99.77%	99.76%
	RMS (WTA)	0.236	0.235
	RMS (TW)	0.350	0.349
Testing	Score (WTA)	99.89%	99.81%
	Score (TW)	99.34%	99.23%
	RMS (WTA)	0.269	0.285
	RMS (TW)	0.390	0.407
Average elapsed time		62.517 s	0.735 s

Tab. 4.2: Results of the benchmarking task for the *s*-ODE LO-TEA and the *s*-analytics LO-TEA models

models. Indeed, a certain level of noise is added to the signal, which add a corresponding level of statistical randomness to the results of a single simulation. The input parameters are presented in Tab.4.1, and the results are gathered in Tab.4.2.

Concerning the results of the benchmarking task, it can be seen that for the training phase, the results are practically identical for both models. The differences are slightly more pronounced for the testing phase, but stay very low : at most 0.11% for the score and 0.017 for the RMS. Given the perfect consistency between the two models observed at subsection 4.1.1, one could suspect these differences to be due to the statistical randomness introduced by the noise.

A striking difference can however be observed at the last row in Tab.4.2 : the average elapsed time needed to perform the benchmarking task. It is roughly 85 **times longer for the *s*-ODE LO-TEA than for the *s*-analytics LO-TEA model**, highlighting the most advantageous feature of the analytical model.

4.2 *s*-analytics LO-TEA vs. *s*-analytics HP-TEA

As seen previously, the *s*-analytics LO-TEA model can be considered as an optimum concerning the required simulation time, while yielding results identical to those that could be obtained with the *s*-ODE LO-TEA model, which involves the standard numerical resolution of the ODE from Eq.2.21. However, one must remember that both of these models express $\Gamma(s)$ as a truncation to the second order of its power expansion. It is hence

important to assess the influence of the consideration or the negligence of the higher-order terms in the expansion of $\Gamma(s)$. As a matter of fact, only the performances of the simulated neural network can vary between the two corresponding models of interest (s-analytics LO-TEA and s-analytics HP-TEA). Indeed, as both models are based on an analytical evaluation of the reduced position of the vortex core (Eq.2.43 and Eq.2.48), the required simulation time is expected to be roughly the same for the two approaches. However, the higher-order terms in the expression of $\Gamma(s)$ are expected to have a non-negligible influence on the dynamics of the simulated STVO.

First, the same input signal as in Fig.4.1 is injected into two oscillators, simulated using the s-analytics LO-TEA and the s-analytics HP-TEA models. The computed output signals are represented in Fig.4.3. As expected, a non-negligible difference is noticeable between the two curves, due to the accounting of the higher-order terms in the s-analytics HP-TEA model. Here, the dynamics is ruled by both time and current density, and the current density increases linearly with time in the injected signal. The maximum relative difference is at $t = 3.43$ ms, and is equal to 14.9%.

It is also possible to observe the same difference between the two models without involving any dependence on time, by plotting the steady-state reduced position of the vortex core $s_\mathrm{f} = s(t = \infty)$ for the same input signal, following the relation of Eq.2.46 and Eq.2.50. In this case, the steady-state position is plotted for each input value independently of the previous values, and the input signal of Fig.4.1 is treated as in a sweep of the bias current density (as the current density increases linearly over time in that signal). In Fig.4.4, one can observe that the relative difference between the two models stays reasonable; it does not exceed 3.95% at 6.816 MA/cm^2. One can hence conclude that in the first case (Fig.4.3), the differences between the two models are propagated and amplified in time due to the time-dependence introduced by the recurrence of the solutions from Eq.2.43 and Eq.2.48 (see model at Fig.3.7).

To assess the influence of these differences on the actual performances of the simulated neural network, the benchmarking task is performed 1000 times, and the results are averaged over all the executions. This high number of runs is made possible in a reasonable amount of time thanks to the short required computation time for both models. Once again, this proves the efficiency of the analytical approaches for obtaining significant results that can be compared without worrying about the possible differences due to the noise introduced in the input signal. The input parameters are the same as in Tab.4.1, and the obtained results are presented in Tab.4.3. Concerning the averaged elapsed time, it can be seen that the two models are practically equally fast, as expected. Concerning the performances at the benchmarking task, one can observe that in most of the cases, the s-analytics HP-TEA model yields slightly better results than the s-analytics LO-TEA model (higher recognition scores and lower RMS values). Given the high number of runs, these differences are quite unlikely to be due to the noise in the input signal, but rather to the models themselves, and to the higher complexity of the representation of the dynamics of the vortex core in the HP-TEA model. The s-analytics HP-TEA model, in addition to being the most accurate so far, thus also yields the better results while performing the benchmarking task.

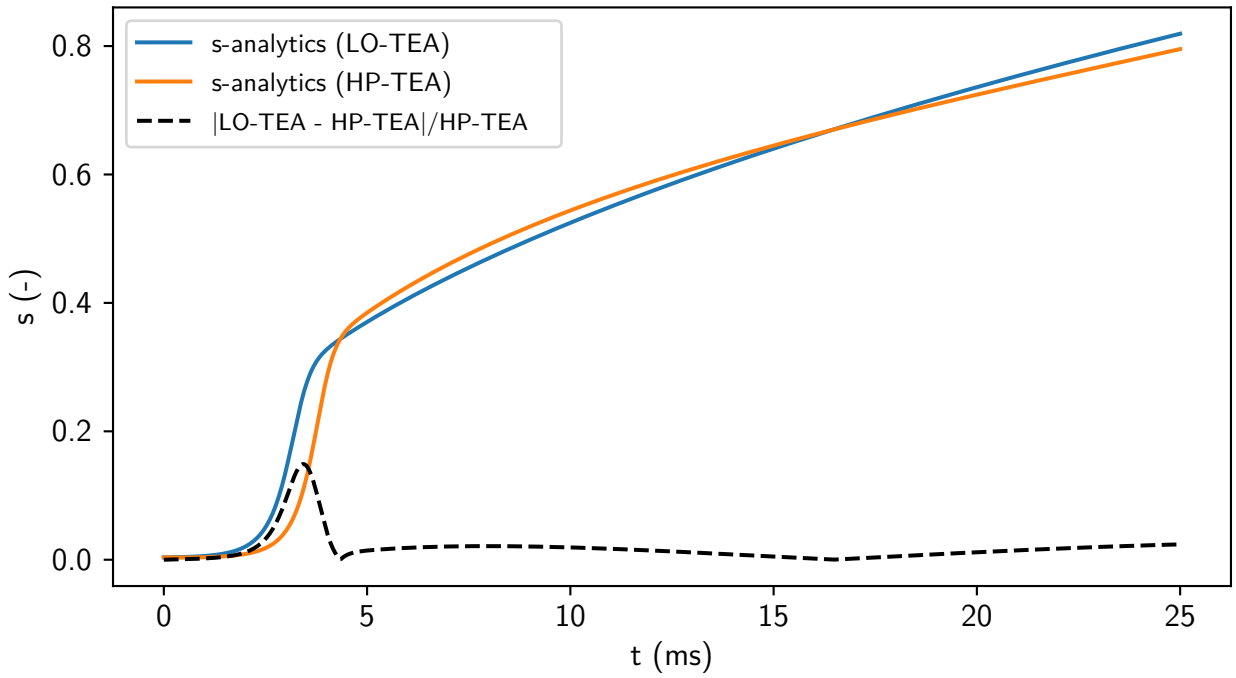

Fig. 4.3: Simulated reduced position s of the vortex core computed using the s-analytics LO-TEA and the s-analytics HP-TEA models for the input signal in Fig.4.1 and the input parameters in Tab.4.1

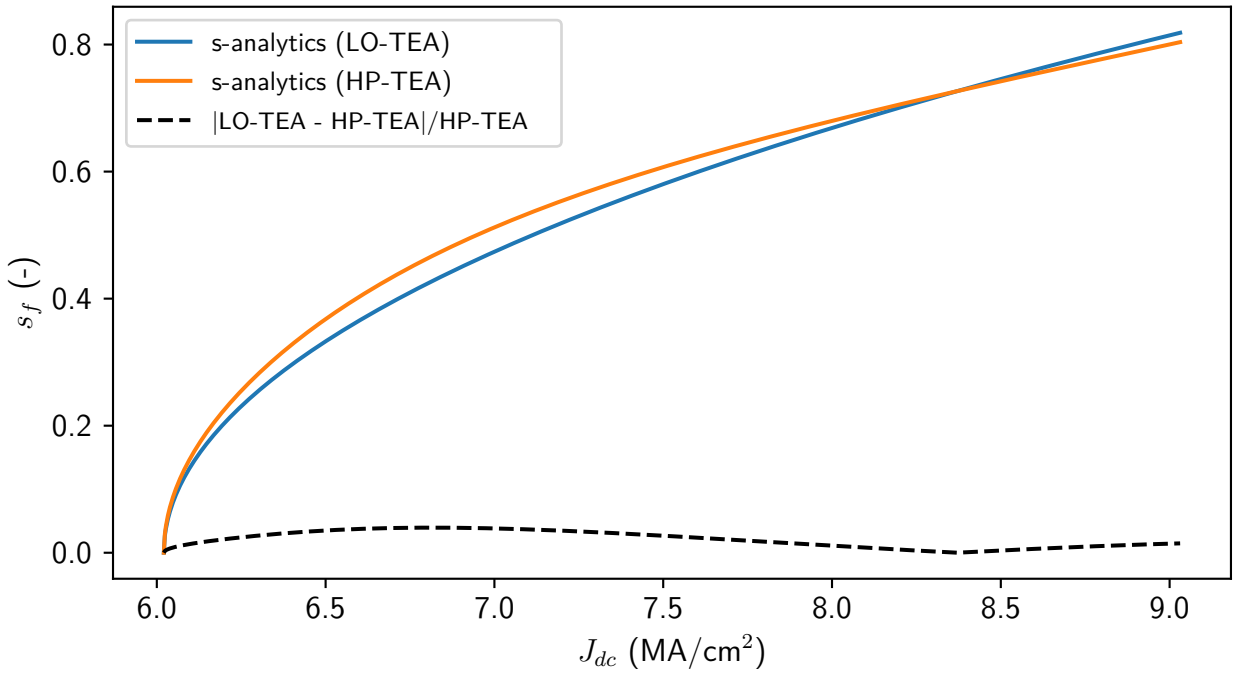

Fig. 4.4: Simulated steady-state reduced position s_f of the vortex core computed using the s-analytics LO-TEA and the s-analytics HP-TEA models

		s-analytics LO-TEA	*s*-analytics HP-TEA
Training	Score (WTA)	99.99%	100%
	Score (TW)	99.77%	99.94%
	RMS (WTA)	0.235	0.198
	RMS (TW)	0.350	0.300
Testing	Score (WTA)	99.82%	99.73%
	Score (TW)	99.26%	99.39%
	RMS (WTA)	0.278	0.245
	RMS (TW)	0.402	0.348
Average elapsed time		0.684 s	0.713 s

Tab. 4.3: Results of the benchmarking task for the *s*-analytics LO-TEA and the *s*-analytics HP-TEA models

Summary

First, the *s*-ODE LO-TEA and the *s*-analytics LO-TEA models are compared. It is shown that the two models yields highly consistent results (to not say identical) when computing the output of the simulated STVO for the same input signal. However, the *s*-analytics LO-TEA allows to obtain the results about 85 times faster than with the *s*-ODE LO-TEA model.
Then, the *s*-analytics LO-TEA and the *s*-analytics HP-TEA models are compared in order to assess the impact of the higher-order complexity considered in the latter on the performances of the NN when performing the benchmarking task. It is shown that the results differ significantly, due to the difference of complexity between the two models. Furthermore, the NN simulated using the *s*-analytics HP-TEA model is shown to be more efficient than the *s*-analytics LO-TEA one in most of the cases. As expected, the computation time required by the two models is similar, as these are both analytical models.

Chapter 5

Impact of the working current intensity and the noise level on the LO-TEA and HP-TEA models

Thanks to the fast completion of the benchmarking task allowed by the two analytical models, a sweep of intensity of the DC bias current and of the noise level can be carried out, in order to observe the dependency of the performances of the NN (recognition score and RMS) on the swept variable.

5.1 Influence of the working current

The parametric sweep of the working current intensity I_{w} is performed using the parameters gathered in Tab.5.1, and the working current is swept from $I_{\mathrm{w,min}} = 1.001 I_{\mathrm{cr1}}$ to I_{cr2}. As the input voltage and the voltage due to the noise are fixed, the sweep is stopped as soon as the sum of the different contributions (DC bias current intensity, input current intensity related to ΔV and noise current intensity related to $\Delta V_{\mathrm{noise}}$) exceeds I_{cr2}. Then, the recognition scores and the RMS values are averaged over all the signals of the *testing* database, and are plotted at the corresponding working current intensity. To get rid of the random fluctuations introduced by the noise, the sweep is performed 200 times and the different plots are averaged. The results are presented in Fig.5.1 and Fig.5.2 for the *s*-analytics LO-TEA model and in Fig.5.3 and Fig.5.4 for the *s*-analytics HP-TEA model.

The first thing that can be noticed is that with both models, the WTA approach

R_{osc}	140.6 Ω
Chirality	C^+
D_{t}	50 ns
ΔV	150 mV
d	200 nm
$\Delta V_{\mathrm{noise}}$	50 mV

Tab. 5.1: Input parameters for the working current parametric sweep

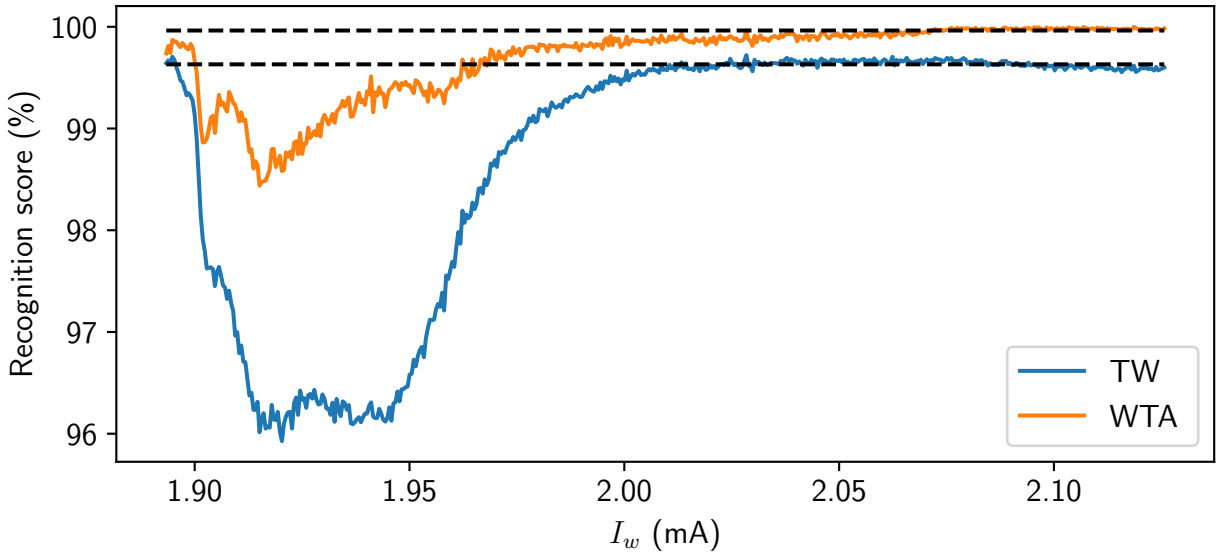

Fig. 5.1: Recognition score of the neural network emulated by a STVO simulated using the *s*-analytics LO-TEA model during a parametric sweep of the working current intensity, averaged over 200 simulations. The asymptotic values are 99.963% for the WTA approach and 99.631% for the TW approach.

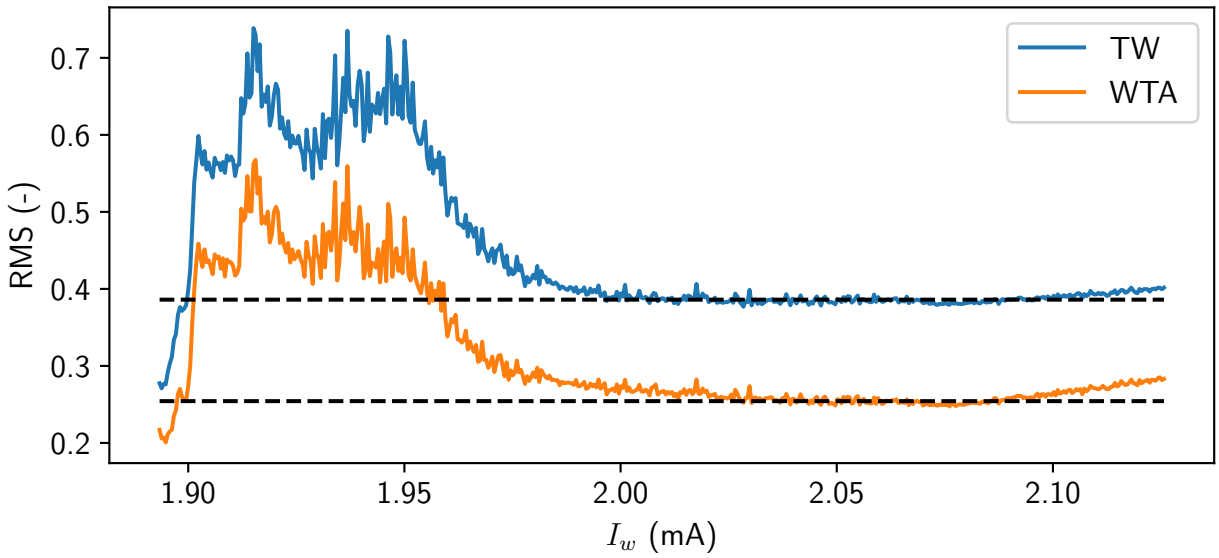

Fig. 5.2: RMS of the recognition of the neural network emulated by a STVO simulated using the *s*-analytics LO-TEA model during a parametric sweep of the working current intensity, averaged over 200 simulations. The asymptotic values are 0.254 for the WTA approach and 0.386 for the TW approach.

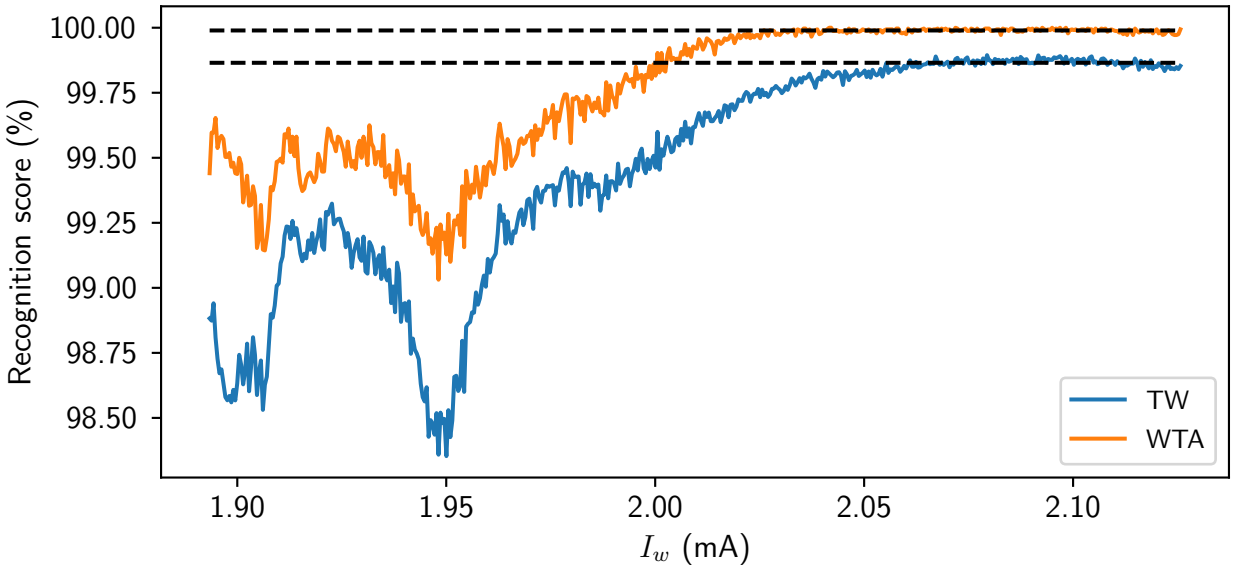

Fig. 5.3: Recognition score of the neural network emulated by a STVO simulated using the *s*-analytics HP-TEA model during a parametric sweep of the working current intensity, averaged over 200 simulations. The asymptotic values are 99.989% for the WTA approach and 99.865% for the TW approach.

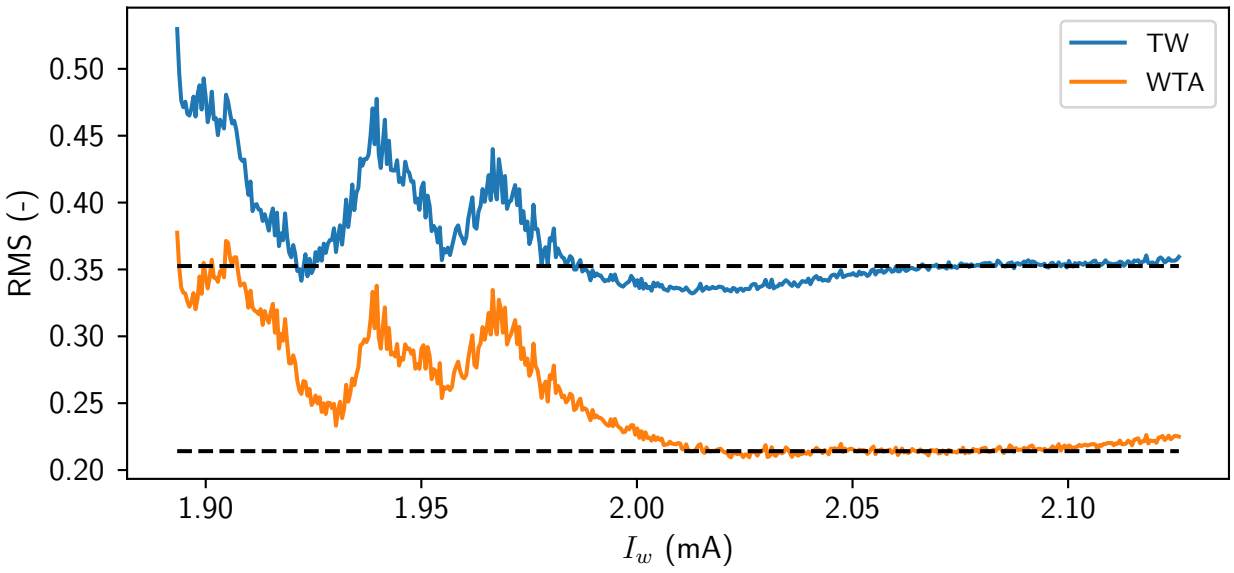

Fig. 5.4: RMS of the recognition of the neural network emulated by a STVO simulated using the *s*-analytics HP-TEA model during a parametric sweep of the working current intensity, averaged over 200 simulations. The asymptotic values are 0.214 for the WTA approach and 0.352 for the TW approach.

		s-analytics LO-TEA	*s*-analytics HP-TEA
Recognition score	WTA	99.963%	99.989%
	TW	99.631%	99.865%
RMS	WTA	0.254	0.214
	TW	0.386	0.352

Tab. 5.2: Asymptotical values at high working current intensities obtained with the *s*-analytics LO-TEA and the *s*-analytics HP-TEA models

systematically yields higher recognition scores and lower RMS values than the TW approach. As explained earlier, the higher score obtained with the WTA approach is due to the involved averaging process that absorbs the small errors (see Fig.1.25). This explanation is also valid for the RMS values, given the beneficial averaging over the whole signal introduced in the RMS computation for the WTA approach (see Eq.3.17).

The second thing to note is the highly similar aspect of the WTA and TW curves during the same study. Except in Fig.5.1, it looks like the two curves are about simply y-shifted. This implies that in most of the cases, the performances for the two approaches (WTA and TW) are the same in average, up to a constant.

It can also be observed that the performances of the emulated neural network eventually reach an asymptotic value at high current intensities, no matter the concerned indicator (recognition score or RMS) or the model used. The recognition score reaches an upper asymptotic value, while the RMS reaches a lower asymptotic value, both after $I_{\mathrm{w}} =\sim 2.0$ mA. This is the case for both models and both approaches (even though higher scores and lower RMS values can be observed at current intensities extremely close to I_{cr1} for the LO-TEA model). These asymptotic values are represented by dashed black horizontal lines in the figures mentioned before. In Fig.5.2 and Fig.5.4, a small divergence from this asymptotical value can also be noticed at the very end of the sweep.

These asymptotical values can be used for the comparison of the efficiency between the different models. Indeed, it can be observed that the asymptotical recognition score obtained with the *s*-analytics LO-TEA model is lower than that obtained with the *s*-analytics HP-TEA model. In the same way, the asymptotical RMS values obtained with the *s*-analytics LO-TEA model are higher than that obtained with the *s*-analytics HP-TEA model. The concerned values, gathered in Tab.5.2, confirm the earlier observations of the better performances of the *s*-analytics HP-TEA model (see Tab.5.2).

This improvement of the performances at higher working current intensities can be due to the increasing SNR. Indeed, as the additional voltage range $\Delta V_{\mathrm{noise}}$ is kept constant during the sweep, the SNR will increase as the intensity of the signal increases, leading to a signal of better quality, and easier to recognize. In order to investigate this assumption, the same sweep is performed by keeping the SNR constant (at higher working current intensities, the noise level will be accordingly higher too). Note that this case is not what happens in reality ; the voltage range due to noise is indeed expected to stay constant during the sweep, and the SNR is thus expected to increase. The results are represented in Fig.5.5 and Fig.5.6 for the *s*-analytics LO-TEA model, and in Fig.5.7 and Fig.5.8 for the *s*-analytics HP-TEA model. The obtained plots are extremely similar to that obtained with a variable SNR (Figures 5.1, 5.2, 5.3 and 5.4). However, several differences are worth to be mentioned.

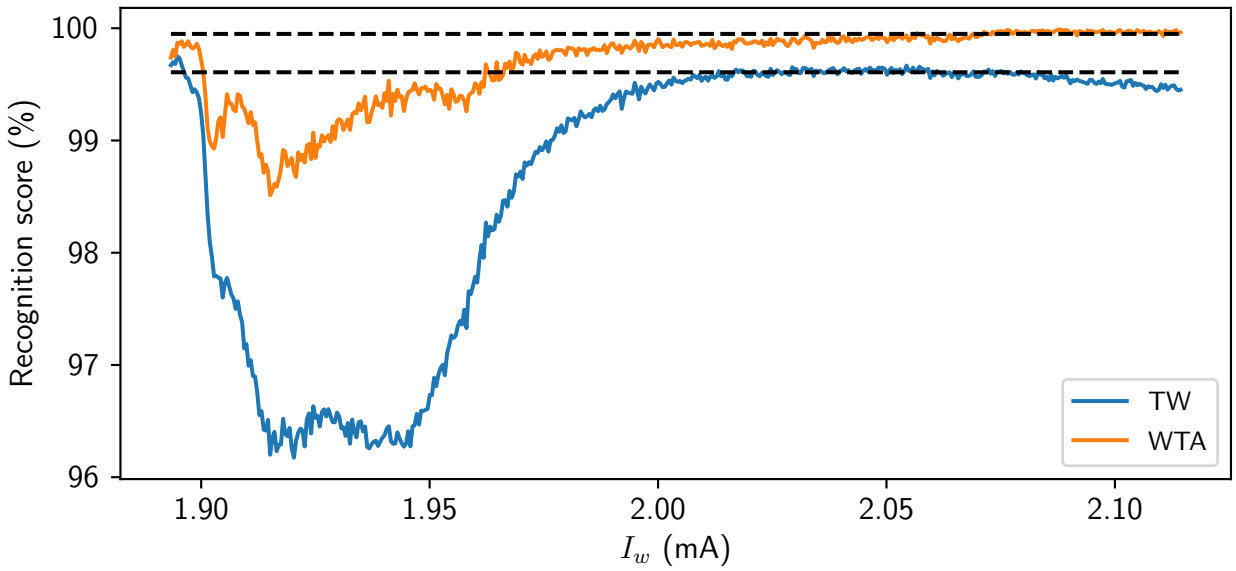

Fig. 5.5: Recognition score of the neural network emulated by a STVO simulated using the s-analytics LO-TEA model, during a parametric sweep of the working current intensity with constant SNR, averaged over 200 simulations. The asymptotic values are 99.950% for the WTA approach and 99.608% for the TW approach.

First, the constant SNR does not allow to get results as good as previously during the sweep. Indeed, the asymptotic results obtained with the constant SNR are slightly worse than with an increasing SNR (lower recognition scores and higher RMS values). This was expected, but the difference between the two cases is so small that the increasing SNR can't be took as responsible for the improvement of the performances at higher current intensities when the level of noise is fixed (in Figures 5.1, 5.2, 5.3 and 5.4). Indeed, with the constant SNR, one can see that the performances still improve at higher current intensities (increase of the recognition score and decrease of the RMS in Figures 5.5, 5.6, 5.7 and 5.8). That means that the shape of the evolution of the performances in the benchmarking task with respect to the working current intensity is due to the intrinsic dynamics of the simulated STVO. Furthermore, the same divergence phenomenon can be seen on the constant SNR plots, meaning that this effect is also probably due to the behavior of the STVO.

One can conclude that higher working current intensities are beneficial for the resolution of the benchmarking task. As higher I_{w} intensities correspond to higher reduced positions s of the vortex core, it implies that larger s allow higher recognition levels. This parametric studies, which would not have been possible in a reasonable time without the use of analytical models, thus allows to direct the further experiments using a physical STVO in the best direction concerning the operating conditions, i.e a working current intensity between 2.05 mA and 2.10 mA (for the same parameters as in Tab.5.1).

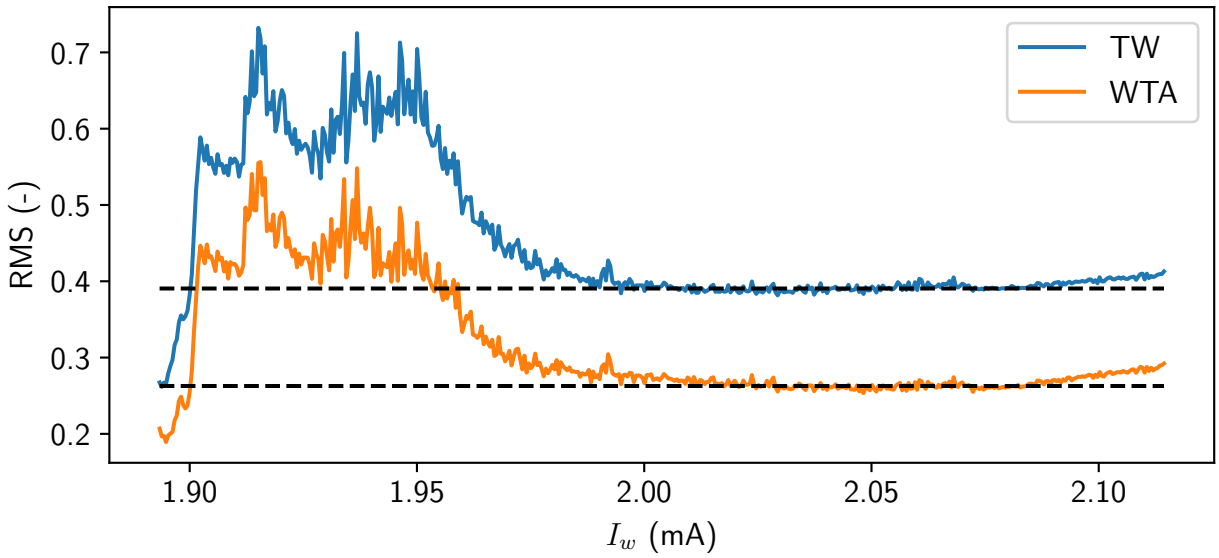

Fig. 5.6: RMS of the neural network emulated by a STVO simulated using the *s*-analytics LO-TEA model, during a parametric sweep of the working current intensity with constant SNR, averaged over 200 simulations. The asymptotic values are 0.263 for the WTA approach and 0.391 for the TW approach.

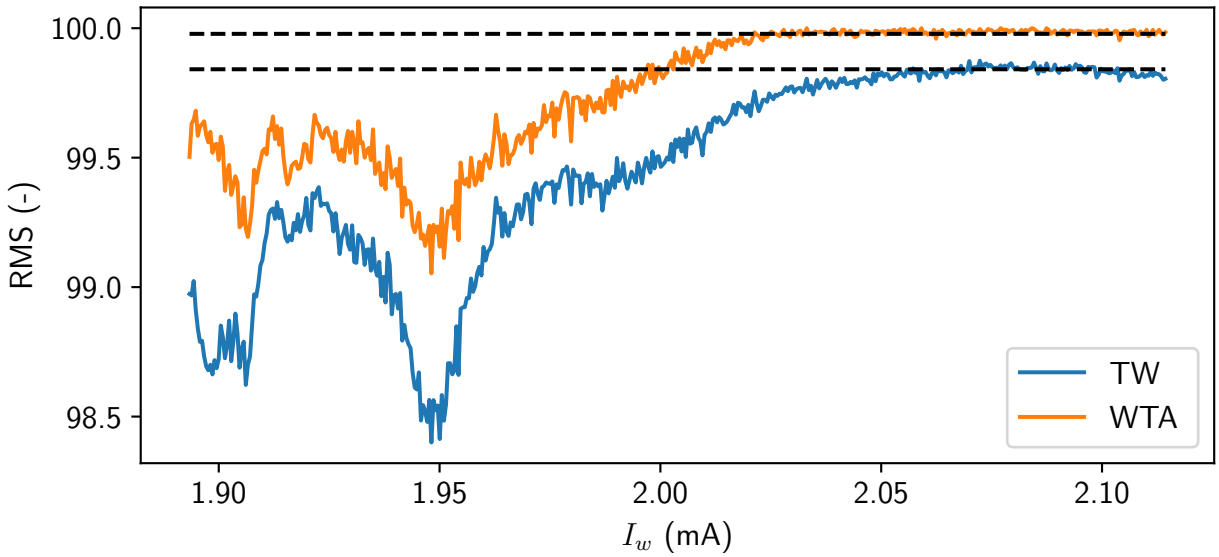

Fig. 5.7: Recognition score of the neural network emulated by a STVO simulated using the *s*-analytics HP-TEA model, during a parametric sweep of the working current intensity with constant SNR, averaged over 200 simulations. The asymptotic values are 99.978% for the WTA approach and 99.841% for the TW approach.

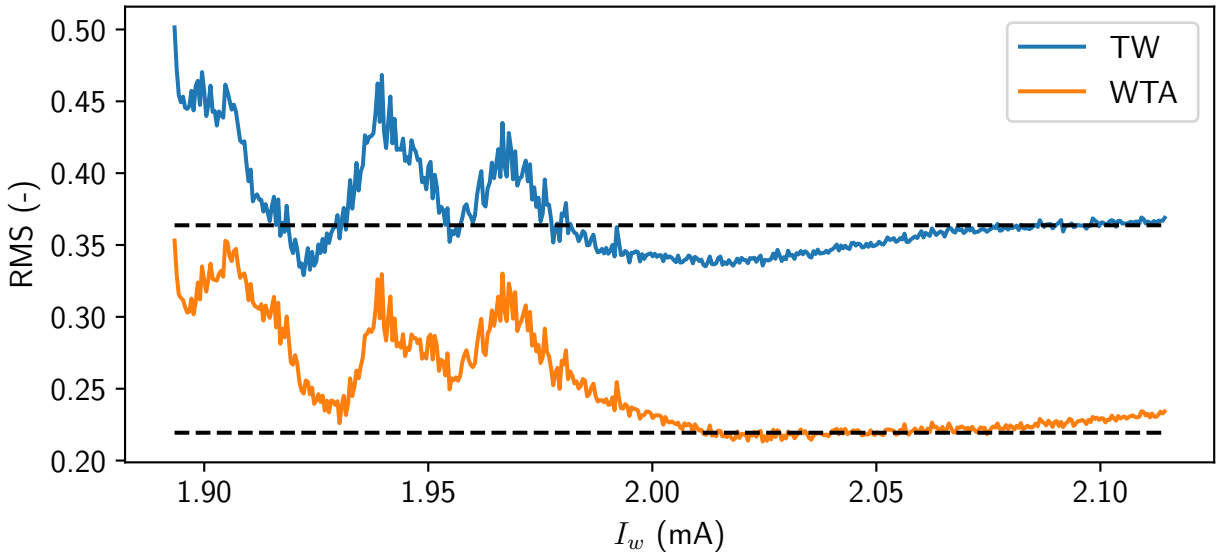

Fig. 5.8: RMS of the neural network emulated by a STVO simulated using the *s*-analytics HP-TEA model, during a parametric sweep of the working current intensity with constant SNR, averaged over 200 simulations. The asymptotic values are 0.219 for the WTA approach and 0.364 for the TW approach.

5.2 Influence of the noise

A parametric study on the noise level was also performed in order to assess the performances in the benchmarking task under varying levels of noise. Indeed, noise is inherent to physical experiments and real life applications. To do so, the performances (recognition score and RMS values) were recorded for signal-to-noise ratios ranging from -100 dB to 100 dB, and with the parameters from Tab.5.1 (the ΔV_{noise} being replaced here by the working current intensity $I_{\text{w}} = 1.05 I_{\text{cr1}} = 1.986$ mA).

An important thing to note is that for negative SNRs (where the average power of the noise is higher than that of the signal), the amplitude of the signal becomes unpredictable. This can have serious consequences on the dynamics of the vortex core. Indeed, for negatives SNRs, high noise amplitudes become probable. Under current intensities exceeding I_{cr2}, the vortex state is expelled, leading to the restoring of an uniform magnetization in the STVO. In the models, this is translated by inconsistent values or errors. Those unwanted values were obviously discarded during the sweep. Hence, to achieve an average on 200 different sweeps as in the I_{w} parametric studies, some simulations needed to be run a lot more than 200 times. This is why the sweeps were particularly difficult to perform for negative SNRs. At extremely low SNRs (under -40 dB), the simulations were often impossible to perform, the nature of the noise being extremely detrimental to the results.

The results obtained with the *s*-analytics LO-TEA model are represented in Fig.5.9 and Fig.5.10. Concerning the recognition score, several things can be noticed directly. First, the sweep was truncated at -40 dB due to the practical impossibility to perform the simulations below that value. This is not a real problem as the most speaking feature of the plot is still visible. Indeed, it can be noticed that as soon as the SNR falls below 0 dB, the recognition score drops to 50%. This result is strongly expected, as it translates

the inability of the neural network to perform efficient recognition when the noise becomes too important compared to the initial signal. As the SNR decreases, the neural networks struggles more and more to distinguish the signal from the noise. Hence, a random choice between the two categories (sine and square signals) is naturally performed by the AI, leading to a stable score of 50%. As soon as the SNR get back to positive values, the recognition score saturates at 100% for the exact opposite reason : the signal becomes clearer and clearer as the SNR increases, making it progressively easier to recognize. Other interesting features can be noticed in Fig.5.9. For example, one can see that the gap between the TW and the WTA curves is much smaller than during the I_{w} sweep. That implies that the mean involved in the WTA approach has almost no beneficial influence on the recognition of noisy signals, leading to the same score as with the TW approach (50%). This is understandable : if all the samples of a signal are randomly classified by the neural network, the mean of all the samples will also lead to a random classification for the whole signal. Finally, it can be seen that the curves can be empirically approximated using a logistic function of the form of Eq.5.1 [31]. For the present case, a satisfactory approximation was found by minimizing the difference between the approximation and the average between the WTA and TW curves. The approximation is thus expressed as Eq.5.2. This allows to say that when SNR = 0 dB, the recognition score will be 88.31% in average, which still might be **better than the human eye in the same conditions** (see Fig.3.12). Once again, this approximation would not have been possible using a non-analytical model.

$$f(\mathrm{SNR}) \approx \frac{A}{(1+\exp{(Bx+C)})} + D \tag{5.1}$$

$$\mathrm{SCORE}(\mathrm{SNR}) \approx \frac{50}{1+\exp\left(\dfrac{-(\mathrm{SNR}+4.75)}{4}\right)} + 50 \tag{5.2}$$

The RMS values recorded during the sweep are represented in Fig.5.10. A striking feature of the plot is the truncation to of the RMS to 2.0 under SNR = 0 dB. Indeed, the RMS quickly exploded when the SNR went below 0 dB : values up to 10^9 were recorded. This is most probably due to the highly heterogeneous amplitude of the signal when high power Gaussian white noise is added to it. On top of that, one can notice that when the SNR increases higher than 0 dB, the RMS reaches the asymptotical value of 0. This result is expected as the signal becomes clearer and easier to recognize. Under 0 dB, despite the presence of the strong outliers, it can be observed that the RMS somewhat reaches a stable value of 1 on a certain range.

During the sweep of the SNR using the s-analytics HP-TEA model, it has been observed that the model was much more sensitive to noise than the s-analytics LO-TEA model. This is expected as this model involves a higher order power expansion of the $\Gamma(s)$ parameter. As the noise can add high outlying values to the input signal (which will be reflected in the reduced position s of the vortex core due to its dependence on the current intensity J_{in}), the reduced position can easily exceed its maximum stable value ($s = 0.8$), leading to a failure in the simulation. Hence, it was extremely difficult to perform simulations below 0 dB, so that the average on 200 simulations was practically impossible to carry out under -10 dB. It was however possible to observe interesting features on the plots. First, one can see that the recognition score quickly tends to 100% for

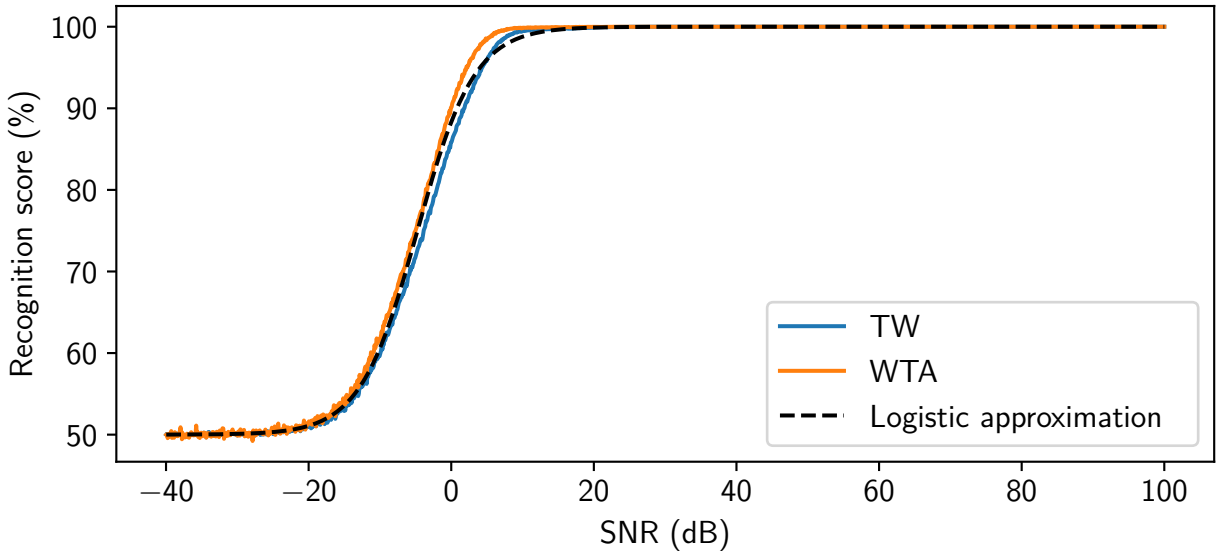

Fig. 5.9: Recognition score of the emulated neural network with respect to the SNR using the s-analytics LO-TEA model, averaged over 200 simulations

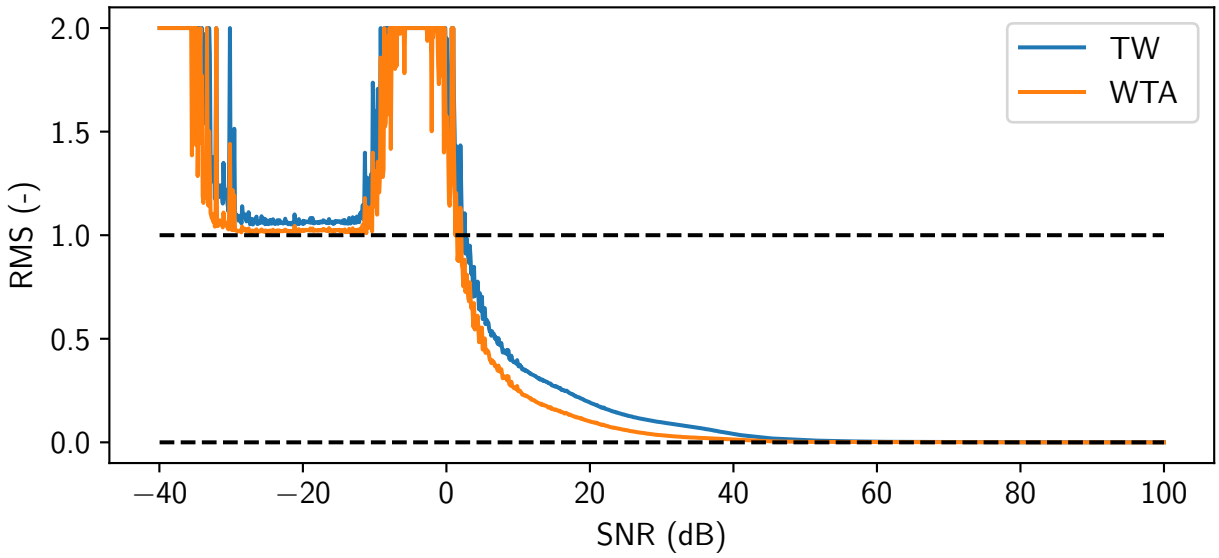

Fig. 5.10: Recognition score of the emulated neural network with respect to the SNR using the s-analytics LO-TEA model, averaged over 200 simulations

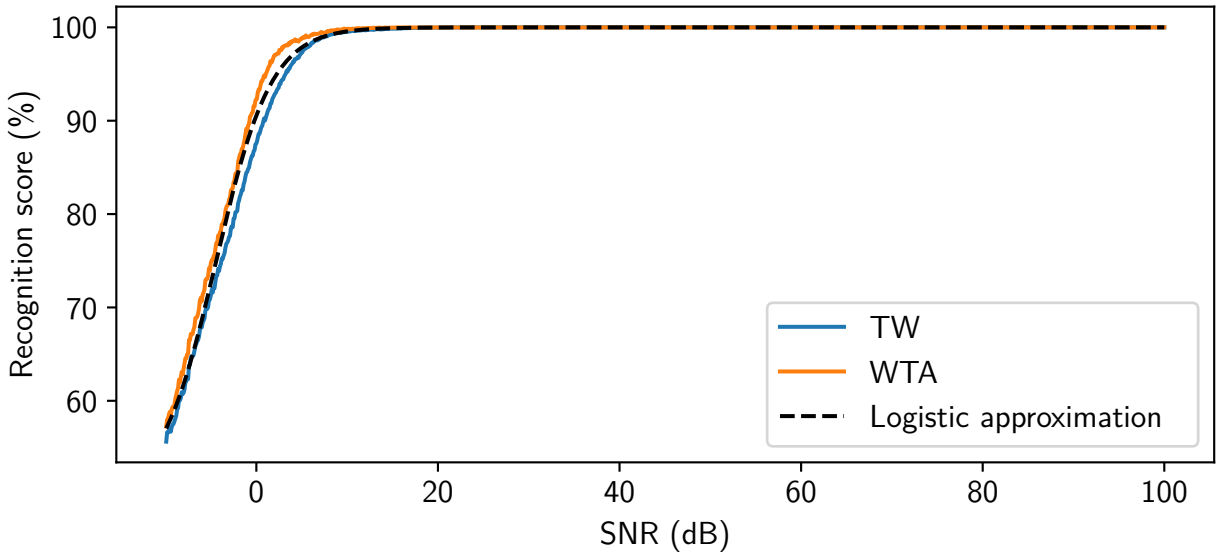

Fig. 5.11: Recognition score of the emulated neural network with respect to the SNR using the *s*-analytics HP-TEA model, averaged over 200 simulations

positive SNRs, similarly to the case with the *s*-analytics LO-TEA model. The recognition score also drops quickly when the SNR becomes negative. It was however not possible to assess the asymptotic value reached for highly negative SNRs. An approximation of the same form as Eq.5.1 could still be found using the same minimization method as with the *s*-analytics LO-TEA results (see Eq.5.3). As a result, the recognition score with a SNR of 0 dB can be estimated to 90.56% in average, which is slightly better than with the *s*-analytics LO-TEA model as expected. Unfortunately, the performances of the *s*-analytics HP-TEA model could not be assessed for SNRs lower than -10 dB, due to the high sensibility of the model to noisy signals.

$$\mathrm{SCORE}(\mathrm{SNR}) \approx \frac{50}{1+\exp\left(\dfrac{-(\mathrm{SNR}+4.43)}{3.04}\right)} + 50 \tag{5.3}$$

Concerning the RMS values, gathered in Fig.5.12, the same lower asymptotical value of 0 for positives SNRs as with the *s*-analytics LO-TEA model is observed. However, no conclusive observations could be made for negative SNRs, due to the difficulty of having successful simulations in this regime. One can however see that the RMS increases when the SNR decreases, which is obviously expected. It can also be observed that unlike the case with the *s*-analytics LO-TEA oscillator, no truncation of the RMS values around SNR$=0$ db was needed here.

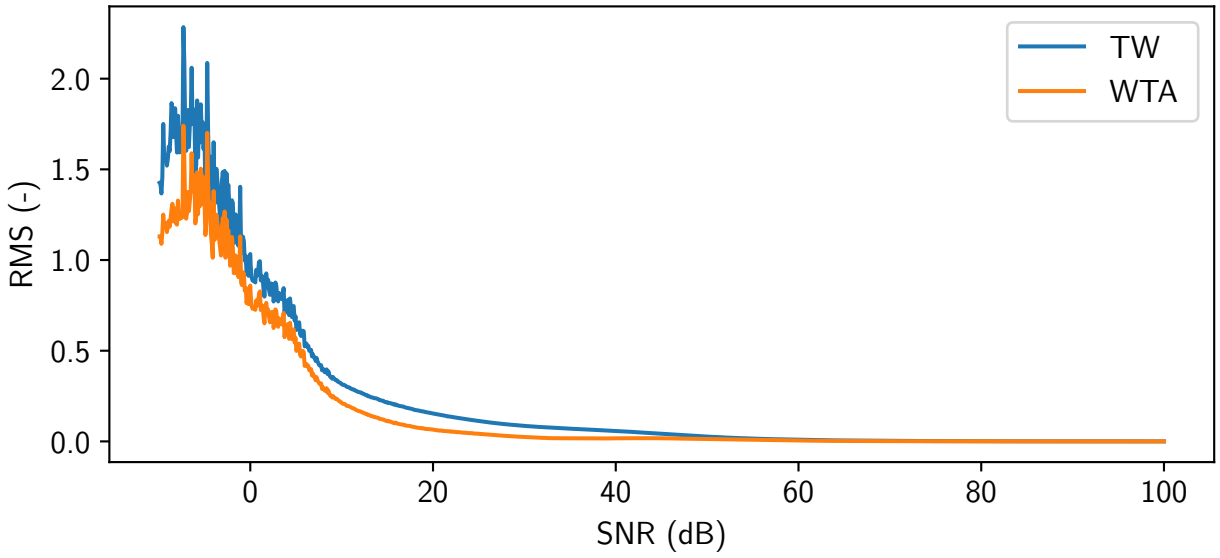

Fig. 5.12: Recognition score of the emulated neural network with respect to the SNR using the *s*-analytics HP-TEA model, averaged over 200 simulations

Summary

The impact of the working current intensity I_{w} on the performances of the neural network emulated with the *s*-analytics LO-TEA model and the *s*-analytics HP-TEA model was assessed during a sweep from $1.001 I_{\mathrm{cr1}}$ to I_{cr2}. It has first been shown that the WTA approach yielded higher recognition scores and lower RMS values than the TW approach, due to the beneficial role of the average involved in the WTA approach. The difference of efficiency between the two approaches can be suspected to be constant in most of the cases. It was also shown that the performances of the NN reached an asymptotical value after ~ 2.0 mA (upper limit for the recognition score and lower limit for the RMS values). This improvement of the performances is mainly due to the intrinsic behavior of the STVO at higher working current intensities, and the influence of the increase of the SNR when the working current intensity increases can be neglected. In general, the performances obtained with the *s*-analytics HP-TEA model were slightly better than the one obtained with the *s*-analytics LO-TEA model. For the operating parameters fixed prior the study, the optimal range of operation for I_{w} is located between 2.05 mA and 2.10 mA.
The parametric study on the noise level, although being rather difficult to carry out for low SNRs, revealed that the recognition score quickly dropped to 50% for negative SNRs, due to the difficulty for the NN to discern the noise from the signal. The score quickly reached 100% for positive SNRs, due to the improvement of the quality of the signal. It was also observed that the gap between the WTA and TW approaches was much smaller than during the I_{w} parametric study, implying that the averaging process involved in the WTA approach is not ineffective for noisy signals. The curves obtained with both models could be approximated using a fitted logistic function, allowing to indicate that the value of the recognition score at 0 dB was between 88% and 90% depending on the model used, which may be better than the human eye in these conditions. The RMS was much more sensitive to noise, hence preventing to drawn relevant conclusions on the impact of the noise on it.

Chapter 6

Speech recognition : Performances of the new models

As highlighted previously, the phenomenological oscillator model (subsection 3.2.1) did not match the experimental results obtained with the corresponding physical STVO (see Fig.1.26) [4]. Now that more accurate models have been developed, the same analysis can be carried out in order to compare the results obtained by simulation with the experimental results. The same study as in [4] was conducted. The TI-46 database was split in 10 subsets of 50 audio files containing the pronunciation of digits from 0 to 9. 9 of those subsets were used to train the neural network, and the last one was used to test its performances in the recognition. 4 different acoustic filters were used, including non-linear filters (MFCC, Spectro HP and Cochleagram) and a linear filter (Linear Spectrogram). The oscillator was simulated using the *s*-analytics LO-TEA and the *s*-analytics HP-TEA models previously presented. The results obtained during the testing phase, echoing Fig.1.26, are represented in Fig.6.1. They are composed the recognition score (or Word Success Rate) obtained with the WTA approach during the testing phase, averaged over the 10 different combinations {training sets; testing set}.

One can see that the performances obtained with the *s*-analytics LO-TEA and *s*-analytics HP-TEA models are almost perfectly matching the performances obtained with the physical sample. The main exception is in the case where the linear acoustic filter was used (leftmost column in Fig.6.1). Indeed, the average recognition score obtained during the simulations is far lower than the ones obtained with the phenomenological oscillator or the physical STVO (about 23%). The same plot for the training phase is shown in Fig.6.2. One can see that recognition score obtained with the *s*-analytics LO-TEA and the *s*-analytics HP-TEA models during the training phase is between 84% and 88%. This can be understood as the fact that the neural network is efficient at classifying the input data, but not at recognizing new unseen data without non-linear pre-treatment of the input. The overfitting estimator, computed as in Eq.6.1, represents the extent at which the neural networks relies too much on the training data, at the expense of the testing data. The noise present in the training data is hence considered as a feature of interest by the neural network, leading to a negative impact on the recognition rate on the test data. The overfitting was estimated to 3.28 in the case of the recognition using the *s*-analytics LO-TEA model without acoustic filter, and 4.68 with the *s*-analytics HP-TEA model. This corresponds to very high values as the mean error during testing is thus 3 to more

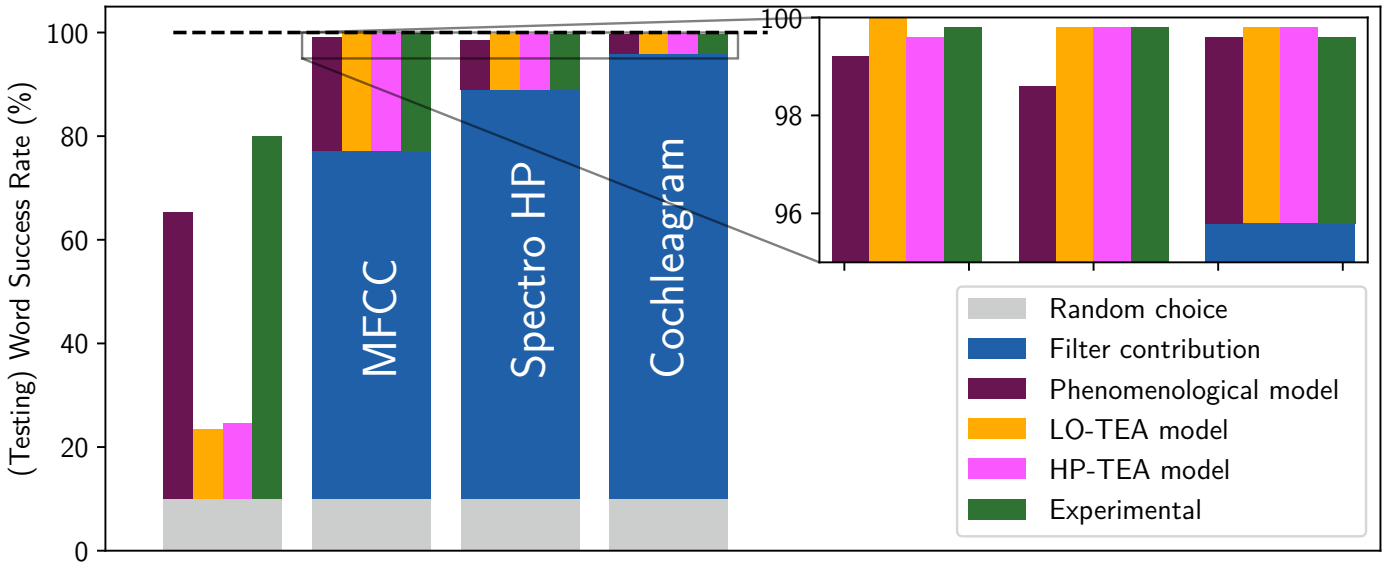

Fig. 6.1: Comparison between the different contributions to the recognition score (WSR), obtained with the WTA approach during testing, with different acoustic filters. The first column represents the case with linear acoustic filtering (linear spectrogram). The green and purples columns are from [4]. The orange columns represent the score obtained with the *s*-analytics LO-TEA model, and the pink columns represent the score obtained with the HP-TEA model.

than 4 times higher than during training. One can conclude that without any acoustic pre-filtering, the neural network emulated using the analytical models developed in this work will overfit the data.

$$\texttt{overfitting} = \frac{n_{\text{train}} \sum\limits_{n_{\text{test}}} \left(\hat{\mathbf{t}}_\sigma - \mathbf{t}_\sigma\right)^2_{\text{testing}}}{n_{\text{test}} \sum\limits_{n_{\text{train}}} \left(\hat{\mathbf{t}}_\sigma - \mathbf{t}_\sigma\right)^2_{\text{training}}} \tag{6.1}$$

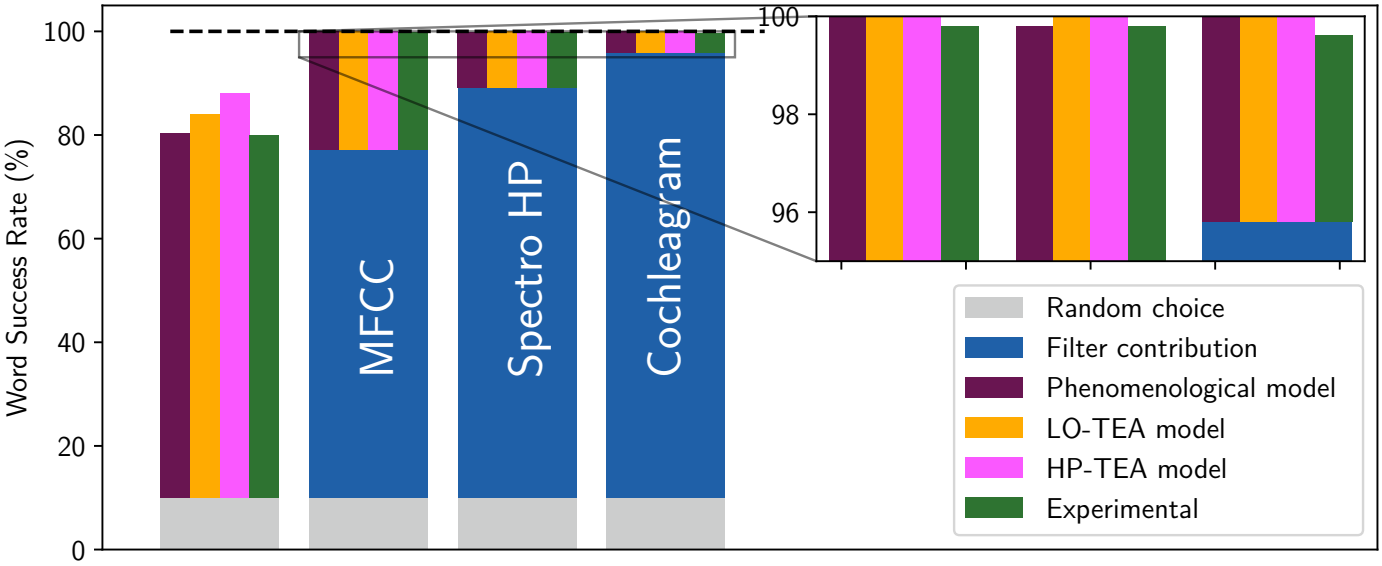

Fig. 6.2: Comparison between the different contributions to the recognition score (WSR), obtained with the WTA approach during training, with different acoustic filters. The first column represents the case with linear acoustic filtering (linear spectrogram). The green and purples columns are from [4]. The orange columns represent the score obtained with the *s*-analytics LO-TEA model, and the pink columns represent the score obtained with the HP-TEA model.

Chapter 7

Discussion

First, the *s*-ODE LO-TEA model and the *s*-analytics LO-TEA model were compared. Although yielding the exact same results when performing the benchmarking task, a striking difference could be observed concerning the simulation time required by both models. It has indeed been shown that **the analytical model allows to reduce the required computation time by two orders of magnitude compared to the numerical model**.

Then, the *s*-analytics LO-TEA model and the *s*-analytics HP-TEA model were compared. In general, the truncation to the second order of the power expansion of the $\Gamma(s)$ parameter could be observed through slight differences in the results yielded by the two models. Indeed, the path followed by the vortex core is significantly different when higher order terms are took into consideration. This is translated by a gap in the performances of the two models in the benchmarking task. As a matter of fact, the *s*-analytics HP-TEA model yielded slightly better results (higher recognition scores and lower RMS values) than its LO-TEA homologue. The required computation times were however practically similar, differing by only 0.03 s. This implies that the higher order, more complex dynamics of the vortex core described by the *s*-analytics HP-TEA model plays a slight, yet non-negligible role in the non-linear treatment of the input data.

Using the analytical models, a sweep of the working current intensity I_{w} was performed. Actually, two variants of the I_{w} sweep were carried out. The first sweep was performed by keeping the voltage range inherent to the noise constant. Hence, the SNR was brought to increase during the sweep, leading to an input signal of progressively better quality. It was shown that the recognition score and the RMS respectively reach an upper and a lower asymptotical limit, the recognition score tending towards 100% and the RMS tending towards 0. However, this behavior was proved to not be due to the increasing SNR as it was also observed during the second I_{w} sweep, where the SNR was kept constant as the working current intensity increased. The main influence of the working current intensity on the performances of the neural network thus concerns the inherent dynamics of the vortex core. Indeed, larger working current intensities lead to oscillations of higher amplitude in the dynamics of the vortex core, which then lead to a better non-linear treatment of the input data. It was also observed in all the cases that the WTA approach allows to obtain significantly better results than the TW approach due to the failure-absorbing character of the averaging process involved in the first method. In most of the cases, the gap of performances between the two approaches was almost constant, meaning

that the advantageous character of the WTA approach doesn't scale with the working current intensity. In fact, this gap between the two approaches doesn't mean that the TW approach is unefficient, as it reflects the ability of the NN to recognize a very small fraction of the signal, while the WTA approach relies on the whole signal to perform the recognition. Finally, it was also observed that the *s*-analytics HP-TEA model was again slightly better than the *s*-analytics LO-TEA model during the sweep, confirming the beneficial role of the higher order dynamics involved in it.

A parametric sweep of the level of noise was also performed, by sweeping the SNR from negative to positive values. The first noticeable result of this study is the recognition score dropping down to 50% as soon as the power of the noise becomes comparable to that of the input signal (SNR= 0 dB). That means that for a strongly noisy signal, the neural network has so much difficulties to classify the data that a random choice is performed between the different categories. When the SNR increases in the positive values, the recognition score quickly reaches 100%, reflecting the easier recognition of a signal of higher quality. Furthermore, it was also seen that the gap of performances between the WTA and the TW approach was barely noticeable, implying that the averaging process involved in the WTA approach cannot compensate for the errors due to a noisy signal (the performances of the WTA approach are still slightly better than with the TW approach though). The difficulty to perform the study at low negative SNRs was due to the increasing probability to add outliers values of high amplitude to the signal, leading to the total input current intensity to exceed I_{cr2}, and hence to the expulsion of the vortex out of the STVO. It was still possible to approximate the recognition score of the neural network as a function of the SNR using a parametrized logistic function. This allowed to see that the *s*-analytics model yields slightly better average scores at 0 dB (90.56%) than its LO-TEA homologue (88.31%). In both case, these results are encouraging as **the signal is barely recognizable for the human eye at the same SNR**. The RMS revealed itself as an indicator much more sensitive to the noise than the recognition score. Indeed, high outlying value could be observed under 0 dB, due to the chaotic nature of the noisy signal. Using both models, it was still possible to see that the RMS dropped to 0 quickly after entering the positive values of SNR. No satisfying conclusion could however be made under 0 dB.

The most striking advantage of the analytical models is obviously the rapidity they offer for the simulations. The parametric studies featured in this work were computed on a cluster of high performance computing, and each of them took about 6 hours in average due to the high number of executions of the benchmarking task (200000 in average) required. It has been shown that the *s*-ODE LO-TEA model requires 85 times more time than the *s*-analytics LO-TEA model to perform the benchmarking task. The first approach used to know the dynamics of the vortex core was based on solving Eq.2.12, and required 10 times more time than the *s*-ODE LO-TEA model itself. Finally, the micromagnetic simulations were proved to require about 1 million times more time than the model based on Eq.2.12. This results in an astonishing **factor of** 850000000 **between the time required to perform simulations using the micromagnetic formalism and the time required with the analytical models (LO-TEA or HP-TEA)**. It also implies that **the parametric studies performed in this work would have required more than** 580000 **years using MMS**.

All these results can also be analyzed under the prism of the splitting phenomenon

highlighted in [28], due to the Ampère-Oersted field and the chirality resulting from the configuration of the setup.

Finally, a speech recognition task was performed using the s-analytics LO-TEA and the s-analytics HP-TEA models. It echoes the results obtained in [4], where the results obtained by simulation of the phenomenological oscillator didn't match the experimental results. Here, the results are much more consistent with the results yielded by the physical STVO in most of the cases (when non-linear acoustic filters were used to pre-treat the data). This highlights the accurate character of the Thiele equation approach. Concerning the gap obtained when a linear acoustic was used, it implies that in the absence of non-linear pre-treatment of the data, both analytical models don't match the physical reality, leading to a neural network which is only efficient to classify the data, but not recognize new unseen data due to strong overfitting.

Conclusion

Neural networks are efficient in performing tasks such as pattern and speech recognition. However, the training process of the latter can be strongly time and energy-consuming. One solution to circumvent this issue is reservoir computing, which allows to decrease the time and the energy needed to train a neural network to a given task. Reservoir computing is also suited for hardware implementations. Hence, it is possible to implement physical neurons using spin torque vortex nano-oscillators, nanoscale oscillators based on a spintronics phenomenon : the spin transfer torque. These oscillators mimic the behavior of biological neurons thanks to their highly non-linear behavior.

In this work, the basics of reservoir computing have been explained, as well as the description of STVOs, and the underlying spintronics phenomena involved. State-of-the-art examples of hardware implementation of a neural network using STVOs have also been explained, showing the high range of applications offered by this type of neural networks.

To ease and speed up the studies on this type of architectures, accurate and efficient simulations are needed. However, micromagnetic simulations, despite being quantitatively accurate, require large amounts of energy and computation time, implying that most efficient models have to be developed. On the other hand, the phenomenological model used up until now has been shown to be inaccurate in front of the experimental results. To overcome these problems, the Thiele equation approach was used, allowing to simulate the dynamics of the vortex core much more accurately than ever. Several different original models and implementations were therefore developed, based either on the numerical or the analytical solving of this equation with more or less efficiency.

To assess the performances of these models, a benchmarking pattern recognition task was designed. It allowed to show that the analytical models allows to speed up the simulations up to two orders of magnitude, compared to the numerical model, which is itself about seven orders of magnitude faster than micromagnetic simulations. The rapidity of the analytical models also allowed to perform parametric studies on several parameters having an impact on the performances of the emulated neural network, which would have taken hundreds of thousands of years using state-of-the-art micromagnetic simulations. Finally, it has been shown that the developed models were much closer to the physical data than ever.

This thesis has allowed to prove that the TEA can be used to describe the dynamics of a STVO more accurately and efficiently than ever, thus answering the open questions raised by the previous works on the topic. In terms of perspectives, the models developed during this thesis could be used to simulate larger hardware neural networks (i.e composed of more than one neuron) using a proper model of the interactions between two adjacent STVOs, taking another step towards the development of concrete hardware neural networks.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436–444, 2015.

[2] S.-C. Wang, "Artificial neural network," *Interdisciplinary Computing in Java Programming*, p. 81–100, 2003.

[3] Q.-F. Ou, B.-S. Xiong, L. Yu, J. Wen, L. Wang, and Y. Tong, "In-memory logic operations and neuromorphic computing in non-volatile random access memory," *Materials*, vol. 13, no. 16, p. 3532, 2020.

[4] F. Abreu Araujo, M. Riou, J. Torrejon, S. Tsunegi, D. Querlioz, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and et al., "Role of non-linear data processing on speech recognition task in the framework of reservoir computing," *Scientific Reports*, vol. 10, no. 1, 2020.

[5] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," *Optics Express*, vol. 20, no. 3, p. 3241, 2012.

[6] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, 2017.

[7] Y. Paquot, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Reservoir computing: A photonic neural network for information processing," *SPIE Proceedings*, 2010.

[8] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, no. 1, 2012.

[9] M. N. Baibich, J. M. Broto, A. Fert, F. N. Van Dau, F. Petroff, P. Etienne, G. Creuzet, A. Friederich, and J. Chazelas, "Giant magnetoresistance of (001)fe/(001)cr magnetic superlattices," *Phys. Rev. Lett.*, vol. 61, pp. 2472–2475, Nov 1988.

[10] P. Dey and J. N. Roy, *Spintronics: Fundamentals and applications.* Springer, 2021.

[11] S. Yuasa and D. D. Djayaprawira, "Giant tunnel magnetoresistance in magnetic tunnel junctions with a crystalline mgo(001) barrier," *Journal of Physics D: Applied Physics*, vol. 40, no. 21, 2007.

[12] I. Appelbaum, B. Huang, and D. J. Monsma, "Electronic measurement and control of spin transport in silicon," *Nature*, vol. 447, no. 7142, p. 295–298, 2007.

[13] D. Ralph and M. Stiles, "Spin transfer torques," *Journal of Magnetism and Magnetic Materials*, vol. 320, no. 7, p. 1190–1216, 2008.

[14] F. Abreu Araujo, *Dynamical and synchronization properties of vortex based spin-torque nano-oscillators*. PhD thesis, Louvain School of Engineering, UCLouvain, 2015.

[15] J. Torrejon, M. Riou, F. Abreu Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, and et al., "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, vol. 547, no. 7664, p. 428–431, 2017.

[16] W. H. Rippard, M. R. Pufall, S. Kaka, T. J. Silva, S. E. Russek, and J. A. Katine, "Injection locking and phase control of spin transfer nano-oscillators," *Physical Review Letters*, vol. 95, no. 6, 2005.

[17] M. Riou, J. Torrejon, B. Garitaine, F. Abreu Araujo, P. Bortolotti, V. Cros, S. Tsunegi, K. Yakushiji, A. Fukushima, H. Kubota, and et al., "Temporal pattern recognition with delayed-feedback spin-torque nano-oscillators," *Physical Review Applied*, vol. 12, no. 2, 2019.

[18] D. Marković, N. Leroux, M. Riou, F. Abreu Araujo, J. Torrejon, D. Querlioz, A. Fukushima, S. Yuasa, J. Trastoy, P. Bortolotti, and et al., "Reservoir computing with the frequency, phase, and amplitude of spin-torque nano-oscillators," *Applied Physics Letters*, vol. 114, no. 1, p. 012409, 2019.

[19] M. Goiriena-Goikoetxea, K. Y. Guslienko, M. Rouco, I. Orue, E. Berganza, M. Jaafar, A. Asenjo, M. L. Fernández-Gubieda, L. Fernández Barquín, A. García-Arribas, and et al., "Magnetization reversal in circular vortex dots of small radius," *Nanoscale*, vol. 9, no. 31, p. 11269–11278, 2017.

[20] F. Abreu Araujo, A. D. Belanovsky, P. N. Skirdkov, K. A. Zvezdin, A. K. Zvezdin, N. Locatelli, R. Lebrun, J. Grollier, V. Cros, G. de Loubens, and et al., "Optimizing magnetodipolar interactions for synchronizing vortex based spin-torque nano-oscillators," *Physical Review B*, vol. 92, no. 4, 2015.

[21] S. de Wergifosse, "Spin-diode effect in vortex based nano-oscillators for hardware artificial intelligence applications," Master's thesis, Louvain School of Engineering, UCLouvain, 2021.

[22] K. Y. Guslienko and V. Novosad, "Comment on "scaling approach to the magnetic phase diagram of nanosized systems"," *Physical Review Letters*, vol. 91, no. 13, 2003.

[23] S. Wittrock, P. Talatchian, M. Romera, S. Menshawy, M. Jotta Garcia, M.-C. Cyrille, R. Ferreira, R. Lebrun, P. Bortolotti, U. Ebels, and et al., "Beyond the gyrotropic motion: Dynamic c-state in vortex spin torque oscillators," *Applied Physics Letters*, vol. 118, no. 1, p. 012404, 2021.

[24] M. Riou, J. Torrejon, B. Garitaine, F. Abreu Araujo, P. Bortolotti, V. Cros, S. Tsunegi, K. Yakushiji, A. Fukushima, H. Kubota, and et al., "Temporal pattern recognition with delayed-feedback spin-torque nano-oscillators," *Physical Review Applied*, vol. 12, no. 2, 2019.

[25] J. Leliaert and J. Mulkers, "Tomorrow's micromagnetic simulations," *Journal of Applied Physics*, vol. 125, no. 18, p. 180901, 2019.

[26] G. S. Abo, Y.-K. Hong, J. Park, J. Lee, W. Lee, and B.-C. Choi, "Definition of magnetic exchange length," *IEEE Transactions on Magnetics*, vol. 49, no. 8, p. 4937–4939, 2013.

[27] A. P. Guimarães, "The basis of nanomagnetism," *Principles of Nanomagnetism*, p. 1–20, 2009.

[28] F. Abreu Araujo, C. Chopin, and S. de Wergifosse, "Ampere-oersted field splitting of the nonlinear spin-torque vortex oscillator dynamics," 2021.

[29] A. A. Thiele, "Steady-state motion of magnetic domains," *Physical Review Letters*, vol. 30, no. 6, p. 230–233, 1973.

[30] Wikipedia, "Six sigma." `https://en.wikipedia.org/wiki/Six_Sigma`. (Accessed on 05/22/2022).

[31] Wikipedia, "Logistic function." `https://en.wikipedia.org/wiki/Logistic_function`. (Accessed on 29/05/2022).

Appendix

Appendix A

Oscillator models

A.1 Phenomenological oscillator

```
def oscillator(Input, OSC):
  dt = OSC.dt*1e9       #(1/sampling rate) (nanoseconds)
  tau = OSC.Trelax*1e9  #relaxation time (nanoseconds)
  Rosc = OSC.Rosc       #DC resistance (Ohm)
  pA = OSC.pA

  Delta_I = OSC.I0 - OSC.Ic
  EXP_term = np.exp(-dt/tau)

  InputV = OSC.AmpV*Input
  Out = np.zeros_like(InputV)

  Out[0] = pA*np.sqrt(np.abs(Delta_I - InputV[0]/Rosc))
  for i in range(1,len(Out)):
    OffsetAsVal = pA*(np.sqrt(np.abs(Delta_I - InputV[i]/Rosc)))
    Out[i] = OffsetAsVal + (Out[i-1] - OffsetAsVal)*EXP_term

  return Out
```

A.2 s-ODE LO-TEA oscillator

```
def s_only(s, t, LOM, Jdc, Jin, Dt):
    aJ = LOM[0]
    bJ = LOM[1]
    a = LOM[2]
    b = LOM[3]
    fl = int(floor(t/Dt))
    J = Jdc + Jin[fl]
    alpha = aJ*J + a    #? MHz
    beta = bJ*J + b     #? MHz
    Gamma = (alpha + beta*s**2)*1e-3 #? GHz
    return Gamma*s
```

A.3 s-analytics LO-TEA oscillator

```
def calc_s_th(si, Ji, Dt, Jdc, LOM):
    aJ = LOM[0]
    bJ = LOM[1]
    a = LOM[2]
    b = LOM[3]
    J = Jdc + Ji
    Jcr = -a / aJ
    alpha = (aJ * J + a) * 1e-3
    beta = (bJ * J + b) * 1e-3
    try :
        interm = si**2 * beta / alpha
        s = si / sqrt( (1 + interm) * exp(-2*alpha*Dt) - interm)
    except :
        pass
    return s
```

A.4 s-analytics HP-TEA oscillator

```
def calc_s_th_full(si, Ji, Dt, Jdc, LOM=[]):
    aJ = LOM[0]
    bJ = LOM[1]
    a = LOM[2]
    b = LOM[3]
    J = Jdc + Ji
    alpha = (aJ * J + a) * 1e-3
    beta = (bJ * J + b) * 1e-3
    b = 4930.675312
    bJ = -3032.315785
    cJ = 741.379846
    dJ = -89.896255
    eJ = 5.425502
    fJ = -0.130746
    beta = (bJ * J + cJ * J**2 + dJ * J**3 + eJ * J**4 + fJ * J**5 + b)
   *1e-3
    b = 157.142220
    bJ = -95.968883
    cJ = 23.535504
    dJ = -2.871111
    eJ = 0.175043
    fJ = -0.004282
    n = (bJ * J + cJ * J**2 + dJ * J**3 + eJ * J**4 + fJ * J**5 + b)
    interm = si**n * beta / alpha
    return si / ( (1 + interm) * exp(-n*alpha*Dt) - interm)**(1/n)
```

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**