

# Nanophotonic Reservoir Computing With Photonic Crystal Cavities to Generate Periodic Patterns

Martin Andre Agnes Fiers, *Member, IEEE*, Thomas Van Vaerenbergh, *Student Member, IEEE*,  
Francis Wyffels, David Verstraeten, Benjamin Schrauwen, Joni Dambre, *Member, IEEE*,  
and Peter Bienstman, *Member, IEEE*

**Abstract**—Reservoir computing (RC) is a technique in machine learning inspired by neural systems. RC has been used successfully to solve complex problems such as signal classification and signal generation. These systems are mainly implemented in software, and thereby they are limited in speed and power efficiency. Several optical and optoelectronic implementations have been demonstrated, in which the system has signals with an amplitude and phase. It is proven that these enrich the dynamics of the system, which is beneficial for the performance. In this paper, we introduce a novel optical architecture based on nanophotonic crystal cavities. This allows us to integrate many neurons on one chip, which, compared with other photonic solutions, closest resembles a classical neural network. Furthermore, the components are passive, which simplifies the design and reduces the power consumption. To assess the performance of this network, we train a photonic network to generate periodic patterns, using an alternative online learning rule called first-order reduced and corrected error. For this, we first train a classical hyperbolic tangent reservoir, but then we vary some of the properties to incorporate typical aspects of a photonics reservoir, such as the use of continuous-time versus discrete-time signals and the use of complex-valued versus real-valued signals. Then, the nanophotonic reservoir is simulated and we explore the role of relevant parameters such as the topology, the phases between the resonators, the number of nodes that are biased and the delay between the resonators. It is important that these parameters are chosen such that no strong self-oscillations occur. Finally, our results show that for a signal generation task a complex-valued, continuous-time nanophotonic reservoir outperforms a classical (i.e., discrete-time, real-valued) leaky hyperbolic tangent reservoir (normalized root-mean-square errors = 0.030 versus NRMSE = 0.127).

**Index Terms**—Integrated optics, optical neural network, pattern generation, photonic reservoir computing, supervised learning.

Manuscript received August 16, 2012; revised June 4, 2013; accepted July 20, 2013. Date of publication August 15, 2013; date of current version January 10, 2014. This work was supported in part by the Interuniversity Attraction Pole Photonics@be program of the Belgian Science Policy Office and the ERC NaResCo Starting grant. The work of M. Fiers was supported by a research fund of Ghent University. The work of T. Van Vaerenbergh was supported by the Flemish Research Foundation under a Ph.D. grant.

M. A. A. Fiers, T. Van Vaerenbergh, and P. Bienstman are with the Department of Information Technology of Ghent University, Ghent 9000, Belgium, and also with the Center for Nano and Biophotonics, Ghent University, Ghent B-9000, Belgium (e-mail: martin.fiers@intec.ugent.be; thomas.vanvaerenbergh@intec.ugent.be; peter.bienstman@ugent.be).

F. Wyffels, D. Verstraeten, B. Schrauwen, and J. Dambre are with the ELIS Department, Ghent University, Ghent 9000, Belgium (e-mail: francis.wyffels@ugent.be; david.verstraeten@ugent.be; benjamin.schrauwen@ugent.be; joni.dambre@ugent.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2013.2274670

## I. INTRODUCTION

RESERVOIR computing (RC) is a recently proposed methodology in the field of machine learning [1], [2]. Originally, it combined a recurrent neural network (RNN) with a linear readout layer, trained on the neuron states. The RNN itself, usually consisting of nonlinear neurons, is left untrained. It transforms the input signal to a high-dimensional state space, containing many linear and nonlinear transformations of the input history. Only a few global parameters, such as the overall gain in the system, the magnitude of the inputs, and the network size are optimized to tune the reservoir dynamics into an appropriate regime for the task at hand. RC performs very well on a variety of tasks, e.g., speech recognition [3], detection of epileptic seizures [4], robot localization [5], and time series prediction [1], [6].

Typically, a reservoir is implemented as a sequential algorithm in software. This obviously limits the speed of operation of the reservoir. The inherent parallelism of the neural network concept makes it a good candidate for parallel hardware implementations. In particular, analog hardware platforms in which the nonlinearity is provided by the physics of the components can be very efficient. A good overview of hardware implementations of RC can be found in [7]. In the electronic domain, there are implementations such as an FPGA [3] and VLSI [8]. Recently, photonic and optoelectronic implementations of RC have been demonstrated [9], [10]. An advantage of photonic implementations is the fact that they can be operated in a coherent regime, where all signals have both an amplitude and a phase, which interact whenever signals are combined. This increases the dimensionality of the state space. Even when the number of observed signals remains unchanged (e.g., only the magnitudes of the neuron states), it generally enriches the realized projection of the input into the observed portion of state space, yielding better reservoirs. Furthermore, the very high carrier frequency of optical signals, which can be carried over dielectrical materials with low loss, allows the system to process signals faster, and potentially with lower power, than electronic implementations.

Vandoorne *et al.* [10] demonstrated, through simulations, that RC with an integrated circuit of semiconductor optical amplifiers (SOAs) as a reservoir can solve a speech recognition task with state-of-the-art performance. These components are, however, relatively large and, because they are active amplifiers, the network consumes a lot of power. Here, we

propose to use an integrated circuit of photonic crystal cavities as alternative to the SOA network. These are passive components, which drastically reduces the power budget, and they are easier to fabricate because no postprocessing is involved (no electrical pads, no additional bonding, and electrical wiring).

In contrast to previous paper, we evaluate the proposed physical reservoir in continuous time, i.e., without the external clock determined by the sampling period. To fully evaluate the potential of continuous-time reservoirs, we also used a continuous-time task: the multiple superimposed oscillator (MSO) problem [11], in which the reservoir is used to predict the evolution of a superposition of two or more sinusoidal waves with different frequencies.

The output weights are adjusted online during training, using a technique called first-order reduced and corrected error (FORCE) [12]. This online learning technique has two distinct advantages. First, adjustments to the weights consider the dynamics and the feedback of the network during training. This is used to stabilize the reservoir. Especially for resonant nonlinear cavities, in which it is usually difficult to control the dynamics, this is a significant advantage. When trained correctly, the system with feedback is very robust and stable, even when the feedback has a considerable delay. Second, it is an online training technique, in contrast to the traditional offline learning techniques, which require the recording of long time traces for all state variables before training. This, and the fact that it does not depend on extensive matrix operations, makes it a good candidate for future analog implementations of the RC training stage.

To summarize, we assess the performance of a continuous-time low-power photonic crystal cavity reservoir, using the FORCE learning technique, on a continuous-time task. In several intermediate steps, we increase the complexity from a discrete-time hyperbolic tangent reservoir to a continuous-time complex-valued photonic crystal cavity reservoir. We discuss the impact of the most relevant design parameters, e.g., the phase difference between connected resonator pairs, the total number of nodes and the number of biased nodes and the delay between the nodes, and identify the optimal parameter ranges. We find that an optical reservoir performs particularly well on the MSO task and even outperforms the classical hyperbolic tangent reservoir. Finally, we link our results to the total information processing capacity of the photonic crystal cavity reservoir for the most relevant parameters, in which the same trends are observed.

The rest of this paper is structured as follows. The RC setup and training method are described in Section II, and the experimental setup is briefly described in Section III. In Section IV, we investigate the difference between a classical discrete-time reservoir and a continuous-time reservoir. Then, we investigate the benefits of using complex-valued states in Section V. In Section VI, we present an optical reservoir on an integrated photonics platform. Additional challenges are addressed, such as the feedback loop, the readout layer, and the weight recalculation. Finally, in Section VII, we show the results of simulations for the optical reservoir using photonic crystal cavities.

## II. RC

### A. Reservoir and the Readout Layer

Generally, a reservoir is simulated in discrete time.<sup>1</sup> The states of a reservoir of size  $N$  at time step  $k$  are then given by  $\mathbf{x}[k]$ ,  $k \in \mathbb{N}$ .  $\mathbf{x}[k]$  is a column vector with dimensions  $(N, 1)$ . The next state of the reservoir is calculated using the following equation:

$$\mathbf{x}[k + 1] = (1 - \lambda)\mathbf{x}[k] + \lambda\mathbf{f}(\mathbf{W}_{\text{in}}\mathbf{u}[k] + \mathbf{W}_{\text{res}}\mathbf{x}[k]) \quad (1)$$

where  $\mathbf{W}_{\text{in}}$  ( $N, K$ ) is the weight matrix to feed  $K$  inputs to the reservoir and  $\mathbf{W}_{\text{res}}$  ( $N, N$ ) is the connection matrix of the reservoir with  $N$  neurons.  $\mathbf{f}$  is the (usually nonlinear) activation function. In echo state networks [1], one of the most commonly used types of discrete-time reservoirs,  $\mathbf{f}$  is a sigmoidal or a hyperbolic tangent function.

For many tasks, it is beneficial that the reservoir operates close to the edge of stability. The spectral radius, a measure for the maximal gain in a reservoir, is often used as an indicator for reservoir stability. It is defined as the largest eigenvalue of the system's Jacobian at its maximal gain state. For a reservoir with hyperbolic tangent neurons, the maximal node gain equals one, therefore the spectral radius can be simplified to

$$\text{SR} = \max(|\text{eig}(\mathbf{W}_{\text{res}})|). \quad (2)$$

In a nonlinear reservoir, the actual gain at each point in time depends on the operating point. The stronger the nonlinearities are driven, the smaller the actual gain.

### B. Training

An RC system can be trained both online and offline. For offline learning, all input is first fed to the reservoir, and the states are recorded into a state matrix. The output weights are then calculated so that the output matches a desired pattern, usually using some form of regularized regression. The need to record reservoir states and the mathematical operations required for training make it difficult to implement offline learning using analog hardware.

In contrast, in online learning, the output weights  $\mathbf{W}_{\text{out}}$  are modified during the training. At regular time intervals, updates are computed from the observed error. Such learning schemes are more easily adaptable to analog hardware implementations, which is why we decided to evaluate a recent and very successful online learning technique, called FORCE learning [12] for training our photonic reservoir architecture. FORCE was proposed as a training technique for tasks in which the system is required to autonomously generate some kind of periodic pattern. Such tasks are particularly difficult, because they require that the reservoir output is fed back into the system. This implies that the system needs to be robust against errors or noise that is being fed back. For such tasks, the desired output is often fed into the system during training (teacher forcing), with some noise added to achieve robustness against noise. This robustness is, however, very limited.

<sup>1</sup>For the rest of this paper, the time variable for a discrete-time signal is surrounded by square brackets. In a continuous-time signal, we will use round brackets.

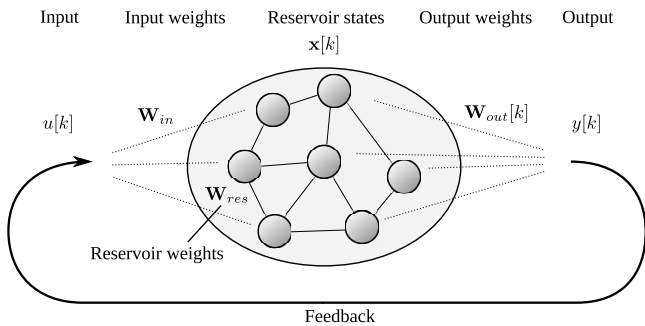


Fig. 1. Reservoir with feedback. The reservoir states  $\mathbf{x}[k]$  weighted by the output weights  $\mathbf{W}_{out}$  to produce the output  $y[k]$ . This output is then fed back to the input. The purpose of this network is to autonomously generate periodic patterns after training  $\mathbf{W}_{out}$  using the rule described in [12].

The FORCE technique succeeds in training the system with the actual output feedback in place. The readout weights are adjusted online, e.g., using the recursive least squares (RLS) modification rule. This way, the training considers the reservoir dynamics and dynamic trajectories, including useful basins of attraction, can be learned. This yields greater robustness and stability, even when delay is present in the feedback loop. This is particularly interesting for hardware implementations, as some delay is always present in the physical systems.

The steps to perform learning are summarized as follows (Fig. 1).

- 1) Simulate the reservoir from  $\mathbf{x}[k]$  to  $\mathbf{x}[k + 1]$ .
- 2) Calculate the output  $y[k + 1] = \mathbf{W}_{out}[k] \mathbf{x}[k + 1]$ .
- 3) Calculate the error with respect to the target signal  $f : e[k + 1] = y[k + 1] - s[k + 1]$ .
- 4) From this error, calculate the new weights  $\mathbf{W}_{out}[k + 1]$  using the RLS rule explained in [12].
- 5) Feedback the output  $y[k + 1]$  to the reservoir for the next time step.

FORCE is usually applied to reservoirs, which initially exhibit chaotic behavior. For the hyperbolic tangent ESNs, this condition is usually met when the spectral radius is considerably larger than one. The training stabilizes the dynamical system consisting of the reservoir and the output feedback.

FORCE learning is typically used in a training and evaluation setup, and is shown in Fig. 2. To be able to scale this learning approach to periodic signal generation tasks with different time scales, the lengths of all training stages are expressed as multiples of  $T_1$ , the period of the slowest frequency component in the signal that is to be generated. Below, we summarize these steps.

- 1) *Warm-Up* ( $15T_1$ ): The initial state  $\mathbf{x}[0]$  is chosen zero. The input during the warm-up is noise, sampled from a standard normal distribution with amplitude one.
- 2) *Training* ( $400T_1$ ): The output weights are adjusted using the proposed RLS rule [12].
- 3) *Free-Run* ( $2000T_1$ ): The output weights are unmodified. If the training converges, the RC system with feedback can now autonomously generate the desired function  $s[k]$ .

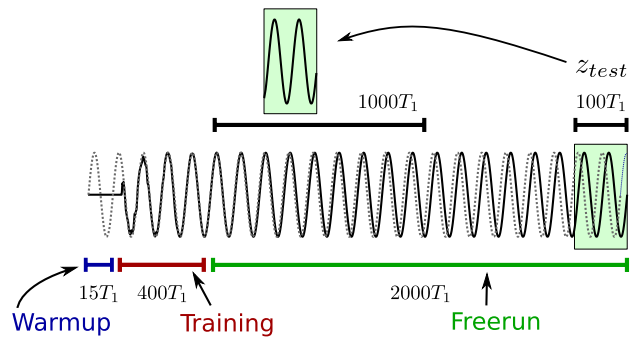


Fig. 2. Simplified illustration of the learning sequence.  $T_1$  is the period of the lowest frequency component of the signal that must be generated. During warm-up, the input of the reservoir is noise, sampled from a uniform distribution. During training, the output weights  $\mathbf{W}_{out}$  are adjusted such that the output (black solid line) follows the target signal (gray dashed line). The output weights are unmodified during free-run. The output can have a slightly different frequency than the target. The last samples  $y_{test}[k]$  are scrolled over a window of the free-run output, each time calculating the NRMSE. The optimal value of the NRMSE is used as performance for this learning sequence.

### III. EXPERIMENTAL SETUP

#### A. Task Description

As this paper aims to evaluate a photonic reservoir implementation—inherently operating in continuous time—trained with FORCE learning, we selected the MSO task [11], a pattern generation task defined in continuous time. This academic task has previously been used to benchmark the performance for the different types of reservoirs. The RC system has to generate a superposition of sine waves with harmonically unrelated frequencies

$$s(t) = \sin(\omega_1 t) + \sin(\omega_2 t). \quad (3)$$

The pulsations of the signals are:  $\omega_1 = 0.2/s$  for the lower frequency and  $\omega_2 = 0.311/s$  for the higher frequency. When this task is used in a discrete-time setup, the target signal is sampled at  $s[k] = s(k\Delta t)$ ,  $k \in \mathbb{N}$ . The period of the first signal is  $T_1 = 2\pi/\omega_1 \simeq 31.42$  s, the period of the second signal is  $T_2 = 2\pi/\omega_2 \simeq 20.20$ . For a classical discrete-time reservoir,  $\Delta t = 1$  s, and we continue using the notation  $s[k] = s(k \cdot 1$  s). The period of the superimposed signal is very long, which increases the challenge of learning the signal.

#### B. Simulation Software

All simulations were performed using a combination of two software packages: 1) OGER and 2) Caphe. OGER, the OrGanic Environment for RC [13], extends the Python MDP toolbox [14], by providing extra functionality in the context of RC. It is used to setup and postprocess RC experiments and to initiate the reservoir simulations. Caphe [15], a simulation framework for arbitrary nonlinear circuits, is used for the actual continuous-time reservoir simulations.

#### C. Performance Measure

Reservoir performance is evaluated throughout this paper by the normalized root-mean-square error (NRMSE) between the output and the target function at the sampling times  $k\Delta t$ .

In signal generation tasks, an important aspect of performance is the stability of the generated output over longer periods of time. Therefore, after the training phase, we allow the reservoir run for some time ( $2315T_1$ ) before calculating the performance. In addition, for periodic signals, a small phase shift between the actual and the desired output is usually acceptable, but it strongly affects the NRMSE. We use the approach of [16], where we calculate the NRMSE for windows of the output signal  $y_{\text{test}}[k] = y[2315T_1 + k]$ ,  $k \in [0, 100T_1]$ , and sliding these windows over a sufficiently long section ( $1000T_1$ ) of the free-run stage. Selecting the minimal value across all window positions effectively evaluates the shape of the desired output, largely cancelling out the impact of any phase shifts. This is shown in Fig. 2. Although computationally intensive, this calculation can be sped up using a convolution.

#### IV. CONTINUOUS-TIME RESERVOIR

The optical network is inherently a continuous-time system, and the MSO task is inherently a continuous-time task, therefore as an intermediate step toward simulating a full photonic reservoir, we initially investigate the performance of a leaky hyperbolic tangent reservoir as a continuous-time system.

Traditional ESNs are defined in discrete time. We can, however, view the neuron update equation (1) as the result of applying Euler integration on ordinary differential equations (ODEs). To more closely resemble this form, we now substitute  $\lambda$  by  $\Delta t/\tau_0$ , where  $\tau_0$  is the dominant time constant of the neuron. In this case, we assume  $\Delta t = 1$  s. To simplify the notation, we omit the explicit argument of  $\mathbf{f}$ , and instead we write  $\mathbf{f}(\mathbf{v})$

$$\mathbf{x}(t + \Delta t) = \left(1 - \Delta t \frac{1}{\tau_0}\right) \mathbf{x}(t) + \Delta t \frac{1}{\tau_0} \mathbf{f}(\mathbf{v}(t)).$$

Restructuring and taking the limit for  $\Delta t \rightarrow 0$  yield the equivalent continuous-time leaky hyperbolic tangent ODE equations

$$\frac{d(\mathbf{x})}{dt} = \frac{1}{\tau_0} (-\mathbf{x}(t) + \mathbf{f}(\mathbf{v}(t))). \quad (4)$$

Note that  $\mathbf{v}(t)$  also needs to be adapted to explicitly include the interconnection delays in the network. In a discrete-time RC system there is inherently a delay of 1 s between the neurons. To most closely resemble this classical case, all delays are in this case equal to the original discrete time step of  $\Delta t = 1$  s

$$\mathbf{v}(t) = \mathbf{W}_{\text{in}} \mathbf{u}(t) + \mathbf{W}_{\text{res}} \mathbf{x}(t - \Delta t).$$

In a more general setting, a delay on the input connections can also be included.

Table I lists the parameters that are used throughout this paper for the leaky hyperbolic tangent reservoirs. Any deviations from these parameters will be explicitly mentioned. We simulate the continuous-time hyperbolic tangent reservoir using the Bulirsch–Stoer integration method with variable step size  $\Delta t'$ , and using a relative accuracy of  $10^{-8}$ . The feedback is also simulated in continuous time, but without delay. Learning is still performed at discrete time steps of  $\Delta t = 1$  s.

TABLE I  
DEFAULT VALUES USED IN THE LEAKY HYPERBOLIC  
TANGENT RESERVOIR

Name	Description	Default value
N	Number of neurons	200
SR	Spectral radius (see eq. (2))	1.5
f	Input scaling, which determines the strength of the feedback	1.0
$\alpha$	Learning rate (smaller value means faster learning)	0.01
$\tau_0$	Leak rate	[2, 7] s

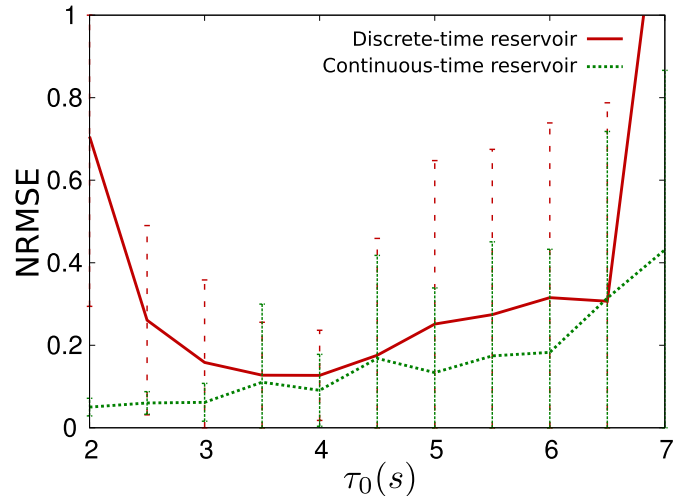


Fig. 3. NRMSE for different  $\tau_0$  for the MSO task described in Section III-A. The error bars show the sample standard deviation over 40 simulations ( $\text{NRMSE} \pm \sigma_{\text{NRMSE}}$ ). Solid red line: the error for a standard discrete-time reservoir, often used in the literature and in the practical applications. Dotted green line: the performance of a reservoir simulated in continuous time, with continuous-time feedback.

We have compared the performance of discrete-time and continuous-time reservoirs on the MSO task, as a function of the neuron time constant  $\tau_0$ . For each value of  $\tau_0$  and each reservoir type, the NRMSE was averaged over 40 experiments with randomized connection matrix, input noise in the warm-up phase, and input weight matrix. To make a fair comparison between discrete and continuous time, all simulations are sampled at  $\Delta t_{\text{sample}} = 1$  s, which means that the training stage receives the same number of samples for each reservoir. This also ensures that the learning rate  $\alpha$  has the same effect in each case.

The average NRMSE values are summarized in Fig. 3. It shows that moving to continuous time yields a decrease of the NRMSE over the full parameter range. This improvement is partly related to the bandwidth: certain frequencies cannot be captured in discrete time. A continuous-time system in theory has unlimited bandwidth, but for simulations this is limited by the  $\Delta t'$ . Because of the adaptive step-size algorithm, the effective bandwidth varies during the simulation.

For both reservoir types, we can identify an optimal value of the neuron time constant. For the discrete-time reservoir, we find an optimal NRMSE of 0.127 at  $\tau_0 = 4.0$  s, with a sample standard deviation of 0.111. For the continuous-time reservoir,

the best NRMSE in the interval considered ( $0.050 \pm 0.022$ ) occurs at the minimal value of  $\tau_0 = 2.0$  s. This shows that, for signal generation tasks, a continuous-time hardware reservoir can perform better than a discrete-time reservoir, or equivalently: the same performance can be reached with fewer neurons, which is an advantage when implementing a hardware reservoir computer.

## V. FROM REAL-VALUED TO COMPLEX VALUED

One of the advantages of using photonics for a hardware reservoir is the fact that, when using coherent light sources, the optical signals are complex valued, i.e., they have an amplitude and phase. To understand how this gives an advantage, we now extend the discrete-time leaky hyperbolic tangent reservoir with complex-valued states and compare this with the original real-valued reservoir. The input weights are also complex valued. The readout layer splits the states  $\mathbf{x}(t)$  into a real and imaginary parts, before multiplying them with  $\mathbf{W}_{\text{out}}$ .

Because the signals are now complex valued, we need to define a new nonlinearity for the neuron. It is tempting to replace the hyperbolic tangent function by  $f(z) = \tanh(z)$ ,  $z \in \mathbb{C}$ . This function has, however, discontinuities at  $f(j(2k+1)\pi/2)$ ,  $k \in \mathbb{Z}$  which are generally not wanted. We want to guarantee continuity and keep the  $\tanh(x)$  behavior with an image that is bound to  $[-1, 1]$ . We can do so by preserving the phase of the signal and applying  $\tanh$  on the absolute value of the signal

$$f(z) = e^{-j\angle(z)} \tanh(|z|) \quad (5)$$

where  $\angle(z)$  is the angle of  $z$ .

The state space of a complex-valued reservoir contains twice as many variables as that of a real-valued reservoir. In our experiments, this was exploited using the real and imaginary parts of the signal in each node as state signals. The performance of the complex-valued reservoir is similar to the performance as a real-valued reservoir with twice as many nodes. This can be observed in Fig. 4. Recall that for the discrete-time case, the optimal NRMSE value was  $0.127 \pm 0.111$ . For  $\Delta t = 1.0$  s, the optimal value is found for  $\tau_0 = 4.5$  s. For the same network, but with complex-valued states, the optimal value is found for  $\tau_0 = 3$  s, and the corresponding NRMSE equals  $0.0546 \pm 0.0238$ . We also compare this to a real-valued network of 400 neurons, which is double the size of the network, also shown in Fig. 4. The performance is similar: the best NRMSE of  $0.0514 \pm 0.0233$  is found for  $\tau_0 = 4$  s. We can conclude that, for this task, the performance of a coherent (i.e., complex valued) reservoir approximates that of a real-valued reservoir with twice the number of neurons.

## VI. PHOTONICS: A HARDWARE RC IMPLEMENTATION

In [10], a network of coupled SOAs was used to solve a speech recognition task. This network requires one electrical input per component to pump the amplifiers, which causes them to use a lot of power. Here, we propose a network of photonic crystal cavities. This platform is promising for a nanophotonic reservoir, because it is CMOS compatible,

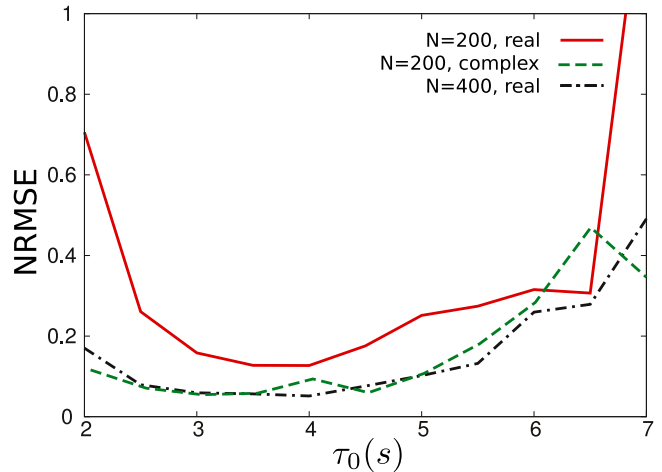


Fig. 4. NRMSE for three reservoirs: standard leaky hyperbolic tangent reservoir with 200 neurons (red, baseline), the same reservoir with complex-valued states (green, dashed), and a standard reservoir with 400 neurons. Clearly, the complex-valued reservoir performs better than the standard reservoir. For most leak rates, the performance is similar to the system with 400 neurons.

which allows us to fabricate very large networks using the mature silicon photonics technology. Because this is a planar technology, there are several constraints on the topology, but it has been proven previously that for rate-based systems the performance is largely independent on the exact topology [17]. Instead of using the speech task, we now use a signal generation task. This task is especially relevant for optics, for example, to create an optical signal generator.

Initially, we will explain the basic principles of a photonic crystal and a photonic crystal cavity, and explain how we model them using the coupled mode theory (CMT). Then, we design a network of photonic crystal cavities and explain what topology is used. In the last section of this paper, we train the MSO task using the photonic crystal cavity reservoir and compare our results to the performance of the hyperbolic tangent reservoir.

### A. Photonic Crystals, Waveguides, and Cavities

A photonic crystal is composed of a periodic nanostructure. Because of this periodicity, light with certain frequencies cannot propagate. The band in which propagation is not allowed is called the photonic band gap. Typically, these periodic structures are holes in a triangular lattice or rods in a rectangular lattice, and the dimensions of one period are approximately the same order of magnitude as the wavelengths, which are used. A photonic crystal waveguide is a photonic crystal with one-line defect, as shown in Fig. 5(a). Because light cannot propagate inside the photonic crystal, it is forced to propagate through this line defect. We can then introduce a cavity by inserting some holes inside this line, Fig. 5(b). Using semiconductor on insulator technology and optical lithography, photonic crystal waveguides have been successfully fabricated and measured [18].

Extensive theoretical [19], [20] and experimental [21], [22] studies have been performed on these devices. The component

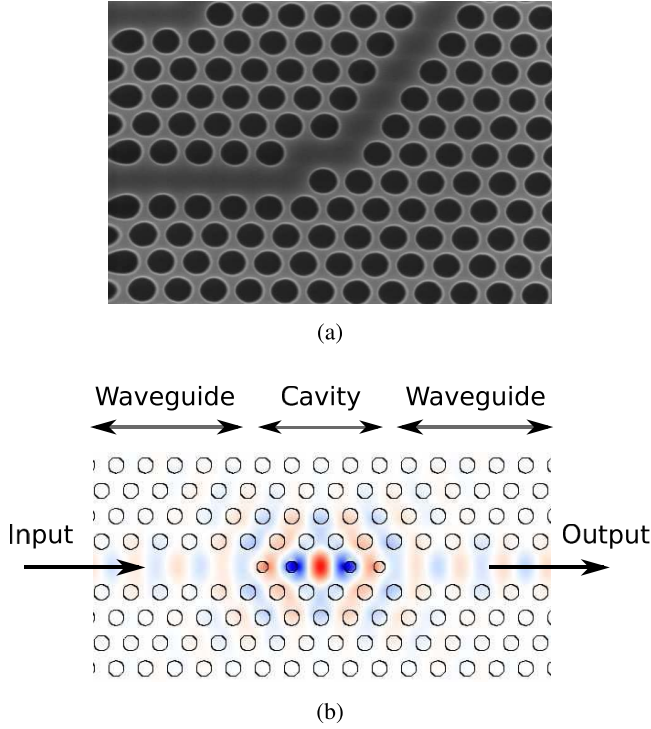


Fig. 5. Photonic crystals are periodic structures, made by, e.g., etching holes in a silicon layer. For certain wavelengths, the light cannot propagate through the periodic structure. Because of this property we can fabricate very small devices that manipulate light. For example, we can (a) guide light by bending it or (b) capture it in a very small volume. The latter is called a photonic crystal cavity, which is the building block we will use to construct a reservoir. (a) Photonic crystal waveguide with a bend. Light propagates through the line defect, and very small bends can be realized [18]. (b) 2-D finite-difference time-domain simulation of a photonic crystal cavity. Light tunnels to the small volume in the center and is trapped, which creates an optical resonator. This cavity is the basic building block for a nanophotonic reservoir.

is a resonator, which means that for certain wavelengths, light is trapped inside the cavity and the power inside the resonator is greatly enhanced. Furthermore, when the materials used to build the resonator are optically nonlinear, there are conditions for which the resonator becomes bistable. Similarly from [19] and [20], the nonlinearity, we use in this photonic crystal cavity is the Kerr effect. It is a second-order nonlinearity, i.e., the refractive index changes proportionally to the intensity of the waves:  $n = n_0 + n_2 I$ . The Kerr effect is very fast, in the order of a few femtosecond.

There are many ways to model these optical resonators. The most accurate way to simulate them is by using finite difference time domain (FDTD) simulations in which the Maxwell equations, which describe light propagation, are discretized directly. The physical layout of the device is transformed to a 3-D grid and this system is then solved in the time domain. These simulations are computationally very intensive. In addition, because the structure is a resonator, light bounces back and forth in the cavity multiple times, and as such a lot of time steps are needed to model the buildup of energy in such a component. Typically, a simulation of one resonator in 2-D takes more than 10 h to complete on an eight-core 3-GHz processor. For example, Fig. 5(b) was rendered by performing a 2-D FDTD simulation in MEEP [23]. In contrast, the approximate CMT is computationally much

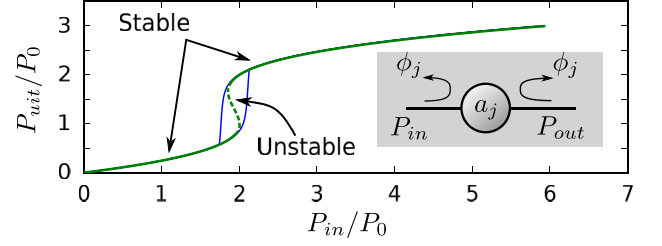


Fig. 6. Output versus input power of a single nonlinear optical resonator (shown in inset) with a detuning  $\delta = -2$ . The resonator has a bistability (the dotted region is an unstable solution). Input light whose wavelength is close to the resonance wavelength is trapped inside the cavity, leading to energy accumulation. Nonlinear effects, which depend on the intensity of the field, are thereby greatly enhanced, leading to the observed bistability. The typical time scale for energy accumulation/dissipation because of the optical resonance is  $\tau \approx 1$  ps.

faster and consumes much less memory. The simulation time for the same system is now less than 1 ms. In CMT, the behavior of the resonator is fully described by only one variable: the complex amplitude of the resonator  $a_j$  (more complex models also include the temperature, the amount of free carriers or even multiple modes). It has been shown that the correspondence between FDTD and CMT is very good for a system of two resonators [19]. To decrease the computational footprint of the simulations while still keeping a good level of accuracy, we use the CMT to simulate the nanophotonic reservoir. Each resonator is governed by the following CMT equations (the time dependency is omitted, for simplicity):

$$\frac{da_j}{dt} = \left[ i((\omega_r - \omega_0) + \delta\omega_j) - \frac{1}{\tau} \right] a_j + ds_{j,in,0} + ds_{j,in,1} \quad (6)$$

$$s_{j,out,0} = \exp(i\phi_j)s_{j,in,0} + da_j \quad (7)$$

$$s_{j,out,1} = \exp(i\phi_j)s_{j,in,1} + da_j. \quad (8)$$

Here,  $\phi_j$  is a phase that depends on the resonator mirror reflection properties (also shown in the inset of Fig. 6). The nonlinear frequency shift is  $\delta\omega_j = -|a_j|^2/(P_0\tau^2)$ , with  $P_0$  the characteristic nonlinear power of the cavity [20], which is directly related to the nonlinear coefficient  $n_2$  (Kerr effect).  $d = i\exp(i\phi_j/2)/\sqrt{\tau}$ , where  $\tau$  is the lifetime of the cavity. The detuning of the cavity is defined as  $\delta = \tau(\omega_0 - \omega_r)$ .  $\omega_r$  is the resonance frequency of the cavity and  $\omega_0$  is the frequency of the input signal.

The input–output behavior of this resonator is shown in Fig. 6. Here, we normalized the power with respect to  $P_0$ . The small-signal power gain at  $P_{in} = 0W$  is 1, 0.5, 0.2 for a detuning of  $\delta$  0,  $-1$ ,  $-2$ , respectively. The onset of bistability occurs for  $\delta < -\sqrt{3}$ .

### B. Network Topology and Hardware Challenges

A hardware implementation implies certain restrictions on the topology. In photonics these constraints are associated with the fact that the chip is planar, crossings induce loss and large fan-in/out increases complexity. It has been proven in [17] that for an analog reservoir (i.e., using rate equations) the performance is largely independent of the sparsity of the

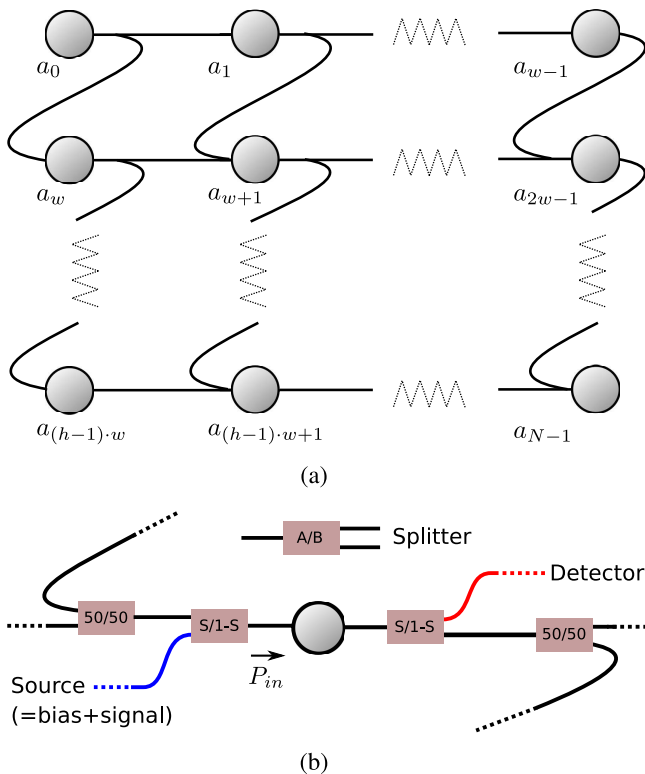


Fig. 7. Proposed topology for the photonic crystal cavity reservoir. A regular mesh topology minimizes the fan-in and crossings, while keeping a good connectivity. An important design parameter is the splitting ratio  $S$ , the amount of power that enters/leaves the network through the source/detector. (a) Because of hardware restrictions, it is difficult to use a random topology. Especially, the fan-in should be minimized, because large fan-in means increasing process variations and design complexity. The mesh topology has very good performance for optical components, can be easily designed, and minimizes the amount of fan-in and fan-out. (b) Typical splitting ratios in a fully connected node in the mesh topology. The fan-in and fan-out have been minimized to three. The splitting ratio  $S$  is the fraction of power that enters the network from the source, and the fraction of power that goes to the detector.

network or the exact topology. Thus, we can safely implement the required topology constraints without expecting a large impact on the final performance.

In photonics, we need to design optical splitters to distribute the light. Splitters with more ports have an increased design complexity and they are more susceptible to process variations. For these reasons, we limit the fan-in of one photonic crystal cavity to three: 1) one input signal and 2) two inputs coming from neighboring cavities. Similarly for the fan-out, one output signal goes to a detector and two outputs go to other cavities. In addition, as the photonic chip is implemented on a planar surface, too many optical crossings should be avoided if possible because these cause additional loss and cross talk. Recent improvements in crossing strategies reduce the loss to only 0.17 dB [24] per crossing, which means that restrictions on the topology are now less stringent. The topology, inspired by [10], is still a regular 2-D mesh of photonic crystal cavities, in which the cavities are connected to close neighbors. This is shown in Fig. 7. Note that the interconnections between two nodes are bidirectional.

When moving to a future hardware implementation, there are two extra challenges to be solved: 1) the feedback loop

and 2) the readout. Conceptually, it is very easy to make the feedback loop all-optical: the output weights can be modified externally, for example, using a Mach Zehnder Interferometer and applying a voltage in one of the arms. This causes constructive/destructive interference, which allows us to modify the readout weight. An amplifier is needed to amplify the output before feeding it back to the input. This can be implemented by designing an SOA in the feedback loop. For example, the amplifier from [25] can be used, which is demonstrated on the same silicon photonics platform.

Another challenge is reading out the states and recalculating the weights: the outputs have to be read out, fed to a computer to calculate the weight adjustment, and then the output weights have to be modulated. Other learning techniques can simplify the calculations, such as a more simple delta-type rule in which we do not need to perform complex matrix operations (also shown in [12]), or using Hebbian learning [26], where the exact error signal is not explicitly needed.

In the used CMT models, which builds upon the results of [19] and [20], free carrier effects (in the order of a few nanosecond) and temperature effects ( $\sim 0.100 \mu\text{s}$ ) have not been considered. The CMT equations and steady-state curves for these type of dynamical effects are, however, very similar, which means that similar conclusions should hold there too.

Depending on the specific design parameters and the material system, some of the effects will be more important than other effects: because of the nonlinear two photon absorption, the free carrier effects in silicon on insulator at  $\lambda = 1.55 \mu\text{m}$  are not negligible. In addition, because of the high powers inside the cavities, the temperature will have a significant effect, albeit on a slow time scale compared with the fast Kerr effect. Moving to different material systems, using techniques to reduce the lifetime of the free carriers, or using different wavelengths can change these conditions.

## VII. PHOTONICS: SIMULATION RESULTS

In this section, we will discuss the results of the MSO task in the optical domain. The reservoir parameters are listed in Table II. The target signal is again given by the sum of two sines

$$s(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t). \quad (9)$$

Here,  $f_1 = 0.3/\tau_0 \simeq 216 \text{GHz}$  and  $f_2 = 0.311/0.2 f_1$ , so the same ratio of frequencies is preserved. This fast frequency is based on the lifetime of the cavity (which is in the order of 1.4 ps), to fully exploit the internal dynamics of the nodes.

We will discuss the influence of several important design parameters for the photonic crystal cavity reservoir, such as the phase between the resonators, the delay in the waveguides connecting the cavities, the splitting ratio  $S$  [Fig. 7(b)] and the network size. To interpret the results physically, it will be useful to draw the comparison with a system of two coupled cavities.

### A. Phase Reflection of the Resonator

The phase reflection  $\phi_j$  strongly influences the dynamics of the optical resonator. We keep the phase between the

TABLE II

DEFAULT VALUES USED IN THE PHOTONIC CRYSTAL CAVITY RESERVOIR

Name	Description	Default value
width	Width of the 2D mesh	20
height	Height of the 2D mesh	10
N	Number of cavities	200
BF	Bias fraction (fraction of nodes that receive bias)	0.1
BP	Bias power	$1.30P_0$
$\phi_j$	Phase reflection of the cavities	$0.2\pi$
FB	The strength of the feedback	1.0
$\lambda$	Input wavelength	1551.83 nm
$\lambda_r$	Cavity resonance wavelength	$1550 \text{ nm} \pm 0.2 \text{ nm}$
$\tau$	Cavity lifetime	1.39 ps
$\delta$	Detuning of the cavity ( $= \tau(\omega_0 - \omega_r)$ )	-2
$f_1$	Target signal frequency	$0.3/\tau_0 \simeq 216 \text{ GHz}$
$\alpha$	Learning rate (smaller: faster)	0.01
$\tau_d$	Delay between the resonators	0 ps

resonators (which is determined by the waveguide between the resonators) fixed, which is a realistic assumption if engineered correctly.

For a dynamical system, a linear stability analysis reveals whether the system is stable or not. This is done by examining the eigenvalues of the Jacobian of the system. If all eigenvalues have a negative real part, the system is stable. In some cases, an unstable fixed point implies chaos or self-pulsation. To distinguish between both, we can calculate the maximal Lyapunov exponent of the system. If the maximal Lyapunov exponent is larger than zero, the system is chaotic. For a stable periodic solution, the maximal Lyapunov exponent is zero.

This is elaborated in detailed in [19]. For example, a series of two resonators will self-pulsate when  $\phi_j \simeq 0.2\pi$ ,  $P_{\text{in}} \simeq P_0$ , and  $\delta = -2$ . For larger circuits with arbitrary topology, it becomes very cumbersome to evaluate the Jacobian and calculate the largest Lyapunov exponent. Numerical simulations of an optical reservoir (with parameters used from Table II) show that there is a region where self-pulsation occurs, and that the condition for the phase is similar for the case of two coupled resonators. We chose  $\phi = 0.2\pi$  and feed input power to a fraction (BF in Table II) of the cavities. When this fraction is sufficiently large ( $>10\%$ ), enough power arrives in the cavities and self-pulsation occurs. Under these conditions, the training and/or free-run are disturbed severely and most of the time, training does not converge.

This shows that preferentially, there should be no self-pulsation or strong synchronized interaction between the cavities. To understand this behavior more quantitatively, we change the phase reflection to  $\phi_j = 0.2\pi + \phi_r \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, 1)$  is sampled from a Gaussian random variable with zero mean and one variance. Physically, this can be done by changing the length of the interconnection between the different cavities. When increasing  $\phi_r$ , we move out of the self-pulsing regime for the case of two cavities, which is an indication that the interaction between the cavities is less synchronized. Indeed, the reservoir performance improves. This can be observed in Fig. 8, where we simulate for different fractions of the bias input, and sweep  $\phi_r$  between 0 and  $2\pi$ . With no bias and randomized phases, the best NRMSE is found and equals  $0.030 \pm 0.021$ .

On the silicon photonics platform [27], the phase errors produced by propagation over a photonic wire are very small.

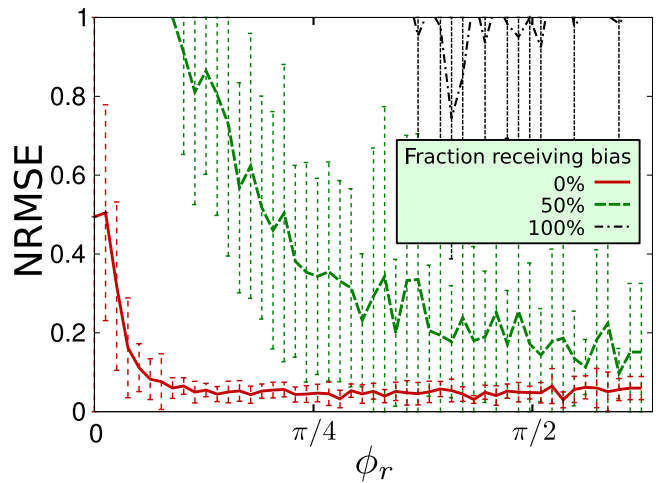


Fig. 8. Error (NRMSE) after training an optical network of photonic crystal cavities for the MSO task. The phase between the resonators is described by  $\phi_j = 0.2\pi + \phi_r \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, 1)$ . This is done for different fraction of cavities that receive bias. The more cavities that receive bias, the more the reservoir dynamics are disturbed by strong interactions between the resonators (e.g., self-pulsation), which causes the network to be unable to generate the signal autonomously. Increasing the randomness in the phase reduces the amount of self-pulsation.

Over a few 100  $\mu\text{m}$ , the phase error for two identically designed waveguides is less than  $0.1\pi$  [28], with the technology improving fast. By changing the length or width of the waveguide, we can change this relative phase.

Variations caused by fabrication imperfections also cause the resonance frequencies of optical resonators to be different from one device to the other. This shift can be up to 1 nm [27]. For our simulations, we used a variation on the resonance of 0.2 nm (Table II). Above 0.3 nm, some of the simulations do not converge, and above 0.4 nm, no reservoir succeeds at reproducing the MSO signal. As the technology improves, this resonance shift because of fabrication errors is becoming increasingly smaller. Conceptually, we can compensate the individual resonances either by trimming the individual devices [29] or adjusting the temperature of each device with heaters [30], [31]. But, also an alternative designs can relax the requirements on the variability of the system. For example, a slower cavity lifetime means the resonance is less pronounced, making it less susceptible to process variations, but requiring higher input powers and faster readout mechanisms.

### B. Delays

Although the phase and the delay of a waveguide with length  $L$  both scale linearly with  $L$ , it is more convenient to investigate the effects separately. This is justified by the fact that we only need 0.05  $\mu\text{m}$  length to rotate the phase by  $0.1\pi$ , while we need at least 50  $\mu\text{m}$  of waveguide length to observe a significant delay (of approx. 0.1 ps). Again, we first look at the dynamics of a sequence of two coupled cavities. The response for different delays is shown in Fig. 9. Here, the delay is very important for the condition for self-pulsation and the shape of the output signal. For  $\phi = 0.2\pi$ ,  $P_{\text{in}} = 1.30P_0$ , and  $\delta = -2$ , and for a delay larger than 2.5 ps (which corresponds to approximately 310  $\mu\text{m}$  on-chip), the self-pulsation is lost



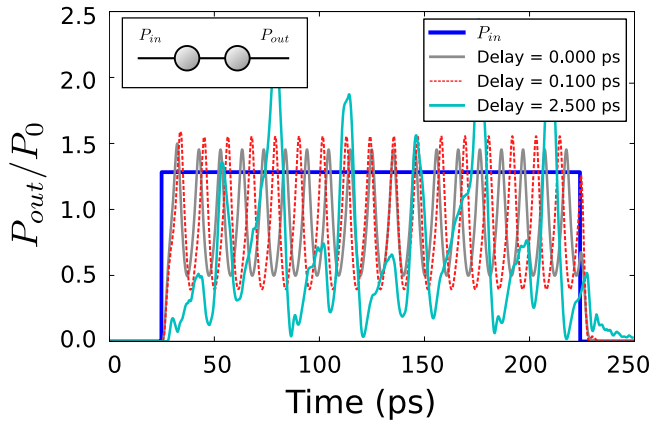


Fig. 9. Dynamics for a sequence of two coupled resonators (shown in inset). For increasing delay, the dynamics change from self-pulsation to chaos. A delay of 0.1 ps corresponds to approximately  $10 \mu\text{m}$  on-chip. Although for two resonators the dynamics change significantly, it has a negligible effect on the training properties of a large reservoir.

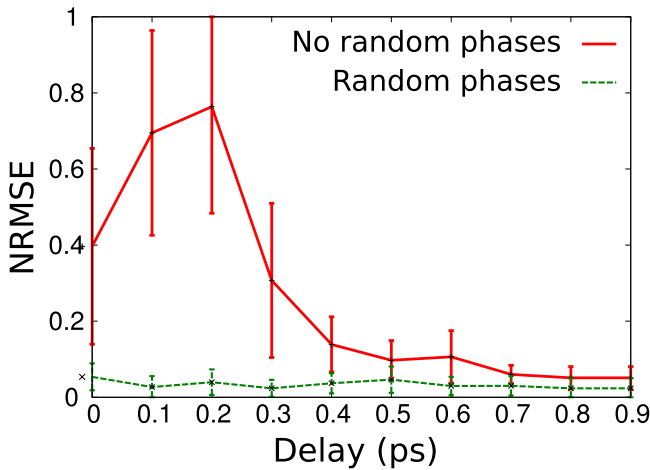


Fig. 10. Error (NRMSE) of the photonic reservoir for the MSO task. As shown in Fig. 8, training fails in certain conditions when the phases are not randomized. By increasing the delay (100-ps delay is approximately  $12.5 \mu\text{m}$  on-chip), the self-pulsation in a series of two cavities is lost (Fig. 9), and this conclusion can be extrapolated to larger circuits where training improves when the self-pulsation locking is no longer present.

and the output becomes chaotic. The influence of the delay for the MSO task is shown in Fig. 10. Where the resonators were initially locked in a self-pulsing regime (which caused learning to fail), increasing the delay now improves the ability to learn. We only simulate up to 1.0 ps as the trend does not change anymore after this delay. If the system was initially capable of learning (e.g., when the phases were randomized), increasing the delay has almost no influence on the performance of the reservoir. For 0.8 ps and a bias fraction of 10% the NRMSE is  $0.024 \pm 0.028$ .

### C. Splitters

For all simulations until now, the splitting ratio  $S$  [Fig. 7(b)] was neglected, and we measured the power of the cavities without disturbing the system. In addition, the input was fed to the cavities without introducing extra loss. In practice, however, when feeding power to a cavity, this has the

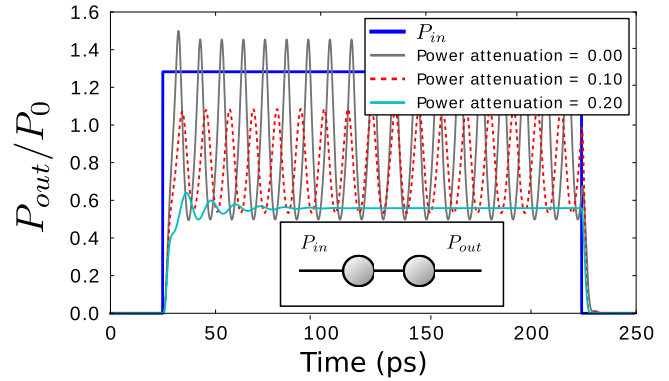


Fig. 11. Dynamics of a series of two coupled resonators for increasing attenuation of the waveguide between the cavities. For a power attenuation of 20%, the self-pulsation is lost.

drawback that power also leaks away through the same port. For example, a splitting ratio of 10/90 ( $S = 0.10$ ) means that 10% of the source power reaches the cavity, but it also means that 10% of the cavity power leaks away through this channel. 90% of the power circulates to other cavities. To be able to draw meaningful comparisons between different values of  $S$ , the input signal amplitude and feedback strength have been scaled by  $1/\sqrt{S}$ . The splitting ratio should be kept small to ensure enough power circulates in the reservoir.

Again, studying the dynamics of a system of two coupled resonators gives insight about the behavior of the reservoir. In this case, we increase the attenuation of the waveguide between the two resonators, which has the same effect as splitting some of the power in the full reservoir. For example, when the splitting ratio  $S$  of Fig. 7(b) equals 0.1, this is equal to  $1 - 0.9^2 = 19\%$  power attenuation in the waveguide (two splitters are passed between two resonators). For this attenuation, the interaction between two cavities reduces considerably. In Fig. 11, we can clearly observe that the self-pulsation is lost when the attenuation becomes bigger.

The influence of the splitter ratio on the training is shown in Fig. 12. When initially the training fails because the bias fraction is large causing strong unwanted interactions between the cavities (for example, self-pulsation), increasing the splitting ratio will decrease these strong interactions between the neurons, which improves the results. In the other case, where the bias fraction is low (and initially the training converges), there is no significant influence of this splitting ratio on the training (although the error increases slightly for larger  $S$ ). Of course, the bias input powers and the feedback strength need to be scaled accordingly by a factor of  $1/\sqrt{S}$ . A larger  $S$  means that less external power needs to be added (bias) or amplified (feedback connection).

### D. Network Size

For the final MSO experiment, we measure the performance of the network as a function of the network size for two slightly different topologies (a square mesh versus a rectangular mesh). From Fig. 13, we can conclude that for the chosen target signal  $s(t)$  (9), using more than 70 resonators does not

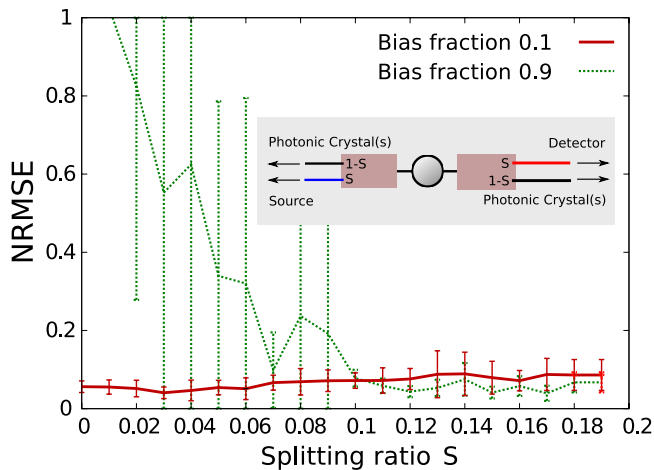


Fig. 12. Influence of the splitter ratio  $S$  [Fig. 7(b) and inset in this figure]. By increasing the splitting ratio, the interaction between two neighboring cavities is decreased. This is advantageous for learning, because the strong self-pulsation which disrupts training disappear (Fig. 11). Parameters: phases random,  $P_{\text{bias}} = 1.3P_0/\sqrt{S}$ ,  $FB = 1.0/\sqrt{S}$ .

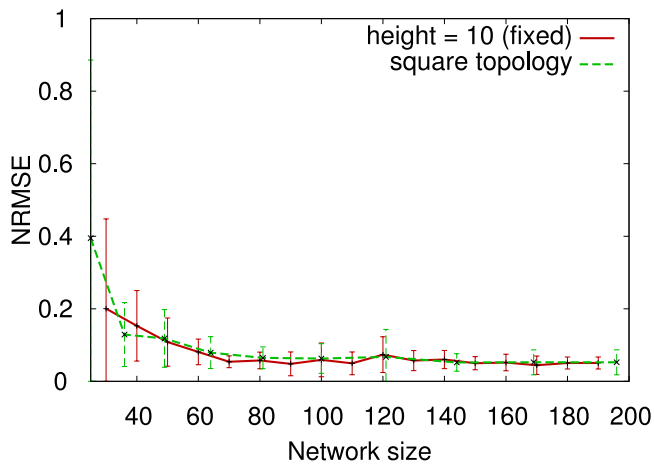


Fig. 13. Influence of the network size on the performance. We have simulated two variations on the mesh topology, once with a square topology and once with a rectangular topology. Clearly, the influence of this slight topology change is negligible. It also shows that, for this specific task, 70 resonators are sufficient, and there is no performance gain using more resonators.

further improve the performance.<sup>2</sup> Furthermore, the difference between a rectangular and square topology is negligible for this task.

### E. Information Processing Capacity

In [32], the (linear) memory capacity of reservoirs was introduced. It quantifies a reservoir's capacity to reproduce past input samples in a task-independent way. For many tasks, a larger linear memory capacity will result in better performance of the reservoir. In [33], this concept was generalized to the total information processing capacity. This measure quantifies the total capacity of a dynamical system to compute transformations, both linear and nonlinear, of its input history.

<sup>2</sup>In classical RC, the network size is usually around 1000 neurons. As this is clearly not realistic for a first hardware implementation, we compared the hyperbolic tangent and nanophotonic reservoir for 200 neurons.

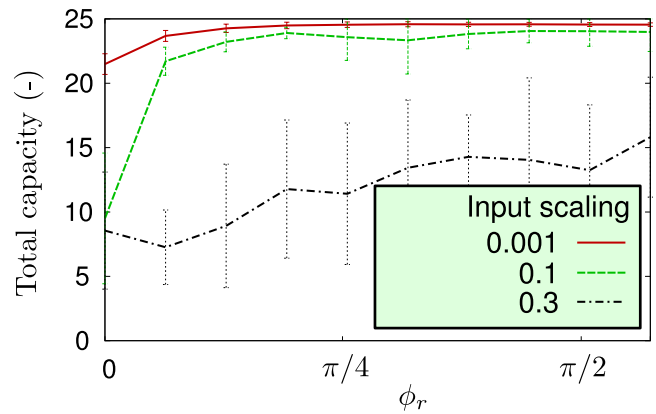


Fig. 14. Total information processing capacity of a  $5 \times 5$  photonic crystal cavity network. The region with low phase randomness and high input powers correspond with self-pulsation regions. These regions are better avoided to increase the total capacity, and hence the performance of the system. This conclusion is in line with previous experiments.

This total capacity is bounded by the number of observed state variables. In our photonic reservoir, this is the number of resonators in the system. This maximal capacity is effectively reached for dynamical systems with fading memory, i.e., that are asymptotically stable during their entire driven operation. Although the experimental quantification of the total capacity can be a bit tedious, reasonable approximations of the total capacity can often be achieved within an acceptable computation time. In practice, the useable capacity decreases rapidly in the presence of noise if the responses of the reservoir states to the input are very similar. To perform well as a reservoir, a dynamical system should first achieve close to its maximal capacity, i.e., be stable and have sufficient variability in its responses to the input signals.

We will now apply this knowledge to a small reservoir of 25 photonic crystal cavities, and consider two important parameters: 1) the input scaling and 2) the phase randomness. The amount of nonlinearity is determined by the power inside a cavity (and hence by the input scaling), so this will have a large influence on the dynamics of the network. In addition, the phase randomness in the phase reflection  $\phi_r$  will greatly influence the performance (as we observe in Fig. 8). The other reservoir parameters are fixed, and are taken from Table II. In this experiment, we have used the magnitude of the output of the neurons instead of using the real and imaginary parts separately. As can be observed from Fig. 14, the total information processing capacity (with a maximum of 25) drops dramatically when the input scaling increases and there are no random phases. This again corresponds to the self-pulsation regime, which we encountered in the previous sections. This means that for typical RC tasks, the performance will be better when not in a self-pulsing regime.

## VIII. CONCLUSION

In this paper, we have studied different reservoir architectures for the generation of periodic patterns. The output weights are trained using an online technique called FORCE. The standard benchmark task we used is the MSO task,

TABLE III  
SUMMARY OF THE NRMSE CALCULATED IN THIS PAPER FOR THE  
DIFFERENT ARCHITECTURES

System	comments	NRMSE
Discrete time		$0.127 \pm 0.111$
Continuous time		$0.050 \pm 0.022$
Discrete time	Complex-valued states	$0.0515 \pm 0.039$
Discrete time	Double size (N=400)	$0.0514 \pm 0.0233$
Photonic crystal cavities	Randomized phase, no bias	$0.030 \pm 0.021$

and the performance is measured by the NRMSE. Table III summarizes the result of all simulations performed in this paper.

Unlike standard RC, where a discrete-time system is used, we use advanced integration routines to simulate a reservoir in continuous time, and find that the reservoir performs better for the MSO task, which is inherently a continuous-time task. In addition, using a complex-valued reservoir improves the general performance because of the increased richness of the internal state space.

We have presented a hardware implementation for RC using photonic crystal cavities. These resonators exhibit bistability because of the Kerr nonlinearity, and they are modeled using CMT. There are several important design parameters such as the topology, the phase difference between the cavities and the delay between the cavities. We show that it is important not to drive the cavities in a self-pulsating regime, because strong interaction between neighboring resonators disturbs the training process and decreases the final performance. After optimizing the parameters of the optical reservoir, we find that it outperforms the classical hyperbolic tangent reservoir: the average NRMSE is 0.03 compared with a NRMSE of 0.127 for the hyperbolic tangent reservoir. This conceptual study shows that photonic crystal cavities are a good candidate for generating periodic patterns in the optical domain. There are, however, some challenges to overcome to create an all-optical hardware implementation: the readout needs to be computed fast enough, many data signals need to be provided on-chip, the variation on the resonance frequencies of the photonic crystal cavities should be small enough, and the feedback signal needs to be strong enough.

## REFERENCES

- [1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [2] W. Maas, T. Natschlagler, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [3] B. Schrauwen, M. D'Haene, D. Verstraeten, and J. V. Campenhout, "Compact hardware liquid state machines on FPGA for real-time speech recognition," *Neural Netw.*, vol. 21, nos. 2–3, pp. 511–523, 2008.
- [4] P. Buteneers, D. Verstraeten, P. van Mierlo, T. Wyckhuys, D. Stroobandt, R. Raedt, H. Hallez, and B. Schrauwen, "Automatic detection of epileptic seizures on the intra-cranial electroencephalogram of rats using reservoir computing," *Artif. Intell. Med.*, vol. 53, no. 3, pp. 215–223, Nov. 2011.
- [5] E. Antonelo, B. Schrauwen, and D. Stroobandt, "Event detection and localization for small mobile robots using reservoir computing," *Neural Netw.*, vol. 21, no. 6, pp. 862–871, Aug. 2008.
- [6] F. Wyffels and B. Schrauwen, "A comparative study of reservoir computing strategies for monthly time series prediction," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1958–1964, Jun. 2010.
- [7] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI - Künstliche Intell. Manuscript*, Mar. 2012, pp. 1–8.
- [8] F. Schürmann, K. Meier, and J. Schemmel, "Edge of chaos computation in mixed-mode VLSI—A hard liquid," in *Proc. NIPS*, 2005.
- [9] L. Appeltant, M. Soriano, G. V. Der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, pp. 1–3, Sep. 2011.
- [10] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1469–1481, Sep. 2011.
- [11] J. J. Steil, "Several ways to solve the MSO problem," in *Proc. ESANN*, 2007, pp. 489–494.
- [12] D. Sussillo and L. Abbott, "Generating coherent patterns of activity from chaotic neural networks," *Neuron*, vol. 63, no. 4, pp. 544–557, Aug. 2009.
- [13] D. Verstraeten, B. Schrauwen, S. Dieleman, P. Brakel, P. Buteneers, and D. Pecevski, "Oger: Modular learning architectures for large-scale sequential processing," *J. Mach. Learn. Res.*, vol. 1, 2011.
- [14] T. Zito, N. Wilbert, L. Wiskott, and P. Berkes, "Modular toolkit for data processing (MDP): A python data processing framework," *Frontiers Neuroinformat.*, vol. 2, no. 8, pp. 1–10, Jan. 2009.
- [15] M. Fiers, T. V. Vaerenbergh, K. Caluwaerts, D. V. Ginste, B. Schrauwen, J. Dambre, and P. Bienstman, "Time-domain and frequency-domain modeling of nonlinear optical components at the circuit-level using a node-based approach," *J. Opt. Soc. Amer. B-Opt. Phys.*, vol. 29, no. 5, pp. 896–900, 2012.
- [16] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin, "Pruning and regularization in reservoir computing," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1534–1546, Mar. 2009.
- [17] L. Busing, B. Schrauwen, and R. Legenstein, "Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons," *Neural Comput.*, vol. 22, no. 5, pp. 1272–311, 2010.
- [18] S. Selvaraja, P. Jaenen, W. Bogaerts, D. Van Thourhout, P. Dumon, and R. Baets, "Fabrication of photonic wire and crystal circuits in silicon-on-insulator using 193-nm optical lithography," *J. Lightw. Technol.*, vol. 27, no. 18, pp. 4076–4083, Sep. 2009.
- [19] B. Maes, M. Fiers, and P. Bienstman, "Self-pulsing and chaos in series of coupled nonlinear micro-cavities," *Phys. Rev. B*, vol. 79, no. 111, pp. 3–15, 2009.
- [20] M. Soljačić, M. Ibanescu, S. Johnson, Y. Fink, and J. Joannopoulos, "Optimal bistable switching in nonlinear photonic crystals," *Phys. Rev. E*, vol. 66, no. 5, pp. 055601-1–055601-4, Nov. 2002.
- [21] T. Asano, B.-S. Song, and S. Noda, "Analysis of the experimental factors (1 million) of photonic crystal nanocavities," *Opt. Exp.*, vol. 14, no. 5, pp. 1996–2002, Mar. 2006.
- [22] K. Srinivasan, P. E. Barclay, O. Painter, J. Chen, A. Y. Cho, and C. Gmachl, "Experimental demonstration of a high quality factor photonic crystal microcavity," *Appl. Phys. Lett.*, vol. 83, no. 10, pp. 1915–1917, 2003.
- [23] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. Joannopoulos, and S. G. Johnson, "Meep: A flexible free-software package for electromagnetic simulations by the FDTD method," *Comput. Phys. Commun.*, vol. 181, no. 3, pp. 687–702, Mar. 2010.
- [24] W. Bogaerts, P. Dumon, D. V. Thourhout, and R. Baets, "Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides," *Opt. Lett.*, vol. 32, no. 19, pp. 2801–2803, 2007.
- [25] M. Tassaert, G. Roelkens, H. J. S. Dorren, D. Van Thourhout, and O. Raz, "Bias-free, low power and optically driven membrane InP switch on SOI for remotely configurable photonic packet switches," *Opt. Exp.*, vol. 19, no. 26, pp. B817–B824, Dec. 2011.
- [26] R. Legenstein, S. M. Chase, A. B. Schwartz, and W. Maass, "A reward-modulated Hebbian learning rule can explain experimentally observed network reorganization in a brain control task," *J. Neurosci.*, vol. 30, no. 25, pp. 8400–8410, 2010.

- [27] S. Selvaraja, W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets, "Subnanometer linewidth uniformity in silicon nanophotonic waveguide devices using CMOS fabrication technology," *IEEE J. Sel. Topics Quantum Electron.*, vol. 16, no. 1, pp. 316–324, Jan. 2010.
- [28] P. Dumon, "Ultra-compact integrated optical filters in silicon-on-insulator by means of wafer-scale technology," mar 2007.
- [29] J. Schrauwen, D. Van Thourhout, and R. Baets, "Trimming of silicon ring resonator by electron beam induced compaction and strain," *Opt. Exp.*, vol. 16, no. 6, pp. 3738–3743, Mar. 2008.
- [30] D. Dai, L. Yang, and S. He, "Ultrasmall thermally tunable microring resonator with a submicrometer heater on Si nanowires," *J. Lightw. Technol.*, vol. 26, no. 6, pp. 704–709, Mar. 2008.
- [31] C. Qiu, J. Shu, Z. Li, X. Zhang, and Q. Xu, "Wavelength tracking with thermally controlled silicon resonators," *Opt. Exp.*, vol. 19, no. 6, pp. 5143–5148, Mar. 2011.
- [32] H. Jaeger, (2001). *The 'Echo State' Approach to Analysing and Training Recurrent Neural Networks* [Online]. Available: <http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf>
- [33] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Eng information processing capacity of dynamical systems," *Eng. Sci. Rep.*, vol. 2, pp. 1–7, Jan. 2012.



**Martin Andre Agnes Fiers** (S'10–M'12) was born in Zottegem, Belgium, in 1985. He received the M.Sc. degree in electrical engineering from Ghent University, Ghent, Belgium, in 2008.

He is currently with the Photonics Research Group, Department of Information Technology. His research topic is photonic reservoir computing. His current research interests include modeling of photonic components and software development for circuit simulation tools.



**Thomas Van Vaerenbergh** (S'11) was born in Ghent, Belgium, 1987. He received the M.Sc. degree in applied physics from Ghent University, Ghent, Belgium, in 2008.

He is currently with the Photonics Research Group, Department of Information Technology. His research topic is photonic reservoir computing. His current research interests include excitability in photonic components and the modeling of microring circuits.



**Francis Wyffels** was born in Belgium in 1983. He received the M.Sc. degree in computer engineering from Ghent University, Ghent, Belgium, in 2007. Since 2007, he has been pursuing the Ph.D. degree with the Electronics Information System Department, Ghent University, where he is investigating how to exploit properties of reservoir computing systems for robot locomotion and time series prediction.



**David Verstraeten** was born in Brussels, Belgium, on January 17, 1979. He received the M.Sc. degree in computer engineering and the Ph.D. degree from Ghent University, Ghent, Belgium, in 2004 and 2009, respectively.

He is currently a Post-Doctoral Researcher with the Department of Electronics and Information Systems, Ghent University. His current research interests include the study of reservoir computing dynamics and machine learning for biomedical applications and speech recognition.



**Benjamin Schrauwen** received the M.Sc. degree in computer science from the University of Antwerp, Edegem, Belgium, in 2000, and the M.Sc. and Ph.D. degrees in computer engineering from Ghent University, Ghent, Belgium, in 2002 and 2008, respectively.

He has been a Professor with Ghent University with the Department of Electronics and Information Systems, Faculty of Engineering, since 2010. He is currently leading a research group over 15 people with interests in reservoir computing, spiking neural networks, and deep learning architectures, with applications in the domains of speech recognition, robotics, and biomedical signal processing. His current research interests include machine learning and biologically inspired computing systems.

Prof. Schrauwen is a member of the American Association for the Advancement of Science.



**Joni Dambre** (S'99–M'03) was born in Ghent, Belgium, in 1973. She received the M.Sc. degree in electronics engineering and the Ph.D. degree in computer engineering from Ghent University, Ghent, in 1996 and 2003, respectively.

She is currently a Post-Doctoral Researcher and Part-Time Professor with the Department of Electronics and Information Systems, Ghent University. She has performed research on very large scale integration architecture and technology evaluation and novel interconnection techniques. Her current research interests include hardware implementations of reservoir computing and the computational power of general dynamical systems.



**Peter Bienstman** (S'97–A'01–M'03) was born in Ghent, Belgium, in 1974. He received the M.Sc. degree in electrical engineering and the Ph.D. degree from Ghent University, Ghent, in 1997 and 2001, respectively.

He is currently an Associate Professor with the Department of Information Technology, Ghent University. From 2001 to 2002, he was with the Joannopoulos Research Group, MIT. His current research interests include several applications of nanophotonics as well as nanophotonics modeling. He has published 50 papers and holds several patents.

Dr. Bienstman has been awarded the European Research Council starting grant for the Naresco-project: Novel paradigms for massively parallel nanophotonic information processing. He is a member of the IEEE Photonics Society.