

Reservoir Computing in MEMS



Guillaume Dion, Anouar Idrissi-El Oudrhiri, Bruno Barazani,
Albert Tessier-Poirier, and Julien Sylvestre

Abstract This chapter explores the use of the Duffing nonlinearity and fast dynamics found in microelectromechanical beam oscillators for reservoir computing applications. General properties of MEMS are discussed, and the Duffing microscale beam characteristics are analyzed through analytical models and simulations. The reservoir computer is then constructed around a single such nonlinear oscillator through temporal multiplexing of the input and self-coupling via delayed feedback. The parameters of the resulting physical system are finally adjusted for optimal performance on computing the parity of a binary input stream, as well as on a spoken digit recognition task.

1 Introduction

Artificial intelligence (AI) and machine learning have progressed tremendously over recent years and are now the focus of an intense interest worldwide within many fields, with applications ranging from self-driving cars (Huval et al. 2015) to health monitoring systems (Witt et al. 2019). This rapid progress has occurred over only a few years and was driven by algorithmic advances and improvements in computing hardware (LeCun et al. 2015) that have resulted in much shorter training and validation times for AI systems. The expectation that better hardware could contribute to further improving AI systems currently fuels a large research effort to find new “computing substrates” for AI. While conventional AI is implemented with software running on general-purpose computers, it is widely accepted that much more efficient hardware implementations of AI must exist; our brains are an existence proof that some computing architectures can far exceed the density and energy efficiency of current microelectronics technology. We have published the first demonstration that microelectromechanical systems (MEMS) were an appropriate substrate for miniature, low energy consumption AI systems (Coulombe et al. 2017; Dion et al.

G. Dion · A. I.-E. Oudrhiri · B. Barazani · A. Tessier-Poirier · J. Sylvestre (✉)
Institut interdisciplinaire d’innovation technologique (3iT), Université de Sherbrooke,
Sherbrooke, Canada
e-mail: julien.sylvestre@usherbrooke.ca

2018). By exploiting the nonlinearity of microfabricated mechanical oscillators, our approach implements the concept of reservoir computing (RC) (Jaeger and Haas 2004) physically in MEMS. As MEMS can be fabricated to small dimensions and therefore have high resonance frequencies (up to the GHz van Beek et al. 2007), our approach has the potential to be used as a highly efficient electrical component to implement reservoir computing.

As MEMS are also the mainstream technology for many modern sensors (Khoshnoud and de Silva 2012), our work further paves the way to the development of a new class of smart sensors with built-in data processing capabilities. As an example, we have demonstrated a MEMS displacement sensor which implements reservoir computing in the mechanical domain (Barazani et al. 2019). As the sensor is moved randomly between two positions separated by $2\ \mu\text{m}$ at 20.8 Hz, it uses the nonlinear dynamics of its resonating mechanical structures to compute at every timestep when the position can change, if it had been in one of the two positions an even or an odd number of times over the last five timesteps. More recently, we have also demonstrated a MEMS accelerometer with similar neuromorphic computing capabilities (Barazani et al. 2020). By using the hardware implementation of reservoir computing in MEMS, these devices offer both sensing and non-trivial computing functions in small, highly integrated structures. We envision a number of applications for MEMS sensors integrating machine learning capabilities through our architecture (Sylvestre et al. 2018), especially in fields where small, energy-efficient systems are required, including the Internet of Things, autonomous systems, as well as mobile and wearable electronic devices.

This chapter provides a general overview of our neuromorphic computing MEMS technology. We start with an introduction to MEMS in Sect. 2, including the unique characteristics of microfabricated devices (relative to conventional devices) which are leveraged to implement computing functionalities. We discuss the modeling and analysis of nonlinear MEMS resonators (Sect. 3), leading to an example of a silicon beam design which has proven to be useful in experiments. Measurements of computing performances are presented in Sect. 4, together with observations on the tuning of the system parameters to optimize performance on different benchmark tasks.

2 Microelectromechanical Systems

Microelectromechanical systems (MEMS) are miniaturized machines able to sense or produce displacements at the micrometer and sub-micrometer scales, typically in the range of $0.1\ \mu\text{m}$ to $100\ \mu\text{m}$. MEMS devices comprise structures such as beams or membranes that are able to move relative to the substrate, providing actuation (MEMS actuators, e.g., micropumps) or detection capabilities (MEMS sensors, e.g., pressure or force meters). However, the design of miniaturized actuators and sensors requires some modifications if compared to the design of conventional machines. At the scale of MEMS structures, surface forces (such as electrostatic and adhe-

sion forces) are dominant compared to volumetric forces (such as gravitational and inertial forces). For instance, water surface tension forces can completely suppress MEMS mobility and are sometimes very difficult to avoid (Van Spengen et al. 2003). On the other hand, MEMS μm -dimensions allow them to be batch produced and assembled in the same chip as the electronic circuits, resulting in cheaper (lower cost per unit), faster, and more compact monolithic devices. Furthermore, MEMS tend to demonstrate higher sensitivity, faster response, and lower energy consumption than conventional mechanisms (Ananthasuresh 2012). MEMS applications can be quite diverse and include for example printers ink-jet nozzles, airbag sensors, mirror arrays in video projectors, focusing systems in smartphone cameras, and accelerometers in smartphones or personal fitness trackers.

2.1 MEMS Fabrication

In order to manufacture MEMS, traditional fabrication methods such as milling and extrusion are replaced by processes with increased precision and resolution, such as photolithography, chemical etching, and plasma etching. MEMS fabrication utilizes processes adapted from the microelectronics industry, which were mainly developed for the handling and processing of silicon substrates (Madou 1997; Liu 2006). This sort of manufacturing consists of multiple steps of deposition and etching of structural (usually silicon) and sacrificial (usually oxide) thin films. At the end of the process, the sacrificial material is removed to enable the structural parts to move relative to the substrate. One simple MEMS fabrication method is the direct etching of silicon on insulator (SOI) wafers. SOI wafers are standardized stacks composed of a device structural layer on the top, an oxide sacrificial layer in the middle, and a handle substrate layer at the bottom. The SOI MEMS fabrication process, illustrated in Fig. 1, can be roughly summarized into two main steps: (1) etching of the device layer, after it is patterned using photolithography; and (2) partial removal of the oxide layer granting motion to the structural parts, which remain connected to the substrate through the oxide that is not etched away (the anchors). The addition of electrical contacts to the fabrication flow allows the induction of motion by the application of electrical voltages. Likewise, measurements of voltage changes can be used to gage MEMS motion.

2.2 Sensing and Driving Methods

There are several techniques used to provide or detect microscale displacements in MEMS. The most common operating principles include electrostatic, electrothermal, piezoelectric, and piezoresistive (Liu 2006). In the great majority of MEMS devices, energy conversion involves an input or an output electrical signal, typically a voltage difference. The electrostatic and electrothermal phenomena, which produce

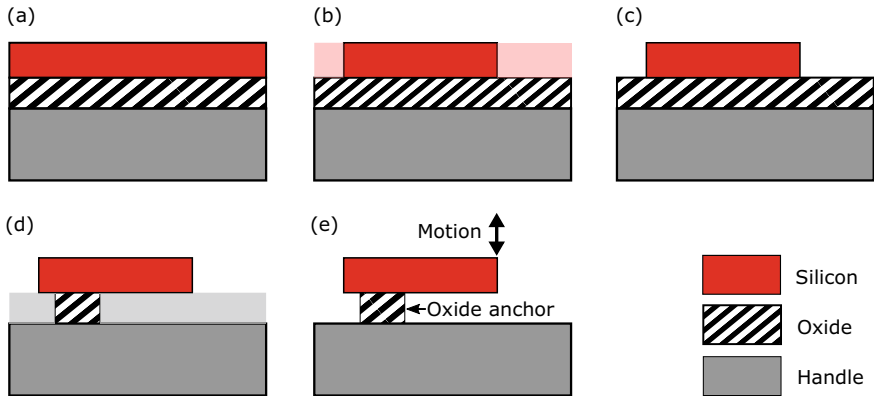


Fig. 1 **a** Initial SOI wafer. **b** Parts of the device layer are selected to be etched away after photolithography. **c** SOI wafer after the etching of the device layer. **d** Oxide areas to be removed in a selective etching procedure. **e** Final device after the oxide removal; the remaining silicon structure is anchored to the handle and is free to flex or move relative to the handle

forces that are usually negligible in conventionally sized mechanisms, are the most traditional configurations for driving and sensing in MEMS. Electrothermal MEMS, for example, produce motion through the thermal expansion of structures (usually beams) caused by Joule heating due to the application of voltage (Lai et al. 2004). In the case of electrostatic MEMS, motion is induced by electrostatic forces between microelectrodes separated by a small gap (Batra et al. 2007). Alternatively, changes in the gaps caused by an external force can be measured by the capacitance change between the electrodes.

MEMS accelerometers, some of the most commercially successful MEMS devices, may present a large variety of design types and working principles (Yazdi et al. 1998). Typically, external inertial forces displace an inertial mass that is suspended by compliant springs. This motion is then converted to an electrical signal that is proportional to the magnitude of this displacement. The transduction principle is usually capacitive (electrostatic) or piezoresistive (changes in the electrical resistance due to mechanical deformations). MEMS accelerometers can detect in-plane or out-of-plane forces depending on their design configurations (Fig. 2). Planar accelerometers commonly use an interdigitated configuration in order to increase the total capacitance and therefore the electrostatic sensitivity of the sensor. Higher sensitivity can also be achieved by reducing the accelerometer's natural frequency, which could be done by diminishing the suspension's stiffness. However, this also reduces the frequency response (bandwidth) of the sensor. Another practice to increase the sensitivity is to increase the signal-to-noise ratio by reducing the system's damping. This is usually done by etching holes along the proof mass or by operating the device under vacuum.

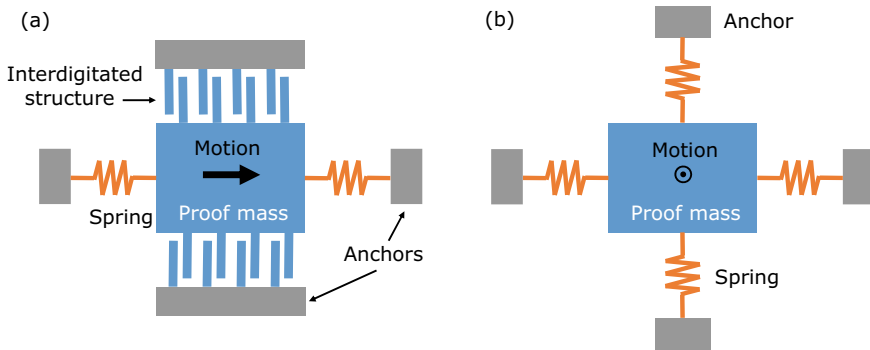


Fig. 2 Schematic illustration of two possible configurations of MEMS accelerometers: **a** the proof mass moves laterally, enabling the detection of in-plane accelerations, and **b** the proof mass moves up and down allowing the device to sense out-of-plane accelerations

2.3 MEMS Dynamics and Nonlinearity

MEMS devices are frequently designed to work in their dynamic regime, as it happens for example in MEMS resonators. As vibrating structures, MEMS exhibit much higher resonance frequencies compared to non-miniaturized mechanisms. This is because of their much higher k/m ratio, where k is the device elastic constant and m is its total mass. In MEMS resonators, shifts in the resonance frequency can be used to detect changes of different physical quantities, enabling the manufacturing of a variety of sensors such as pressure, force, and temperature sensors (Tilmans et al. 1992). The resonance frequency of MEMS tends to be very well defined (small bandwidth) due to their typically large quality factor (Q), which is a measure of the energy dissipation of oscillating structures. High values of Q indicate low energy dissipation, which leads to lower energy consumption, higher sensitivity, and lower noise. Energy dissipation can be classified as intrinsic or extrinsic (Ekinici and Roukes 2005). The former is associated with losses due to the material microstructure while the latter is mainly related to losses induced by the media surrounding the device. Extrinsic damping effects such as drag forces or squeezed films (when structures are too close) are usually the dominant sources of energy dissipation.

Another observed characteristic of MEMS resonators is their nonlinearity. Micromechanical oscillating structures demonstrate nonlinear behavior when driven above a certain critical amplitude (Husain et al. 2003; Ekinici and Roukes 2005). Frequently, the Duffing equation for nonlinear oscillators is used to describe the motion of MEMS resonators. Essentially, when oscillating at very large amplitudes (above critical), changes in the structure's stiffness result in nonlinear shifts of the resonance frequency. In the case of a clamped–clamped microbeam (i. e. both ends anchored) vibrating in its flexural mode, large driving amplitudes generate tensile forces that increase the beam stiffness resulting in an increase of its resonance frequency (Tilmans et al. 1992). The onset of nonlinearity in microstructures has been

explored elsewhere (Buks and Yurke 2006; Tadokoro et al. 2018). In this study, the nonlinearity of a clamped–clamped microbeam is used to set up a reservoir computing system able to perform non-trivial computing tasks.

3 Driven Oscillators with Duffing Nonlinearities

The Duffing model was first introduced to describe the hardening spring effect observed in mechanical systems (Duffing 1918). It is considered as one of the most common models used to describe the jump phenomenon observed in highly deformed mechanical resonators, where a slight change of forcing frequency leads to an abrupt discontinuous change in the steady-state amplitude (Guckenheimer and Holmes 2002; Kalmar-Nagy and Balachandran 2011). It keeps a simple mathematical form and accepts, under some approximations, analytical solutions (Ali 1995; Worden 1996).

3.1 Duffing Oscillator

Several micromechanical structures behave as nonlinear systems for high levels of excitation (Ekinici and Roukes 2005; Zaitsev et al. 2012). The Duffing equation with damping and external harmonic forcing is

$$\ddot{x} + \frac{\omega_0}{Q}\dot{x} + \omega_0^2x + \beta x^3 = A \cos(\Omega t), \quad (1)$$

where x , t , ω_0 , Q , A , Ω , and β are the displacement, time, undamped angular frequency, quality factor, excitation amplitude, angular excitation frequency, and cubic stiffness parameter, respectively. Dots denote derivatives with respect to time. As can be seen, Eq. (1) reduces to the forced damped linear oscillator when the anharmonic term is ignored ($\beta = 0$). An approximative solution for the position $x(t)$ can be obtained for small ω_0/Q , β , and A values and assuming the forcing is close to resonance, with $\Omega - \omega_0$ also small. Equation (1) can then be viewed as a perturbation of the autonomous harmonic oscillator. The perturbation technique known as “averaging” gives an approximative steady-state solution $x(t) = r \cos(\Omega t + \phi)$ where r is the oscillation amplitude and ϕ is the phase (see Guckenheimer and Holmes 2002 or Jan 2007 for details). Averaging gives a frequency response curve (Jan 2007),

$$(-2\omega_0(\Omega - \omega_0)r + \frac{3}{4}\beta r^3)^2 + 4(\omega_0^2/Q)^2 r^2 - A^2 = 0, \quad (2)$$

which can be solved for r .

Figure 3 shows the frequency response curve for $\beta = 0$ (from the exact solution of the linear problem) and curves from averaging for $\beta = \pm 0.05\text{Hz}^2/\text{m}^2$. The introduc-

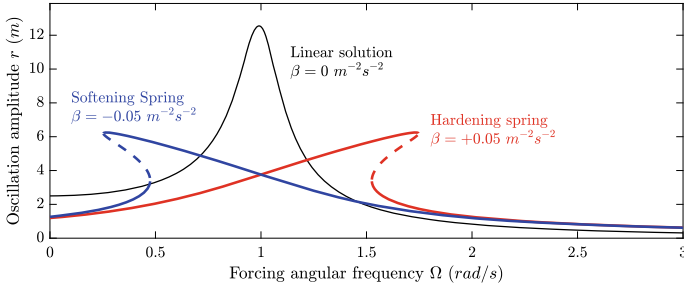


Fig. 3 Amplitude–frequency response curves for the linear system ($\beta = 0 \text{ Hz}^2/\text{m}^2$) from the exact solution and by averaging for stiffness parameter $\beta = \pm 0.05 \text{ Hz}^2/\text{m}^2$. Stable and unstable solutions are denoted as solid and dashed lines, respectively. The parameters used to construct these curves were $A = 2.5\text{m/s}^2$, $Q = 5$ and $\omega_0 = 1 \text{ rad/s}$

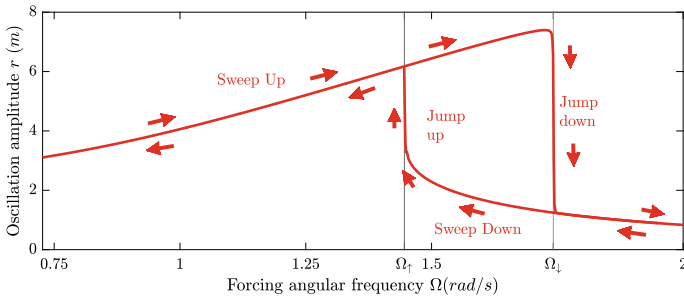


Fig. 4 Amplitude–frequency response curve obtained numerically by a sweep up followed by a sweep down of Ω ; the jump and the hysteresis are apparent. The parameter values used to construct these response curves were $A = 2.5 \text{ m/s}^2$, $Q = 5$, $\beta = 0.05\text{m}^{-2} \text{ s}^{-2}$, and $\omega_0 = 1 \text{ rad/s}$

tion of the cubic nonlinearity tilts the curve to the right for $\beta > 0$ (hardening spring) and to the left for $\beta < 0$ (softening spring). Furthermore, close to the peak, there are three possible solutions for a given Ω (two stable ones and an unstable one, denoted as a dashed line). Figure 4 shows numerical solutions to Eq. 1 for $\beta = 0.05\text{m}^{-2} \text{ s}^{-2}$, as the forcing angular frequency Ω is swept up and down. Once Ω is increased above the angular frequency of the peak Ω_\downarrow , the oscillation amplitude abruptly jumps to the lower branch, which is the only remaining solution. As Ω is reduced again, the oscillation amplitude follows the lower stable branch and jumps back to the upper branch once it reaches the unstable solution, at Ω_\uparrow . Since $\Omega_\downarrow > \Omega_\uparrow$, the nonlinear system exhibits hysteresis.

Figure 5 shows the phase-space plot of three distinct motion regimes. For low forcing amplitudes or when the anharmonic term is not taken into account in the Duffing equation (1), the motion of the resonator resembles a linear harmonic device where the response in phase-space is an ellipse. At intermediate forcing, the system can have more complex dynamics due to the stiffening characteristic of the resonator: there can be more than one harmonic component in the oscillator motion, as studied

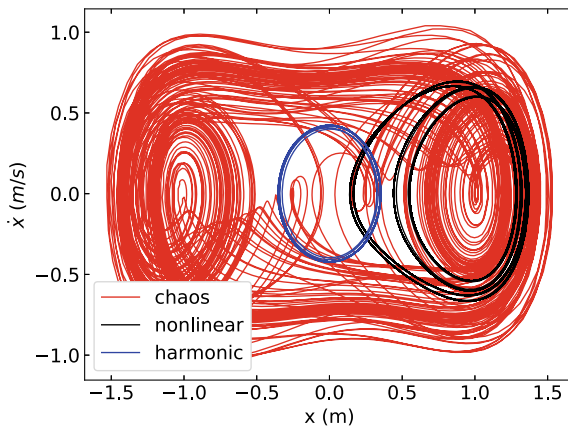


Fig. 5 Phase-space plots obtained from Eq. (1) for three motion conditions: harmonic oscillations with weak forcing $A = 0.2 \text{ m/s}^2$ and cubic stiffness corresponding to zero (blue line), moderate forcing $A = 0.29 \text{ m/s}^2$ with $\beta = 1 \text{ Hz}^2/\text{m}^2$ (black line), and chaotic oscillations at high forcing level $A = 0.5 \text{ m/s}^2$, $\beta = 1 \text{ Hz}^2/\text{m}^2$ (red line). The other parameters used to construct these curves were $Q = 3.33$, $\omega_0 = 1 \text{ rad/s}$, and $\Omega = 1.2 \text{ rad/s}$

in Kalmar-Nagy and Balachandran (2011). Large forcing amplitudes lead to a chaotic motion and the system becomes very sensitive to the initial conditions.

For nonlinear Duffing systems, sudden jumps in the resonance response are observed, as in Fig. 6. The jump frequency depends on the direction of the frequency sweep and the type of nonlinearity (softening or stiffening) (Malatkar and Nayfeh 2002). For lightly damped Duffing oscillator, Brennan et al. presented a simple approximated non-dimensional expression which gives the maximum oscillation amplitude r_{\max} at the jump frequency Ω_{\downarrow} (Brennan et al. 2008). The relationship between the jump-down frequency and the cubic stiffness can be written in a dimensional form as (Tang et al. 2016)

$$\Omega_{\downarrow}^2 = \frac{3}{4}\beta r_{\max}^2 + \omega_0^2. \quad (3)$$

Solving for r_{\max} gives the so-called “backbone curve” presented by the dashed line in Fig. 6. It can be used to predict the frequency response of the system (Cammarano et al. 2014; Arroyo and Zanette 2016).

3.2 Clamped–Clamped Beams

A clamped–clamped beam is an oscillator exhibiting an anharmonic behavior at higher excitation amplitudes. Multiple studies have demonstrated that the Duffing

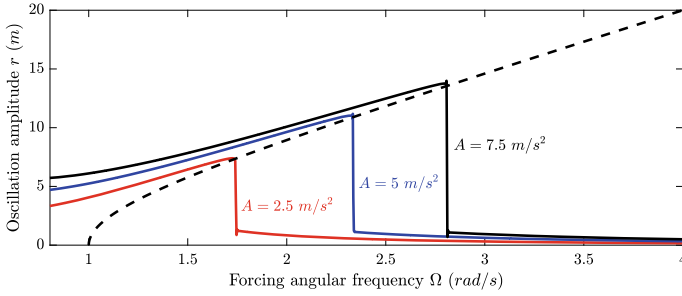


Fig. 6 Example of backbone curve (dashed line) for a Duffing oscillator swept up in forcing frequency. The parameter values used to construct these response curves were $Q = 5$, $\beta = 0.05 \text{ Hz}^2/\text{m}^2$, and $\omega_0 = 1 \text{ rad/s}$

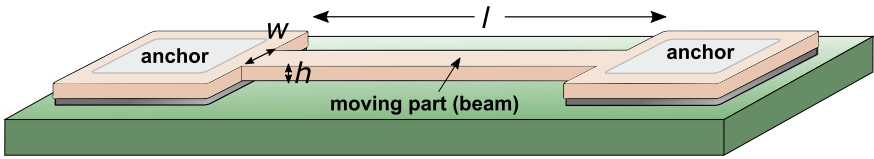


Fig. 7 Schematic description of a clamped–clamped beam: l , w , and h correspond to the length, width, and thickness of the beam

model may describe the nonlinear behaviors observed in the beam dynamics (Verbridge et al. 2006; Antonio et al. 2012; Abdolvand et al. 2016).

Figure 7 depicts a simplified schematic of a clamped–clamped structure.

3.2.1 Linear Analysis

The mass–damper–spring system represents the simplest model used to describe the linear resonator motions. It corresponds to Eq. (1) for which the nonlinear term β is null. The damper is associated here with energy losses in the system. The fundamental frequencies of excited clamped–clamped beam can be determined by solving the differential equation from Euler–Bernoulli beam theory. We assume that the beam deflection follows the fundamental mode vibration. The expression of the undamped resonance frequency for a clamped–clamped beam subjected to a lateral surface excitation can then be written as (Tilmans et al. 1992; Bao 2005)

$$\omega_0 = \frac{\lambda^2}{l^2} \sqrt{\frac{EI}{\rho wh}}, \tag{4}$$

where I , E , ρ , l , w , and h are quadratic moment, Young’s modulus, mass density, length, width, and thickness of the beam, respectively. λ is a constant satisfying $\cosh(\lambda)\cos(\lambda) = 1$. Equation (4) indicates that the resonance frequency is closely

related to the mechanical structure geometry. It corresponds, for instance, to 389 kHz for a 300 μm silicon beam with a width and a thickness of 4 μm and 10 μm, respectively ($\lambda = 4.73$ in that case).

3.2.2 Nonlinearity Effects

In a clamped–clamped beam, the nonlinear parameter caused by the elongation of the beam can be approximated from (Postma et al. 2005)

$$\beta = \frac{E}{18\rho} \left(\frac{2\pi}{l} \right)^4 . \tag{5}$$

For example, the calculated nonlinear coefficient is equal to $7.75 \times 10^{23} (\text{Hz/m})^2$ using Eq. (5) for a 300 μm silicon beam.

To better understand the nonlinear dynamics of a clamped–clamped beam, a finite element modeling using the ANSYS software (Theory Reference for the Mechanical 2017) was developed. Figure 8a) presents a deformed silicon beam in its fundamental mode. The anchors, substrate, and gages are also considered in the simulation. An initial modal simulation is used to identify the resonance modes of the beam. Using an explicit time analysis, the system is then excited in the proximity of a resonant peak by a time-varying lateral force applied in the middle of the beam. This analysis takes into account the nonlinear phenomena induced by large geometrical deformations and the mechanical dissipation that occurs during the structure motion.

The simulation results are depicted in Fig. 8b). We first note that the “hardening” phenomenon, characteristic of Duffing oscillator, is present. Unlike the symmetric response in the linear case, the peak amplitudes shift to the higher frequencies when the excitation force increases. The jumps are also observed. The cubic stiffness

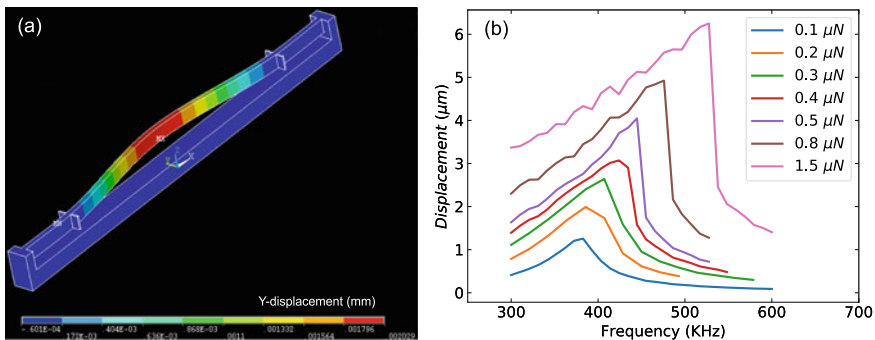


Fig. 8 **a** Displacement mapping of 300 μm clamped–clamped silicon beam obtained by ANSYS modal analysis. **b** Results of the analysis of transient finite elements on the clamped–clamped beam for different force amplitudes. The width w and thickness h of the beam were 4 μm and 10 μm, respectively

parameter can be determined from a fit to Eq. (3) and is equal to $(1.87 \pm 0.26) \times 10^{23}$ $(\text{Hz/m})^2$. This result is similar to the one obtained theoretically (Eq. (5)).

3.2.3 Damping Effects

The energy dissipation mechanisms of the mechanical system are associated with damping effects. The parameter indicating the damping and the efficiency of the resonator systems, the so-called quality factor Q , can be defined as the ratio of dissipated energy per period, Δ , to the energy stored in the oscillator (here, $kr^2/2$) (Tilmans et al. 1992; Bao and Yang 2007)

$$Q = 2\pi \times \frac{kr^2/2}{\Delta}. \tag{6}$$

Figure 9 depicts the amplitude–frequency curve for three damping conditions using numerical Duffing solutions (Eq. (1)). The larger damping effect corresponds to the smaller factor (black line) while the peak amplitude is higher for smaller damping (blue line). Note that the peak amplitude would be infinite in the absence of damping.

There are several sources of damping in mechanical structures. A quality factor Q_i can be attributed to each dissipation mechanism. The total quality factor Q can be written as Matthiessen’s rule (Matthiessen and Vogt 1864; Naeli and Brand 2009)

$$\frac{1}{Q} = \sum_i \frac{1}{Q_i}. \tag{7}$$

The extrinsic damping caused by the surrounding air can often be ignored for conventional mechanical systems. However, as air damping is related to the surface area of the resonator, viscous air damping can be significant for micromechanical

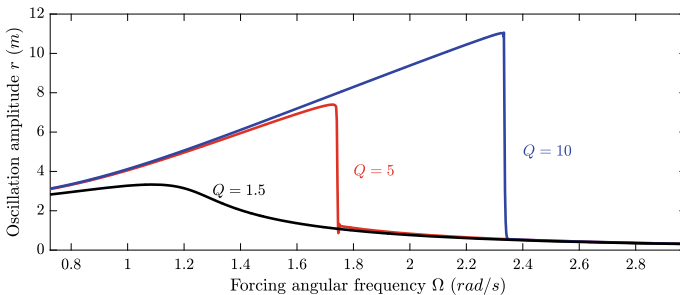


Fig. 9 Effect of damping on the amplitude–frequency response curve: small damping (blue line), intermediate damping (red line), and high damping (black line). The parameter values used to construct these response curves were $A = 2.5\text{m}/\text{rms}^2$, $\beta = 0.05 \text{ Hz}^2/\text{m}^2$, and $\omega_0 = 1 \text{ rad/s}$

devices. The first damping mechanism highlighted is the drag force. It represents the effect caused by the surrounding gas on the resonator when the beam is far away from any surrounding object. From Naeli and Brand (2009), the quality factor describing gas dissipation in microbeams is

$$Q_d = \frac{\rho h w \omega_0}{3\pi (\mu + w \sqrt{\rho_a \mu \omega_0 / 16})}, \quad (8)$$

where μ is the air dynamic viscosity and ρ_a is the air mass densities. This factor can be reduced experimentally by placing mechanical devices under vacuum (Tilmans and Legtenberg 1994; Gui et al. 1995).

A driving electrode must be close to the beam in order to electrostatically drive the mechanical resonator. If the gap d between the beam and the electrode is small compared to the beam thickness h , the main damping mechanism is the “squeezed-film effect” due to the incompressible character of the gas. This is all the more important when the gap is reduced. The corresponding analytical expression of squeezed-film damping is (Starr 1990; Bao 2005)

$$Q_s = \frac{\rho w d^3 \omega_0}{\mu h^2}. \quad (9)$$

For a silicon beam with $(w, h, l) = (4, 10, 300) \mu\text{m}$, where the gap d corresponds to $6 \mu\text{m}$, one has $Q_d = 529$ and $Q_s = 2740$. From Eq. (7), the combined quality factor Q is then 457. For additional effects comprising, for instance, the thermoelastic mechanism, we refer the reader to Verbridge et al. (2006), Naeli and Brand (2009), and Younis (2010). Note that the anchors in the clamped–clamped beams can also have a significant effect on the dynamics of the resonator (Lee et al. 2008; Naeli and Brand 2009).

4 Reservoir Computing in a MEMS

As highlighted in the previous sections, MEMS technology can reliably produce small and energy-efficient devices exhibiting rich dynamical behaviors often not accessible for mechanical structures at larger scales. Exploiting these dynamics for neuromorphic hardware thus seems a promising alternative to computing using conventional electronics, which keep struggling with power dissipation issues. As a result, the following section explores the use of a micromachined clamped–clamped silicon beam as the single dynamical node of a delay-coupled reservoir computer trained to perform simple classification tasks.

4.1 The MEMS Nonlinear Node

Construction of a hardware reservoir computer (RC) begins with the choice of a suitable physical node, which should have a nonlinear activation function in order to be able to model nonlinear processes. The stiffening Duffing behavior of a clamped–clamped silicon beam oscillating at large amplitudes can provide the nonlinearity in MEMS RC. An order of magnitude for the minimum oscillation amplitude to obtain sufficient nonlinear behavior is the amplitude r_c associated with the onset of bistability (Lifshitz and Cross 2010):

$$r_c = \left(\frac{4}{3}\right)^{3/4} \sqrt{\frac{\omega_0^2}{Q\beta}}. \quad (10)$$

For the beam studied in this section, the onset of the nonlinearity is around $r_c = 150$ nm.

The beam shown in Fig. 10 was microfabricated on a (100) silicon on insulator (SOI) substrate with a nominal resistivity of $(0.003 \pm 0.002) \Omega \text{ m}$ and a sacrificial oxide thickness of $1.5 \mu\text{m}$. It has a length of $L = 500 \mu\text{m}$, a width of $w = 10 \mu\text{m}$, corresponding to the SOI device layer thickness, and an in-plane thickness (normal to its displacement) of $h = 4 \mu\text{m}$. The device was wirebonded to a chip carrier and placed in a Faraday cage for the experiments, but was otherwise unpackaged. This lack of proper packaging makes the beams sensitive to dust in their environment, which has the undesirable effect of modifying their resonant frequency over time. For instance, one beam has had its natural frequency lowered by as much as 20% over the course of one year. The experimental quality factor of the MEMS was 167 ± 2 . This value, which is independent of the oscillation amplitude, is comparable to the analytical value of 204 obtained using Eqs. 7–9 for the nominal dimensions of the beam. Fabrication tolerances could account for this gap between the two values, as

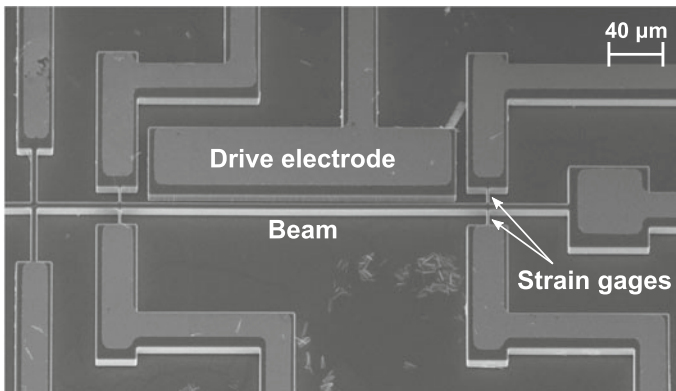


Fig. 10 SEM image of the MEMS

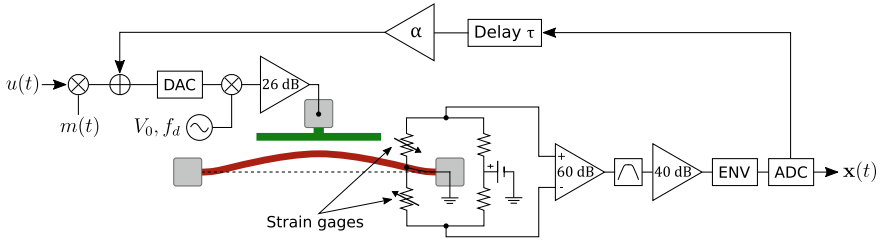


Fig. 11 Experimental setup for the reservoir computer. The masking procedure as well as the delayed feedback loop are implemented in the digital domain, while the post-processing (extraction of the displacement amplitude) is carried out through custom analog electronics

well as other dissipation mechanisms such as anchor loss and the proximity of the substrate. In the linear regime, the beam naturally oscillated at $f_0 = 155$ kHz, compared to a calculated value of 144.2 kHz (Eq. 4), although the maximum of the frequency response shifted to higher frequencies as the drive amplitude was increased, a behavior which corresponds to a stiffening Duffing oscillator. The Duffing parameter for the beam shown in Fig. 10 was estimated to 1.9×10^{23} Hz²m⁻² by adjusting Eq. 3 to experimental data of the beam's response. Equation 5 yields a comparable value of 1.1×10^{23} Hz²m⁻².

Among the plethora of possible transduction methods presented in Sect. 2.2, an appropriate choice for RC MEMS is to drive the beam electrostatically and sense its displacement piezoresistively. By polarizing a 300 μ m long drive electrode placed 6 μ m away from the beam in Fig. 10 with a voltage signal of the form $V_d(t) = V_0 \cos(2\pi f_d t)$, a force $F_d \propto V_d^2(t) = \frac{V_0^2}{2} (1 + \cos(4\pi f_d t))$ can be applied between the beam and the fixed electrode such that vibrations of the beam are solicited at twice the input voltage frequency f_d . The piezoresistive transduction of the beam motion to an electrical signal, carried out through 12 μ m long by 1.2 μ m wide piezoresistive strain gages patterned on the device, was chosen for its linearity (to ensure that nonlinear mapping comes exclusively from the beam's displacement) and sensitivity (transduction coefficient of $\sim 10^2$ V/m). Two external resistors were combined with the two piezoresistive gages, as illustrated in Fig. 11, to form a Wheatstone bridge, allowing for a differential measurement of the beam's motion. Compared to a single-ended measurement, the differential configuration has the advantage of reducing the system sensitivity to noise in the DC voltage source polarizing the Wheatstone bridge, but more importantly, it also cancels the feedthrough drive signal at the readout. This unwanted signal is symmetrically coupled to both readout points (ends of the piezoresistive gages) through parasitic capacitors (much larger than the ~ 10 fF capacitor formed by the beam and drive electrode) present in the device, while the displacement signal is of opposite sign in each branch (one gage stretches when the other gets compressed), so only the latter gets amplified by the instrumentation amplifier. The differential input stage is followed by a bandpass filter with a bandwidth of 80 kHz to further reduce the noise contribution, and a second amplification stage brings the displacement signal, initially of a few tens of μ V, to a

level suitable for the envelope detection stage that follows. This last step produces an appropriate output by extracting the amplitude of the beam displacement signal, yielding a signal-to-noise ratio (SNR) of 35 dB, essentially limited by the Johnson noise generated by the resistor bridge.

4.2 Training with Delayed Feedback

The use of a single physical node (Appeltant et al. 2011) greatly simplifies hardware implementation of an RC by drastically reducing the number of structures to couple physically, drive, and measure, with the main drawbacks of requiring a more refined preprocessing scheme and a serialization of the network (and thus of the computation). Indeed, since a single physical node is available, the reservoir consists of a virtual network created by time-division multiplexing of the input signal. While a space-coupled network would possess a multitude of physical nodes (typically $\sim 10^2$) coupled in space and use the ring-down time of the oscillators as a form of memory (the behavior of the oscillators depends on their history), a delay-coupled reservoir instead uses this decay time to couple adjacent virtual nodes in the time domain: the input signal is masked by a function of period τ , which in the simplest case is a function alternating randomly between two values after each time interval θ . τ is an integer multiple of θ which defines the number of virtual nodes ($N = \tau/\theta$). By choosing $\theta < T$, where $T = Q/(\pi f_0) = 330 \mu\text{s}$ is the decay time of the oscillator, the beam response during a given interval θ depends on its response during previous intervals. Since the oscillator decay time T is much shorter than the characteristic time τ of the input, the reservoir activation does not persist between two timesteps of the input signal, and the virtual network requires an additional feedback loop in order to have access to some form of memory. A feedback signal is thus added, with a delay τ and gain α , to the input for the next timestep. As a result, a given virtual node is driven by a superposition of the (masked) input and of its response to the input from the previous timestep:

$$V_d(t) = V_0 [u(t)m(t) + \alpha x(t - \tau) + 1] \cos(2\pi f_d t), \quad (11)$$

where $x(t)$ is the displacement amplitude signal at time t , $m(t)$ is the temporal mask, and $u(t)$ is the input signal.

The nonlinear nature of the beam's amplitude response (Dion et al. 2018) guided the choice of amplitude modulation of the sinusoidal pump for the RC input. In the case of a Duffing oscillator, the nonlinearity can be tuned to a certain extent by adjusting the drive frequency. The resulting system is schematized in Fig. 11. The input $u(t)$ is first scaled so that it is restricted to the empirically determined range $[0.60, 0.75]$, then it is sampled and held for a time τ and multiplied by the temporal binary mask of period τ and characteristic time θ . For the MEMS RC, optimization of the mask with respect to the RC success rate yielded mask values of 0.45 and 0.70. The result, $u(t) \times m(t)$, is used to modulate the amplitude of the sinusoidal pump

(Sect. 4.4.2 discusses adjusting the pump in more detail). Sampling the envelope (ENV) of the displacement signal at a rate θ^{-1} with an analog to digital converter (ADC) yields a vector $\mathbf{x}(t)$, containing the N virtual node states at timestep t . These values are then combined linearly to produce a scalar output:

$$y(t) = \mathbf{w}^T \mathbf{x}(t). \quad (12)$$

The goal of the training phase is to compute the appropriate vector \mathbf{w} of weights by adjusting them so that the response of the RC to a series of training examples approximates as well as possible a known target $y'(t)$. If the task for which the RC is trained is to process a signal which changes at every time period τ , for instance, then a series of M training periods can be presented to the system, each with an input value $u_k = u(k\tau)$ for $k = 1, \dots, M$, resulting in M outputs $y_k = y(k\tau)$ which can be compared to the desired outputs $y'_k = y'(k\tau)$ with the mean squared error

$$\frac{1}{M} \sum_{k=1}^M (y_k - y'_k)^2. \quad (13)$$

A similar mean squared error can be defined for the classification of input sequences of different lengths (with $y(t)$ sampled at the end of each input sequence).

The training process is done offline and consists in computing the vector \mathbf{w} minimizing the mean squared error between $y(t)$ and $y'(t)$. The result is

$$\mathbf{w} = \mathbf{y}'\mathbf{X}^T (\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I})^{-1}, \quad (14)$$

where \mathbf{y}' is the vector of desired outputs and \mathbf{X} is a matrix with each row corresponding to the state \mathbf{x} of the virtual nodes after one of the inputs u_k from the training set has been processed. γ is a regularization parameter that increases numerical stability and prevents overfitting. A value of $\gamma = 10^{-4} \text{ V}^2$ proved adequate for both benchmarks investigated below.

4.3 Performance Metrics

Following the training phase, it is customary to test the performance of the RC with inputs that were not part of the training set, so that the generalization capability of the RC can be assessed. In order to highlight its universal character, the MEMS RC discussed above was tested on two different benchmarks with the same set of hyperparameters: a network of $N = 400$ virtual nodes sampled every $\theta = 0.1$ ms with a feedback gain $\alpha = 1.1$ and a beam driven at $f_d = 80.3$ kHz, $V_0 = 72.5$ V, with the piezoresistive gages biased at 2.5 V.

4.3.1 Parity Benchmark

The parity benchmark is a conceptually simple task that can be nonlinear and requires memory. As such, it is well suited for a first evaluation of the system's performance. It consists of computing the parity of $n \geq 1$ successive input bits after an initial delay $\delta \geq 0$:

$$P_{n,\delta}(t) = \prod_{i=0}^{n-1} u(t - (i + \delta)\tau). \quad (15)$$

$P_{1,0}$ is linear and does not require memory, but for $\delta > 0$ or $n > 1$, the target depends on the history of the input signal, so the system must be able to store a transformed version of the input for a finite time. In this chapter, we will only report results with no delay, i.e., for $P_n = P_{n,0}$. For this task, the input $u(t)$ is a binary sequence randomly alternating between -1 and +1 at each time $t = k\tau$. It is thus first shifted and scaled to $[0.60, 0.75]$ before being fed to the RC, as discussed in Sect. 4.2.

Figure 12a shows the RC output for this task overlaid on the target after a training phase of 2000 samples. The performance is quantified by comparing the signs of the prediction and of the target over the whole 2000 samples of the testing set. The accuracy of the classification is the same for P_2 to P_4 since the raw RC output is thresholded, but the trace is more noisy for P_4 . By increasing n or δ , the complexity of the task is increased and this translates to a decrease in the prediction success rate. This performance drop can be counterbalanced up to a degree by increasing the number of nodes or the number of training samples, as evidenced by Fig. 14, or by a finer tuning of the nonlinearity (see Fig. 15). For the network of $N = 400$ nodes used to produce Fig. 12a, the mask period is $\tau = N\theta = 40$ ms, such that the bitstream is processed at a rate of 25 bits/s. On the other hand, a network of 10 virtual nodes is sufficient to process P_2 with less than 1% error, which leads to a classification rate of 10^3 bits/s. This means that for a given physical node with immutable characteristics, processing speed can be optimized for a specific task by adjusting the number of virtual nodes.

4.3.2 Spoken Digit Classification

With the same set of hyperparameters, the MEMS RC was also trained to classify the digits zero to nine spoken by sixteen different speakers, male and female, using the TI-46 dataset (Lieberman 1993). Since sounds have an inherent temporal dependence, this task seems well adapted to the RC approach, as evidenced by its predominance as a RC benchmark (Appeltant et al. 2011; Brunner et al. 2013; Coulombe et al. 2017; Dion et al. 2018; Dupont et al. 2012; Larger et al. 2012, 2017; Martinenghi et al. 2012; Paquot et al. 2012; Soriano et al. 2015; Torrejon et al. 2017; Verstraeten 2005). Whether it is obtained through RNNs or by using other means, state-of-the-art performance for this task is usually accompanied by spectral preprocessing to model the human ear, such as the Mel-Frequency Cepstral Coefficients (MFCC) or

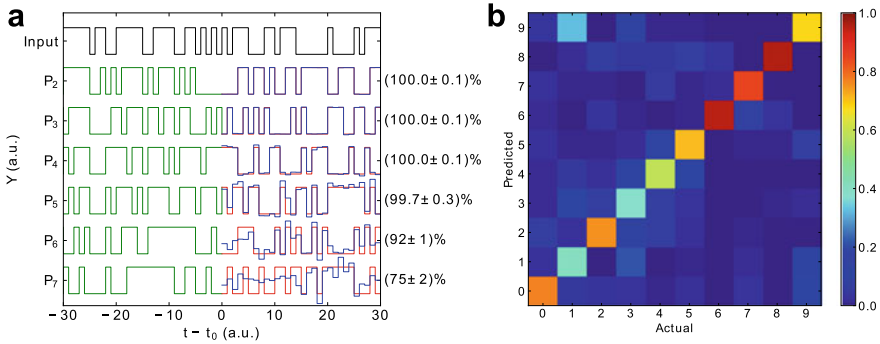


Fig. 12 **a** Performance for the parity benchmark. After a training phase (green), the RC response (blue) to the input (black) is compared to the target (red). **b** Confusion matrix for the spoken digit classification task. Colors indicate the probability that an input digit (columns) is assigned to a given class (rows) by the RC

the Lyon Passive Ear model (Lyon 1982). For this study, the preprocessing was kept minimal in anticipation of eventually interfacing the MEMS RC directly with sound pressure, as opposed to feeding samples from recorded waveforms. Each randomly selected utterance is first lowpass filtered 30 Hz and resampled at 60 samples/s, then it is normalized and scaled so that the complete sequence of waveforms is restricted to the range $[0.60, 0.75]$. In order to save processing time, silences before and after the utterance are cropped, which results in an average of $\bar{\eta} = 29$ samples per word. After being masked as described in Sect. 4.2, those samples are then fed sequentially to the reservoir without any pause between them. The output of a given virtual node for a given utterance is then the mean of its responses over the whole utterance (i.e., $x_i = (1/\eta) \sum_{j=0}^{\eta-1} x(i\theta + j\tau)$ for node i). Ten output layers are trained for the same reservoir activation: one boolean classifier is used for each individual digit. Since there are ten different possible classes for this task, the length M of the training sequence was increased to 6000 utterances so that the RC is trained on a sufficient number of examples for each digit.

Figure 12b shows that the confusion matrix for this task is almost diagonal, although some phonetically similar digits such as “1” and “9” or “4” and “5” are more often misclassified by the RC. The global success rate is $(70 \pm 2) \%$, and slightly better performance (Dion et al. 2018) could be obtained by optimizing the hyperparameters with respect to this particular task. Despite the fact that the training procedure lasts a few hours, the trained 400 node RC processes words at a rate of 1 per second, fast enough so that one could envision using such a system for real-time speech processing.

4.4 Hyperparameter Optimization

Finding optimal parameters for successful reservoir computing can be a tedious task, as RC performance typically depends on the appropriate combination of multiple hyperparameter values. Moreover, these parameters cannot be tuned independently: modifying one of them can shift the optimal value of other parameters. Choosing a random set of parameters will most often result in no computational success at all, and the accuracy landscape may display multiple local minima, making gradient descent optimization impractical. A gridsearch may seem like a foolproof optimization method, but without any indication of the location of the success region, the search space is vast and of high dimensionality. Besides, the region of non-zero success can be limited to a rather narrow region, as will become apparent later in this section, so that if the gridsearch is too coarse, the optimal parameter set can be missed altogether. Expert knowledge is thus necessary to set bounds for the different parameters of the gridsearch in a principled way or to perform a manual search in order to find a starting point with non-trivial success for optimization. To circumvent this obstacle, different methods are investigated in the RC literature (Bala et al. 2018), such as using genetic algorithms (Dale et al. 2016; Ferreira and Ludermir 2009, 2011), particle swarm optimization (Zhou 2010; Sergio and Ludermir 2012; Jubayer Alam Rabin et al. 2013; Salah et al. 2017), differential evolution (Zhang et al. 2013; Rigamonti et al. 2018; Wang et al. 2018), or hybrid variants thereof which combine different metaheuristics.

Temporal traces of reservoir activation such as those presented in Fig. 13 can also guide the initial optimization. By detuning a single parameter such as the drive frequency f_d , the feedback gain α , or the virtual node duration θ , the traces for healthy and unhealthy reservoirs can be compared and a few empirical criteria for successful RC can be deduced. Such criteria include the dynamic range and saturation of the response and its correlation with the input signal.

The optimization of hyperparameters shown below was performed using the parity benchmark, as the total training and testing time is much lower than the spoken word recognition benchmark: a training example for parity is composed of a single sample, while a spoken digit utterance contains tens of samples to feed to the RC. Nevertheless, the resulting parameter set can be used as a starting point for optimization with respect to a different task.

4.4.1 Number of Training and Testing Samples, Reservoir Size

The number of examples used for testing is one parameter that can be chosen in a principled way. Its only effect is on the uncertainty of the performance measurement. Considering that for all the benchmarks investigated here the testing phase is a series of Bernoulli trials (i.e., is the sample correctly classified?), the precision of the obtained success rates can be quantified using a binomial proportion confidence interval, such as the Agresti–Coull interval (Agresti and Coull 1998). In this specific

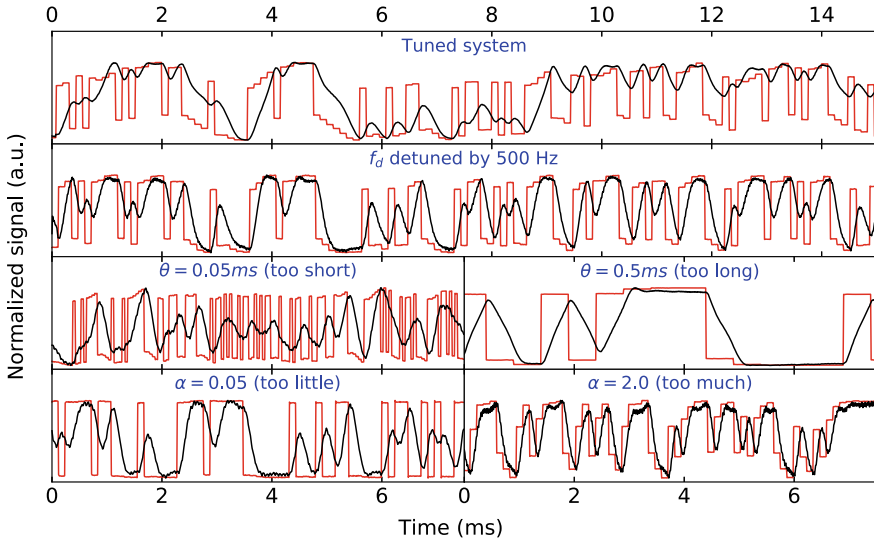


Fig. 13 Normalized drive signal envelope (red) and beam displacement amplitude (black) for a well-performing RC (top panel) and for various detuned configurations (lower panels). Note that the beam response is oversampled compared to normal operation where it is only sampled when the mask value is updated

case, the measurement error decreases as the number of trials and success rate are increased. A longer testing phase thus increases the measurement accuracy, but it also increases the acquisition time, making the results more susceptible to the effects of parameter drifts in the MEMS. This is where cross-validation becomes relevant: the training data can be reused for testing (and testing data for training), and thus not increase acquisition time but still get more measurement accuracy. A testing set of 2000 samples was deemed sufficient for the results presented here, as it is a good compromise between acquisition speed (~ 3 min for one complete training and testing experiment) and accuracy ($< 2\%$).

Figure 14 shows the P_3 to P_6 success rate for different pairs of (N, M) values. For this task, the minimum length of the training set (M) insuring optimal performance increases with the number of virtual nodes (N) in the explored region, and the number of nodes needs to be increased as the complexity of the task is increased from P_3 to P_6 in order to keep a constant success rate. A narrow region, centered around $M = N$, seems to prohibit adequate results. This could be due to overfitting, since this region does not respect the rule of thumb stating that N should not exceed $M/10$ to $M/2$ (Jaeger 2002). Training another output layer on the same data with $\gamma = 10^{-2} \text{ V}^2$ (to reduce overfitting by increasing regularization) increases performance for $M = N$ but considerably degrades performance otherwise. Good performance is also possible in a region where $N > M$, although unless the training set is of limited size, it is advisable to choose $N < M$ as the speed and energy cost of increasing the number of nodes is generally higher than using a longer training phase.

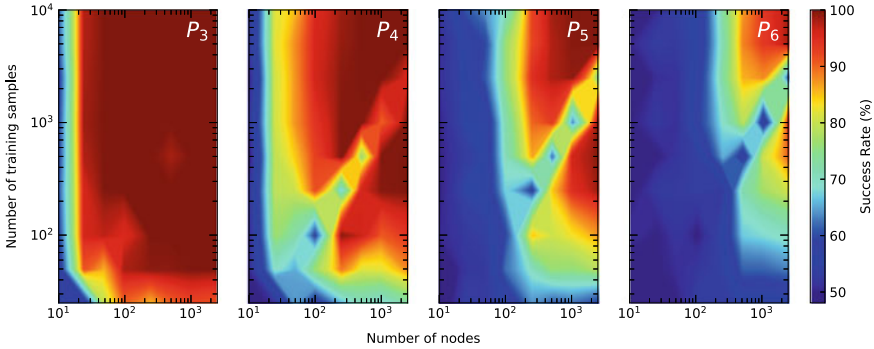


Fig. 14 Interpolated success rate in the number of nodes (N)—number of training samples (M) plane for P_3 to P_6 (left to right)

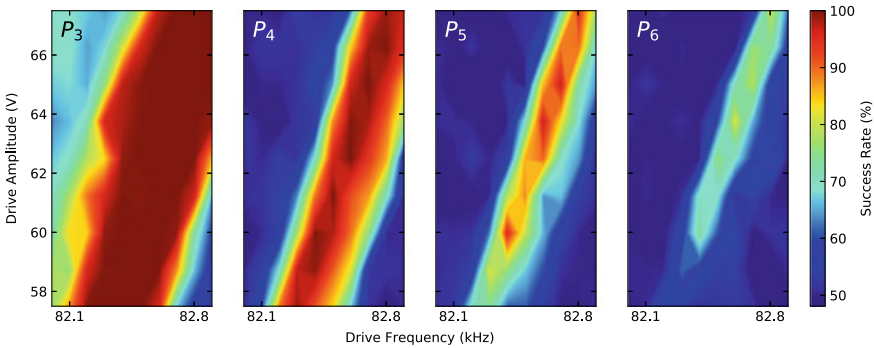


Fig. 15 Success rate in the drive frequency—drive amplitude plane for the P_3 to P_6 tasks. Note that this figure was produced earlier than the other figures when the oscillator had a slightly higher natural frequency. This slow drift of f_0 merely translates the features in this figure horizontally

4.4.2 Tuning the Nonlinearity

Figure 15 shows that good performance for P_3 to P_6 is limited to a rather narrow, tilted band in the drive frequency—drive amplitude plane. The more nonlinear task P_6 requires higher drive amplitudes for optimal success, corresponding to higher beam oscillation amplitudes and thus a more pronounced impact of the cubic term in the Duffing equation (Eq. 1). Figure 13 shows the effect of operating the system with the wrong combination of drive amplitude and frequency. At 500 Hz below the proper operating frequency, the dynamic range of the readout signal is reduced and its shape more closely resembles the input due to the more linear behavior of the beam. Such detuning can occur for example during the MEMS life if a large enough foreign particle gets attached to (or detached from) the beam, shifting its natural frequency.

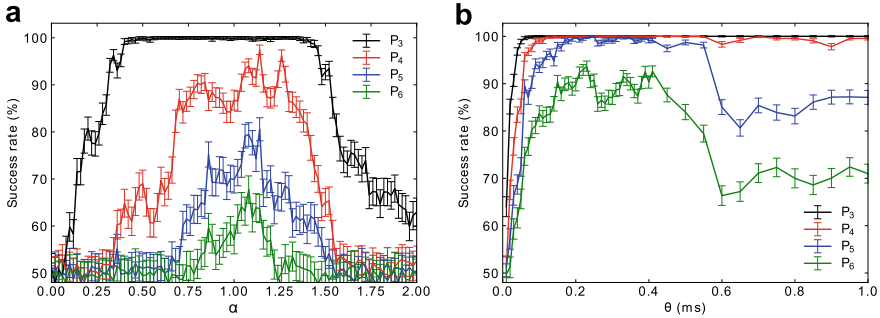


Fig. 16 The success rate for the parity function strongly depends on both the feedback gain α (a) and the mask update rate θ (b)

4.4.3 Feedback Strength

By plotting the success rate for the parity benchmark against the feedback strength α as in Fig. 16a, it can be seen that there is an intermediate value of α providing optimal results for all the investigated tasks. Below this value of $\alpha \simeq 1.1$, the system has less memory and success eventually vanishes at $\alpha = 0$. For values of α which are too large, the RC may not exhibit the fading memory property (Jaeger 2001) (or it may fade too slowly), and the system also tends to saturate (see bottom panel of Fig. 13), negatively impacting performances.

4.4.4 Coupling Strength

Figure 13 shows the effect of increasing or decreasing θ on the dynamics of the system. For $\theta = 0.05$ ms $\ll T$, the dynamic range is limited: the beam cannot respond quickly enough to the rapidly alternating low and high mask bits, and only behaves appropriately when there is a succession of identical mask values. This translates into a lower correlation coefficient of 0.05 between the input and output amplitudes, compared to a correlation coefficient of 0.44 for the optimized RC. For the case $\theta = 0.5$ ms $\gg T$, the response saturates as soon as there are two or more successive identical mask values, such that the readout (points sampled at the end of each period θ) essentially only visits two points of the transfer function (low level and high level saturation). The correlation coefficient is 0.60 and feedback has little effect, as the signal is less dynamical and more closely tied to the input due to the weak coupling between adjacent virtual nodes. The weak coupling regime ($\theta \lesssim T$), where a given virtual node state is only dependent on the state of its neighbor, is analogous to a linear chain of space-coupled oscillators.

Figure 16b shows the success rate for P_3 to P_6 as a function of θ , which essentially controls the connectivity matrix of the reservoir. While using a value of $\theta = 0.2$ ms gives slightly better results, a virtual node duration of $\theta = 0.1$ ms $\simeq T/3$ was used

for the results presented here as the computation is two times faster (~ 2 min). For higher values of θ , the longer acquisition time increases the effect of medium-term drifts in the system on the results: optimal weights may evolve over time but our offline training method doesn't allow adapting them through the acquisition.

5 Conclusion

MEMS devices form the basis of many of today's sensor technologies and are expected to play an important role in the development of new technologies related to artificial intelligence and machine learning, in the context of producing "big data" from autonomous systems (e.g., self-driving cars) or distributed sensor systems (e.g., the Internet of Things). We have presented in this chapter key concepts for using MEMS to construct neuromorphic computing devices, as well as key experimental results showing that reservoir computing can be implemented efficiently and robustly in MEMS. As MEMS can be small, energy-efficient, and function at high speeds, they could constitute a very attractive hardware substrate for unconventional AI computing. When used as "pure" computing devices (with an analog electrical input and an analog electrical output), they could implement AI functionalities with performance levels exceeding those of conventional electronics (Coulombe et al. 2017). Perhaps more interestingly, our MEMS devices can implement both neuromorphic computing *and* sensing functionalities in the same device. This is a fairly new idea, which could bring significant gains in system size and energy consumption through integration: instead of building mechatronic systems with a discrete sensor coupled to separate signal processing electronics, one could envision building a trainable sensor which exploits the nonlinearity of its sensing mechanism to implement computing functions on the measured data. We are developing this idea in MEMS, but similar ideas might also be relevant for optical sensors and RC systems, for instance.

Deep learning, as the most productive line of research for artificial intelligence today, relies on training complex systems (artificial neural networks) using large amounts of data. The separation between data generation and data processing has traditionally been very clear in such deep neural networks. One might however consider the example of biological brains, which actually integrate the sensing and computing functionalities in some sensory neurons (Pitkow 2015), perhaps as a strategy to increase efficiency, robustness, or adaptiveness. Nature might have discovered long ago that such integration was an effective way to build faster, smaller, and more energy-efficient intelligent biological systems, which are able to respond efficiently to sensory inputs collected from their environment (i.e., systems which are sophisticated integrated sensing *and* computing devices).

References

- R. Abdolvand, B. Bahreyni, J. Lee, F. Nabki, Micromachined resonators, a review. *Micromachines* **7**(9), 160 (2016)
- A. Agresti, B.A. Coull, Approximate is better than “Exact” for interval estimation of binomial proportions. *Am. Stat.* **52**(2), 119 (1998)
- G. Ananthasuresh, *Micro and Smart Systems: Technology and Modeling* (Wiley, Hoboken, 2012)
- D. Antonio, D.H. Zanette, D. Lopez, Frequency stabilization in nonlinear micromechanical oscillators. *Nat. Commun.* **3**(1) (2012)
- L. Appeltant, M.C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C.R. Mirasso, I. Fischer, Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011)
- S.I. Arroyo, D.H. Zanette, Duffing revisited: phase-shift control and internal resonance in self-sustained oscillators. *Eur. Phys. J. B* **89**(1) (2016)
- A. Bala, I. Ismail, R. Ibrahim, S.M. Sait, Applications of metaheuristics in reservoir computing techniques: a review. *IEEE Access* **6**, 58012–58029 (2018)
- M. Bao, *Analysis and Design Principles of MEMS Devices*, 1st edn. (Elsevier, Amsterdam, 2005). OCLC: 254583926
- M. Bao, H. Yang, Squeeze film air damping in MEMS. *Sens. Actuators A* **136**(1), 3–27 (2007)
- B. Barazani, G. Dion, A. Idrissi-El Oudrhiri, F. Ghaffari, J. Sylvestre, Micromachined neuro-processing accelerometer. To appear in *27th Canadian congress of Applied Mechanics*, vol. 3 (2019)
- B. Barazani, G. Dion, J.-F. Morissette, L. Beaudoin, J. Sylvestre, M. Neuroaccelerometer, Integrating sensing and reservoir computing in MEMS. *J. Microelectromech. Syst.* **29**(3), 338–347 (2020)
- R.C. Batra, M. Porfiri, D. Spinello, Review of modeling electrostatically actuated microelectromechanical systems. *Smart Mater. Struct.* **16**(6), R23–R31 (2007)
- M.J. Brennan, I. Kovacic, A. Carrella, T.P. Waters, On the jump-up and jump-down frequencies of the Duffing oscillator. *J. Sound Vib.* **318**(4–5), 1250–1261 (2008)
- D. Brunner, M.C. Soriano, C.R. Mirasso, I. Fischer, Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013)
- E. Buks, B. Yurke, Mass detection with a nonlinear nanomechanical resonator. *Phys. Rev. E* **74**(4) (2006)
- A. Cammarano, T.L. Hill, S.A. Neild, D.J. Wagg, Bifurcations of backbone curves for systems of coupled nonlinear two mass oscillator. *Nonlinear Dyn.* **77**(1), 311–320 (2014)
- J.C. Coulombe, M.C.A. York, J. Sylvestre, Computing with networks of nonlinear mechanical oscillators. *PLOS ONE* **12**(6), e0178663 (2017)
- M. Dale, S. Stepney, J.F. Miller, M. Trefzer, Reservoir computing in materio: an evaluation of configuration through evolution, in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, December 2016, Athens, Greece (IEEE, 2016), pp. 1–8
- G. Dion, S. Mejaouri, J. Sylvestre, Reservoir computing with a single delay-coupled non-linear mechanical oscillator. *J. Appl. Phys.* **124**(15) (2018)
- G. Duffing, *Erzwungene Schwingungen bei veranderlicher Eigenfrequenz und ihre technische Bedeutung*. Brunswick (1918)
- F. Dupont, B. Schneider, A. Smerieri, M. Haelterman, S. Massar, All-optical reservoir computing. *Opt. Express* **20**(20), 22783 (2012)
- K.L. Ekinci, M.L. Roukes, Nanoelectromechanical systems. *Rev. Sci. Instrum.* **76**(6) (2005)
- A.A. Ferreira, T.B. Ludermir, Genetic algorithm for reservoir computing optimization, in *2009 International Joint Conference on Neural Networks*, June 2009, Atlanta, Ga, USA (IEEE, 2009), pp. 811–815
- A.A. Ferreira, T.B. Ludermir, Comparing evolutionary methods for reservoir computing pre-training, in *The 2011 International Joint Conference on Neural Networks*, July 2011, San Jose, CA, USA (IEEE, 2011), pp. 283–290

- J. Guckenheimer, P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer, New York, 2002). OCLC: 51506830
- C. Gui, R. Legtenberg, M. Elwenspoek, J.H. Fluitman, Q-factor dependence of one-port encapsulated polysilicon resonator on reactive sealing pressure. *J. Micromech. Microeng.* **5**(2), 183–185 (1995)
- A.C. Harrie, Tilmans and Rob Legtenberg, Electrostatically driven vacuum-encapsulated polysilicon resonators: Part II. Theory and performance. *Sens. Actuators A: Phys.* **45**(1), 67–84 (1994)
- A. Husain, J. Hone, H.W.Ch. Postma, X.M.H. Huang, T. Drake, M. Barbic, A. Scherer, M.L. Roukes, Nanowire-based very-high-frequency electromechanical resonator. *Appl. Phys. Lett.* **83**(6), 1240–1242 (2003)
- B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, A.Y. Ng, An empirical evaluation of deep learning on highway driving, April 2015 (2015), [arXiv:1504.01716](https://arxiv.org/abs/1504.01716) [cs]
- H. Jaeger, The echo state approach to analysing and training recurrent neural networks—with an erratum note. *GMD Technical Report* **148**(34), 13, German National Research Center for Information Technology, Bonn, Germany (2001)
- H. Jaeger, A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach (2002), p. 46
- H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**(5667), 78–80 (2004)
- Md. Jubayer Alam Rabin, Md. Safayet Hossain, Md. Solaiman Ahsan, Md. Abu Shahab Mollah, Md. Tawabur Rahman, Sensitivity learning oriented nonmonotonic multi reservoir echo state network for short-term load forecasting, in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2013, Dhaka, Bangladesh (IEEE, 2013), pp. 1–6
- T. Kalmar-Nagy, B. Balachandran, Forced harmonic vibration of a duffing oscillator with linear viscous damping, in *The Duffing Equation*, ed. by I. Kovacic, M.J. Brennan (John Wiley & Sons, Ltd., Chichester, 2011), pp. 139–174
- F. Khoshnoud, C.W. de Silva, Recent advances in MEMS sensor technology—mechanical applications. *IEEE Instrum. Meas. Mag.* **15**(2), 14–24 (2012)
- Y. Lai, J. McDonald, M. Kujath, T. Hubbard, Force, deflection and power measurements of toggled microthermal actuators. *J. Micromech. Microeng.* **14**(1), 49–56 (2004)
- L. Larger, M.C. Soriano, D. Brunner, L. Appeltant, J.M. Gutierrez, L. Pesquera, C.R. Mirasso, I. Fischer, Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**(3), 3241 (2012)
- L. Larger, A. Baylon-Fuentes, R. Martinenghi, V.S. Udaltsov, Y.K. Chembo, M. Jacquot, High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Phys. Rev. X* **7**(1) (2017)
- Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**(7553), 436–444 (2015)
- J.E.-Y. Lee, Y. Zhu, A.A. Seshia, A bulk acoustic mode single-crystal silicon microresonator with a high-quality factor. *J. Micromech. Microeng.* **18**(6) (2008)
- D.A. Lieberman, *Learning: Behavior and Cognition*, 2nd edn. (Thomson Brooks/Cole Publishing Co., Belmont, CA, US, 1993)
- R. Lifshitz, M.C. Cross, Nonlinear dynamics of nanomechanical resonators, in *Nonlinear Dynamics of Nanosystems*, ed. by G. Radons, B. Rumpf, H.G. Schuster (Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2010), pp. 221–266
- H. Lin Wang, X.-Y.A. Huanling, H. Liu, Effective electricity energy consumption forecasting using echo state network improved by differential evolution algorithm. *Energy* **153**, 801–815 (2018)
- C. Liu, *Foundations of MEMS* (Pearson Prentice Hall, Upper Saddle River, 2006)
- R.F. Lyon, A computational model of filtering, detection, and compression in the cochlea **7**, 1282–1285 (1982)
- M. Madou, *Fundamentals of Microfabrication* (CRC Press, Boca Raton, 1997)
- P. Malatkar, A.H. Nayfeh, Calculation of the jump frequencies in the response of S.D.O.F. Nonlinear systems. *J. Sound Vib.* **254**(5), 1005–1011 (2002)

- R. Martinenghi, S. Rybalko, M. Jacquot, Y.K. Chembo, L. Larger, Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **108**(24) (2012)
- A. Matthiessen, C. Vogt, On the influence of temperature on the electric conducting-power of alloys. *Philos. Trans. R. Soc. Lond.* 167–200 (1864)
- K. Naeli, O. Brand, Dimensional considerations in achieving large quality factors for resonant silicon cantilevers in air. *J. Appl. Phys.* **105**(1) (2009)
- A.H. Nayfeh, D.T. Mook, *Nonlinear Oscillations* (Wiley, 1995)
- Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, S. Massar, Optoelectronic reservoir computing. *Sci. Rep.* **2**(1) (2012)
- X. Pitkow, S. Liu, D.E. Angelaki, G.C. DeAngelis, A. Pouget, How can single sensory neurons predict behavior? *Neuron* **87**(2), 411–423 (2015)
- H.W.Ch. Postma, I. Kozinsky, A. Husain, M.L. Roukes, Dynamic range of nanotube- and nanowire-based electromechanical systems. *Appl. Phys. Lett.* **86**(22) (2005)
- M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, S. Poll, Ensemble of optimized echo state networks for remaining useful life prediction. *Neurocomputing* **281**, 121–138 (2018)
- S.B. Salah, I. Fliss, M. Tagina, Echo state network and particle swarm optimization for prognostics of a complex system, in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, October 2017, Hammamet (IEEE, 2017), pp. 1027–1034
- J.A. Sanders, F. Verhulst, J. Murdock, *Averaging Methods in Nonlinear Dynamical Systems*, 2nd edn., Applied Mathematical Sciences (Springer, New York, 2007)
- A.T. Sergio, T.B. Ludermir, PSO for reservoir computing optimization, in *Artificial Neural Networks and Machine Learning - ICANN 2012*, vol. 7552, ed. by D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, A.E.P. Villa, W. Duch, P. Erdi, F. Masulli, G. Palm (Springer, Berlin, 2012), pp. 685–692
- M.C. Soriano, S. Ortin, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, 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)
- J.B. Starr, Squeeze-film damping in solid-state accelerometers, in *IEEE 4th Technical Digest on Solid-State Sensor and Actuator Workshop* (1990), pp. 44–47
- J. Sylvestre, G. Dion, B. Barazani, Provisional US patent application 62/780,589 (2018)
- Y. Tadokoro, H. Tanaka, M.I. Dykman, Driven nonlinear nanomechanical resonators as digital signal detectors. *Sci. Rep.* **8**(1) (2018)
- B. Tang, Mj. Brennan, V. Lopes, S. da Silva, 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)
- Theory Reference for the Mechanical APDL and Mechanical Applications (2017)
- H.A.C. Tilmans, M. Elwenspoek, J.H.J. Fluitman, Micro resonant force gauges. *Sens. Actuators A: Phys.* **30**(1), 35–53 (1992)
- 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, J. Grollier, Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**(7664), 428–431 (2017)
- J.T.M. van Beek, G.J.A.M. Verheijden, G.E.J. Koops, K.L. Phan, C. van der Avoort, J. van Wingerden, D. Ernur Badaroglu, J.J.M. Bontemps, Scalable 1.1 GHz fundamental mode piezo-resistive silicon MEMS resonator, in *2007 IEEE International Electron Devices Meeting*, December 2007, Washington, DC (IEEE, 2007), pp. 411–414
- W.M. Van Spengen, R. Puers, I. De Wolf, On the physics of stiction and its impact on the reliability of microstructures. *J. Adhes. Sci. Technol.* **17**(4), 563–582 (2003)
- S.S. Verbridge, J.M. Parpia, R.B. Reichenbach, L.M. Bellan, H.G. Craighead, High quality factor resonance at room temperature with nanostrings under high tensile stress. *J. Appl. Phys.* **99**(12), 124304 (2006)

- D. Verstraeten, B. Schrauwen, D. Stroobandt, Isolated word recognition using a liquid state machine, in *Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN)* (2005), pp. 435–440
- D. Witt, R. Kellogg, M. Snyder, J. Dunn, Windows into human health through wearables data analytics. *Curr. Opin. Biomed. Eng.* (2019)
- K. Worden, On jump frequencies in the response of the duffing oscillator. *J. Sound Vib.* (1996)
- N. Yazdi, F. Ayazi, K. Najafi, Micromachined inertial sensors. *Proc. IEEE* **86**(8), 20 (1998)
- M.I. Younis, *MEMS Linear and Nonlinear Statics and Dynamics*, vol. 20, Microsystems (Springer, New York, 2010). OCLC: ocn495781913
- S. Zaitsev, O. Shtempluck, E. Buks, O. Gottlieb, Nonlinear damping in a micromechanical oscillator. *Nonlinear Dyn.* **67**(1), 859–883 (2012)
- Y. Zhang, Y. Yu, D. Liu, The application of modified ESN in chaotic time series prediction, in *2013 25th Chinese Control and Decision Conference (CCDC)*, May 2013, Guiyang, China (IEEE, 2013), pp. 2213–2218
- H. Zhou, Y. Wang, K. Xing, Modeling of McKibben pneumatic artificial muscles using optimized echo state networks, in *2010 8th World Congress on Intelligent Control and Automation*, July 2010, Jinan, China (IEEE, 2010), pp. 1723–1728