ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Interferometric stabilisation of a fibre-based optical computer

Experimental study

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil physicien à finalité spécialisée

Denis Verstraeten

Directeur
Professeur Marc Haelterman

Co-Promoteur
Professeur Serge Massar

Superviseur
Lorenz Butschek

Service
Opera

Année académique
2018 - 2019

Abstract

Acknowledgements

Contents

Chapter 1

Introduction

Chapter 2

Reservoir Computing

Reservoir Computing (RC) is a bio-inspired artificial Recurrent Neural Network (RNN) which is based on the Echo State Network (ESN) introduced by Herbert Jaeger in [10]. This computation scheme is well suited for real-time data processing and for chaotic time series prediction[10, 11, 16], and achieves state of the art performances in those domains, as well as in speech recognition[27, 24, 14], nonlinear channel equalisation[10] and financial forecasting [3].

The first section of this chapter introduces the different concepts linked with RC. It starts by giving a brief overview on the Neural Network (NN) computation paradigm from which RC has been derived. After that, the structure, operating principles and features of Reservoir Computer (RC) are presented. Then, a few elements of Machine Learning (ML) are mentioned in order to have a first glimpse of the training procedure for RC. The second section is devoted to the mathematical model governing a RC. The last section is about Photonic Reservoir Computers. During the last decade, several physical implementations of RC have been proposed, some of which being optical setups. This kind of RC is presented, and most specifically reservoir involving Time Division Multiplexing (TDM). Finally, a RC is used to resolve two different benchmark tasks and the results on the computations are shown.

2.1 Reservoir Computing framework

2.1.1 Artificial Neural Network

A RC is specific kind of NN, which is a computation paradigm mimicking the behaviour of a biological brain. The artificial neurons are simply interconnected entities carrying an activation level. The way the activation level is updated depends on the scheme, but the basic idea is common for all of them: a neuron receives a linear combination of the activation level of the neurons to which it is connected, and then computes a nonlinear transformation of this value. This gives the new activation level. On figure 2.1, a feedforward NN is depicted. It is called feedforward because the computation goes from left (input neurons in red) to right (output neurons in green). Feedforward NN are organised in layers, and a neuron from one layer can only be influenced by neurons in the adjacent layers. This is shown on the Figure by the arrows representing the connections. The gray neurons in the middle belong to the hidden layers, which are used to improve the computing power of such networks. The results of a computation can be read on the activation level of the output neurons [21, p.727][4, p.225].

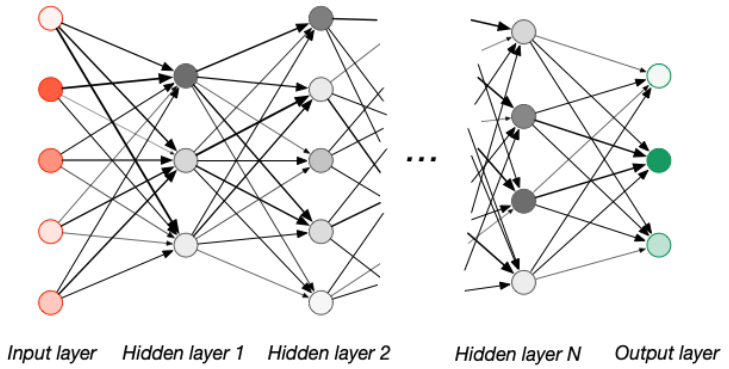


Figure 2.1: Schematic representation of a feedforward NN

2.1.2 Reservoir Computing

RC have been designed to process time dependent inputs, so their structure is inherently different from that of a feedforward NN, because they need to exhibit other properties. In this scheme, all the neurons are interconnected and form what is called a reservoir. The reservoir is fed with the time dependent input signal it should process. When the reservoir is properly set up, the activation level of each of the neurons becomes a systematic transformed version of the input signal [10]. This operating point is called the echo state and allows RC to reach their best performances [9, 11]. In this regime, RC exhibit a short-term memory of the previous inputs [10], which could explain why they perform so well in time dependent situations. There are many physical implementations of RC proposed in the literature, many of which are based on optical setups [22], that is why section 2.3 is devoted to them.

On figure 2.2, a RC is shown. The neurons of the reservoir are represented in orange. They characterise what is called the state of the reservoir, which is encoded by $\mathbf{x}(t)$. They are coupled by the connection matrix $\mathbf{W}$. The input signal $u(t)$ is fed into the input neuron (blue) and is coupled to the reservoir *via* the input matrix $\mathbf{W}^{\text{in}}$. The output $y(t)$ is read on the output neuron (red) and is obtained thanks to the output matrix $\mathbf{W}^{\text{out}}$. This matrix is the only one that needs to be updated when the reservoir is learning. This task is not straightforward, that is why the next paragraph takes care of introducing the different approaches that can be followed to compute $\mathbf{W}^{\text{out}}$. For some applications, it can be useful to also have a feedback of the output sent back into the reservoir. This can be achieved by the feedback matrix $\mathbf{W}^{\text{fb}}$. The mathematical aspects mentioned in this paragraph are detailed in section 2.2.

2.1.3 Machine Learning

Regardless of the learning scheme used to train a NN, the basic idea is always to minimise the difference between the desired and the actual outputs. In practice, this is achieved by updating the different connection coefficients of the NN [4, p.233][21, p.733]. This procedure often turns out to be a really complicated task for feedforward NN, which explains why the development of efficient ML algorithms is such a hot topic nowadays. In contrast, as can be seen on figure 2.3, RC only need their output weights to be adjusted when being trained, which makes them computationally lighter [10]. This is due to the fact that the connections of the reservoir should not contain any information about the task, but should only be used to reach the ESN regime, as mentioned in the previous paragraph. There are two main families of training methods for RC [12]. On the one

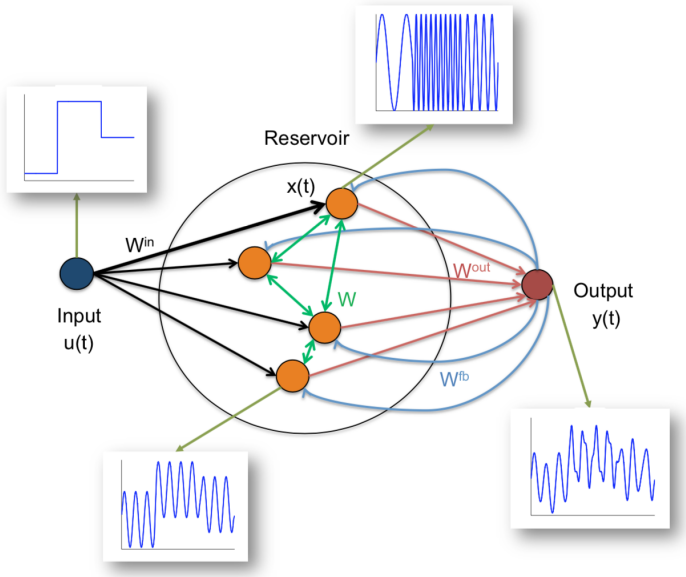


Figure 2.2: Schematic representation of a Reservoir Computer [3]

hand, there is the *batch learning*, which comprises the methods requiring to first store a wealth of data regarding the task being taught before being able to actually compute the output weights. Once enough data is gathered, this kind of algorithms returns the optimal weights all at once. They present the advantage of involving only one training phase, after which the RC are ready to perform. However, the need for vast amount of data and the inability for the RC to adapt to an input evolving out of the range for which it has been trained are two drawbacks. On the other hand, *online learning* methods allow to iteratively improve the output weights. Therefore, starting from a first guess, these algorithms can converge to suitable output weights. They are much more adaptable than the batch learning ones. However, their convergence is not guaranteed and can be slow [13, 23].

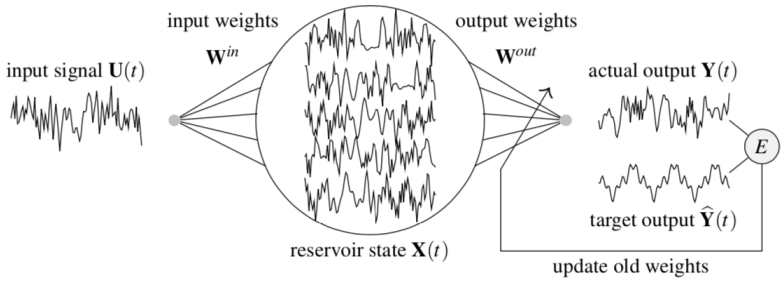


Figure 2.3: Learning procedure for Reservoir Computer [9]

2.2 Mathematical Model

In this section, an overview of the mathematical framework is given. First, the different objects are formally defined, and their dynamics is presented. Then, a few key elements about the computation of the output weights are introduced.

2.2.1 Dynamics of the reservoir

let us define the *state* of the reservoir. As said previously, the RC can be fully described by the activation level of each of its neurons. The state is therefore defined as a vector whose components are the activation levels of the neurons. If the number of neurons making up the reservoir is N, and if x_i is the activation level of the i^{th} neuron, then the state vector reads as follows:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix} \tag{2.1}$$

The dynamics governing the state vector and the output of the reservoir proposed in [11] are presented below. In practice, it is too general for the implementations studied in this work. However, the equations are introduced without loss of generality, and simplifying assumptions applying the photonic implementations of RC will be specified in the section devoted to them.

$$\mathbf{x}(n+1) = \mathbf{f}\left(\mathbf{W}^{\text{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{\text{fb}}\mathbf{y}(n)\right) \tag{2.2}$$

$$\mathbf{y}(n+1) = \mathbf{f}^{\text{out}}\left(\mathbf{W}^{\text{out}}\left(\mathbf{x}(n+1), \mathbf{u}(n+1), \mathbf{y}(n)\right)\right) \tag{2.3}$$

Different elements need to be defined:

- $n \in \{1, \ldots, T\}$ is the discrete time variable
- $\mathbf{u} \in \mathbb{C}^k$ is the input vector which enters the reservoir through the input neurons
- $\mathbf{W}^{\text{in}} \in \mathbb{C}^{N \times k}$ is the input matrix. It indicates how the k input neurons are connected to the neurons of the reservoir
- $\mathbf{x} \in \mathbb{C}^N$ is the state vector, as said previously
- $\mathbf{W} \in \mathbb{C}^{N \times N}$ is the synaptic matrix, or the connection matrix which has already been introduced
- $\mathbf{y} \in \mathbb{C}^m$ is the output vector of the reservoir whose value can be read out on the output neurons
- $\mathbf{W}^{\text{fb}} \in \mathbb{C}^{N \times m}$ is the feedback matrix. It couples the output back into the reservoir
- $\mathbf{f} : \mathbb{C}^N \mapsto \mathbb{C}^N$ is the nonlinear function mapping the linear combination it receives as argument to a valid state vector
- $(\mathbf{x}(n+1), \mathbf{u}(n+1), \mathbf{y}(n))$ is the concatenation of those three vectors
- $\mathbf{W}^{\text{out}} \in \mathbb{C}^{m \times (N+k+m)}$ is the output matrix of the reservoir. It is optimised through ML
- $\mathbf{f}^{\text{out}} : \mathbb{C}^m \mapsto \mathbb{C}^m$ is the output function of the reservoir

2.2.2 Computation of the output weights

To determine the output matrix $\mathbf{W}^{\text{out}}$ in the batch learning approach, one needs to perform a ridge (or Tikhonov) regression [24], which is an improved version of multivariate linear regression that improves the numerical stability of the scheme, and that prevents overfitting of the data. By restricting the desired output to a scalar function $\hat{y}(n)$ and by taking a learning period of T time steps, one defines the matrices $\mathbf{X}$ and $\hat{\mathbf{Y}}$ and solves the following equation to find the output weights vector $\mathbf{W}^{\text{out}}$:

$$\mathbf{X} = \begin{bmatrix} x_0(0) & x_1(0) & \dots & x_N(0) \\ x_0(1) & x_1(1) & \dots & x_N(1) \\ \vdots & & & \vdots \\ x_0(T) & x_1(T) & \dots & x_N(T) \end{bmatrix}, \quad \hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}(0) \\ \hat{y}(1) \\ \vdots \\ \hat{y}(T) \end{bmatrix} \tag{2.4}$$

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \epsilon\mathbf{I}\right)\mathbf{W}^{\text{out}} = \mathbf{X}^{\mathrm{T}}\hat{\mathbf{Y}} \tag{2.5}$$

Here ϵ is the constant used for the Tikhonov regression. By setting ϵ to 0, one finds the *normal equation* that comes up when solving a linear regression [9] in the sense of the least squares. This procedure can be generalised to higher dimensions desired output vectors $\hat{\mathbf{y}}(n)$. Different optimisation algorithms can be used to compute the output weights in practice, but their description is out of the scope of this work, see [15] for more details.

Different metrics can be used to capture the distance between the actual and the desired outputs. In the literature, one of the most frequent ones is the Normalised Mean Square Error (NMSE) [7] (with $\hat{\mathbf{y}}(n)$ the target vector, $\langle \dots \rangle_n$ the average with respect to n, and $\| \dots \|$ the euclidean norm):

$$\text{NMSE} = \frac{\langle \|\hat{\mathbf{y}}(n) - \mathbf{y}(n)\|^2 \rangle_n}{\langle \|\hat{\mathbf{y}}(n) - \langle \hat{\mathbf{y}}(n) \rangle_n\|^2 \rangle_n} \tag{2.6}$$

2.3 Introduction to Photonic Reservoir Computing

As already mentioned, different implementations of Photonic Reservoir Computing (PRC) have been proposed [22]. In this section, systems in which neurons are multiplexed in time are studied because they constitute a good first approach to PRC and because they bring insights that are interesting for the scheme explored in this thesis. However, it is worth mentioning that one can find among others spatially distributed RC based on fully integrated silicon-chip with nonlinearities stemming from Semiconductor Optical Amplifiers [25], and on diffractively coupled Vertical-Cavity Surface-Emitting Lasers [5].

In this section, the assumptions applying to reservoir with TDM of the neurons are first presented. Indeed, the equations introduced in the previous section can be substantially simplified when one is working with this kind of reservoir. After that, setups where neurons are encoded in the intensity of the light are considered. Finally, experiments in which the neurons are represented in the phaser of the electric field are presented. Recalling that the light intensity is proportional to the squared modulus of the phaser of the electric field, it is shown that the outputs of these two kinds of reservoir are different, but analogous, and that they rely on the same mathematical interpretation. The latter scheme is studied with greater length since it reaches state of the art performances in classical benchmark tasks and since the novel implementation which is the main topic of

this work shares some features with it. As a last remark, and to give a better understanding on the flexibility of RC, in [8] the researchers managed to perform speech recognition and to resolve the XOR problem[1] in a bucket of water.

2.3.1 Time Division Multiplexing of the neurons

Many implementations of RC based on TDM of the neurons have been proposed in the literature [18, 2, 7, 6, 26, 28]. Let T be the time scale of the input signal. It should be close to the Round-Trip Time (RTT) of the delay line, but not exactly the same in order to be able to couple the neurons. The detuning between the RTT and T is controlled by the integer parameter k. On figure 2.4, one can observe how the neurons can be multiplexed in time. One can see that the allocated window for each of the neurons lasts $\theta = T/N$. Since θ cannot be arbitrarily small in practice[2], this implies that the greater the number of neurons, the greater T and thus the slower the time scale of the input data. This suggests that one should look for a tradeoff between accuracy and speed in data processing for this kind of RC. For example, in [28], the refresh rate of the input is around 0.9 GHz.

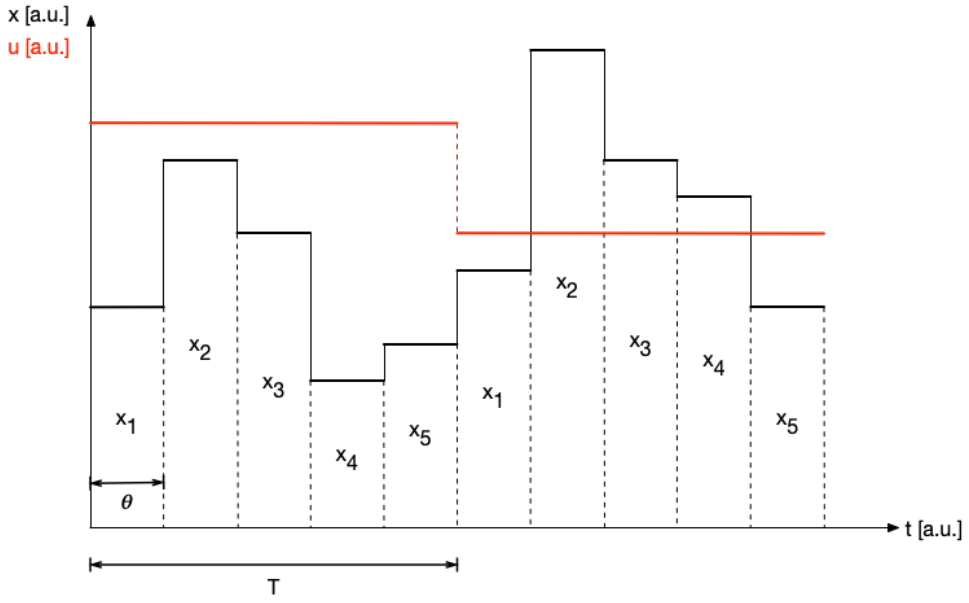


Figure 2.4: Schematic representation of the evolution of the neurons in time for a TDM reservoir

2.3.2 Simplifying assumptions

In this section, the mathematical framework introduced in section 2.2 is adapted to describe the behaviour of TDM PRC in a suitable way. Further details concerning specific types of PRC are added in the following sections which are devoted to them.

[1]The XOR task consists in reproducing the behaviour of a logical XOR gate, which is a task of historical importance for NN [17].

[2]If θ gets too short, it exceeds the bandwidth of measurement devices, so it becomes impossible to measure the neurons.

Input and connection matrices The input of the reservoir is real scalar function $u(n)$ for PRC, therefore the expression of $\mathbf{W}^{\text{in}}$ becomes a real vector of length N which is called the *input mask* $\mathbf{m}$ in the literature. The input mask can be chosen in different ways: in [7], they use a sinusoidal input mask whereas in [2, 28, 18] the input masks are randomly chosen.

Very few constraints apply to the creation of the connection matrix $\mathbf{W}$. It can be randomly generated and sparse. However, to make the occurence of the echo state more likely to happen, one wants to work with a spectral radius $\rho(\mathbf{W}) < 1$. If this condition is not verified, as well as degrading the performance of the reservoir, this can also lead to instabilities [15].

The matrices $\mathbf{W}$ and $\mathbf{W}^{\text{in}}$ can be rescaled, and this scaling can alter the performances of the reservoir. One should therefore design the PRC in such a way that those scaling factors are easily accessible experimentally. If one defines α and β, the feedback[3] and input gains, the input and connection matrices become:

$$\mathbf{W} \longrightarrow \alpha\mathbf{A}, \quad \mathbf{W}^{\text{in}} \longrightarrow \beta\mathbf{m} \tag{2.7}$$

Feedback and output The output signal of a PRC is a real scalar function $y(n)$, which means that the output matrix $\mathbf{W}^{\text{out}}$ becomes a real vector. Furthermore, the concatenation of $\mathbf{x}(n+1), \mathbf{u}(n+1), y(n)$ appearing in equation (2.3) is not used, but only the state vector $\mathbf{x}$, hence the fact that $\mathbf{W}^{\text{out}}$ is of dimension N. Regarding the feedback of the output into the reservoir, it is currently not doable in practice. This is due to the fact that, when running a RC experiment, some post-processing of the data needs to be performed on a computer in order to obtain the output.

With all these simplifications, equations (2.2) and (2.3) reduce to:

$$\mathbf{x}(n+1) = \mathbf{f}\left(\alpha\mathbf{A}\mathbf{x}(n) + \beta\mathbf{m}u(n+1)\right) \tag{2.8}$$

$$y(n+1) = f^{\text{out}}\left(\mathbf{W}^{\text{out}}\mathbf{x}(n+1)\right) \tag{2.9}$$

2.3.3 Neurons encoded in light intensity

There are two major families of TDM PRC. The first kind of PRC are those using optical components exhibiting nonlinear behaviour, such as Mach-Zehnder Modulator (MZM) [7, 18, 2], Semiconductor Optical Amplifier (SOA) [26] or semiconductor saturable absorber mirror [6] to couple the neurons. In an actual optical experiment, the measurements have to be done with photodiodes. These devices can only inform about the intensity of the light, which is the squared modulus of the phaser representation of the electric field, and not about the actual electric field. However, in the scheme presented above, the input and the activation level of the neurons are real valued functions appropriately encoded in the intensity of the light, and can therefore be directly read out by a photodiode, hence this simple expression for the output of the reservoir:

$$y(n+1) = \sum_{i=1}^{N} W_i^{\text{out}} x_i(n+1) = (\mathbf{W}^{\text{out}})^{\text{T}} \cdot \mathbf{x}(n+1) \tag{2.10}$$

[3]This may seem like a misnomer at this point since it has nothing to do with $\mathbf{W}^{\text{fb}}$, but this name is used because α acts as a gain for the activation level of the neurons being fed back into the reservoir.

2.3.4 Neurons encoded in phaser representation of the electric field

On the other hand, in [28], coherent light is used and the neurons are encoded in the complex phaser representation of the electric field and are linearly coupled using a delay line. In this scheme, the reservoir is linear, as can be seen on equation (2.12). This kind of RC is described with greater length because the new approach studied in this thesis relies on on a linear reservoir as well. In this equation, α and β are the feedback and input gains, respectively, A_0 is the input bias, ϕ is the phase acquired by the electric field during by going through the delay line, and k is the detuning parameter.

$$x_i(n+1) = \begin{cases} \alpha e^{j\phi} x_{i-k}(n) + \beta\left(m_i u(n) + A_0\right) & \text{if } k \leq i \leq N \\ \alpha e^{j\phi} x_{N+i-k}(n-1) + \beta\left(m_i u(n) + A_0\right) & \text{if } 0 \leq i \leq k \end{cases} \tag{2.11}$$

The equation can be rewritten in a compact way:

$$\mathbf{x}(n+1) = \alpha \mathbf{A}\mathbf{x}(n) + \beta(\mathbf{m}u(n+1) + \mathbf{A_0}) \tag{2.12}$$

It thus appears more clearly why the condition mentioned previously regarding $\rho\left(\alpha\mathbf{A}\right)$ is relevant. Indeed, by neglecting the inputs, if $\mathbf{x}_j$ is an eigenvector of $\alpha\mathbf{A}$ with eigenvalue $\lambda_j > 1$, the above equation will lead to an exponential divergence $\mathbf{x}_j(n) \sim \lambda_j^n \mathbf{x}_j(0)$. On the other hand, if λ_j is too small, the state $\mathbf{x}_j$ will be damped too quickly, deteriorating the short-term memory capabilities of the reservoir and preventing it from reaching the echo state.

In the linear reservoir, since the neurons and the input are inscribed in the complex electric field, the nonlinearity is introduced by the read out photodiodes. The output is therefore given by:

$$y(n+1) = \sum_{i=1}^{N} W_i^{\text{out}} |x_i(n+1)|^2 \tag{2.13}$$

From a historical point of view, it is interesting to notice that this reservoir shares features with the first artificial NN, the *perceptron*, introduced by F. Rosenblatt in 1958 [20], which also had a linear connection matrix, and a nonlinear output function.

Equations (2.10) and (2.13) give an interesting intuition on the meaning of the ESN. Indeed, as said previously, when a RC reaches the echo state, each of the neurons tends to systematically reproduce a modified version of the input, the actual modification being an individual characteristic of each neuron [10]. One can see this feature in the perspective of linear algebra. When the RC is fed with an input, it creates a set of time varying functions that can be seen as a basis in which one can try to express the output of the reservoir, which is a vector in the vectorial space of functions. This is why it is often said in the literature that a RC maps an input to a higher dimensional space. Therefore, one can in theory approach any target function with an arbitrary precision, depending on the number of neurons in the reservoir [10]. The higher the number of neurons, the closer to a genuine series development one gets.

2.3.5 Simulations

In this section, the performance of the linear reservoir with quadratic output from [28] are estimated with simulations. Two benchmark task are tackled, first NARMA10 and then nonlinear channel equalisation.

NARMA10

Nonlinear Auto-Regressive Moving Average (NARMA) 10 is a model often used to simulate chaotic time series because it exhibits similar properties [18]. If $u(n)$ is a random variable uniformly distributed along the interval $[-0.5, 0.5]$, the recurrent equation for NARMA10 reads:

$$\hat{y}(n+1) = 0.3\hat{y}(n) + 0.05\hat{y}(n)\left(\sum_{i=0}^{9}\hat{y}(n-i)\right) + 1.5u(n-9)u(n) + 0.1 \tag{2.14}$$

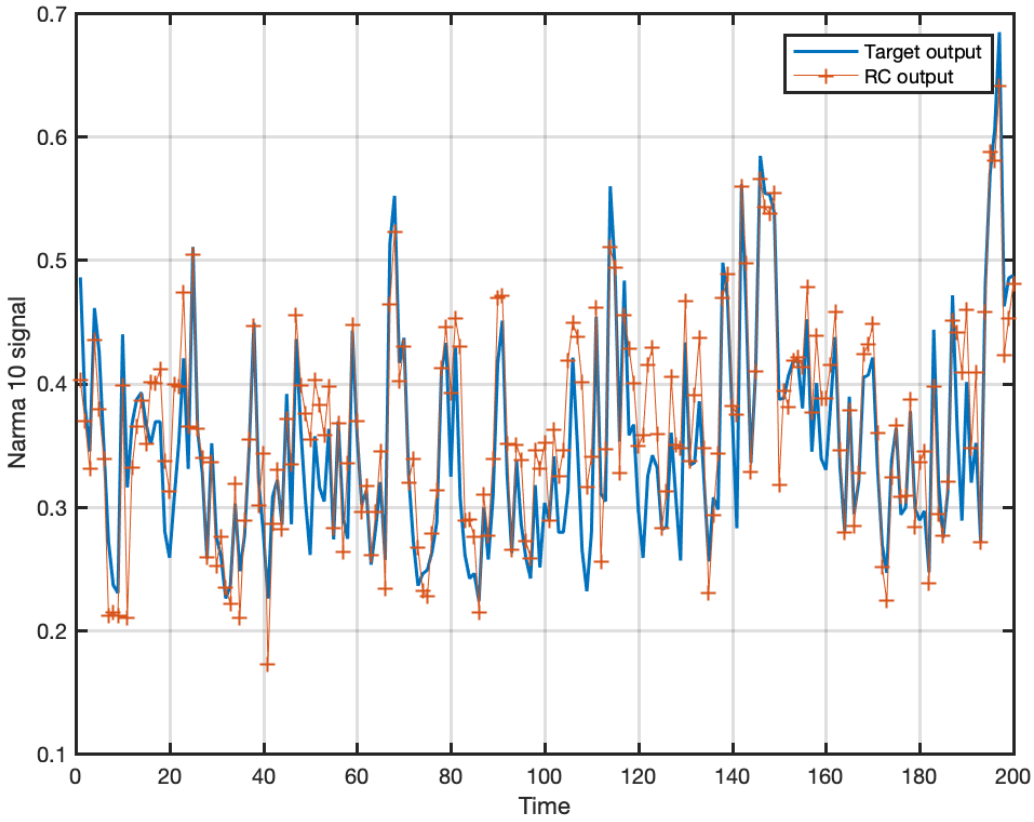


Figure 2.5: NARMA10 task with NMSE equal to 0.1541. This reservoir is made of 50 neurons. $\alpha = 0.5$, $k = 7$, $\phi = 0$ rad, $\beta = 1$, $A_0 = 1$, $\mathbf{m}$ is a random vector distributed between 0 and 1. The first 300 time steps were discarded in order to let enough time to the reservoir to enter the echo state (washout). Then the reservoir was trained for 3000 time steps and tested over 6000 time steps. This reservoir is particularly well suited for NARMA10 since the nonlinearities in the signal are mostly quadratic.

Nonlinear channel equalisation

This task consists in the reconstruction of a signal after it has travelled through a nonlinear channel. The emitted signal $\hat{y}$ is randomly drawn from the symbol set $\{-3, -1, 1, 3\}$. It is first superposed with following and preceding symbols as can be seen in (2.15). After that, a third degree polynomial transformation is applied to the mixed signals, and a Gaussian noise, whose intensity can be set in order to adjust the signal to noise ratio, is added in (2.16). This metric used to evaluate the performance of the reservoir for this type of task is the Signal Error Rate (SER) and is defined as the ratio between the number of erroneous symbols and the total number of symbols.

$$\begin{aligned} q(n) = {} & 0.08\hat{y}(n+2) - 0.12\hat{y}(n+1) + \hat{y}(n) + 0.18\hat{y}(n-1) \\ & - 0.1\hat{y}(n-2) + 0.091\hat{y}(n-3) - 0.05\hat{y}(n-4) \\ & + 0.04\hat{y}(n-5) + 0.03\hat{y}(n-6) + 0.01\hat{y}(n-7) \end{aligned} \tag{2.15}$$

$$u(n) = q(n) + 0.036q(n)^2 - 0.011q(n)^3 + \nu(n) \tag{2.16}$$

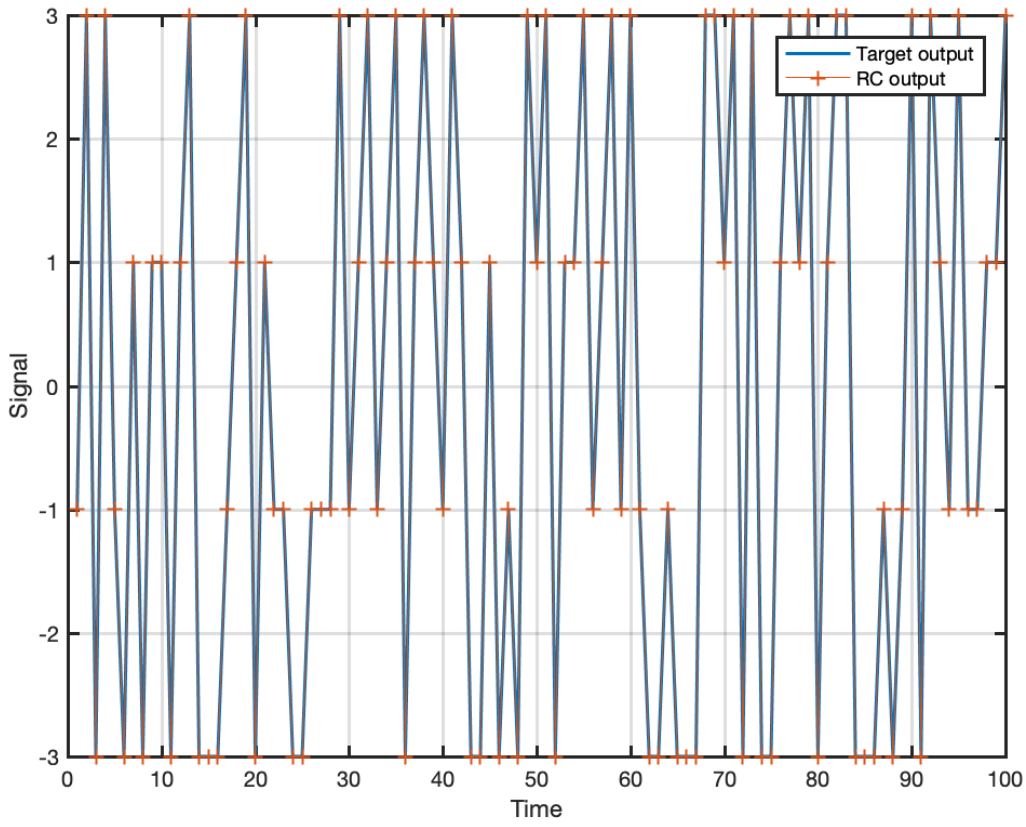


Figure 2.6: Nonlinear channel equalisation task with a signal error rate of 5 10^{-4}, with signal to noise ratio equal to 32 dB. This reservoir is made of 50 neurons. $\alpha = 0.5$, $k = 7$, $\phi = 0$ rad, $\beta = 1$, $A_0 = 1$, $\mathbf{m}$ is a random vector distributed between 0 and 1. The first 300 time steps were discarded in order to let enough time to the reservoir to enter the echo state (washout). Then the reservoir was trained for 3000 time steps and tested over 6000 time steps. The symbols predicted by the RC are found by changing the continuous valued output by the closest symbol.

Chapter 3

Photonic Reservoir Computer with Wavelength Division Multiplexed neurons

In this chapter, a novel approach of PRC is proposed, in which the neurons are no longer multiplexed in time, but in wavelength (or frequency) instead. This scheme was introduced in [1] and its end goal is to overcome the speed limitations imposed by the TDM of the neurons, as exposed in section 2.4. Indeed, by multiplexing the neurons in the frequency domain, the input signal can reach them all in the same time, there is no need to slow down its pace to let it alter the neurons sequentially. As an illustration, in [25], they proposed the first PRC with parallelisation of the neurons. In the experiment described in the paper, the neurons were spatially multiplexed. Doing so, the authors were able to reach data processing rates up to 12.5 GHz on tasks such as header recognition or boolean logic, which is an increase of one order of magnitude compared to those reported in [28]. This is encouraging for research in parallel PRC and motivates the study of the scheme explored in this thesis.

First, in section 3.1, a description of the working principle is given. It starts with a high-level overview of the scheme in which different features and components are presented. After that, attention is brought to the frequency coupling mechanism used to let the neurons interact between one another. It is shown that this can be achieved using a optoelectronic Phase Modulator (PM), even though this device has some practical drawbacks. Finally, the scheme is described with more details. In section **??**, the stabilisation issue, which is main topic of this thesis, is introduced.

3.1 Description of the scheme

In this section, the working principle of Wavelength Division Multiplexing (WDM) PRC is first given. This scheme takes advantage of the wave character of light and uses different wavelength to encode the neurons. It is fibre-based and relies on an optical cavity made of a fibre loop. Inside the resonator, the different neurons need to be able to interact, therefore one has to provide some coupling mechanism for the different frequencies. This issue is tackled in section 3.1.2 and provides interesting mathematical insights that are eventually used to derive a suitable model for this other kind of linear RC.

3.1.1 Working principle

The working principle of the new reservoir is schematised on figure 3.1. The whole setup is fibre-based, and the fibre used is a polarisation maintaining, single mode fibre.

Input The input time series $u(n)$ is converted into a continuous signal $\tilde{u}(t)$ by keeping its value constant during one period τ (sample and hold procedure), which leads to $\tilde{u}(t) = u(n)$, with $t \in [n\tau, (n+1)\tau[$. The continuous, coherent light emitted by a narrow band laser at frequency ω_0 is modulated in intensity by the input signal $\tilde{u}(t)$ using a MZM.

Cavity The modulated data is then coupled to a ring cavity around 20 metres long[1]. Inside this cavity, a PM mixes the different optical frequencies. More information about the PM can be found in section 3.1.2. Losses are introduced inside the cavity by two main sources, the intrinsic losses of the fibre and the insertion losses of the PM, hence the presence of an intra-cavity amplifier used to compensate for them. The neurons are encoded in the electric field evolving inside this cavity, which is schematised on the figure by the three different colours for the fibre loops. Therefore, from now on, the cavity is referred to as the *reservoir*. The holding time τ introduced earlier should be equal to the RTT of the reservoir. This is done in order to update the discrete dynamics of the RC each time the light completes a trip around the cavity. Doing so, one ensures that the value of the neuron x_j at discrete time $n+1$ depends on the values of the neurons x_i, with $i \in \{1, \dots, N\}$, at precisely time n.

Readout The electric field encoding the neurons exits the cavity through another coupler, and is then demultiplexed in frequency. Each of the frequency component is measured using a Photodiode (PD), which gives a value proportional to the squared modulus of the electric field, as explained in the previous chapter. The resulting signals are linearly combined using the trained output weights to produce the output of the reservoir $y(n)$.

$$y(n) = \sum_{i=1}^{N} W_i^{\text{out}} |x_i(n)|^2 \tag{3.1}$$

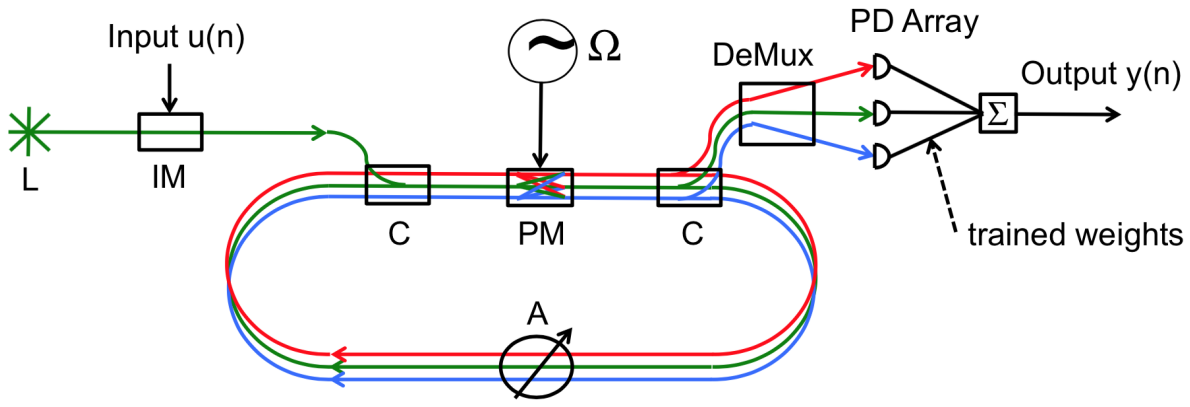


Figure 3.1: Schematic representation of the working principle of the WDM PRC [1]

[1]The length of the cavity can be derived from its Free Spectral Range (FSR), which is studied in the next chapter

Training and testing The training scheme used for this setup is the batch learning. Thus, to compute the output weights $\mathbf{W}^{\text{out}}$, one can use the ridge regression technique introduced in section 2.2.2. To do so, one needs to discard the first state vectors $\mathbf{x}$ because they correspond to the transient of the reservoir, and then to store them[2] as well as the desired outputs $\hat{y}$ during the whole learning period to create the matrices $\mathbf{X}$ and $\hat{\mathbf{Y}}$. After that, it is straightforward to solve equation (2.5) and to find the output weights. Once the RC is trained, it can move on to the testing phase, during which the outputs $y(n)$ are compared to the desired outputs. From that, one can quantify the performance of the RC using the appropriate metric, such as the NMSE for example.

3.1.2 Frequency coupling of the neurons

To mix the different frequencies present inside the reservoir a PM is used. This kind of device is well known for creating equidistant sidebands. Let $Ee^{i\omega t}$ be the input electric field entering the PM. Since a PM is an optoelectronic device, it needs to be externally driven by a voltage generator with a RF modulation signal $V_{\text{RF}}(t) = A\sin(\Omega t)$, with $\Omega \ll \omega$. Using the *Anger-Jacobi* expansion that allows to express the exponential of a trigonometric function in the basis of its harmonics, one can express the effect of a PM:

$$\begin{aligned} Ee^{i\omega t} &\xrightarrow{\Omega} Ee^{i\omega t}e^{im\sin(\Omega t)} \\ &= E\sum_{n=-\infty}^{\infty} J_n(m)e^{i(\omega+n\Omega)t} \end{aligned} \tag{3.2}$$

This formula indicates that the resulting wave is a discrete superposition of planar waves whose frequencies are evenly spaced ($\Omega/2\pi$ between each frequency component) and centred on the input electric field frequency $\omega/2\pi$. The coefficient $J_n(m)$ weighting the superposition is the n^{th} order Bessel function and m is the modulation depth. The latter is related to the modulation amplitude A, but the link between them is not straightforward and is experimentally investigated in the next chapter.

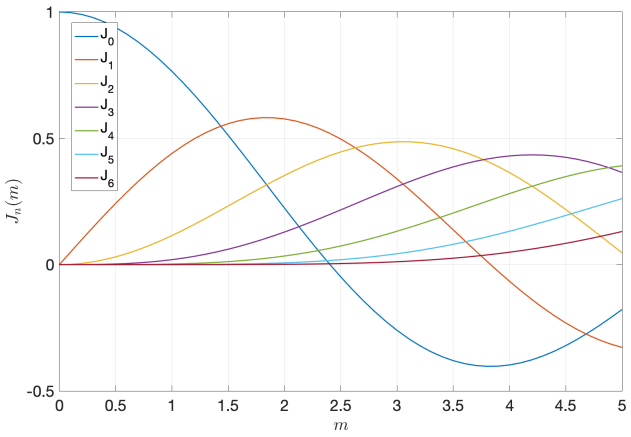


Figure 3.2: Bessel function of order $n \in \{0, \ldots, 6\}$

However, what can already be said about m is that it cannot exceed 2 in practice. This is due to the fact that when one increases the modulation frequency $\Omega/2\pi$, the modulation

[2]Since the output y is constructed using the squared modulus of the activation of the neurons, the matrix $\mathbf{X}$ is filled with $|x_i(n)|^2$ instead of what is indicated in section 2.2.2, namely $x_i(n)$

amplitude A cannot be brought to arbitrarily large values. As a first approximation regarding the RF generator, one could say that the product ΩA is upper bounded. On figure 3.2, one can observe the behaviour the Bessel functions $J_n(m)$ with $n \in \{0, \dots, 6\}$. For $m \in [0, 2]$, $J_n(m)$ becomes less significant for increasing order n. This provides an insight on an intrinsic limitation of this kind of RC, which is the limited number of neurons. A possible solution to improve the scheme would be to use a longitudinal multimode laser[3] to inject energy in a wider range of neurons. Yet, the coupling between distant neurons would still be very faint.

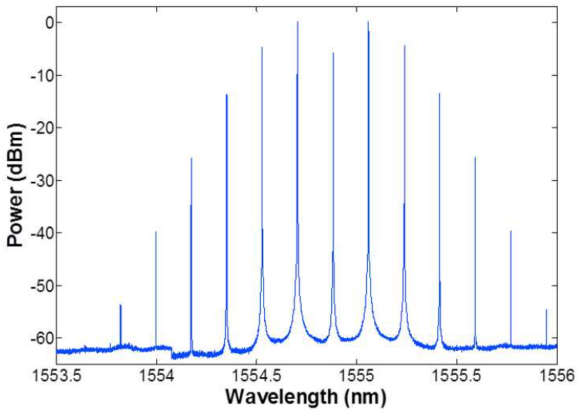


Figure 3.3: Experimental spectral density inside the reservoir [1]

On figure 3.3, the elements regarding the fast decrease of the intensity of the sidebands brought to light in the previous paragraph are confirmed experimentally. On this graph, one can see that only 13 neurons are usable in practice. One also notices that the spectrum is symmetric with respect to the central neuron. This can be explained by the fact that the height of the n^{th} peak is $\propto |J_n(m)|^2$, and by using a property of the Bessel functions that says that for $n \in \mathbb{Z}$, $J_{-n}(m) = (-1)^n J_n(m)$:

$$|J_{-n}(m)|^2 = |(-1)^n J_n(m)|^2 = |J_n(m)|^2 \tag{3.3}$$

3.1.3 Mathematical model

This section tackles the derivation of the discrete equation governing the dynamics of the reservoir. This calculation relies on the result stemming from the previous section regarding the effect of a PM on an incident electric field. As a first step, the state $\mathbf{x}$ is expressed using a convenient formalism. The end goal here is to be able to use linear algebra because it makes the derivation easier. Let N be the total number of neurons. This implies that $\mathbf{x}$ should be a N-dimensional vector that can developed as a linear combination of basis vectors:

$$\mathbf{x} = \sum_{i=1}^{N} x_i \hat{\mathbf{e}}_i \tag{3.4}$$

If the basis vectors are adequately chosen, the state vector $\mathbf{x}$ can reduce to the value of the electric field inside the reservoir. The basis vectors are defined as:

$$\hat{\mathbf{e}}_n = e^{i\omega_n t} = e^{i(\omega + n\Omega)t} \tag{3.5}$$

[3] A longitudinal multimode laser emits light at different frequencies

Moreover, to be consistent, the numbering of the neurons should be modified such that $\hat{\mathbf{e}}_0$ corresponds to the central frequency. Let η be the new variable used to iterate through the neurons such that the sum in the linear combination will go from $-\eta$ to η instead of from 1 to N. η is given by $\lfloor N/2 \rfloor$. The state vector $\mathbf{x}$ now reads:

$$\mathbf{x} = \sum_{i=-\eta}^{\eta} x_i \hat{\mathbf{e}}_i \tag{3.6}$$

This is precisely equal to the electric field inside the reservoir. The results from the previous section con now be used to derive the discrete dynamics equation. Knowing the value of each of the neurons at discrete time n and the effect of a PM, one can compute the new state vector. To do so, linear algebra is used. One knows that the reservoir is linear, therefore, it should be possible to find a linear mapping between two successive states. If there exists a linear mapping, since one is working in discrete space, there is a matrix representation of this mapping. Furthermore, given that one knows the effect of a PM on each of the basis vectors, one can explicitly define this matrix.

$$\mathbf{J} = \begin{bmatrix} J_0(m) & J_{-1}(m) & \dots & J_{-\eta}(m) & \dots & J_{-2\eta} \\ J_1(m) & J_0(m) & \dots & J_{-\eta+1}(m) & \dots & J_{-2\eta+1} \\ \vdots & \vdots & & \vdots & & \vdots \\ J_{2\eta}(m) & J_{2\eta-1}(m) & \dots & J_{\eta}(m) & \dots & J_0 \end{bmatrix} \tag{3.7}$$

One should note that each time the light goes around the cavity, it acquires some phase which is dependent on the wavelength. This should be taken into account in the derivation as well. If ϕ_i is the phase acquired by the neuron propagating with the frequency $\omega_i/2\pi$, one can define the phase matrix:

$$\mathbf{\Phi} = \begin{bmatrix} e^{i\phi_{-\eta}} & 0 & \dots & \dots & \dots & 0 \\ 0 & e^{i\phi_{-\eta+1}} & 0 & \dots & \dots & 0 \\ \vdots & 0 & \ddots & & & \vdots \\ \vdots & \vdots & & e^{i\phi_i} & & \vdots \\ \vdots & \vdots & & & \ddots & 0 \\ 0 & 0 & \dots & \dots & 0 & e^{i\phi_\eta} \end{bmatrix} \tag{3.8}$$

Both $\mathbf{J}$ and $\mathbf{\Phi}$ are $N \times N$ matrices, with associated linear operators $\hat{\mathbf{J}}$ and $\hat{\mathbf{\Phi}}$. Finally, recalling that α and β are the feedback and input gains, respectively, and that the source laser only injects light in the central wavelength, one can express the discrete time dynamics of the reservoir:

$$\mathbf{x}(n+1) = \alpha \hat{\mathbf{\Phi}} \hat{\mathbf{J}} \mathbf{x}(n) + \beta u(n+1) \hat{\mathbf{e}}_0 \tag{3.9}$$

A few things need to be mentioned. The value of α can be adjusted experimentally by changing the intra-cavity amplification and that of β numerically in the computer controlling the experiment. Finally, to be consistent with the model derived here, the figure 3.1, coming directly from [1], should be slightly modified. Indeed, the PM should be positioned before the input coupler and not between the input and readout coupler, because according to the figure, there should be no inhomogeneous term corresponding to the input signal.

Chapter 4

Interferometric stabilisation of reservoir cavity

4.1 Introduction

In this introductory section, the concept of interferometry is presented. As the name of the chapter suggests, this technique is used to stabilise the reservoir cavity. The reason why an optical cavity needs stabilisation will appear clearer later, but basically, this is due to the fact that light is a wave and that it can interfere with itself inside the cavity. The interferences can be constructive, destructive, or can behave in any intermediate way. Moreover, it will be shown that the interferometric properties are wavelength dependent. Since several wavelengths coexist inside the reservoir, this gives a first glimpse on the complexity entailing its stabilisation. To gain some insight on interferometry, and before moving on to the study of an actual ring cavity, the features of the well known Fabry-Perot (FP) interferometer are recalled. After that, it is shown that the properties studied for the FP can be translated to ring cavities with close to no modification. Finally, under the light of the basic notions of interferometry developed, the difficulties linked to the stabilisation of the reservoir cavity, which is at the heart of the scheme introduced in this thesis, are presented.

4.1.1 Fabry-Perot interferometer

The FP plays an important role in modern optics as it is really ubiquitous. This can be explained by the fact that, despite its great simplicity, it can reach good performance using high reflectivity mirrors, which can be produced using nowadays technologies. In practice, a FP cavity is simply made of two facing mirrors as can be viewed on figure 4.1. On this figure, one can see the two mirrors, represented by the vertical black lines, and the different electric fields. The resonance condition, namely the regime where the transmitted electric field E_t is maximum, can be seen intuitively as a situation where the intra-cavity field E_1 is in phase with the incident field E_in, which leads to the build up of a very intense intra-cavity electric field. On the other hand, the anti-resonance condition is met when E_in and E_1 are out of phase. The transmissivity of the FP interferometer, which is defined as the ratio $|E_\mathrm{t}|^2/|E_\mathrm{in}|^2$, is given by [19]:

$$\mathcal{T}(\omega) = \frac{1}{1 + \mathcal{F}\sin^2\left(\frac{\omega}{\mathrm{FSR}}\right)} \tag{4.1}$$

In this expression, $\mathcal{F}$ is the finesse of the cavity, ω is the angular frequency of the incident electric field, and FSR is the Free Spectral Range of the cavity. In a stationary

regime, the energy inside the cavity does not evolve, therefore the energy carried by the incident electric field E_{in} can either be transmitted or reflected, which implies that the reflectivity of the cavity which is defined as the ratio $|E_{\text{ref}}|^2/|E_{\text{in}}|^2$ is simply given by:

$$\mathcal{R}(\omega) = 1 - \mathcal{T}(\omega) \tag{4.2}$$

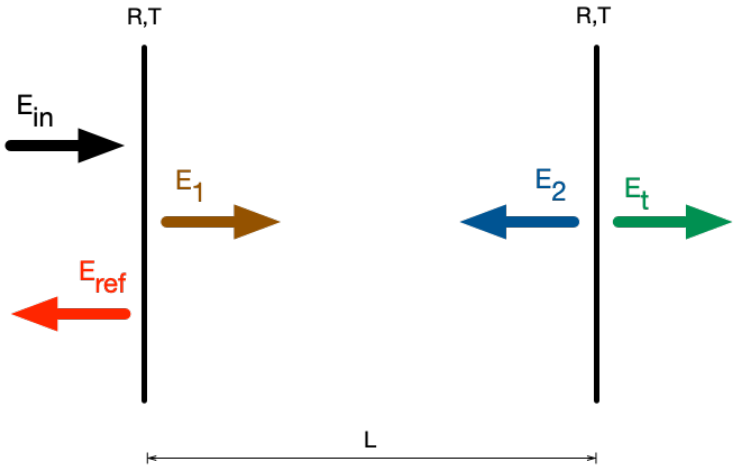


Figure 4.1: Schematic representation of a Fabry-Perot interferometer. E_{in} is the incident electric field, E_{ref} is the reflected electric field, E_{t} is the transmitted electric field, E_1 is the intra-cavity electric field propagating from left to right, E_2 is the intra-cavity electric field propagating from right to left, R and T are the reflectivity and transmissivity of the mirrors and L is the distance between them.

On figure 4.2, the transmissivity (right) and reflectivity (left) can be viewed. These graphs are made of peaks which are distant of the FSR in the spectral domain. Recalling that FSR $= c/2nL$, one can see that the FSR is linked to the length of the cavity, with c the speed of light and n the refractive index of the medium that could be present between the two mirrors (nL is the optical path). The finesse $\mathcal{F}$ is related to the width of the peaks and depends on the reflectivity of the mirrors as $\mathcal{F} = 4R/(1-R)^2$. As the reflectivity of the mirrors tends to 1, the finesse tends to infinity, and the peaks get infinitely narrow. On the other hand, with a lower reflectivity, more energy can leak out of the cavity even outside the resonance condition. Seeing the broadening of the peaks as energy leakage will be useful when drawing a parallel between FP and ring cavity interferometers.

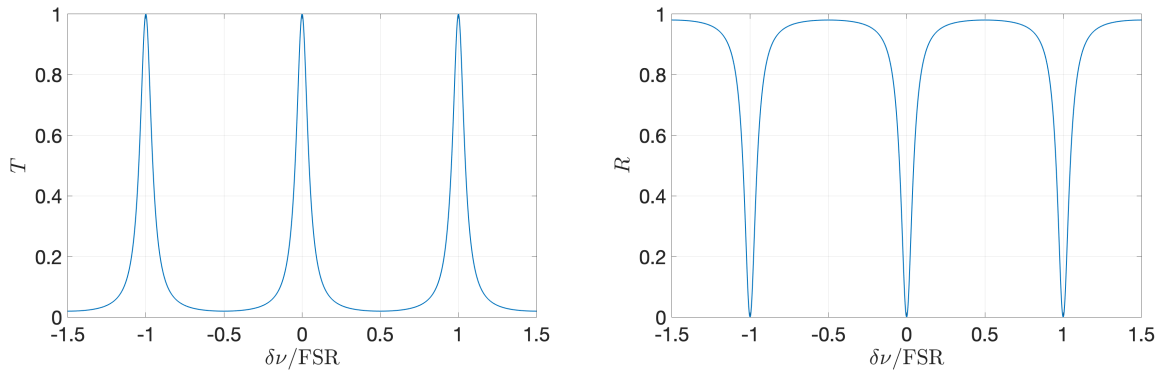


Figure 4.2: Transmissivity $\mathcal{T}$ (left) and reflectivity $\mathcal{R}$ (right) of the cavity. Finesse $\mathcal{F} = 50$. $\delta\nu$ denotes the deviation from a resonant frequency.

Equation (4.1) and (4.2) indicate that $\mathcal{T}$ and $\mathcal{R}$ depend on ω/FSR. This value can be rearranged as:

$$\frac{\omega}{\text{FSR}} = kL = \phi \tag{4.3}$$

Where k is the wave number defined as $n\omega/c$ and ϕ is the phase acquired by the electric field when propagating along the cavity. Therefore, by measuring the transmitted or reflected power, one can gain information about the phase of the electric field. This is the idea underlying interferometry.

4.1.2 Ring cavity

A ring cavity exhibits the same behaviour as a FP interferometer. The structure of a ring cavity interferometer is displayed on figure 4.3. This is a fibre-based setup in which the incident electric field E_{in} penetrates the cavity from the left through a coupler. E_{ref} denotes the electric field exiting the cavity and E_1 and E_2 refer to the fields entering and leaving the fibre loop, respectively. The nomenclature for the fields was chosen in such a way that the analogy with the FP cavity can be understood. Indeed, one could see the coupler acting as the leftmost mirror from the figure 4.1, and the fibre loop as the one on the right side because it turns E_1 into E_2 and dissipates energy through fibre losses, whereas for the mirror it was by leakage out of the cavity.

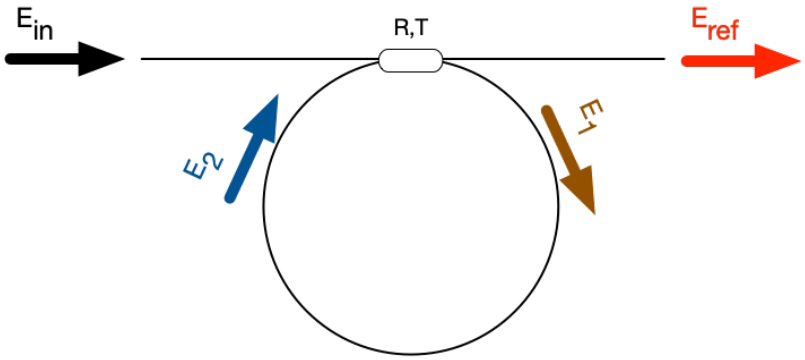


Figure 4.3: Schematic view of a ring cavity

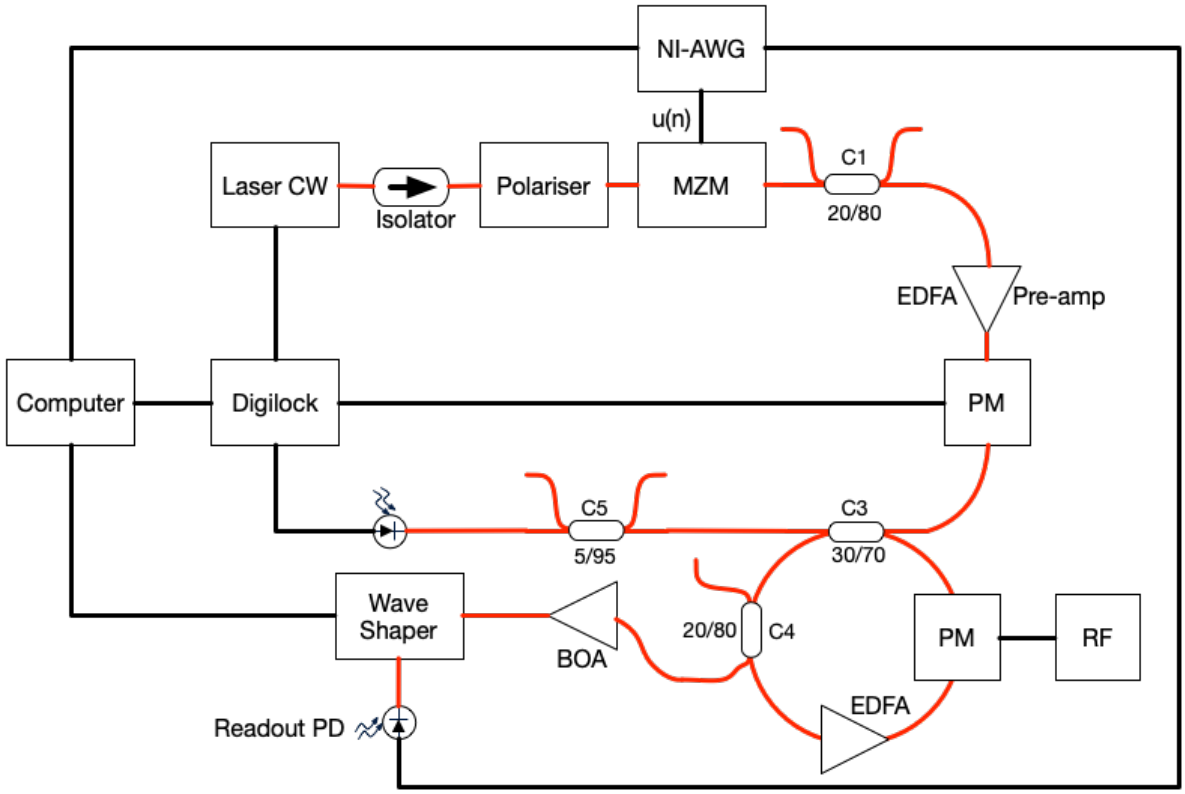
Because of the similarities between ring cavities and FPs, the former show the same transmissivity and reflectivity as the latter. Therefore, by measuring the reflected power, one can determine the phase acquired by the electric field inside the cavity.

4.1.3 Challenge

The basic principle of interferometry has been introduced through the presentation of the FP interferometer, and in the discussion that followed, it has been shown that a ring cavity, such as the reservoir cavity studied in this thesis, can be used as an interferometer. Moreover, it has also been showed that $\mathcal{R}$ depend on the frequency (or wavelength) and on the length of the cavity and that the phase acquired by the incident electric field after one round trip could be determined by studying the reflected power.

In the reservoir cavity, many different wavelengths coexist because they encode the different neurons. Furthermore, after each round trip, phase acquired by each neuron should be a constant, as shown on equation (3.9). Recalling the phase matrix $\mathbf{\Phi}$, the phase acquired by the i^{th} neuron is given by Φ_{ii}.

4.2 Experimental setup



4.3 Characterisation of the reservoir

4.3.1 Introduction

4.3.2 Transfer function of the cavity

Mathematical model

Simulations

Experimental results

4.3.3 Effective losses

4.3.4 Modulation depth

4.4 Pound-Drever-Hall stabilisation technique

4.4.1 Introduction

4.4.2 Error function

Mathematical model

Simulation

Experimental results

4.5 Characterisation of the stabilisation performance for different regimes

4.5.1 Introduction

4.5.2 Approach

4.5.3 Results

Chapter 5

Conclusion

Acronyms

ESN Echo State Network 7, 8, 14

FP Fabry-Perot 22–24

FSR Free Spectral Range 18, 22, 23

ML Machine Learning 7, 8, 10

MZM Mach-Zehnder Modulator 13, 18

NARMA Nonlinear Auto-Regressive Moving Average 15

NMSE Normalised Mean Square Error 11, 15, 19

NN Neural Network 7, 8, 12, 14

PD Photodiode 18

PM Phase Modulator 17–21

PRC Photonic Reservoir Computing 11–13, 17, 18

RC Reservoir Computer 7–12, 14, 16–20

RC Reservoir Computing 7, 10, 13, 14

RNN Recurrent Neural Network 7

RTT Round-Trip Time 12, 18

SER Signal Error Rate 16

SOA Semiconductor Optical Amplifier 11, 13

TDM Time Division Multiplexing 7, 11–13, 17

WDM Wavelength Division Multiplexing 17, 18

Bibliography

[1] A. Akrout et al. "Parallel photonic reservoir computing using frequency multiplexing of neurons". In: (2016).

[2] Piotr Antonik et al. "Online Training of an Opto-Electronic Reservoir Computer Applied to Real-Time Channel Equalization". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11 (Nov. 2017), pp. 2686–2698. DOI: 10.1109/tnnls.2016.2598655. URL: https://doi.org/10.1109/tnnls.2016.2598655.

[3] A. Bernal, S. Fok, and R. Pidaparthi. "Financial Market Time Series Prediction with Recurrent Neural Networks". In: (2012). URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.278.3606&rep=rep1&type=pdf.

[4] Christopher Bishop. *Pattern recognition and machine learning*. New York: Springer, 2006. ISBN: 978-0387-31073-2.

[5] Daniel Brunner and Ingo Fischer. "Reconfigurable semiconductor laser networks based on diffractive coupling". In: *Optics Letters* 40.16 (Aug. 2015), p. 3854. DOI: 10.1364/ol.40.003854. URL: https://doi.org/10.1364/ol.40.003854.

[6] Antoine Dejonckheere et al. "All-optical reservoir computer based on saturation of absorption". In: *Optics Express* 22.9 (Apr. 2014), p. 10868. DOI: 10.1364/oe.22.010868. URL: https://doi.org/10.1364/oe.22.010868.

[7] François Duport et al. "Fully analogue photonic reservoir computer". In: *Scientific Reports* 6.1 (Mar. 2016). DOI: 10.1038/srep22381. URL: https://doi.org/10.1038/srep22381.

[8] Chrisantha Fernando and Sampsa Sojakka. "Pattern Recognition in a Bucket". In: *Advances in Artificial Life*. Springer Berlin Heidelberg, 2003, pp. 588–597. DOI: 10.1007/978-3-540-39432-7_63. URL: https://doi.org/10.1007/978-3-540-39432-7_63.

[9] Alireza Goudarzi et al. "A Comparative Study of Reservoir Computing for Temporal Signal Processing". In: *CoRR* abs/1401.2224 (2014).

[10] H. Jaeger. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". In: *Science* 304.5667 (Apr. 2004), pp. 78–80. DOI: 10.1126/science.1091277. URL: https://doi.org/10.1126/science.1091277.

[11] H. Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. 2001.

[12] Herbert Jaeger. "Adaptive Nonlinear System Identification with Echo State Networks". In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 609–616. URL: http://dl.acm.org/citation.cfm?id=2968618.2968694.

[13] Herbert Jaeger. "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach". In: *GMD-Forschungszentrum Informationstechnik, 2002*. 5 (Jan. 2002).

[14] Herbert Jaeger et al. "Optimization and applications of echo state networks with leaky- integrator neurons". In: *Neural Networks* 20.3 (Apr. 2007), pp. 335–352. DOI: 10.1016/j.neunet.2007.04.016. URL: https://doi.org/10.1016/j.neunet.2007.04.016.

[15] M. Lukoševičius and H. Jaeger. "Reservoir computing approaches to recurrent neural network training". In: *Computer Science Review* 3.3 (Aug. 2009), pp. 127–149. DOI: 10.1016/j.cosrev.2009.03.005. URL: https://doi.org/10.1016/j.cosrev.2009.03.005.

[16] M. Lukoševičius, M. Jaeger, and B. Schrauwen. "Reservoir Computing Trends". In: *KI - Künstliche Intelligenz* 26.4 (May 2012), pp. 365–371. DOI: 10.1007/s13218-012-0204-5. URL: https://doi.org/10.1007/s13218-012-0204-5.

[17] Marvin Minsky. *Perceptrons; an introduction to computational geometry*. Cambridge, Mass: MIT Press, 1969. ISBN: 9780262130431.

[18] Y. Paquot et al. "Optoelectronic Reservoir Computing". In: *Scientific Reports* 2.1 (Feb. 2012). DOI: 10.1038/srep00287. URL: https://doi.org/10.1038/srep00287.

[19] A. Perot and Charles Fabry. "On the Application of Interference Phenomena to the Solution of Various Problems of Spectroscopy and Metrology". In: *The Astrophysical Journal* 9 (Feb. 1899), p. 87. DOI: 10.1086/140557. URL: https://doi.org/10.1086/140557.

[20] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain". In: *Psychological Review* (1958), pp. 65–386.

[21] Stuart Russell. *Artificial intelligence : a modern approach*. Upper Saddle River, New Jersey: Prentice Hall, 2010. ISBN: 978-0-13-604259-4.

[22] Guy Van der Sande, Daniel Brunner, and Miguel C. Soriano. "Advances in photonic reservoir computing". In: *Nanophotonics* 6.3 (Jan. 2017). DOI: 10.1515/nanoph-2016-0132. URL: https://doi.org/10.1515/nanoph-2016-0132.

[23] Benjamin Schrauwen, David Verstraeten, and Jan Campenhout. "An overview of reservoir computing: Theory, applications and implementations". In: Jan. 2007, pp. 471–482.

[24] Fabian Triefenbach et al. "Phoneme Recognition with Large Hierarchical Reservoirs". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 2307–2315. URL: http://papers.nips.cc/paper/4056-phoneme-recognition-with-large-hierarchical-reservoirs.pdf.

[25] Kristof Vandoorne et al. "Experimental demonstration of reservoir computing on a silicon photonics chip". In: *Nature Communications* 5.1 (Mar. 2014). DOI: 10.1038/ncomms4541. URL: https://doi.org/10.1038/ncomms4541.

[26] Kristof Vandoorne et al. "Toward optical signal processing using Photonic Reservoir Computing". In: *Optics Express* 16.15 (July 2008), p. 11182. DOI: 10.1364/oe.16.011182. URL: https://doi.org/10.1364/oe.16.011182.

[27] D. Verstraeten, B. Schrauwen, and D. Stroobandt. "Reservoir-based techniques for speech recognition". In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006. DOI: 10.1109/ijcnn.2006.246804. URL: https://doi.org/10.1109/ijcnn.2006.246804.

[28] Quentin Vinckier et al. “High-performance photonic reservoir computer based on a coherently driven passive cavity”. In: *Optica* 2.5 (Apr. 2015), p. 438. DOI: 10.1364/optica.2.000438. URL: https://doi.org/10.1364/optica.2.000438.