

<https://doi.org/10.1038/s44172-024-00193-5>

# In-sensor human gait analysis with machine learning in a wearable microfabricated accelerometer

Check for updates

Guillaume Dion<sup>1</sup>, Albert Tessier-Poirier<sup>1</sup>, Laurent Chiasson-Poirier<sup>1</sup>, Jean-François Morissette<sup>1</sup>, Guillaume Brassard<sup>1</sup>, Anthony Haman<sup>1</sup>, Katia Turcot<sup>2</sup> & Julien Sylvestre<sup>1</sup> ✉

In-sensor computing could become a fundamentally new approach to the deployment of machine learning in small devices that must operate securely with limited energy resources, such as wearable medical devices and devices for the Internet of Things. Progress in this field has been slowed by the difficulty to find appropriate computing devices that operate using physical degrees of freedom that can be coupled directly to degrees of freedom that perform sensing. Here we leverage reservoir computing as a natural framework to do machine learning with the degrees of freedom of a physical system, to show that a micro-electromechanical system can implement computing and the sensing of accelerations by coupling the displacement of suspended microstructures. We present a complete wearable system that can be attached to the foot to identify the gait patterns of human subjects in real-time. The computing efficiency and the power consumption of this in-sensor computing system is then compared to a conventional system with a separate sensor and digital computer. For similar computing capabilities, a much better power efficiency can be expected for the highly-integrated in-sensor computing devices, thus providing a path for the ubiquitous deployment of machine learning in edge computing devices.

The design flexibility of micromechanical devices has enabled a vigorous research effort to develop mechanical computing elements<sup>1</sup>, such as memory cells<sup>2</sup> or logic gates<sup>3</sup>. These computing elements are built, for instance, using contact switches or resonators, and they are small, operate at high frequencies, and dissipate a small amount of energy. They might outperform their conventional, electronic counterparts in future computing applications that operate on small power budgets, including battery-powered sensors that perform edge computing in wearable devices or for the Internet of Things<sup>4</sup>.

By virtue of their operating in the mechanical domain, mechanical computing devices are especially interesting for their integration with sensors that measure sound, acceleration, strain, shape, adsorbed mass, or other mechanical properties. Ultimately, the mechanical computing device in a sensor could be driven directly by a measured mechanical stimulus to produce without transduction and, in the most efficient manner, a processed output, such as a detection, classification, or control signal. Examples that have been discussed in the literature include resonating structures that are sensitive to sound and can identify certainly spoken commands<sup>5</sup>,

metamaterial structures that can identify specific shapes from the positions of a set of contact points<sup>6</sup>, as well as various proposals for *in materio* computing<sup>7</sup>. The defining characteristic of these devices is that they can perform complex data processing, including the prediction or the classification of time-varying signals, with minimal or no electronic components that, when they are required, do not contribute substantially to the computing functions of the system. The term *in-sensor computing* has been used to refer to the information processing of physical stimuli by devices that perform both the sensing and the processing of these stimuli in the same physical domain<sup>8</sup>. This is a radically different approach compared to conventional systems that use sensors that first convert the physical stimuli to electrical signals and then perform the processing using analog or digital electronic computers.

Our work provides strong evidence that machine learning with in-sensor computing is a viable technological option for real-time, secure, and highly efficient wearable devices and edge computing devices. We present a demonstration of a complete sensor with built-in machine-learning capabilities via a wearable accelerometer that uses neuromorphic

<sup>1</sup>Institut Interdisciplinaire d'Innovation Technologique, Université de Sherbrooke, Sherbrooke, Canada. <sup>2</sup>Department of Kinesiology, Faculty of Medicine, Interdisciplinary Center for Research in Rehabilitation and Social Integration (Cirris), Université Laval, Québec, Canada. ✉e-mail: [julien.sylvestre@usherbrooke.ca](mailto:julien.sylvestre@usherbrooke.ca)

computing in the mechanical domain to detect subtle gait patterns in human subjects. Our device implements its sensing and computing functions using the displacements of mechanical structures, and as a result, it can perform machine learning on sensor data in an especially efficient manner. The device is attached to the left foot, and is able to detect in real-time changes in the gait far better than a linear signal classifier. Unlike conventional benchmark tasks<sup>9</sup>, this gait analysis task provides a difficult real-world testbed to analyze the benefits of in-sensor computing, as it requires a fully-integrated wearable device that must also be robust against variability in the gait accelerations, walking speed, morphology of the test subjects, and noise and non-ideal sensor characteristics.

Our fully integrated device allows a direct comparison of in-sensor computing with a conventional solution (i.e., machine learning software executing on a commercial microcontroller). Building on previous work, our device uses a custom micro-electromechanical system (MEMS) accelerometer that couples the inertial movement of a suspended proof mass (for acceleration sensing) to the nonlinear oscillations of a doubly clamped beam (for computation)<sup>10,11</sup>. Here, the MEMS device is additionally shielded against mechanical and electromagnetic perturbations, thermally stabilized, and, most importantly, tightly packaged with all its auxiliary electronics so that it can be tested as a miniaturized wearable device. As it is commonly done in the field of *physical reservoir computing*<sup>12,13</sup>, the auxiliary electronic components are used to drive our device and to increase the dimensionality of its acceleration data representation using a feedback mechanism.

Our prototype device is shown to consume power at a level that is similar to a heavily optimized commercial micro-controller implementing the same gait classification task in software (see the section “In-sensor computing for wearable devices”). The total power consumed by the MEMS prototype device was measured and compared to a model summing the calculated power consumed by each subsystem ( $970 \pm 10$  mW total measured power, 958 mW calculated, see the “Methods” subsection “Power consumption”). This analysis showed that a simple re-design of certain subsystems in our prototype device would further reduce its power consumption by an order of magnitude (down to an estimated 94 mW, see “Methods” subsection “Power consumption”), thus providing a significant advantage over conventional microcontroller-based solutions and paving the way for wearable devices that are much smaller or function much longer on a battery charge. Our results thus firmly establish in-sensor computing in MEMS accelerometers as a competitive alternative to conventional embedded software solutions in a device that is integrated and effective enough for a gait classification task that is difficult and relevant for real-world applications.

In addition, only the classification labels are transmitted (wirelessly) by our in-sensor computing device. The raw physiological data (i.e., the accelerations) are nonlinearly transformed through the dynamics of the MEMS before they are digitized. The digital data in the electronics feedback mechanism cannot be inverted to retrieve the raw accelerations because of the complexity of the dynamics of the MEMS, thus providing built-in data security. This feature of in-sensor computing could be especially relevant for wearable medical devices to protect the privacy of their users and to save energy by reducing the amount of transmitted data and the use of encryption algorithms.

The most fundamental challenge for in-sensor computing has arguably been the development of an appropriate coupling between the computing functions and the sensor functions of a device. Our work shows that this coupling is conceptually simpler to implement in a device when the computing functions are realized using degrees of freedom that can also be influenced by the physical stimuli that are being measured. We achieve this by coupling a nonlinear resonator to a suspended-proof mass in a micro-fabricated accelerometer. Data processing is realized in the acceleration sensor by leveraging the computing model known as *reservoir computing*<sup>14</sup>, which has been tremendously influential in the development of computing systems that use the measurable degrees of freedom of physical dynamical systems (aka. physical reservoir computers<sup>9</sup>). Recent examples of physical

reservoir computers with in-sensor computing capabilities have included the coupling of radio-frequency electromagnetic waves to electron-spin reservoir computers<sup>15</sup> and a light-sensitive memristor reservoir computer<sup>16</sup>. Our work advances the nascent field of in-sensor computing by demonstrating the energy efficiency, compactness, and data processing capabilities of machine learning concepts implemented in the same physical domain as the sensory information in a highly functional wearable device.

## Results

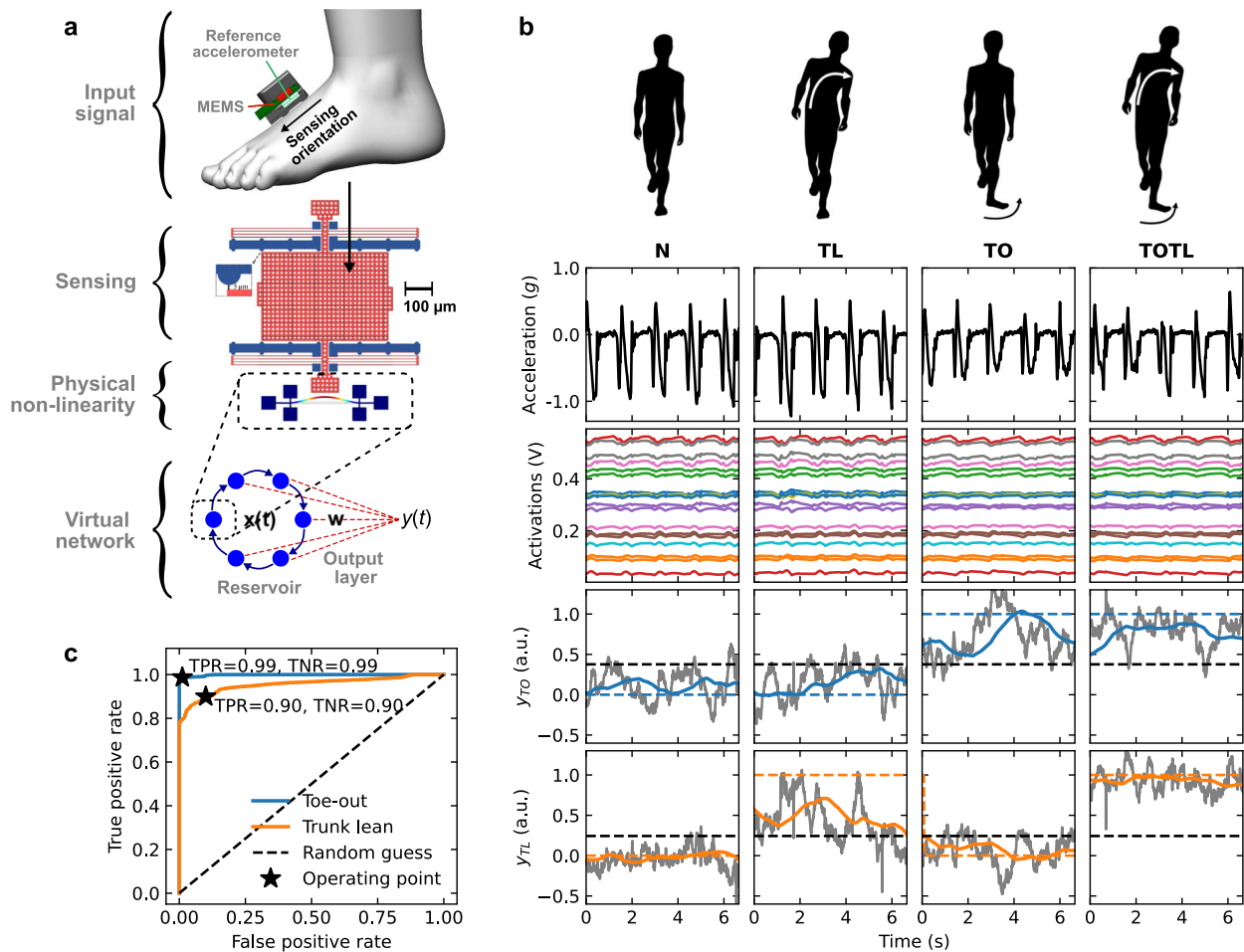
### MEMS device with sensing and trainable computing capabilities

We address a human gait classification task (Fig. 1a–c) with a microelectromechanical (MEMS) accelerometer that performs both sensing and computing in the mechanical domain. The mechanical reservoir computer is implemented using the dynamical nonlinearity of a thin silicon beam clamped at both ends, which is similar to the device described previously<sup>10</sup>. When the beam is driven by an oscillating electrostatic force that is sufficiently large, the amplitude of its oscillations exhibits complex nonlinear dynamics (Fig. 2a, b), that are used for reservoir computing. The driving electrostatic force on the clamped beam is applied through a suspended proof mass that moves when accelerations are applied to the device (Figs. 1a and 2d), as described previously<sup>11</sup>. The amplitude of the driving force varies with the distance between the proof mass and the equilibrium position of the beam, and the parameters of the system are chosen so that the amplitude of the oscillations of the beam is a complex nonlinear function of the position of the proof mass (and therefore, of the accelerations, see Fig. 1b). In order to increase the complexity of the signals that can be generated from the amplitude response of the beam, a feedback technique<sup>12</sup> is employed to create multiple different virtual responses using a time-multiplexing technique (Fig. 1b), each being a different nonlinear function of the accelerations. The virtual responses are sampled at regular intervals with conventional electronics to generate a vector of ‘activation values’ at each time interval (a complete block diagram of the system is shown in Supplementary Fig. 2). The scalar product between this vector and a trained weights vector is finally computed in a conventional microcontroller at each time interval to produce the output classification for the type of gait (Fig. 1b).

The feedback approach to generate virtual activation values has been used successfully in a wide variety of physical reservoir computers<sup>9</sup>, as it allows to adjust the memory of a physical reservoir (which is often fixed by the hardware) to the memory required to solve a task<sup>10,17</sup>, and as it allows to leverage the fast or energy-efficient physical reservoirs to perform complex nonlinear computations, that are then efficiently processed by conventional electronics. As we demonstrate in this work, even sub-optimized physical reservoir computers that rely on a feedback technique can achieve performance levels that are competitive with conventional electronics<sup>18–20</sup>.

A different weight vector is used for each of the two gait patterns (toe-out, TO, and trunk-lean, TL, Fig. 1b), so the device can discriminate between a normal gait (N) or an abnormal gait with either (TO or TL) or both (TOTL) of the TO and TL patterns. The MEMS device was packaged and integrated on a small printed circuit board (Fig. 2e) so that it could be attached to a shoe with minimal interference to the gait. Additional details about the MEMS device and the weights training process are provided in the “Methods” subsections “MEMS design and fabrication” and “MEMS packaging and board integration”. Similar MEMS devices have also been described elsewhere<sup>21–27</sup>.

For our MEMS device, the in-sensor computing that is done in the mechanical domain is the result of the modulation of the high-frequency driving voltage (around 250 kHz) that is the input to the beam reservoir computer by the low-frequency displacement (below 400 Hz) of the proof mass, which is sensitive to external accelerations. As the electrostatic driving force is proportional to the square of the drive voltage, the beam is driven into fast in-plane oscillations (around 500 kHz). As displayed in Fig. 1b and Supplementary Fig. 1, gait acceleration signals have a low-frequency content (0.5–15 Hz) and small amplitude (~1 g) as a result of the biomechanics of the human body<sup>28,29</sup>. The inertial sensing component of our MEMS system is designed to respond to these acceleration signals (see the sensitivity curve in



**Fig. 1 | General description of the MEMS gait analysis system.** **a** Schematic description of the MEMS gait analysis system, showing the position of the MEMS and reference accelerometers on the foot, the MEMS accelerometer sensing element coupled to the non-linear beam resonator, and the network of virtual activations used to generate the output signal  $y(t)$ . **b** Schematic representation of the gait patterns (top row), accelerations measured by the reference accelerometer (second row), virtual activations for 18 randomly selected nodes (third row), and gait discrimination output signals  $y_{TO}$  for the toe-out gait and  $y_{TL}$  for the trunk-lean gait

(fourth and fifth rows, respectively, gray for the raw signals, and color lines for the time-averaged signals), together with the threshold levels (fourth and fifth rows, black dashed lines) and detector binary outputs (fourth and fifth rows, dashed color lines). **c** Receiver operating characteristic curve for both walking patterns. MEMS micro-electromechanical system, N normal gait, TL trunk-lean gait, TO toe-out gait, TOTL toe-out and trunk-lean gait, a.u. arbitrary units, TPR true positive rate, TNR true negative rate.

Fig. 2d). By coupling the inertial proof mass to the driving force acting on the doubly clamped beam, efficient computing is achieved on a timescale set by the short ( $\tau = 150 \mu\text{s}$ ) decay time of the beam oscillator (Fig. 2a). The delayed feedback loop (details in section Supplementary Note 3) and leaky integrator (details in the section “Training the MEMS system”) are used to increase this relatively short intrinsic memory, to match the frequency content of the gait acceleration signals. For the virtual network with  $N = 100$  nodes that were used in this work, the envelope of the beam oscillation was sampled at a rate of  $\theta^{-1} = 1/70 \text{ MHz} > \tau^{-1}$  to produce the activation signals so that the raw classification signals were updated from the linear combination of the activation signals at a rate of 142.85 Hz. Since this update rate is too high for the frequency content of the biomechanical task, a moving average was performed over 300 time steps. This resulted in an effective computing timescale of  $\sim 2.1 \text{ s}$ , which was well matched to the gait accelerations, and also filtered noise at frequencies higher than the bandwidth of the acceleration signals.

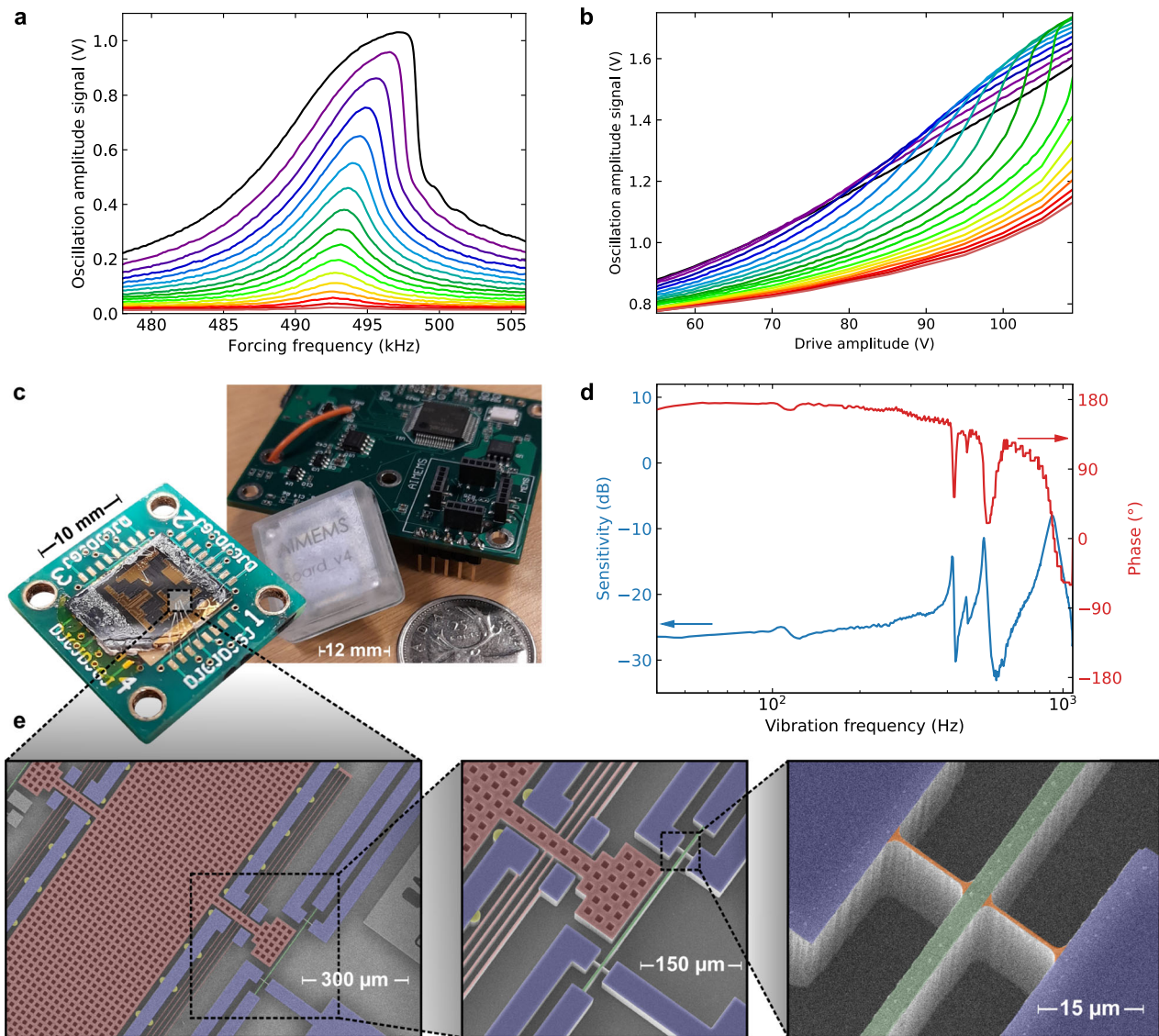
### In-sensor identification of human gait patterns

In-sensor computing with our MEMS device was applied to the detection of the four different gait patterns (N, TO, TL, TOTL) using a single device

attached to the left foot. The automated identification of these gait patterns is relevant clinically, as it may be used for the management of certain musculoskeletal diseases, such as knee osteoarthritis, for instance, via gait retraining using real-time biofeedback<sup>30</sup>. While gait pattern identification can be performed in laboratories equipped with 3D motion capture systems or using multiple inertial measurement units<sup>31–33</sup>, the difficulty of the task is compounded by our use of a single accelerometer, which is motivated by our objective to develop the simplest, lowest cost, and most unintrusive wearable device.

In our study, ten healthy subjects were instructed to walk on a treadmill with the MEMS system attached to their left shoe while alternating between the four gait patterns (see the “Methods” subsection “Gait analysis protocol” and Supplementary Note 2 for details). Relatively slow walking speeds, between 0.36 and 0.72 m/s, were set on the treadmill to correspond to a range of speeds in a population with knee pain. This range ( $\pm 33\%$  around the median) is much larger than the expected range of natural variations for a given person (e.g.,  $\pm 3\%$ )<sup>34</sup>. The virtual activations from the MEMS reservoir computer and the accelerations measured by a conventional accelerometer co-located with the MEMS were sent to a computer in the laboratory. The training and the testing phases were performed





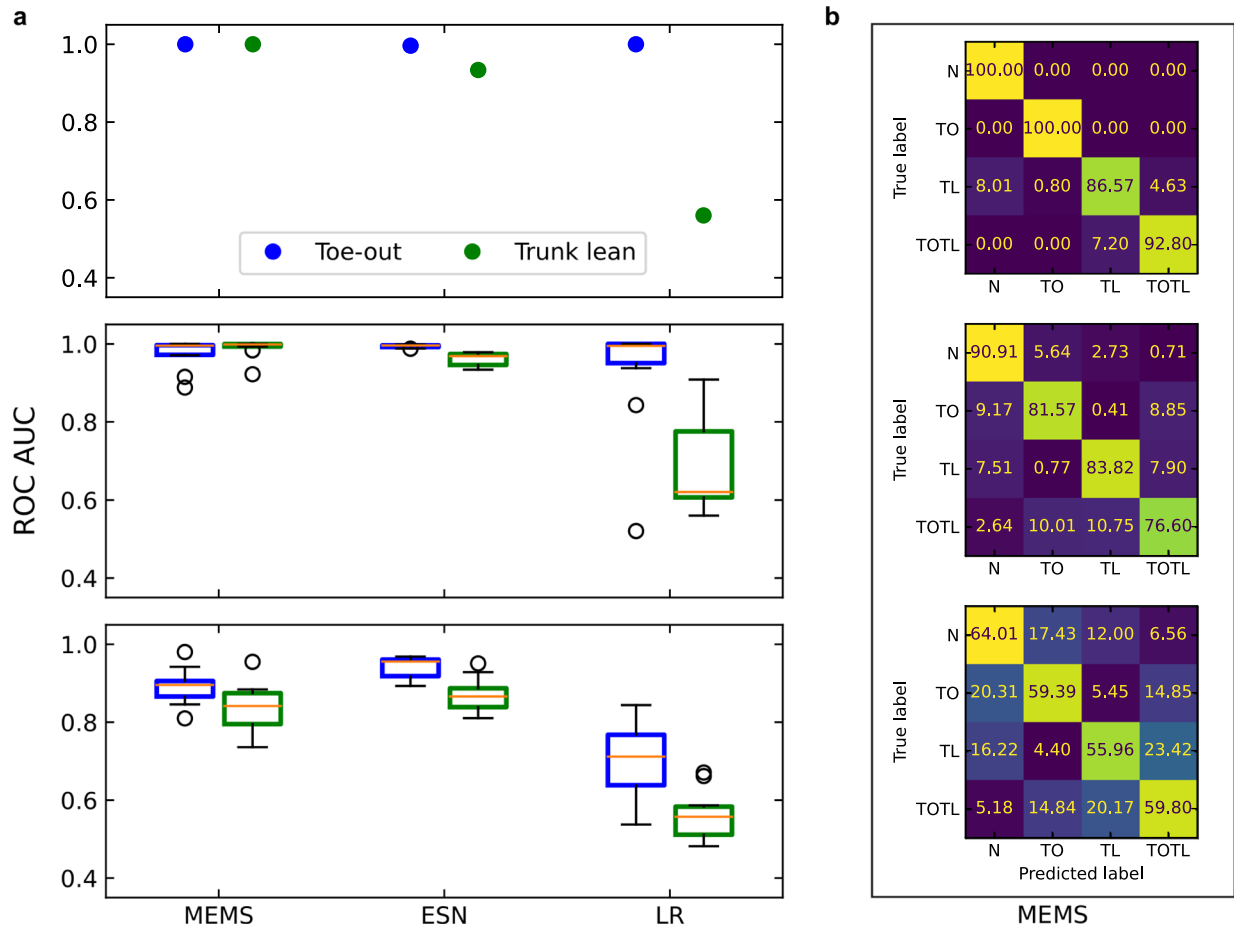
**Fig. 2 | Mechanical response of the gait analysis system.** **a** Frequency response curves for the silicon beam for a drive voltage amplitude that is increased from 30 V (red) to 109 V (black) in steps of 5 V (up to 105 V, black curve at 109 V). The Duffing stiffening nonlinearity of the silicon beam can be observed in the asymmetry of the frequency response curves. **b** Drive amplitude voltage sweeps for the silicon beam for a forcing frequency that is increased from 493.4 kHz (black curve) to 501.0 kHz (red curve) in steps of 400 Hz. These sweeps show that the deviation from the linear regime is increased as the drive voltage amplitude and forcing frequency are

increased. **c** Photographs of the packaged MEMS and its main electronics board. **d** Inertial response of the MEMS, where the sensitivity is displayed as the amplitude of the oscillation signal (in V) per unit of acceleration applied (in standard gravity  $g$ ), converted to decibels (dB) using a reference level of 1 V per  $g$  (at 0 dB). **e** Manually colored scanning electron microscope images of a device (red: inertial mass and spring suspension, blue: anchors, green: clamped beam, yellow: inertial mass stoppers, orange: strain gages).

independently for each participant to obtain personalized classifiers with good performance. For each participant, we used a four-fold cross-validation (see the “Methods” subsection “Training the MEMS system” for details). For each of the four splits of the cross-validation, the training was performed on three folds using a ridge regression, and the testing was performed on the remaining fold. The classification performance was assessed by computing the average of the area under the curve (AUC) of the receiver operating characteristic (ROC) curve, computed over the four splits (an AUC of 0.5 corresponds to a random classifier while an AUC of 1.0 corresponds to a perfect classifier). See Fig. 1c for an example of a ROC curve and Fig. 3 for a characterization of the performance of the MEMS system. Figure 3 also presents for comparison the performance of two conventional software solutions (an echo-state network, ESN, and logistic regression, LR, details in the “Methods” subsection “ESN and logistic regression”) that were

not implemented using in-sensor computing and that instead used the data acquired from the conventional accelerometer. The weight vectors obtained in training could finally be transferred back to the MEMS system, which was then used to classify gait patterns in real-time. The real-time performance of the system was then evaluated while a subject was walking on the treadmill with different gait patterns (see Supplementary Movie 1 for a demonstration).

Our results show that the TO and the TL gait detectors could both perform very well, with a typical probability for correctly identifying the TO pattern while the subject was walking with that pattern (TPR) above 99% and a typical probability for incorrectly identifying the TO pattern while the subject was not walking with that pattern (FPR) below 1% (respectively, 90% and 10% for the harder TL pattern, see Fig. 1c). This level of performance for training customized to a single subject could be useful in the context of



**Fig. 3 | Performance of in-sensor classification.** **a** Areas under the receiver characteristic curve for the toe-out (blue) and trunk-lean (green) gaits, for the MEMS, ESN, and LR classifiers, for a single participant walking at a steady pace (top panel), for all participants walking at a single speed of 0.63 m/s (middle panel), and for all participants, all walking at five different speeds (bottom panel). The box plots displayed in the bottom two panels show the first and third quartiles (boxes), the median (orange lines), 1.5 times the interquartile range (whiskers), and the data

points outside the whiskers (circles). **b** Corresponding confusion matrices for the MEMS classifier showing the probability (in percentages) of the detector to predict a gait pattern (columns) for a prescribed gait pattern (rows). MEMS micro-electromechanical system, ESN echo-state network, LR logistic regression, ROC receiver operating characteristic, AUC area under the curve, N normal gait, TL trunk-lean gait, TO toe-out gait, TOTL toe-out and trunk-lean gait.

personalized medicine<sup>35</sup>, where a device could be trained in a clinical environment to provide a high-accuracy gait classifier to achieve a specific therapeutic objective. Figure 3a further shows the AUC for subtasks with various levels of complexity. It can be observed that the MEMS classifications were robust against variations in the gait patterns across individual subjects when they all walked at the same speed, even if there were substantial differences within the group of subjects (10 participants, age  $33 \pm 15$  years, height  $173 \pm 8$  cm and weight  $75 \pm 13$  kg, reported as mean  $\pm$  standard deviation). When the difficulty of the gait identification task was further increased by using a single set of weight vectors for all walking speeds, the performance for the MEMS classifications was reduced (median AUC of 90% for TO and 84% for TL), principally due to an increase in the frequency of errors between the normal and TO patterns, and between the TL and TOTL patterns (Fig. 3b).

This level of classification performance with the MEMS device is very good compared to what can be achieved with linear classifiers, especially for the harder TL identification task. Figure 3a shows, for instance, that the median AUC for the TL pattern with all participants and all speeds is much better with the MEMS device (84%) than with a logistic regression classifier that did not perform significantly better than random guessing (median AUC of 56%). A similar logistic regression classifier could perform well for TO identification at a fixed speed, but its performance was much lower than

the performance of the MEMS device on all other tasks (TL and TO with combined speeds). This establishes that only nonlinear classifiers can perform well under the conditions of our gait identification task and further illustrates the usefulness of sophisticated in-sensor processing for this application.

### In-sensor computing for wearable devices

In order to have a relevant technological impact, in-sensor computing devices need to generalize while performing complex computations, i.e., they must be robust against variations in the inputs that were not seen during training, as well as a certain level of noise. This robustness is a hallmark of neuromorphic computing techniques, including reservoir computing. In addition, these techniques can often model the nonlinearities in sensor systems<sup>36–38</sup>, so they perform well even with non-ideal sensors. This could be leveraged to alleviate the design requirements of many sensors, thus potentially lowering their cost of fabrication or calibration.

The performance of our prototype device as a pure sensor (Fig. 2d) was far from the state-of-the-art for MEMS accelerometers, with a sensitivity between 0.05 V per g and 0.1 V per g in a bandwidth of 400 Hz, some in-band resonances and other problems (see Supplementary Note 1 for details). Nevertheless, the beam reservoir computer was able to learn these performance limitations, to classify gait patterns as well as a neural network

operating on data from a commercial accelerometer. Figure 3a shows a comparison of the classification performance of the MEMS device and of a conventional system built using a commercial accelerometer and software ESN<sup>39</sup> executed on a microprocessor. It can be observed that the MEMS device and the ESN both learned the variability between subjects and speeds to achieve similar classification performance and to perform better than the linear classifier. The MEMS device further had to learn the non-ideal behavior of its acceleration sensor so that it was arguably solving a more difficult task. It could, of course, be expected that more elaborate classifier software could achieve better performance with the commercial accelerometer data; the comparison was performed here with an ESN with optimized hyperparameters and with the same number of reservoir nodes as the number of virtual activations in the MEMS device (see the “Methods” subsection “ESN and logistic regression” for details).

This demonstration that the MEMS solves the gait classification task, as well as an ESN, further allows a direct comparison of the energy consumed by our system and by a conventional electronics system. The MEMS was packaged on a small electronic board that was used to implement some non-essential functions such as data logging, subsystem control, wireless communication, voltage level conversion, and parameter tuning for the signal processing chain. It was also used to implement the functions that are essential for the current MEMS, including the battery, a high voltage drive with an amplitude modulator, an analog-to-digital converter, a microcontroller implementing a buffer of activation values, and a scalar product with a weight vector, a digital-to-analog converter, and a signal processing chain to measure the position of the beam. With all functions active, the power consumed by our prototype system was measured to be  $970 \pm 10$  mW. A system executing the ESN on a state-of-the-art microcontroller and using data from a commercial accelerometer was measured to similarly consume  $280 \pm 40$  mW of power. By breaking down the power consumption of our prototype by subsystems, eliminating the non-essential functions, and improving some simple elements of the MEMS design, we can calculate that a better-optimized version of our system could consume only 94 mW of power, already much better than the heavily optimized ESN system built with commercial components. More complex design modifications could further reduce the power consumption below 12 mW, for instance, by co-integrating the control electronics with the MEMS<sup>40</sup> or increasing the quality factor of the MEMS resonator with vacuum packaging<sup>41</sup> (see the “Methods” subsection “Power consumption” for details). The lowest power consumption could arguably be achieved by eliminating the feedback circuitry to create activations entirely in the mechanical domain and by forming the linear combination of the activations in the analog domain, as discussed in the “Methods” subsection “Power consumption”.

## Discussion

We have described a device that is both an acceleration sensor and a neuromorphic computer and that performs its sensing and nonlinear computing functions in the mechanical domain via the displacement of suspended microstructures. This wearable device was shown to successfully implement the concept of in-sensor computing for the difficult task of identifying the gait pattern used by a human subject, using only the acceleration measured on one foot. The most important characteristics of the device that enabled this successful demonstration were its computing capabilities (linear classifiers cannot solve the gait classification task), robustness against data variations and non-ideal sensor behavior, small size, and low power consumption. These characteristics are natural benefits of the co-integration of the sensing and neuromorphic computing functions in a single device. Another benefit of in-sensor computing is that the sensor data never leave the device or are never actually converted to the electrical or digital domains, thus offering a high level of privacy that could be especially relevant for medical devices.

These benefits that are important for medical devices are also highly desirable for edge computing devices in applications for the Internet of

Things. In these applications, in-sensor computing could further allow the amount of data transmitted by the edge sensors to be drastically reduced, from the full (compressed) bandwidth at the Nyquist frequency of the signal down to the rate of identification of relevant features in the signal. In addition to alleviating the data congestion issues that are typical of Internet of Things applications<sup>42</sup>, this could greatly increase the battery life of edge sensors, thus facilitating their field deployment at a large scale. Our demonstration of a single device that solves a complex real-world task by performing both sensing and neuromorphic computing functions directly in the mechanical domain can thus be considered an important milestone for the broad deployment of sensors and machine learning capabilities in emerging applications for wearable medical devices and for the Internet of Things.

## Methods

### MEMS design and fabrication

The suspended inertial mass and doubly clamped beam were defined by photolithographically patterning their shapes in the AZ MIR 701 photoresist, deep reactive-ion etching (Bosch process) of the 50  $\mu\text{m}$  device layer of a P-type (boron dopant, 0.02  $\Omega\text{cm}$ ) silicon on insulator (SOI) substrate, and then releasing by HF vapor etching of the 4  $\mu\text{m}$  buried oxide (BOX) through perforations in the proof mass (10  $\mu\text{m}$ -side squares on a pitch of 20  $\mu\text{m}$ ), while their anchors (minimum width of 35  $\mu\text{m}$ ) remained attached to the 350  $\mu\text{m}$ -thick handle layer due to their larger surface area and absence of perforations. The device was finalized by a metallization step where a stainless steel laser-cut stencil attached over the die allowed the deposition of a Cr–Au film over the electrical traces using electron-beam evaporation. More details of the fabrication process are given elsewhere<sup>11</sup>.

The 50  $\mu\text{m}$  thickness of the device layer provided sufficient out-of-plane stiffness to prevent the pull-in of the inertial mass towards the handle across the 4  $\mu\text{m}$  air gap left by the release step (etching of the BOX), but more importantly, provided substantial mass (49  $\mu\text{g}$ ) in a small footprint (1  $\text{mm}^2$ ), thus providing a sufficient sensitivity to uniaxial accelerations. Four pairs of 3  $\mu\text{m}$ -wide folded flexure springs (436 and 488  $\mu\text{m}$  long segments) were used to suspend the inertial mass. They were tuned to provide sufficient longitudinal displacement in reaction to accelerations (6.8 N/m spring constant) while restricting transverse and rotational displacements. An 8  $\mu\text{m}$ -wide electrostatic transduction gap separated the inertial mass from the beam oscillator. The beam width (4  $\mu\text{m}$ ) and length (300  $\mu\text{m}$ ) were adjusted<sup>10,11</sup> for fast oscillations (much above the resonance frequency of the inertial mass, in a range where the resonator is not sensitive to inertial forces), as well as to easily achieve nonlinear dynamics through mechanical stiffening at large displacements. Thin 1.5  $\mu\text{m} \times 12$   $\mu\text{m}$  piezoresistive strain gauge pairs attached to the beam were included in the design to allow the differential piezoresistive measurement of its oscillations. Sets of stopper structures (18  $\mu\text{m}$  diameter half-discs) surrounded the inertial mass in order to limit its range of motion to 5  $\mu\text{m}$ , to protect the device from shocks and electrostatic pull-in, which could otherwise damage the device by short-circuiting the capacitive transduction gap.

### MEMS packaging and board integration

A schematic description of the system is shown in Fig. 4. The beam oscillation amplitude signals were preprocessed by an analog measurement circuit before being sampled by the analog-to-digital converter. The digitized samples were then fed to a leaky integrator which outputs the virtual node activations that were used during inference by the output layer, which was trained using a ridge regression. The activations were also delayed and fed back to the MEMS through the high-voltage drive circuit driven by a digital-to-analog converter. The detected gait pattern was obtained by thresholding the temporally averaged linear combinations of the activations.

The MEMS die was wire bonded to a 20 mm  $\times$  20 mm printed circuit board (PCB) chip carrier, and a plastic cover was attached to the PCB in order to protect the device from its environment (see Fig. 2c for a photograph of the packaged MEMS). Fine-pitch male headers soldered on the flip





### ESN and logistic regression

The ESN was implemented in the software. It used the single-channel acceleration signal  $u(n)$  from the reference accelerometer as input and produced a state  $x(n)$  according to

$$\tilde{x} = \tanh\left(W_{in} \begin{bmatrix} 1 \\ u(n) \end{bmatrix} + W_r x(n-1)\right) \quad (3)$$

$$x(n) = (1 - \alpha)x(n-1) + \alpha\tilde{x}(n), \quad (4)$$

where the hyperparameter  $\alpha$  is the leaking rate,  $W_{in} \in \mathbb{R}^{N \times 2}$  is the input weight matrix, and  $W_r \in \mathbb{R}^{N \times N}$  is the reservoir weight matrix. Both  $W_{in}$  and  $W_r$  were initialized randomly from uniform distributions (between  $-1$  and  $+1$ ), then sparsified by randomly replacing entries by 0 (with a probability given by the hyperparameter called the *sparsity probability*) and finally scaled by multiplying by a scaling hyperparameter (for  $W_{in}$ , one scaling hyperparameter was used for the input and another was used for the bias). The ESN hyperparameters were taken from an earlier study on a gait event detection task<sup>44</sup> and are given in Supplementary Note 6. In that study, the hyperparameters with the largest influence on the performance (the input scaling, the bias scaling, the spectral radius, and the leaking rate) were optimized by the CHARC method<sup>45</sup>, and the other hyperparameters were taken from previous studies and not optimized further since they did not have much influence on the performance. In the CHARC method, the hyperparameters were selected to maximize the ratio between the kernel rank (which measures the ability of the ESN to distinguish distinct inputs) and the generalization rank (which must be decreased to improve the ability of the ESN to recognize similar inputs). After discarding the initial 1000 timesteps to account for the initial ESN transient, the states from a recording with  $M$  timesteps were assembled in a matrix  $X \in \mathbb{R}^{(2+N) \times M}$ , together with a row of ones to add a bias term and a row corresponding to the acceleration signal. The rest of the procedure was then the same as for the MEMS system: for each split of the cross-validation, a matrix  $W_{out}$  was obtained by ridge regression, predictions were generated using the matrix-vector product of the state-bias-acceleration vector with  $W_{out}$  and a moving average of 300 timesteps was applied.

For the logistic regression, a digital low-pass filter (FIR filter with a Hamming window, cutoff frequency of 71.5 Hz, order 40) was applied to the acceleration signal from the reference accelerometer. The first 1000 acceleration values were removed in order to keep the same training data length as for the MEMS and ESN methods. Then, at each timestep  $n$ , the last 715 acceleration values (a window corresponding to the last 5 s) were down-sampled by a factor of 2, and the resulting 358 acceleration values, as well as the mean of the last 715 points, were used as features for the logistic regression. These features were standardized by subtracting their means and by dividing by their standard deviations, with the means and standard deviations obtained from the training dataset. A two-row matrix  $Y_p$  containing the probabilities for the TO and TL classes was calculated by applying the logistic function to a linear combination of the features plus a bias term. The parameters of the linear combination and the bias were obtained during training by a coordinate descent algorithm implemented in the liblinear solver in the scikit-learn package<sup>46</sup>. In this algorithm, the regularization parameter  $C$  was set to 1.0, and the maximum number of iterations was 100. The same procedure as for the MEMS and the ESN was then followed (for each split of the cross-validation,  $Y_p$  was computed, a moving average over 300 timesteps was applied, and performance was evaluated).

### Power consumption

In order to compare the power consumption of the MEMS system to a conventional implementation that uses separate sensor and processing components, the ESN described in section M4 was also implemented on an Adafruit Industries HUZAZH32 Feather board, that is built around an Espressif Systems ESP32 microcontroller. The board was connected to an STMicroelectronics ISM330DHCX inertial measurement unit. The WiFi,

radio, and Bluetooth functions were deactivated to save power. Power was supplied through the board USB connector, and everything was packaged in a 3D-printed plastic enclosure similar to the one of the MEMS system.

Power measurements were performed on our MEMS prototype as well as on the ESN running on the Feather system by connecting an ammeter in series with a 5 V benchtop power supply replacing the batteries or USB power of either system. Power dissipation was measured in inference mode, without data transmission. For the MEMS system, a simple LED board addressed by the microcontroller was used to provide real-time feedback about the detected gait pattern.

Current and projected energy requirements for the MEMS system were calculated by summing the power dissipation contribution of every electronic and electromechanical subsystem. These individual contributions, shown in Supplementary Table 3 and Supplementary Table 4, were obtained by first measuring the signal characteristics (root mean square (RMS) voltage ( $V_{out}$ ) and dominant frequency) at the output of each active device that powers these subsystems. For operational amplifiers, the dominant frequency of signals was then used to compute equivalent load impedances ( $Z_{load}$ ) from nominal values of passive load components. This in turn, allowed the conversion of the measured voltages into currents being sourced and sunk by the amplifiers ( $I_{load}$ ). Finally, an estimate of power dissipation of all subcircuits ( $P_{total}$ ) was obtained by adding the integrated circuit quiescent power dissipation ( $P_{quiescent} = V_{supply} I_{quiescent}$ , where  $V_{supply}$  is the supply voltage difference, and  $I_{quiescent}$  is the quiescent current, taken from the datasheet) to the power dissipated due to the application of a signal ( $P_{load} = V_{supply} I_{load}/2$ , since analog signals are bipolar and biased at 0 V). Power dissipation values for the microcontroller, the voltage-controlled oscillator (VCO) and the three voltage regulators were obtained from their respective datasheets. The power outputs of the voltage regulators, calculated using values from Supplementary Tables 3 and 4, were used to deduce their power dissipation through the efficiency curves obtained from their respective datasheets. The high-voltage amplifier power dissipation was deduced by disconnecting it from the system and by measuring the subsequent drop in the power supplied to the system.

A total power requirement of 958 mW was calculated with this method for the current version of the MEMS system (see Supplementary Table 3), consistently with the measured value of  $970 \pm 10$  mW. This level of agreement validates our power analysis methodology, which thus provides a way to identify subsystems that can be optimized and to calculate expected power requirements for future designs. As detailed in Supplementary Table 4, a large reduction in power consumption could be achieved by simply changing the currently supplied voltages ( $+3.3$ ,  $+5$ , and  $-5$  V) to a single 2.5 V power rail, selecting lower power active chips and scaling component values accordingly, and eliminating unnecessary or data logging subcircuits such as the EEPROM chip, some voltage offset and pre-amplification stages. In addition, the multiplier integrated circuit, which dissipates an order of magnitude more power than other active components in our current implementation, could be replaced by a digital potentiometer controlling the gain of a voltage amplifier in order to implement amplitude modulation more efficiently. With these straightforward modifications, the power consumption would be reduced from its current value of  $970 \pm 10$  mW (measured) down to 94 mW (calculated).

Lower power consumption levels could arguably be achieved by eliminating the feedback circuitry to create activations entirely in the mechanical domain. This could be achieved using multiple resonators<sup>47</sup> or multiple proof masses<sup>48</sup>. A completely mechanical implementation with the number of activations (approximately 100) and memory (on timescales on the order of seconds) required for complex time-series processing (such as gait analysis) has yet to be experimentally demonstrated, but hybrid systems with a few resonators or proof masses<sup>25,49</sup> could be a stepping stone toward this goal, that proportionally reduces the power consumption of the feedback electronics. Any fully integrated system should also have efficient drive and read-out mechanisms with efficient electrical implementations<sup>50</sup>. Finally, a complete MEMS system should implement a mechanism to adjust the weights in the linear combination of all its activation functions, to



support learning. It is unlikely that this could be achieved by tuning the structure of the MEMS, due to the economical costs of creating lithographic masks and, more fundamentally, to the inherent variability of the fabricated devices. An intriguing possibility would be to perform the linear combination on analog signals instead of in the digital domain, using an array of adaptable elements to adjust the weights, such as memristors<sup>51</sup>. The non-linear processing of the physical signals could be performed efficiently by MEMS resonators with a high-quality factor, while low currents would flow from the memristor array into a summing amplifier with a high input impedance.

### Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

### Data availability

The recorded reference acceleration signals and the virtual activation signals for all participants, gait patterns, and walking speeds are available online<sup>52</sup>.

Received: 13 September 2023; Accepted: 5 March 2024;

Published online: 16 March 2024

### References

1. Yasuda, H. et al. Mechanical computing. *Nature* **598**, 39–48 (2021).
2. Charlot, B., Sun, W., Yamashita, K., Fujita, H. & Toshiyoshi, H. Bistable nanowire for micromechanical memory. *J. Micromech. Microeng.* **18**, 045005 (2008).
3. Hafiz, Ma. A., Kosuru, L. & Younis, M. I. Microelectromechanical reprogrammable logic device. *Nat. Commun.* **7**, 11137 (2016).
4. Jin, X., Li, L., Dang, F., Chen, X. & Liu, Y. A survey on edge computing for wearable technology. *Digit. Signal Process.* **125**, 103146 (2022).
5. Dubcek, T. et al. Metamaterials you can talk to: speech recognition with elastic neural networks. *J. Acoust. Soc. Am.* **151**, A129 (2022).
6. Sylvestre, J. & Morissette, J.-F. Neuromorphic metamaterial structures. *Mater. Design* **210**, 110078 (2021).
7. Krushynska, A. O. et al. Emerging topics in nanophononics and elastic, acoustic, and mechanical metamaterials: an overview. *Nanophotonics* **12**, 659–686 (2023).
8. Zhou, F. & Chai, Y. Near-sensor and in-sensor computing. *Nat. Electron.* **3**, 664–671 (2020).
9. Tanaka, G. et al. Recent advances in physical reservoir computing: a review. *Neural Netw.* **115**, 100–123 (2019).
10. Dion, G., Mejaouri, S. & Sylvestre, J. Reservoir computing with a single delay-coupled non-linear mechanical oscillator. *J. Appl. Phys.* **124**, 152132 (2018).
11. Barazani, B., Dion, G., Morissette, J.-F., Beaudoin, L. & Sylvestre, J. Microfabricated neuroaccelerometer: integrating sensing and reservoir computing in MEMS. *J. Microelectromech. Syst.* **29**, 338–347 (2020).
12. Appeltant, L. et al. Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 1–6 (2011).
13. Brunner, D. et al. Tutorial: photonic neural networks in delay systems. *J. Appl. Phys.* **124**, 152004 (2018).
14. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
15. Leroux, N. et al. RF signal classification in hardware with an RF spintronic neural network. ArXiv:2211.01131 [cond-mat] (2022).
16. Sun, L. et al. In-sensor reservoir computing for language learning via two-dimensional memristors. *Sci. Adv.* **7**, eabg1455 (2021).
17. Soriano, M. C. et al. Delay-based reservoir computing: noise effects in a combined analog and digital implementation. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 388–393 (2015).
18. Paquot, Y. et al. Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
19. Torrejon, J. et al. Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428–431 (2017).
20. Zhong, Y. et al. A memristor-based analogue reservoir computing system for real-time and power-efficient signal processing. *Nat. Electron.* **5**, 672–681 (2022).
21. Takemura, H., Mizumoto, T., Banerjee, A., Hirotsu, J. & Tsuchiya, T. Physical reservoir computing using nonlinear MEMS resonator having high memory capacity at “Edge of Chaos”. In *2023 IEEE 36th International Conference on Micro Electro Mechanical Systems (MEMS)*, 515–518 (IEEE, 2023).
22. Mizumoto, T., Hirai, Y., Banerjee, A. & Tsuchiya, T. Mem reservoir computing using frequency modulated accelerometer. In *2022 IEEE 35th International Conference on Micro Electro Mechanical Systems Conference (MEMS)*, 487–490 (IEEE, 2022).
23. Guo, X. et al. Input-output-improved reservoir computing based on duffing resonator processing dynamic temperature compensation for MEMS resonant accelerometer. *Micromachines* **14**, 161 (2023).
24. Sun, J. et al. Enhancing the recognition task performance of MEMS resonator-based reservoir computing system via nonlinearity tuning. *Micromachines* **13**, 317 (2022).
25. Zheng, T. et al. Enhancing performance of reservoir computing system based on coupled MEMS resonators. *Sensors* **21**, 2961 (2021).
26. Zheng, T. et al. Processing IMU action recognition based on brain-inspired computing with microfabricated MEMS resonators. *Neuromorph. Comput. Eng.* **2**, 024004 (2022).
27. Sun, J. et al. Novel nondelay-based reservoir computing with a single micromechanical nonlinear resonator for high-efficiency information processing. *Microsyst. Nanoeng.* **7**, 1–11 (2021).
28. Antonsson, E. K. & Mann, R. W. The frequency content of gait. *J. Biomech.* **18**, 39–47 (1985).
29. MacDougall, H. G. & Moore, S. T. Marching to the beat of the same drummer: the spontaneous tempo of human locomotion. *J. Appl. Physiol.* **99**, 1164–1173 (2005).
30. Rynne, R., Le Tong, G., Cheung, R. T. H. & Constantinou, M. Effectiveness of gait retraining interventions in individuals with hip or knee osteoarthritis: a systematic review and meta-analysis. *Gait Posture* **95**, 164–175 (2022).
31. Hulleck, A. A., Menoth Mohan, D., Abdallah, N., El Rich, M. & Khalaf, K. Present and future of gait assessment in clinical practice: towards the application of novel trends and technologies. *Front. Med. Technol.* **4**, 901331 (2022).
32. Tao, W., Liu, T., Zheng, R. & Feng, H. Gait analysis using wearable sensors. *Sensors* **12**, 2255–2283 (2012).
33. Roetenberg, D., Luinge, H. & Slycke, P. *Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors*. Technical Report (Xsens Motion Technologies BV, 2013).
34. Kawai, H. et al. Intra-day variation in daily outdoor walking speed among community-dwelling older adults. *BMC Geriatr.* **21**, 417 (2021).
35. Napier, C., Esculier, J.-F. & Hunt, M. A. Gait retraining: out of the lab and onto the streets with the benefit of wearables. *Br. J. Sports Med.* **51**, 1642–1643 (2017).
36. Thuruthel, T. G., Shih, B., Laschi, C. & Tolley, M. T. Soft robot perception using embedded soft sensors and recurrent neural networks. *Sci. Robot.* **4**, eaav1488 (2019).
37. Oldfrey, B., Jackson, R., Smitham, P. & Miodownik, M. A deep learning approach to non-linearity in wearable stretch sensors. *Front. Robot. AI* **6**, 27 (2019).
38. Patra, J., Chakraborty, G. & Meher, P. Neural-network-based robust linearization and compensation technique for sensors under nonlinear environmental influences. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **55**, 1316–1327 (2008).
39. Jaeger, H. *The “Echo State” Approach to Analysing and Training Recurrent Neural Networks*. Technical Report (German National

- Research Center for Information Technology GMD, Bonn, Germany, 2001).
40. Fedder, G. K. et al. Monolithically integrated inertial sensors. In *Advanced Micro and Nanosystems* 1 edn (eds Brand, O. & Fedder, G. K.) 137–191 (Wiley, 2005).
  41. Asadian, M. H., Askari, S. & Shkel, A. M. An ultrahigh vacuum packaging process demonstrating over 2 million Q-factor in MEMS vibratory gyroscopes. *IEEE Sensors Lett.* **1**, 1–4 (2017).
  42. Srinidhi, N., Dilip Kumar, S. & Venugopal, K. Network optimizations in the Internet of Things: a review. *Eng. Sci. Technol.* **22**, 1–21 (2019).
  43. Chen, J. et al. Optoelectronic graded neurons for bioinspired in-sensor motion perception. *Nat. Nanotechnol.* **18**, 882–888 (2023).
  44. Chiasson-Poirier, L., Younesian, H., Turcot, K. & Sylvestre, J. Detecting gait events from accelerations using reservoir computing. *Sensors* **22**, 7180 (2022).
  45. Dale, M., Miller, J. F., Stepney, S. & Trefzer, M. A. A substrate-independent framework to characterize reservoir computers. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **475**, 20180723 (2019).
  46. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
  47. Coulombe, J. C., York, M. C. A. & Sylvestre, J. Computing with networks of nonlinear mechanical oscillators. *PLoS ONE* **12**, e0178663 (2017).
  48. Nikfarjam, H., Megdadi, M., Okour, M., Pourkamali, S. & Alsaleem, F. Energy efficient integrated MEMS neural network for simultaneous sensing and computing. *Commun. Eng.* **2**, 19 (2023).
  49. Ortín, S. & Pesquera, L. Reservoir computing with an ensemble of time-delay reservoirs. *Cogn. Comput.* **9**, 327–336 (2017).
  50. Duport, F., Smerieri, A., Akrouf, A., Haelterman, M. & Massar, S. Fully analogue photonic reservoir computer. *Sci. Rep.* **6**, 22381 (2016).
  51. Prezioso, M. et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
  52. Dion, G. et al. *Dataset—In-sensor Human Gait Analysis With Machine Learning in a Wearable Microfabricated Accelerometer* <https://zenodo.org/records/10593481> (2024).

## Acknowledgements

The authors thank Mathieu Bergeron, Cyril Bounakoff, and Adrien Merel for their help. J.S. and K.T. disclose support for the research of this work from the New Frontiers in Research Fund [grant number NFRFE-2018-00598].

## Author contributions

Guillaume Dion: Formal analysis, investigation, resources, software, writing—original draft, Albert Tessier-Poirier: Formal analysis, software, writing—

original draft, Laurent Chiasson-Poirier: Formal analysis, investigation, software, Jean-François Morissette: Writing—review & editing, project administration, Guillaume Brassard: Software, Anthony Haman: Software, Katia Turcot: Funding acquisition, Julien Sylvestre: Conceptualization, writing—original draft, supervision, funding acquisition.

## Competing interests

The authors declare the following competing interests: a patent application was filed by Guillaume Dion and Julien Sylvestre for the MEMS technology used in this paper (WO/2020/124219). The other authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s44172-024-00193-5>.

**Correspondence** and requests for materials should be addressed to Julien Sylvestre.

**Peer review information** *Communications Engineering* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editors: Anastasiia Vasylychenkova and Rosamund Daw. A peer review file is available.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024