Journal of Applied Physics

Reservoir computing with a single delay-coupled non-linear mechanical oscillator

Cite as: J. Appl. Phys. **124**, 152132 (2018); https://doi.org/10.1063/1.5038038
Submitted: 30 April 2018 • Accepted: 03 August 2018 • Published Online: 16 October 2018

Guillaume Dion, Salim Mejaouri and Julien Sylvestre

COLLECTIONS

F This paper was selected as Featured

openaccess

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Machine learning based on reservoir computing with time-delayed optoelectronic and photonic systems
Chaos: An Interdisciplinary Journal of Nonlinear Science **30**, 013111 (2020); https://doi.org/10.1063/1.5120788

Physical reservoir computing based on spin torque oscillator with forced synchronization
Applied Physics Letters **114**, 164101 (2019); https://doi.org/10.1063/1.5081797

Photonic neuromorphic information processing and reservoir computing
APL Photonics **5**, 020901 (2020); https://doi.org/10.1063/1.5129762

AIP Publishing

Reservoir computing with a single delay-coupled non-linear mechanical oscillator

Guillaume Dion, Salim Mejaouri, and Julien Sylvestre[a)]
Department of Mechanical Engineering, Université de Sherbrooke, Sherbrooke J1K 2R1, Canada

(Received 30 April 2018; accepted 3 August 2018; published online 16 October 2018)

Reservoir computing was achieved by constructing a network of virtual nodes multiplexed in time and sharing a single silicon beam exhibiting a classical Duffing non-linearity as the source of non-linearity. The delay-coupled electromechanical system performed well on time series classification tasks, with error rates below 0.1% for the 1st, 2nd, and 3rd order parity benchmarks and an accuracy of (78 ± 2)% for the TI-46 spoken word recognition benchmark. As a first demonstration of reservoir computing using a non-linear mass-spring system in MEMS, this result paves the way to the creation of a new class of compact devices combining the functions of sensing and computing.

https://doi.org/10.1063/1.5038038

I. INTRODUCTION

The discovery of faster numerical methods to adjust the parameters of artificial neural networks (aka training) has led in the last decade to a resurgence of interest in using these networks to implement complex functions, which are constructed from a finite (albeit large) training set of examples, and which can exhibit impressive generalization capabilities when applied to inputs which were not part of the training set.[1] Recurrent neural networks (RNN) are especially efficient at modeling time dependent data,[2,3] as they feed information from the "top" parts of the network at a given iteration to "lower" parts of the network at the next iteration. They are universal computers (in the sense described in Ref. 4) but are considered difficult to train.[5] Under certain conditions, RNN form a so-called reservoir computer (RC), in which case the weights of the recurrent network are initialized randomly and are left untrained, while the weights of a simple output layer are adjusted to train the network for a desired output.[6]

The concept of RC has led to interesting numerical applications[7–9] but, more importantly, it has been the trigger for a variety of hardware implementations of computing systems with functionalities similar to those of artificial neural networks. In these hardware implementations, the dynamics of a physical system are (often) left untrained and provide memory and non-linear computing capabilities to a simple, trainable output system. Hardware RC results have been reported for optical systems,[10–12] mechanical devices,[13,14] memristor arrays,[15] and spintronic devices,[16] for instance.

Hardware RC can be interesting as computing systems if they provide significant gains over conventional computers in terms of speed[17,18] or energy efficiency.[6,19,20] In addition, RC implemented in micro-mechanical devices, in particular, could serve the dual purpose of sensing and of computing for various force stimuli,[21] to create new classes of devices in distributed sensing or in control applications where the device dimensions or energy requirements are a limiting factor. As a step toward the realization of such devices combining mechanical sensing with neural-like computing, we describe in this paper a micro-fabricated silicon beam, with non-linear dynamics which can be harnessed to perform non-trivial computing tasks.

The dynamics of our oscillating beam, which exhibit a classical Duffing non-linearity (Sec. II), are coupled to a feedback mechanism in a scheme[22] that has been used to create hardware RC implementations from various relatively simple non-linear systems.[16,17,23,24] The resulting MEMS RC is trained (Sec. III) to process streams of bits and to classify spoken words, to demonstrate a small, energy-efficient computing device, which encodes information in the mechanical domain and therefore has the potential to function as both a sensor and a computer (as discussed in Sec. IV).

II. METHODS

A. The non-linear node

The non-linear expansion of input data into a higher dimensional state space is a necessary property of RC.[25] In this experiment, it is achieved using the non-linear dynamics of a single clamped-clamped silicon beam at large oscillation amplitudes. Figure 1 shows a typical device, which incorporates two mechanically coupled beams, although this study focuses on a single one of them (highlighted in red). The other beam is not actuated and the coupled system will be the subject of future work on a hybrid reservoir architecture.[26]

The device is micro-fabricated on a (100) p-doped silicon on insulator (SOI) substrate with a sacrificial oxide thickness of 1.5 μm. Both device and handle layers have a resistivity of (0.003 ± 0.002) Ωm. The device layer thickness of 10 μm defines the width of the beam (normal to its displacement), while its length and in-plane thickness were chosen to be 500 μm and 4 μm, respectively, in order to satisfy previously

[a)]julien.sylvestre@usherbrooke.ca

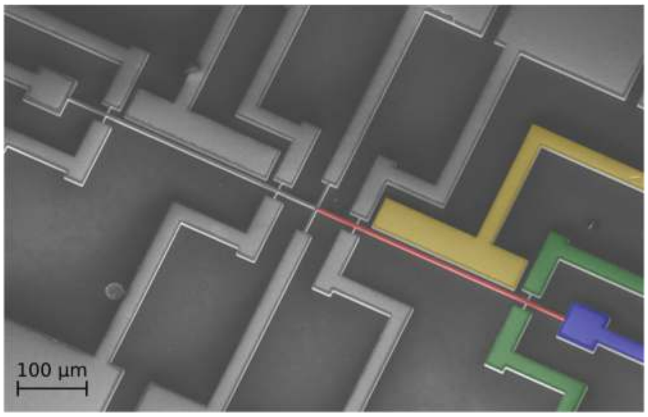

FIG. 1. SEM image of the device. The two coupled silicon beams (with one of them highlighted in red) can be actuated and read out individually by independent drive electrodes (yellow), piezoresistive gage pairs (biased using the electrical traces highlighted in green), and measure pads (blue).

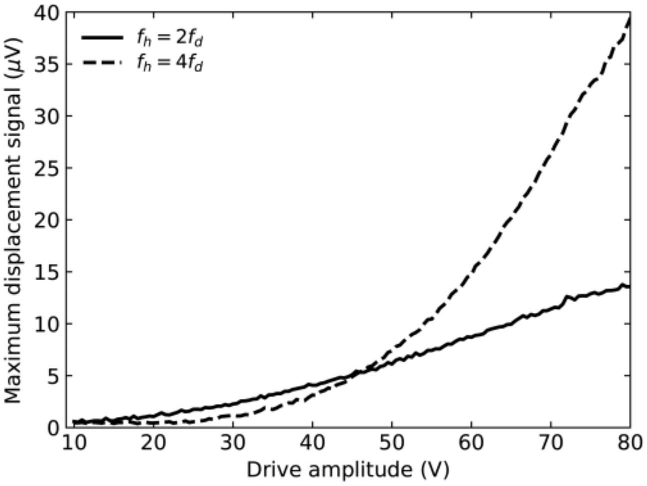

FIG. 2. Maximum of the frequency response for various drive amplitudes, measured at twice ($f_h = 2f_d$) and four times ($f_h = 4f_d$) the drive frequency. The spectrum is otherwise flat within the detection resolution, except for a large feedthrough signal at the drive frequency.

identified requirements.[21] These requirements include a relatively low quality factor (~100) so that the beam "forgets" about previous inputs rapidly enough, a high natural frequency ($f_n > 10^5$ Hz) so that the characteristic time of the oscillator is short enough to allow a reasonable processing speed (as discussed in Sec. IV), and a sufficiently non-linear behavior, insuring rich reservoir dynamics. This last prerequisite demands for a high value of β ($\sim 10^{23}\,\mathrm{Hz}^2\,\mathrm{m}^{-2}$), the coefficient controlling the amount of non-linearity in the restoring force for a Duffing oscillator whose displacement is described by

$$\ddot{x}(t) = -\frac{\omega_0}{Q}\dot{x}(t) - \omega_0^2 x(t) - \beta x^3(t) + F(t), \tag{1}$$

where $F(t)$ is the force per unit mass driving the beam, $\omega_0 = 2\pi f_0$ is the natural angular frequency of the oscillator in its linear regime, and Q is the quality factor of the MEMS. A value of f_0=228 kHz was obtained for the beam studied here, and its quality factor, measured in the time domain, is 100 ± 3, independent of the oscillation amplitude.

Oscillations of the beam are excited in the plane of the substrate by a 300 μm long drive electrode (yellow in Fig. 1) placed 6 μm away from the beam. For the electrostatic actuation scheme employed here, the force acting on the beam is proportional to the square of the driving voltage. Vibrations of the beam are thus solicited at twice the sinusoidal drive frequency f_d, which allows one to filter out the feedthrough signal stemming from parasitic capacitive coupling between the drive electrode and the measurement electrode.

Reproducing a strategy that has proven successful,[27] identical strain gage (1.2 μm large by 12 μm long) pairs are also patterned on the device to allow for the piezoresistive transduction of the mechanical vibrations, measured at the mid-voltage point between the two gages (blue pad in Fig. 1). The gages have a nominal resistance of $R_g = 50\,\Omega$ and are biased at ± 2.5 V (green traces in Fig. 1). When displaced, the beam acts as a lever and causes axial tension ($R_g \rightarrow R_g + \Delta R$) in one gage and compression ($R_g \rightarrow R_g - \Delta R$) in the other such that the combined effect on the readout voltage V_o is additive:

$$V_o = V_B \frac{\Delta R}{R_g}, \tag{2}$$

where $V_B = 2.5$ V is the DC bias voltage.

Electrical connections to the device are established by wirebonding the chip to a printed circuit board, with the substrate grounded. The reason for grounding the handle layer is twofold: it negates the possibility of a charged substrate exerting a static downward force on the beam, modifying its dynamics, and it mitigates feedthrough by removing the main feedthrough pathway (constituted by the two large series capacitors established around the oxide layer by the drive electrode, handle layer, and sense pad). The piezoresistive displacement signal being of small amplitude ($< 100\,\mu$V) and riding on top of a relatively large feedthrough signal,

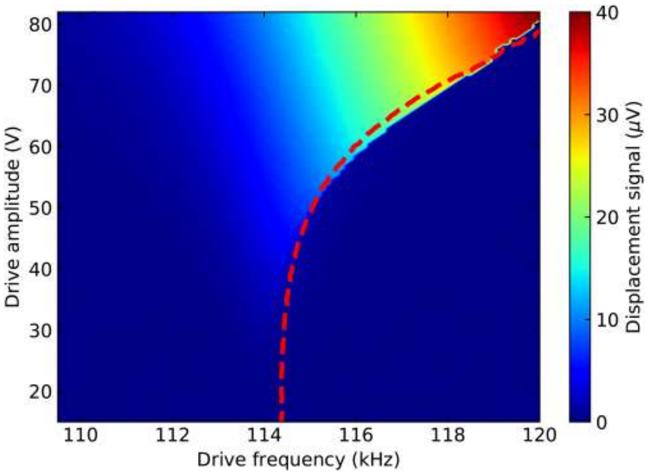

FIG. 3. Response of the system at four times the drive frequency ($f_h = 4f_d$) in the drive amplitude–drive frequency (A', f_d) plane. Colours indicate the measured signal V_o. The "backbone curve" (dashed red line) gives the natural frequency as a function of oscillation amplitude for a Duffing oscillator.

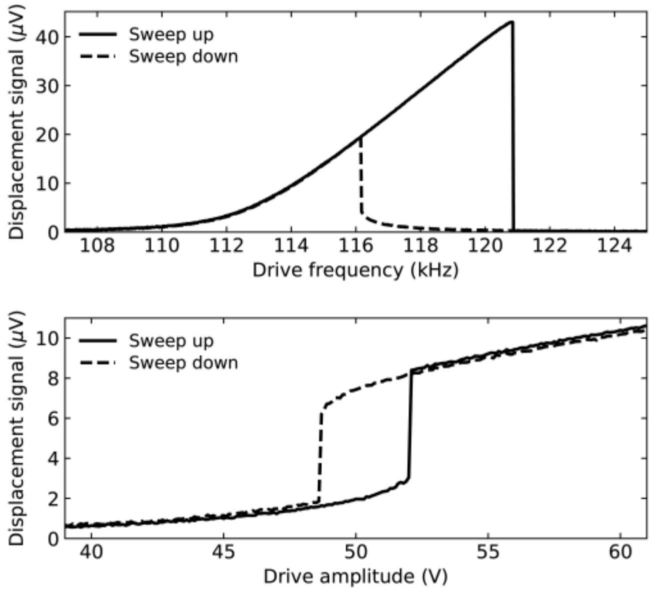

FIG. 4. Top: response of the beam to a drive frequency sweep for a fixed drive amplitude of 80 V. Bottom: response to a drive amplitude sweep for a fixed drive frequency of 115.1 kHz. All curves show the response at 4 times the drive frequency.

its proper digitization requires amplification with a gain of 80 dB, bandpass filtering with a center frequency of $f_h > f_d$ to attenuate the feedthrough component and reduce the noise contribution, and finally envelope detection to extract the oscillation amplitude.

Figure 2 shows the amplitude of the piezoresistive displacement signal V_o (before its amplification) as a function of the drive amplitude A', for the oscillator driven at its natural frequency f_n. For low drive amplitudes, the device behaves as a damped harmonic oscillator and the response is essentially contained in the component at twice the drive frequency. As the amplitude is increased, a component at four times the drive frequency begins to dominate, which indicates the onset of nonlinear effects. The response of the system to a drive frequency and amplitude sweep, filtered around this fourth harmonic, is shown in Fig. 3. It features a discontinuity for drive amplitudes larger than 45 V, which coincides with the amplitude at which the response at $4f_d$ starts to dominate over the component at $2f_d$ (see Fig. 2). The frequency at which the discontinuity appears increases as the drive amplitude is increased, which suggests that the beam indeed behaves as a stiffening Duffing oscillator. The red dashed line in Fig. 3 verifies the Duffing nature of the nonlinearity by modeling the location of the discontinuity (jump down) with[28]

$$f_{jump}^2 = \frac{3}{16\pi^2}\beta x_{max}^2 + f_0^2, \tag{3}$$

where x_{max} is the maximum displacement of the beam (right before the jump) which, for a linear transduction scheme, is proportional to the measured signal, and f_0 is the natural frequency of the oscillator in its linear regime. A value of $\beta = 1.9 \times 10^{23}\,\mathrm{Hz}^2\,\mathrm{m}^{-2}$ and a transduction coefficient of $90\,\mu\mathrm{V}/\mu\mathrm{m}$ reproduce the data well and are coherent with values obtained by finite element analysis and analytical methods for the dimensions of the beam.

Note that Fig. 3 was constructed by sweeping the drive signal from low to high frequency, increasing the drive amplitude between each sweep. Decreasing the drive voltage or amplitude during the sweeps changes the location of the discontinuity. This results in the hysteresis loops shown in Fig. 4, constructed by fixing the drive amplitude (frequency) and sweeping up and down the drive frequency (amplitude).

B. The reservoir

For RC to be realized on a single physical node, virtual nodes need to be created in order to generate a rich enough reservoir state, which can then be collapsed to a usable output by a simple linear combination of the individual node states from the reservoir state. This is done through the

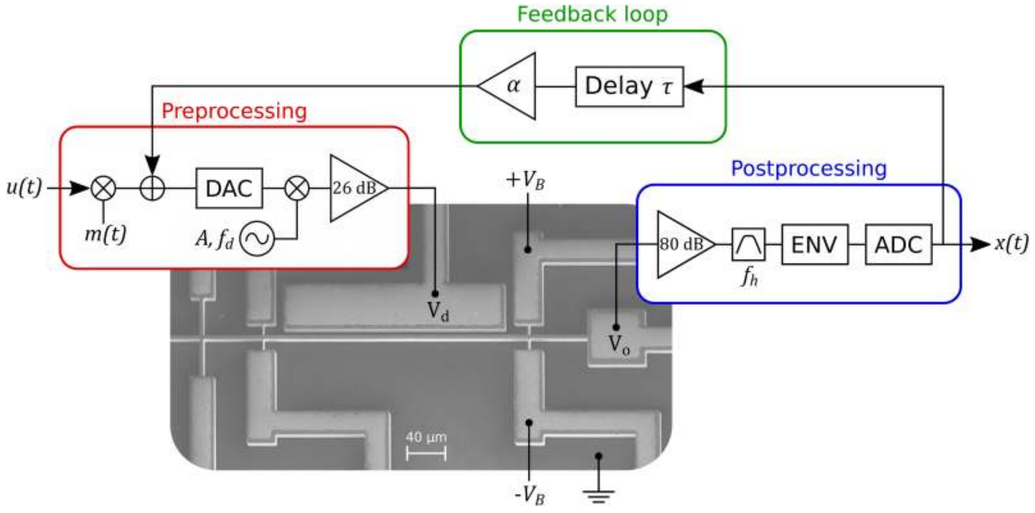

FIG. 5. Signal chain for the system and SEM image of the device. Before being supplied to the drive electrode, the input signal is preprocessed in the digital domain, goes through a digital to analog converter (DAC), modulates a sinusoidal drive of amplitude A and frequency f_d, and is amplified with a gain of 26 dB. Measuring the response involves amplifying the piezoresistive signal V_0 with a gain of 80 dB, bandpass filtering it around $f_h = 4f_d$, detecting its envelope (ENV) and digitizing it (ADC), before reinjecting it, with a delay τ and a gain α, in the preprocessing stage.

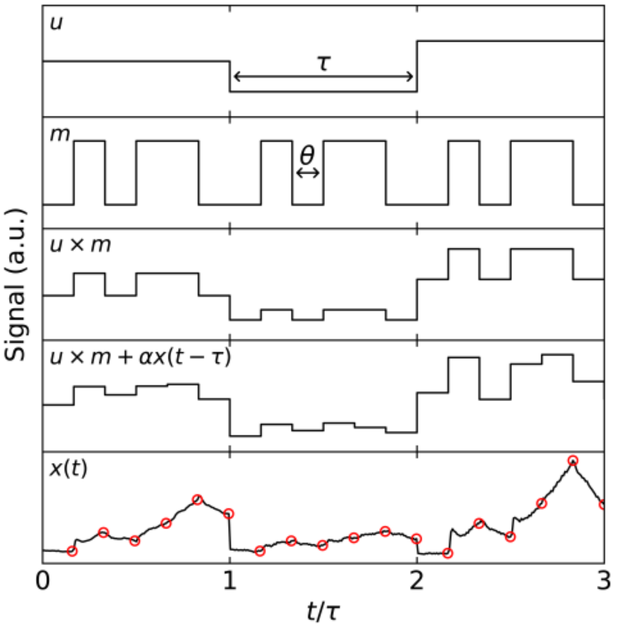

FIG. 6. From top to bottom: three timesteps of the sampled input signal $u(t)$, the associated periodic mask sequence $m(t)$ with $N = 6$, the preprocessed input $u(t) \times m(t)$, the preprocessed input with added feedback, and the reservoir response $x(t)$, where the red markers show how the signal is sampled to obtain the individual virtual node states.

increasingly common delay-based approach,[22] where virtual nodes are created by time-domain multiplexing: the input signal $u(t)$ is sampled and held for a time τ while a random binary mask of length N is applied at a rate θ^{-1} ($\tau = N\theta$) (see Figs. 5 and 6). The mask defines the input weights and keeps the system from saturating by regularly switching between two levels. For both tasks investigated in Sec. III, good results are obtained with mask values randomly chosen to be 0.45 or 0.70, with the input signal scaled such that $u(t)$ is limited to the range [0.60, 0.75]. Since coupling between adjacent virtual nodes is done entirely in the time domain, the reservoir interconnection sparsity can be tuned to some extent by varying the virtual node length θ: decreasing the ratio θ/T, where $T = 2Q/\omega_0$ is the decay time of the oscillator, makes the behavior of a given virtual node more dependent on the response of previous nodes since the signal is given less time to settle. For our experiments, best results were obtained for $\theta \sim T$ (see Sec. IV). The masked signal is then used to modulate the amplitude A of the drive at frequency f_d. After amplification with a gain of 26 dB to bring the drive amplitude from A to A', the resulting amplitude modulated waveform is used to drive the silicon beam into a non-linear state (see Figs. 3 and 4). The N virtual node states are then acquired by measuring the oscillation amplitude $x(t)$ at the end of each interval of length θ. This is done by bandpass filtering the amplified readout signal (gain of 80 dB) around $f_h = 4f_d$ and sampling its envelope at a rate θ^{-1} with a resolution of 16 bits. The result is a vector $\mathbf{X}$ of length N containing the output layer state at a given timestep of the input function. This readout, scaled by a factor α, is then added to the masked input for the next timestep, closing the feedback loop. With a proper choice of α, this feedback loop ensures that the RC has the fading memory[29] property, which means that perturbations propagate in the reservoir for a finite time and eventually vanish. The oscillator is thus driven by a voltage signal of the form:

$$V_d(t) = A'[u(t)m(t) + \alpha x(t - \tau) + 1] \cos(2\pi f_d t). \tag{4}$$

Finally, the readout vector is used to form a scalar output

$$y(t) = \mathbf{w}^T \mathbf{x}(t), \tag{5}$$

with $\mathbf{w}$ being a vector of weights. In order to obtain a desired output $y'(t)$ for a series of inputs $u(t)$, the weights are adjusted during a supervised training period, which consists of minimizing the mean squared error between y and y' by applying a ridge regression with Tikhonov regularization:

$$\mathbf{w} = \mathbf{y}'\mathrm{X}^T\left(\mathrm{X}\mathrm{X}^T + \lambda \mathrm{I}\right)^{-1}, \tag{6}$$

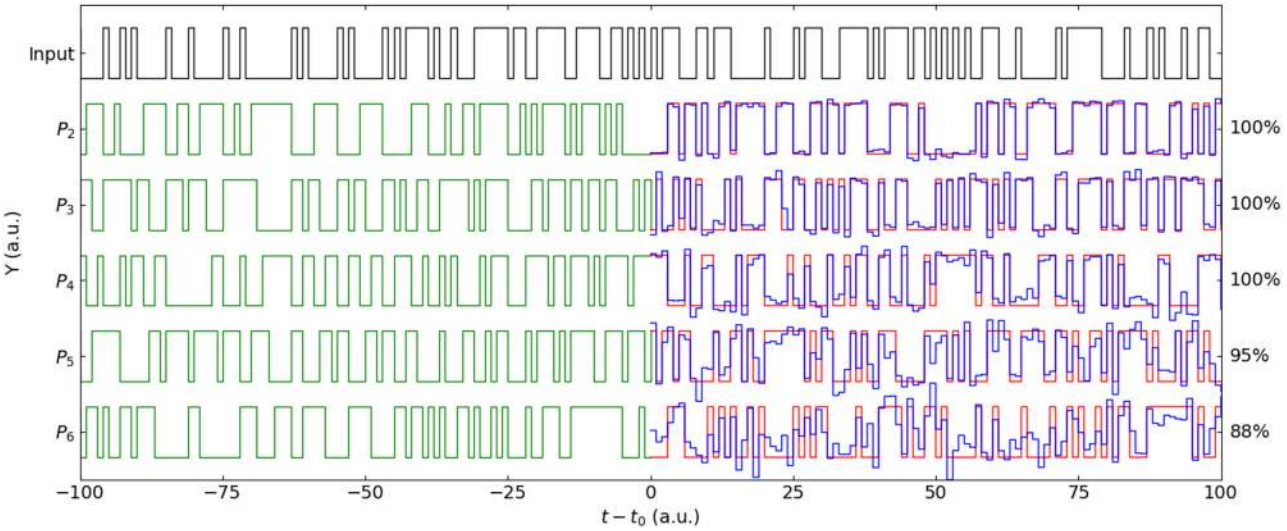

FIG. 7. Performance of the system for the parity benchmark. The topmost waveform is the input signal $u(t)$, randomly alternating between values of −1 and 1. The other waveforms show the prediction of the system $y(t)$ (blue) overlaid on the target $y'(t)$ (red) after the training phase (green) for $P_{n,0}$ with $n = 2, 3, 4, 5, 6$. Success rates for the individual tasks, evaluated for 1000 timesteps of the input function, are indicated on the right.

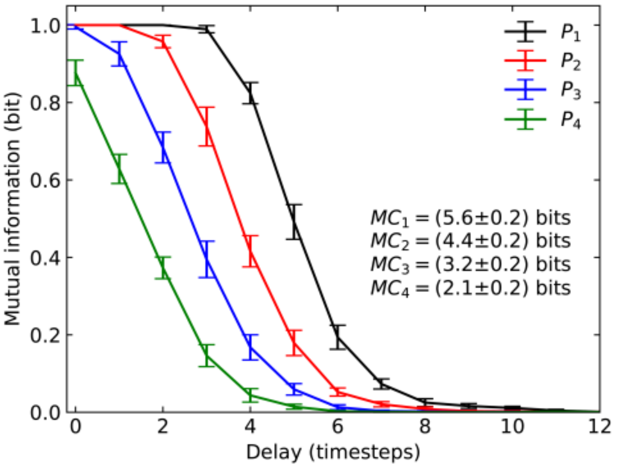

FIG. 8. The mutual information [Eq. (8)], computed for P_1, P_2, P_3, and P_4 and averaged over 10 different trials. Error bars show the standard deviation.

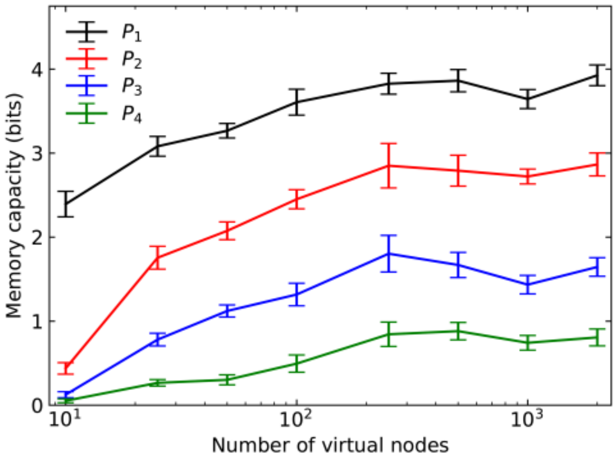

FIG. 9. Memory capacity for different reservoir sizes N, averaged over 10 trials, with the error bars showing the standard deviation.

where $\mathbf{y}'$ and X are the target vector and the data matrix, obtained by concatenating the discrete entries of $y(t)$ and $\mathbf{x}(t)$, respectively. The regularization parameter λ prevents overfitting, thus increasing robustness. A value of $\lambda \sim 10^{-1}$ V^2 improves the stability in our system.

III. RESULTS

A. Parity benchmark

The performance of the system was first assessed by means of the parity benchmark. For this benchmark, a random binary sequence $u(t)$, with values drawn from $\{-1, 1\}$, is fed to the network at a rate τ^{-1}, following the feedback system described in Sec. II B. The n^{th}-order parity function with delay,

$$P_{n,\delta}(t) = \prod_{i=0}^{n-1} u[t - (i + \delta)\tau], \qquad (7)$$

where $\delta \in \mathbb{Z}^*$ is the delay, is then computed by the reservoir computer. Equation (7) shows that this task requires the reservoir to store information about a nonlinear transform of previous inputs, except for the case $n = 1$, which is only a delay line (and the identity for $\delta = 0$). For $n > 1$, the parity function is not linear separable[30] and depends on the input at times $t - (i + \delta)\tau$, while the readout is a linear transform of the reservoir output. This benchmark thus tests the system's memory capacity as well as its nonlinear processing capability. To verify that a non-linear behavior of the physical node is necessary to implement the parity function, we ran a series of numerical simulations for this task with $\beta = 0$. For the linear case ($n = 1$), the simulations could achieve near perfect success rates, even for large values of δ. However, as soon as non-linear computations were introduced ($n \geq 2$), we could not find any parameter set allowing even the slightest success.

Figure 7 shows the system's prediction overlaid on the target [Eq. (7)] for $n = 2, 3, 4, 5$ and $\delta = 0$. Unless otherwise noted, all results for this benchmark were obtained by setting the acquisition parameters to $N = 400$, $\theta = 0.1$ ms, $\tau = 40$ ms, $A' = 82$ V, $f_d = 115.1$ kHz, and $\alpha = 0.5$. The success rate, displayed beside each trace of Fig. 7, represents the fraction of correct bits, computed by comparing the signs of the prediction and of the target during a test phase with 1000 input bits. Training is done offline after acquiring the reservoir states for the first 3000 timesteps. While thresholding the prediction for P_4 still gives a near bit-perfect reproduction of the target, the raw signal becomes more noisy. The prediction effectively becomes more uncertain as the difficulty and memory requirement of the task is increased from P_1 to P_6.

The mutual information,

$$MI_{n,\delta} = p_{n,\delta} \log_2(2p_{n,\delta}) + (1 - p_{n,\delta}) \log_2[2(1 - p_{n,\delta})], \qquad (8)$$

with $p_{n,\delta}$ being the success probability for the δ-delayed n^{th}-order parity function, can be used to estimate the memory capacity:[30]

$$MC_n = \sum_{\delta=0}^{\infty} MI_{n,\delta}. \qquad (9)$$

Figure 8 shows the mutual information obtained for $n = 1, 2, 3, 4$ as a function of the delay δ. In all cases, the

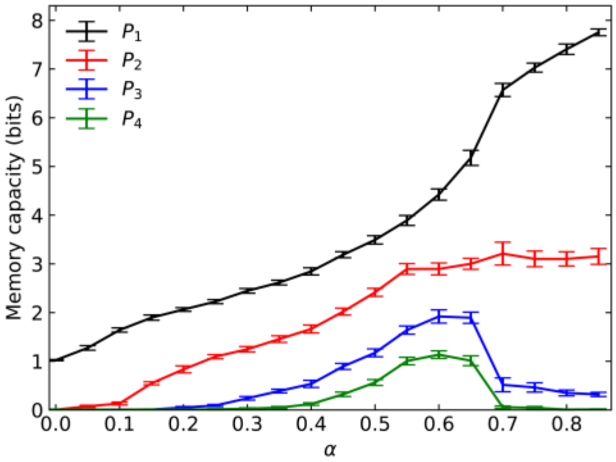

FIG. 10. Memory capacity for the parity tasks (P_1, P_2, P_3, P_4) for different values of α. Error bars are the standard deviation for 10 trials.

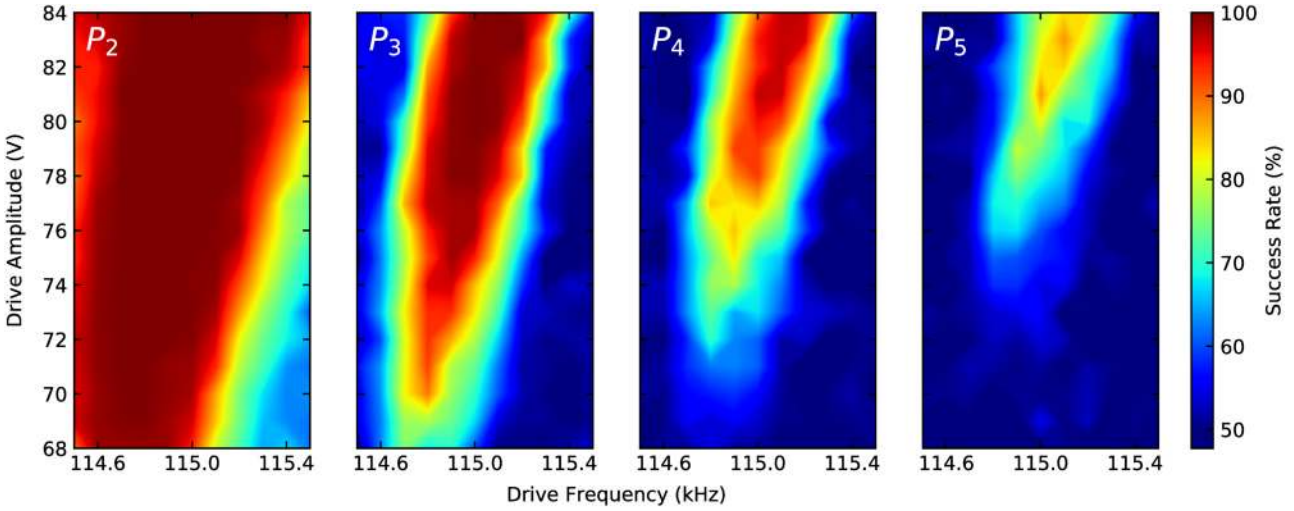

FIG. 11. Success rate for the parity function ($n = 2$ to $n = 5$, left to right) in the drive frequency–drive amplitude (f_d, A') plane. The fixed parameters are $N = 400$, $\theta = 0.1$ ms, $\tau = 40$ ms, and $\alpha = 0.5$.

mutual information becomes negligible for $\delta > 8$ such that the sum of Eq. (9) can be truncated at $\delta = 8$, yielding a linear memory capacity of (5.6 ± 0.2) bits (P_1), and non-linear memory capacities of (4.4 ± 0.2), (3.2 ± 0.2), and (2.1 ± 0.2) bits for P_2, P_3, and P_4, respectively, which is comparable to results for other types of reservoirs.[21,30,31]

Figure 9 shows the memory capacity as a function of the number of virtual nodes N, for a fixed value of $\theta = 0.1$ ms ($\tau = N\theta$ thus increases with N). With an increase in the number of virtual nodes, the readout function [Eq. (5)] has richer dynamics at its disposal for the reconstruction of the target such that a monotonic increase in memory capacity could be expected.[30] On the other hand, the fixed training set size of 3000 samples may not be sufficient for the larger reservoirs (the required training set length should scale up with N[32]), which could explain the saturation or even slight decline of the memory capacity observed for larger N. Indeed, a training set of 3000 examples was deemed sufficient (RC performance did not significantly increase past this point) for the $N = 400$ reservoir studied in the rest of this paper, but this has not been studied for larger reservoirs. Nevertheless, the decrease in memory capacity for more difficult tasks can be compensated up to a certain point by increasing the reservoir size. The curves presented in Fig. 9 show lower memory capacities than curves in Fig. 8 due to a less thorough parameter optimization. Detuning the system shifts these curves vertically but does not affect the general trend mentioned above.

As shown in Fig. 10, the memory capacity is also highly correlated to the feedback strength α. It drops rapidly for low feedback levels as the system has no way of storing information about previous inputs, while too much feedback also causes a decline in memory capacity. In the latter case, the reservoir does not exhibit the required fading memory[33] property and becomes unstable for $\alpha \geq 0.7$, affecting the results.

The success rate in the drive frequency–drive amplitude plane, interpolated from a 10 by 16 points grid search, is shown for P_2 to P_5 in Fig. 11. It can be seen that the area of significant success (>50% since a random guess will in average produce a success rate of 50%) is limited to a rather specific region of the parameter space that shrinks as the complexity of the task is increased from P_2 to P_5. Since the drive amplitude and the drive frequency (along with the feedback gain) essentially determine the level of non-linearity in our system, this shows that even though RCs, and more generally RNNs, can operate with a multitude of different non-linear nodes, fine tuning of the non-linearity can be necessary, especially for more demanding tasks.

Although Fig. 11 shows that at 115.1 kHz a drive amplitude of 82 V produces good results, this sinusoidal drive is modulated by the masked input such that the drive signal explores a certain range of voltages during normal operation. For this experiment, it was found that the reservoir performed well when the mask levels were adjusted such that the drive signal envelope spanned voltage values strictly larger than 49 V. From Fig. 4, it can be seen that the drive signal thus visits a non-linear part of the transfer function but never explores the lower branch of the hysteresis loop. Fully spanning the hysteresis loop likely causes the response to be chaotic, and various studies show that a RC operates best at the *edge of chaos*,[34] in the ordered phase.

B. Spoken word classification

The reservoir computer's ability to classify more complex signals was assessed using a spoken word classification task. For this task, a subset consisting of the numbers zero to nine spoken by 8 different females and 8 different males is drawn from the TI-46 corpus.[35] Utterances are then randomly selected from this subset and processed by the RC. Compared to usual approaches,[7] the preprocessing applied here is intentionally rudimentary to anticipate implementation in a simple system where sound pressure would be directly coupled to the oscillator. It involves only the normalization of each waveform by its standard deviation, low-pass filtering, and sampling of the envelope at twice the low-pass corner frequency of 30 Hz. The output layer state for a given utterance is obtained by integrating the consecutive readouts

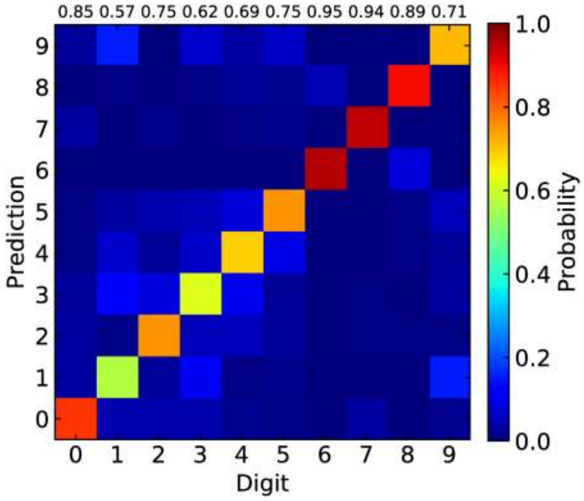

FIG. 12. The word classification benchmark confusion matrix shows the probability (colors) that a digit presented to the device (columns) is classified to a certain value (lines) by the reservoir computer. The numbers at the top of each column indicate the success probability for the individual digits. Acquisition parameters are $N = 400$, $\theta = 0.1\,\text{ms}$, $\tau = 40\,\text{ms}$, $A' = 78\,\text{V}$, $f_d = 115.0\,\text{kHz}$, and $\alpha = 0.6$.

for each individual virtual node during the whole utterance. During the training phase, a vector of weights is computed for each of the ten digits in the vocabulary. Each classifier is trained to output a value of 1 when presented with an utterance of the digit corresponding to its class, and -1 otherwise. Prediction during the test period is then carried out by choosing the digit with the highest classifier output value.

Figure 12 shows the confusion matrix for this task. The prediction probabilities were obtained with a ten-fold cross-validation: 5000 utterances were processed by the RC and divided into subgroups of 500. Each of these subgroups was used exactly once for testing, with the rest of the data used for training the RC each time. Correct classification is achieved in $(78 \pm 2)\%$ of occurrences, with digits such as $\{6,8\}$ being correctly classified more often and others such as $\{1,3,4\}$ being harder to discriminate. While much lower error rates are common in the literature,[10,16,17,36] they are invariably associated with a preprocessing of the utterances that often involves some sort of spectral analysis. Our intentional lack of an elaborate preprocessing scheme certainly calls for a more thorough parameter optimization in order to obtain lower error rates.

IV. DISCUSSION

While the use of delay systems can simplify physical reservoir implementation of RC, it has the drawback of lowering the computation rate. The equivalent N-node traditional reservoir can process the same input signal at N times the rate of the delay-coupled reservoir because of its parallel processing configuration.

For the device studied here, low error rates are achieved for $\theta \geq 0.1\,\text{ms}$, which is comparable to the oscillator characteristic time $T = 2Q/\omega_0 = 0.13\,\text{ms}$. This differs from the optimal value of $\theta \sim T/5$ presented elsewhere.[22] While longer values of θ tend to slightly increase the success rate, they also reduce the processing rate. For $\theta = 0.1\,\text{ms}$ and $N = 200$, which give adequate performance in our system, the sample clock of the input signal cycles at a rate $\tau^{-1} = N^{-1}\theta^{-1} = 50\,\text{Hz}$. While this only allows for a much slower classification rate than the 10^5 to 10^6 words per second rates[17,18] achieved with photonic reservoirs, it is sufficient to classify spoken words in real time, as only the low frequency components are necessary for this task.

The device demonstrated here occupies a relatively large surface ($\sim$0.1 mm^2), but it is a non-issue considering a single beam is used, and the additional space and power required for the external circuitry for the actuation, readout, and feedback loop is more concerning. Gains from size reduction instead impact the processing speed: by reducing the beam dimensions, its natural frequency increases while the quality factor can be kept relatively low by inducing additional viscous damping, resulting in an increase in clock rate since $\tau \propto \theta \sim T \propto Q/\omega_0$. The same piezoresistive transduction method could be applied[27] to a nanomechanical oscillator resonating at hundreds of MHz, allowing gains of many orders of magnitude in processing speed to be made.

Although fast high resolution ADCs and DACs are nowadays relatively cheap and accessible, a purely analog implementation of our system would have its appeals, such as being less demanding on the control circuitry, or even enabling the possibility of directly coupling different external inputs such as pressure or acceleration to the input layer of the reservoir. While both pre-processing and post-processing (given that the vector of weights is known) could easily be carried out entirely in the analog domain (which has been done for a photonic system[37]), the long delay required for the feedback loop remains nearly impossible to implement in analog electronics. Other means, such as a mechanical delay line, could eventually be employed to bypass this bottleneck.

Adjusting the system parameters can be a challenging task due to the high number of possible parameter combinations. Our approach of scanning the parameter space thus suffered from poor resolution and limited range. Nevertheless, once tuned the RC was found to be quite robust. For instance, the drive frequency can be incremented much more finely than the $\sim$100 Hz span insuring good performance for the P_4 task (see Fig. 11). Thus, fluctuations of device properties such as its natural frequency f_n and quality factor Q, which can happen over time or between nominally identical silicon beams due to fabrication tolerances, can be accounted for by adjusting the operating point as to negate their adverse effect on RC performance. The bottom panel of Fig. 6 shows that the measured signal contains white noise. This is essentially due to the thermal noise generated by the piezoresistive gages in a measurement bandwidth of 80 kHz, which limits the SNR to $\sim$ 30 dB. Yet, the results obtained here for two non-trivial tasks seem to indicate that the system is robust to this level of noise and thus well suited for its projected implementation as a smart sensor.

V. CONCLUSION

The Duffing non-linearity of a silicon beam, in conjunction with a delay-coupled reservoir architecture, allowed for the first experimental demonstration of a MEMS reservoir computer. The system is capable of predicting time series and classifying spoken words with relatively good accuracy, reliably predicting the parity of up to three consecutive input bits (the P_1 to P_3 tasks) with an error rate under 0.1%, and correctly classifying isolated spoken digits with an accuracy of (78 ± 2)%. While performance for the parity function exceeds numerically obtained results,[21] state-of-the-art performance for the spoken digit classification task would require a more thorough parameter optimization or a refined preprocessing scheme.

The system's memory capacity could be tuned to a certain extent by adjusting the feedback strength α (Fig. 10), although it could not be stabilized for delays longer than 8τ. Adding a second feedback loop with delay 2τ and strength $\alpha' < \alpha$ would likely improve the memory capacity.[38,39] An unsynchronized approach,[10] where a given node is fed back the response of a different node from the previous time step, could also potentially increase the performance of the RC. Other parameters could benefit from a more principled optimization.

This experiment has demonstrated that a MEMS oscillator can be used to construct a system capable of time series prediction and spoken word classification. It is a first step to conceiving computing substrates capable of also sensing their environment. More in-depth study of the rich non-linear dynamics of our oscillators will be the subject of a future communication.

ACKNOWLEDGMENTS

The authors thank the Ministère de l'économie, de la science et de l'innovation du Québec and the Natural Sciences and Engineering Research Council of Canada for financial support.

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**(7553), 436–444 (2015).

[2] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)* (Omnipress, 2011), pp. 1017–1024.

[3] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," preprint arXiv:1502.04623 (2015).

[4] K.-i. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," Neural Netw. **6**(6), 801–806 (1993).

[5] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Netw. **5**(2), 157–166 (1994).

[6] H. Jaeger, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**(5667), 78–80 (2004).

[7] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Isolated word recognition using a liquid state machine," in *Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN)* (D-Side Publishing, 2005), pp. 435–440.

[8] P. Buteneers, D. Verstraeten, B. Van Nieuwenhuyse, D. Stroobandt, R. Raedt, K. Vonck, P. Boon, and B. Schrauwen, "Real-time detection of epileptic seizures in animal models using reservoir computing," Epilepsy Res. **103**(2-3), 124–134 (2013).

[9] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-p. Martens, "Phoneme recognition with large hierarchical reservoirs," Adv. Neural Inf. Process. Syst. **23**, 2307–2315 (2010).

[10] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," Sci. Rep. **2**(1), 287 (2012).

[11] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," Opt. Express **20**(3), 3241 (2012).

[12] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," Nat. Commun. **5**, 3541 (2014).

[13] K. Caluwaerts, M. D'Haene, D. Verstraeten, and B. Schrauwen, "Locomotion without a brain: Physical reservoir computing in tensegrity structures," Artif. Life **19**(1), 35–66 (2013).

[14] J. Degrave, K. Caluwaerts, J. Dambre, and F. Wyffels, "Developing an embodied gait on a compliant quadrupedal robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 4486–4491.

[15] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, "Reservoir computing using dynamic memristors for temporal information processing," Nat. Commun. **8**(1), 2204 (2017).

[16] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, "Neuromorphic computing with nanoscale spintronic oscillators," Nature **547**(7664), 428–431 (2017).

[17] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," Nat. Commun. **4**, 1364 (2013).

[18] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," Phys. Rev. X **7**(1), 011015 (2017).

[19] C. Merkel, Q. Saleh, C. Donahue, and D. Kudithipudi, "Memristive reservoir computing architecture for epileptic seizure detection," Procedia Comput. Sci. **41**, 249–254 (2014).

[20] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," Front. Neurosci. **9**, 502 (2016).

[21] J. C. Coulombe, M. C. A. York, and J. Sylvestre, "Computing with networks of nonlinear mechanical oscillators," PLoS One **12**(6), e0178663 (2017).

[22] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," Nat. Commun. **2**, 468 (2011).

[23] M. L. Alomar, M. C. Soriano, M. Escalona-Moran, V. Canals, I. Fischer, C. R. Mirasso, and J. L. Rossello, "Digital implementation of a single dynamical node reservoir computer," IEEE Trans. Circuits Syst. II: Express Briefs **62**(10), 977–981 (2015).

[24] M. C. Soriano, S. Ortin, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. van der Sande, "Delay-based reservoir computing: Noise effects in a combined analog and digital implementation," IEEE Trans. Neural Netw. Learn. Syst. **26**(2), 388–393 (2015).

[25] M. Lukoševčius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Comput. Sci. Rev. **3**(3), 127–149 (2009).

[26] A. Rohm and K. Ludge, "Reservoir computing with simple oscillators: Virtual and real networks," e-print arXiv:1802.08590 (2018).

[27] M. Sansa, G. Gourlat, G. Jourdan, M. Gely, P. Villard, G. Sicard, and S. Hentz, "Compact heterodyne NEMS oscillator for sensing applications," Solid-State Electron. **125**, 214–219 (2016).

[28] B. Tang, M. Brennan, V. Lopes, S. da Silva, and R. Ramlan, "Using nonlinear jumps to estimate cubic stiffness nonlinearity: An experimental study," Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci. **230**(19), 3575–3581 (2016).

[29] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," IEEE Trans. Circuits Syst. **32**(11), 1150–1161 (1985).

[30] N. Bertschinger and T. Natschläger, "Real-time computation at the edge of chaos in recurrent neural networks," Neural Comput. **16**(7), 1413–1436 (2004).

[31] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. R. Mirasso, and J. M. Gutiérrez, "A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron," Sci. Rep. **5**(1), 14945 (2015).

[32] L. Appeltant, G. Van der Sande, J. Danckaert, and I. Fischer, "Constructing optimized binary masks for reservoir computing with delay systems," Sci. Rep. **4**(1), 3629 (2015).

[33] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks—with an Erratum note," Technical Report GMD Report 148 (German National Research Center for Information Technology), 2010.

[34] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," Neural Netw. **20**(3), 323–334 (2007).

[35] M. Liberman, R. Amsler, K. Church, E. Fox, C. Hafner, J. Klavans, M. Marcus, B. Mercer, J. Pedersen, P. Roossin, D. Walker, S. Warwick, and A. Zampolli, Ti 46-word LDC93S9. Web download (Linguistic Data Consortium, Philadelphia, 1993).

[36] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Opt. Express **20**(20), 22783 (2012).

[37] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," Sci. Rep. **6**(1), 22381 (2016).

[38] L. Appeltant, "Reservoir computing based on delay-dynamical Systems," Ph.D. thesis, Vrije Universiteit Brussel, Universitat de les Illes Balears, 2012.

[39] Y. Hou, G. Xia, W. Yang, D. Wang, E. Jayaprasath, Z. Jiang, C. Hu, and Z. Wu, "Prediction performance of reservoir computing system based on a semiconductor laser subject to double optical feedback and optical injection," Opt. Express **26**(8), 10211 (2018).