Review article **Open Access**

Guy Van der Sande*, Daniel Brunner and Miguel C. Soriano

Advances in photonic reservoir computing

DOI 10.1515/nanoph-2016-0132
Received July 31, 2016; revised November 30, 2016; accepted December 23, 2016

Abstract: We review a novel paradigm that has emerged in analogue neuromorphic optical computing. The goal is to implement a reservoir computer in optics, where information is encoded in the intensity and phase of the optical field. Reservoir computing is a bio-inspired approach especially suited for processing time-dependent information. The reservoir's complex and high-dimensional transient response to the input signal is capable of universal computation. The reservoir does not need to be trained, which makes it very well suited for optics. As such, much of the promise of photonic reservoirs lies in their minimal hardware requirements, a tremendous advantage over other hardware-intensive neural network models. We review the two main approaches to optical reservoir computing: networks implemented with multiple discrete optical nodes and the continuous system of a single nonlinear device coupled to delayed feedback.

Keywords: analogue computing; artificial neural networks; nonlinear optics; optical computing.

PACS: 42.79.Ta; 42.79.Hp; 42.65.-k; 42.82.-m; 85.60.-q; 42.55.Px; 05.45.-a; 07.05.Mh.

1 Introduction

Novel methods for information processing are highly desired in our information-driven society. Traditional von Neumann computer architectures or Turing approaches work very efficiently when it comes to executing algorithmic instructions. In terms of efficiency they run into trouble for highly complex or abstract computational tasks such as speech recognition or facial recognition. Our brain functions in a different way and seems to be excellently equipped for this kind of tasks. In our daily lives, we are constantly fed with impressions stemming from sensory information. Seeing a vehicle or a familiar face, hearing the ongoing traffic and conversations, and smelling the food stalls – all these external impulses instantly produce large neural activity in our brain and allow us to recognize the passing bus, a good friend, a car horn, or that smell of freshly baked waffles inducing the physical response of hunger. The neural network system that constitutes our brain is constantly processing these stimuli and uses underlying structures to interpret reality. In this, the human brain is highly efficient. Today, except for mathematical operations, our brain functions faster and much more efficient than any supercomputer. A recent estimate by Dharmendra Modha (IBM) suggests that emulating a human brain requires ~30 PFlops. Even today, only supercomputers provide such enormous computational performance – at an astronomical power consumption of ~10 MW. The human brain suffices with a mere ~20 W.

The information processing core of the human brain is formed by a neural network. Until today, research into information processing via artificial neural networks (ANN) is strongly dependent on advances in simulating ANN on von Neumann computing platforms. The highly successful deep learning algorithm can be seen as an illustrating example. Training deep neural networks requires a vast number of iterations optimizing the internal connections of the ANN. Though already in the 1970s identified as a promising computational concept [1], such ANNs could only be implemented recently by employing the newest generation of graphical processing units [2]. This technological breakthrough led to record-breaking state-of-the-art performances on several benchmarks such as computer vision [3]. Recently, AlphaGo, a deep learning algorithm by Google DeepMind trained for playing the board game Go [4], defeated a human professional player in the full-sized game of Go. The algorithm of AlphaGo is based on deep neural (feedforward) networks that are trained by a combination of supervised and reinforcement learning from games of self-play. Nevertheless, AlphaGo

***Corresponding author: Guy Van der Sande,** Applied Physics Research Group (APHY), Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium, e-mail: guy.van.der.sande@vub.ac.be
Daniel Brunner: UMR CNRS FEMTO-ST 6174/Optics Department, Université de Bourgogne Franche-Comté, 15 Avenue des Montboucon, F-25030 Besançon Cedex, France
Miguel C. Soriano: Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (CSIC-UIB), Campus Universitat de les Illes Balears, 07122 Palma de Mallorca, Spain

©2017, Guy Van der Sande et al., published by De Gruyter.
This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 3.0 License.
Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

consumes approximately 1 MW of power, exceeding power consumption of our brain by roughly four orders of magnitude.

To reproduce some of the brain's computational capabilities while circumventing limitations such as excessive power consumption, the field of neuromorphic computing starts to diverge from single-core von Neumann computer architecture. For example, IBM has developed the neuromorphic TrueNorth chip, consisting of more than one million spiking neurons. Consuming only 70 mW, the chip has a strongly improved power efficiency when implementing ANNs [5]. Furthermore, the SpiNNaker spiking neural network processor is used as part of the Human Brain Project's neuromorphic platform [6, 7]. Both systems present a large step towards an efficient hardware implementation of ANNs. Still, they rely on serial communication between neurons and a von Neumann approach to compute neuron responses. Though distributed in a highly parallel manner and located close to neurons, information is still stored in an isolated memory. Rather than true axon-like connections between individual neurons, connections are based on a serial bus technology. As a consequence, the system update rate is orders of magnitude below the bandwidth of individual components. Early work by Hopfield and Tank [8] suggested densely connected networks of *micro electronic neurons* for implementing ANN, while Denz [9] elaborated on optical implementations of ANN in nonlinear optical media. These foresighted suggestions had a common divisor: the analog hardware implementation of all aspects of an ANN, nodes (neurons) and network connections (axions and dendrites).

Structurally speaking, ANNs differ fundamentally from von Neumann processors. In a neural network, a large number of node and connection values are processed simultaneously. Contrarily, the core of each von Neumann processor can only compute a single value at a time. Due to its inherent parallelism, photonic technology is expertly suited for the creation of such networks. The exploitation of such parallelism in optics was first realized two decades ago. Early work exploited volume holographic elements for establishing connections between light-emitting diodes and detector arrays [10]. Using other bulk optical components, e.g. lenslet arrays, others focused on the implementation of an optical vector-matrix multiplier or of optical correlators [11]. While initially seen as promising, the developed schemes where limited by their fundamental working principle. Firstly, a one-to-one translation of ANN concepts to optical systems requires the implementation and accurate control of a large number of optical connections, something which quickly was found to be unrealistic. Secondly, early schemes relied on the nonlinear transformations by the neurons to be implemented electronically, limiting the energy efficiency of such systems.

Recently, interest into neuromorphic computing using photonics has been reinvigorated. The driving force behind this development is a novel paradigm of neuromorphic computing referred to as reservoir computing (RC). Crucially, the term *reservoir* originally referred to a large, randomly connected fixed network of nonlinear nodes or neurons. It was quickly realized that such random and fixed connections radically reduce the complexity for a hardware implementation in photonics as compared to earlier efforts in the 1990s. With RC there is no need for reconfigurable optical connection links. Various (more standard) photonic techniques can now be used to implement optical networks for RC with a wide range of network topologies. In this review, we will show how such recurrent networks for optical RC have been implemented either on a photonic chip or using diffractive optics. In addition, we show how novel, excitable spiking photonic devices move the field even closer to its biological inspiration, the human brain.

Going even further, not all reservoirs are neural networks. Analog physical systems such as the nonlinear behavior of ripples on a water surface have been used for information processing based on the RC paradigm [12]. RC therefore enables the implementation of neuromorphic computing avoiding the need of interconnecting large numbers of discrete neurons. In this review, we will revisit the concept of delay embedded RC, using only a single nonlinear node with delayed feedback. Contrary to optical network-based RC, nodes of a delay-based reservoir are implemented in a spatially continuous medium (i.e. the delay line). These nodes are considered virtual as they are not implemented as components or units in hardware. As a consequence, coupling between the virtual nodes of such a reservoir are intrinsically fixed. Nevertheless, delay-based RCs have shown similar performance as spatially distributed RCs, with the advantage that the hardware requirements are minimal as no complex interconnection structure needs to be formed. In photonics, it allows even for the use of hardware that is more traditionally associated with optical communications.

RC rekindled neuromorphic computing activities in photonics. Today, multiple photonic RC systems show great promise for providing a practical yet powerful hardware substrate for neuromorphic computing, both systems based on discrete nonlinear nodes (neurons), as well as implementations based on nondiscrete continuous systems. This development now opens the field of

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

nanophotonics for advanced implementations of neuro-inspired computational systems.

2 Reservoir computing

In Figure 1, we show the functional architecture of a standard reservoir computer as it is most generally implemented in software. To ease the discussion, we will consider that the reservoir is an ANN consisting of discrete neurons. It consists of three neural layers: an input layer (in red), the reservoir itself (in green), and one output layer (in blue). Here, for simplicity, we show a system with a one-dimensional readout. Information is injected into the reservoir according to input connectivity matrix W_{in}. For k input channels and a reservoir of N nodes, W_{in} is of dimension $(k \times N)$. The internal connectivity of the reservoir is defined by a connectivity matrix W_{int} of dimension $(N \times N)$. W_{out} determines the connection weights between reservoir and readout layer nodes. Here, an $(l \times N)$-dimensional readout matrix simultaneously creates l computations with a single reservoir.

RC finds its main merit in a simplification resulting from the particular properties of matrices W_{in}, W_{int}, and W_{out}. Introduced independently by Jaeger and Haas [13] and Maass et al. [14], their most striking innovation was the usage of random distributions for input connections W_{in} and internal connections W_{int}. Both matrices remain constant over time and do not participate in the training procedure. Training therefore is restricted to modifications to matrix W_{out}. Due to the random distribution of W_{int}, the resulting randomly connected reservoir naturally features a fraction of recurrent connections. As such, RC conceptually belongs to the field of recurrent neural networks (RNNs). Similar to connections with finite propagation speed in the brain, recurrence forms temporally delayed connectivity loops within the reservoir. Therefore, the current state depends on information originating from different earlier time steps: RNNs provide short-term or working memory. RC in particular allows feature extraction on complex time series with excellent performance [13–17]. Inside RNNs, this memory merges with computation, clearly going beyond the von Neumann concept. The processing capabilities are clearly due to the analogue dynamics within the network, although digital implementations have also been considered. The output layer is the only one which can be altered. This allows for parallel processing with an output layer which is higher dimensional. It also allows for the processing of new tasks after a rewiring of said layer alone.

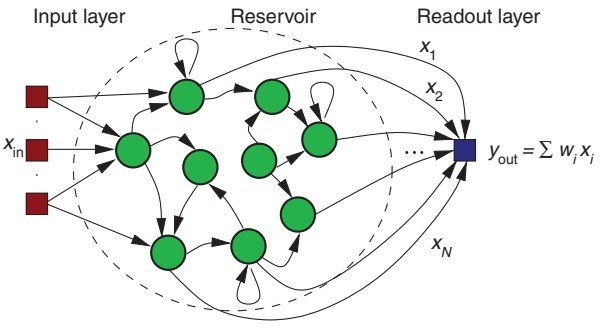

Figure 1: Standard layout of a reservoir computer, comprising an input layer (red), the reservoir (green) with randomized but fixed connections, and the linear readout layer (blue). Here, for simplicity a one-dimensional readout layer is drawn ($l = 1$).

The term *reservoir computing* was coined by Verstraeten et al. [17] in 2007. It was meant to unify two closely related RNN structures: the echo state network by Jaeger [18], described in 2001, and the liquid state machine by Maass et al. [14], in 2002. Though highly attractive, RNNs are notoriously difficult systems to train. The main intention of Jaeger and Maass was a strong simplification of the training algorithm of RNNs, originally motivating the random injection and internal connectivity matrices W_{in} and W_{int}. However, soon it was realized that random and temporally fixed connections are of enormous benefit for implementing RC in hardware. These features particular to RC will be described in depth in the following sections.

2.1 The hardware reservoir

Already indicated by one of its original names, liquid state machine, a reservoir corresponds to a nonlinear dynamical system. Allowing for a random W_{int}, the scheme is tolerant against variations of the internal interactions with the RNN. One of the few requirements on a reservoir is the following: internal weights W_{int} need to be scaled such that the reservoir is put into a suitable dynamical regime. Though task-dependent, typically one has to bring the reservoir close to an instability by globally scaling W_{int}, i.e. multiplying each interconnection strength with the same amount. Without external input the system should return to a quiescent state. Under such conditions a reservoir experiences fading memory: the longer ago information was injected, the less influence it is supposed to exert onto the current state of the reservoir.

These particular features make RC excellent for a implementation in complex dynamical systems. The state of the reservoir $X(n)$ is determined by

$$X(n) = f(W_{in} x_{in} + W_{int} X(n-1) + b), \quad (1)$$

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

where b is the vector of biases and $x_{in}(t)$ is sequentially injected input data. Generally, $x_{in}(t)$ is a k-dimensional input vector, which might be discrete or continuous in nature. The exact internal connectivity is not crucial as long as it is globally scalable. In fact, Rodan and Tino [19] showed that very simple and nonrandom internal connectivity leads to very promising computational performance. The same applies to data injection. Going even further, in Eq. (1), no specific nonlinear function $f(x)$ is defined. Countless nonlinear and high-dimensional hardware systems are therefore suitable for implementing Eq. (1).

Shortly after the conditions on Eq. (1) were strongly relaxed, the number of nonlinear dynamical systems exploited for RC increased in a short amount of time. Only 4 years after the initial demonstration of RC in hardware, experimental realization included a Mackey-Glass type nonlinearity [20], $\sin^2$ nonlinearities [21, 22], and a semiconductor optical amplifier (SOA) [23] as well as a semiconductor laser nonlinearity [24]. Among the main conclusions of these studies is the robustness of computational performance. Each of these systems has different system-specific properties such as dynamical parameters, different nonlinearities $f(x)$, and different internal (W_{int}) and external (W_{ext}) connectivity. Still, all systems fundamentally produced comparable figures of merit on a wide range of computational tasks.

Combining the simplifications and advantages introduced, RC has opened lines of research that go beyond common digital implementations and even beyond neural networks consisting of discrete elements. In principle, any dynamical system which has a high-dimensional phase space is a good candidate for RC [25]. The RC concept offers a highly attractive approach to neuromorphic computation in hardware. Substrate and reservoir implementation dimension can be chosen to maximally exploit system-specific properties for computation. Typical reservoirs comprise several hundred nodes, a complexity readily present in physical systems. Full analog physical implementations such as water ripples [12], mechanical oscillators [26], tensegrity structures [27, 28], soft bodies [29], and the optical devices and circuits which are the subject of this review have all been implemented. RC has indeed grown to include systems which are not necessarily based on a network topology of discrete components.

As opposed to digital implementations simulating RNNs, a physical system promises higher bandwidths, and parallelism and lower power consumption. The main obstacle for the practical implementation of reservoir computers (and indeed of all ANNs of considerate size) in physical substrates is that each of the many nodes has to be built and connected to the others: i.e. the connectivity problem. Integration on a photonic chip is one way to deterministically tackle this interconnection problem either by integrated waveguides or by imaging techniques. Nevertheless, the technology imposes limitations on the number and strength of the interconnections that can be achieved. Another approach is to use the natural circular connectivity present in delays systems and as such radically simplify the interconnection topology. We are convinced that nanophotonic concepts and devices could not only result in a breakthrough in terms of integration density and computation speeds in the presently considered optical RC implementations but also lead to novel architectures benefitting from nanophotonic's inherent properties. In this way, optical computing devices could revolutionize tasks where fast and energy-efficient processing is of the essence. Example applications include optical header recognition, optical signal recovery, and fast control loops.

2.2 The input layer

As defined in W_{in}, the connection between k-dimensional input data and an N-dimensional reservoir is randomly distributed. In a physical implementation, such a structured injection matrix corresponds to a random connection between a dynamical system's multiple degrees of freedom (reservoir) and an external modulation signal (injected information). The original motivation behind random injection was the creation of a highly diverse reservoir response [14, 18]. The computational power of ANNs generally relies on creating multiple nonlinear transformations of the same input information. Only then will a training procedure be capable to approximate the functional relationship for the desired computational operation. A random injection matrix therefore is a natural choice when aiming at maximizing the diversity of the reservoir responses without specifically optimizing W_{int}.

As with the random internal connectivity W_{int}, a random W_{in} as proposed by Jaeger and Maass had additional and strongly beneficial effects for hardware implementations. Physical implementation of such connections is highly practical. As is the case for the internal connectivity W_{int}, the injection connectivity W_{in} only has to be globally scaled. Combined with the fading memory property mentioned in Section 2.1, W_{in} and W_{int} have to be set such that the data-driven reservoir system features (i) the approximation and (ii) the separation property. The approximation property corresponds to consistency of a driven system [30, 31]: for multiple repetitions of identical input data, the system has to produce comparable

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

responses. The hardware reservoir then is robust against reservoir inherent and input data noise. The separation property demands that reservoir responses will sufficiently differ for multiple input data differing by more than noise. As shown by Uchida et al. [30] and Oliver [31], nonlinear dynamical systems typically can be brought into such state using a small number of global control parameters.

As for a hardware implementation of the reservoir itself, the data input procedure is highly flexible. The implementation dimension can again be chosen such that system-specific properties are maximally exploited. This is illustrated by comparing spatially and temporally implemented reservoir. The first approach requires the parallel realization of a large number of different spatial coupling coefficients. In photonics, this can be realized, e.g. by multimode imaging [24]. Still, a spatially distributed reservoir might require more involved device control. If one therefore targets the reduction of experimental complexity, one can implement a reservoir in a single device via a delay system. Information injection then changes to a single modulation signal multiplexed in time [20].

2.3 The training procedure

As introduced in the previous sections, Jaeger [18] and Maass et al. [14] created a neuromorphic computational concept utilizing dimensionality expansion based on random nonlinear mapping. The exploitation of such a mapping for information processing is illustrated in Figure 2. Information originating from an arbitrary measurement process is typically underrepresented: not all dimensions of a system's phase space can be accessed simultaneously. Going even further, for abstract problems it is often not apparent which those dimensions are. As a consequence, relevant computations can typically not be obtained from a linear operation upon available data. This situation corresponds to Figure 2A. Two different classes within the sample data (red and yellow spheres) cannot be linearly separated within the two-dimensional sample space. Mapping the two-dimensional data onto a high-dimensional reservoir, the random nonlinear mapping results in a dimensionality expansion. For a reservoir with a sufficient number of independent projections, one can find additional dimensions which allow for such a linear separation. This case is illustrated in Figure 2B. The possibility to find dimensions suitable for a linear separation becomes more likely with an increasing size of the reservoir.

Mathematically, the readout of a reservoir computer can therefore be formed by a linear combination of the reservoir states. During training, T consecutive input vectors are applied to the reservoir. The N will respond to these input sample, and each node state will therefore change. These node states are consecutively collected in an $N\times T$ state matrix S. For a multivariate l-dimensional output, the readout layer is defined by the $l\times N$ weights matrix W_{out}. Now, the goal is to choose this output weights matrix in such a way that the actual output $Y = W_{\text{out}}S$ matches the desired output $\hat{Y}$ as close as possible in the least-squares sense. This is a linear problem, for which the solution is calculated by using the Moore-Penrose pseudoinverse $S^{\dagger}$ of the state matrix S:

$$S^{\dagger} = (S^T S)^{-1} S^T, \tag{2}$$

$$W_{\text{out}} = (\hat{Y} S^{\dagger})^T. \tag{3}$$

Typically, standard RNN training is a formidable problem that is computationally involved. In fact, in RC only weighted connections from the state of the dynamical system to the output are trained. As a consequence, the reservoir itself and input layer connections remain unaltered and do not need to be reconfigured individually.

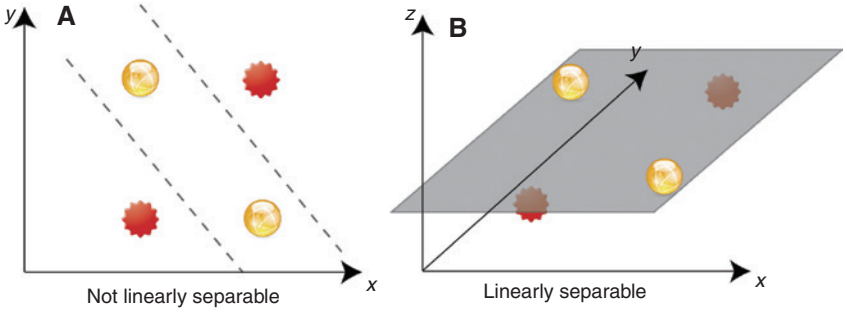

Figure 2: Illustration of the operation principle of reservoir computing. In its original (x, y) representation, two different classes of data cannot be linearly separated (left panel). Upon adding an additional dimension, (x, y, z) representation, a linear separation via a linear hyperplane can be found. Reservoir computing is based on the creation of additional dimensions to provide such hyperplanes. Figure adapted from Appeltant et al. [20].

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

Once the readout layer weights have been determined, the system can be used with new and unseen test data of the same class. Besides the significant reduced complexity of the training, the limited number of connections which have to be modified individually strongly aids implementations in hardware and mass production.

This is not to say that RC is without its own challenges. It is possible that the resulting system is overtrained and cannot generalize sufficiently to unseen inputs. Therefore, the training procedure in principle needs to be complemented with Tikhonov regularization or ridge regression techniques. However, in experimental implementations noise stemming from internal physical processes and from the measurement itself may be sufficient to counter overfitting [32]. All the optical implementations of RC discussed in this review use this training method. Many variations to this scheme exist, and an overview of current RC trends and software applications is given in Lukoševičius et al. [33].

Because the reservoir itself is not trained and only the output connections are, the RC concept can rely on any nonlinear dynamical system, as long as it exhibits consistent responses, a high-dimensional state space, and fading memory. As the training procedure does not modify the dynamical state of the reservoir, the l number of readout layer nodes corresponds to l-independent computations executed in parallel. The computational concept is therefore fully parallel, starting with multi-valued input data, continuing with the creation of high-dimensional reservoir responses until final computation of multi-valued output data.

3 Spatially distributed reservoir computing

Spatially extended reservoirs are the most intuitive implementations of a RNN. RNNs are typically illustrated as a complex network of spatially distributed nonlinear nodes. Such spatially extended networks allow for the implementation of various connection topologies. In the following, we will introduce multiple concepts how such spatially extended photonic networks have been implemented or suggested.

3.1 On chip silicon photonics reservoir computer

Motivated by a large-scale industry, silicon photonics is a platform of unparalleled appeal for technological implementations [34]. Once realized, photonic circuitry can be produced with the mature production technology of the silicon semiconductor industry. As such, a silicon photonic RC chip is an attractive system for ultra high speed and low-power consumption optical computing.

Already in 2008, Vandoorne et al. [35] suggested the implementation of photonic RC in an on-chip network of SOAs. Consequently, the computational performance of SOAs connected in a waterfall topology was evaluated numerically. The power-saturation behavior of a SOA resembles the nonlinear function of a hyperbolic tangent (tan$h$). This function can be computed efficiently, and as such numerical RC systems are often based on the tan$h$ as the nonlinearity of the nodes. For the first photonic RC, it was therefore intended to optically reproduce the encouraging performance of the numerical counterparts [35–37]. It was, however, quickly realized that constantly driving a SOA into power saturation results in poor energy efficiency.

For the first realization in hardware, Vandoorne et al. [38] therefore chose a different approach. A linear photonic network consisting of optical waveguides, optical splitters, and optical combiners was implemented using a Silicon-on-Insulator system. The resulting chip is shown in Figure 3A. Network nodes are indicated by the colored dots; blue arrows indicate topology of the network. In this way, the system acts as a very complex and random interferometer. The realization of such passive components is technologically mature. Still, the choice of a linear system might be perplexing at first glance, as it lacks an essential ingredient for RC: nonlinearity. What Vandoorne et al. realized is that the detection process via a standard fast photo diode solves this problem. A photodetector always detects optical power; hence, the detected signal of their photonic reservoir will be given by $X_n \propto ||E||^2$. However, the system cannot be operated all optically as it fundamentally relies on an optoelectronic conversion in the detector.

One advantage of working with passive elements is that they are relatively broad-band (few nanometers). Therefore, no precise control of wavelength is needed, and even several wavelengths could be sent through the system at the same time realizing parallel processing at different wavelength. Precise control of the optical phase between nodes is also not needed as it is exactly that diversity that leads to good processing capabilities. Novel learning techniques could be used to accommodate for phase drift over longer times. As a drawback, one could consider the increased optical losses as the chip is scaled to more nodes and the difficulty of measuring the response on all the nodes in parallel.

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

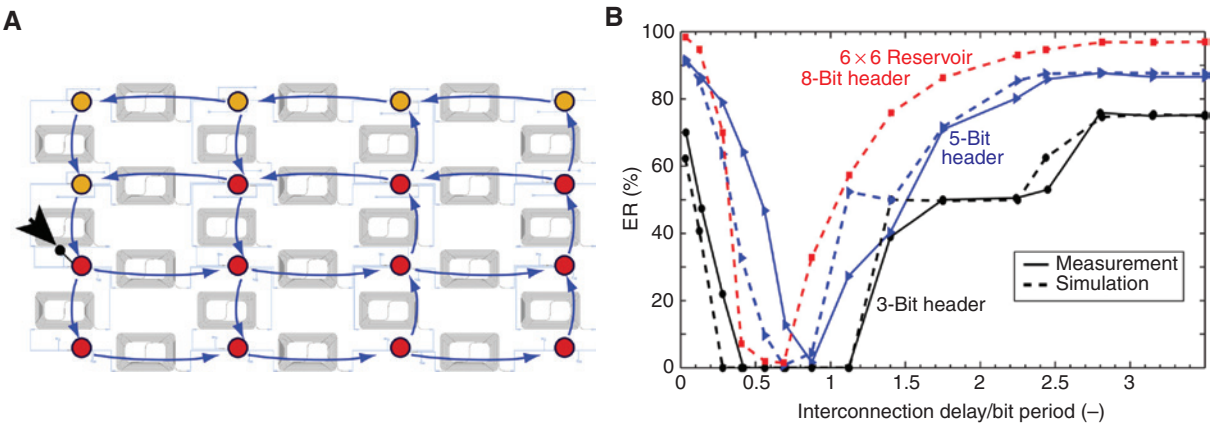

Figure 3: Reservoir implemented on a passive silicon chip [38]. (A) A linear optical network is implemented using optical waveguides, splitters, and combiners. Blue arrows illustrate the implemented reservoir connectivity. (B) Experimental and numerical evaluation of optical header recognition via on-chip RC (figure courtesy of Peter Bienstman).

In a network of passive elements, the timescale of the working memory and hence the input data clock frequency is dictated by the propagation delay between individual nodes. For typical distances on a photonic waveguide chip this would require hundreds of Gbit/s injection rates. While attractive for future technological implementations, experimental characterizations would be unrealistic using current modulation, detection, and arbitrary waveform generator technology. Vandoorne et al. therefore additionally separated each node by a photonic delay line in form of a spiral waveguide of 2 cm length. The spiral delay line structures can be seen in Figure 3A. The system can then be fed at data rates in the range of 0.12 up to 12.5 Gbit/s. A negative side effect of this downscaling mechanism is a rather large footprint of 16 mm^2 for a chip of 16 nodes, besides the increased optical losses in the bended waveguides.

Computational performance of the system was evaluated via multiple tasks. For that, information was optically injected (1531 nm) into the reservoir at a single point (black arrow in Figure 3A), which represents a massive simplification to the random input connectivity considered in software RC. Node responses were sampled individually by repeating the experiment several times and recorded by an optical sampling scope. Only 11 nodes where then used for experimentally evaluating the computational performance of the system (depicted in red). Figure 3B shows experimental results for the recognition or classification of optical headers of different bit lengths. For optical headers with a 5 bit length excellent results were obtained experimentally in a large range of the ratio of interconnection delay and bit period around 1. For longer bit sequences, larger chips need to be designed and the results shown are so far only numerical [38]. This functionality can be equivalently framed as matched filtering synthesized from random filter responses. Multiple further tests such as Boolean operations with memory and the classification of spoken digits were evaluated, both experimentally and in a numerical implementation of the system. In all tests, the system produced adequate results.

3.2 Diffractively coupled VCSEL as a RC

A second approach to the implementation of a spatially extended photonic reservoir is based on diffractive imaging using a standard diffractive optical element (DOE). In this way, Brunner and Fischer [24] demonstrated coupling inside a network of vertical cavity surface emitting lasers (VCSEL). Figure 4A shows a chip from Princeton Optronics hosting an array of 8×8 VCSELS, regularly spaced by a pitch of 250 μm. Due to the structure of a VCSEL, optical emission is directed vertical to the surface of the chip. A special feature of this device is that bias current of each laser can be controlled individually.

Coupling between individual lasers was realized based on diffractive multiplexing in an imaging setup. As schematically illustrated in Figure 4B, an image of the VCSEL array is formed on the left side of the imaging lens. Here, the VCSEL array lattice pitch combined with the focal distance of the imaging lens results in an angle ϕ between principal rays of neighboring lasers. This angle can be adjusted via the lens focal length. At the same time, a DOE beam splitter creates multiple copies of an incoming ray. These copies correspond to different diffractive orders, which are respectively spaced by the DOE angle offset θ. Based on the small angle approximation, it can then be shown that images formed by the

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

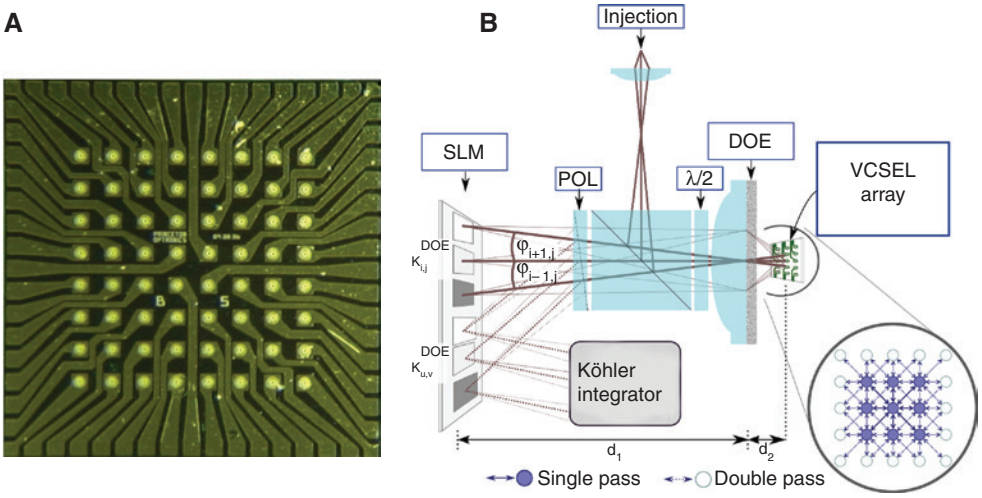

Figure 4: (A) Array in single-mode VCSEL laser diodes, Princeton Optronics. Implementing such an array in a diffractive resonator design creates coupling between individual lasers (panel B) [24]. Such a network can then be injected via a single external laser; readout weights can be implemented via a SLM.

diffractive orders of one laser will overlap with the non-diffracted image of its neighbors. As shown in Figure 4B, coupling between individual elements can be established by placing a reflector in the image plane of the setup. For the used DOE, a 5×5 coupling matrix was established. In the experiment shown in Figure 4B, a spatial light modulator (SLM) was located in the imaging plane, allowing for practical control of the networks coupling weights. Multiple semiconductor inherent properties result in a highly nonlinear response of the semiconductor lasers. In general, VCSELs are highly energy efficient and allow for modulation bandwidths reaching tens of gigahertz. Once coupled, the system showed nonlinear dynamics induced by the network coupling [24]. A Köhler integrator follows the photonic VCSEL array reservoir. In a Köhler integrator, microlens arrays decompose a complex optical field into multiple single-mode fields, which consecutively can be focused by a lens to a single, uniform focal spot. Modifying the input field via a SLM and placing a detector at the Köhler integrator focal spot therefore realizes the integrated and weighted network state needed for RC.

Inherent to the manufacturing process, lasers located on the array are subject to parameter variations across the laser chip. The introduced diffractive network therefore is subject to diversity, and an inherently complex network was established. The reservoir dynamical timescale is given by the external coupling delay time, which in Brunner and Fischer [24] was ~2 ns. The global network state is therefore updated at a rate of 0.5 GHz. Beside the network coupling, diffractive imaging also allows for optically modulating multiple lasers in parallel. Brunner and Fischer [24] locked eight semiconductor lasers of the introduced array to an external injection laser which was intensity modulated with a sine wave of 33 MHz. The responses of the eight lasers were recorded, and, due to the diverse nonlinear responses of the eight lasers, they were able to synthesize several different nonlinear transformations of the input data using a linear combination in the readout layer. The experimental results are shown in Figure 5, in which the computed functions are compared to the corresponding target ones. Nonlinear transformations were created offline. The readout weights for each computed function are shown in Figure 5 (right).

The small number of lasers coupled in Brunner and Fischer [24] was mainly limited by optical aberrations of the imaging setup. For smaller DOE diffraction angles Θ, the scheme should be scalable to networks consisting of hundreds of nodes in an area smaller than 1 mm^2. As such, the scheme would allow for all-optical RC with networks of competitive sizes. While the scheme is flexible, bulk optics are a significant limitation for a commercial application. For a technologically relevant implementation the miniaturization of the introduced approach would first have to be demonstrated. As this system is based on injection locking, careful attention needs to be paid to the wavelength uniformity of the laser array.

3.3 Excitable photonic devices for RC

A different approach to the photonic implementation of RC is by exploiting the excitability properties of specific photonic devices. The spiking behavior of excitable photonic devices, which can be implemented by

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

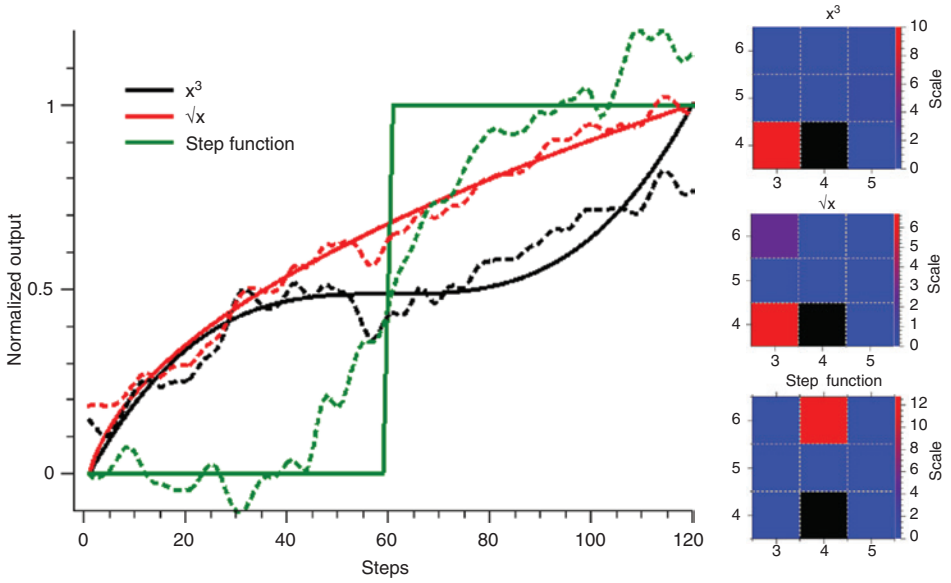

Figure 5: Several nonlinear transformations were synthesized from the network of eight injection locked lasers [24]. The *y* axis corresponds to rescaled values of optical intensities, the *x* axis to sample points of the input signal (sampling time 100 ps). The right panels show the numerically implemented readout weights of each individual laser.

semiconductor technology, resembles the properties of biological neurons [39–44]. Networks of such excitable, nanophotonic devices would therefore correspond to a neural network implementation very close to their biological inspiration. Also, the typical spike energies can be in the fJ–pJ range, leading to a very advantageous power consumption for RC based on excitable photonic devices.

When biased adequately close to a stability threshold, a laser with a saturable absorber becomes an excitable system [39, 41, 43, 44]. In their dynamical behavior, these devices approximate the integrate and fire behavior of biological neurons with remarkable quality. With current semiconductor technology, highly energy-efficient lasers can be implemented thanks to high-quality Bragg mirrors and large differential optical amplification. In addition, saturable absorbers can be realized and incorporated in such lasers.

Shastri et al. [44] numerically simulated a simple optical network consisting of two excitable lasers. As saturable absorber they implemented a single layer of graphene. In 2014, the same group evaluated larger networks based on electro-optically excited semiconductor lasers [45]. Wavelength division multiplexing technology was used for addressing single lasers, which results in a limitation of ~60 nodes per chip.

In 2011, Barbay et al. [39] demonstrated excitable neuron-like pulsing behavior of a monolithic semiconductor micro-pillar laser. The system is highly simplistic, only adding an additional quantum well in order to achieve excitability. In 2014, Selmi et al. [46] demonstrated that a similar system experiences a refractory period. Upon initial excitation by an optical pulse, the system remains nonexcitable during its refractory period. This behavior is also present in biological neurons.

Optical excitability has also been observed in microring and disk lasers [40, 42]. The dynamical behavior of these integrated laser devices is not related to a saturable absorber but rather to internal symmetry breaking properties.

So far, no hardware implementations of such excitable systems have been exploited for the implementation of ANNs or RC. However, the latest developments demonstrate the possibilities of photonics for the realization of spiking ANNs of unparalleled speed. Advances in this field might circumvent the von Neumann approach to calculating neuron responses in the TrueNorth and SpiNNaker architectures.

4 Delay-based reservoir computing

The concept of delay line-based RC, using only a single nonlinear node with delayed feedback, was introduced some years ago by Appeltant et al. [20] and Pacquot et al. [47] as a means of minimizing the expected hardware complexity in photonic systems. The first working prototype was developed in electronics in 2011 by Appeltant et al.

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

[20], and performant optical systems followed quickly after that [21, 22].

Nonlinear systems with delayed feedback and/or delayed coupling, often simply put as *delay systems*, are a class of dynamical systems that have attracted considerable attention, because they arise in a variety of real-life systems [48]. In optics, delayed coupling often coming from unwanted reflections was originally considered a nuisance leading to oscillations and chaos. By now, intentional delayed optical feedback or coupling has led to many applications [49], such as secure chaos communications [50], high-speed random bit generation [51], and now also RC.

Due to the circular symmetry of a single delay line, delay systems have been interpreted as an implementation of a discrete reservoir with a circular connection topology [20]. However, a key property of neural networks is the notion of a discrete node that exhibits a nonlinear relation between an output and multiple inputs. The network can take on a number of, if not arbitrary, directed graph topologies. This was the case in all of the optical RC implementations covered in Section 3. The spatially distributed reservoir computers covered in this review have many network degrees of freedom, even though they are fixed artificially. In contrast, delay-based photonic reservoirs are fixed intrinsically: they take the form of a time-delayed dynamical system with a single nonlinear state variable. Thus, from a network perspective, there is only one (hardware) node. Mathematically, delay systems are described by delay differential equations (DDE) that differ fundamentally from ordinary differential equations as the time-dependent solution of a DDE is not uniquely determined by its state at a given moment. For a DDE, the continuous solution on an interval of one delay time needs to be provided in order to define the initial conditions correctly. As such, a low-dimensional system with delayed feedback offers the high-dimensional phase space which is the basis for RC. Hence, the delay-based approach allows for a far simpler system structure, even for very large reservoir sizes. The tremendous advantage of delay-based RC lies in the minimal hardware requirements as compared to the more hardware-intensive systems from Section 3. It allows even for the use of hardware that is more traditionally associated with optical communications.

In essence, the idea of delay line RC constitutes an exchange between space and time: what has been done spatially with many nodes as in Sections 2 and 3 is now done in a single node that is multiplexed in time. There is a price to pay for this hardware simplification: compared to an N-node standard spatially distributed reservoir, the dynamical behavior in the system has to run at an N times higher speed. In Figure 6, we show a diagram of a delay

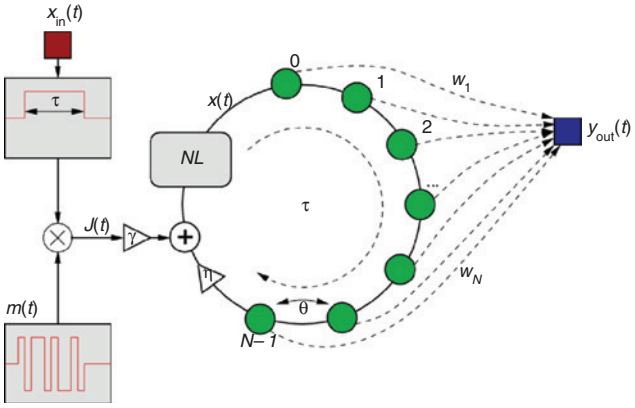

Figure 6: Structure of a delay line based reservoir computer. A one-dimensional input signal (in red) is first preprocessed using the masking function $m(t)$. Virtual nodes are defined along the delay line and form the reservoir (in green). The output layer (in blue) is unaltered from the standard RC structure.

line-based RC. The input signal $x_{in}(t)$ undergoes a sample and hold operation every τ, which is also exactly the duration of the delay in the feedback loop. One could also say that a new input sample is applied every τ. Then, this input is multiplied with a masking signal $m(t)$. This mask repeats every τ, and within one period, it is a piecewise constant function with a fixed sequence of N values. This sequence is chosen from a certain set $\{m_1, m_2, \ldots\}$, and these are spaced

$$\theta = \frac{\tau}{N} \tag{4}$$

in time. The θ-spaced points in the delay line are called *virtual nodes* or *virtual neurons*. Therefore, θ is also called the virtual node separation or distance. The mask together with the inertia of the nonlinear node controls the connectivity between the virtual nodes, and a virtual interconnection structure is created. The masked signal $J(t)$ is then applied to a nonlinear time-dependent node, which also receives input from the delayed feedback. The mask is introduced to diversify the response of the virtual nodes to the input signal and plays a similar role as the input weights in Figure 1. The optimal choice for the amount of different mask values and their exact values is task and system dependent. No research has been done on the distribution types used to draw the mask values. Limited research has been done for two-valued mask functions, suggesting a nonrandom mask construction procedure based on maximum length sequences [52]. The node distance θ has to be sufficiently short to keep the nonlinear node in a transient state throughout the entire delay line. Typically, a number of 20% of an internal timescale is

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

quoted [20, 32]. However, there is no reason to assume that this could not be task and system bias dependent. If θ is too short, the nonlinear node will not be able to follow the high-bandwidth input signal, and the response signal will be too small to measure. If θ is too long, the virtual interconnection structure between the virtual nodes is lost. However, if one would slightly misalign the period of the mask and the input sampling period to the length of the delay line, a slightly different virtual network structure would be recovered [47]. It is clear that the operation speed of a delay-based RC is limited by the delay length τ as input date samples are fed in at this period. The delay itself is defined by the numbers of virtual nodes N that are necessary to compute a specific task and the node distance θ.

As shown in Figure 6, the masked input $J(t)$ is scaled by an input scaling factor γ and the feedback by a feedback strength η. This is to bias the nonlinear node in the optimal dynamical regime. Optimal values for the input scaling γ and η depend on the task at hand, as well as the specific dynamical behavior of the nonlinear node. Finding the optimal point for these parameters is a nonlinear problem which can be approached by, for example, a gradient descent or by simple scanning of the parameter space. After each τ interval a new output value $y_{out}(n)$ is obtained. It is calculated as a linear combination of the θ-spaced taps on the delay line, which comprise the virtual neurons. This output value is kept constant over an entire delay time τ. Each virtual node is a measuring point or tap in the delay line. However, these taps do not have to be physically realized. Since the x signal revolves unaltered in the delay line anyway, a single measuring point suffices. For training, the reservoir state $x(t)$ is sampled per time step θ. The samples are then reorganized in a state matrix S as before having width N and length equal to T, the number of input samples. The ith column of S represents the time series of the ith virtual node. Its precise content is determined by the input, the masking, and the dynamical behavior of the nonlinear node. The state matrix S is built using one input sample at a time. The corresponding node states are recorded when one entire input sample – stretched over one delay – has passed the nonlinear node, i.e. when the entire delay line is filled with responses to the same data sample. From there on, the training proceeds exactly as described in Section 2.3.

4.1 Optoelectronic delay-based reservoir computing

The first optical hardware implementations of RC were independently developed by Larger et al. [22] and Paquot et al. [21]. Both implementations were based on the optoelectronic implementation of an Ikeda-like ring optical cavity [53, 54]. The optoelectronic implementation of RC is schematically depicted in Figure 7. The optical part of the setup, which is fiber based, includes a laser source, a Mach-Zehnder modulator, and a long optical fiber spool. The Mach-Zehnder modulator provides the nonlinear modulation transfer function ($\sin^2$ – function), while the long optical fiber provides the delayed feedback loop. The electronic part of the setup is typically composed of a photodiode, a filter, and an amplifier. The injection of the external input and the extraction of the system output are both done in the electronic part.

The optoelectronic system has been widely employed for RC. A number of classification, prediction, and system modelling tasks have been performed with state-of-the-art results. To name a few, excellent performance has been obtained for speech recognition [21, 22, 55], chaotic time series prediction [22, 56, 57], nonlinear channel equalization [21, 57–59], and radar signal forecasting [58, 59]. A summary of the results obtained for each task is given in Table 1. The operating speed of optoelectronic RC implementations is in the megahertz range, although this kind of setup has the potential to operate at gigahertz speeds [60].

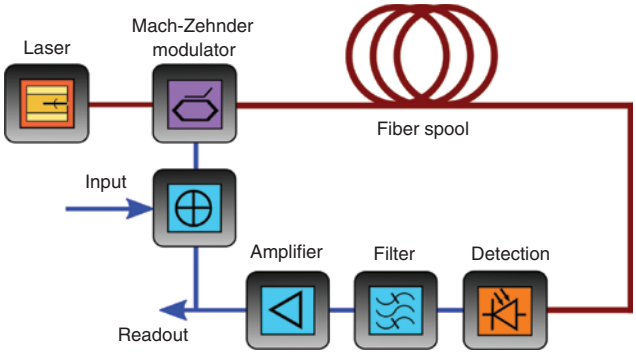

Figure 7: Scheme of the optoelectronic reservoir computer. The optical (electronic) path is depicted in red (blue) color.

Table 1: Summary of the best reported results obtained by the optoelectronic reservoir computer for computationally hard tasks.

Task	Result
Isolated spoken digit recognition	0.04% (WER) [22]
Santa Fe time series prediction	0.02 (NMSE) [56]
Nonlinear channel equalization (SNR 20 dB)	10^{-3} (SER) [21]
Nonlinear channel equalization (SNR 28 dB)	10^{-4} (SER) [21]
Radar signal forecasting (LSS, 1 day)	10^{-3} (NMSE) [59]
Radar signal forecasting (LSS, 5 days)	10^{-2} (NMSE) [59]

WER, word error rate; NMSE, normalized mean square error; SER, symbol error rate; SNR, signal-to-noise ratio; LSS, low sea level.

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

An important factor contributing to the extensive use of the optoelectronic version of RC is that it can be modelled with high accuracy. The systems presented in Larger et al. [22] and Paquot et al. [21] can be respectively modelled with similar, but not identical, scalar equations. First, we present the first-order DDE that describes the temporal evolution of the system introduced in Larger et al. [22]:

$$\varepsilon\dot{x}(s)+x(s)=\beta\sin^2[\eta x(s-1)+\gamma J(s-1)+\Phi], \quad (5)$$

where β is the nonlinearity gain, Φ denotes the offset phase of the Mach-Zehnder modulator, γ is the relative weight of the input information J compared to the system signal x, and η corresponds to the feedback scaling. Parameter $\varepsilon=T_R/\tau$ is the oscillator response time T_R normalized to the delay time τ, and $s=t/\tau$ is the normalized time. Equation (5) describes a system with inertia. In contrast to Eq. (5), the system introduced in Paquot et al. [21] can be described with a simple model that considers the nonlinear transformation to be instantaneous.

$$x(s)=\sin[\eta x(s-1)+\gamma J(s)+\Phi], \quad (6)$$

Equation (6) is a valid approximation of the optoelectronic system when the frequency responses of all hardware components are significantly faster than the input injection rate, which is of the order $1/\tau$. Both experimental systems provide direct access to all variables and parameters [21, 22]. The system modelled by Eq. (6) can be used for RC purposes as long as the delay time and the length of the mask are slightly misaligned [21]. This ensures that the virtual nodes are interconnected as a RNN. A detailed comparison of the technical implications derived from the differences between Eqs. (5) and (6) is beyond the scope of this review. However, it is worth noting that the performance of both approaches for RC purposes is comparable. By taking either Eq. (5) or (6) as a theoretical model, it is possible to evaluate the computational properties of the optoelectronic system. As a result, the system depicted in Figure 7 and its variants have served as a paradigmatic example for several experimental and numerical studies of photonic RC [61, 62].

Most hardware implementations of optoelectronic RC focus on the practical demonstration of the reservoir layer. The input and output layers are emulated offline on a standard computer. There are, however, first works aiming at the complete implementation of the three layers of RC on analogue hardware. In this way, a proof of concept for standalone optoelectronic reservoir computers has been demonstrated [58]. For the analogue input layer, a mask that combines two sinusoidals with different frequencies suffices. The design of the analogue output layer is more involved. Figure 8 shows the components required

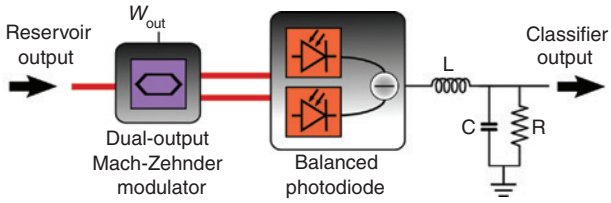

Figure 8: Scheme of the analogue readout layer in Duport et al. [58].

to implement the linear readout $y_{out}=\sum W^i_{out}x_i$, with $i=[1, \ldots, N]$. The optical signal from the reservoir is modulated with a dual-output Mach-Zehnder modulator, with the readout weights W_{out} computed during the training phase. Since the optical signal is strictly positive, a balanced photodiode is placed after the dual-output Mach-Zehnder modulator, allowing for positive and negative response values. The output signal of the balanced photodiode is filtered by an RLC filter that is carrying out the analogue summation of the weighted reservoir values. The resulting signal of the analogue output layer is the corresponding output of the reservoir computer y_{out}.

The issue of how to train the readout weights of hardware RC implementations is a relevant aspect that has been seldom considered. Typically, the training of the readout weights is performed off-line on a standard computer after the responses of the reservoir to the input examples have been recorded. An interesting approach to train on-line the readout weights is to use dedicated hardware such as field-programmable gate arrays (FPGAs) [63]. Although FPGAs are digital electronic devices, they are prepared to interact with analogue signals via on-board analogue-to-digital and digital-to-analogue converters. Interestingly, the increasing clock speeds of commercial FPGAs can easily reach hundreds of megahertz. The on-line training can then be realized by employing gradient descent or genetic algorithms. On-line learning capabilities offer the possibility to adapt to changing environments [63].

Going beyond RC, we would like to discuss two other machine learning approaches that have been implemented on the optoelectronic hardware depicted in Figure 7. To start with, the optoelectronic system without delayed feedback loop has served as a hardware platform to implement the extreme learning machine (ELM) concept [57]. ELMs are feedforward neural networks with a single hidden layer, i.e. a reservoir without internal connectivity, in which the input layer is randomly mapped to the hidden layer [64]. In Ortín et al. [57], it has been shown that the optoelectronic implementation of ELM yields comparable results to RC as long as past inputs are explicitly included in the input layer. A more powerful approach is that of implementing general machine

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

learning models on a hardware device [65]. Full optimization of the complete system, including input mask, system parameters, and output weights, is possible thanks to the back-propagation through time (BPTT) algorithm. It has been experimentally and numerically shown that the performance of the optoelectronic system can be greatly enhanced when BPTT is used as a training procedure [65, 66]. This paves the way to implement advanced machine learning concepts on high-speed physical hardware devices.

4.2 All optical delay-based reservoir computing

In this review, the classification between optoelectronic and all-optical implementations of delay-based RC is done on the basis of the nature of the input and the reservoir. We discuss in this section those implementations with an optical input to an all-optical reservoir. Several practical implementations, either based on, e.g. semiconductor lasers [67], SOAs [23], or passive optical cavities [68] fall in this category. The number of virtual reservoir nodes in these hardware implementations is typically in the range 50–400.

The first two experimental realizations of all-optical RC were based on active devices. Duport et al. [23] employed the nonlinear response of a SOA placed in a ring optical cavity, while Brunner et al. [67] employed the nonlinear response of a semiconductor laser subject to feedback. In both cases, the external input was injected as a modulated optical field. The output layer was implemented off-line after detection. These experimental realizations demonstrate the potential of the RC paradigm in photonics for computationally hard tasks. In particular, the photonic reservoir based on a semiconductor laser with feedback has shown unconventional information processing capabilities at Gbyte/s rates [67], the fastest reservoir computer up to date. This system is schematically depicted in Figure 9, where a semiconductor laser is subject to the injection of a modulated laser (external input) and an optical feedback loop with a delay of 77.6 ns forming the reservoir. Since the injection data rate in Brunner et al. [67] was 5 GSamples/s, the node distance θ was 200 ps, and the total number of nodes was 388. This reservoir computer was used to classify spoken digits and to forecast chaotic time series with high accuracy. A similar system was also employed to demonstrate high-speed optical vector and matrix operations [69]. Numerical simulations of this kind of system suggest that the information processing capabilities can still be improved

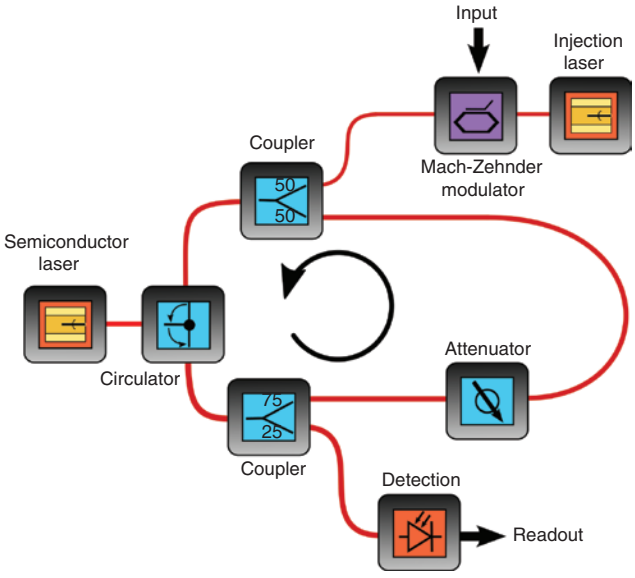

Figure 9: Scheme of the all-optical reservoir computer based on a semiconductor laser subject to delayed optical feedback. The experimental setup comprises the laser diode, a tunable laser source to optically inject the information, a Mach-Zehnder modulator, an optical attenuator, a circulator, couplers, and a fast photo diode (PD) for signal detection.

by either increasing the injection strength [70] or by changing the input mask [71].

All-optical RC based on delay systems has the potential to be integrated in a photonic chip. Nguimdo et al. [72] has shown numerically that the necessary optical bias injection can increase the optical modulation bandwidth of semiconductor lasers, allowing for shorter virtual node distances and hence shorter delay times on the order of a few nanoseconds rather than the 70 ns employed by Brunner et al. [67]. Nguimdo et al. [73] has suggested that an on-chip semiconductor ring laser subject to optical feedback can be used to simultaneously solve two different tasks, e.g. a classification task and a time series prediction task. The phase sensitivity found on semiconductor lasers with short optical external cavities, typical of integrated systems, can be avoided if the readout layer is slightly modified [74]. This proves that RC based on delay systems can be transferred to photonic integrated circuits.

An important step towards the development of high-speed, low-consumption, analogue, photonic computers is the use of passive devices. Here, we discuss two implementations of all-optical RC that use passive optical devices. First, Dejonckheere et al. [75] demonstrated a photonic RC system based on a semiconductor saturable absorber mirror (SESAM) that is placed in a ring-like optical cavity. A schematic view of this experimental

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

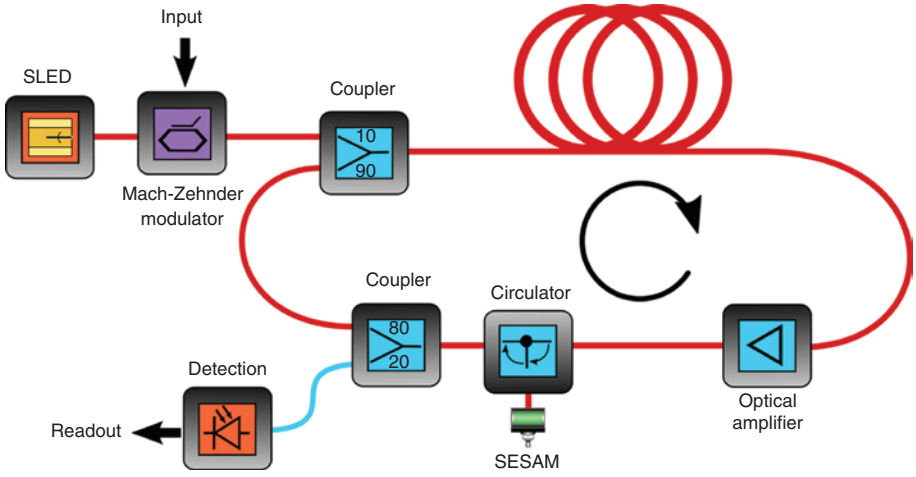

Figure 10: Scheme of the all-optical reservoir computer based on the saturation of absorption. The input optical signal is injected into the ring cavity by means of a fiber coupler. The cavity itself consists of a fiber spool used as a delay line, an optical amplifier, a circulator, and a SESAM. A fiber coupler is used to send 20% of the cavity intensity to the readout photodiode.

setup can be seen in Figure 10. The external input modulates a superluminescent light-emitting diode, whose light is injected into the delay-based photonic reservoir. This system yields performances similar to other photonic reservoir computers [75], with the SESAM being a nonlinear passive element. Finally, we discuss a photonic RC based on a coherently driven passive cavity. Vinckier et al. [68] demonstrated that a simple linear fiber cavity can be used as a reservoir computer as long as the output layer is nonlinear. The optical field that propagates in the cavity is detected with a photodiode, which performs a quadratic transformation of the impinging optical field. This nonlinear transformation is sufficient to perform computationally hard tasks such as nonlinear channel equalization or spoken digits recognition [68]. As such, this is a simple all-optical reservoir computer with high power efficiency.

5 Outlook

The combination of nanophotonics and the RC paradigm has the potential to disrupt photonic information processing in the coming years. The two main advantages offered by photonic hardware implementations of RC are the low power consumption and the high processing speeds compared to other traditional approaches. Combined with optical sensors, this technology could transform the way photonic information is being processed. We foresee a clear trend towards the integration and miniaturization of the hardware implementations, which so far have been realized in systems with a relatively large footprint with a few notable exceptions [38]. Here, one can envision the true synergetic potential when combining nanophotonics with novel neuromorphic computing approaches.

Photonic RC systems bring intelligence to optical systems in a native platform. The range of applications that could benefit from such devices is extremely broad, from optical header recognition and signal recovery to fast control loops.

Acknowledgments: We would like to thank the promoters of the Photonic Reservoir Computing community in Europe for fruitful discussions: Ingo Fischer, Claudio Mirasso, Laurent Larger, Luis Pesquera, Gordon Pipa, Juergen Kurths, Jan Danckaert, Serge Massar, Joni Dambre, Benjamin Schrauwen, and Peter Bienstman. We would also like to thank Lennert Appeltant, Silvia Ortín, and Lars Keuninckx for helpful comments and Guy Verschaffelt for proofreading the manuscript. M.C.S. was supported by the Conselleria d'Innovació, Recerca i Turisme del Govern de les Illes Balears and the European Social Fund. G.V. acknowledges support from FWO, and the Research Council of the VUB. This work benefited from the support of the Belgian Science Policy Office under Grant No IAP-7/35 'photonics@be'.

References

[1] Ivakhnenko AG. Polynomial theory of complex systems. IEEE T Syst Man Cyb 1971;1:364–78.
[2] Cireşan D, Meier U, Maria Gambardella L, Schmidhuber J. Deep, big, simple neural nets for handwritten digit recognition. Neural Comput 2010;22:3207–20.

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

[3] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012;25:1106–14.
[4] Silver D, Huang A, Maddison CJ, et al. Mastering the game of go with deep neural networks and tree search. Nature 2016;529:484–9.
[5] Merolla PA, Arthur J, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 2014;345:668–72.
[6] Furber S, Temple S. Neural systems engineering. J R Soc Interf 2006;4:193–206.
[7] Rast A, Galluppi F, Davies S, et al. Concurrent heterogeneous neural model simulation on real-time neuromimetic hardware. Neural Networks 2011;24:961–78.
[8] Hopfield J, Tank D. "Neural" computation of decisions in optimization problems. Biol Cybern 1985;52:141–52.
[9] Denz C. Optical neural networks. In: Tschudi T., ed. Wiesbaden, Springer Vieweg, 1998.
[10] Psaltis D, Brady D, Gu X-G, Lin S. Holography in artificial neural networks. Nature 1990;343:325.
[11] Jutamulia S, Yu F. Overview of hybrid optical neural networks. Opt Laser Technol 1996;28:59–72.
[12] Fernando C, Sojakka S. Pattern recognition in a bucket. In: Banzhaf W., Ziegler J., Christaller T., Dittrich P., Kim J.T., eds. Advances in Artificial Life. ECAL 2003. Lecture Notes in Computer Science, vol 2801. Berlin, Heidelberg, Springer, 2003.
[13] Jaeger H, Haas H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 2004;304:78–80.
[14] Maass W, Natschläger T, Markram H. Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput 2002;14:2531–6.
[15] Steil J. Backpropagation-decorrelation: online recurrent learning with O(N) complexity. IJCNN 2004;1:843–8.
[16] Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 2009;3:127–49.
[17] Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D. An experimental unification of reservoir computing methods. Neural Networks 2007;20:391–403.
[18] Jaeger H. Short term memory in echo state networks. German National Research Center for Information Technology, Technical Report GMD Report 152, 2001.
[19] Rodan A, Tino P. Minimum complexity echo state network. IEEE Trans Neural Netw 2011;22:131–44.
[20] Appeltant L, Soriano MC, Van der Sande G, et al. Information processing using a single dynamical node as complex system. Nat Commun 2011;2:468.
[21] Paquot Y, Duport F, Smerieri A, et al. Optoelectronic reservoir computing. Sci Rep 2012;2:287.
[22] Larger L, Soriano MC, Brunner D, et al. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. Opt Express 2012;20:3241–9.
[23] Duport F, Schneider B, Smerieri A, Haelterman M, Massar S. All-optical reservoir computing. Opt Express 2012;20: 22783–95.
[24] Brunner D, Fischer I. Reconfigurable semiconductor laser networks based on diffractive coupling. Opt lett 2015;40:3854.
[25] Dambre J, Verstraeten D, Schrauwen B, Massar S. Information processing capacity of dynamical systems. Sci Rep 2012;2:514.
[26] Sylvestre J. "Mechanical computations", presentation at "BEYOND! von Neumann" workshop, Berlin May 18–20 (2016).
[27] Caluwaerts K, D'Haene M, Verstraeten D, Schrauwen B. Locomotion without a brain: physical reservoir computing in tensegrity structures. Artif Life 2012;19:35–66.
[28] Hauser H, Ijspeert AJ, Füchslin RM, Pfeifer R, Maass W. Towards a theoretical foundation for morphological computation with compliant bodies. Biol Cybern 2011;105:355–70.
[29] Nakajima K, Li T, Hauser H, Pfeifer R. Exploiting short-term memory in soft body dynamics as a computational resource. J R Soc Interf 2014;11:20140437.
[30] Uchida A, Yoshimura K, Davis P, Yoshimori S, Roy R. Local conditional Lyapunov exponent characterization of consistency of dynamical response of the driven Lorenz system. Phys Rev E Stat Nonlin Soft Matter Phys 2008;78:036203.
[31] Oliver N. Consistency properties of a chaotic semiconductor laser driven by optical feedback. Phys Rev Lett 2015;114:123902.
[32] Soriano MC, Ortín S, Keuninckx L, et al. Delay-based reservoir computing: noise effects in a combined analog and digital implementation. IEEE Trans Neural Netw Learn Syst 2015;26:388–93.
[33] Lukoševičius M, Jaeger H, Schrauwen B. Reservoir computing trends. KI Künstliche Intelligenz 2012;26:365–71.
[34] Simply silicon. Nat Photon 2010;4:491.
[35] Vandoorne K, Dierckx W, Schrauwen B, et al. Toward optical signal processing using photonic reservoir computing. Opt Express 2008;16:11182.
[36] Vandoorne K, Dambre J, Verstraeten D, Schrauwen B, Bienstman P. Parallel reservoir computing using optical amplifiers. IEEE Trans Neural Netw 2011;22:1469–81.
[37] Salehi MR, Dehyadegari L. Optical signal processing using photonic reservoir computing. J Mod Opt 2014;61:144–5.
[38] Vandoorne K, Mechet P, Van Vaerenbergh T, et al. Experimental demonstration of reservoir computing on a silicon photonics chip. Nat Commun 2014;5:1–6.
[39] Barbay S, Kuszelewicz R, Yacomotti AM. Excitability in a semiconductor laser with saturable absorber. Opt Lett 2011;36:4476–8.
[40] Coomans W, Gelens L, Beri S, Danckaert J, Van der Sande G. Solitary and coupled semiconductor ring lasers as optical spiking neurons. Phys Rev E 2011;84:036209.
[41] Hurtado A, Schires K, Henning ID, Adams MJ. Investigation of vertical cavity surface emitting laser dynamics for neuromorphic photonic systems. Appl Phys Lett 2012;100. Paper number: 103703.
[42] Vaerenbergh TV, Fiers M, Mechet P, et al. Cascadable excitability in microrings. Opt Express 2012;20:20292–308.
[43] Nahmias MA, Tait AN, Shastri BJ, de Lima TF, Prucnal PR. Excitable laser processing network node in hybrid silicon: analysis and simulation. Opt Express 2015;23:26800–13.
[44] Shastri BJ, Nahmias MA, Tait AN, Rodriguez AW, Wu B, Prucnal PR. Spike processing with a graphene excitable laser. Sci Rep 2016;6:19126.
[45] Tait AN, Member S, Nahmias MA, Shastri BJ, Prucnal PR. Broadcast and weight: an integrated network for scalable photonic spike processing. J Lightwave Technol 2014;32:3427–39.
[46] Selmi F, Braive R, Beaudoin G, Sagnes I, Kuszelewicz R, Barbay S. Relative refractory period in an excitable semiconductor laser. Phys Rev Lett 2014;112:183902.
[47] Paquot Y, Dambre J, Schrauwen B, Haelterman M, Massar S. Reservoir computing: a photonic neural network for

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM

information processing," in Proc. SPIE 7728, Nonlinear Optics and Applications IV 2010;7728:77280B–12.
[48] Erneux T. Applied delayed differential equations. New York, Springer Science Business Media, 2009.
[49] Soriano MC, Garca-Ojalvo J, Mirasso CR, Fischer I. Complex photonics: dynamics and applications of delay-coupled semiconductors lasers. Rev Mod Phys 2013;85:421–70.
[50] Argyris A, Syvridis D, Larger L, et al. Chaos-based communications at high bit rates using commercial fibre-optic links. Nature 2005;438:343–6.
[51] Uchida A, Amano K, Inoue M, et al. Fast physical random bit generation with chaotic semiconductor lasers. Nat Photon 2008;2:728–32.
[52] Appeltant L, Van der Sande G, Danckaert J, Fischer I. Constructing optimized binary masks for reservoir computing with delay systems. Sci Rep 2014;4:3629.
[53] Ikeda K, Daido H, Akimoto O. Optical turbulence: chaotic behavior of transmitted light from a ring cavity. Phys Rev Lett 1980;45:709.
[54] Goedgebuer J-P, Larger L, Porte H, Delorme F. Chaos in wavelength with a feedback tunable laser diode. Phys Rev E 1998;57:2795.
[55] Martinenghi R, Rybalko S, Jacquot M, Chembo YK, Larger L. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. Phys Rev Lett 2012;108:244101.
[56] Soriano MC, Ortín S, Brunner D, et al. Optoelectronic reservoir computing: tackling noise-induced performance degradation. Opt Express 2013;21:12–20.
[57] Ortín S, Soriano MC, Pesquera L, et al. A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. Sci Rep 2015;5:14945.
[58] Duport F, Smerieri A, Akrout A, Haelterman M, Massar S. Fully analogue photonic reservoir computer. Sci Rep 2016;6:22381.
[59] Duport F, Smerieri A, Akrout A, Haelterman M, Massar S. Virtualization of a photonic reservoir computer. J Lightwave Technol 2016;34:2085–91.
[60] Lavrov R, Jacquot M, Larger L. Nonlocal nonlinear electro-optic phase dynamics demonstrating 10 Gb/s chaos communications. IEEE J Quantum Elect 2010;46:1430–5.
[61] Woods D, Naughton TJ. Optical computing: photonic neural networks. Nat Phys 2012;8:257–9.
[62] Soriano MC, Brunner D, Escalona-Morán M, Mirasso CR, Fischer I. Minimal approach to neuro-inspired information processing. Front Comput Neurosci 2015;9:68.
[63] Antonik P, Duport F, Smerieri A, Hermans M, Haelterman M, Massar S. Online training of an opto-electronic reservoir computer. In: Neural Information Processing. 1em plus 0.5em minus 0.4em Springer, 2015, pp. 233–240.
[64] Huang G-B, Wang DH, Lan Y. Extreme learning machines: a survey. Int J Mach Learn Cyber 2011;2:107–22.
[65] Hermans M, Soriano MC, Dambre J, Bienstman P, Fischer I. Photonic delay systems as machine learning implementations. J Mach Learn Res 2015;16:2081–97.
[66] Hermans M, Dambre J, Bienstman P. Optoelectronic systems trained with backpropagation through time. IEEE Trans Neural Netw Learn Syst 2015;26:1545–50.
[67] Brunner D, Soriano MC, Mirasso CR, Fischer I. Parallel photonic information processing at gigabyte per second data rates using transient states. Nat Commun 2013;4:1364.
[68] Vinckier Q, Duport F, Smerieri A, et al. High-performance photonic reservoir computer based on a coherently driven passive cavity. Optica 2015;2:438–46.
[69] Brunner D, Soriano MC, Fischer I. High-speed optical vector and matrix operations using a semiconductor laser. IEEE Photonics Tech Lett 2013;25:1680–3.
[70] Hicke K, Escalona-Morán MA, Brunner D, Soriano MC, Fischer I, Mirasso CR. Information processing using transient dynamics of semiconductor lasers subject to delayed feedback. IEEE J Sel Top Quant 2013;19:1501610.
[71] Nakayama J, Kanno K, Uchida A. Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal. Opt Express 2016;24:8679–92.
[72] Nguimdo RM, Verschaffelt G, Danckaert J, Van der Sande G. Fast photonic information processing using semiconductor lasers with delayed optical feedback: role of phase dynamics. Opt Express 2014;22:8672–86.
[73] Nguimdo RM, Verschaffelt G, Danckaert J, Van der Sande G. Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback. IEEE Trans Neural Netw Learn Syst 2015;26:3301–7.
[74] Nguimdo RM, Verschaffelt G, Danckaert J, Van der Sande G. Reducing the phase sensitivity of laser-based optical reservoir computing systems. Opt Express 2016;24:1238–52.
[75] Dejonckheere A, Duport F, Smerieri A, et al. All-optical reservoir computer based on saturation of absorption. Opt Express 2014;22:10868–81.

Brought to you by | Beijing Post and Telecommunication Unive
Authenticated
Download Date | 6/25/18 4:56 PM