Neural Networks 115 (2019) 100–123

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Review

# Recent advances in physical reservoir computing: A review

Gouhei Tanaka [a,b,*], Toshiyuki Yamane [c], Jean Benoit Héroux [c], Ryosho Nakane [a,b], Naoki Kanazawa [c], Seiji Takeda [c], Hidetoshi Numata [c], Daiju Nakano [c], Akira Hirose [a,b]

[a] *Institute for Innovation in International Engineering Education, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan*
[b] *Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan*
[c] *IBM Research – Tokyo, Kanagawa 212-0032, Japan*

## ARTICLE INFO

*Article history:*
Received 15 August 2018
Received in revised form 24 February 2019
Accepted 7 March 2019
Available online 20 March 2019

*Keywords:*
Neural networks
Machine learning
Reservoir computing
Nonlinear dynamical systems
Neuromorphic device

## ABSTRACT

Reservoir computing is a computational framework suited for temporal/sequential data processing. It is derived from several recurrent neural network models, including echo state networks and liquid state machines. A reservoir computing system consists of a reservoir for mapping inputs into a high-dimensional space and a readout for pattern analysis from the high-dimensional states in the reservoir. The reservoir is fixed and only the readout is trained with a simple method such as linear regression and classification. Thus, the major advantage of reservoir computing compared to other recurrent neural networks is fast learning, resulting in low training cost. Another advantage is that the reservoir without adaptive updating is amenable to hardware implementation using a variety of physical systems, substrates, and devices. In fact, such physical reservoir computing has attracted increasing attention in diverse fields of research. The purpose of this review is to provide an overview of recent advances in physical reservoir computing by classifying them according to the type of the reservoir. We discuss the current issues and perspectives related to physical reservoir computing, in order to further expand its practical applications and develop next-generation machine learning systems.

© 2019 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

**Contents**

1. Introduction ........ 101
2. Reservoir computing (RC) ........ 102
    2.1. Basic framework ........ 102
    2.2. Recent trends ........ 103
3. Dynamical systems models for RC ........ 105
    3.1. Delayed dynamical systems ........ 105
    3.2. Cellular automata ........ 105
    3.3. Coupled oscillators ........ 106
4. Electronic RC ........ 106
    4.1. Analog circuits ........ 107
    4.2. FPGAs ........ 107
    4.3. VLSIs ........ 108
    4.4. Memristive RC ........ 109
        4.4.1. Neuromemristive circuits ........ 109
        4.4.2. Memristive systems and devices ........ 109
5. Photonic RC ........ 110
    5.1. Optical node arrays ........ 110
    5.2. Time-delay systems ........ 111
        5.2.1. Opto-electronic and optical feedback with gain ........ 111
        5.2.2. Optical feedback in a laser cavity ........ 112

---

[*] Corresponding author at: Institute for Innovation in International Engineering Education, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan.
*E-mail address:* gouhei@sat.t.u-tokyo.ac.jp (G. Tanaka).

https://doi.org/10.1016/j.neunet.2019.03.005
0893-6080/© 2019 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

6. Spintronic RC ..... 113
7. Mechanical RC ..... 114
8. Biological RC ..... 115
  8.1. Brain regions ..... 115
  8.2. in-vitro cultured cells ..... 116
9. Others ..... 117
10. Conclusion and outlook ..... 117
Acknowledgments ..... 118
References ..... 118

**Abbreviations in alphabetical order**

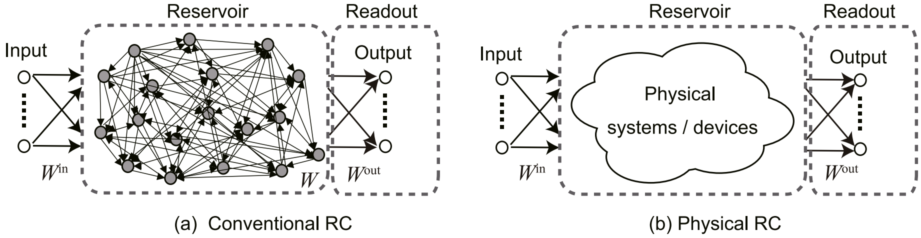
| | |
|---|---|
| AD: | Analog-to-Digital |
| ANN: | Artificial Neural Network |
| ASIC: | Application Specific Integrated Circuit |
| ASN: | Atomic Switch Network |
| BMI: | Brain Machine Interface |
| BPDC: | Backpropagation Decorrelation |
| BPTT: | Backpropagation Through Time |
| CA: | Cellular Automaton |
| CPU: | Central Processing Unit |
| DA: | Digital-to-Analog |
| DNA: | Deoxyribonucleic acid |
| ECG: | Electrocardiogram |
| EEG: | Electroencephalogram |
| EMG: | Electromyogram |
| ESN: | Echo State Network |
| fMRI: | functional Magnetic Resonance Imaging |
| FNN: | Feedforward Neural Network |
| FORCE: | First Order Reduced and Controlled Error |
| FPGA: | Field-Programmable Gate Array |
| GPU: | Graphics Processing Unit |
| LIF: | Leaky Integrate-and-Fire |
| LLG: | Landau–Lifshitz–Gilbert |
| LSM: | Liquid State Machine |
| LSTM: | Long Short-Term Memory |
| MEA: | Microelectrode Array |
| MFCC: | Mel-Frequency Cepstrum Coefficient |
| MTJ: | Magnetic Tunnel Junction |
| NARMA: | Nonlinear Autoregressive Moving Average |
| ODE: | Ordinary Differential Equation |
| PM | Particulate Matter |
| RC: | Reservoir Computing |
| RNN: | Recurrent Neural Network |
| RTRL: | Real-Time Recurrent Learning |
| SNN: | Spiking Neural Networks |
| SOA: | Semiconductor Optical Amplifier |
| STDP: | Spike-Timing-Dependent Plasticity |
| STO: | Spin Torque Oscillator |
| VCSEL: | Vertical Cavity Surface Emitting Laser |
| VLSI: | Very Large Scale Integration |
| XOR: | Exclusive OR |
| YIG: | Yttrium Iron Garnet |

## 1. Introduction

Artificial neural networks (ANNs) constitute the core information processing technology in the fields of artificial intelligence and machine learning, which have witnessed remarkable progress in recent years, and they are expected to be increasingly employed in real-world applications (Samarasinghe, 2016). ANNs are computational models that mimic biological neural networks. They are represented by a network of neuron-like processing units interconnected via synapse-like weighted links. Network architectures of ANNs are typically classified into feedforward networks (Schmidhuber, 2015) and recurrent networks (Mandic, Chambers, et al., 2001), the choice of which depends on the type of computational task. Feedforward neural networks (FNNs) are mainly used for *static* (non-temporal) data processing, as individual input data are independently processed even if they are given sequentially. In short, FNNs are capable of approximating nonlinear input–output functions. On the other hand, recurrent neural networks (RNNs) are suited for *dynamic* (temporal) data processing, as they can embed temporal dependence of the inputs into their dynamical behavior. In other words, RNNs are capable of representing dynamical systems driven by sequential inputs owing to their feedback connections.

Reservoir computing (RC) is originally an RNN-based framework and is therefore suitable for temporal/sequential information processing (Jaeger & Haas, 2004). Specifically, RC is a unified computational framework (Lukoševičius & Jaeger, 2009; Verstraeten, Schrauwen, d' Haene, & Stroobandt, 2007), derived from independently proposed RNN models, such as echo state networks (ESNs) (Jaeger, 2001) and liquid state machines (LSMs) (Maass, Natschläger, & Markram, 2002). The backpropagation decorrelation (BPDC) learning rule (Steil, 2004, 2007) for RNNs is also regarded as a predecessor of RC. Similar concepts and models in special cases were reported in earlier studies as summarized in Jaeger (2007), including sequential associative memory models (Gallant & King, 1988), neural oscillator network models for learning handwriting movements (Schomaker & Richardus, 1991, 1992), context reverberation networks consisting of linear threshold units for sequential learning (Kirby, 1991; Kirby & Day, 1990), cortico-striatal models for context-dependent sequence learning (Dominey, 1995; Dominey, Arbib, & Joseph, 1995), and biological neural network models for temporal pattern discrimination (Buonomano & Merzenich, 1995).

In RC, input data are transformed into spatiotemporal patterns in a high-dimensional space by an RNN in the *reservoir*. Then, a pattern analysis from the spatiotemporal patterns is performed in the *readout*, as shown in Fig. 1(a). The main characteristic of RC is that the input weights ($W^{\mathrm{in}}$) and the weights of the recurrent connections within the reservoir ($W$) are not trained whereas only the readout weights ($W^{\mathrm{out}}$) are trained with a simple learning algorithm such as linear regression. This simple and fast training process makes it possible to drastically reduce the computational cost of learning compared with standard RNNs, which is the major advantage of RC (Jaeger, 2002b). RC models have been successfully applied to many computational problems, such as temporal pattern classification, prediction, and generation. To enhance computational performance of RC, it is necessary to appropriately represent sample data and optimally design the RNN-based reservoir. The methods for obtaining effective reservoirs, which have been summarized in Lukoševičius and Jaeger



(a) Conventional RC (b) Physical RC

**Fig. 1.** RC frameworks where the reservoir is fixed and only the readout weights $W^{\rm out}$ are trained. (a) A conventional RC system with an RNN-based reservoir as in ESNs and LSMs. (b) A physical RC system in which the reservoir is realized using a physical system or device.

(2009), are categorized into task-independent generic guidelines and task-dependent reservoir adjustments.

The role of the reservoir in RC is to nonlinearly transform sequential inputs into a high-dimensional space such that the features of the inputs can be efficiently read out by a simple learning algorithm. Therefore, instead of RNNs, other nonlinear dynamical systems can be used as reservoirs. In particular, *physical RC* using reservoirs based on physical phenomena has recently attracted increasing interest in many research areas (Fig. 1(b)). Various physical systems, substrates, and devices have been proposed for realizing RC. A motivation for physical implementation of reservoirs is to realize fast information processing devices with low learning cost. For hardware implementation of normal RNNs where training is necessary, we often rely on advanced technologies of neural network hardware (Misra & Saha, 2010) and neuromorphic hardware (Hasler & Marr, 2013). In contrast, physical implementation of reservoirs can be achieved using a variety of physical phenomena in the real world, because a mechanism for adaptive changes for training is not necessary. Actually, physical RC is one of the candidates of unconventional computing paradigms based on novel hardware (Hadaeghi, He, & Jaeger, 2017). Although design principles for conventional RC, such as ESNs (Lukoševičius, 2012; Ozturk, Xu, & Príncipe, 2007) and LSMs (Maass, 2011), have been examined comprehensively, the following issues require further investigation: how to design physical reservoirs for achieving high computational performance and how much computational power can be attained by individual physical RC systems.

The purpose of this review is to provide an overview of recent advances in RC, with a special focus on physical RC. Published studies on physical RC can be found not only in neural network journals but also in specialized journals related to the respective physical systems. Our objective is to provide a comprehensive viewpoint with regard to interdisciplinary studies on physical RC by classifying them according to the type of the physical phenomenon utilized for the reservoir. Toward this end, we summarize the characteristics of individual physical reservoirs. Our classification, which highlights the similarities and differences among different physical reservoirs, is useful for gaining insights into further developments in physical RC.

Some physical reservoirs are promising for developing next-generation machine learning hardware devices and chips. Such hardware-based RC can significantly speed up data processing compared to software-based RC, and it is often motivated by the need to reduce the power consumed by machine learning hardware. In particular, the RC framework using a simple readout is suited to low-cost real-time computation, which is not achievable with other machine learning frameworks based on iterative learning algorithms leading to high learning cost. Currently, many online services rely on cloud computing (Armbrust et al., 2010) where tasks are performed at computers far from devices of end users on the Internet. Owing to the ever-growing amount of data at network edges, increasing communication latency is becoming a bottleneck for high-speed cloud computing. An emerging alternative computing paradigm for reducing the latency is edge computing (Shi, Cao, Zhang, Li, & Xu, 2016) where tasks are performed at computers close to devices of end users and sensors, preferably in real time. Machine learning hardware for real-time temporal data processing, such as RC hardware, enables efficient edge computing, and therefore, is expected to become increasingly significant in the future. On the other hand, physical constraints make it difficult to optimize the conditions of reservoirs in physical RC. Many issues remain to be addressed in order to realize efficient physical RC systems and devices for practical applications. Our review introduces potential candidates of physical reservoirs for such devices.

The remainder of this review is organized as follows. In Section 2, we briefly describe the basic concept of RC and its recent trends. In the subsequent sections, we introduce different types of physical RC systems, including RC based on well-known dynamical systems models (Section 3), electronic RC (Section 4), photonic RC (Section 5), spintronic RC (Section 6), mechanical RC (Section 7), biological RC (Section 8), and others (Section 9). Finally, we discuss the current issues and future research directions in Section 10.

## 2. Reservoir computing (RC)

First, we outline the fundamental concept of RC in Section 2.1. For a more detailed background and overview of conventional RC, readers may refer to the following well-organized articles: the survey paper (Lukoševičius & Jaeger, 2009), the review papers (Lukoševičius & Schrauwen, 2012; Scardapane & Wang, 2017; Schrauwen, Verstraeten, & Van Campenhout, 2007), and the special issue on ESNs and LSMs in *Neural Networks* (Jaeger, Maass, & Principe, 2007). Then, we discuss recent trends in RC in Section 2.2.

### 2.1. Basic framework

Since the 1980s, RNNs have been used for temporal/sequential pattern recognition. They are characterized by feedback (recurrent) connections for generating history-dependent dynamical responses to external inputs. These models are described as non-autonomous dynamical systems. Although a special type of RNNs without external inputs (the so-called Hopfield networks Hopfield, 1982) have also been widely studied (Tananka et al., 2019), we will not treat such models represented as autonomous dynamical systems in this review. The two major gradient-based learning algorithms for general RNNs with external inputs are backpropagation through time (BPTT) (Rumelhart, Hinton, & Williams, 1985; Werbos, 1990) and real-time recurrent learning (RTRL) (Doya, 1998; Williams & Zipser, 1989). These classical methods for training RNNs are discussed in a tutorial on RC (Jaeger, 2002b). In BPTT, an RNN is unfolded in time and regarded as an FNN with shared weights. Then the FNN is trained with the standard backpropagation algorithm (Rumelhart et al., 1985). In practice, a variant of this algorithm, truncated BPTT, is performed using only a finite history of data to adaptively update the trained model and save computational cost. Therefore, this method has a difficulty in learning long-term dependencies of sequential data. On the other hand, RTRL shows excellent performance in online learning, but its time complexity (i.e. computational time) is high. Most RNN-based methods, including long short-term memory (LSTM) networks (Hochreiter & Schmidhuber, 1997), use one of the above-mentioned algorithms or a combination of them.

In the early 2000s, ESNs (Jaeger, 2001; Jaeger & Haas, 2004) and LSMs (Maass, 2011; Maass et al., 2002) were independently proposed as seminal RC models. They are different from conventional RNNs in that the weights on the recurrent connections in the reservoir are not trained but only the weights in the readout are trained (Lukoševičius & Jaeger, 2009; Schrauwen et al., 2007). The nonlinear mapping of an input signal into a high-dimensional space in the reservoir is effective for pattern analysis of the input information as in the kernel method (Hofmann, Schölkopf, & Smola, 2008). To apply a simple machine learning method to the readout, the reservoir should be appropriately designed in advance. The characteristics of the two above-mentioned models, ESNs and LSMs, are briefly summarized below.

The ESN model was proposed by Jaeger (Jaeger, 2001, 2007; Jaeger & Haas, 2004). This model uses an RNN-based reservoir consisting of discrete-time artificial neurons (Fig. 1(a)). When the feedback from the output to the reservoir is absent, the time evolution of the neuronal states in the reservoir is described as follows (Jaeger, 2001):

$$\mathbf{x}(n) = \mathbf{f}(W^{\mathrm{in}}\mathbf{u}(n) + W\mathbf{x}(n-1)), \tag{1}$$

where $n$ denotes discrete time, $\mathbf{x}(n)$ is the state vector of the reservoir units, $\mathbf{u}(n)$ is the input vector, $W^{\mathrm{in}}$ is the weight matrix for the input-reservoir connections, and $W$ is the weight matrix for the recurrent connections in the reservoir. The function $\mathbf{f}$ represents an element-wise activation function of the reservoir units, which is typically a sigmoid-type activation function. Eq. (1) represents a non-autonomous dynamical system forced by the external input $\mathbf{u}(n)$. The output is often given by a linear combination of the neuronal states as follows:

$$\mathbf{y}(n) = W^{\mathrm{out}}\mathbf{x}(n), \tag{2}$$

where $\mathbf{y}(n)$ is the output vector and $W^{\mathrm{out}}$ is the weight matrix in the readout. In supervised learning, this weight matrix is trained to minimize the difference between the network output and the desired output for a certain time period (see Sec. 8 of Lukoševičius and Jaeger (2009) for details). The performance of the ESN depends on the design of the RNN-based reservoir. In order to approximate a teacher output signal, the RNN-based reservoir must have the *echo state property*, whereby it asymptotically eliminates any information from the initial conditions (Jaeger, 2001). It is empirically observed that the echo state property is obtained for any input if the spectral radius (i.e. the maximum absolute eigenvalue of $W$) is adjusted to be smaller than unity. A practical guide for reservoir design in ESNs can be found in Lukoševičius (2012). The echo state property is essentially the same as the fading memory property (Boyd & Chua, 1985; Matthews, 1993). An input/output system (or a filter) for temporal signals has fading memory when the outputs corresponding to inputs that are close in the recent past are close even if those inputs are very different in the distant past. Recently, it was theoretically shown that ESNs have universal approximation ability in terms of discrete-time fading memory filters with uniformly bounded inputs defined on negative infinite times (Grigoryeva & Ortega, 2018).

The LSM was proposed by Maass et al. (Maass, 2011; Maass et al., 2002) to explore the computational capability of neural microcircuits in the brain (Maass & Markram, 2004). The purpose of LSMs is to develop biologically relevant learning models using spiking neural networks (SNNs) with recurrent connectivity. The architecture of the LSM is similar to that shown in Fig. 1(a) and the reservoir units are typically given by excitatory and inhibitory spiking neurons. Although the units are principally modeled with leaky integrate-and-fire (LIF) neurons, other biologically plausible spiking neuron models can also be used (Grzyb, Chinellato, Wojcik, & Kaminski, 2009; Wojcik & Kaminski, 2007). The topology and connectivity of the RNN in the LSM follow the constraints of biological neural networks. Specifically, the probability that two neurons are connected depends on the distance between their positions. Such a reservoir is often called a *liquid* and the LSM operation is called *liquid computing* because it is similar to excitable media exhibiting ripples in response to external stimulation inputs. The reservoir dynamics is generally described as follows (Maass et al., 2002):

$$x^M(t) = (L^M u)(t), \tag{3}$$

where $t$ denotes continuous time, $x^M$ is the reservoir state (neuronal activation patterns), $u(\cdot)$ is the input encoded as a spike sequence, and $L^M$ is the filter for transforming the input into the reservoir state. The output is given by

$$y(t) = f^M(x^M(t)), \tag{4}$$

where $y(t)$ is the output and $f^M$ is a memory-less readout map. A simple machine learning algorithm or a biologically plausible learning rule can be adopted to train the readout map. The LSMs can incorporate new findings about biological mechanisms of information processing in the brain. It was shown that any given time-invariant filter (a transformation from $u(\cdot)$ to $y(\cdot)$) with the fading memory property can be approximated by LSMs to any degree of precision, if $L^M$ is chosen from a class of time-invariant filters with fading memory that has a point-wise separation property and $f^M$ is chosen from a class of functions that satisfies an approximation property (Maass & Markram, 2004; Maass et al., 2002). Recent studies have treated the universal approximation property of LSMs and other RC systems in a more mathematically rigorous way (Gonon & Ortega, 2018; Grigoryeva & Ortega, 2018).

### *2.2. Recent trends*

The number of studies on RC has been rapidly increasing in recent years. In this subsection, we discuss the recent trends in RC studies from several viewpoints, including applications, methods, and physical realizations.

First, RC has been successfully applied to many practical problems involving real data. One of the reasons for this success is that researchers outside the neural network community have recognized the advantages of RC. The simplicity of the training method in RC is attractive for non-expert developers. Table 1 lists examples of subjects that have been addressed using RC. Most of these studies are involved in machine learning applications, such as pattern classification, time series forecasting, pattern generation, adaptive filtering and control, and system approximation. In particular, RC meets the demands for low training cost and real-time processing in these applications. Some benchmark tasks related to these applications are listed in Table 2. The input and output information for an RC system are determined depending on the task. In pattern classification tasks, the input is a time series and the output is a discrete value (label) representing a pattern class. More specifically, in a spoken digit recognition task, the input is a sound signal corresponding to one of ten different utterances of the digits from zero to nine and the output is one of the ten digits. The goal of this task is to output the correct digit number from a sound signal of an unknown digit. To remove unnecessary information and noise, the original sound signal is

**Table 1**
Examples of subjects in RC applications.

| Category | Examples |
|---|---|
| Biomedical | EEG, fMRI, ECG, EMG, heart rates, biomarkers, BMI, eye movement, mammogram, lung images. |
| Visual | Images, videos. |
| Audio | Speech, sounds, music, bird calls. |
| Machinery | Vehicles, robots, sensors, motors, compressors, controllers, actuators. |
| Engineering | Power plants, power lines, renewable energy, engines, fuel cells, batteries, gas flows, diesel oil, coal mines, hydraulic excavators, steam generators, roller mills, footbridges, air conditioners. |
| Communication | Radio waves, telephone calls, Internet traffic. |
| Environmental | Wind power and speed, ozone concentration, PM2.5, wastewater, rainfall, seismicity. |
| Security | Cryptography. |
| Financial | Stock price, stock index, exchange rate. |
| Social | Language, grammar, syntax, smart phone. |

**Table 2**
Applications and related benchmark tasks of RC.

| Applications | Benchmark tasks |
|---|---|
| Pattern classification | Spoken digit recognition (Verstraeten, Schrauwen, Stroobandt and Van Campenhout, 2005)<br>Waveform classification (Paquot et al., 2012)<br>Human action recognition (Soh & Demiris, 2012)<br>Handwritten digit image recognition (Jalalvand, Van Wallendael, & Van de Walle, 2015) |
| Time series forecasting | Chaotic time series prediction (Jaeger, 2001)<br>NARMA time series prediction (Jaeger, 2003) |
| Pattern generation | Sine-wave generation (Jaeger, 2002b)<br>Limit cycle generation (Hauser, Ijspeert, Füchslin, & Maass, 2012) |
| Adaptive filtering and control | Channel equalization (Jaeger & Haas, 2004) |
| System approximation | Temporal XOR task (Bertschinger & Natschläger, 2004)<br>Temporal parity task (Bertschinger & Natschläger, 2004) |
| Short-term memory | Memory capacity (Jaeger, 2002a) |

typically transformed into feature values such as mel-frequency cepstrum coefficient (MFCC) in a preprocessing step and then given to the reservoir. Although RC is suited for temporal pattern recognition, it can be applied to image recognition by transforming an image into a sequence of pixel values. As RC can deal with any sequential data in principle, further expansion of its application fields is widely expected.

Second, many variants of RC models have been proposed to improve the performance of the original ones. New RC models have been devised by using new architectures such as multiple reservoirs (Akiyama & Tanaka, 2019; Gallicchio, Micheli, & Pedrelli, 2017; Malik, Hussain, & Wu, 2017) and evolving reservoirs (Qiao, Li, Han, & Li, 2017), combining RC with other feature extraction methods such as untrained convolutional neural networks (Tong & Tanaka, 2018) and reinforcement learning (Chang et al., 2018; Murakami, Kröger, Birkholz, & Triesch, 2015), incorporating new learning algorithms such as the FORCE learning (Sussillo & Abbott, 2009) and its variants (Pyle & Rosenbaum, 2018), and/or employing a diversity of reservoir elements (Inubushi & Yoshimura, 2017; Tanaka, Nakane, Yamane and Nakano et al., 2016; Xia, Jahanchahi, & Mandic, 2015). In addition, theoretical studies have provided a deeper understanding of the relationship between the computational performance of RC and the dynamics of reservoirs, in terms of nonlinear dynamical systems theory, information theory, and statistical theory. A comprehensive overview of these studies is beyond the scope of this review.

Finally, physical realizations of RC models have attracted considerable attention. A straightforward method is to implement RNNs using neural network hardware or neuromorphic computing techniques. Another method is to employ other dynamical systems instead of RNNs. Any dynamical system has the potential to serve as a reservoir if it can exhibit dynamical responses to inputs. Such reservoirs were previously regarded as "exotic" ones (Lukoševičius & Jaeger, 2009), but the number of studies on physical RC has been rapidly increasing. Various physical reservoirs have been proposed using different types of physical systems, substrates, and devices. Some physical RC systems are aimed at developing energy-efficient machine learning hardware and others are at exploring natural computing based on novel substrates.

There are several requirements for a physical reservoir to efficiently solve computational tasks. (i) *High dimensionality* is necessary to map inputs into a high-dimensional space. This property facilitates the separation of originally inseparable inputs in classification tasks and allows reading out spatiotemporal dependencies of inputs in prediction tasks. The dimensionality is related to the number of independent signals obtained from the reservoir. (ii) *Nonlinearity* is necessary for a reservoir to operate as a nonlinear mapping. This property allows inputs that are not linearly separable to be transformed into those that are linearly separable in classification tasks. It is also useful for effectively extracting nonlinear dependencies of inputs in prediction tasks. (iii) *Fading memory* (or short-term memory) (Boyd & Chua, 1985; Maass et al., 2002; Maass, Natschläger, & Markram, 2004) is necessary to ensure that the reservoir state is dependent on recent-past inputs but independent of distant-past inputs. It is also referred to as the echo state property, indicating that the influence of past inputs on the current reservoir states and outputs asymptotically fades out (Jaeger, 2001; Yildiz, Jaeger, & Kiebel, 2012). Such a property is particularly important for representing sequential data with short-term dependencies. (iv) *Separation property* is required to separate the responses of a reservoir to distinct signals into different classes. On the other hand, a reservoir should be insensitive to unessential small fluctuations, such as noise, so that similar inputs are classified into the same class. Therefore, when a system parameter variation causes a transition between non-chaotic and chaotic regimes, it is often recommended that the parameter be set close to the transition point (the so-called edge of chaos (Bertschinger & Natschläger, 2004; Legenstein & Maass, 2007)) where the transformation by a reservoir is neither very expanding nor very contracting.

The responses of physical RC systems are used to train a readout that is realized using physical devices or software-based computations. Linear regression or another simple machine learning algorithm is used in the readout of ESN-type RC (Lukoševičius & Jaeger, 2009), while a perceptron-like local learning rule or a synaptic plasticity-based one is used for the readout neurons in LSM-type RC (Maass et al., 2002).

An intriguing example of physical RC in an early study is a fluidic RC with water in a bucket for pattern recognition as shown in Fig. 2 (Fernando & Sojakka, 2003). The input signals are transmitted to electric motors that generate ripples on the water surface, and the ripples are recorded using a video camera. From the recorded signals, a software-based readout is trained. The performance of the liquid computer is demonstrated in an XOR task and a spoken digit recognition task. Subsequently, many dynamical systems models and physical systems have been employed as potential reservoirs. We classify these RC systems, substrates, and devices, depending on the type of the physical phenomenon in the reservoir and review the individual reservoirs in the following sections.

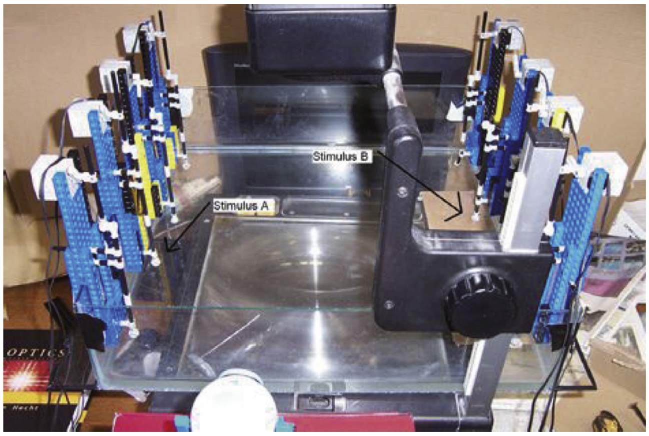


**Fig. 2.** A fluidic RC where the reservoir is the water in a bucket (Fernando & Sojakka, 2003). Figure reproduced with permission from Springer Nature.

## 3. Dynamical systems models for RC

In this section, we review several types of reservoirs based on well-known nonlinear dynamical systems models, including delayed dynamical systems (Section 3.1), cellular automata (Section 3.2), and coupled oscillators (Section 3.3).

### 3.1. Delayed dynamical systems

The reservoirs in ESNs and LSMs generate high-dimensional signals using a network of interacting neuron nodes, which are regarded as a special class of high-dimensional dynamical systems. Another way to generate high-dimensional patterns is to use a time-delayed dynamical system, as described in the following form (Lepri, Giacomelli, Politi, & Arecchi, 1994):

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = F(t, x(t), x(t-\tau)), \tag{5}$$

where $t$ represents continuous time, $x$ is the state variable, $F$ is a function determining the flow of this system, and $\tau > 0$ is the delay period. This system is capable of exhibiting rich nonlinear behavior including periodic oscillations and deterministic chaos depending on the system parameter setting. In the first proposal of a single-node reservoir with delayed feedback (Appeltant et al., 2011), the reservoir was implemented using electronic circuits with a feedback loop (see Section 4.1 for details). As shown in Fig. 3, the input signal is time-multiplexed by a mask function (Appeltant, Van der Sande, Danckaert, & Fischer, 2014) and fed to the single nonlinear node. The virtual nodes are set at $N$ time points that equally divide the delay period $\tau$. The time interval between two consecutive nodes is $\theta \equiv \tau/N$. The states at these virtual nodes, $x(t-(N-i)\theta)$ for $i = 1, \dots, N$, are used as the reservoir state at time $t$ and then fed to the output layer through weighted connections. These connection weights are trained in the readout. The system was successfully applied to the spoken digit recognition task and the nonlinear autoregressive moving average (NARMA)-10 time series prediction task.

The architecture of the single-node reservoir with delayed feedback was extended in two ways (Ortín & Pesquera, 2017). One is an ensemble of two separate time-delayed reservoirs whose outputs are combined at the readout. The other is a circular concatenation of the delay lines of two reservoirs, forming a longer delay line. These extended architectures were shown to achieve better performance, faster processing speed, and higher robustness than the single-node reservoir. An extensive amount of work has been performed on single-node reservoirs with delayed feedback (Brunner et al., 2018).

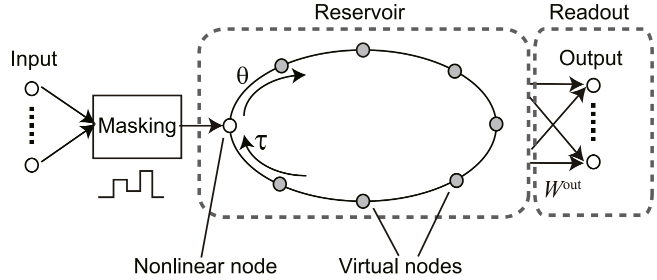


**Fig. 3.** RC using a single nonlinear node reservoir with time-delayed feedback (Appeltant et al., 2011).

The simplicity of the single-node reservoir with delayed feedback is advantageous for physical implementation compared with network-based reservoirs consisting of a large number of nodes. In fact, single-node reservoirs have been widely employed for electronic RC (Section 4) and photonic RC (Section 5.2).

### 3.2. Cellular automata

A cellular automaton (CA) is a simple dynamical systems model where both state and time are discrete (Wolfram, 2018). The discrete states on the cells are updated according to a given (local) evolution rule. Depending on the rule, the CA can exhibit rich behavior, including ordered, critical (or the edge of chaos), and chaotic dynamics, in spite of its simplicity. It is heuristically conjectured that the computational capability of CA is maximized at the edge of chaos.

This conjecture has been confirmed in a numerical study on RC based on a random Boolean network, which is an extended version of CA (Snyder, Goudarzi, & Teuscher, 2013). Other studies employed CA-based reservoirs as shown in Fig. 4 (Yilmaz, 2014,2015a,2015b). Binary or non-binary inputs are randomly mapped onto the initial states of CA through an encoding procedure. According to the predefined evolution rule, CA exhibits nonlinear dynamical behavior, through which the input data are projected onto an expressive and discriminative space. The entire state of the CA evolution is vectorized and used as a feature vector for processing in the readout. The CA reservoirs are binary in nature and suitable for symbolic computation including Boolean logic. The CA-based RC system can perform 5-bit and 20-bit temporal memory tasks, which require long short-term memory capability, with less computation compared to ESNs (Yilmaz, 2015a). The binary operations and simple update rules of CA reservoirs are advantageous for implementation with parallel hardware, such as field-programmable gate arrays (FPGAs) and graphics processing units (GPUs). In a recent study (Morán, Frasser, & Rosselló, 2018), different CA evolution rules were tested for the MNIST handwritten character recognition by numerical simulations and the best rule giving the highest accuracy was specified as rule 90 (Martin, Odlyzko, & Wolfram, 1984). Then, a CA-based reservoir with this rule was implemented with FPGA and applied to the MNIST task. Compared to other FPGA-based neural networks, the proposed method achieved competitive results in terms of accuracy, speed, and power dissipation.

The architecture of the CA reservoir can be extended in several ways. For instance, a parallel loosely coupled architecture (Nichele & Gundersen, 2017) and a layered deep architecture (Nichele & Molund, 2017) have been proposed. The rules of CA can also be extended by employing two CA rules of different classes in the reservoir layer to satisfy two competing requirements of the CA reservoir, i.e., sensitivity to the current input and

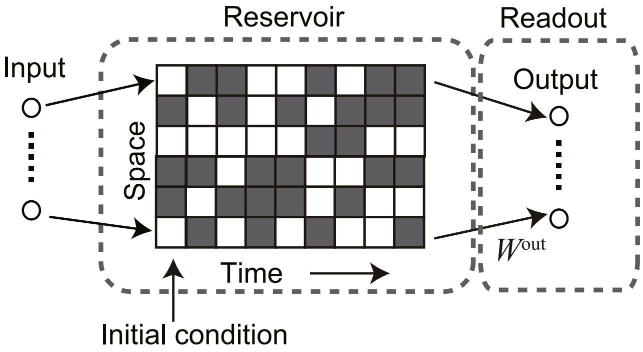


**Fig. 4.** RC using cellular automata (Yilmaz, 2015b).

asymptotic insensitivity to past inputs (McDonald, 2017). The first half of the CA reservoir is driven by a rule generating chaos or the edge of chaos for hyperdimensional projection of the inputs, while the second half is driven by a rule generating an ordered state for short-term memory.

### 3.3. Coupled oscillators

Coupled nonlinear oscillators are ubiquitous in mechanical, chemical, electronic, biological, optical, spintronic, and quantum mechanical systems. They can be used as physical reservoirs. When each oscillator is described with a first-order ordinary differential equation (ODE), a system of $N$ coupled oscillators can be described in the following general form:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = F(x_i(t)) + G(x_1(t), \ldots, x_N(t)), \quad \text{for } i = 1, \ldots, N, \tag{6}$$

where $t$ represents continuous time, $x_i(t)$ is the state of oscillator $i$ at time $t$, $F$ is a function determining the dynamics of isolated oscillators, and $G$ is a coupling function. The coupled oscillator model consists of the term representing the dynamics of individual oscillators and the coupling term representing the interactions between oscillators.

A reservoir based on nonlinear mechanical oscillators has been proposed (Coulombe et al., 2017). Such a reservoir is composed of multiple inertial masses arranged in a chain, which are coupled with their nearest neighbors by springs, as illustrated in Fig. 5(a). The masses are grounded by linear or nonlinear springs with damping. The entire equation of motion is described by coupled Duffing oscillators (second-order ODEs) driven by periodic forcing. The input signal is applied to the reservoir through amplitude modulation of the periodic forcing. The outputs from the reservoir are envelope signals containing only the low-frequency amplitude variations in the high-frequency mass position signals. The output signal is calculated by a linear combination of the weighted envelope signals. A reservoir with nonlinear mechanical oscillators can be compactly fabricated using microelectromechanical systems technology in an energy-efficient manner. The performance of such a reservoir computer was evaluated in a parity task and a spoken digit recognition task. Other mechanical reservoirs are reviewed in Section 7.

Chemical reactions are often modeled with coupled chemical oscillators to reproduce their oscillatory behavior far from a steady state. A deoxyribonucleic acid (DNA) reservoir was designed with coupled deoxyribozyme-based oscillators at molecular scale as shown in Fig. 5(b) (Goudarzi et al., 2013). The reservoir consists of different DNA species interacting via biochemical reactions in a microfluidic reaction chamber, which can be mathematically represented as coupled ODEs with state variables representing concentrations of the product molecules of the DNA species. The input signal is encoded as fluctuations in the influx of species into the reactor, and the reservoir state is monitored by fluorescent probes. A software-based readout is trained to map the oscillator dynamics to a target output. A chemical oscillator-based RC with three DNA species was applied to a temporal signal-tracking task.

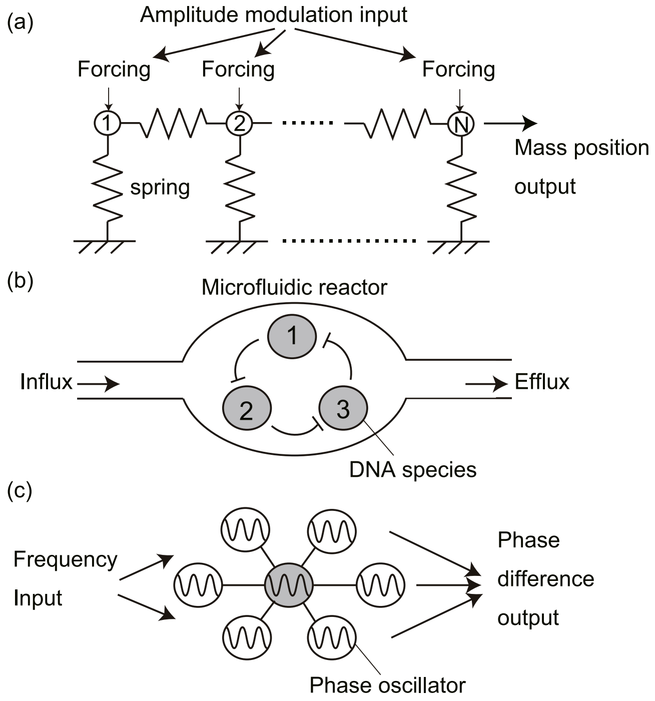


**Fig. 5.** RC using coupled oscillators. (a) A mechanical reservoir with coupled nonlinear oscillators (Coulombe, York, & Sylvestre, 2017). (b) A DNA reservoir with coupled chemical oscillators (Goudarzi, Lakin, & Stefanovic, 2013). (c) A synchronization-based RC with coupled phase oscillators (Yamane, Katayama, Nakane, Tanaka, & Nakano, 2015).

When coupled nonlinear oscillators show limit cycle motions, they can be reduced to coupled phase oscillators under the assumption of weak interactions (Nakao, 2016). After the reduction, the amplitude of the motion is eliminated and the dynamics is restricted to the phase domain. Coupled phase oscillators can exhibit rich dynamical behavior, including phase transition, clustering, and phase synchronization, which are available for RC. A phase-based RC encodes the input signal as phases of the oscillators and adopt phase synchronization for computation as shown in Fig. 5(c) (Yamane et al., 2015). In this method, the phase coupling function is appropriately designed to perform function approximations. This approach based on synchronization can contribute to development of phase-based information processing (Parihar, Shukla, Jerry, Datta, & Raychowdhury, 2017) and wave-based neuromorphic computing (Katayama, Yamane, Nakano, Nakane, & Tanaka, 2016).

## 4. Electronic RC

RC systems implemented with electronic circuits and devices have been actively studied for developing machine learning devices with low training cost. Any existing ANN and neuromorphic circuits are available in principle as electronic reservoirs, but simpler configurations have been explored to reduce energy consumption, speed up computation, and cope with imperfection and noise in hardware. In this section, we start with single-node reservoirs implemented with analog circuits (Section 4.1) and then introduce RC systems implemented with FPGAs which

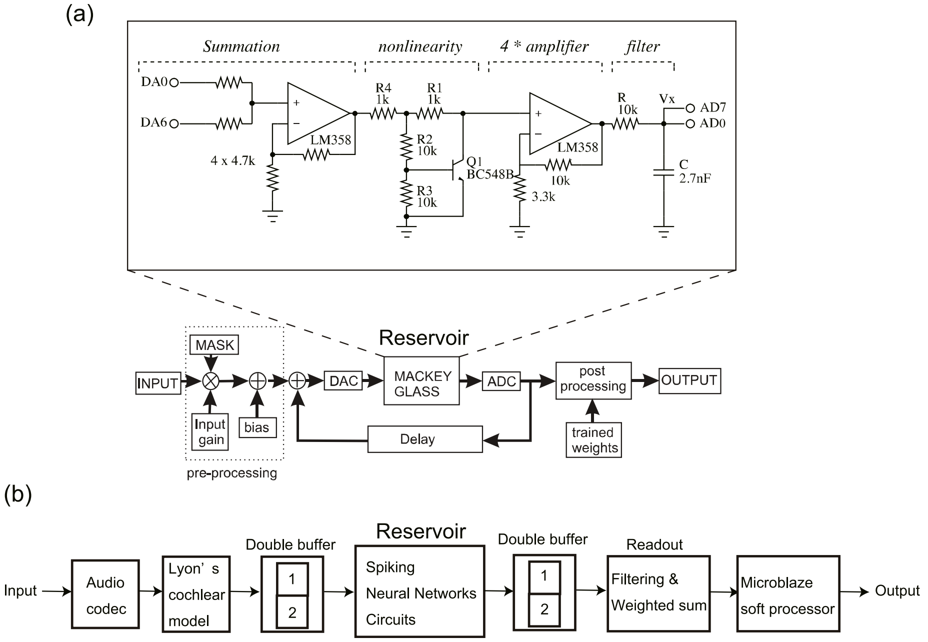


**Fig. 6.** (a) Electronically implemented RC system using a single-node reservoir based on the Mackey–Glass equation with delay (Soriano, Ortín and Keuninckx et al., 2015). Figure reproduced and modified with permission from Springer Nature. (b) Architecture of an FPGA-based LSM system for speech recognition (Schrauwen, D' Haene, Verstraeten, & Van Campenhout, 2008). Figure reproduced with permission from Elsevier.

are common reconfigurable hardware devices consisting of large arrays of simple configurable logic blocks and configurable interconnection structures (Section 4.2). Subsequently, we review very-large-scale integrated circuit (VLSI) designs for RC devices (Section 4.3). Finally, we focus on memristive RC based on memristive units (Section 4.4).

### 4.1. Analog circuits

As described in Section 3.1, a single nonlinear node with delayed feedback works well as a reservoir where the input information is transformed into the states of the virtual nodes. The single-node reservoir imposes less hardware requirements compared to a network-type reservoir consisting of a large number of units and interconnections (Soriano, Brunner, Escalona-Morán, Mirasso and Fischer, 2015). A nonlinear analog electronic circuit was implemented for constructing a single-node reservoir with a delay line, in combination with other digital hardware components for pre- and post-processing (Appeltant et al., 2011; Soriano, Ortín and Keuninckx et al., 2015). The system architecture is schematically shown in Fig. 6(a). The input signal is time-multiplexed with a mask that defines the connection weights from the input to the virtual nodes on the delay line (Appeltant et al., 2014) and kept to be positive by adding a bias voltage by digital processing. Then, the converted signal is injected into the single nonlinear node implemented with an analog Mackey–Glass nonlinear element circuit with a delayed feedback (the upper panel in Fig. 6(a)), corresponding to the following equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = -x(t) + \frac{\eta(x(t-\tau)+\gamma I(t))}{1+(x(t-\tau)+\gamma I(t))^p}, \tag{7}$$

where $t$ is dimensionless time, $x$ is the dynamical variable, $\tau$ is the delay in the feedback loop, $\eta$ is the feedback strength, $\gamma$ is the input scaling, $I(t)$ is the external input current, $p$ is the parameter for tuning the nonlinearity. The states of the virtual nodes are linearly combined to produce the output in the digital postprocessing part. The training of the output weights are performed using a linear regression algorithm. The digital and analog parts are interfaced by digital-to-analog (DA) and analog-to-digital (AD) converters with 12-bits resolution. The proposed system was successfully applied to spoken digit recognition, memory capacity estimation, and time series prediction, by appropriately adjusting the feedback strength $\eta$. The effect of the quantization noise caused by AD and DA conversion on the computational performance was also investigated in comparison with numerical simulations. A recent study proposed an extended system composed of multiple delayed feedback reservoirs based on the Mackey–Glass circuit in a deep layer structure (Li, Bai, Liu, & Yi, 2018). The presented system was applied to time series prediction tasks with Santa Fe dataset and ECG signals. Another study considered a single-node reservoir implemented with the Chua's circuit driven by external input forcing, which exhibits chaotic dynamics, and applied it to non-temporal nonlinear tasks (Jensen & Tufte, 2017).

On the other hand, a spike-based single-node reservoir with a delay loop was proposed with its analog implementation design (Zhao et al., 2016). The information transmission is based on spike signals for power efficiency, instead of time-continuous analog signals requiring peripheral modules for signal conversion and amplification. The spike-based delayed feedback circuit demonstrated nonlinear transformation from input spike sequences to output spike sequences. A spike timing-dependent encoder for encoding analog inputs as temporal spike trains and a masking process were used to design an RC system with a spike-based single-node reservoir (Li, Zhao, Hamedani, & Yi, 2017). Numerical experiments showed that the optimal mask for yielding high computational performance was different depending on the task.

### 4.2. FPGAs

FPGAs have often been used to implement ANNs as their reconfigurability is suited for concurrent processing and adaptive weight updating in ANNs (Omondi & Rajapakse, 2006; Zhu & Sutton, 2003). In the context of RC, many FPGA implementations have been studied to realize reservoirs and/or readouts (Antonik, 2018). Network-based reservoirs can be constructed using a variety of components, such as binary neurons, sigmoid neurons, stochastic neurons, and spiking neurons. A significant issue is how to choose hardware-friendly network components and efficiently implement them.

Binary neurons are often employed to compose neural network circuits as they are suited to be handled in digital platforms. In an early study, a reservoir composed of stochastic bitstream neurons was implemented on an FPGA board to explore efficient hardware implementation (Verstraeten, Schrauwen and Stroobandt, 2005). The stochastic bitstream neuron is based on stochastic computing where its state value is represented as a statistical characteristic of a sufficiently long bitstream, e.g. as a probability of "1" in a binary sequence (Bade & Hutchings, 1994). Whereas sigmoid neurons require a large number of multipliers and adders for updating their states, stochastic bitstream neurons enable to simplify hardware implementation of those arithmetic operations. The proposed system with a small-world network of 25 neurons was applied to a simple task of generating a phase-shifted version of the input sinusoidal signal. A similar reservoir based on stochastic logic was implemented on an FPGA board using a random network of neuron units (Alomar, Canals, Martínez-Moll, & Rosselló, 2014). The proposed system composed of 10 neurons was applied to a nonlinear time series prediction task. The above FPGA-based reservoir computers use offline batch learning. This limitation was overcome by the first FPGA implementation of a reservoir computer with an online learning algorithm (Antonik, Smerieri, Duport, Haelterman, & Massar, 2015). The neuron units in this reservoir have a sinusoidal activation function and the linear readout provides a gradient descent algorithm for real-time learning. It was confirmed that the reservoir computer on the FPGA board has a significant advantage in the high-speed processing over the software-based reservoir implemented in a high-end laptop. It was demonstrated that the proposed system can successfully solve channel equalization problems with invariable channels and those with variable channels.

Implementations of recurrent SNNs for LSMs on FPGAs have been reported in many studies. The hardware LSMs can be regarded as a special form of neuromorphic hardware. The information is coded by temporal patterns in spike sequences, unlike the rate coding in analog neuron hardware. Efficient hardware architectures for implementing LSMs on an FPGA were explored for real-time information processing and applied to isolated digit speech recognition (Schrauwen et al., 2008). The overview of the system architecture for speech processing is shown in Fig. 6(b). The speech signals are transformed into cochleograms by the Lyon cochlear ear model (Lyon, 1982) and then converted to spike signals via Bens Spiker Algorithm (Schrauwen & Van Campenhout, 2003) before being fed to the SNN reservoir. The spikes generated in the SNN are filtered by a low-pass filter and sampled, and the weighted sum of the sampled signals is used to produce the system output in the postprocessing step. The SNN consists of 200 LIF neurons and the exponential synapse models (Gerstner & Kistler, 2002). A novel implementation technique for digital SNNs was devised on the basis of serial arithmetic operations instead of parallel ones, in order to enhance scalability of the hardware device. Another study presented an efficient design and architecture for general-purpose FPGA-based LSMs based on LIF neuron models with a biologically plausible learning rule (Wang, Jin, & Li, 2015). As a method to enhance energy efficiency, a light mode where some neuron units in the SNN are powered off was considered for simple tasks. The proposed neuromorphic LSM was successfully applied to speech and image recognition tasks with four different datasets. Furthermore, it was shown that the stand-alone FPGA-based LSM with parallel processing can speed up the runtime and reduce the energy consumption using the firing-activity dependent power gating and approximate arithmetic, compared to the implementation on a general-purpose CPU (Wang, Li, & Li, 2016; Wang, Li, Shao, Dey, & Li, 2017). Other FPGA implementations of LSMs were demonstrated using stochastic spiking neurons in a ring network (Alomar, Canals, Morro, Oliver, & Rossello, 2016), multiplier-less reconfigurable architectures (Ghani et al., 2015), and a spike-time-dependent encoder (Yi et al., 2016). In a recent work, a novel digital neuromorphic architecture called a spiking temporal processing unit was proposed by incorporating temporal buffers in each spiking neuron to model arbitrary synaptic response functions (Smith et al., 2017). This scheme was adopted to implement an LSM on an FPGA chip, and its pattern classification ability was evaluated in a spoken digit recognition task. In another FPGA-based study, improvements in cost and energy efficiency of hardware LSMs were achieved by developing a reservoir tuning method based on a hardware-friendly spike-timing-dependent plasticity (STDP) learning algorithm in speech and image recognition (Liu, Jin, & Li, 2018).

The remaining FPGA-based reservoirs are the single-node reservoirs with delayed feedback (see Sections 3.1 and 4.1). An FPGA-based single-node reservoir with a delay loop was demonstrated using a Boolean logic element (Haynes, Soriano, Rosin, Fischer, & Gauthier, 2015). The Boolean logic element was set to operate as an XOR gate and the delay loop was realized as a cascade of pairs of inverter gates. The states of the virtual nodes are collected and linearly combined to produce an output in an external computer. It was shown that, by appropriately adjusting parameters such as the length of the time delay, the transient dynamics of the single-node reservoir was useful for classification of short input patterns. In another study, the single nonlinear node was realized with a digitized version of the Mackey–Glass equation in Eq. (7) and the delay line was implemented with the random access memory block (Alomar et al., 2015). The training of output weights is conducted on the FPGA board. The presented system was successfully applied to a waveform pattern classification task and a time series prediction task.

### 4.3. VLSIs

Some efforts have made to efficiently integrating reservoir circuits in VLSIs. In an early study, an electronic reservoir was implemented with a general-purpose ANN application-specific integrated circuit (ASIC) which is a mixed-mode hardware using analog computation and digital signaling (Schürmann, Meier, & Schemmel, 2005). This system consists of 256 McCulloch–Pitts binary neuron nodes and 33k analog synapses. The input and output information are given by binary sequences. Experimental results showed that the computational performance of the RC system in a temporal 3-bit parity task was maximized when the neural network exhibited critical dynamics at the edge of chaos, as in the case of previous results of software simulations (Bertschinger & Natschläger, 2004).

Neuromorphic approaches to RC based on spikes and pulses have recently attracted much attention. Prototypes of pulse-based information processors were fabricated on a printed circuit board and an ASIC, on which RC systems can be implemented (Petre & Cruz-Albrecht, 2016). These systems are based on asynchronous pulse processing that uses a time encoder for converting analog signals into pulse domain signals and an event-driven computing scheme for power efficiency. In another study, a digital design architecture for implementing LSMs on reconfigurable platforms was proposed for real-time processing of input data (Polepalli, Soures, and Kudithipudi, 2016a,2016b). The reservoir consists of LIF neurons with random connectivity having spatial locality. The readout layer consists of a two-layer perceptron and the output weights are adjusted based on a gradient descent method. The performance of the proposed model was numerically evaluated in epileptic seizure detection from EEG signals and user identification from walking patterns. Hardware-friendly readout implementation techniques have been focused on in some studies for energy-efficient learning in VLSI-based LSMs. Inspired by dendrites in biological neurons, a hardware-friendly readout architecture for LSMs and an efficient learning rule with adaptive network rewiring were proposed (Roy, Banerjee, & Basu, 2014). The efficiency of the proposed architecture was demonstrated in a spike train classification task and an approximation task of retrieving the sum of firing rates of input spike trains. In another study, a biologically inspired local learning rule was presented for low-power VLSI implementation of LSMs to reduce hardware implementation costs (Zhang, Li, Jin, & Choe, 2015). It was numerically shown that the overhead of hardware implementation can be reduced by the new learning rule in a speech recognition task. The tradeoff between hardware overhead and computational performance in hardware-implemented LSMs was discussed in a simulation study (Jin & Li, 2017).

### 4.4. Memristive RC

This section focuses on reservoirs implemented with memristive circuits and devices. The main characteristic of a memristive element, differentiating from other fundamental circuit elements, is that its resistance changes with time depending on the current flow that has passed through it (Chua, 1971; Williams, 2014). The existence of a memristor (or memory resistor) was first predicted from theoretical consideration (Chua & Kang, 1976) and after a long time physically realized using titanium dioxide ($TiO_2$) (Strukov, Snider, Stewart, & Williams, 2008). Although there are some debates about whether the memristor is truly counted as a fundamental passive circuit element (Abraham, 2018), memristive devices indeed exist in reality and are promising for computing. We classify memristive reservoirs into two types: neuromemristive reservoirs consisting of both neuron circuits and memristor synapses (Section 4.4.1) and memristive reservoirs without neuron units (Section 4.4.2).

#### 4.4.1. Neuromemristive circuits

Neuromorphic computing with non-von Neumann architecture has rapidly progressed in recent years. Neuromorphic devices/chips have been fabricated by different research groups (Walter, Röhrbein, & Knoll, 2015). Neuromemristive systems are a subclass of neuromorphic computing systems that use memristors to mimic the synaptic plasticity in biological neurons, where the memristor conductance corresponds to the synaptic weight (Indiveri, Linares-Barranco, Legenstein, Deligeorgis, & Prodromakis, 2013; Thomas, 2013). Nano-scale memristive synapses have considerable potential as elements of energy-efficient neuromorphic devices.

A memristor-based ESN with cellular neural network structures (i.e., locally connected arrays) was proposed using staircase memristor models (Yang, Chen, & Wang, 2016). Numerical simulation showed that, due to the simple structure, the performance of the proposed model was successful but worse than that of the original ESN model with a random connection topology in a time series prediction task. Other ESN-type reservoirs were designed using memristor crossbar arrays (Donahue et al., 2015; Merkel, Saleh, Donahue, & Kudithipudi, 2014). Memristor crossbar arrays are often used to implement direct synaptic connections in neuromorphic devices, because they are suitable for vector–matrix multiplication, and the conductances of memristive synapses can be adaptively updated by applying voltage pulses. Double crossbar arrays were used to realize recurrent connections in ESNs (Hassan, Li, & Chen, 2017) and LSMs (Soures, Hays, & Kudithipudi, 2017). The use of memristor crossbar arrays in the readout part of ESNs was proposed for a digital (or mixed signal) reservoir with a doubly twisted toroidal structure (Kudithipudi, Saleh, Merkel, Thesing, & Wysocki, 2016). A more general on-chip system using analog memristive nanosynapses was presented for emulating a reservoir computer and performing spoken digit and MNIST image recognition tasks (Bennett, Querlioz, & Klein, 2017).

#### 4.4.2. Memristive systems and devices

Memristive systems and devices are capable of exhibiting nonlinear dynamics and responding to inputs in a history-dependent manner. Even without neuron units, memristors can exhibit nonlinear transformation of input signals. By exploiting these favorable properties for a dynamic reservoir, some studies have proposed reservoirs based on memristive systems and devices. There are various memristive devices having different current–voltage characteristics, but ideally, they can be formulated as memristive systems categorized into a special class of dynamical systems (Chua, 1971; Chua & Kang, 1976). A current-controlled memristor with time-invariant characteristic is generally described as follows:

$$V = R(\mathbf{w}, I)I, \tag{8}$$

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = f(\mathbf{w}, I), \tag{9}$$

where $t$ represents continuous time, $V$ is the voltage, $R$ is the time-varying resistance, $I$ is the current, and $\mathbf{w}$ is a vector representing the internal state of the system. The function $f$ determines how the internal state evolves depending on the input current.

A network of memristors can be used as a reservoir to nonlinearly map an input signal into a high-dimensional feature space, as shown in Fig. 7(a). The first memristor-based reservoir was proposed in Kulkarni and Teuscher (2012). The proposed model was applied to a wave pattern classification task and an associative memory task, but the readout was a genetic algorithm which is not a linear algorithm. Therefore, the merit of the memristive reservoir was not clear from the results. As the connectivity of the memristors affects the computational performance of memristor-based RC, subsequent studies have extended the structure of the memristor network to a regular mesh structure for investigating the variation tolerance of RC (Bürger & Teuscher, 2013) and to a hierarchical structure for enhancing the computational capability of RC (Bürger, Goudarzi, Stefanovic, & Teuscher, 2015b). Moreover, the correlation between the computational capacity and the energy consumption of RC with random memristor networks was investigated to clarify its potential and limitations (Bürger J. Goudarzi, Stefanovic, & Teuscher, 2015a). In the above-mentioned studies, the behavior of memristor-based reservoirs was computed using circuit simulators. On the other hand, a general mathematical model of memristor networks was explicitly formulated to promote theoretical and numerical analyses of memristive RC (Tanaka et al., 2017). This formulation is applicable to any memristor network if the memductance (memristor conductance) of a single memristor is expressed as a function of its magnetic flux as in the linear drift model (Strukov et al., 2008). The numerical simulation showed that the variability in the memristive elements, which is generally thought to be undesired, can be beneficial if the network topology and the input scaling are appropriately selected. In Carbajal, Dambre, Hermans, and Schrauwen (2015), new memristor models with volatility were proposed to introduce the fading memory property into the standard non-volatile memristor model. Other memory devices, such as memcapacitors, are potential components of a network-type reservoir with low power consumption (Tran & Teuscher, 2017).

Several studies have demonstrated physical implementations of memristive reservoirs. A memristor-based reservoir device was

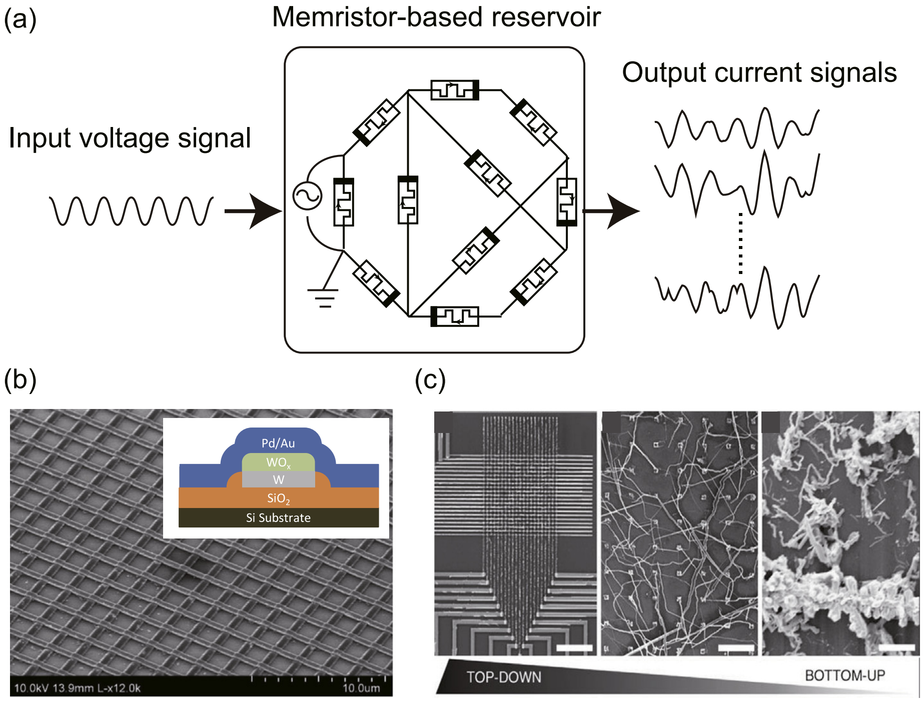


**Fig. 7.** (a) Schematic illustration of an RC system based on a memristor-based reservoir. (b) A fabricated memristor array where several cells are selected to work as a reservoir on the whole (Du et al., 2017). The inset shows tungsten oxide (WO$_x$) memristor with a metal–insulator–metal structure. Figure reproduced from Du et al. (2017), licensed under CC-BY 4.0. (c) An atomic switch network device composed of self-assembled silver nanowires (Stieg et al., 2014). Figure reproduced with permission from Springer Nature.

recently fabricated as shown in Fig. 7(b) (Du et al., 2017), where the reservoir is not a network of memristors but a group of independent memristors. An input signal is divided into multiple segments, and the segments are then separately transformed into output signals by the individual memristive devices. The collection of the output signals is used as the state of the entire reservoir. It was experimentally demonstrated that this RC device performed well in image recognition and time series prediction tasks. Memristive behavior is observed also in atomic switch networks (ASNs), which are aggregations of a number of silver nanowires interconnected with each other and formed via a thermodynamically driven self-organized growth process, as shown in Fig. 7(c) (Stieg et al., 2012). The RC device based on ASNs was applied to a waveform generation task (Sillin et al., 2013; Stieg et al., 2014).

## 5. Photonic RC

In this section, the rich literature on photonic reservoir implementations is categorized and briefly summarized. An alternative discussion on this specific area can be found in Van der Sande, Brunner, and Soriano (2017). Reviews on topics related to photonic spike processing (Prucnal, Shastri, de Lima, Nahmias, & Tait, 2016) and photonics for neuromorphic applications (Ferreira, Shastri, Tait, Nahmias, & Prucnal, 2017) have also been published recently. In this review, optical reservoirs are categorized as spatially distributed array reservoirs (Section 5.1) and reservoirs with delay feedback (Section 5.2). On this last category, a detailed tutorial has recently been published (Brunner et al., 2018).

### *5.1. Optical node arrays*

A photonic reservoir computer, with the potential advantages of low power consumption and extremely fast computation, was first proposed in 2008 and subsequently refined on the basis of numerical simulations (Vandoorne, Dambre, Verstraeten, Schrauwen, & Bienstman, 2011; Vandoorne et al., 2008). The design is based on a chip-integrated device with single-mode waveguides. Semiconductor optical amplifiers (SOAs) are assembled in a 4 × 4 array, with each node connected to a maximum of four neighbors in a swirl configuration, as shown in Fig. 8. The effects of non-uniformity of the nodes and the delay and phase shift between interconnections were studied. An experimental prototype was reported in 2014 (Vandoorne et al., 2014). Readout was performed at 11 of the 16 nodes by photodetectors providing nonlinearity, owing to the limitations imposed by optical loss. The signal input was inserted via a single node and the training of the reservoir and processing of the output were performed offline. Experimental demonstrations include a 2-bit XOR logical operator, header recognition, and spoken digit classification.

A proposed related design is a platform in which the nodes are microring resonators providing nonlinear responses (Mesaritakis, Papataxiarhis, & Syvridis, 2013). With a randomly interconnected 6 × 6 array, successful operation for a pattern recognition task was numerically demonstrated. Another related study developed a node-based framework adapted to a coupled-mode theory to simulate a large number of cavities efficiently (Vaerenbergh, Fiers, Dambre, & Bienstman, 2015). A drawback of photonic reservoirs is that a compact design results in short time delays; hence, the required input and readout operation rates may be too high for practical implementation. A digital mask modulation technique was reported to alleviate this problem and decrease the input signal rate by a factor of 40 (Schneider, Dambre, & Bienstman, 2016). There is a design choice for the number of nodes that are linked to the input signal. This aspect was recently studied, and it was found that the power efficiency was improved for an input signal fully connected to the array (Katumba, Freiberger, Bienstman, & Dambre, 2017). An intrinsic limitation of single-mode waveguide-integrated passive optical reservoirs is loss accumulation for a large number of nodes and long delay lines. A chip-integrated multimode photonic circuit with a low-loss optical Y-junction combiner was proposed, simulated, and applied to a header recognition task (Katumba et al., 2018).

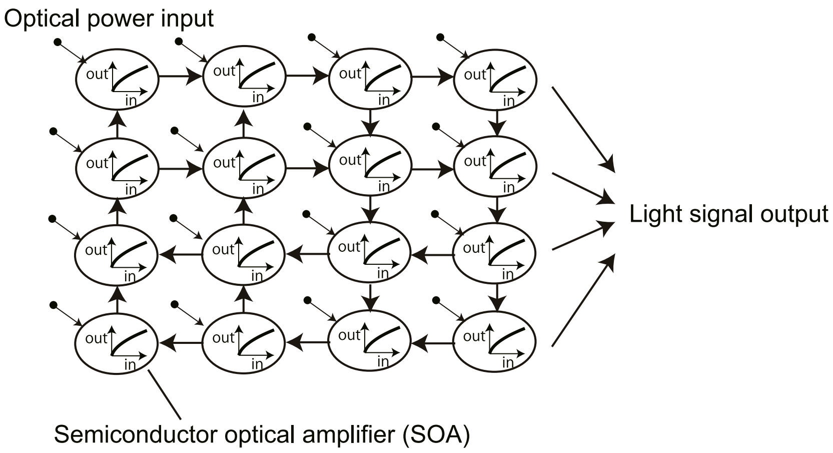


**Fig. 8.** A 4 × 4 node array optical reservoir with a swirl connection pattern (Vandoorne et al., 2011).

The concept of a device built instead on a photonic crystal platform was first proposed for a multiple superimposed oscillator waveform prediction task (Fiers et al., 2014). Good simulation results were reported for a reservoir structure made of an array of resonator cavities for which there is bistability in the optical power. More recently, a reservoir built from photonic crystal cavities with multiple waveguide inputs and outputs was designed to exploit field mixing dynamics and was shown to exhibit memory up to 6 bits (Laporte, Katumba, Dambre, & Bienstman, 2018).

Another family of reservoirs with photonic nodes is based on free-space optics principles. A configuration with a diffraction grating and Fourier imaging was designed with randomly interconnected microring resonators and applied to an imaging pattern task (Mesaritakis, Bogris, Kapsalis, & Syvridis, 2015). Another reservoir with a diffractive optical element was recently described on the basis of an 8 × 8 laser array and a spatial light modulator. Rich interaction dynamics was experimentally observed, and the potential for scaling up to a complex configuration with low power consumption was illustrated (Brunner & Fischer, 2015; Van der Sande et al., 2017). Further modifications to this setup, including the use of a laser illumination field and digital micro-mirror device, led to the realization of an RNN with hundreds of nonlinear optical nodes applied to a time series prediction task (Bueno et al., 2018). A reservoir in which node interaction comes from the scattering of light passing through a microscopic slide with a simple layer of white paint was also proposed and applied to logical function operation, with signal input via a micro-mirror array (Dong, Gigan, Krzakala, & Wainrib, 2018).

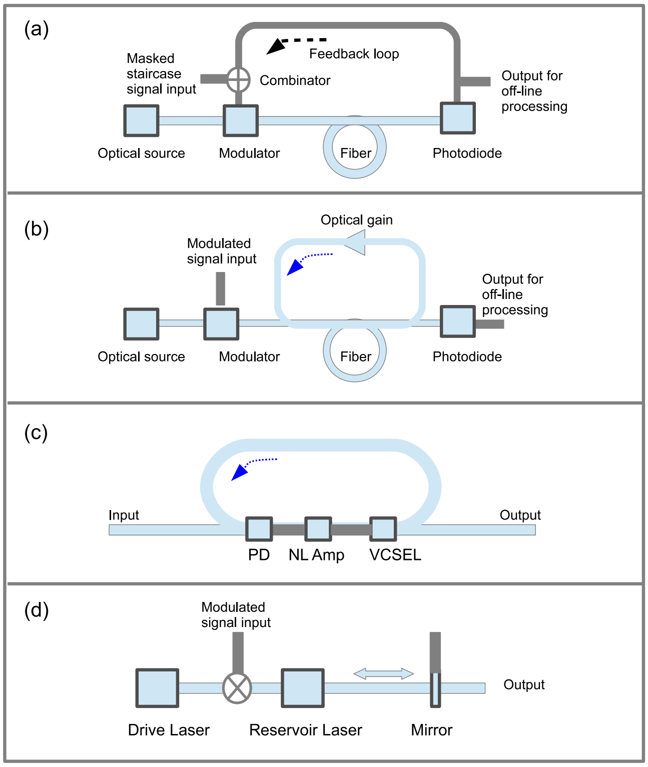


**Fig. 9.** Time-delay reservoir configuration examples. Blue and gray lines represent optical and electronic signals, respectively. (a) Opto-electronic feedback loop. (b) All-optical feedback loop with gain. (c) Passive optical feedback and opto-electronic node with gain. (d) Feedback into a laser cavity by a partially reflecting mirror . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 5.2. Time-delay systems

### 5.2.1. Opto-electronic and optical feedback with gain

The most extensively studied implementation of an optical reservoir computer is a configuration that uses a single physical node with a time-delayed feedback signal (Appeltant et al., 2011), as explained in Section 3.1. The input signal is converted into a staircase waveform by a sample-and-hold procedure. A weight mask related to the number of virtual nodes in the feedback loop is applied to each symbol step. The weights of the output signal are calculated offline during the training procedure. In this section we describe this class of systems as well as related designs.

In the first experimental demonstrations (Larger et al., 2012; Paquot et al., 2012), the feedback loop was optoelectronic. Light from a continuously emitting laser source is sent to a modulator for information processing. The optical output reaching a photodiode is amplified and electronically combined with the input signal to drive the modulator, as shown in a generic form in Fig. 9(a). Readout is performed either by sampling the feedback photodiode output or with an optical splitter collecting part of the light in the loop and a second photodiode. Demonstrations include generation of the NARMA equation of order 10, equalization of a nonlinear channel, and spoken digit recognition. In these pioneering demonstrations, the signal input rate was in the MHz range.

A number of related published studies have reported different components, design optimization, and the addition of new functionality. A reservoir in which the input signal is fed by directly modulating the laser, node nonlinearity is achieved by a birefringent plate, and multiple feedback loops with different delay times for enhanced connectivity are realized with an FPGA, was reported (Martinenghi, Rybalko, Jacquot, Chembo, & Larger, 2012). Designs with an all-optical feedback in Fig. 9(b) in which the electronics in the loop was replaced by a semiconductor optical amplifier and a fiber coupler (Duport, Schneider, Smerieri, Haelterman, & Massar, 2012) or a fiber amplifier, semiconductor saturable absorber, and fiber coupler (Dejonckheere et al., 2014) have also been demonstrated experimentally.

A critical limiting factor for reservoir prototypes is the impact of noise on performance. This issue was investigated numerically and experimentally, and the importance of a preprocessing mask weight with multiple values, as opposed to binary values, was demonstrated (Soriano, Ortín, Brunner et al., 2013). It was also shown that when an optoelectronic loop setup was used for a reservoir computer, the system could perform learning tasks even if the feedback was removed to obtain an extreme learning machine configuration if the response was slower than the masked input steps to provide some interaction between neighboring neurons (Ortín et al., 2015). The system without feedback was found to be capable of performing a cancer type classification task with genetic microarray data.

A chip-integrated device that combines physical optical nodes and time memory functions by connecting in series several ring resonators with a time-encoded input optical signal was also described (Zhang et al., 2014) and is included here for completeness even though the feedback is not amplified. Performance for signal classification and chaotic time series prediction tasks was numerically investigated. In another scheme, a reservoir with a coherently-driven passive cavity was reported, in which the low optical loss led to good operation (Vinckier et al., 2015). State-of-the-art results were obtained numerically and experimentally for a spoken digit recognition task. A parallel reservoir computer based on a fiber loop with frequency multiplexing of neurons was also reported (Akrout et al., 2016), where the realization of nonlinear channel equalization and isolated spoken digit recognition was simulated.

Another example of added functionality is the simultaneous computation of three independent tasks in a time-delay optical system with three so-called virtual reservoirs, as the system typically has an inherent bandwidth higher than the input signal rate (Duport, Smerieri, Akrout, Haelterman, & Massar, 2016b). To avoid electronic preprocessing of the input signal into a step waveform as an input mask, an implementation of a fully analog reservoir with time delay was also proposed and only slight performance degradation compared to a step signal was observed (Duport, Smerieri, Akrout, Haelterman, & Massar, 2016a). At the output layer, the signal was split between a readout photodiode and a modulated portion applying the readout coefficients online. In another publication, the potential of an optoelectronic feedback reservoir computer for high speed processing was clearly demonstrated in the realization of a setup capable of performing a million-words-per-second classification task using fast electronics for input and output processing (Larger et al., 2017). Phase modulators were used in the feedback loop, and the delay between the virtual nodes corresponded to an input signal rate of approximately 17 GHz.

An interesting application was shown in Qin, Zhao, Yin, Jin, and Liu (2017), namely the identification and classification of a packet header for switching in an optical network application. The low hardware requirements compared to a traditional neural network were emphasized. This concept was further extended to a system with two feedback loops to perform simultaneous recognition of packet headers for two optical channels (Zhao, Yin, & Zhu, 2018). An optoelectronic reservoir computer was also applied to the task of generating a long-range periodic time series and the emulation of a chaotic system (Antonik, 2018; Antonik, Haelterman and Massar, 2017; Antonik, Hermans, Haelterman and Massar, 2017). The underlying concept of this work was to feed the signal output processed by an FPGA back into the reservoir, in addition to the usual feedback signal. Periodic frequency and random patterns were successfully generated.

Some recently published articles have reported the potential of novel schemes for online training. A new method based on gradient descent training with BPTT was recently proposed and experimentally shown with an optoelectronic feedback loop (Hermans, Antonik, Haelterman, & Massar, 2016; Hermans, Dambre and Bienstman, 2015; Hermans, Soriano, Dambre, Bienstman and Fischer, 2015). The scheme cannot be strictly labeled as a reservoir, because the input weights are optimized to enhance performance, but may have a significant impact on the development of future related systems. Results obtained for the TIMIT phoneme recognition task were similar to those reported for a fully offline trained RC with a larger number of nodes. By connecting an FPGA performing a gradient descent algorithm at the loop output, a device with a very low symbol error rate was also demonstrated for a distorted wireless signal recovery task, particularly for a high signal-to-noise ratio environment (Antonik, Duport et al., 2017).

A multimode polymer waveguide-based design in which coupling and propagation losses are small has recently been proposed (Héroux, Numata, Kanazawa, & Nakano, 2018; Héroux & Nakano, 2017). The feedback is all-optical and the nodes are optoelectronic, composed of photodiode, amplifier, and vertical cavity surface emitting laser (VCSEL) chip arrays, as shown in Fig. 9(c). Simulations with this reservoir yielded promising results for signal recovery and nonlinear time series prediction tasks. As the gain is located in the neurons, there is no fundamental limitation on the aggregation of a large number of physical nodes in a future device. Moreover, as the optical path is on the centimeter scale with low loss and signals can cross each other, there is added flexibility, compared to a chip-integrated device, in the implementation of multiple relatively long delay feedbacks on a solid substrate in a compact format.

### 5.2.2. *Optical feedback in a laser cavity*

A second class of time-delayed optical reservoir computers has been investigated in recent years, in which the nonlinearity comes not from a discrete element in the loop but from the nonlinear response of a laser when a delayed signal re-enters its cavity. Light input and output with the same aperture is achieved by polarization rotation with a circulator component or another suitable mechanism. In the first reports (Brunner, Soriano, Mirasso and Fischer, 2013; Hicke et al., 2013), good results were experimentally obtained with a semiconductor laser for spoken digit recognition and chaotic time series prediction tasks. In another study that is related but not strictly categorized as a reservoir as there was no feedback component, a similar system based on a nonlinear resonant cavity was used to demonstrate optical vector–matrix product operations (Brunner, Soriano and Fischer, 2013).

In an alternative configuration, a semiconductor ring laser was implemented as the nonlinear node, with the potential for on-chip integration and simultaneous processing of two tasks by modes with opposite propagation directions. Numerical studies have shown good results for chaotic time series prediction and nonlinear channel equalization tasks (Nguimdo, Verschaffelt, Danckaert, & der Sande, 2015). Delayed systems are sensitive to phase change, but in a subsequent study, it was explained that this problem can be alleviated with a modified readout layer (Nguimdo, Verschaffelt, Danckaert, & der Sande, 2016).

In another related setup, information was sent into the laser cavity via phase modulation, and a partially transparent mirror split the laser output into a delayed component going back into the cavity and a transmitted component for readout (Nakayama, Kanno, & Uchida, 2016), as shown schematically in Fig. 9(d). The efficiency of a chaotic input mask generated from a separate optical cavity subjected to feedback was compared to other digital and analog masking schemes and numerically studied for a chaotic time series prediction task. Performance improvement was observed, provided that the mask frequency was near the relaxation oscillation of the ring cavity laser. It was experimentally confirmed that a chaotic or colored-noise mask gives better results with a properly selected cut-off frequency (Kuriki, Nakayama, Takano, & Uchida, 2018). In another study, numerical simulation of a reservoir in which the input signal was applied to a tunable Bragg reflector providing external feedback to the laser cavity was proposed, and the reservoir was applied to a waveform classification task (Takeda, Nakano, Yamane and Tanaka et al., 2016).

Another application was described in Qin et al. (2016), where a delayed optical feedback setup with a polarization circulator was adopted for optical packet header identification at 10 GB/s. The optimal feedback parameters were found and a low recognition error was obtained provided that the signal-to-noise ratio was above 15 dB. Calculations with a ring laser system for dual-channel packet header recognition have also been reported (Bao, Zhao, Yin, & Qin, 2018).

The optimal operation conditions for an all-optical feedback loop system were experimentally investigated in Bueno, Brunner, Soriano, and Fischer (2017). The effects of detuning of the frequency between an injection laser and the reservoir laser as well as the locking of the laser state were studied for a chaotic time series prediction task. Moreover, very recently a system with two optical feedback loops was studied numerically for a time series prediction task with an information processing rate of 1 GB/s (Hou et al., 2018). Performance was improved over a single loop configuration owing to the rich dynamics.

## 6. Spintronic RC

Several reservoirs based on spin electronics (spintronics) have been proposed. Spintronics is an emerging research field of nanoscale electronics involving both charge and spin of electrons for developing new electronic devices, such as non-volatile storage (Wolf et al., 2001). Spin systems are potential candidates for low-power and small-scale reservoir devices. Three types of spin-based reservoirs with spin oscillations, spin waves, and skyrmions are summarized.

A reservoir with a spin torque oscillator (STO) was experimentally demonstrated (Torrejon et al., 2017). The spin torque oscillator was fabricated with a magnetic tunnel junction (MTJ) element composed of two ferromagnets (the top free layer and the bottom pinned layer) separated by a thin insulator, as shown in Fig. 10(a). When a constant DC current is injected into an MTJ, the spin direction in the free layer rotates owing to the spin torque that originates from the spin-polarized electron current generated by the pinned layer. After a transient time, the oscillation is stabilized at a frequency depending on the magnitude of the input current. The reservoir uses the nonlinear relationship between the input current and the oscillation frequency as well as history-dependent transient motions of the spin in the free layer. In a spoken digit recognition task, the sound data was given to the single STO as an input current after a preprocessing step and the voltage output of the STO was used to train the readout. It was shown that the digit prediction performance of the proposed method was better than that in the case without STO. The possibility of using chaotic dynamics in an STO with time-delayed feedback for RC has been explored in Williame, Accioly, and Kim (2017). By adjusting the amplitude and delay time of the feedback current, good conditions for spin motions in RC can be specified. Another study numerically investigated the effect of memory and nonlinearity of STO-based reservoirs with single and multiple MTJs in a short-memory task and a parity check task (Furuta et al., 2018). A random binary voltage was used as an input and the time-varying resistance of the MTJ device was used as an output. The spin dynamics in the free layer follows the Landau–Lifshitz–Gilbert (LLG) equation (Lakshmanan, 2011). It was demonstrated that the performance in the two tasks depends on the duration of the input pulse voltage.

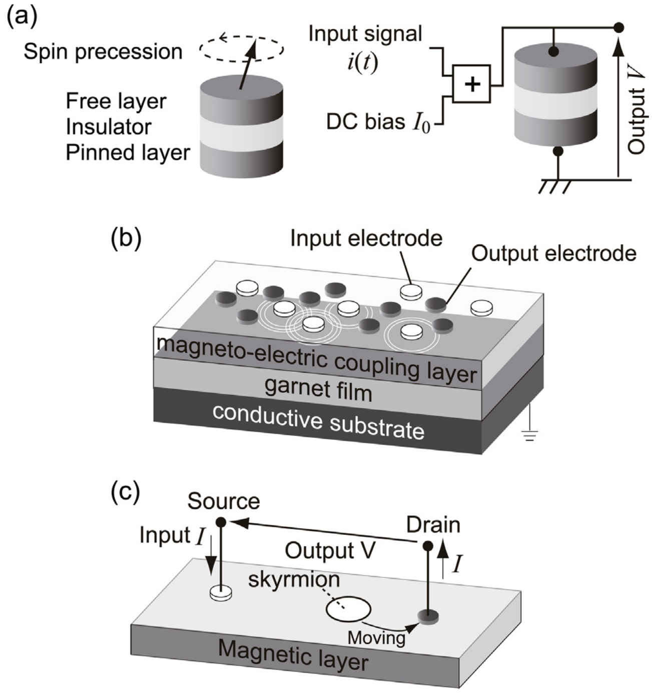


**Fig. 10.** (a) Schematic illustration of the spin torque oscillator (left) and the reservoir device structure with a bias current input and a voltage output (right) (Torrejon et al., 2017). (b) Schematic illustration of a spin-wave-based reservoir device (Nakane, Tanaka, & Hirose, 2018). Figure reproduced with permission from IEEE. (c) A possible structure of a skyrmion-based reservoir device (Prychynenko et al., 2018).

Spatial propagation of spin motions, or spin waves, can also be used for spintronic reservoir (Nakane et al., 2018). The proposed reservoir device structure consists of a thin Yttrium iron garnet (YIG) film between a magneto-electric coupling layer and a conductive substrate, as shown in Fig. 10(b). The input voltage signals applied at the input electrodes on the top layer cause changes in the uniaxial magnetic anisotropy constant of the middle garnet layer due to the magneto-electric effect. This stimulation disturbs the alignment of spins in the same direction; thus, the phase difference between neighboring spins is transmitted through a material as spin waves. The spin motions in a three-dimensional space were simulated by solving the LLG equation. The spin waves can show input history-dependent transient motions and nonlinear interferences (Stancil & Prabhakar, 2009). Spin motions measured at the output electrodes were used to train a regression model in the readout. Numerical demonstration showed that a characteristic of the input pattern can be well estimated by the spin-wave-based RC system when the electrode positions and the duration of the spin motions used for computation are appropriately selected.

Another type of spin-based reservoir device could be realized using a magnetic skyrmion, which is a nano-scaled magnetic

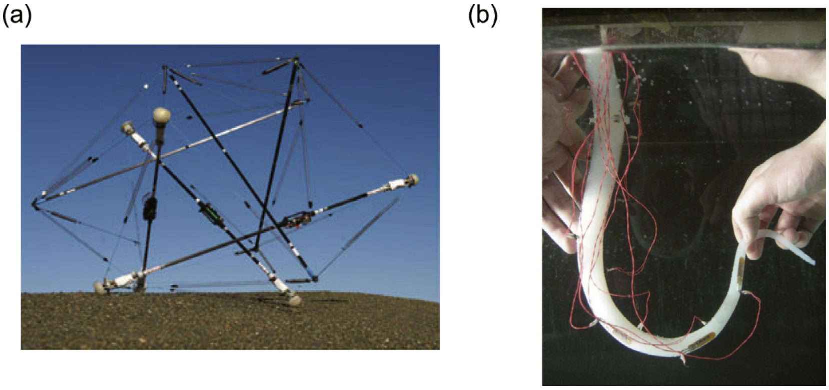


**Fig. 11.** (a) A tensegrity structure-based reservoir (Caluwaerts et al., 2014). Figure reproduced with permission from NASA Ames Research Center. (b) A reservoir based on a soft octopus robot (Nakajima, Hauser, Li, & Pfeifer, 2015). Figure reproduced from (Nakajima et al., 2015), licensed under CC-BY 4.0.

vortex in a magnetic material, as shown in Fig. 10(c) (Prychynenko et al., 2018). It is known that a current-induced transfer of a skyrmion can show nonlinear dynamics and history-dependent spin responses, which are favorable for a reservoir. In the proposed device, the input is an electron current at the source and the output is the voltage between the source and the drain. By injection of a constant current, a skyrmion initially present between the source and the drain moves toward the drain and is then annihilated at the drain. The output voltage changes with the distance between the skyrmion position and the drain, in response to the input current. Interacting multiple skyrmions can generate more complex behavior, and skyrmion fabrics are potentially attractive options for implementing a reservoir (Bourianoff, Pinna, Sitte, & Everschor-Sitte, 2018).

## 7. Mechanical RC

Mechanical systems, such as soft and compliant robots, are possible options for physical reservoirs. Soft and compliant robots with flexible bodies are difficult to control due to their complex body dynamics compared with rigid robots with stiff bodies. However, such complex behavior can be favorably leveraged to generate rich nonlinear dynamics required for RC. The idea of outsourcing computation to a physical body is known as *morphological computing* in the field of robotics (Pfeifer & Bongard, 2006).

Soft and compliant robots are typically composed of deformable bodies. A primitive model available as a physical reservoir is a mass–spring network, which can be regarded as coupled mechanical oscillators as described in Section 3.3. A mass–spring network reservoir where mass points are randomly connected to neighboring mass points via nonlinear springs was proposed in Hauser, Ijspeert, Füchslin, and Maass (2011). The motion of each nonlinear spring can be described as follows:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2, \tag{10}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = -p(x_1) - q(x_2) + u, \tag{11}$$

where $t$ represents continuous time, $x_1$ is the displacement of the spring from its rest length, $x_2$ is the velocity of the spring motion, $p$ and $q$ are nonlinear functions representing the properties of the spring, and $u$ is the sum of external forces acting on the spring. The input signal is given to some randomly chosen nodes as the external force, inducing nonlinear responses of the mass–spring network. The output signal is obtained as a linear combination of the actual lengths of the springs. Simulations demonstrated the computing power of RC based on the mass–spring network in time series approximation and robot arm tasks. By adding feedback loops from the output, the reservoir of a mass–spring network can be applied to pattern generation tasks, which are useful for producing locomotion of robots and biological organisms (Hauser et al., 2012). In another extensive study, a reservoir model of a mass–spring–damper network was numerically investigated to clarify the link between the property of the mechanical reservoir and its computational ability in locomotion learning (Urbain, Degrave, Carette, Dambre, & Wyffels, 2017).

By replacing point masses in a mass–spring network by stiff bars, a tensegrity-like structure is obtained. Tensegrity (tension integrity) indicates a structural principle that uses isolated compression elements loaded in a continuous network of tension elements, leading to a physical structure combining strength and flexibility. The compliant body of a tensegrity robot was exploited as a physical reservoir and successfully applied to stable gait pattern generation and terrain pattern classification in numerical experiments (Caluwaerts & Schrauwen, 2011). Moreover, how to effectively implement control in physical reservoirs with tensegrity structure was investigated by considering various learning rules including a reward-modulated Hebbian learning rule (Burms, Caluwaerts, & Dambre, 2015; Caluwaerts, D'Haene, Verstraeten, & Schrauwen, 2013). A physical prototype of a reservoir compliant tensegrity robot (ReCTeR, Fig. 11(a)) was developed for creating planetary rovers (Caluwaerts et al., 2014). The highly compliant body of ReCTeR consists of 24 passive spring-cable assemblies and 6 actuated spring-cable assemblies connecting non-parallel struts. The reservoir states obtained from the sensors attached to the struts were used to control the robot in order to approximate desired signals under various control strategies. In another study, better conditions for a tensegrity-based reservoir computer were explored (Fujita, Yonekura, Nishikawa, Niiyama, & Kuniyoshi, 2018).

Another type of physical body is a muscular hydrostat system inspired by octopus limbs, as shown in Fig. 11(b) (Nakajima et al., 2013). This is an extreme embodiment of a soft robotic arm without a skeleton, involving virtually unlimited degrees of freedom. Even though the hydrodynamic effect yields highly complex and time-varying dynamics of the soft robotic arm, its motion was successfully learned by an ESN-based controller in a simulation study (Kuwabara et al., 2012) and in an experimental study using a real robot made of silicone rubber (Li, Nakajima, Cianchetti, Laschi, & Pfeifer, 2012). These studies suggested that the complex dynamics of the soft body can be used for RC. The computational capability of the soft body was demonstrated in nonlinear system approximations and body dynamics control without an external controller (Nakajima et al., 2015; Nakajima, Li, Hauser, & Pfeifer, 2014).

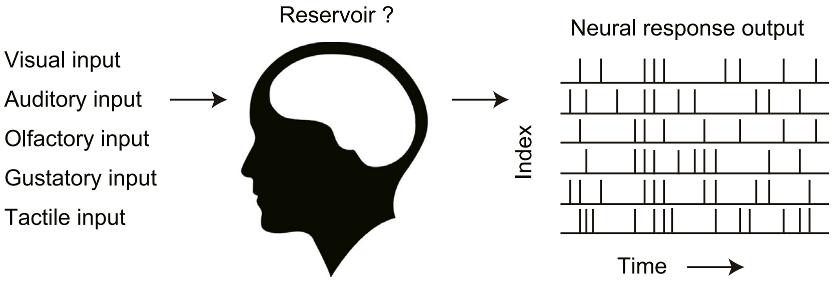


**Fig. 12.** A possibility that some brain regions work as reservoirs.

Other attempts to control real robots and generate different robot behaviors, based on the concept of RC, can be found in studies on a pneumatically driven soft robot arm (Eder, Hisch, & Hauser, 2017), a spine-driven quadruped robot (Zhao, Nakajima, Sumioka, Hauser, & Pfeifer, 2013), a dog-like quadruped robot (Wyffels et al., 2010), and a much less compliant quadrupedal robot (Degrave, Caluwaerts, Dambre, & Wyffels, 2015).

## 8. Biological RC

The mechanism behind the computing ability of the brain is one of the mysteries in neuroscience. Considerable efforts have been made to explain the brain computing function using an analogy to computational models in ANNs and machine learning. In attempts to understand the relationship between temporal information processing in the brain and RC, researchers have speculated about which part of the brain can be regarded as a reservoir or a readout as well as about how subnetworks of the brain work in the RC framework. On the other hand, physical RC based on *in vitro* biological components has been proposed to investigate the computational capability of biological systems in laboratory experiments. We review recent studies on RC hypotheses in brain regions (Section 8.1) and RC using *in vitro* cultured cells (Section 8.2).

### *8.1. Brain regions*

A specific case of RC principle can be found in a series of neurocognitive studies on cortico-striatal models for context-dependent sequential information processing (Dominey, 1995, 2013; Dominey et al., 1995). A more general computational framework was later proposed as the LSM, motivated by the real-time information processing of time-varying input streams in cortical microcircuits (Maass & Markram, 2004; Maass et al., 2002) as described in Section 2.1. Subsequently, researchers have debated on whether reservoir computation is actually conducted in the brain and which brain regions can be interpreted to use this computational framework (Fig. 12). It is suggested that RC is one of the general frameworks for state-dependent computation in cortical networks, emerging from the interaction between the incoming stimulation and the internal dynamics in RNNs (Buonomano & Maass, 2009). Some researchers have hypothesized that the spatiotemporal information processing in cortical and subcortical networks can be interpreted as RC through neurophysiological experiments and/or computational model analyses, as described below.

First, we summarize RC-related studies on cortical regions including prefrontal and visual areas. The prefrontal cortex is a front part of the cerebral cortex, which is associated with the planning of cognitive behavior, personality expression, decision making, and moderating social behavior (Fuster, 2015). An early RC-type model of the cortico-striatal system was proposed to understand the mechanism of context-dependent oculomotor (eye movement) behavior (Dominey, 1995; Dominey et al., 1995). The model consists of the prefrontal cortex neurons with fixed recurrent connections ("reservoir") and modifiable connections from the prefrontal cortex neurons to neurons in the striatum ("readout"). The neural activity in the cortical network, responding to sequential visual inputs, is associated with the outputs that represent the corresponding oculomotor movements through reinforcement learning. An extended work combined the cortico-striatal model (called a temporal recurrent network) for learning serial and temporal structure of sequential inputs with an additional abstract recurrent network which has a short term memory for encoding an abstract structure (or a hidden rule) of sequential inputs (Dominey & Ramus, 2000). The cortico-striatal model was further combined with neurophysiological models of language processing to learn grammatical constructions in sentence processing (Dominey & Inui, 2009; Dominey, Inui, & Hoen, 2009) and with an RC scheme to improve the performance of the learning algorithm (Hinaut & Dominey, 2013). A detailed history of the cortico-striatal model and its developments is summarized in a review paper (Dominey, 2013). On the other hand, the RC properties in monkey prefrontal cortex were investigated with both model simulation and neurophysiological experiments to explore how contexts of sensory inputs are represented in cortical dynamics (Enel, Procyk, Quilodran, & Dominey, 2016). A reservoir of model neurons was compared with the monkey prefrontal cortex in terms of their representational and dynamical properties during a complex cognitive task. Then, it was shown that the reservoir of randomly connected RNNs can obtain a dynamic form of mixed selectivity (Rigotti et al., 2013) and thus perform a complex context-dependent cognitive task as in cortical neurons. This result suggests that RC can be a model of cognitive processing in local generic prefrontal cortical networks. A discussion on the computational role of the thalamus (Dehghani & Wimmer, 2018) also highlights the close relationship between the computational properties of the prefrontal cortex and the main features of RC. It is suggested that the context-dependent computing in the prefrontal cortex relies on the task-dependent modulation of the cortical reservoir by thalamic functions.

The visual cortex is a part of the cerebral cortex that processes visual information. The visual cortex located in the occipital lobe is divided into the primary visual cortex (early visual areas) and visual association cortex (higher-order visual areas) (Grill-Spector & Malach, 2004). An experimental study showed evidence that early visual areas have fading memory properties demanded by LSMs (Nikolić, Häusler, Singer, & Maass, 2009). In this experiment, *in vivo* data of neuronal spiking activities in cat primary visual cortex for different visual stimulus were obtained as a reservoir state and they were classified by a simple linear classifier emulating linear integrate-and-fire readout neurons. The new perspective that the primary visual cortex can perform time-dependent computation for sequential inputs based on memory mechanisms is in contrast to the conventional one that it executes frame-by-frame computation based on memory-less hierarchically organized feedforward architectures (Serre et al., 2005). The limitation of the conventional viewpoint is that the frame-by-frame computation cannot respond to changes in the context and environment behind visual inputs.

Second, we move on to studies on RC in subcortical structures including the cerebellum and the basal ganglia. In terms of learning styles, the cerebellum and the basal ganglia can be characterized by supervised learning and reinforcement learning, respectively, which are complementary to each other (Doya, 2000). The cerebellum is mainly involved in the coordination of body movement and is responsible for the learning of motor and non-motor skills (Rapoport, van Reekum, & Mayberg, 2000). The classical hypothesis is that the cerebellum is a learning device, such as a simple perceptron (Albus, 1971; Marr, 1969). Another viewpoint from a model-based study is that the functional role of cerebellar circuits is interpreted as an LSM (Yamazaki & Tanaka, 2007). In the proposed model, the granular layer constructed with a recurrent inhibitory circuit is regarded as a reservoir that receives inputs through mossy fibers, and the Purkinje cells are regarded as readout neurons. Numerical studies showed that the cerebellar circuit model successfully learned Boolean functions. This LSM hypothesis highlights an indirect functional pathway from the precerebellar nucleus to the cerebellar nucleus through the LSM model consisting of the granular layer and the Purkinje cells, in addition to the direct pathway from the precerebellar nucleus to the cerebellar nucleus, which is regarded as a simple perceptron as in the classical hypothesis. Further exploration in this direction has made it possible to incorporate recent anatomical and physiological findings on cerebellar microcircuits into computational models (D'Angelo et al., 2016).

The striatum is the primary input module of the basal ganglia and a critical component of the reward system. It plays a significant role not only in motor control and planning but also in reward-modulated decision making. However, it is not obvious how the environmental states are represented in the reinforcement learning of the basal ganglia. Based on the hypothesis that the striatum responds to diverse inputs from different cortical sources and plays a computational role for discriminating inputs in reinforcement learning-based decision making, the LSM properties of a striatal microcircuit were studied with computational models (Toledo-Suárez & Morrison, 2014). It was demonstrated that the separation and approximation properties required for the LSM were generated using a model network of medium spiny neurons and fast spiking interneurons coupled via inhibitory synapses in a supervised learning task.

Third, we focus on the discussion about the relationship between RC and working memory. Working memory is a cognitive process that temporarily stores and manages information for carrying out complex cognitive tasks (Baddeley, 2003; Daneman & Carpenter, 1980) and is believed to be involved in multiple cortical and subcortical regions (Dutta, Shah, Silvanto, & Soto, 2014; Eriksson, Vogel, Lansner, Bergström, & Nyberg, 2015). However, elucidating the mechanism for task-dependent switching of the role of neural circuitry remains an unresolved issue. Multiple working memory models based on RC and their variants have been proposed, including the working memory model with generic cortical microcircuit models with feedback (Maass, Joshi, & Sontag, 2007), the ESN-based working memory model (Pascanu & Jaeger, 2011), the reservoir model for storing different time constants of memory traces for reward expectation in reinforcement learning (Bernacchia, Seo, Lee, & Wang, 2011), the autonomously emerged working memory model through reward-modulated online learning rule that enables the same neural circuit to solve different tasks (Hoerzer, Legenstein, & Maass, 2012), and the comparative analysis of three working memory models including the RC-based one (Barak, Sussillo, Romo, Tsodyks, & Abbott, 2013).

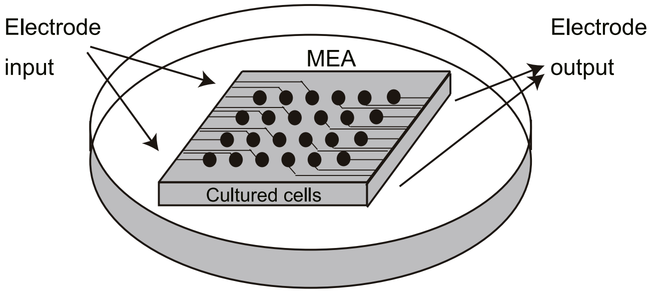


**Fig. 13.** A reservoir of *in-vitro* cell cultures.

As described above, the biological plausibility of RC in the brain regions has been examined in many studies, but it requires further investigation. To address this issue, the RC models should be evaluated from multiple aspects of structural, dynamical, and functional properties of cognitive systems. For instance, an investigation focused on the robustness and resilience of reservoir models against structural perturbations in cortical circuits (Vincent-Lamarre, Lajoie, & Thivierge, 2016). Examination of RC from the neuroscience viewpoint is expected to be useful for its applications in brain machine interfacing, disease care, and robot control.

### 8.2. *in-vitro cultured cells*

A model system to gain insights into the spatiotemporal information processing *in vivo* is cultured biological components *in vitro*.

RC with biological neurons has been demonstrated to investigate the computational function of an assembly of biological components. Microelectrode arrays (MEAs) are widely used to electrically stimulate neuronal cultures and measure their responses as shown in Fig. 13 (Obien, Deligkaris, Bullmann, Bakkum, & Frey, 2015). An RC system that combines a reservoir based on cultured cells on MEAs and a computer-aided readout is regarded as a hybrid biological-silicon computer. An early study demonstrated an LSM using *in vitro* rat cortical neurons plated on MEAs Hafizovic et al. (2007). Two stimulation patterns with rectangular voltage pulses were given to the electrodes for generating the action potentials of the cultured neurons. The spatiotemporal patterns of spike events recorded with MEAs were transformed into a time-continuous reservoir state by a leaky integrator. In the readout on an external computer, a support vector machine (SVM) was employed for classification of the stimulation patterns. We note that this readout is not a linear classifier and thus the effectiveness of this reservoir remains undetermined. This work was followed by similar experiments. In one such study, the classification ability was investigated with a larger number of stimulation patterns and SVM classifiers, and the tradeoff between the number of patterns and the separability was evaluated (Ortman, Venayagamoorthy, & Potter, 2011). In Dockendorf, Park, He, Príncipe, and DeMarse (2009), two types of electrical stimulation protocols were used. One is the low-frequency stimulation inducing bursts and the other is the high-frequency stimulation suppressing the burst response. Living cortical networks react to these stimulus patterns differently, generating distinct spatiotemporal spike trains. The goal is to reconstruct the input spatial stimulation pattern from the temporal structures of the spike trains. The results show that the input reconstruction is successful if an appropriate window size is chosen, suggesting the separation property of the biological cell networks. In another study, input coding and output decoding methods for LSMs with cultured neurons were considered (George et al., 2014). The proposed input coding method enabled to generate a large number of input patterns from stimulations through a small number of electrodes. The spatiotemporal spike patterns in the reservoir of cultured cells were transformed into spatial patterns such that a linear classifier works well in the readout.

Instead of electrical stimuli, an optogenetic approach for controlling neuronal activities more precisely has been used in RC experiments. Optical stimuli using random dot patterns were projected onto rat cortical neurons cultured on MEAs as a reservoir (Dranias, Ju, Rajaram, & VanDongen, 2013). Through experiments on short-term memory for optical inputs, it was shown that the results closely matched those of corresponding *in vivo* experiments in Nikolić et al. (2009). This suggests that the mechanism of state-dependent information processing *in vivo* can be studied using cultured neuronal networks. Based on a similar experimental setting, more complex tasks in the time domain were tested (Ju, Dranias, Banumurthy, & VanDongen, 2015). In classification tasks of spike template patterns and musical styles, high classification performance was achieved if appropriate MEA channels were selected before the readout process and the spatiotemporal memory processes lasted for several seconds.

In addition to the input separability, the control of biological reservoirs has received considerable attention in terms of robot control. In this case, the reservoir receives feedback inputs from the output. The FORCE learning (Sussillo & Abbott, 2009) was used for a closed loop reservoir with cultured neurons to control the output at a target constant value (Takahashi, Yasuda, Yada, & Kanzaki, 2016). This technique was adopted to control a mobile robot, by which the robot can avoid obstacles and traverse a maze. An application of the concept of RC to robot control is discussed in Aaser et al. (2017).

Finally, reservoirs based on other living organisms are discussed. A reservoir with bacterium *Escherichia coli* (*E. coli*) reacts to different chemical inputs and generate complex temporal patterns (Jones, Stekel, Rowe, & Fernando, 2007). A population of *E. coli* was placed in a flask and exposed to sequential inputs that indicate time-varying combinatorial conditions of chemical environments and temperatures. Then, samples of *E. coli* were moved to a microarray to measure the mRNA and protein concentrations, representing the state of the gene regulatory network. Perceptron learning was used in the readout to classify the stimuli into the appropriate classes. Numerical simulations corresponding to the above experimental design confirmed that the proposed RC was partially successful in an XOR problem and showed good performance in a separation problem. This study motivated a corresponding experiment in a wet lab. In Didovyk et al. (2014), genetically engineered bacterial cell cultures were used for realizing a variety of simple classifiers separating different chemical inputs, which are aggregated to form a more complex classifier. This conceptual design of machine learning in synthetic biology can be extended to a reservoir computer.

## 9. Others

There are many other candidates for physical reservoirs. Attempts to exploit novel physical reservoirs for RC can contribute to exploration of unconventional computing methods (Hadaeghi et al., 2017). We introduce two types of physical reservoirs that are not categorized into the aforementioned types.

The first one is RC realized with nano-scale materials and substrates that exhibit stimulation-dependent changes. An example is the atomic switch network introduced in Section 4.4.2. Another proposal is a nano-scale reservoir with quantum dots and chemical compounds that change their absorption spectrum depending on the pH or redox potential in their environment (Obst et al., 2013). An input signal is given as a change in chemical properties of the compounds, which affect the signal transfer between quantum dots randomly dispersed in a space, encoded as an emission pattern. Simulations confirmed the potential computational performance in an image recognition task. In another study, reservoirs were configured with physical substrates consisting of carbon nanotubes and polymer mixtures (Dale, Miller, Stepney and Trefzer, 2016; Dale, Stepney, Miller and Trefzer, 2016). A computer-aided evolutionary algorithm determines the control voltage signals and the locations of input/output electrodes for finding an optimal configuration of the reservoir. Experimental results showed that the material-based RC with the optimized reservoir configuration was successful in time series prediction benchmark tasks.

Another recently proposed idea is to use complex quantum dynamics for RC (Fujii & Nakajima, 2017). A quantum system consists of multiple qubits (or quantum bit, indicating the minimum unit of information in quantum computing). For $N$ qubits, there are $2^N$ basis states for a pure quantum state. In quantum RC, each individual node of a quantum reservoir is defined by the system's basis states, not by qubits; therefore, a large number of hidden nodes can be implicitly implemented behind true nodes that are monitored by ensemble measurements. In response to an input signal injected into the first qubit, the quantum system temporally evolves under a Hamiltonian. Then, the states of the qubits obtained by ensemble average measurements are transformed with time multiplexing into a set of signals that are linearly combined to yield an output signal in the readout. High computational performance of the quantum RC was demonstrated in benchmark tasks. In a subsequent study (Nakajima, Fujii, Negoro, Mitarai, & Kitagawa, 2018), spatial multiplexing was proposed to enhance the computational power of quantum RC by constructing a system of multiple decoupled quantum reservoirs. Physical implementation of a quantum reservoir system is a challenge for the future.

## 10. Conclusion and outlook

This review has summarized recent trends in RC, with special focus on physical RC. Physical reservoirs have been categorized based on the type of dynamical system (Section 3) and physical phenomenon (Sections 4–9). Most physical reservoirs are designed and configured to satisfy multiple requirements, such as high-dimensionality, nonlinearity, input history-dependent response, fading memory property (echo state property), and/or separation property (Section 2.2). However, dynamical behavior produced by different types of physical reservoirs exhibits high diversity in terms of the degree of nonlinearity, transient response time, signal transmission speed, and spatial dimension. This diversity makes physical reservoirs appropriate for various tasks and data.

Physical reservoirs are classified into several types according to the architecture, including network-type reservoir, single-node reservoir with time-delayed feedback, and excitable continuous medium reservoir. They are characterized as follows:

- Network-type reservoirs consist of interacting nonlinear elements, such as artificial neurons (Section 2.1), oscillators (Section 3.3), standard circuit elements (Sections 4.2 and 4.3), memristive elements (Section 4.4), optical nodes (Section 5.1), springs (Section 7), biological cells (Section 8.2). They can be simply scaled up by increasing the number of network elements for higher dimensionality. However, realization of a large-scale reservoir requires sophisticated technology to implement massive recurrent interconnections.

- Single nonlinear node reservoirs with time-delayed feedback generate input-dependent patterns in a high-dimensional space using virtual nodes on a delay loop (Section 3.1). Such reservoirs have been intensively studied for electronic RC (Sections 4.1 and 4.2) and photonic RC (Section 5.2). These reservoirs can avoid the problem of massive interconnections and thus are more hardware friendly. However, designing and implementing an appropriate delayed feedback loop is not a straightforward task.
- Excitable continuous medium reservoirs use propagation of waves triggered by stimulation inputs. Propagation phenomena are widely observed in fluids (Section 2.2), cellular automata (Section 3.2), magnetic materials (Section 6), and elastic media (Section 7). They have the potential to realize extremely efficient physical RC by harnessing rich physical properties of waves, such as interference, resonance, and synchronization (Katayama et al., 2016). However, their computational power has not been fully understood thus far.

There are general issues to be addressed in physical RC. First, appropriate preprocessing is necessary for maximizing the computational power of each physical RC system. In fact, some studies have empirically demonstrated that adequate information transformation of input data is necessary for satisfactory performance. In addition, temporal and spatial scaling of input data critically affects the computational performance as the dynamic range is limited in physical reservoirs. Second, each physical reservoir needs to be optimized by the selection of the best-suited material or substrate as well as the tuning of hyper-parameters, such as the shape and size of the reservoir. Mathematical modeling and analysis of a physical reservoir, under physical constraints and practical conditions, are useful for determining the reservoir settings systematically. It is also practically important to ensure scalability in a single reservoir as well as in combined multiple reservoirs. Third, a training algorithm in the readout should be suitably chosen to be compatible with the physical property of RC. Even if a physical reservoir has high signal processing speed, the total computation speed of the entire physical RC system for real-time information processing is limited by the training speed in the readout. One possible solution to this problem is to use a physically realized readout instead of a software-based one.

Moreover, evaluation of computational performance, processing speed, memory, power efficiency, and scalability of physical RC systems is necessary for comparison with other relevant methods including current digital computers. The computational performance is evaluated with a task-dependent measure, such as classification accuracy in classification tasks and prediction error in prediction tasks. The measures called kernel quality and generalization ability (Büsing, Schrauwen, & Legenstein, 2010; Legenstein & Maass, 2007) are also useful for evaluating the effectiveness of a physical reservoir. The processing speed is measured by how much information is processed in a unit time for each task. For instance, in speech recognition tasks, the number of words processed per second under a very low word error rate condition is a measure for processing speed (Larger et al., 2017). The memory in reservoirs means the length of time during which the past input information is kept in the reservoir dynamics. The memory capacity is a standard measure for evaluating the memory performance (Jaeger, 2002a). The power efficiency is evaluated by power consumption required for performing a specific task. For instance, in a study on photonic reservoir (Vinckier et al., 2015), the total optical power injected into a passive optical cavity reservoir is considered for assessing its power efficiency aside from power consumption for the other hardware components in preprocessing steps and the readout. The scalability of physical reservoirs is determined by how much the reservoir components can be miniaturized and whether they can be efficiently implemented on a small-scale chip. The evaluation of emerging RC technologies from these different aspects would highlight the advantage and disadvantage of each reservoir.

Some physical RC systems are suited for realizing next-generation machine learning hardware devices and chips with low power consumption. Such RC hardware is potentially capable of high-speed online computation for dynamic data, in contrast to relatively expensive computation in deep learning hardware for static data. This type of hardware is in great demand as edge computing devices for reducing the communication load in the IoT society, where a massive amount of data is produced and transmitted. Developments of device technology and findings of new materials for this purpose are highly expected. Other physical RC systems are useful for exploring the applicability of natural phenomena to information processing under realistic constraints in biology, chemistry, physics, and engineering. In particular, studies on biological reservoirs are expected to provide new insights into the mechanism of real-time information processing in a variety of brain regions (Section 8.1).

Physical RC still remains in an early stage of development. It seems premature to compare different physical RC technologies in terms of performance, speed, memory, power efficiency, and scalability, because these characteristics highly depend on the implementation method. It is an ongoing challenge to explore efficient implementation methods for each type of reservoir. For full-scale development of physical RC, further investigation is required from various aspects, such as performance evaluations in practical applications, developments of implementation technology, and theoretical understanding of dynamics and computational function. It is intriguing to exploit novel physical phenomena for RC and to combine physical RC with other machine learning algorithms/hardware. We expect this review to facilitate future interdisciplinary research on physical RC.

## Acknowledgments

We would like to thank Ze Hei for his support in collecting the relevant papers and the anonymous reviewers for their useful comments that improved the quality of this review. We would also like to express our gratitude to the action editor for his valuable comments and suggestions. This work was partially supported by JSPS KAKENHI, Japan Grant Number JP16H00326 (GT) and partially based on results obtained from a project subsidized by the New Energy and Industrial Technology Development Organization (NEDO), Japan.

## References

Aaser, P., Knudsen, M., Huse Ramstad, O., van de Wijdeven, R., Nichele, S., Sandvig, I., Tufte, G., Bauer, U. S., Halaas, S., Sandvig, A., & Valderhaug, V. (2017). Towards making a cyborg: a closed-loop reservoir-neuro system. In *Proc. the European conference on artificial life* (pp. 430–437). MIT Press.

Abraham, I. (2018). The case for rejecting the memristor as a fundamental circuit element. *Scientific Reports*, *8*, 10972.

Akiyama, T., & Tanaka, G. (2019). Analysis on characteristics of multi-step learning echo state networks for nonlinear time series prediction. In *Proc. International joint conference on neural networks (IJCNN)*. IEEE, (in press).

Akrout, A., Bouwens, A., Duport, F., Vinckier, Q., Haelterman, M., & Massar, S. (2016). Parallel photonic reservoir computing using frequency multiplexing of neurons, arXiv preprint arXiv:1612.08606.

Albus, J. S. (1971). A theory of cerebellar function. *Mathematical Biosciences*, *10*, 25–61.

Alomar, M. L., Canals, V., Martínez-Moll, V., & Rosselló, J. L. (2014). Low-cost hardware implementation of reservoir computers. In *Proc. 24th international workshop on power and timing modeling, optimization and simulation (PATMOS)* (pp. 1–5). IEEE.

Alomar, M. L., Canals, V., Morro, A., Oliver, A., & Rossello, J. L. (2016). Stochastic hardware implementation of liquid state machines. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 1128–1133). IEEE.

Alomar, M. L., Soriano, M. C., Escalona-Morán, V., Fischer, I., Mirasso, C. R., et al. (2015). Digital implementation of a single dynamical node reservoir computer. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *62*, 977–981.

Antonik, P. (2018). *Application of FPGA to real-time machine learning: hardware reservoir computers and software image processing*. Springer.

Antonik, P., Duport, F., Hermans, M., Smerieri, A., Haelterman, M., & Massar, S. (2017). Online training of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Transactions on Neural Networks and Learning Systems*, *28*, 2686–2698.

Antonik, P., Haelterman, M., & Massar, S. (2017). Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems. *Physical Review A*, *7*, 054014.

Antonik, P., Hermans, M., Haelterman, M., & Massar, S. (2017). Random pattern and frequency generation using a photonic reservoir computer with output feedback. *Neural Processing Letters*, 1–14.

Antonik, P., Smerieri, A., Duport, F., Haelterman, M., & Massar, S. (2015). FPGA Implementation of reservoir computing with online learning. In *Proc. the 24th Belgian-Dutch conference on machine learning*.

Appeltant, L., Van der Sande, G., Danckaert, J., & Fischer, I. (2014). Constructing optimized binary masks for reservoir computing with delay systems. *Scientific Reports*, *4*, 3629.

Appeltant, L., Soriano, M. C., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., et al. (2011). Information processing using a single dynamical node as complex system. *Nature Communications*, *2*, 468.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*, 50–58.

Baddeley, A. (2003). Working memory: looking back and looking forward. *Nature Reviews Neuroscience*, *4*, 829–839.

Bade, S. L., & Hutchings, B. L. (1994). FPGA-Based stochastic neural networks - implementation. In *Proc. IEEE Workshop on FPGAs for custom computing machines* (pp. 189–198). IEEE.

Bao, X., Zhao, Q., Yin, H., & Qin, J. (2018). Recognition of the optical packet header for two channels utilizing the parallel reservoir computing based on a semiconductor ring laser. *Modern Physics Letters B*, *32*, 1850150.

Barak, O., Sussillo, D., Romo, R., Tsodyks, M., & Abbott, L. (2013). From fixed points to chaos: three models of delayed discrimination. *Progress in Neurobiology*, *103*, 214–222.

Bennett, C. H., Querlioz, D., & Klein, J.-O. (2017). Spatio-temporal learning with arrays of analog nanosynapses. In *Proc. IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 125–130). IEEE.

Bernacchia, A., Seo, H., Lee, D., & Wang, X.-J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nature Neuroscience*, *14*, 366–372.

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*, 1413–1436.

Bourianoff, G., Pinna, D., Sitte, M., & Everschor-Sitte, K. (2018). Potential implementation of reservoir computing models based on magnetic skyrmions. *AIP Advances*, *8*, 055602.

Boyd, S., & Chua, L. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, *32*, 1150–1161.

Brunner, D., & Fischer, I. (2015). Reconfigurable semiconductor laser networks based on diffractive coupling. *Optics Letters*, *40*, 3854–3857.

Brunner, D., Penkovsky, B., Marquez, B., Jacquot, M., Fischer, I., & Larger, L. (2018). Tutorial: photonic neural networks in delay systems. *Journal of Applied Physics*, *124*, 152004.

Brunner, D., Soriano, M. C., & Fischer, I. (2013). High-speed optical vector and matrix operations using a semiconductor laser. *IEEE Photonics Technology Letters*, *25*, 1680–1683.

Brunner, D., Soriano, M. C., Mirasso, C. R., & Fischer, I. (2013). Parallel photonic information processing at Gigabyte per second data rates using transient states. *Nature Communications*, *4*, 1364.

Bueno, J., Brunner, D., Soriano, M. C., & Fischer, I. (2017). Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback. *Optics Express*, *25*, 2401–2412.

Bueno, J., Maktoobi, S., Froehly, L., Fischer, I., Jacquot, M., Larger, L., et al. (2018). Reinforcement learning in a large-scale photonic recurrent neural network. *Optica*, *5*, 756–760.

Buonomano, D. V., & Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, *10*, 113–125.

Buonomano, D. V., & Merzenich, M. M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, *267*, 1028–1030.

Bürger, J., Goudarzi, A., Stefanovic, D., & Teuscher, C. (2015b). Hierarchical composition of memristive networks for real-time computing. In *Proc. IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 33–38). IEEE.

Bürger, J., & Teuscher, C. (2013). Variation-tolerant computing with memristive reservoirs. In *Proc. the 2013 IEEE/ACM international symposium on nanoscale architectures* (pp. 1–6). IEEE Press.

Bürger J. Goudarzi, A., Stefanovic, D., & Teuscher, C. (2015a). Computational capacity and energy consumption of complex resistive switch networks. *AIMS Materials Science*, *2*, 530–545.

Burms, J., Caluwaerts, K., & Dambre, J. (2015). Reward-modulated hebbian plasticity as leverage for partially embodied control in compliant robotics. *Frontiers in Neurorobotics*, *9*, 9.

Büsing, L., Schrauwen, B., & Legenstein, R. (2010). Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation*, *22*, 1272–1311.

Caluwaerts, K., Despraz, J., Işçen, A., Sabelhaus, A. P., Bruce, J., Schrauwen, B., et al. (2014). Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of the Royal Society Interface*, *11*, 20140520.

Caluwaerts, K., D'Haene, M., Verstraeten, D., & Schrauwen, B. (2013). Locomotion without a brain: physical reservoir computing in tensegrity structures. *Artificial Life*, *19*, 35–66.

Caluwaerts, K., & Schrauwen, B. (2011). The body as a reservoir: locomotion and sensing with linear feedback. In *Proc. the 2nd international conference on morphological computation (ICMC)* (pp. 45–47).

Carbajal, J. P., Dambre, J., Hermans, M., & Schrauwen, B. (2015). Memristor models for machine learning. *Neural Computation*, *27*, 725–747.

Chang, H.-H., Song, H., Yi, Y., Zhang, J., He, H., & Liu, L. (2018). Distributive dynamic spectrum access through deep reinforcement learning: a reservoir computing based approach. *IEEE Internet of Things Journal*, http://dx.doi.org/10.1109/JIOT.2018.2872441.

Chua, L. (1971). Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, *18*, 507–519.

Chua, L. O., & Kang, S. M. (1976). Memristive devices and systems. *Proceedings of the IEEE*, *64*, 209–223.

Coulombe, J. C., York, M. C., & Sylvestre, J. (2017). Computing with networks of nonlinear mechanical oscillators. *PLoS ONE*, *12*, e0178663.

Dale, M., Miller, J. F., Stepney, S., & Trefzer, M. A. (2016). Evolving carbon nanotube reservoir computers. In *Proc. International conference on unconventional computation and natural computation* (pp. 49–61). Springer.

Dale, M., Stepney, S., Miller, J. F., & Trefzer, M. (2016). Reservoir computing in materio: an evaluation of configuration through evolution. In *Proc. IEEE symposium series on computational intelligence (SSCI)* (pp. 1–8). IEEE.

Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, *19*, 450–466.

D'Angelo, E., Antonietti, A., Casali, S., Casellato, C., Garrido, J. A., Luque, N. R., et al. (2016). Modeling the cerebellar microcircuit: new strategies for a long-standing issue. *Frontiers in Cellular Neuroscience*, *10*, 176.

Degrave, J., Caluwaerts, K., Dambre, J., & Wyffels, F. (2015). Developing an embodied gait on a compliant quadrupedal robot. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4486–4491). IEEE.

Dehghani, N., & Wimmer, R. D. (2018). A computational perspective of the role of Thalamus in cognition, arXiv preprint arXiv:1803.00997.

Dejonckheere, A., Duport, F., Smerieri, A., Fang, L., Oudar, J.-L., Haelterman, M., et al. (2014). All-optical reservoir computer based on saturation of absorption. *Optics Express*, *22*, 10868–10881.

Didovyk, A., Kanakov, O. I., Ivanchenko, M. V., Hasty, J., Huerta, R., & Tsimring, L. (2014). Distributed classifier based on genetically engineered bacterial cell cultures. *ACS Synthetic Biology*, *4*, 72–82.

Dockendorf, K. P., Park, I., He, P., Príncipe, J. C., & DeMarse, T. B. (2009). Liquid state machines and cultured cortical networks: the separation property. *Biosystems*, *95*, 90–97.

Dominey, P. F. (1995). Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, *73*, 265–274.

Dominey, P. F. (2013). Recurrent temporal networks and language acquisition – from corticostriatal neurophysiology to reservoir computing. *Frontiers in Psychology*, *4*, 500.

Dominey, P., Arbib, M., & Joseph, J.-P. (1995). A model of corticostriatal plasticity for learning oculomotor associations and sequences. *Journal of Cognitive Neuroscience*, *7*, 311–336.

Dominey, P. F., & Inui, T. (2009). Cortico-striatal function in sentence comprehension: insights from neurophysiology and modeling. *Cortex*, *45*, 1012–1018.

Dominey, P. F., Inui, T., & Hoen, M. (2009). Neural network processing of natural language: II. towards a unified model of corticostriatal function in learning sentence comprehension and non-linguistic sequencing. *Brain and Language*, *109*, 80–92.

Dominey, P. F., & Ramus, F. (2000). Neural network processing of natural language: I. sensitivity to serial, temporal and abstract structure of language in the infant. *Language and Cognitive Processes*, *15*, 87–127.

Donahue, C., Merkel, C., Saleh, Q., Dolgovs, L., Ooi, Y. K., Kudithipudi, D., et al. (2015). Design and analysis of neuromemristive echo state networks with limited-precision synapses. In *Proc. IEEE symposium on computational intelligence for security and defense applications (CISDA)* (pp. 1–6). IEEE.

Dong, J., Gigan, S., Krzakala, F., & Wainrib, G. (2018). Scaling up echo-state networks with multiple light scattering. In *Proc. IEEE statistical signal processing workshop (SSP)* (pp. 448–452). IEEE.

Doya, K. (1998). Recurrent networks: supervised learning. In *The handbook of brain theory and neural networks* (pp. 796–800). MIT Press.

Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, *10*, 732–739.

Dranias, M. R., Ju, H., Rajaram, E., & VanDongen, A. M. (2013). Short-term memory in networks of dissociated cortical neurons. *Journal of Neuroscience*, *33*, 1940–1953.

Du, C., Cai, F., Zidan, M. A., Ma, W., Lee, S. H., et al. (2017). Reservoir computing using dynamic memristors for temporal information processing. *Nature Communications*, *8*, 2204.

Duport, F., Schneider, B., Smerieri, A., Haelterman, M., & Massar, S. (2012). All-optical reservoir computing. *Optics Express*, *20*, 22783–22795.

Duport, F., Smerieri, A., Akrout, A., Haelterman, M., & Massar, S. (2016a). Fully analogue photonic reservoir computer. *Scientific Reports*, *6*, 22381.

Duport, F., Smerieri, A., Akrout, A., Haelterman, M., & Massar, S. (2016b). Virtualization of a photonic reservoir computer. *Journal of Lightwave Technology*, *34*, 2085–2091.

Dutta, A., Shah, K., Silvanto, J., & Soto, D. (2014). Neural basis of non-conscious visual working memory. *Neuroimage*, *91*, 336–343.

Eder, M., Hisch, F., & Hauser, H. (2017). Morphological computation-based control of a modular, pneumatically driven, soft robotic arm. *Advanced Robotics*, *32*, 1–11.

Enel, P., Procyk, E., Quilodran, R., & Dominey, P. F. (2016). Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS Computational Biology*, *12*, e1004967.

Eriksson, J., Vogel, E. K., Lansner, A., Bergström, F., & Nyberg, L. (2015). Neurocognitive architecture of working memory. *Neuron*, *88*, 33–46.

Fernando, C., & Sojakka, S. (2003). Pattern recognition in a bucket. In *European conference on artificial life* (pp. 588–597). Springer.

Ferreira, d. L. T., Shastri, B. J., Tait, A. N., Nahmias, M. A., & Prucnal, P. R. (2017). Progress in neuromorphic photonics. *Nanophotonics*, *6*, 577–599.

Fiers, M. A. A., Vaerenbergh, T. V., Wyffels, F., Verstraeten, D., Schrauwen, B., Dambre, J., et al. (2014). Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 344–355.

Fujii, K., & Nakajima, K. (2017). Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review A*, *8*, 024030.

Fujita, K., Yonekura, S., Nishikawa, S., Niiyama, R., & Kuniyoshi, Y. (2018). Environmental and structural effects on physical reservoir computing with tensegrity. *Journal of the Institute of Industrial Applications Engineers*, *6*, 92–99.

Furuta, T., Fujii, K., Nakajima, K., Tsunegi, S., Kubota, H., Suzuki, Y., et al. (2018). Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions. *Physical Review Applied*, *10*, 034063.

Fuster, J. (2015). *The prefrontal cortex*. Academic Press.

Gallant, S. I., & King, D. (1988). Experiments with sequential associative memories. In *Proc. the tenth annual conference of the cognitive science society, vol. 4047*.

Gallicchio, C., Micheli, A., & Pedrelli, L. (2017). Deep reservoir computing: a critical experimental analysis. *Neurocomputing*, *268*, 87–99.

George, J. B., Abraham, G. M., Singh, K., Ankolekar, S. M., Amrutur, B., & Sikdar, S. K. (2014). Input coding for neuro-electronic hybrid systems. *Biosystems*, *126*, 1–11.

Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models: single neurons, populations, plasticity*. Cambridge University Press.

Ghani, A., See, C. H., Migdadi, H., Asif, R., Abd-Alhameed, R. A., & Noras, J. M. (2015). Reconfigurable neurons-making the most of configurable logic blocks (CLBs). In *Proc. Internet technologies and applications (ITA)* (pp. 475–478). IEEE.

Gonon, L., & Ortega, J.-P. (2018). Reservoir computing universality with stochastic inputs, arXiv preprint arXiv:1807.02621.

Goudarzi, A., Lakin, M. R., & Stefanovic, D. (2013). DNA Reservoir computing: a novel molecular computing approach. In *Proc. International workshop on DNA-based computers* (pp. 76–89). Springer.

Grigoryeva, L., & Ortega, J.-P. (2018). Echo state networks are universal. *Neural Networks*, *108*, 495–508.

Grill-Spector, K., & Malach, R. (2004). The human visual cortex. *Annu. Rev. Neurosci.*, *27*, 649–677.

Grzyb, B. J., Chinellato, E., Wojcik, G. M., & Kaminski, W. A. (2009). Which model to use for the liquid state machine?. In *Proc. International joint conference on neural networks* (pp. 1018–1024). IEEE.

Hadaeghi, F., He, X., & Jaeger, H. (2017). Unconventional information processing systems, novel hardware: a tour d'Horizon Technical Report IRC-Library, Information Resource Center der Jacobs University Bremen. urn:nbn:de:gbv:579-opus-1007495.

Hafizovic, S., Heer, F., Ugniwenko, T., Frey, U., Blau, A., Ziegler, C., et al. (2007). A CMOS-based microelectrode array for interaction with neuronal cultures. *Journal of Neuroscience Methods*, *164*, 93–106.

Hasler, J., & Marr, H. B. (2013). Finding a roadmap to achieve large neuromorphic hardware systems. *Frontiers in Neuroscience*, *7*, 118.

Hassan, A. M., Li, H. H., & Chen, Y. (2017). Hardware implementation of echo state networks using memristor double crossbar arrays. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 2171–2177). IEEE.

Hauser, H., Ijspeert, A. J., Füchslin, R., & Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biological Cybernetics*, *106*, 595–613.

Hauser, H., Ijspeert, A. J., Füchslin, R., & Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, *105*, 355–370.

Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I., & Gauthier, D. J. (2015). Reservoir computing with a single time-delay autonomous Boolean node. *Physical Review E*, *91*, 020801.

Hermans, M., Antonik, P., Haelterman, M., & Massar, S. (2016). Embodiment of learning in electro-optical signal processors. *Physical Review Letters*, *117*, 128301.

Hermans, M., Dambre, J., & Bienstman, P. (2015). Optoelectronic systems trained with backpropagation through time. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 1545–1550.

Hermans, M., Soriano, M. C., Dambre, J., Bienstman, P., & Fischer, I. (2015). Photonic delay systems as machine learning implementations. *Journal of Machine Learning Research (JMLR)*, *16*, 2081–2097.

Héroux, J. B., Numata, H., Kanazawa, N., & Nakano, D. (2018). Optoelectronic reservoir computing with VCSEL. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 1–6). IEEE.

Héroux, H., & Nakano, D. (2017). Polymer waveguide-based reservoir computing. In *Proc. International conference on neural information processing (ICONIP)* (pp. 840–848). Springer.

Hicke, K., Escalona-Morán, D., Soriano, M. C., Fischer, I., & Mirasso, C. R. (2013). Information processing using transient dynamics of semiconductor lasers subject to delayed feedback. *IEEE Journal of Selected Topics in Quantum Electronics*, *19*, 1501610.

Hinaut, X., & Dominey, P. F. (2013). Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. *PLoS ONE*, *8*, e52946.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Hoerzer, G. M., Legenstein, R., & Maass, W. (2012). Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral Cortex*, *24*, 677–690.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*, 1171–1220.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Hou, Y., Xia, G., Yang, W., Wang, D., Jayaprasath, E., Jiang, Z., Hu, C., & Wu, Z. (2018). Prediction performance of reservoir computing system based on a semiconductor laser subject to double optical feedback and optical injection. *Optics Express*, *26*, 10211–10219.

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., & Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology*, *24*, 384010.

Inubushi, M., & Yoshimura, K. (2017). Reservoir computing beyond memory-nonlinearity trade-off. *Scientific Reports*, *7*, 10199.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks, Technical Report GMD Report 148, German National Research Center forInformation Technology.

Jaeger, H. (2002a). Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology.

Jaeger, H. (2002b). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. Technical Report GMD Report 159, German National Research Center for Information.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In *Advances in neural information processing systems* (pp. 609–616).

Jaeger, H. (2007). Echo state network. *Scholarpedia*, *2*, 2330. http://dx.doi.org/10.4249/scholarpedia.2330, Revision #188245.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, *304*, 78–80.

Jaeger, H., Maass, W., & Principe, J. (2007). Special issue on echo state networks and liquid state machines. *Neural Networks*, *20*, 287–289.

Jalalvand, A., Van Wallendael, G., & Van de Walle, R. (2015). Real-time reservoir computing network-based systems for detection tasks on visual contents. In *Proc. 7th international conference on computational intelligence, communication systems and networks (CICSyN)* (pp. 146–151). IEEE.

Jensen, J. H., & Tufte, G. (2017). Reservoir computing with a chaotic circuit. In *Proc. the European conference on artificial Life, vol. 14* (pp. 222–229). MIT Press.

Jin, Y., & Li, P. (2017). Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition. *Neurocomputing*, *226*, 145–160.

Jones, B., Stekel, D., Rowe, J., & Fernando, C. (2007). Is there a liquid state machine in the bacterium Escherichia Coli. In *Proc. IEEE symposium on artificial Life (ALIFE)* (pp. 187–191). Ieee.

Ju, H., Dranias, M. R., Banumurthy, G., & VanDongen, A. M. (2015). Spatiotemporal memory is an intrinsic property of networks of dissociated cortical neurons. *Journal of Neuroscience*, *35*, 4040–4051.

Katayama, Y., Yamane, T., Nakano, D., Nakane, R., & Tanaka, G. (2016). Wave-based neuromorphic computing framework for brain-like energy efficiency and integration. *IEEE Transactions on Nanotechnology*, *15*, 762–769.

Katumba, A., Freiberger, M., Bienstman, P., & Dambre, J. (2017). A multiple-input strategy to efficient integrated photonic reservoir computing. *Cognitive Computation*, *9*, 307–314.

Katumba, A., Heyvaert, J., Schneider, B., Uvin, S., Dambre, J., & Bienstman, P. (2018). Low-loss photonic reservoir computing with multimode photonic integrated circuits. *Scientific Reports*, *8*, 2653.

Kirby, K. (1991). Context dynamics in neural sequential learning. In *Proc. Florida AI research symposium* (pp. 66–70).

Kirby, K. G., & Day, N. (1990). The neurodynamics of context reverberation learning. In *Proc. The twelfth annual international conference on engineering in medicine and biology society* (pp. 1781–1782). IEEE.

Kudithipudi, D., Saleh, Q., Merkel, C., Thesing, J., & Wysocki, B. (2016). Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing. *Frontiers in Neuroscience*, *9*, 502.

Kulkarni, M. S., & Teuscher, C. (2012). Memristor-based reservoir computing. In *Proc. IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 226–232).

Kuriki, Y., Nakayama, J., Takano, K., & Uchida, A. (2018). Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers. *Optics Express*, *26*, 5777–5788.

Kuwabara, J., Nakajima, K., Kang, R., Branson, D. T., Guglielmino, E., Caldwell, D. G., et al. (2012). Timing-based control via echo state network for soft robotic arm. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Lakshmanan, M. (2011). The fascinating world of the Landau–Lifshitz–Gilbert equation: an overview. *Philosophical Transactions of the Royal Society of London A (Mathematical and Physical Sciences)*, *369*, 1280–1300.

Laporte, F., Katumba, A., Dambre, J., & Bienstman, P. (2018). Numerical demonstration of neuromorphic computing with photonic crystal cavities. *Optics Express*, *26*, 7955–7964.

Larger, L., Baylón-Fuentes, A., Martinenghi, R., Udaltsov, V. S., Chembo, Y. K., & Jacquot, M. (2017). High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Physical Review X*, *7*, 011015.

Larger, L., Soriano, M. C., Brunner, D., Appeltant, L., Gutierrez, J. M., Pesquera, L., et al. (2012). Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Optics Express*, *20*, 3241–3249.

Legenstein, R., & Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, *20*, 323–334.

Lepri, S., Giacomelli, G., Politi, A., & Arecchi, F. (1994). High-dimensional chaos in delayed dynamical systems. *Physica D: Nonlinear Phenomena*, *70*, 235–249.

Li, J., Bai, K., Liu, L., & Yi, Y. (2018). A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system. In *Proc. the 19th international symposium in quality electronic design (ISQED)* (pp. 308–313).

Li, T., Nakajima, K., Cianchetti, M., Laschi, C., & Pfeifer, R. (2012). Behavior switching using reservoir computing for a soft robotic arm. In *Proc. IEEE international conference on robotics and automation (ICRA)* (pp. 4918–4924). IEEE.

Li, J., Zhao, C., Hamedani, K., & Yi, Y. (2017). Analog hardware implementation of spike-based delayed feedback reservoir computing system. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 3439–3446). IEEE.

Liu, Y., Jin, Y., & Li, P. (2018). Online adaptation and energy minimization for hardware recurrent spiking neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, *14*, 11.

Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade* (pp. 659–686). Springer.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*, 127–149.

Lukoševičius, H., & Schrauwen, B. (2012). Reservoir computing trends. *KI-Künstliche Intelligenz*, *26*, 365–371.

Lyon, R. (1982). A computational model of filtering, detection, and compression in the cochlea. In *Proc. IEEE international conference on acoustics, speech, and signal processing (ICASSP), vol. 7* (pp. 1282–1285). IEEE.

Maass, W. (2011). Liquid state machines: motivation, theory, and applications. In *Computability in context: computation and logic in the real world* (pp. 275–296). World Scientific.

Maass, W., Joshi, P., & Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS Computational Biology*, *3*, e165.

Maass, W., & Markram, H. (2004). On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, *69*, 593–616.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, *14*, 2531–2560.

Maass, W., Natschläger, T., & Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal de Physiologie (Paris)*, *98*, 315–330.

Malik, Z. K., Hussain, A., & Wu, Q. J. (2017). Multilayered echo state machine: a novel architecture and algorithm. *IEEE Transactions on cybernetics*, *47*, 946–959.

Mandic, D. P., Chambers, J. A., et al. (2001). *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley Online Library.

Marr, D. (1969). A theory of cerebellar cortex. *The Journal of Physiology*, *202*, 437–470.

Martin, O., Odlyzko, A. M., & Wolfram, S. (1984). Algebraic properties of cellular automata. *Communications in Mathematical Physics*, *93*, 219–258.

Martinenghi, R., Rybalko, S., Jacquot, M., Chembo, Y. K., & Larger, L. (2012). Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Physical Review Letters*, *108*, 244101.

Matthews, M. B. (1993). Approximating nonlinear fading-memory operators using neural network models. *Circuits, Systems, and Signal Processing*, *12*, 279–307.

McDonald, N. (2017). Reservoir computing & extreme learning machines using pairs of cellular automata rules. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 2429–2436). IEEE.

Merkel, C., Saleh, Q., Donahue, C., & Kudithipudi, D. (2014). Memristive reservoir computing architecture for epileptic seizure detection. *Procedia Computer Science*, *41*, 249–254.

Mesaritakis, C., Bogris, A., Kapsalis, A., & Syvridis, D. (2015). High-speed all-optical pattern recognition of dispersive fourier images through a photonic reservoir computing subsystem. *Optics Letters*, *40*, 3416–3419.

Mesaritakis, C., Papataxiarhis, V., & Syvridis, D. (2013). Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system. *Journal of the Optical Society of America. B Optical Physics*, *30*, 3048–3055.

Misra, J., & Saha, I. (2010). Artificial neural networks in hardware: a survey of two decades of progress. *Neurocomputing*, *74*, 239–255.

Morán, A., Frasser, C. F., & Rosselló, J. L. (2018). Reservoir computing hardware with cellular automata, arXiv preprint arXiv:1806.04932.

Murakami, M., Kröger, B., Birkholz, P., & Triesch, J. (2015). Seeing [u] aids vocal learning: babbling and imitation of vowels using a 3d vocal tract model, reinforcement learning, and reservoir computing. In *Proc. Joint IEEE international conference on development and learning and epigenetic robotics (ICDL-EpiRob)* (pp. 208–213). IEEE.

Nakajima, K., Fujii, K., Negoro, M., Mitarai, K., & Kitagawa, M. (2018). Boosting computational power through spatial multiplexing in quantum reservoir computing, arXiv preprint arXiv:1803.04574.

Nakajima, K., Hauser, H., Kang, R., Guglielmino, E., Caldwell, D. G., & Pfeifer, R. (2013). A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience*, *7*, 91.

Nakajima, K., Hauser, H., Li, T., & Pfeifer, R. (2015). Information processing via physical soft body. *Scientific Reports*, *5*, 10487.

Nakajima, K., Li, T., Hauser, H., & Pfeifer, R. (2014). Exploiting short-term memory in soft body dynamics as a computational resource. *Journal of the Royal Society Interface*, *11*, 20140437.

Nakane, R., Tanaka, G., & Hirose, A. (2018). Reservoir computing with spin waves excited in a garnet film. *IEEE Access*, *6*, 4462–4469.

Nakao, H. (2016). Phase reduction approach to synchronisation of nonlinear oscillators. *Contemporary Physics*, *57*, 188–214.

Nakayama, J., Kanno, K., & Uchida, A. (2016). Laser dynamical reservoir computing with consistency: An approach of a chaos mask signal. *Optics Express*, *24*, 8679–8692.

Nguimdo, R. M., Verschaffelt, G., Danckaert, J., & der Sande, G. V. (2015). Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 3301–3307.

Nguimdo, R. M., Verschaffelt, G., Danckaert, J., & der Sande, G. V. (2016). Reducing the phase sensitivity of laser-based optical reservoir computing systems. *Optics Express*, *24*, 1238–1252.

Nichele, S., & Gundersen, M. S. (2017). Reservoir computing using non-uniform binary cellular automata, arXiv preprint arXiv:1702.03812.

Nichele, S., & Molund, A. (2017). Deep reservoir computing using cellular automata, arXiv preprint arXiv:1703.02806.

Nikolić, D., Häusler, S., Singer, W., & Maass, W. (2009). Distributed fading memory for stimulus properties in the primary visual cortex. *PLoS Biology*, *7*, e1000260.

Obien, M. E. J., Deligkaris, K., Bullmann, T., Bakkum, D. J., & Frey, U. (2015). Revealing neuronal function through microelectrode array recordings. *Frontiers in Neuroscience*, *8*, 423.

Obst, O., Trinchi, A., Hardin, S. G., Chadwick, M., Cole, I., Muster, T. H., et al. (2013). Nano-scale reservoir computing. *Nano Communication Networks*, *4*, 189–196.

Omondi, A. R., & Rajapakse, J. C. (2006). *FPGA implementations of neural networks, vol. 365*. Springer.

Ortín, S., & Pesquera, L. (2017). Reservoir computing with an ensemble of time-delay reservoirs. *Cognitive Computation*, *9*, 327–336.

Ortín, S., Soriano, M. C., Pesquera, L., Brunner, D., San-Martín, D., Fischer, I., et al. (2015). A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. *Scientific Reports*, *5*, 14945.

Ortman, R. L., Venayagamoorthy, K., & Potter, S. M. (2011). Input separability in living liquid state machines. In *Proc. International conference on adaptive and natural computing algorithms* (pp. 220–229). Springer.

Ozturk, M. C., Xu, D., & Príncipe, J. C. (2007). Analysis and design of echo state networks. *Neural Computation*, *19*, 111–138.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., et al. (2012). Optoelectronic reservoir computing. *Scientific Reports*, *2*, 287.

Parihar, A., Shukla, N., Jerry, M., Datta, S., & Raychowdhury, A. (2017). Computing with dynamical systems based on insulator–metal-transition oscillators. *Nanophotonics*, *6*, 601–611.

Pascanu, R., & Jaeger, H. (2011). A neurodynamical model for working memory. *Neural networks*, *24*, 199–207.

Petre, P., & Cruz-Albrecht, J. (2016). Neuromorphic mixed-signal circuitry for asynchronous pulse processing. In *Proc. IEEE international conference on rebooting computing (ICRC)* (pp. 1–4). IEEE.

Pfeifer, R., & Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.

Polepalli, A., Soures, N., & Kudithipudi, D. (2016a). Digital neuromorphic design of a liquid state machine for real-time processing. In *Proc. IEEE international conference on rebooting computing (ICRC)* (pp. 1–8). IEEE.

Polepalli, A., Soures, N., & Kudithipudi, D. (2016b). Reconfigurable digital design of a liquid state machine for spatio-temporal data. In *Proc. The 3rd ACM international conference on nanoscale computing and communication* (p. 15). ACM.

Prucnal, P. R., Shastri, B. J., de Lima, T. F., Nahmias, M. A., & Tait, A. N. (2016). Recent progress in semiconductor excitable lasers for photonic spike processing. *Advances in Optics and Photonics*, *8*, 228–299.

Prychynenko, D., Sitte, M., Litzius, K., Krüger, B., Bourianoff, G., Kläui, M., et al. (2018). Magnetic skyrmion as a nonlinear resistive element: a potential building block for reservoir computing. *Physical Review A*, *9*, 014034.

Pyle, R., & Rosenbaum, R. (2018). A model of reward-modulated motor learning with parallelcortical and basal ganglia pathways, arXiv preprint arXiv:1803.03304.

Qiao, J., Li, F., Han, H., & Li, W. (2017). Growing echo-state network with multiple subreservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, *28*, 391–404.

Qin, J., Zhao, Q., Xu, D., Yin, H., Chang, Y., & Huang, D. (2016). Optical packet header identification utilizing an all-optical feedback chaotic reservoir computing. *Modern Physics Letters B*, *30*, 1650199.

Qin, J., Zhao, Q., Yin, H., Jin, Y., & Liu, C. (2017). Numerical simulation and experiment on optical packet header recognition utilizing reservoir computing based on optoelectronic feedback. *IEEE Photonics Journal*, *9*, 1–11.

Rapoport, M., van Reekum, R., & Mayberg, H. (2000). The role of the cerebellum in cognition and behavior: a selective review. *The Journal of Neuropsychiatry and Clinical Neurosciences*, *12*, 193–198.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., et al. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, *497*, 585.

Roy, S., Banerjee, A., & Basu, A. (2014). Liquid state machine with dendritically enhanced readout for low-power, neuromorphic VLSI implementations. *IEEE Transactions on Biomedical Circuits and Systems*, *8*, 681–695.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. Technical Report California Univ. San Diego La Jolla Inst. for Cognitive Science.

Samarasinghe, S. (2016). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Auerbach publications.

Van der Sande, G., Brunner, D., & Soriano, M. C. (2017). Advances in photonic reservoir computing. *Nanophotonics*, *6*, 561–576.

Scardapane, S., & Wang, D. (2017). Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks*, *61*, 85–117.

Schneider, B., Dambre, J., & Bienstman, P. (2016). Using digital masks to enhance the bandwidth tolerance and improve the performance of on-chip reservoir computing systems. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 2748–2753.

Schomaker, L., & Richardus, B. (1991). *Simulation and recognition of handwriting movements: a vertical approach to modeling human motor behavior*. Nijmegen: NICI (NIjmegen Institute of Cognitive Information), Katholieke Universiteit Nijmegen.

Schomaker, L., & Richardus, B. (1992). A neural oscillator-network model of temporal pattern generation. *Human Movement Science*, *11*, 181–192.

Schrauwen, B., D' Haene, M., Verstraeten, D., & Van Campenhout, J. (2008). Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Networks*, *21*, 511–523.

Schrauwen, B., & Van Campenhout, J. (2003). BSA, a fast and accurate spike train encoding scheme. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 2825–2830). IEEE.

Schrauwen, B., Verstraeten, D., & Van Campenhout, J. (2007). An overview of reservoir computing: theory, applications and implementations. In *Proc. the 15th European symposium on artificial neural networks* (pp. 471–482).

Schürmann, F., Meier, K., & Schemmel, J. (2005). Edge of chaos computation in mixed-mode VLSI - "a hard liquid". In *Proc. Advances in neural information processing systems* (pp. 1201–1208).

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., & Poggio, T. (2005). *A theory of object recognition:computations and circuits in the feedforward path of the ventral stream in primate visual cortex*: *Technical Report No. 259 Center for Biological and Computational Learning (CBCL)*, MIT.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: vision and challenges. *IEEE Internet of Things Journal*, *3*, 637–646.

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., et al. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, *24*, 384004.

Smith, M. R., Hill, A. J., Carlson, K. D., Vineyard, C. M., Donaldson, J., Follett, D. R., et al. (2017). A novel digital neuromorphic architecture efficiently facilitating complex synaptic response functions applied to liquid state machines. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 2421–2428). IEEE.

Snyder, D., Goudarzi, A., & Teuscher, C. (2013). Computational capabilities of random automata networks for reservoir computing. *Physical Review E*, *87*, 042808.

Soh, H., & Demiris, Y. (2012). Iterative temporal learning and prediction with the sparse online echo state gaussian process. In *Proc. International joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Soriano, M. C., Brunner, D., Escalona-Morán, C. R., & Fischer, I. (2015). Minimal approach to neuro-inspired information processing. *Frontiers in Computational Neuroscience*, *9*, 68.

Soriano, M. C., Ortín, S., Brunner, D., Larger, L., Mirasso, C. R., Fischer, I., et al. (2013). Optoelectronic reservoir computing: tackling noise-induced performance degradation. *Optics Express*, *21*, 12–20.

Soriano, M. C., Ortín, S., Keuninckx, L., Appeltant, L., Danckaert, J., Pesquera, L., & Van der Sande, G. (2015). Delay-based reservoir computing: noise effects in a combined analog and digital implementation. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 388–393.

Soures, N., Hays, L., & Kudithipudi, D. (2017). Robustness of a memristor based liquid state machine. In *Proc. International conference on neural networks (IJCNN)* (pp. 2414–2420). IEEE.

Stancil, D. D., & Prabhakar, A. (2009). *Spin waves*. Springer.

Steil, J. J. (2004). Backpropagation-decorrelation: online recurrent learning with O(N) complexity. In *Proc. International joint conference on neural networks (IJCNN), vol. 2* (pp. 843–848). IEEE.

Steil, J. J. (2007). Online reservoir adaptation by intrinsic plasticity for backpropagation - decorrelation and echo state learning. *Neural Networks*, *20*, 353–364.

Stieg, A. Z., Avizienis, A. V., Sillin, H. O., Aguilera, R., Shieh, H.-H., Martin-Olmos, C., et al. (2014). Self-organization and emergence of dynamical structures in neuromorphic atomic switch networks. In *Memristor networks* (pp. 173–209). Springer.

Stieg, A. Z., Avizienis, A. V., Sillin, H. O., Martin-Olmos, C., Aono, M., & Gimzewski, J. K. (2012). Emergent criticality in complex Turing B-Type atomic switch networks. *Advanced Materials*, *24*, 286–293.

Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, *453*, 80–83.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*, 544–557.

Takahashi, H., Yasuda, S., Yada, Y., & Kanzaki, R. (2016). Reservoir computing with dissociated neuronal culture. *Frontiers in Neuroscience*, *27*.

Takeda, S., Nakano, D., Yamane, T., Tanaka, G., Nakane, R., Hirose, A., et al. (2016). Photonic reservoir computing based on laser dynamics with external feedback. In *Proc. International conference on neural information processing (ICONIP)* (pp. 222–230). Springer.

Tanaka, G., Nakane, R., Yamane, T., Nakano, D., Takeda, S., Nakagawa, S., et al. (2016). Exploiting heterogeneous units for reservoir computing with simple architecture. In *Proc. International conference on neural information processing (ICONIP)* (pp. 187–194). Springer.

Tanaka, G., Nakane, R., Yamane, T., Takeda, S., Nakano, D., Nakagawa, S., et al. (2017). Waveform classification by memristive reservoir computing. In *Proc. International conference on neural information processing (ICONIP)* (pp. 457–465). Springer.

Tananka, G., Nakane, R., Takeuchi, T., Yamane, T., Nakano, D., Katayama, Y., et al. (2019). Spatially arranged sparse recurrent neural networks for energy efficient associative memory. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2019.2899344, (in press).

Thomas, A. (2013). Memristor-based neural networks. *Journal of Physics D: Applied Physics*, *46*, 093001.

Toledo-Suárez, R., & Morrison, A. (2014). Liquid computing on and off the edge of chaos with a striatal microcircuit. *Frontiers in Computational Neuroscience*, *8*, 130.

Tong, Z., & Tanaka, G. (2018). Reservoir computing with untrained convolutional neural networks for image recognition. In *Proc. 24th international conference on pattern recognition (ICPR)* (pp. 1289–1294). IEEE.

Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., et al. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, *547*, 428.

Tran, S. D., & Teuscher, C. (2017). Memcapacitive reservoir computing. In *Proc. IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 115–116). IEEE.

Urbain, G., Degrave, J., Carette, B., Dambre, J., & Wyffels, F. (2017). Morphological properties of mass–spring networks for optimal locomotion learning. *Frontiers in Neurorobotics*, *11*, 16.

Vaerenbergh, T. V., Fiers, M., Dambre, J., & Bienstman, P. (2015). Efficient simulation of optical nonlinear cavity circuits. *Optical and Quantum Electronics*, *47*, 1471–1476.

Vandoorne, K., Dambre, J., Verstraeten, D., Schrauwen, B., & Bienstman, P. (2011). Parallel reservoir computing using optical amplifiers. *IEEE Transactions on Neural Networks*, *22*, 1469–1481.

Vandoorne, K., Dierckx, W., Schrauwen, B., Verstraeten, D., Baets, R., Bienstman, P., et al. (2008). Toward optical signal processing using photonic reservoir computing. *Optics Express*, *16*, 11182–11192.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., et al. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communications*, *5*, 3541.

Verstraeten, D., Schrauwen, B., d' Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, *20*, 391–403.

Verstraeten, D., Schrauwen, B., & Stroobandt, D. (2005). Reservoir computing with stochastic bitstream neurons. In *Proc. the 16th annual prorisc workshop* (pp. 454–459).

Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, *95*, 521–528.

Vincent-Lamarre, P., Lajoie, G., & Thivierge, J.-P. (2016). Driving reservoir models with oscillations: a solution to the extreme structural sensitivity of chaotic networks. *Journal of Computational Neuroscience*, *41*, 305–322.

Vinckier, Q., Duport, F., Smerieri, A., Vandoorne, K., Bienstman, P., Haelterman, M., et al. (2015). High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica*, *2*, 438–446.

Walter, F., Röhrbein, F., & Knoll, A. (2015). Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks. *Neural Networks*, *72*, 152–167.

Wang, Q., Jin, Y., & Li, P. (2015). General-purpose lsm learning processor architecture and theoretically guided design space exploration. In *Proc. IEEE biomedical circuits and systems conference (BioCAS)* (pp. 1–4). IEEE.

Wang, Q., Li, Y., & Li, P. (2016). Liquid state machine based pattern recognition on fpga with firing-activity dependent power gating and approximate computing. In *Proc. IEEE international symposium on circuits and systems (ISCAS)* (pp. 361–364). IEEE.

Wang, Q., Li, Y., Shao, B., Dey, S., & Li, P. (2017). Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA. *Neurocomputing*, *221*, 146–158.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*, 1550–1560.

Williame, J., Accioly, A. D., & Kim, J.-V. (2017). Chaotic dynamics in a macrospin spin-torque nano-oscillator with time-delayed feedback, arXiv preprint arXiv:1709.04310.

Williams, R. S. (2014). How we found the missing memristor. In *Memristors and memristive systems* (pp. 3–16). Springer.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270–280.

Wojcik, G. M., & Kaminski, W. A. (2007). Liquid state machine and its separation ability as function of electrical parameters of cell. *Neurocomputing*, *70*, 2593–2597.

Wolf, S., Awschalom, D., Buhrman, R., Daughton, J., Von Molnar, S., Roukes, M., et al. (2001). Spintronics: a spin-based electronics vision for the future. *Science*, *294*, 1488–1495.

Wolfram, S. (2018). Cellular automata and complexity: collected papers. CRC Press.

Wyffels, F., D'Haene, M., Waegeman, T., Caluwaerts, K., Nunes, C., & Schrauwen, B. (2010). Realization of a passive compliant robot dog. In *Proc. the 3rd IEEE RAS and EMBS international conference on biomedical robotics and biomechatronics (BioRob)* (pp. 882–886). IEEE.

Xia, Y., Jahanchahi, C., & Mandic, D. P. (2015). Quaternion-valued echo state networks. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 663–673.

Yamane, T., Katayama, Y., Nakane, R., Tanaka, G., & Nakano, D. (2015). Wave-based reservoir computing by synchronization of coupled oscillators. In *Proc. International conference on neural information processing (ICONIP)* (pp. 198–205). Springer.

Yamazaki, T., & Tanaka, S. (2007). The cerebellum as a liquid state machine. *Neural Networks*, *20*, 290–297.

Yang, X., Chen, W., & Wang, F. Z. (2016). Investigations of the staircase memristor model and applications of memristor-based local connections. *Analog Integrated Circuits and Signal Processing*, *87*, 263–273.

Yi, Y., Liao, Y., Wang, B., Fu, X., Shen, F., Hou, H., et al. (2016). FPGA Based spike-time dependent encoder and reservoir design in neuromorphic computing processors. *Microprocessors and Microsystems*, *46*, 175–183.

Yildiz, I. B., Jaeger, H., & Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, *35*, 1–9.

Yilmaz, O. (2014). Reservoir computing using cellular automata, arXiv preprint arXiv:1410.0162.

Yilmaz, O. (2015a). Machine learning using cellular automata based feature expansion and reservoir computing. *Journal of Cellular Automata*, *10*, 435–472.

Yilmaz, O. (2015b). Symbolic computation using cellular automata-based hyperdimensional computing. *Neural Computation*, *27*, 2661–2692.

Zhang, H., Feng, X., Li, B., Wang, Y., Cui, K., Liu, F., et al. (2014). Integrated photonic reservoir computing based on hierarchical time-multiplexing structure. *Optics Express*, *22*, 31356–31370.

Zhang, Y., Li, P., Jin, Y., & Choe, Y. (2015). A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 2635–2649.

Zhao, C., Li, J., Liu, L., Koutha, L. S., Liu, J., & Yi, Y. (2016). Novel spike based reservoir node design with high performance spike delay loop. In *Proc. the 3rd ACM international conference on nanoscale computing and communication* (p. 14). ACM.

Zhao, Q., Nakajima, K., Sumioka, H., Hauser, H., & Pfeifer, R. (2013). Spine dynamics as a computational resource in spine-driven quadruped locomotion. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1445–1451). IEEE.

Zhao, Q., Yin, H., & Zhu, H. (2018). Simultaneous recognition of two channels of optical packet headers utilizing reservoir computing subject to mutual-coupling optoelectronic feedback. *Optik*, *157*, 951–956.

Zhu, J., & Sutton, P. (2003). FPGA Implementations of neural networks - a survey of a decade of progress. In *Proc. International conference on field programmable logic and applications* (pp. 1062–1066). Springer.