

SCIENTIFIC REPORTS



OPEN

Fully analogue photonic reservoir computer

François Duport¹, Anteo Smerieri¹, Akram Akrou², Marc Haelterman¹ & Serge Massar²

Received: 28 October 2015

Accepted: 08 February 2016

Published: 03 March 2016

Introduced a decade ago, reservoir computing is an efficient approach for signal processing. State of the art capabilities have already been demonstrated with both computer simulations and physical implementations. If photonic reservoir computing appears to be promising a solution for ultrafast nontrivial computing, all the implementations presented up to now require digital pre or post processing, which prevents them from exploiting their full potential, in particular in terms of processing speed. We address here the possibility to get rid simultaneously of both digital pre and post processing. The standalone fully analogue reservoir computer resulting from our endeavour is compared to previous experiments and only exhibits rather limited degradation of performances. Our experiment constitutes a proof of concept for standalone physical reservoir computers.

Reservoir computing is a bio-inspired approach for processing time dependent information^{1–5}. A reservoir computer can be decomposed into three parts, see Fig. 1. The “input layer” couples the input signal into a non-linear dynamical system that constitutes the “reservoir layer”. The internal variables of the dynamical system, also called “reservoir states”, provide a nonlinear mapping of the input into a high dimensional space. Finally the time-dependent output of the reservoir is computed in the “output layer” as a linear combination of the internal variables. The readout weights used to compute this linear combination are optimized so as to minimize the mean square error between the target and the output signal, leading to a simple and easy training process. On the other hand, the values of the internal coupling weights within the input layer and within the reservoir layer are not critical, and can be chosen at random up to some global parameters that are tuned to get the best performance.

One of the key advantages of reservoir computers is that, because only the output layer is trained, training algorithms are efficient and rapidly converge to the global optimum. This simplicity enables reservoir computers to solve a large range of complex tasks on time dependent signals, such as speech recognition⁶, nonlinear channel equalization^{3,7,8}, detection of epileptic seizures⁹, robot control¹⁰, time series prediction^{1,3}, financial forecasting, handwriting recognition, etc..., often with state of the art performance. We refer to^{11–13} for recent reviews.

This simplicity and flexibility has also allowed for a breakthrough in analogue information processing, and in particular in optical information processing. The experimental implementations^{14–30} of reservoir computing (most of them optical) often report error rates comparable to the best digital algorithms. Most of these experiments, and in particular those that have been able to tackle the most complex tasks, are based on an architecture, introduced experimentally in¹⁵ (see also the earlier report³¹ and the theoretical proposal^{32,33}), consisting of a single nonlinear node and a delay line in which the reservoir states are time multiplexed. These experimental demonstrations are further complemented by extensive studies in simulation of alternative or improved optical implementations^{34–41}.

Despite this intensive research, the potential of reservoir computing in terms of processing easiness and speed has not yet been fully considered. In particular, all previous experiments required either digital pre-processing of the inputs, or digital post-processing of the outputs, or both (i.e. at least either the input layer or the output layer were digitally implemented). This is indeed a major limitation if one intends to use physical reservoir computers as versatile and efficient standalone solutions. Moreover, besides the advantages of speed and versatility, a fully analogue device would allow for the feedback of the output of the reservoir into the reservoir itself, enabling new training techniques⁴² as well as the exploitation of reservoir computers to new kinds of tasks, such as pattern generation^{3,43}.

Note that some steps towards a fully analogue reservoir have already been taken. In our unpublished manuscript⁴⁴ we showed how to implement an analogue input layer. In fact an analogue input layer is comparatively simpler to implement, as it consists of multiplying the input signal with random weights. The exact values of these

¹OPERA-Photonique, CP 194/5, Université Libre de Bruxelles (U.L.B.), Avenue Adolphe Buyl 87, 1050 Brussels, Belgium. ²Laboratoire d'Information Quantique, CP 224, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, 1050 Brussels, Belgium. Correspondence and requests for materials should be addressed to S.M. (email: smassar@ulb.ac.be)

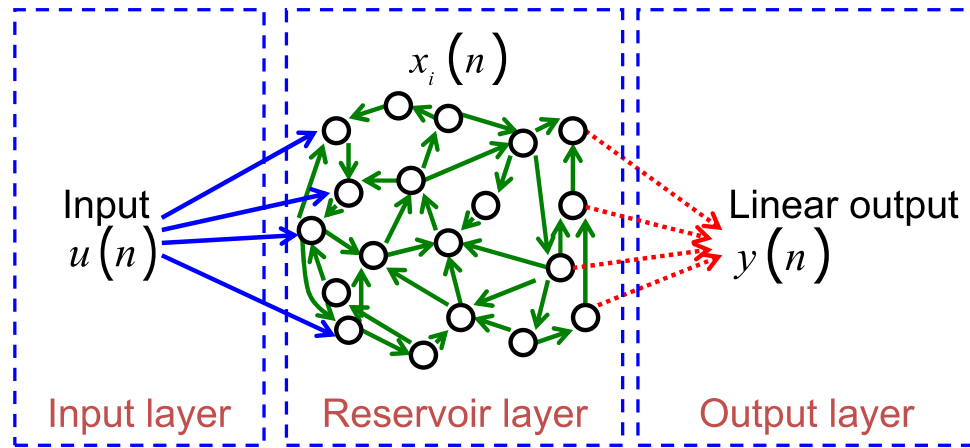


Figure 1. Schematic of a reservoir computer. The input $u(n)$ is sent to the reservoir layer by the input layer. The reservoir layer is a nonlinear recurrent dynamical system whose internal variables are denoted $x_i(n)$. The output $y(n)$, produced in the output layer, is a linear combination of the internal variables. Only the linear output layer is adjusted during the training phase.

weights are not very important, as they can be chosen at random up to some global scaling. Optimisation of the input weights has been considered in^{22,28,40}.

The first report of a reservoir computer with an analogue output layer was given in²⁰. This solution was tested on a single task, and the results obtained were not as good as those obtained using a digital output. The difficulty of constructing an analogue output layer is due to the nature of the computation that needs to be carried out. Indeed the output of the reservoir computer is a linear combination, with positive and negative weights, of many internal states, which requires a very high computation accuracy. While this accuracy is obtained naturally with a digital computer, it is rather challenging to reach it with an analogue integrating device.

In the present work we report the first fully analogue reservoir computer. Our implementation takes as input an analogue optical signal, and produces as output an analogue electrical signal proportional to the output requested by the task. We thereby prove that the concept of reservoir computing can be entirely implemented by means of analogue signals handled by analogue components. This opens up the route to new promising developments based on high-bandwidth standalone reservoirs as well as feedback loop reservoirs.

In what follows we first introduce the concept of reservoir computer, and then review the optoelectronic reservoir exploited in the present work. This reservoir, based on a single nonlinear node and a fibre delay loop, was introduced in^{17,18}. In Sec. 4 we describe the analogue input layer that we implemented in the present work on the basis of our previous study⁴⁴. The input mask (i.e. the weights that determine the coupling between the input and the reservoir states) is given by the sum of two sine functions which could be easily generated by oscillators and should be therefore more suitable for future integration. The analogue output layer discussed in Sec. 5 is an improved version of the solution developed in²⁰. In Sec. 6 we study the performance of the fully autonomous reservoir computer on three tasks that are traditionally considered in the reservoir computing community, namely NARMA10, nonlinear channel equalization and radar signal forecasting. Finally we discuss the implications of our work for the future development of photonic reservoir computing.

Reservoir Computing Basics

A reservoir computer, see^{11–13} for general presentations, is composed of three layers: an input layer, the reservoir itself and an output layer. The input signal is a time series $u(n)$ where $n \in \mathbb{Z}$ is the discrete time. The internal variables of the reservoir, also called reservoir states, are denoted $x_i(n)$, $i = 1, \dots, N$ with N the number of internal variables. The reservoir is a nonlinear dynamical system that recurrently couples the internal states to each other. The input layer distributes the input $u(n)$ to the reservoir states with coupling coefficients that vary for each internal state. These coefficients, also called the input mask, are often drawn from a random distribution. The fact that the input mask coefficients vary enriches the dynamics of the reservoir by breaking the symmetry that would occur if the same image of the input would be distributed to all the internal variables. The output layer linearly combines the internal variables $x_i(n)$ to construct the output $y(n)$ of the system. Whereas in traditional recurrent neural networks, all the layers are optimised, the reservoir computing technique proposes to only optimise the output layer while the input layer and interconnections within the reservoir are fixed. The optimization of the output layer consists in a linear regression, which greatly simplifies the training of reservoir computers. It is thus possible to use large reservoirs and obtain processing capabilities as good as with other machine learning techniques.

The dynamics (in discrete time) of the internal variables are described by the evolution equation

$$x_i(n) = F_{NL} \left(\alpha \sum_j A_{ij} x_j(n-1) + \beta m_i u(n) \right), \quad (1)$$

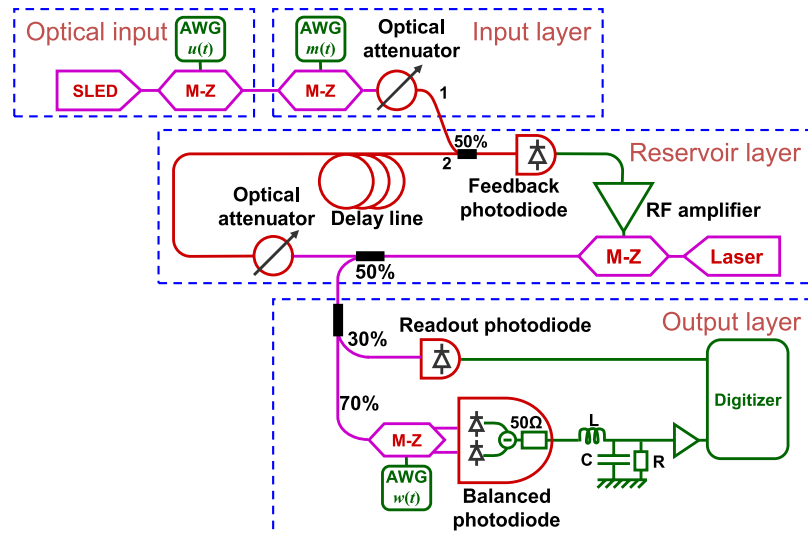


Figure 2. Schematic of the experimental reservoir computer. The optical input produces the signal that must be processed. The input layer multiplies the input signal by the input mask. The reservoir layer is a delay dynamical system in which a M-Z modulator acts as non-linearity. The output layer produces an analogue electric signal proportional to the desired output. Electric components are in green, optical components in red and purple, with purple used for polarization maintaining fibre components (used to avoid the use of polarisation controllers) and red for non polarisation maintaining fibre components. AWG denotes Arbitrary Waveform Generator; RF amplifier denotes Radio Frequency amplifier; R, L, C denote resistor, inductor and capacitor, respectively.

where the matrix A_{ij} represents the interconnections between the internal variables x_i . This matrix is normalized so that its largest eigenvalue is equal to one. The vector m_i represents the input mask. The two parameters α and β are used to adjust the strength of the feedback signals inside the reservoir and the strength of signals injected into the reservoir. In digital implementations, the nonlinear function F_{NL} is often taken to be a hyperbolic tangent, but many other nonlinear functions are satisfactory (below we use a sine function). The reservoir output $y(n)$ consists in a linear combination of the internal variables x_i with weights given by the vector W_i , that is,

$$y(n) = \sum_i W_i x_i(n). \quad (2)$$

The output weights W_i are chosen as follows. For a given choice of parameters α and β , a “training” input series $u(n)$ is injected in the reservoir and the internal variables are recorded. One then computes the W_i that minimize the Normalized Mean Square Error (NMSE) between the output of the reservoir $y(n)$ and the targeted output $\hat{y}(n)$

$$NMSE = \frac{\langle (\hat{y}(n) - y(n))^2 \rangle_n}{\langle (\hat{y}(n) - \langle \hat{y}(n) \rangle)^2 \rangle_n}. \quad (3)$$

Finally, the quality of the optimized output is evaluated by means of a “test” input series $u(n)$ injected into the reservoir while the coefficients W_i are kept fixed to their computed optimum values.

Optoelectronic Reservoir Layer

The experimental setup is depicted in Fig. 2. In the present section we describe the reservoir layer, and the next two sections are devoted to the input and readout layers. Further details on the experimental setup can be found in the Supplementary Methods and Supplementary Figure.

The reservoir layer consists of a delay line and a single nonlinear node. Similar systems have been studied previously in the general context of nonlinear dynamics, see e.g.^{45–47}. This reservoir layer is essentially identical to the optoelectronic reservoir used in¹⁷, as well as in^{18,20,22,26,28,44}. For a general overview of reservoir computing with delay dynamical systems see⁴⁸. Such nonlinear dynamical systems with delayed feedback have intrinsically continuous time dynamics, while the fundamental equation of reservoir computing equation (1) is expressed in discrete time. Here below we will go back and forth between the continuous and the discrete time descriptions depending on the context. The correspondence between the two time scales is clearly stated when necessary.

The delay line consists of a spool of optical fibre (~1.7 km of SMF28e). The internal variables x_i are time multiplexed along the delay line. They are represented by the light intensity that travels along the delay line within fixed temporal windows. At the end of the fibre the optical feedback signal is added to the optical input signal injected into the cavity by the use of a 50% fibre coupler and converted to a voltage by the feedback photodiode (TTI TIA525 with a cut-off frequency of 125 MHz). The resulting signal is then amplified (ZHL-32A amplifier with a bandwidth of 50 kHz–130 MHz and a gain of 27 dB) to drive a Mach-Zehnder (M-Z) light intensity modulator

(Photline MXAN-LN-10 with a bandwidth of 10 GHz). It modulates the light of a DFB laser (Covega DFB laser -SFL-1550p-NI- with a wavelength around 1550 nm). The sine response of this M-Z modulator is used as the non-linearity of the reservoir (nonlinear function F_{NL} in eq. (1)). During the experiments, the bias point of the M-Z modulator is regularly tuned to ensure a proper sine response. In other words, if no signal is applied to the RF port of the M-Z modulator, its transparency is at 50%. (Note that in some works¹⁸ the bias point of the M-Z is considered as a tuneable parameter which allows one to modify the nonlinear function F_{NL} . Here the bias point, and hence F_{NL} is kept fixed). At the output of the M-Z modulator, 50% of the light intensity is picked up for the readout layer, the remaining 50% is attenuated by a tuneable optical attenuator (Agilent 81571A) before going into the optical delay line. This optical attenuator allows adjusting the feedback gain of the cavity (α coefficient in eq. (1)). The round-trip time of the cavity is $T \approx 8.4 \mu\text{s}$. If we omit the constant part of the signal (that is in any case filtered out by the amplifier), the dynamics of the system without input can be approximated by

$$x(t) = \sin(\alpha x(t - T)). \quad (4)$$

In order to carry out computation we drive the cavity with a desynchronised signal as in¹⁷. (Note that an alternative way to couple the internal variables to each other proposed in^{15,18} exploits a low-pass filter in the cavity). To this end, consider a reservoir of N internal variables. We define the duration θ of each internal variable by the relation $T = (N + k)\theta$ where we recall that T is the round-trip time, and k denotes the degree of desynchronisation. We convert the discrete time input $u(n)$ into a continuous signal $u(t)$ by a sample and hold procedure for duration $T' = N\theta$. Thus in continuous time, the input of the reservoir computer is represented by a step signal given by $u(t) = u(n)$ for $t \in](n - 1)T', nT']$.

The value of the internal variable $x_i(n)$ is given by the average value of $x(t)$ in the interval $[(i - 1)\theta + (n - 1)T', i\theta + (n - 1)T']$. The internal variable $x_i(n)$ is set to the input $u(n)$ multiplied by the input mask value m_i . In continuous time the input mask is represented by a periodic function $m(t)$ of period T' .

The continuous time dynamics of the driven system can thus be described by

$$x(t) = \sin(\alpha x(t - T) + \beta m(t)u(t)). \quad (5)$$

In discrete time, the variable $x_i(n)$ is connected to $x_{i-k}(n-1)$ if $i > k$ or to $x_{N+i-k}(n-2)$ if $i \leq k$. The corresponding dynamics in discrete time is thus given by

$$\begin{cases} x_i(n) = \sin(\alpha x_{i-k}(n-1) + \beta m_i u(n)), & k < i \leq N \\ x_i(n) = \sin(\alpha x_{N+i-k}(n-2) + \beta m_i u(n)), & 1 \leq i \leq k \end{cases} \quad (6)$$

which is a particular case of equation (1).

Analogue Input Layer

In a reservoir computer based on a delay dynamical system with a single nonlinear node, the input mask m_i plays a crucial role as it breaks the symmetry of the system, giving each internal variable $x_i(n)$ a different dependence on the inputs $u(n)$. For this reason the optimisation of the input mask has been the subject of several studies^{22,28,40}. In the present implementation the input mask $m(t)$ is introduced independently of the input and is intrinsically continuous, which greatly simplifies its hardware implementation.

The optical input signal is generated as follows. A broadband source (superluminescent light emitting diode, SLED Denselight DL-CS5254A) is modulated using a Mach Zehnder (M-Z) modulator (Photline MXAN-LN-10) to generate an optical signal proportional to the input $u(t)$ of the reservoir computer (see methods for the pre-compensation). This precompensated input signal is generated by an Arbitrary Waveform Generator (AWG) with a sample rate close to 200 MS/s and a resolution of 16 bits (NI PXI-5422). The intensity profile of the optical signal sent to the reservoir computer is thus given by

$$I_{in}(t) = I_0 u(t), \quad (7)$$

where the input has been scaled to belong to the interval $u(t) \in [0,1]$.

The multiplication by the input mask is achieved with the same sample rate and resolution (200 MS/s and 16 bits) by another AWG (Tabor WW2074) that drives an additional M-Z modulator (Photline MXAN-LN-10). The optical signal after multiplication by the input mask has intensity

$$I(t) = m(t)I_{in}(t) = m(t)u(t)I_0, \quad (8)$$

where the mask is scaled to belong to the interval $m(t) \in [0,1]$, and for simplicity we have not written the insertion losses of the M-Z modulator.

A tuneable optical attenuator (Agilent 81571A) is used to adjust the strength of the input signal (β coefficient in eqs (1), (5) and (6)). The use of an incoherent light source (the SLED) avoids interferences between the signal injected into the cavity and the signal already present in the cavity (which is coherent since it comes from a laser). Therefore, at the output of the 50% fibre coupler, the feedback photodiode produces an electrical signal proportional to $\alpha x(t - T) + \beta m(t)u(t)$ (compare with eq. (5)).

Concerning the choice of input mask $m(t)$, we use sinusoidal signals (as in our earlier work⁴⁴). The simplest mask signal of this type would be a single sine at frequency p/T' with p integer. In this case the AWG produces the voltage

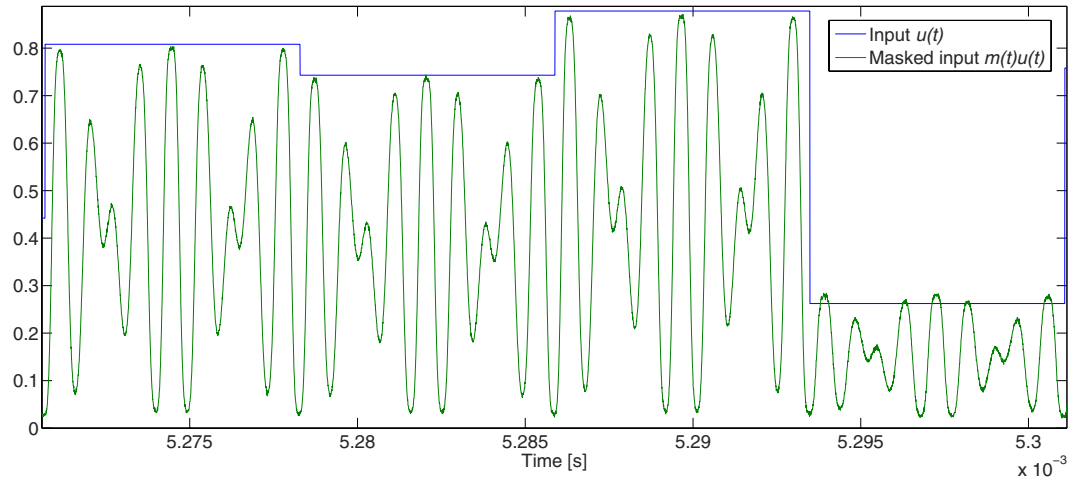


Figure 3. Signals injected into the reservoir layer. The blue curve is the optical input $I_{in}(t) = I_0 u(t)$. The green curve is a record of the masked input signal $m_{pq}(t)I_{in}(t)$ with $p = 7$ and $q = 9$, as measured by a photodiode and digitizer. The vertical axis is scaled so that its maximum range is $[0,1]$, i.e. $I_0 = 1$.

$$f_p(t) = -\frac{V_\pi}{2} \cos\left(2\pi \frac{p}{T'} t\right), \quad (9)$$

where V_π is the half-wave voltage of the M-Z modulator. We do not use any precompensation (as it would be incompatible with the use of a simple oscillator), which does not affect the system performance⁴⁴. The operating point of the M-Z modulator is adjusted so that the mask is

$$m_p(t) = \frac{1}{2} \left(1 + \sin\left(-\frac{\pi}{2} \cos\left(2\pi \frac{p}{T'} t\right)\right) \right), \quad (10)$$

Note that the phase of the cosine in equation (9) is chosen in order to ensure that the mask vanishes at times $t = nT'$ when the input $u(t)$ has discontinuities. The signal sent into the cavity is thus a smooth function without any discontinuity and the synchronisation between the input signal and the mask is drastically simplified.

With the simple input mask equation (10), the obtained performances depend strongly on the value of p . For a good choice of p , the results are close to those we can obtain with a random input mask. However, this is true only when the output is postprocessed digitally. When the results are obtained with the experimental analogue readout layer they are significantly less good than those resulting from a random mask (we do not have a good explanation for this).

The performance is significantly improved when we use an input mask m_{pq} containing two frequencies p/T' and q/T'

$$m_{pq}(t) = \frac{1}{2} \left(1 + \sin\left(-\frac{\pi}{4} \cos\left(2\pi \frac{p}{T'} t\right) - \frac{\pi}{4} \cos\left(2\pi \frac{q}{T'} t\right)\right) \right). \quad (11)$$

The resulting optical signal fed into the cavity is again null for $t = nT'$, ensuring continuity. The mask equation (11) is the one that is used in the experiments reported here below. A trace of the masked input signal is given in Fig. 3. Note that it was necessary to scan the values of p and q to get good results.

Analogue Output Layer

General principle of the output layer. The readout layer is in charge of producing the output $y(n)$ of the reservoir. It consists of two parts, the first measures the internal states $x(t)$ of the reservoir. The second produces the output itself.

As shown in Fig. 2, 30% of the light intensity sent to the reservoir layer is detected by the readout photodiode (TTI TIA525). The resulting signal is recorded by a digitizer (NI PXI-5124) at 200 MS/s with a resolution of 12 bits and a bandwidth of 150 MHz. This signal is used during the training phase to compute the values of the internal variables $x_i(n)$ and of the readout coefficients W_i (following the method described below). The remaining 70% of light intensity is modulated by a dual output M-Z modulator (Photline MXDO-LN-10 with 10 GHz of bandwidth) using a signal produced by an AWG (Tabor WW2074). The two outputs of this modulator are complementary and detected by a balanced photodiode (TTI TIA527 with a cut-off frequency of 125 MHz and output impedance 50 Ω). The bias point of this modulator is regularly tuned to have a sine response. In other words, if no signal is applied on the RF port of the M-Z modulator, both outputs have a transparency of 50% and the signal at the output of the balanced photodiode is null. If a positive (negative) voltage drives the M-Z modulator, the signal at the output of the balance photodiode is positive (negative). The reason for constructing the readout layer in

this way is that the internal variables are given by the optical intensity inside the reservoir, hence their values are positive. But for processing information with the reservoir computer, positive and negative readout coefficients W_i are required. Using a dual output M-Z modulator coupled to a balanced photodiode enables us to modulate the internal variables by coefficients that are either positive or negative. The signal from the balanced photodiode is filtered by a low-pass RLC filter whose role is to carry out an analogue summation of the weighted internal variables. The output of the low-pass filter is then amplified before being recorded by the second channel of the digitizer (NI PXI-5124). The value of the resulting signal at every instant $t = nT'$ is the output of the reservoir $y(n)$.

Computation of the readout coefficients. The balanced M-Z modulator in the output layer is driven by a signal produced by an AWG. Using the method described below, one computes a continuous time weight function $w(t)$. The signal produced by the AWG is precompensated so that the signal at the output of the balanced photodiode is proportional to $w(t)x(t)$.

Denoting by $h(t)$ the impulse response of the RLC filter followed by the amplifier, the signal $y_c(t)$ detected by the second channel of the digitizer can be expressed as:

$$y_c(t) = (w(t)x(t)) * h(t) = \int w(\tau)x(\tau)h(t - \tau)d\tau. \quad (12)$$

Since we use a real (causal) filter, the integration in equation (12) is done over the interval $\tau \in]-\infty, t]$. The continuous time weight function $w(t)$ is a stepwise function of period T' defined by:

$$w(t) = W_i \text{ for } nT' + (i - 1)\theta \leq t < nT' + i\theta \text{ with } 1 \leq i \leq N \text{ and } n \in \mathbb{Z}, \quad (13)$$

where we recall that θ is the duration of each internal variable. The output of the reservoir computer $y(n)$ is a function of discrete time. It is equal to the continuous output $y_c(t)$ at time nT' : $y(n) = y_c(nT')$. It can be expressed as

$$y(n) = y_c(nT') = \sum_{r=0}^{n-1} \sum_{i=1}^N W_i \int_{rT'+(i-1)\theta}^{rT'+i\theta} x(\tau)h(nT' - \tau)d\tau. \quad (14)$$

In order to calculate the readout coefficients for the analogue readout layer, new internal variables $x_i(n)$ are defined by

$$x_i(n) = \sum_{r=0}^{n-1} \int_{rT'+(i-1)\theta}^{rT'+i\theta} x(\tau)h(nT' - \tau)d\tau. \quad (15)$$

In practice, the impulse response of the readout layer has a finite length. Let l be an integer such that the impulse response is shorter than lT' (i.e. $h(t) \approx 0$ for $t > lT'$), the sum over r in equations (14) and (15) can be limited to values of r from $n-1-l$ to $n-1$. Note that since the impulse response lasts longer than T' , the current output $y(n)$ contains contributions from the light intensity $x(t)$ up to l input periods in the past, which is a small difference with respect to the traditional reservoir computer, see equation (2). In our experiment, for the channel equalization task we use $l = 10$, and for NARMA10 and the radar signal forecasting, $l = 15$.

At the beginning of the experiment, we record the step response (response to the Heaviside function) of the analogue readout layer by applying a voltage step on the dual output M-Z modulator. The derivative of the recorded signal is the impulse response $h(t)$ of the analogue readout layer.

Note that a key point to obtain good results is to optimize the extinction of the signal when a readout weight equal to zero is applied. Because the two arms of the balanced M-Z have different insertion losses and different extinction ratios, the extinction of the signal is not obtained with a null voltage. That is why, after tuning the working point of the M-Z, we measure the voltage that corresponds to zero readout weight, and take this into account when we precompensate the readout mask.

During the training phase, we record the output $x(t)$ of the reservoir using the readout photodiode (first channel of the digitizer). This record is then combined with the impulse response $h(t)$ of the analogue readout layer to compute the new internal variables $x_i(n)$ (see eq. (15)). From these internal variables we compute the readout weights W_i using Tikhonov (Ridge) regularisation. The corresponding stepwise periodic signal $w(t)$ is normalized with the highest absolute value of W_i , so as to fit the maximum modulation capabilities of the analogue readout layer. The corresponding gain (the highest absolute value of W_i) is applied on the recorded signal after acquisition of $y_c(t)$ and finally an offset correction is applied.

Note that the AWG that produces the output signal $w(t)$ has a finite resolution, and therefore exhibits quantification noise which degrades the quality of the output $y_c(t)$. This effect is minimised if the amplitudes of the W_i are all comparable. This can be enforced by increasing the Ridge regularisation parameter. In the present experiments we found it useful to take a Ridge regularisation parameter 10 times larger than when we use a digital output layer. (Note that the Ridge parameter is generally used to avoid overfitting on a limited training data set. If the data set is large enough, and in the absence of the quantification noise, the Ridge parameter should be taken as small as possible in order to increase the precision of the output).

The performance of the analogue output layer is obviously dependent on the impulse response $h(t)$, and different tasks work better with different impulse responses. In practice we first tested numerically different choices of R, L, and C, and then implemented experimentally those that provide good results. Typical values used are R in the range 1.6 k Ω –10 k Ω , C in the range 760 pF to 1.2 nF, with L = 1.8 mH.

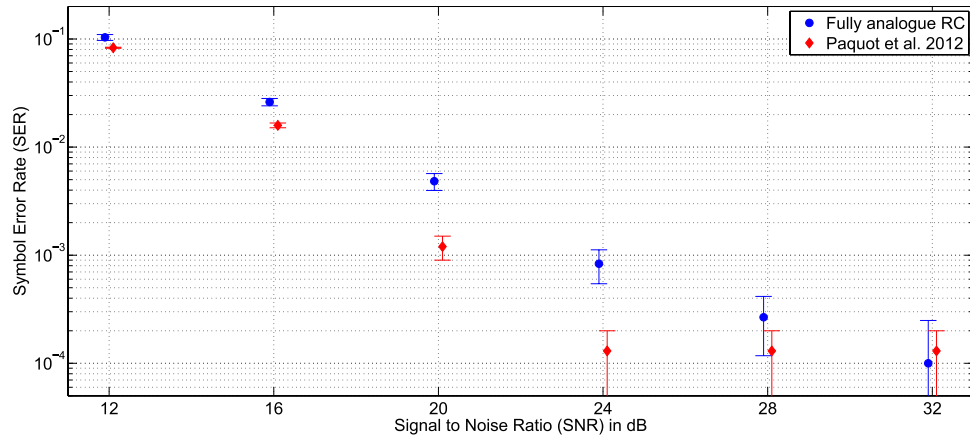


Figure 4. Results obtained for the equalization of the nonlinear channel for signal to noise ratios (SNR) ranging from 12–32 dB. For each SNR, the symbol error rate (SER) is given with its corresponding error bar over 5 datasets. The blue circles are the results obtained with the full analogue reservoir, and the red diamonds are the results presented in¹⁷ (similar optoelectronic reservoir computer, but without the analogue input and output layers).

Results

We tested the fully analogue reservoir computer on three tasks commonly considered in the reservoir computing community, namely equalization of a nonlinear communication channel, NARMA10, and the forecast of a radar signal. We compare the results with those obtained in¹⁷ in which a practically identical optoelectronic reservoir computer was used, and in the case of the radar task with the all optical reservoir¹⁹. Both¹⁷ and¹⁹ used a similar number of internal variables, but without analogue input and output layers.

In all cases we used 47 internal variables, so $N = 47$ and $k = 5$. The two frequencies of the input mask p/T' and q/T' , are $7/T'$ and $9/T'$. Either numerically before the experiment or during the experiment itself, the feedback gain (α) and the input gain (β) are scanned in order to find their optimal values. For each set of parameters, several datasets are used in the experiment in order to have sufficient statistics. In our experiment, a feedback gain α equal to 1 is obtained when the optical attenuator inside the loop is set to 9.5 dB. At this attenuation, when no input signal is applied small oscillations appear in the cavity. This corresponds to a maximum optical power received by the feedback photodiode (i.e. at maximum transparency of the M-Z modulator inside the loop) of 264.4 μ W. For comparison, the optical signal received by the feedback photodiode when the input is on, and the optical attenuator in the input layer is set to 0 dB, is 1.46 mW. When the input of the reservoir belongs to the interval $[0, 1]$, the input optical attenuation to obtain a β coefficient of 1 is around 7.4 dB.

It is important to note that we do not carry out any time averaging on the acquired signal $y_c(t)$. For this reason the output suffers from quantification noise (see discussion below). Moreover, note that because each data set is sent to the experiment twice (once to measure $x(t)$ and compute $w(t)$, once to measure $y_c(t)$), the stability of the experiment is more important than in experiments with digital postprocessing. To ensure stability, we regularly adjust the working points of all M-Z modulators.

Nonlinear channel equalization. The aim of this task is to compensate for the distortion of a wireless communication link affected by a small nonlinearity and a memory effect. It was used previously in the reservoir computing literature, see e.g.^{3,33}. A sequence of symbols $d(n)$, randomly drawn from the set of values $\{-3, -1, 1, 3\}$, passes through a channel model with inter-symbols interferences (due to multi-path travels and/or band-pass filters at the channel ends) followed by a nonlinear transformation:

$$\begin{aligned} q(n) &= 0.08d(n+2) - 0.12d(n+1) + d(n) + 0.18d(n-1) - 0.1d(n-2) \\ &\quad + 0.091d(n-3) - 0.05d(n-4) + 0.04d(n-5) + 0.03d(n-6) + 0.01d(n-7) \\ u(n) &= q(n) + 0.036q^2(n) - 0.011q^3(n) + \text{noise}. \end{aligned} \quad (16)$$

The signal to noise ratio (SNR) is scanned from 12–32 dB using a step of 4 dB. The input of the reservoir computer is the noisy and distorted sequence $u(n)$, while the target output is the original sequence of symbols $d(n)$. For each SNR, the quality of the equalization is given by the symbol error rate (SER). We use 5 different datasets. For each dataset, the reservoir is trained over 3000 time steps, and then a second sequence of 6000 time steps is used to test its performances (evaluate the SER). Results are presented with their corresponding standard deviation in Fig. 4. A slight degradation is observed compared to the results obtained in¹⁷. The presented results are significantly better than those presented in²⁰ (for instance at SNR of 32 dB, a SER of 10^{-4} compared to 10^{-2}). This is due in part to the larger number of internal variables that are used (47 instead of 28), but also to a better characterisation of the output layer, and a better choice of the output filter impulse response.

For this task, the feedback optical attenuator is set to 11.25 dB, the input optical attenuator is set to 5 dB, and the analogue output layer had parameters $R = 1.6 \text{ k}\Omega$, $C = 1.2 \text{ nF}$, $L = 1.8 \text{ mH}$. The measured impulse and step

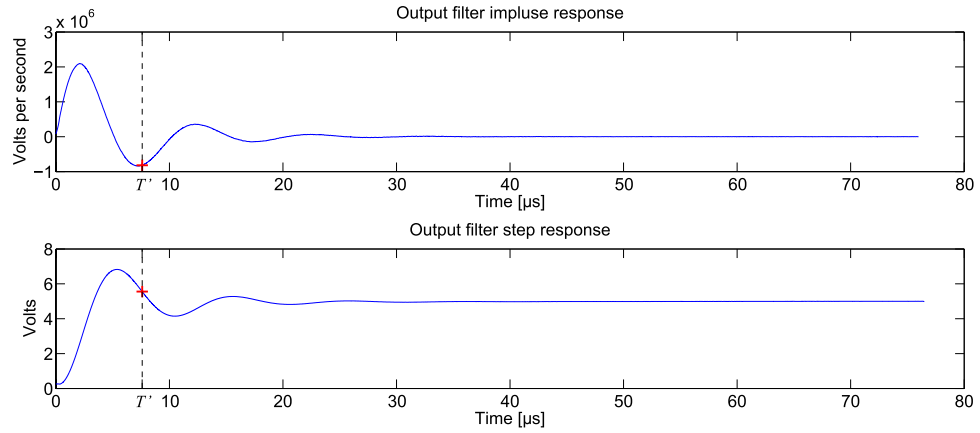


Figure 5. Impulse and step responses of the analogue readout layer used for the equalization of a nonlinear channel. The step response is recorded at the beginning of the experiment. Its derivative gives the impulse response of the analogue readout layer. The red cross gives the signal value at $T^* = 7.598 \mu\text{s}$.

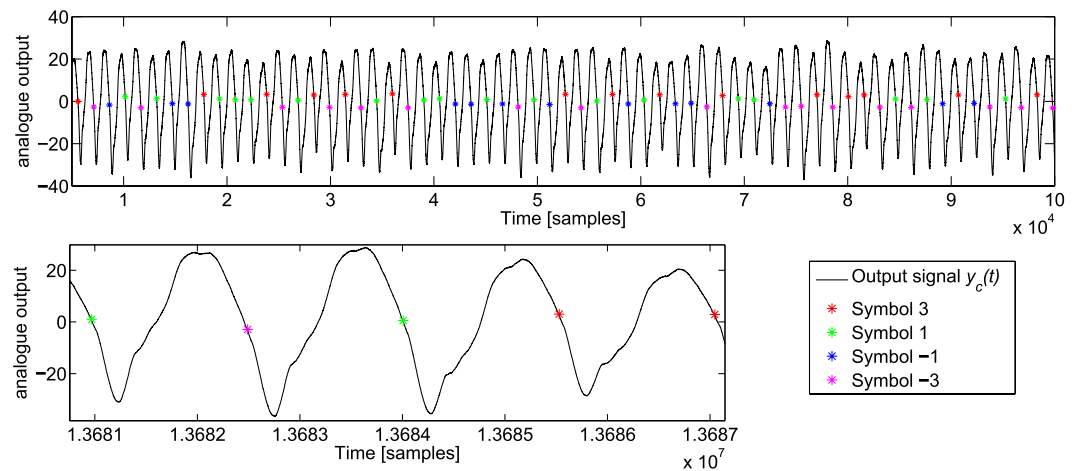


Figure 6. Signal at the output of the analogue readout layer for the nonlinear channel equalization task. The time is in number of samples (at 200 MS/s). The black curve is the acquired signal $y_c(t)$ with a final gain correction (multiplication by the maximum absolute value of the readout weights W_i). The stars are the output values $y_c(nT^*) = y(n)$. The different colours correspond to the different symbol values.

responses of the analogue readout layer are given in Fig. 5. A sample of the readout signal $y(t)$ is given in Fig. 6. A plot of the readout weights W_i is given in the supplementary material.

NARMA10. The aim of this task is to train a reservoir computer to behave like a 10th order Nonlinear Auto Regressive Moving Average system (NARMA10) in which an input $u(n)$, randomly drawn from a uniform distribution over the interval $[0,0.5]$, is injected. The following equation defines the targeted output:

$$\hat{y}(n+1) = 0.3\hat{y}(n) + 0.5\hat{y}(n) \left(\sum_{i=0}^9 \hat{y}(n-i) \right) + 1.5u(n-9)u(n) + 0.1. \quad (17)$$

For this task, the reservoir is trained over a sequence of 1000 time steps and tested over another sequence of 1000 time steps, this process is repeated 10 times to obtain the statistics. The performance on this task is measured using the *NMSE*. This task is commonly studied in the reservoir computing community, see e.g.^{33,49}.

For this task, the feedback optical attenuator is set to 9.2 dB (i.e. slightly above the threshold for oscillations), the input optical attenuator is set to 9.5 dB, and the analogue output layer had parameters $R = 10 \text{ k}\Omega$, $C = 760 \text{ pF}$, $L = 1.8 \text{ mH}$. The impulse response of the analogue readout layer is given in Fig. 7.

The test *NMSE* for the all-analogue system is 0.230 ± 0.023 . For the sake of comparison note that a reservoir that carries out no computation (i.e. produces a time independent output $y(n) = \text{const}$) has a *NMSE* = 1, the system reported in¹⁷ provides a *NMSE* of 0.168 ± 0.015 , an ideal linear shift register (no nonlinearity in the reservoir) can reach a *NMSE* of 0.16, and using a different experimental architecture based on a coherently driven passive

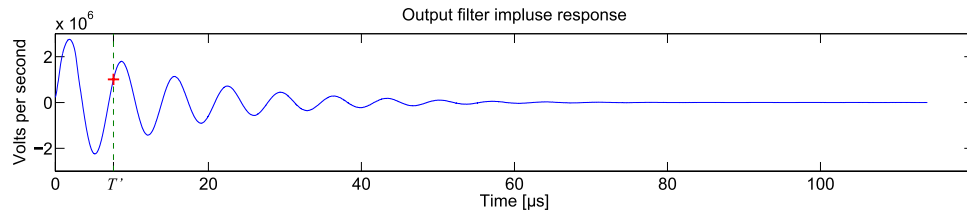


Figure 7. Impulse response of the analogue readout layer for NARMA10. The red cross gives the signal value at $T' = 7.598 \mu\text{s}$.

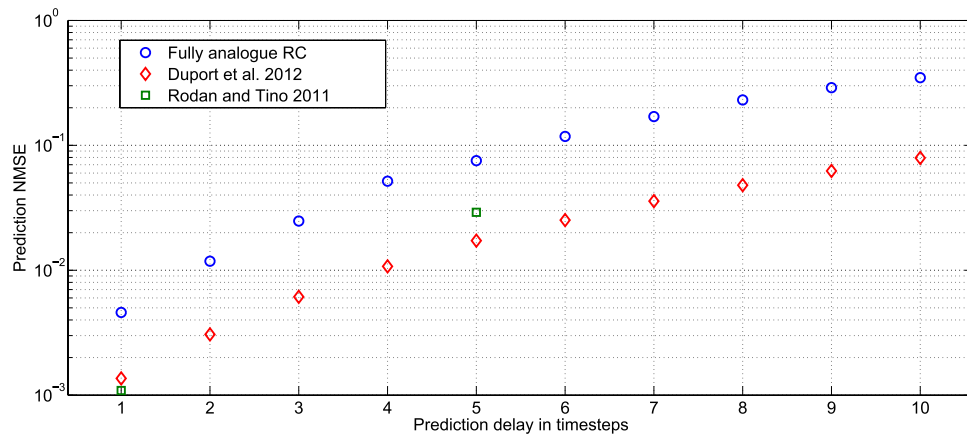


Figure 8. Radar signal forecasting error (*NMSE*) with respect to the number of time steps of the prediction (one to ten time steps in the future). The blue circles are the results obtained with the fully analogue reservoir, the red diamonds are the results published in¹⁹ obtained with an all optical reservoir computer with similar number of internal variables, but without the analogue input and output layers. The green squares are the numerical results of³³.

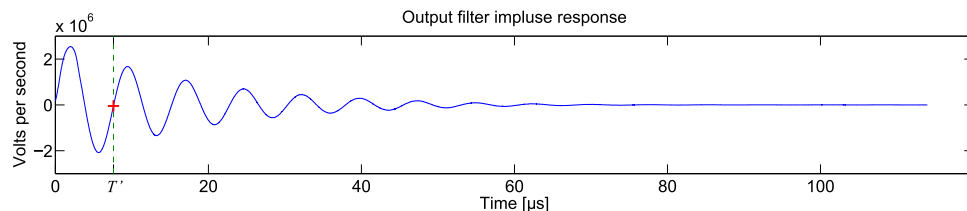


Figure 9. Impulse response of the analogue readout layer used for the radar signal forecasting. The red cross gives the signal value at $T' = 7.598 \mu\text{s}$.

cavity a *NMSE* as low as 0.107 ± 0.012 was reported in³⁰. Note that the all-analogue performance is slightly worse than that of the linear shift register but significantly better than a system that carries out no computation.

Radar signal forecasting. This task consists in predicting a radar signal one to ten time steps in the future from a radar signal backscattered from the ocean surface (data collected by the McMaster University IPIX radar). The quality of the forecasting is evaluated by computing the *NMSE* between the predicted signal and the actual data one to ten time steps in the future. The experiment uses a single recorded radar signal under low sea state conditions, corresponding to an average wave height of 0.8 meters (max 1.3 m). The recorded signal has two dimensions, corresponding to the in-phase and in-quadrature outputs (respectively, *I* and *Q*) of the radar demodulator. Therefore for each dataset, the in-phase and in-quadrature signals are successively processed (predicted) by the experiment. The training and test sequences contain 1000 inputs each. This task has been previously used to evaluate the performance of reservoir computers, see e.g.^{33,50}. The results are presented in Fig. 8.

For this task, the feedback optical attenuator is set to 9.9 dB, the input optical attenuator varied between 7 and 10 dB, and the analogue output layer had parameters $R = 10 \text{ k}\Omega$, $C = 810 \text{ pF}$, $L = 1.8 \text{ mH}$. The impulse response of the analogue readout layer is given in Fig. 9.

Discussion

The novel feature of the experimental reservoir computer presented here is the simultaneous inclusion of analogue input and output layers. The interest of this configuration is that it represents the necessary step towards develop standalone reservoir computers for future complex and high-bandwidth applications. The only role of the external computer in our experiment is to compute the output weight function $w(t)$.

Concerning the analogue input layer, we proposed the use of sinusoidal functions as input mask, as these will be simple to generate in future hardware implementations. Upon using the sum of two sines as input mask, we did not observe significant degradation of performance compared to using the standard random step function.

The present analogue input layer is well suited to scalar input signals. However there are many problems in which reservoir computers are used to process a vector input signal $u_i(n)$. This is for instance the case in the radar task of Sec. 6.3 in which both the I and Q channels may be used as input (we used here either I or Q), and in many speech recognition tasks where the input consists of several auditory channels. We shall address this issue in forthcoming works.

Concerning the analogue output layer whose aim is to produce a linear combination of the internal states that yields the desired output, the key difficulty is the accuracy needed in the summation that involves a large number of adjustable factors (the output weights).

The results presented here are obtained without any temporal averaging of the recorded signal, which makes them sensitive to quantification noise. This is important in our case since the total range of the output signal $y_c(t)$ is much larger than the range of the outputs $y_c(nT) = y(n)$, see Fig. 6. In the case of the channel equalisation task, which essentially constitutes a classification task, quantification noise is not such a problem since a signal that is correctly classified will in general continue to be so if a small amount of noise is added. But in the case of NARMA10 and the radar task, we measure the performance by how close the output is to the desired output using the NMSE. Quantification noise then directly affects the performances. (Note that the effects of quantification noise and methods to counteract it have been studied previously in the context of reservoir computing in^{22,26}).

Quantification noise also affects the readout mask $w(t)$. For this reason the Ridge regularisation parameter was optimised in order to minimise the range of $w(t)$, as discussed in section 5.2.

We emphasize that for different tasks, different output filters were used (values of the constants R , L and C). We do not have a complete explanation of why the optimal filters are different for each task. In our previous experiment²⁰, we used a simpler RC filter. This filter typically has a long impulse response, but on the other hand the resulting signal is much smaller, which leads to an increase of the output quantification noise. In the present work, we used a second order RLC filter that also exhibits a long impulse response, but keeps a larger signal range.

In summary, we proposed in this work the first study of a fully analogue reservoir computer. At stake is the development of future analogue computers dedicated to complex and high-bandwidth signal-processing tasks. Due to the added complexity of our experiment, some degradation of performance is naturally observed compared to previous experiments in which the input and output layers were implemented digitally through digital pre and post processing. However, the present experiment can be considered as a proof of principle that suggests the feasibility of fully standalone reservoir computers. In this sense our work can also be seen as an important step towards the development of novel applications in which reservoir computers are cascaded or looped on themselves. As emphasized in the above discussion, many technical problems remain to be solved. For instance, to circumvent some of the difficulties related to the use of fast-electronics we are currently addressing the possibility of implementing an all-optical output layer.

References

1. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks. *Technical Report GMD Report 148*, German National Research Center for Information Technology (2001).
2. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
3. Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
4. Steil, J. Backpropagation-decorrelation: online recurrent learning with $O(N)$ complexity. *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, IEEE, Vol. 2, 843–848; doi: 10.1109/IJCNN.2004.1380039 (2004).
5. Verstraeten, D., Schrauwen, B., D’Haene, M. & Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Netw.* **20**(3), 391–403 (2007).
6. Triefenbach, F., Jalal, A., Schrauwen, B. & Martens, J.-P. Phoneme recognition with large hierarchical reservoirs. *Adv. Neural Inf. Proces. Syst.* **23**, 2307–2315 (2010).
7. Boccatto, L., Lopes, A., Attux, R. & Von Zuben, F. J. An echo state network architecture based on Volterra filtering and PCA with application to the channel equalization problem. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, IEEE, 580–587; doi: 10.1109/IJCNN.2011.6033273 (2011).
8. Boccatto, L., Lopes, A., Attux, R. & Von Zuben, F. J. An extended echo state network using Volterra filtering and principal component analysis. *Neural Netw.* **32**, 292–302 (2012).
9. Buteneers, P. *et al.* Automatic detection of epileptic seizures on the intra-cranial electroencephalogram of rats using reservoir computing. *Artif. Intell. Med.* **53**, 215–223 (2011).
10. Antonelo, E. A., Schrauwen, B. & Stroobandt, D. Event detection and localization for small mobile robots using reservoir computing. *Neural Netw.* **21**, 862–871 (2008).
11. Lukoševičius, M., Jaeger, H. & Schrauwen, B. Reservoir Computing Trends. *KI - Künstliche Intelligenz* **26**, 365–371 (2012).
12. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* **3**, 127–149 (2009).
13. Lukoševičius, M. A Practical Guide to Applying Echo State Networks In *Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science* **7700**, 659–686 (Springer Berlin/Heidelberg, 2012).
14. Fernando, C. & Sojakka, S. Pattern recognition in a bucket in *Banzhaf*, Vol. 2801 (ed. Ziegler, W. *et al.*) *Advances in Artificial Life, Lecture Notes in Computer Science*, 588–597 (Springer Berlin/Heidelberg, 2003).
15. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).

16. Martinenghi, R., Rybalko, S., Jacquot, M., Chembo, Y. K. & Larger, L. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **108**, 244101 (2012).
17. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
18. Larger, L. *et al.* Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**, 3241–3249 (2012).
19. Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783–22795 (2012).
20. Smerieri, A. *et al.* Analogue readout for optical reservoir computers. *Adv. Neural Inf. Proces. Syst.* **25**, 944–952 (2012).
21. Brunner, D. *et al.* Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
22. Soriano, M. C. *et al.* Optoelectronic reservoir computing: tackling noise-induced performance degradation. *Opt. express* **21**, 12–20 (2013).
23. Hicke, K. *et al.* Information processing using transient dynamics of semiconductor lasers subject to delayed feedback. *IEEE J. Sel. Top. Quantum Electron.* **19**, 1501610 (2013).
24. Brunner, D., Soriano, M. C. & Fischer, I. High-speed optical vector and matrix operations using a semiconductor laser. *IEEE Photon. Technol. Lett.* **25**, 1680–1683 (2013).
25. Dejonckheere, A. *et al.* All-optical reservoir computer based on saturation of absorption. *Opt. Express* **22**, 10868–10881 (2014).
26. Soriano, M. C. *et al.* Delay-based Reservoir Computing: noise effects in a combined analogue and digital implementation. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 388–393 (2014).
27. Vandoorne, K. *et al.* Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun* **5**, 3541 (2014).
28. Hermans, M., Soriano, M. C., Dambre, J., Bienstman, P. & Fischer, I. Photonic Delay Systems as Machine Learning Implementations. *Journal of Machine Learning Research* **16**, 2081–2097 (2015).
29. Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I. & Gauthier, D. J. Reservoir computing with a single time-delay autonomous boolean node. *Phys. Rev. E* **91**, 020801(R) (2015).
30. Vinckier, Q. *et al.* High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica* **2**, 438 (2015).
31. Paquot, Y., Dambre, J., Schrauwen, B., Haelterman, M. & Massar, S. Reservoir computing: a photonic neural network for information processing. *Proc. SPIE* **7728**, 77280B; doi: 10.1117/12.854050 (2010).
32. Rodan, A. & Tino, P. Simple deterministically constructed recurrent neural networks. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, 11th International Conference, Paisley, UK, September 1–3, 2010. Proceedings, 267–274; doi: 10.1007/978-3-642-15381-5_33 (2010).
33. Rodan, A. & Tino, P. Minimum complexity echo state network. *IEEE Trans. Neural Netw.* **22**, 131–144 (2011).
34. Vandoorne, K. *et al.* Toward optical signal processing using photonic Reservoir Computing. *Optics Express* **16**, 11182–11192 (2008).
35. Vandoorne, K., Dambre, J., Verstraeten, D., Schrauwen, B. & Bienstman, P. Parallel reservoir computing using optical amplifiers. *IEEE Trans. Neural Netw.* **22**, 1469–1481 (2011).
36. Mesaritakis, C., Papataxiarhis, V. & Syvridis, D. Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system. *JOSA B* **30**, 3048–3055 (2013).
37. Nguimdo, R. M., Verschaffel, G., Danckaert, J. & Van der Sande, G. Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics. *Opt. Express* **22**, 8672–8686 (2014).
38. Zhang, H. *et al.* Integrated photonic reservoir computing based on hierarchical time-multiplexing structure. *Opt. Express* **22**, 31356–31370 (2014).
39. Fiers, M. A. A. *et al.* Nanophotonic Reservoir Computing With Photonic Crystal Cavities to Generate Periodic Patterns. *IEEE Trans. Neural Netw. Learn. Syst.* **25**, 344–355 (2014).
40. Appeltant, L., Van der Sande, G., Danckaert, J. & Fischer, I. Constructing optimized binary masks for reservoir computing with delay systems. *Sci. Rep.* **4**, 3629 (2014).
41. Mesaritakis, C., Bogris, A., Kapsalis, A. & Syvridis, D. High-speed all-optical pattern recognition of dispersive Fourier images through a photonic reservoir computing subsystem. *Opt. Lett.* **40**, 3416–3419 (2015).
42. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**(4), 544–557 (2009).
43. Jaeger, H., Lukosevicius, M., Popovici, D. & Siewert, U. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural netw.* **20**(3), 335–52 (2007).
44. Duport, F., Akrou, A., Smerieri, A., Haelterman, M. & Massar, S. Analogue input layer for optical reservoir computers. arXiv preprint arXiv:1406.3238 [cs.LG] (2014).
45. Larger, L., Lacourt, P. A., Poinot, S. & Hanna, M. From flow to map in an experimental high-dimensional electro-optic nonlinear delay oscillator. *Phys. Rev. Lett.* **95**, 1–4 (2005).
46. Chembo, Y. K., Colet, P., Larger, L. & Gastaud, N. Chaotic Breathers in Delayed Electro-Optical Systems. *Phys. Rev. Lett.* **95**, 2–5 (2005).
47. Peil, M., Jacquot, M., Chembo, Y. K., Larger, L. & Erneux, T. Routes to chaos and multiple time scale dynamics in broadband bandpass nonlinear delay electrooptic oscillators. *Phys. Rev. E* **79**, 026208 (2009).
48. Soriano, M. C., Brunner, D., Escalona-Moran, M., Mirasso, C. R. & Fischer, I. Minimal approach to neuro-inspired information processing. *Frontiers in Computational Neuroscience* **9**, 68 (2015).
49. Jaeger, H. Adaptive nonlinear system identification with echo state networks, In *Advance in Neural Information Processing Systems*, Vol. 15 (eds Becker, S. *et al.*), 593–600 (MIT Press, 2003).
50. Xue, Y., Yang, L. & Haykin, S. Decoupled echo state networks with lateral inhibition. *Neural Netw.* **20**, 365–376 (2007).

Acknowledgements

The authors acknowledge financial support by Interuniversity Attraction Poles Program (Belgian Science Policy) project Photonics@be IAP P7-35, by the Fonds de la Recherche Scientifique FRS-FNRS, by the Action de Recherche Concertée through grant AUWB-2012-12/17-ULB9.

Author Contributions

F.D., M.H. and S.M. conceived the experiment. F.D., A.A. and A.S. performed the experiment and the numerical simulations, supervised by S.M. and M.H. All authors contributed to the discussion of the results. The manuscript was written by F.D. and S.M. with help by the other authors.

Additional Information

Supplementary information accompanies this paper at <http://www.nature.com/srep>

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Duport, F. *et al.* Fully analogue photonic reservoir computer. *Sci. Rep.* **6**, 22381; doi: 10.1038/srep22381 (2016).



This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>