DFR: An Energy-efficient Analog Delay Feedback Reservoir Computing System for Brain-inspired Computing

KANGJUN BAI and YANG YI, Virginia Polytechnic Institute and State University, Virginia Tech

Neuromorphic computing, which is built on a brain-inspired silicon chip, is uniquely applied to keep pace with the explosive escalation of algorithms and data density on machine learning. Reservoir computing, an emerging computing paradigm based on the recurrent neural network with proven benefits across multifaceted applications, offers an alternative training mechanism only at the readout stage. In this work, we successfully design and fabricate an energy-efficient analog delayed feedback reservoir (DFR) computing system, which is built upon a temporal encoding scheme, a nonlinear transfer function, and a dynamic delayed feedback loop. Measurement results demonstrate its high energy efficiency with rich dynamic behaviors, making the designed system a candidate for low power embedded applications. The system performance, as well as the robustness, are studied and analyzed through the Monte Carlo simulation. The chaotic time series prediction benchmark, NARMA10, is examined through the proposed DFR computing system, and exhibits a 36%–85% reduction on the error rate compared to state-of-the-art DFR computing system designs. To the best of our knowledge, our work represents the first analog integrated circuit (IC) implementation of the DFR computing system.

CCS Concepts: • **Hardware** → **Integrated circuits**; • **Computing methodologies** → Model development and analysis;

Additional Key Words and Phrases: Reservoir computing, brain-inspired computing, spiking neural network, edge of chaos regime, delayed feedback system, analog integrated circuit design

ACM Reference format:

Kangjun Bai and Yang Yi. 2018. DFR: An Energy-efficient Analog Delay Feedback Reservoir Computing System for Brain-inspired Computing. *J. Emerg. Technol. Comput. Syst.* 14, 4, Article 45 (December 2018), 22 pages.

<https://doi.org/10.1145/3264659>

1 INTRODUCTION

Hindered by the gap between physical limitations of the complementary metal-oxide-semiconductor (CMOS) technology and the storage capacity, modern von Neumann computing architectures can no longer double its performance every 18 months, which indicates the end of Moore's prediction (Kish 2002). Due to the explosive escalation of data density, the extraction of valuable information has become computationally expensive, even for supercomputers. The amount of energy required for supercomputers poses doubt on whether the increased

This material is based upon work funded by Air Force Research Lab (AFRL), under AFRL Grant No. FA8750-16-2-0120.

Authors' addresses: K. Bai and Y. Yi, The Bradley Department of Electrical and Computing Engineering (ECE) at Virginia Polytechnics and Institute State University (Virginia Tech), 302 Whittemore (0111), 1185 Perry Street, Virginia Tech, Blacksburg, VA 24061; emails: {kangjun, yangyi8}@vt.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

© 2018 Association for Computing Machinery.

1550-4832/2018/12-ART45 \$15.00

<https://doi.org/10.1145/3264659>

performance is affordable or even sustainable. The accumulative amount of energy spent on data processing with the traditional computing architecture has become a burden to the world's energy consumption (Dayarathna Wen and Fan 2016).

As humans, our brains process sophisticated capabilities for learning, memorizing and analyzing information with a mere 10W of power consumption (Yu et al. 2011). Inspired by the biology and proposed by Dr. Carver Mead in the 1980s, neuromorphic computing has matured to provided intelligent systems that able to imitate natural neuro-biological processes through highly parallelized computing architectures; such systems typically model the function of neural network through very-large-scaled-integrated (VLSI) circuits (Mead 1990). As powerful as mammalian brains, the neuromorphic computing can efficiently solve complex tasks with applications in pattern recognition and optimization (Chu et al. 2015; Hu et al. 2016), anomaly detection (Begg and Kamruzzaman 2005; Chen et al. 2013), automated data processing (Esser et al. 2015), as well as information fusion (Nasrabadi 2007). For instance, TrueNorth, the brain-inspired chip prototyped by IBM in 2014, classifies multiple objects within 240×240 pixel three-color video input at 30 frames per second with merely 65mW of power consumption, which is five orders of magnitude more efficient than the traditional computing system (Merolla et al. 2014; Modha 2017). Loihi, the latest neuromorphic chip prototyped by Intel in 2017, contains 130,000 neurons through 130,000,000 synapses with a mere 1/1,000 power consumption of the one used by a classic computer (Davies et al. 2018; Mayberry 2017).

To mimic network behaviors, hardware-based artificial neural networks (ANNs) employ electronic circuits to imitate relevant functions of neurological systems. Two well-known ANNs, feed-forward neural networks (FNNs) and recurrent neural networks (RNNs), are powerful ANNs that are capable of learning, which is not probable in the classic computing system reliant on pre-programmed instructions. RNNs, which are constructed upon FNNs with random recurrent connections, closely mimic feedback operations in neurological systems and have the capability to process temporal information (Graves et al. 2006). However, although RNNs are more powerful in performing temporal tasks than FNNs, the training complexity of recurrent connections are computationally expensive.

Introduced in the early 2000s by Jaeger (Jaeger 2001a) and Maass (Maass Natschläger and Markram 2002), respectively, the echo state network (ESN) and the liquid state machine (LSM) exploit dynamic behaviors of RNNs. The ESN and the LSM, a branch of the reservoir computing, are considered as a simplification of conventional RNNs with simple classification and regression techniques. In both ESN and LSM models, a layer of nonlinear neurons, which carry out the nonlinear transformation, with untrained synaptic weights is constructed, namely, the reservoir layer. By only training output weights, the training complexity is greatly reduced. In the past decade, both ESN and LSM models have been fully developed in both software and hardware with the field programmable gate arrays (FPGA), and proven their benefits across multifaceted applications (Lin Yang and Song 2011; Skowronski and Harris 2007; Wang Li and Li 2016; Zhang et al. 2015). However, the realization of an enormous number of neurons hinders the development of the integrated circuit (IC) implementation in both ESN and LSM.

As is well known, the delay is ubiquitous in every biological system (Beuter Bélair and Labrie 1993; Haken 2002; Kandel et al. 2000). With the embedded delay property, the operating mechanism of ESN and LSM models has conceptually evolved to the delayed feedback reservoir (DFR) computing system (Appeltant et al. 2011; Grigoryeva et al. 2015; Ortín and Pesquera 2017). In this context, reservoirs function as a delayed feedback network to construct the persistent memory for data processing. In recent years, the photonic implementation of the DFR computing system has attracted widespread attention (Callan et al. 2010; Larger and Dudley 2010; Larger et al. 2012). The DFR computing system does not only have a resemblance to mammalian brains but also exhibits

a near chaotic behavior (Verstraeten et al. 2007). However, to the best of our knowledge, a pure analog IC implementation of the DFR computing system has not been designed and fabricated yet.

In this work, we design and fabricate an energy-efficient analog DFR computing system that built upon a temporal encoding scheme, a nonlinear transfer function, and a dynamic delayed feedback loop. Contributions of our work include the following:

1. In the endeavor to imitate how human beings process information, our DFR computing system processes analog signal directly, resulting in higher energy efficiency and smaller design area;
2. With considerations of the system robustness and the noise immunity, our DFR chip has been successfully fabricated and tested. To the best of our knowledge, the fabricated DFR chip is the first analog IC implementation of the DFR computing system;
3. Measurement results demonstrate DFR's rich dynamic behaviors, which is indicative of a system that has the capability to emulate the neurological transmission with merely $526\mu\text{W}$ of power consumption;
4. Experimental results in chaotic time series prediction benchmark exhibit 36%–85% reduction on the error rate compared to state-of-the-art DFR computing system designs.

In this article, an overview of the reservoir computing is described in Section 2. The architecture and circuit implementation of the designed analog DFR computing system is demonstrated in Sections 3 and 4, respectively, followed by the performance evaluation in Section 5. The article is then concluded in Section 6.

2 RESERVOIR COMPUTING

2.1 Overview

In past several decades, the neuron function was modeled as a linear integration of multiple synaptic inputs followed by a threshold nonlinearity. However, relatively recent neural information processing research has conceptually evolved to a much more sophisticated mixed-signal evaluation with highly adaptive synaptic elements (Koch and Segev 2000). This data suggest that the spatial arrangement and filtering properties of biological synapses and dendrites lead to the temporal-spatial characteristics of the synaptic input to highly nonlinear. These two opposing temporal-spatial nonlinear transformation processes enable the dynamic of neurons to transit between linear and sublinear regimes. Such property enables the projection of linear excitatory input patterns onto higher-dimensional spaces with the potential to classify otherwise non-separable functions. This single-neuron nonlinear-processing property has attracted widespread attention, not only in the neuroscience but also in the field of ANNs (Giles and Maxwell 1987). RNNs, which successfully adopt the temporal-spatial characteristics within its neural layer, imitate the working mechanism of neurological systems. RNNs are constructed upon FNNs with random recurrent connections within its neural layer. These recurrent connections enable RNNs to solve complex temporal tasks, such as temporal classification or prediction (Schrauwen Verstraeten and Van Campenhout 2007). However, due to the state of the network with recurrent connections, neural states can possibly depend on all previous input values, which result in a computationally expensive training process.

The architectural concept behind the reservoir computing, as illustrated in Figure 1, is to exploit dynamic behaviors of conventional RNNs and computationally simplify its training mechanism. In general, the input layer serves as a scaling function to transfer scaled input patterns to the reservoir layer. Unlike conventional RNNs, in the reservoir computing, connections within the reservoir layer are built upon a layer of nonlinear neurons with fixed and untrained synaptic weights. The output layer performs a linear regression to estimate a predicted signal from the excited reservoir

Fig. 1. The general architecture of the reservoir computing.

Fig. 2. Activation functions of (a) step function, (b) sigmoid function, and (c) hyperbolic tangent function.

layer. Two well-known and representation reservoir computing models, ESN and LSM, employ the strength of conventional RNNs in which synaptic connections within the reservoir layer are not trained (Jaeger 2007). By only training output weights, the complexity of the training process is significantly reduced, resulting in less computational power.

In ESN, synaptic weights within the reservoir layer are sampled uniformly from the interval of $[-1, 1]$, and the spectral radius, λ , is set to be less than 1 to satisfy the ESN property (Jaeger 2001a). Similar to the ESN, the LSM is also constructed based on adapting the RNN as its reservoir layer; such that the internal noise has taken into consideration. The major difference sets these two models of reservoir computing apart is the signal format. In LSM, spiking signals are used as pre- and post-neural signals, while analog signals are examined in ESN. The general node state of the reservoir computing could be expressed as

$$s(t) = f \left[W_{sys} \cdot s(t-1) + W_{in} \cdot x(t-1) \right], \quad (1)$$

where $s(t)$ is the node state at time t , $x(t-1)$ is input patterns, W_{sys} and W_{in} represent randomly generated synaptic weights and input weights, respectively. The output, $\hat{y}(t)$, could be expressed in terms of input patterns and weight connections,

$$y(t) = W_{out} \cdot s(t) + W_{fb} \cdot x(t-1) + W_{bias}, \quad (2)$$

where W_{out} is output weights from the reservoir layer, W_{fb} is feedback weights from the output layer to the reservoir layer, and W_{bias} is the bias weight.

These neuron-like nodes within the reservoir layer process the function of (1) high-dimensional projection and (2) fading memory, whereby the biological neuron's behavior is achieved. These advantages make the reservoir computing especially suitable for neuromorphic computing paradigms. The performance metrics of reservoir computing outperforms conventional RNNs by three orders and two orders of magnitude more accurate in chaotic dynamic predictions and nonlinear channel equalization, respectively (Jaeger and Haas 2004; Lukoševicius 2012). The

Fig. 3. Simplified illustration of delay differential equation (DDE).

Reservoir computing has proven its benefit across multifaceted applications, for instance, speech recognition (Alalshekmubarak and Smith 2014; Ghani 2010; Overton 2014), character recognition (Jin et al. 2015; Verstraeten Schrauwen and Stroobandt 2005), grammar modeling (Hinaut and Dominey 2012), noise modeling (Goudarzi Lakin and Stefanovic 2014) as well as the generation and prediction of chaotic time series (Jaeger 2001b; Schrauwen and Stroobandt 2008).

2.2 Nonlinear Activation Function and Delay System

The correlation between the input and output is mathematically represented by using activation functions, which can be either linear or nonlinear depending on applications. The step function is the most commonly used linear function in the past century (Myers and Hutchinson 1989). However, the recent neural information processing research has conceptually evolved to a much more sophisticated mixed-signal evaluation with highly adaptive synaptic elements, linear functions will no longer be a candidate when mimicking the functionality of neurological systems. In the past several decades, sigmoid and hyperbolic tangent functions are the most popular activation functions that exhibit the capability of nonlinear transformation. For instance, the hyperbolic tangent function is the widely used nonlinear activation function in ESN and LSM for high-dimensional projection (Jaeger 2007; Vandoorne et al. 2008). Obviously, both sigmoid and hyperbolic tangent functions have similar shapes but different ranges. It can be observed that the hyperbolic tangent function is antisymmetric with respect to the origin, which is proven to converge faster than sigmoid function (Haykin 2001). However, both sigmoid and hyperbolic tangent functions do not have the delay property to emulate the biological behavior of neurological systems.

Inevitably, the delay is ubiquitous in almost every system, especially biological systems (Beuter Bélair and Labrie 1993; Haken 2002; Kandel et al. 2000). For instance, diffusion or transport of substances (cellular metabolites, hormones, oxygen, and carbon dioxide in the blood), conduction time of nerves, as well as intrinsic times for synthesis, growth, and reproduction (Milton 2015). Within mammalian brains, the delay takes place between two neurons because of the axonal conduction delay (Haken 2002). With the delay embedded into the system, such computing architecture has the capability to exhibit rich dynamic behaviors during operations. In recent years, the delay differential equation (DDE), which falls into the category of nonlinear activation functions, with the delay property depicted in Figure 3 (Wu 1998; Xu and Yang 2005). Originating from the ordinary differential equation (ODE), the DDE defines a system in which the dynamics depend on both current and previous state, such that the dynamic behavior of the system changes accordingly as the delay varies. The DDE is generally expressed as

$$\frac{dy}{dx} = f[x(t), x(t - \tau)], \quad (3)$$

where $f[\]$ represents a nonlinear activation function of the current state, $x(t)$, as well as its previous state, $x(t - \tau)$, and τ is the delay time. In biological systems, delays between two neurons

arise from axonal conduction delays. The introduced delay can either stabilize or destabilize a dynamical system to transform the dynamic behavior from periodic to chaotic, or vice versa. It has been proven that the best computational performance occurs at the transition region between stable and chaotic regimes (Legenstein and Maass 2007a, 2007b; Schrauwen and Legenstein 2010), namely, the “edge of chaos” regime. Neither sigmoid nor hyperbolic tangent functions can provide such benefits. DDE triumphs sigmoid and hyperbolic tangent functions over its embedded delay property, which enables the system to achieve the high-dimensional projection and to operate at the edge of the chaos regime (Ikeda and Matsumoto 1987).

2.3 DFR Computing System

While classic reservoir computing has been fully developed in the past decade to reduce the training complexity of RNNs, the development of its hardware implementations is hindered by an enormous number of nonlinear neurons. Benefiting from the embedded delay property, the training mechanism and computing architecture of the classic reservoir computing have conceptually evolved to the DFR computing system. In a DFR computing system, the reservoir layer is built upon a single nonlinear neuron with a static delayed feedback loop. In this context, DFR utilizes the feedback signal as a persistent memory to process the time-series input patterns. The dynamic behavior of the DFR computing system could be influenced by controlling the delay time of the system. In Appeltant et al. (2011), numerical results show that a DFR computing system has an identical performance compared to the classic reservoir computing. In general, output nodes in the DFR computing system are linear weighted sums of the tapped states within the delayed feedback loop, which could be given as

$$\hat{y}(t) = \sum_{i=1}^N W_{out(i)} \cdot x \left[t - \frac{\tau}{N} \cdot (N - i) \right]. \quad (4)$$

To satisfy both high-dimensional projection and delay properties, the DFR computing system offers simplification in the hardware implementation of classic reservoir computing.

2.4 State-of-the-art Hardware Implementations of Reservoir Computing

As discussed in Section 2.1, reservoir computing is considered as the simplification of conventional RNNs; in the past decade, both ESN and LSM models have been fully developed in the FPGA for data recognition and classification. In Wang Li and Li (2016), a LSM model is implemented with 135 neurons in a FPGA for pattern recognition, and yielded 96.4% recognition accuracy. Moreover, in Schrauwen et al. (2008) and (Zhang et al. 2015), two LSM models with 200 and 343 neurons, respectively, are built in a FPGA for speech recognition, and yielded 95% and 99.79% recognition accuracy, respectively. Last, in Alomar et al. (2016), an ESN model is implemented in a FPGA for chaotic-time series forecasting; its simulated mean squared error (MSE) is as low as 2.39×10^{-4} .

In recent years, the photonic implementation of the delayed feedback system has attracted widespread attention (Callan et al. 2010; Larger and Dudley 2010; Larger et al. 2012). DFR networks with photonic implementation introduce the phenomenon of optical chaos; such dynamic system has proven its benefit to multifaceted applications (Larger and Dudley 2010). In Larger et al. (2012) and (Paquot et al. 2012), a photonic implementation of the reservoir with the semiconductor optical amplifiers (SOA) is discussed. The photonic implementation offers high-speed optical processing and high bandwidth; most importantly, photonic devices are nonlinear in nature. The photonic implementation, however, can require expensive peripheral devices such as a digitizer and waveform generator for signal processing, which results in complex design technique, large design area, and

Fig. 4. Overview of designed analog DFR computing system.

high power consumption. An IC implementation, however, offers compact design area with low power consumption, but would be susceptible to noise, which makes it difficult to implement.

In state-of-the-art DFR computing system designs (Amil Cabeza and Marti 2015; Appeltant et al. 2011; Soriano et al. 2015), digital implementation is usually preferred due to its advantages in ease of implementation and noise immunity (Eryilmaz et al. 2016). However, real-time operations require interfacing with analog signals. Hence, power-hungry peripheral devices, such as analog multipliers, analog-to-digital (A/D) and digital-to-analog (D/A) converters, and so on, are needed to facilitate the computation. Obviously, with the interface of these peripheral components, digital implementation often results in high power consumption and a large design area. Unlike the digital implementation, analog implementation computes analog signals directly without A/D conversion, hence, dramatically reduce the power consumption and design area. More importantly, analog implementation closely mimics the physical characteristic of neurological systems.

3 ARCHITECTURE AND OPERATING PRINCIPLE

Figure 4 depicts an overview of our designed analog DFR computing system, which is built upon a temporal encoding scheme, a nonlinear transfer function, and a dynamic delayed feedback loop.

Similar to classic DFR computing systems, our designed analog DFR computing system is constructed with three layers, namely, the input layer, the reservoir layer, and the output layer. Unlike the classic DFR computing system, in our analog DFR computing system design, the masking interface within the input layer is substituted by a temporal encoder (Zhao et al. 2016; Zhao et al. 2017), such that pre- and post-neuron signals are represented by a temporal spike train. In a temporal spike train, the carried information from input patterns is encoded into the time intervals, D_i , between spikes. Such encoded pattern allows each spike to be the reference frame with respect to each other, as illustrated in Figure 5. The temporal encoder is built with four major function modules, which include a voltage reference, a signal regulator, a neuron pool, and a signal integrator. Neuron pool is the key function module to generate the temporal information. In the circuit implementation, the neuron pool is built with four leaky integrate-and-fire (LIF) neurons.

Time intervals, D_i , between spikes could be expressed as

$$D_i = f_c(C_m, V_{th}) - f_n(C_m, V_{th}), \quad (5)$$

Fig. 5. Example of temporal code with the fabricated temporal encoder (Zhao et al. 2017).

Fig. 6. Classification with (a) nonlinear separation in low-dimensional spaces; (b) linear separation in high-dimensional spaces.

where functions, $f_c(\cdot)$ and $f_n(\cdot)$, indicate the spike position of the current state and its next state, respectively, which could be written as

$$f(C_m, V_{th}) = C_m \cdot [\beta \cdot (V_{th} - \gamma) + \theta], \quad (6)$$

where β , γ , and θ are design parameters that relate to the firing and refractory period, and C_m and V_{th} are the membrane capacitance and firing threshold potential, respectively.

As depicted in Figure 4, the analog input signal is first encoded into a spike train by the temporal encoder, followed by interfacing with the reservoir layer. Within the reservoir layer, the encoded input signal is projected onto a higher-dimensional space for linear separation by the nonlinear transfer function, followed by injecting into the dynamic delayed feedback loop. High-dimensional projection in the reservoir computing, as illustrated in Figure 6, is the key module for separating input patterns into different categories (Appeltant et al. 2011). For instance, with low-dimensional spaces, two different objects cannot be linearly separated by a single cut-off rule, as depicted in Figure 6(a). However, by projecting input patterns onto higher-dimensional spaces, from two-dimensional to three-dimensional, the separability of the system changes accordingly. As demonstrated in Figure 6(b), the same objects are linearly separated by a single cut-off plane without changing their original xy position. This high-dimensional projection property enables our designed analog DFR computing system to become linearly separable for future classification.

The nonlinear transformed temporal spike train travels along the dynamic delayed feedback loop and eventually integrates with the next incoming input data to construct the persistent memory. Under the influence of varied delays, the reservoir layer exhibits a transient response, which enables the functionality of temporal information processing. To facilitate the efficient

Fig. 7. Illustration of persistent memory.

data processing without losing the computational accuracy, the dynamic delayed feedback loop is built with a decoding operation and a signal gain regulation, such that the new incoming input data is dominant. In our analog DFR computing system, the dynamic behavior can be controlled by regulating the total delay time within the system. Such property is realized by multiple integrate-and-fire (I&F) delay neurons with a delay calibration module. Along the dynamic delayed feedback loop, the total delay time, T , is divided into N equidistant delay neurons with an identical delay time constant of τ_{delay} , such that

$$\tau_{delay} = \frac{T}{N}. \quad (7)$$

In classic reservoir computing, represented by ESN, the short-term memory within the reservoir layer fades in time due to losses and the way that neurons are sparsely connected (Jaeger 2007); such fading memory limits the performance of computation. In general, gradient-based methods and the backpropagation algorithm are the commonly used training mechanism for the classic reservoir computing (Goodfellow et al. 2016). In such training algorithm, weights within the neural network update proportionally to the gradient of the error function, which examines the difference between the predicted value and the target value. According to Kong and Takatsuka (2017), the general idea of the backpropagation algorithm is written as

$$\frac{\partial E}{\partial W_{in}} = \frac{\partial E}{\partial f_i} \frac{\partial f_i}{\partial W_{in}}, \quad (8)$$

where W_{in} is the matrix of input weights and f_i denotes the output vector at i^{th} layer. Obviously, the gradient is proportional to the factor of $\frac{\partial E}{\partial f_i}$, which represents the gradient flow from the upper layer, as well as the factor of $\frac{\partial f_i}{\partial W_{in}}$, which denotes the local gradient. However, the training process becomes inefficient, since weights cannot be updated if one of these gradient factors becomes vanishingly small or decay towards zero. With the delayed feedback system embedded, our analog DFR computing system not only reduces the implementation complexity but also overcomes the drawback of the fading memory limitation. Within the reservoir layer, our analog DFR computing system utilizes the dynamic delayed feedback loop to formulate a persistent memory, as depicted in Figure 7. This memory is integrated with the delayed signal from the previous state and the new incoming input data. Such functionality enables the new incoming input data to carry the information from its previous state(s). The expression of N^{th} output, S_N , could be simplified as

$$S_N = \sum_{i=1}^N I_N(i) \cdot A v^{N-N(i)}, \quad (9)$$

where $I_N(i)$ is the N^{th} input data, Av is the finite gain of the gain regulator within the reservoir layer, and $N(i)$ determine the N^{th} I/O term. Because of the property of persistent memory, our

Fig. 8. Key components within the reservoir layer.

Fig. 9. (a) Design scheme of the nonlinear node; (b) simulated DC characteristic of nonlinear transformation.

analog DFR computing system has the capability to carry out complex nonlinear computations that depend on the past information, such that the system would learn from its previous states.

4 CIRCUIT IMPLEMENTATION

Our analog DFR computing system was designed with careful consideration of its system robustness and noise immunity. In this section, the detailed design considerations of three key components—a nonlinear node, a dynamic delayed feedback loop, and a delay calibration module—are presented, as illustrate in Figure 8.

4.1 Nonlinear Transformation

We observed that the function of nonlinear transformation is a key parameter that plays an important role in the performance of the high-dimensional projection. The simplified design scheme of the nonlinear node is depicted in Figure 9(a).

The nonlinear transformation is achieved by sensing the potential level of the input signal, V_{in} , through the nonlinear transformer, as shown in the dashed box in Figure 9(a). During the reset period, the input signal is charged up to V_{DD} , such that $(V_{DD} - V_{in}) < V_{th,p}$, where $V_{th,p}$ is the threshold potential of the input trigger (SW_1 and SW_2). Such high-voltage signal disenable the input trigger to discharge the nonlinear transformer, thereby, the output signal is reset to 0. In the decision-making period, the nonlinear transformer continuously senses the input signal and charges up its intrinsic capacitor to regulate the biasing voltage, V_{NT} , of M_5 ; consequently, the drain-to-source voltage of M_5 , V_{DS5} , increases quickly and eventually reaches its saturation region, thus, the diode-connected structure of M_4 fully enables the output current mirror ($M_{4,5,6}$) to generate the corresponding current signal. Meanwhile, the feedback current mirror ($M_{1,2,3}$) forms a positive feedback loop to generate a high voltage at V_{FB} to disenable the input trigger,

Fig. 10. Design scheme of I&F delay neuron.

such that the nonlinear node remains resting until the arrival of next input data. As such, the nonlinear transformation of one input signal is completed.

The nonlinearity of the transfer function is mainly controlled by the intrinsic coefficient, η_{NT} , of the nonlinear transformer. To facilitate the accurate high-dimensional projection, a large η_{NT} is required to achieve a high nonlinearity of the transfer function; however, large η_{NT} often results in a lower computational efficiency with a larger design area because of the increment of the parasitic capacitor. Considering the tradeoff between the computational efficiency, system robustness and design area, the accuracy of the system is maintained by implementing a large η_{NT} .

4.2 I&F Delay Neuron and Dynamic Delayed Feedback Loop

With the delay embedded, the dynamic behavior of our analog DFR computing system varies accordingly as the delay changes. Such property can be realized by the dynamic delayed feedback loop, which is built by multiple I&F delay neurons. The design scheme of a single I&F delay neuron is depicted in Figure 10.

The I&F delay neuron utilizes the capacitor-sensing technique from the classic LIF neuron model (Koch and Segev 2000). During the operation, the sensing capacitor (C_S) continuously tracks the current that is generated from the delay calibration module and charges up its potential. When the voltage potential across the sensing capacitor, $V_{th(c)}$, exceeds the firing threshold voltage of input transistors ($M_{1,2}$), two cascading inverters ($M_{3,4}$ and $M_{7,8}$) fire a spike as output. Meanwhile, the positive feedback loop ($M_{5,6}$) induces a high voltage at V_{reset} , such that the sensing capacitor can be fully discharged by the resetting switch (SW_2). As such, the firing process for one output spike is accomplished.

In the I&F delay neuron, the delay time constant, τ_{delay} , can be regulated by the integrating time over the sensing capacitor, which is controlled by the current from the delay calibration module within the reservoir layer. The delay time constant could be expressed as

$$\tau_{delay} = C_S \cdot \frac{V_{th(c)}}{I_{cal}}, \quad (10)$$

where C_S is the sensing capacitance, $V_{th(c)}$ is the firing threshold voltage of input transistors, and I_{cal} is the calibration current. In general, the mathematical model of the delay time constant is represented by values of resistance and capacitance. In our I&F delay neuron, the input resistance, R_{in} , is equivalent to $\frac{V_{th(c)}}{I_{cal}}$, thus, the delay time constant, τ_{delay} , could be simplified as

$$\tau_{delay} = C_S \cdot R_{in}. \quad (11)$$

Fig. 11. Design area illustration in (a) *RC* delay model and (b) designed I&F delay neuron.

Fig. 12. Dynamic delayed feedback loop.

Compared to the traditional *RC* delay model, as illustrated in Figure 11(a), which is built upon a large design area of resistors and capacitors, our designed I&F delay neuron has the capability to process the spiking signal directly with a superior dynamic range of controllable delay time and a small design area.

The dynamic delayed feedback loop, which is constructed upon multiple I&F delay neurons, as illustrated in Figure 12, is implemented by using the output spike train from the previous neuron as the clock triggering signal to its following. For instance, when the temporal spike train is generated from the first delay neuron, τ_1 , within the delay loop, it resets the following delay neuron, τ_2 ; meanwhile, the voltage potential across the sensing capacitor of τ_2 starts to charge up. Over the period of τ_{delay} , τ_2 fires a spike as output, which results in the input spike train at a given delay time.

4.3 Delay Calibration

To facilitate the functionality of dynamic behavior regulation, from periodic to chaotic or vice versa, the delay calibration module, as depicted in Figure 13(a), is implemented within the reservoir layer.

The delay calibration module utilizes the voltage-to-current conversion technique from the conventional trans-conductance amplifier. The current mirror array, as shown in the dashed box in Figure 13(a), keeps sensing the variation of the input signal, V_{cal} , and linearly generates the corresponding calibration current, I_{cal} , which could be expressed as

$$I_{cal} = (V_{cal} - V_N) \cdot G_m, \quad (12)$$

where V_N is the feedback signal from the current mirror, and G_m is the trans-conductance of the operational amplifier (op-amp). Within the dynamic delayed feedback loop, each I&F delay neuron

Fig. 13. Design scheme of (a) delay calibration module and (b) operational amplifier.

Fig. 14. Die photo of (a) whole DFR chip; (b) single analog DFR computing system module; (c) reservoir layer.

has equidistant delay time constant, thus, the current mirror array within the delay calibration module is designed to achieve identical calibration currents.

5 PERFORMANCE EVALUATION

Our analog DFR computing system was successfully fabricated using the standard GlobalFoundries (GF) 130nm CMOS technology. In this section, the detailed computational performance of three key functions—nonlinear projection, delay regulation, and dynamic behavior—are evaluated. Furthermore, the system robustness was studied and analyzed through the Monte Carlo simulations in the Cadence Virtuoso platform.

5.1 Analog DFR Chip

Figure 14 shows a die photo of the fabricated analog DFR computing system. The design area of the whole analog DFR chip occupies $1.5\text{mm} \times 1.5\text{mm}$ while each analog DFR module takes up to $175\mu\text{m} \times 56\mu\text{m}$. The chip comprises 12 analog DFR modules, which are separated into three major groups. Two separated function modules generate reference voltages to trigger the signal regulator and integrator within the input layer of the system. The filtering capacitor array is implemented to eliminate the signal-vibration between each analog DFR and function module.

Fig. 15. (a) Measured characteristic of nonlinear transformation; (b) simulated nonlinearity of transfer function with varied nonlinear coefficients.

5.2 Results Discussion

5.2.1 Nonlinear Transformation. The characteristic of the nonlinear transformation is measured by varying the input signal from GND to V_{DD} , as plotted in Figure 15(a). The mathematical operation of DDE system is separated into two modules, namely the nonlinear transformation and dynamic delay regulation, as illustrated in Figure 3. As depicted in Figure 15(a), the measured nonlinear correlation between the input and output is successfully achieved. Figure 15(b) demonstrates the simulated nonlinear transfer function with a various nonlinear coefficient, η_{NT} , as discussed in Section 4. As the nonlinear coefficient increases, the nonlinearity of the transfer function changes accordingly.

In the circuit implementation, the robustness of the nonlinear transformation could be enhanced by increasing the nonlinear coefficient within the nonlinear node because of the channel length modulation, which could be written as

$$Error \% = \frac{|\Delta L|}{L}, \quad (13)$$

where L is the designed channel length of transistors within the nonlinear transformer, and ΔL represents the difference between the fabricated length due to the process variation and the ideal designed length. Since the error percentage is inverted-proportional to the designed channel length, the system is more reliable and robust with the implementation of a large nonlinear coefficient. However, increasing the nonlinear coefficient often results in a lower computational efficiency, since the parasitic capacitance increases accordingly. In our analog DFR computing system, to facilitate the high-accuracy computation, such nonlinear mapping system is designed with a low operating frequency.

5.2.2 Dynamic Delay Regulation. The dynamic range of the delay time, τ_{delay} , of a single I&F delay neuron is characterized by the calibration current, I_{cal} , as plotted in Figure 16. With testing results on the delay characteristic, it can be observed that the designed I&F delay neuron has a superior dynamic range of delay times from 20ns to 1.38 μ s.

To acquire 1.38 μ s delay time via the traditional RC delay model, such system requires a 100k Ω resistor and a 13.8pF capacitor, which results in a large design area. Due to the capacitor-sensing technique within the I&F neuron model, a single I&F delay neuron is capable of achieving a delay time constant as large as 1.38 μ s with merely 54.72nA of calibration current; the equivalent input resistance, R_{in} , reaching 10.96M Ω . This large delay time constant can be achieved with a 150fF capacitor with the total design area of 19 μ m $\times$ 16 μ m, as illustrated in Figure 11. It is reasonable to

Fig. 16. Characteristic of the dynamic delayed feedback loop in (a) regulation of delay time constant; (b) demonstration of analog spiking signal.

Fig. 17. Simulated system robustness with process variation in (a) nonlinear transformation; (b) dynamic delay regulation.

conclude that our I&F delay neuron indeed achieves a superior dynamic range of delay time with a small design area and processes the spiking signal directly.

5.2.3 System Robustness and Performance. The system robustness of the nonlinear transformation and the dynamic delay regulation were evaluated through the Monte-Carlo simulation by introducing the process variation with 500 sampling points at the room temperature. In this task, the input signal of the nonlinear node and I&F delay neuron was set to be 0.8V and 250nA, respectively. As depicted in Figure 17, the average offset for the nonlinear node and I&F delay neuron is 6mV and 1.99ns with a standard deviation of 8mV and 2.83ns, respectively. Simulated results indicate that 100% of data points lie within a band of 3σ region.

The temperature variation was analyzed by simulating the temperature from 0°C to 60°C in the Cadence Virtuoso platform, as plotted in Figure 18. With the simulated results in temperature variation, the average error rates remain below 3.5% for both nonlinear transformation and dynamic delay regulation if the temperature is below 32°C. As the temperature increases, error rates increase up to 15.5% and 8.6% for the nonlinear transformation and dynamic delay regulation, respectively.

The measured normalized errors of the nonlinear node and I&F delay neuron were evaluated through 20 sampling points at room temperature, as depicted in Figure 19. In this task, the input signal of the nonlinear node and I&F delay neuron was set to be 0.8V and 250nA, respectively. The normalized errors were evaluated by examining the difference between the predicted value from the simulation and the actual value from the measurement. With the testing results in the

Fig. 18. Simulated system robustness with temperature variation in (a) nonlinear transformation; (b) dynamic delay regulation.

Fig. 19. Normalized error in (a) nonlinear transformation; (b) dynamic delay regulation.

measured system robustness, the average normalized error for the nonlinear transformation and dynamic delay regulation is 1.97% and 3.48%, respectively.

5.2.4 Dynamic Behavior of analog DFR Computing System. From solutions of differential equations, the Lyapunov stability analysis shows that the dynamic behavior of a given system could be achieved with a simple delay, and it has been proven that the best performance of the reservoir computing is found when operating at the edge of chaos regime (Hegger et al. 1998; Ikeda and Matsumoto 1987; Legenstein and Maass 2007a). In Bai and Yi (2018), Bai et al. (2018), and Junges and Gallas (2012), phase portraits are used to demonstrate the transition of intrinsic dynamic behavior from a theoretical point of view. A phase portrait is a graphics tool to visualize how the solutions of a given system of DDE would behave in the long run.

As shown in Figure 20, plotted phase portraits were obtained from measurements using two signals within the reservoir layer where one of them was collected with time delay. By varying the total delay time within the feedback loop, the dynamic behavior of the system changes from order to chaotic behavior as the delay increases. When the total delay time within the system is maintained around $1\mu\text{s}$, the delayed signal, $I(t - \tau)$, repeated traces of its initial path even in a long run, resulting in the periodic behavior plotted in Figure 20(b). As the total delay time within the system increases to $1.4\mu\text{s}$, the delayed signal diverges from its initial path without off-tracking from the equilibrium point after a long run, resulting in the edge of chaotic behavior plotted in Figure 20(c).

Fig. 20. Measured phase portrait of dynamic system in (a) $T = 0.64\mu s$; (b) $T = 1\mu s$; (c) $T = 1.4\mu s$.

Fig. 21. (a) Error and power analysis in terms of sampling frequency; (b) power distribution of the analog DFR computing system with the sampling frequency at 1MHz.

From testing results on dynamic behaviors, one can see that our analog DFR computing system does go through a range of dynamic behaviors. It is reasonable to conclude that our fabricated analog DFR chip successfully implemented the desired functionality of delay and richness of dynamic behavior. This indicates that the our analog DFR computing system closely mimics neurological systems with conduction time delay.

5.2.5 Power Consumption. In general, the total power used by CMOS IC is constituted by two parts, the static power and the dynamic power (Sah 1991). The static power defines the power used when transistors are not in the switching process. The static power is independent to the speed, since input signals remain unchanged. However, the change of input signals charges or discharges the parasitic capacitor of a transistor and its corresponding loading capacitor, thereby, the dynamic power changes accordingly. Since the rate of charging and discharging processes is proportional to the sampling frequency, the dynamic power is dependent on the frequency, which could be expressed as

$$P_{dynamic} = (C_p + C_l) \cdot f \cdot V_{DD}^2, \quad (14)$$

where C_p is the parasitic capacitance of a transistor, C_l is loading capacitance, and f is the sampling frequency.

Table 1. The Design Specification of our Fabricated Analog DFR Chip and State-of-the-art DFR Designs

	(Amil, Cabeza, and Marti 2015)	(Appeltant et al. 2011)	(Soriano et al. 2015)	This Work
Design Structure	DFR	DFR	DFR	DFR
Implementation	PCB	FPGA/PCB	FPGA/PCB	IC
Technology	/	/	/	CMOS 130nm
Signal Pattern	Mixed-signal	Mixed-signal	Mixed-signal	Pure Analog
Supply Voltage	$\pm 10V$	1.8V~2.5V	1.8V~2.5V	1.2V
Design Area	/	/	/	$175 \times 56 \mu m^2$

As depicted in Figure 21(a), the measured average error rate and simulated power consumption versus the sampling frequency are plotted. When the sampling frequency is less than 5MHz, the average error rate is below 1% with the average power consumption of $529 \mu W$. However, as the sampling frequency increases, the average error rate and power consumption increase up to 3.5% and $578 \mu W$, respectively. The power distribution of our analog DFR computing system with the sampling frequency at 1MHz is illustrated in Figure 21(b). The overall power consumption reaches $526 \mu W$; the nonlinear node requires 68% of the total power consumption, while the temporal encoder and the dynamic delayed feedback loop take up to 13% and 19% of the total power consumption, respectively. Table 1 summarizes the design specification of the fabricated analog DFR chip and other state-of-the-art reservoir computing designs.

5.2.6 NARMA10 Benchmark. To evaluate the precision of the designed analog DFR computing systems, a chaotic time series prediction benchmark, the tenth-order nonlinear autoregressive moving average system (NARMA10), is carried out, which is governed by the following equation:

$$o(t) = 0.3 \cdot o(t) + 0.05 \cdot o(t) \cdot \sum_{i=0}^9 o(t-i) + 1.5 \cdot d(t-9) \cdot d(t) + 0.1, \quad (15)$$

where $d(t)$ is the random input signal at time t , and $o(t)$ is the output signal.

In this task, a total of 10,000 sampling points were generated for training and testing phases. One hundred samples were used for the initialization, 5,900 samples were used for the training, and 4,000 samples were used for the testing. The prediction error was examined using the normalized root-mean-square error (NRMSE) and compared to state-of-the-art reservoir computing models. During the training process, output weights were trained by minimizing the deviation between predicted and target outputs. Both training and testing errors were achieved by the NRMSE, which could be defined as

$$NRMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{n\sigma_y^2}}, \quad (16)$$

where y_i is the predicted output, $\hat{y}_i$ is the target output, n is the total number of samples, and σ_y^2 is the output variance.

The NRMSE of training and testing was then evaluated through the Equation (16). NRMSE results along with the comparison to state-of-the-art DFR designs are tabulated in Table 2. It can be observed that the NRMSE from our DFR computing system exhibits a 36%–85% reduction on the error rate compared to state-of-the-art reservoir computing modules. Experimental result of predicted output signals against target outputs with our DFR computing system is plotted in Figure 22.

Table 2. Performance Comparison in Different Models

	Model	Training Error(NRMSE)	Testing Error(NRMSE)	Error RateReduction
(Rodan and Tino 2011)	ESN	/	0.1075	36.5%
(Appeltant et al. 2011)	DFR	/	0.15	54.5%
(Goudarzi et al. 2014)	DFR	0.065	0.464	85.3%
(Ortín and Pesquera 2017)	DFR	/	0.17	59.8%
This Work	DFR	0.0849	0.0683	/

Fig. 22. Target signals versus predicted signals for NARMA10 benchmark (demonstrated with first 100 samples in testing).

6 CONCLUSIONS

In this article, we presented an analog IC implementation of the DFR computing system along with its corresponding measurement results. By mimicking dynamic aspects of the neural information processing, our analog DFR computing system is built upon a temporal encoder to achieve spiking based information processing and a nonlinear transfer function to project input patterns onto higher-dimensional spaces. Our analog DFR computing system was fabricated with the standard GF 130nm CMOS technology where high energy efficiency and small design area are accomplished. As demonstrated in measurement results, our analog DFR chip exhibits richness in dynamic behaviors, signaling the successful implementation of the designed DFR architecture that imitates neurological systems with merely $526\mu\text{W}$ of power consumption. The measured normalized error of the proposed nonlinear transformation and the dynamic delay regulation could achieve an error rate as low as 1.97% and 3.48%, respectively. The performance of our DFR computing system was also evaluated through the NARMA10 chaotic time series prediction benchmark. Compared to state-of-the-art reservoir computing modules, our DFR computing system exhibits a 36%–85% reduction on the error rate. To the best of our knowledge, our work represents the first analog IC implementation of a DFR computing system.

ACKNOWLEDGMENTS

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL. Authors thank Dr. Bryant Wysocki and Mr. Clare Thiem, both at AFRL/RITB, for reviewing the material and providing feedback.

REFERENCES

- Abdulrahman Alalshekmubarak and Leslie S. Smith. 2014. On improving the classification capability of reservoir computing for arabic speech recognition. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 225–232.
- Miquel L. Alomar, Vincent Canals, Nicolas Perez-Mora, Víctor Martínez-Moll, and Josep L. Rosselló. 2016. FPGA-based stochastic echo state networks for time-series forecasting. *Comput. Intell. Neurosci.* 2016 (2016), 1–14.
- Pablo Amil, Cecilia Cabeza, and Arturo C. Marti. 2015. Exact discrete-time implementation of the Mackey–glass delayed model. *IEEE Trans. Circ. Syst.* 62, 681–685.
- Lennert Appeltant, Miguel Cornelles Soriano, Guy Van der Sande, Jan Danckaert, Serge Massar, Joni Dambre, Benjamin Schrauwen, Claudio R. Mirasso, and Ingo Fischer. 2011. Information processing using a single dynamical node as complex system. *Nature Commun.* 2, 468.
- Kangjun Bai, Jialing Li, Kian Hamedani, and Yang Yi. 2018. Enabling a new era of brain-inspired computing: Energy-efficient spiking neural network with ring topology. In *Proceedings of the 55th Annual Design Automation Conference*. ACM, 166.
- Kangjun Bai and Yang Yi. 2018. A path to energy-efficient spiking delayed feedback reservoir computing system for brain-inspired neuromorphic processors. In *Proceedings of 19th International Symposium in Quality Electronic Design (ISQED'18)*.
- Rezaul Begg and Joarder Kamruzzaman. 2005. A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *J. Biomech.* 38 401–408.
- Anne Beuter, Jacques Bélair, and Christiane Labrie. 1993. Feedback and delays in neurological diseases: A modeling study using dynamical systems. *Bull. Math. Biol.* 55, 525–541.
- Kristine E. Callan, Lucas Illing, Zheng Gao, Daniel J. Gauthier, and Eckehard Schöll. 2010. Broadband chaos generated by an optoelectronic oscillator. *Phys. Rev. Lett.* 104, 113901.
- Qiuwen Chen, Qinru Qiu, Hai Li, and Qing Wu. 2013. A neuromorphic architecture for anomaly detection in autonomous large-area traffic monitoring. In *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 202–205.
- Myonglae Chu, Byoungho Kim, Sangsu Park, Hyunsang Hwang, Moongu Jeon, Byoung Hun Lee, and Byung-Geun Lee. 2015. Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. *IEEE Trans. Industr. Electron.* 62, 2410–2419.
- Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham China, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, and Shweta Jain. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99.
- Miyuru Dayarathna, Yonggang Wen, and Rui Fan. 2016. Data center energy consumption modeling: A survey. *IEEE Commun. Surveys Tutor.* 18, 732–794.
- Sukru Burc Eryilmaz, Siddharth Joshi, Emre Neftci, Weier Wan, Gert Cauwenberghs, and H.-S. Philip Wong. 2016. Neuromorphic architectures with electronic synapses. In *Proceedings of the 17th International Symposium on Quality Electronic Design (ISQED'16)*. IEEE, 118–123.
- Steve K. Esser, Rathinakumar Appuswamy, Paul Merolla, John V. Arthur, and Dharmendra S. Modha. 2015. Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems*. NIPS, 1117–1125.
- Arfan Ghani. 2010. Neuro-inspired speech recognition based on reservoir computing. In *Advances in Speech Recognition*, InTech.
- C. Lee Giles and Tom Maxwell. 1987. Learning, invariance, and generalization in high-order neural networks. *Appl. Optics* 26, 4972–4978.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep Learning*. MIT Press, Cambridge.
- Alireza Goudarzi, Peter Banda, Matthew R. Lakin, Christof Teuscher, and Darko Stefanovic. 2014. A comparative study of reservoir computing for temporal signal processing. Arxiv preprint Arxiv:1401.2224.
- Alireza Goudarzi, Matthew R. Lakin, and Darko Stefanovic. 2014. Reservoir computing approach to robust computation using unreliable nanoscale networks. In *Proceedings of the International Conference on Unconventional Computation and Natural Computation*. Springer, 164–176.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 369–376.
- Lyudmila Grigoryeva, Julie Henriques, Laurent Larger, and Juan-Pablo Ortega. 2015. Optimal nonlinear information processing capacity in delay-based reservoir computers. *Sci. Rep.* 5, 12858.
- Hermann Haken. 2002. Brain dynamics (synchronisation and activity patterns in pulse-coupled neural nets with delays and noise). *Springer Series in Synergetics*.
- Simon S. Haykin. 2001. *Neural Networks: A Comprehensive Foundation*, Tsinghua University Press.

- Rainer Hegger, Martin J. Bünner, Holger Kantz, and Antonino Giaquinta. 1998. Identifying and modeling delay feedback systems. *Phys. Rev. Lett.* 81, 558.
- Xavier Hinaut and Peter F. Dominey. 2012. On-line processing of grammatical structure using reservoir computing. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 596–603.
- Miao Hu, John Paul Strachan, Zhiyong Li, Emmanuelle M. Grafals, Noraica Davila, Catherine Graves, Sity Lam, Ning Ge, Jianhua Joshua Yang, and R. Stanley Williams. 2016. Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication. In *Proceedings of the 53rd Annual Design Automation Conference*. ACM, 19.
- Kensuke Ikeda and Kenji Matsumoto. 1987. High-dimensional chaotic behavior in systems with time-delayed feedback. *Physica D: Nonlin. Phenom.* 29, 223–235.
- Herbert Jaeger. 2001a. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *German National Research Center for Information Technology GMD Technical Report* 148 13, Bonn, Germany.
- Herbert Jaeger. 2001b. *Short Term Memory in Echo State Networks*, GMD-Forschungszentrum Informationstechnik.
- Herbert Jaeger. 2007. Echo state network. *Scholarpedia* 2, 2330.
- Herbert Jaeger and Harald Haas. 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication’. *Science* 304, 78–80.
- Yu Jin, Qingchun Zhao, Hongxi Yin, and Hehe Yue. 2015. Handwritten numeral recognition utilizing reservoir computing subject to optoelectronic feedback. In *Proceedings of the 11th International Conference on Natural Computation (ICNC’15)*. IEEE, 1165–1169.
- Leandro Junges and Jason A. C. Gallas. 2012. Intricate routes to chaos in the Mackey–glass delayed feedback system. *Phys. Lett. A*, 376, 2109–2116.
- Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. James Hudspeth. 2000. *Principles of Neural Science*, McGraw-Hill, New York.
- Laszlo B. Kish. 2002. End of Moore’s law: Thermal (noise) death of integration in micro and nano electronics. *Phys. Lett. A* 305, 144–149.
- Christof Koch and Idan Segev. 2000. The role of single neurons in information processing. *Nature Neurosci.* 3, 1171–1177.
- Shumin Kong and Masahiro Takatsuka. 2017. Hexpo: A vanishing-proof activation function. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN’17)*. IEEE, 2562–2567.
- Laurent Larger and John M. Dudley. 2010. Nonlinear dynamics: Optoelectronic chaos. *Nature* 465, 41–42.
- Laurent Larger, Miguel C. Soriano, Daniel Brunner, Lennert Appeltant, Jose M. Gutiérrez, Luis Pesquera, Claudio R. Mirasso, and Ingo Fischer. 2012. Photonic information processing beyond Turing: An optoelectronic implementation of reservoir computing. *Optics Express* 20, 3241–3249.
- Robert Legenstein and Wolfgang Maass. 2007a. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Netw.* 20, 323–334.
- Robert Legenstein and Wolfgang Maass. 2007b. What makes a dynamical system computationally powerful. *New Directions in Statistical Signal Processing: From Systems to Brain*. 127–154.
- Xiaowei Lin, Zehong Yang, and Yixu Song. 2011. Intelligent stock trading system based on improved technical analysis and echo state network. *Expert Syst. Appl.* 38, 11347–11354.
- Mantas Lukoševicius. 2012. Reservoir computing and self-organized neural hierarchies. Jacobs University, Bremen.
- Wolfgang Maass, Thomas Natschläger, and Henry Markram. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* 14 2531–60.
- M. Mayberry. 2017. Intel’s new self-learning chip promises to accelerate artificial intelligence. Intel.
- Carver Mead. 1990. Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636.
- Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, and Yutaka Nakamura. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673.
- John G. Milton. 2015. Time delays and the control of biological systems. *IFAC PapersOnLine* 48, 87–92.
- Dharmendra S. Modha. 2017. Introducing a brain-inspired computer. Retrieved from <http://www.research.ibm.com/articles/brain-chip.shtml>.
- D. J. Myers and R. A. Hutchinson. 1989. Efficient implementation of piecewise linear activation function for digital VLSI neural networks. *Electron. Lett.* 25, 1662.
- Nasser M. Nasrabadi. 2007. Pattern recognition and machine learning. *J. Electron. Imag.* 16, 049901.
- Silvia Ortín and Luis Pesquera. 2017. Reservoir computing with an ensemble of time-delay reservoirs. *Cogn. Comput.* 9, 3 (2017), 327–336.
- G. Overton. 2014. Photonic reservoir computing—A new tool for speech recognition. Retrieval from <https://www.laserfocusworld.com/articles/2014/09/photonic-reservoir-computing-a-new-tool-for-speech-recognition.html>.

- Yvan Paquot, Francois Duport, Antoneo Smerieri, Joni Dambre, Benjamin Schrauwen, Marc Haelterman, and Serge Massar. 2012. Optoelectronic reservoir computing. *Sci. Rep.* 2, 287.
- Ali Rodan and Peter Tino. 2011. Minimum complexity echo state network. *IEEE Trans. Neural Netw.* 22, 131–144.
- Chih-Tang Sah. 1991. *Fundamentals of Solid State Electronics*, World Scientific Publishing Company, Singapore.
- Benjamin Schrauwen, Michiel D’Haene, David Verstraeten, and Jan Van Campenhout. 2008. Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Netw.* 21, 511–523.
- Benjamin Schrauwen and Robert Legenstein. 2010. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation* 22, 5 (2010), 1272–1311.
- Benjamin Schrauwen and Dirk Stroobandt. 2008. Using reservoir computing in a decomposition approach for time series prediction. In *Proceedings of the European Symposium on Time Series Prediction (ESTSP’08)*. Multiprint Oy/Otamedia, 149–158.
- Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. 2007. An overview of reservoir computing: Theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*. 471–482.
- Mark D. Skowronski and John G. Harris. 2007. Automatic speech recognition using a predictive echo state network classifier. *Neural Netw.* 20 414–23.
- Miguel C. Soriano, Silvia Ortin, Lars Keuninckx, Lennert Appeltant, Jan Danckaert, Luis Pesquera, and Guy Van derSande. 2015. Delay-based reservoir computing: Noise effects in a combined analog and digital implementation. *IEEE Trans. Neural Netw.d Learn. Syst.* 26, 388–393.
- Kristof Vandoorne, Wouter Dierckx, Benjamin Schrauwen, David Verstraeten, Roel Baets, Peter Bienstman, and Jan Van Campenhout. 2008. Toward optical signal processing using photonic reservoir computing. *Optics Express* 16, 11182–11192.
- David Verstraeten, Benjamin Schrauwen, Michiel d’Haene, and Dirk Stroobandt. 2007. An experimental unification of reservoir computing methods. *Neural Netw.* 20, 391–403.
- David Verstraeten, Benjamin Schrauwen, and Dirk Stroobandt. 2005. Reservoir computing with stochastic bitstream neurons. In *Proceedings of the 16th Annual Prorisc Workshop*. 454–459.
- Qian Wang, Youjie Li, and Peng Li. 2016. Liquid state machine based pattern recognition on FPGA with firing-activity dependent power gating and approximate computing. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS’16)*. IEEE, 361–364.
- Jianhong Wu. 1998. Symmetric functional differential equations and neural networks with memory. *Trans. Amer. Math. Soc.* 350, 4799–4838.
- Daoyi Xu and Zhichun Yang. 2005. Impulsive delay differential inequality and stability of neural networks. *J. Math. Anal. Appl.* 305, 107–120.
- Shimeng Yu, Yi Wu, Rakesh Jeyasingh, Duygu Kuzum, and H.-S. Philip Wong. 2011. An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electron. Dev.* 58, 2729–2737.
- Zhang Yong, Peng Li, Yingyezhe Jin, and Yoonsuck Choe. 2015. A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Trans. Netw. Learn. Syst.* 26, 2635–2649.
- Chenyuan Zhao, Jialing Li, Lingjia Liu, Lakshmi Sravanthi Koutha, Jian Liu, and Yang Yi. 2016. Novel spike based reservoir node design with high performance spike delay loop. In *Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication*. ACM, 14.
- Chenyuan Zhao, Yang Yi, Jialing Li, Xin Fu, and Lingjia Liu. 2017. Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors. *IEEE Trans. Very Large Scale Integr. Syst.* 25, 2193–2205.

Received December 2017; revised June 2018; accepted August 2018