



**HAL**  
open science

## **Role of non-linear data processing on speech recognition task in the framework of reservoir computing**

Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Sumito Tsunegi, Damien Querlioz, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D Stiles, et al.

### ► **To cite this version:**

Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Sumito Tsunegi, Damien Querlioz, et al.. Role of non-linear data processing on speech recognition task in the framework of reservoir computing. Scientific Reports, 2020, 10 (1), 10.1038/s41598-019-56991-x . hal-03041228

**HAL Id: hal-03041228**

**<https://hal.science/hal-03041228>**

Submitted on 4 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

OPEN

# Role of non-linear data processing on speech recognition task in the framework of reservoir computing

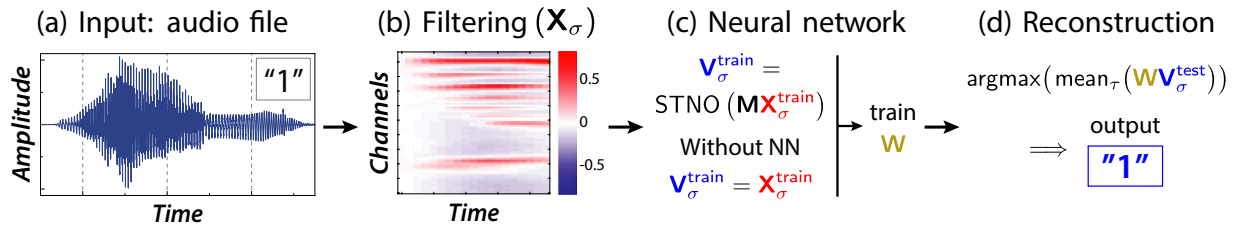
Flavio Abreu Araujo<sup>1\*</sup>, Mathieu Riou<sup>2</sup>, Jacob Torrejon<sup>3</sup>, Sumito Tsunegi<sup>4</sup>, Damien Querlioz<sup>5</sup>, Kay Yakushiji<sup>4</sup>, Akio Fukushima<sup>4</sup>, Hitoshi Kubota<sup>4</sup>, Shinji Yuasa<sup>4</sup>, Mark D. Stiles<sup>6</sup> & Julie Grollier<sup>2</sup>

The reservoir computing neural network architecture is widely used to test hardware systems for neuromorphic computing. One of the preferred tasks for bench-marking such devices is automatic speech recognition. This task requires acoustic transformations from sound waveforms with varying amplitudes to frequency domain maps that can be seen as feature extraction techniques. Depending on the conversion method, these transformations sometimes obscure the contribution of the neuromorphic hardware to the overall speech recognition performance. Here, we quantify and separate the contributions of the acoustic transformations and the neuromorphic hardware to the speech recognition success rate. We show that the non-linearity in the acoustic transformation plays a critical role in feature extraction. We compute the gain in word success rate provided by a reservoir computing device compared to the acoustic transformation only, and show that it is an appropriate bench-mark for comparing different hardware. Finally, we experimentally and numerically quantify the impact of the different acoustic transformations for neuromorphic hardware based on magnetic nano-oscillators.

Artificial neural network algorithms outperform humans on recognition tasks like image or speech recognition, by leveraging deep networks of interconnected non-linear units called formal neurons<sup>1</sup>. The goal of neural networks is to extract the features and classify input data through learned non-linear transformation. Running such algorithms on a classical computer is costly energetically: to overcome this issue, neuromorphic approaches<sup>2,3</sup> propose to implement them physically. In particular, reservoir computing<sup>4,5</sup> is a kind of recurrent neural network that has been widely used to test the efficiency of hardware for neuromorphic computing<sup>6–8</sup> because it has a simplified architecture and learning procedure. The input is sent to a neural network with fixed recurrent connections called a reservoir. The goal of the reservoir is to separate the different kinds of inputs, such that after this transformation, the classification can be done by a linear transformation. The response of the neurons of the reservoir are combined linearly with trained connections to construct the output. Since the connections in the reservoir are random and fixed, it is easier to fabricate it in hardware and then train the output connections, often emulated in software, with linear regression.

Speech recognition is a widely used class of benchmark tasks performed to test the efficiency of a neural network. It is especially employed in the case of reservoir computing because the recurrent connections of the reservoir create an intrinsic memory that is useful to classify time-varying inputs. Generally, this task requires frequency decomposition<sup>9–11</sup> prior to the neural network because the acoustic features are contained in the frequency rather than in the amplitude of the time-varying signal. These decompositions return the amplitude of the signal in different frequency channels as a function of time. The neural network then extracts the acoustic features contained in the frequency information. Several frequency decomposition methods have been reported in the literature: Mel-frequency cepstral coefficients (MFCC) and Lyon's cochlear model (cochleagram) are the most

<sup>1</sup>Institute of Condensed Matter and Nanosciences, Université catholique de Louvain, Place Croix du Sud 1, 1348, Louvain-la-Neuve, Belgium. <sup>2</sup>Unité Mixte de Physique, CNRS, Thales, Université Paris-Sud, Université Paris-Saclay, 91767, Palaiseau, France. <sup>3</sup>Service de Physique de l'Etat Condensé, DSM/IRAMIS/SPEC CNRS UMR 3680 CEA Saclay, 91191, Gif-sur-Yvette Cedex, France. <sup>4</sup>National Institute of Advanced Industrial Science and Technology (AIST), Spintronics Research Center, Tsukuba, Ibaraki, 305-8568, Japan. <sup>5</sup>Centre de Nanosciences et de Nanotechnologies, CNRS, Université Paris-Sud, Université Paris-Saclay, 91405, Orsay, France. <sup>6</sup>Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, Maryland, 20899-6202, USA. \*email: [flavio.abreuaraujo@uclouvain.be](mailto:flavio.abreuaraujo@uclouvain.be)



**Figure 1.** Principle of spoken digit recognition. (a) Audio waveform corresponding to the digit 1 pronounced by speaker 1. (b) Filtering to frequency channels for acoustic feature extraction. The signal during each time interval  $\tau$  is decomposed in  $N_f$  frequency channels. The cochlear model filters each point of the audio waveform in 78 frequency channels (13 in the case of the MFCC model and 65 for the spectrogram model). The frequency channels are concatenated in intervals of duration  $\tau$  to form the filtered input. (c) The filtered input is injected in the neural network or directly used to construct the output (No neural network). The neural network is composed of  $N$  interconnected filtered inputs. (d) For each digit, the response of the neural network (or directly the filtered output) is constructed from a linear combination of neuron states  $V_{\theta_r, \sigma}$  (there are 10 classifiers in total).

common methods since they mimic the filtering that occurs biologically<sup>9,12,13</sup>. However, the actual contribution of the acoustic filter to the total speech recognition rate is generally not investigated while performing speech recognition benchmarks with reservoir computing hardware, even if its influence on the final recognition rate may not be negligible<sup>8</sup>. Furthermore, both of these methods were developed before reservoir computing became popular and, thus they were designed to extract the useful features of an audio signal independent of modern machine learning.

Here, we first show how the choice of different filtering methods drastically affects the final speech recognition rate. We quantify the contributions of the acoustic filtering and the neural network, respectively, for a spoken digit recognition task using four frequency decomposition methods with different non-linear characters: Lyon's ear cochleagram, MFCC filter, linear spectrogram ( $\Re(\text{Spectro})$ ), and Spectro HP ( $\text{Spectro HP} = |\sin \sqrt{|\Re(\text{Spectro})|} - |\cos \sqrt{|\Im(\text{Spectro})|}|$ ). In a first step, we show that the cochleagram, the Spectro HP and the MFCC filter are powerful stand-alone features extractors that can achieve by themselves (without additional processing by a neural network) very high recognition levels: up to 95.8%, 89.0%, and 77.2% for cochleagram, Spectro HP, and MFCC, respectively. In contrast, the linear spectrogram never achieves recognition levels statistically better than random sampling, 10%. However, by adding various levels of non-linearity to the real part of the spectrogram we were able to show a large increase of the recognition rate from about 10% (linear) to 88% (strong non-linearity). These results indicate that the high recognition level of the cochleagram and MFCC approaches is mainly due to the non-linear character of these frequency decomposition methods and not to the reservoir itself.

In a second step, we evaluate the gain in recognition rate provided by a particular hardware approach to reservoir computing, based on magnetic nano-oscillators. In order to compare to other hardware implementations in the literature, we model a neural network based on a single dynamical non-linear magnetic node in the framework of the reservoir computing approach<sup>6–8,14–16</sup>. We find that the contribution of the neural network is dominant for linear spectrogram filter and only plays a small role for the non-linear cochleagram and MFCC filter. Finally, we present experimental results using a non-linear and tunable magnetic nano-oscillator exhibiting excellent agreement with our simulations.

## Methods

We perform a benchmark task called spoken digit recognition that is common in the reservoir computing community for software<sup>10</sup> and hardware<sup>7,8,14,17–21</sup> implementations. The input data, taken from the TI-46 database, are audio waveforms of clean isolated spoken digits (0 to 9) pronounced by five different female speakers (see example in Fig. 1a), as it is usual in the hardware reservoir computing community<sup>6–8,10,14,17,20,22–25</sup>.

The chosen part of the TI-46 spoken digit database contains 500 (5 speakers  $\times$  10 digits  $\times$  10 utterances) audio files, which we index using the Greek letter  $\sigma$ . To perform speech recognition on these spoken digits, each audio temporal trace in the database is transformed from time-domain to a mixed time/frequency domain with different acoustic filters, two of which are known to create a better representation of human voice characteristics. These acoustic filters give rise to different instances of our speech database containing the following elements:  $X_{f_r, \sigma}^{\text{filter}}$  where filter  $\in \{\text{Cochlear}, \text{MFCC}, \text{Spectro}, \text{Spectro HP}\}$ ,  $f$  is the index for the different frequency channels, and  $\tau$  is the index of a new time representation that depends on the time frame window used while performing the time- to frequency-domain transformation. The number of time steps  $N_\tau$  naturally depends on the digit length, while the number of frequency channels  $N_f$  only depends the type of acoustic filter. For instance,  $N_f^{\text{Cochlear}} = 78$ ,  $N_f^{\text{MFCC}} = 13$ , and  $N_f^{\text{Spectro}} = 65$  while  $N_\tau^{\text{Cochlear}}$  ranges from 16 to 41,  $N_\tau^{\text{MFCC}}$  ranges from 31 to 83, and  $N_\tau^{\text{Spectro}}$  ranges from 24 to 67. Digits with  $N_\tau$  lesser than the maximum value are padded with zeros.

The construction of the supervised learning task in the reservoir computing framework starts with associating each digit  $X_\sigma$  with its corresponding target  $T_\sigma \in \mathbb{R}^{N_d \times N_\tau}$  where  $N_d$  is the number of categories to classify (here  $N_d = 10$  as the goal is to recognise the 10 different digits). Each target matrix  $T_\sigma$  is constructed column-wise and the  $N_\tau$  columns correspond to the same target vector  $t_\sigma$  (the target vector  $t_{d, \sigma}$  with  $d \in [0..9]$  is zero almost everywhere but is one where  $d$  is equal to the corresponding digit number). The  $T_\sigma$  matrices would allow us to perform

$\tau$ -wise recognition (partial digits for instance) but in this study we choose to make entire-digit-wise recognition by averaging out the estimated target matrices  $\hat{\mathbf{t}}_\sigma$  over the different columns ( $\tau$  direction) to end up with estimators  $\hat{\mathbf{t}}_\sigma$  of the target vectors  $\mathbf{t}_\sigma$  as shown below (see Eq. (3)).

Our reservoir is a time multiplexed single device as described in ref.<sup>8</sup>. Rather than a set of  $N_\theta$  physical neurons, our reservoir consists of a single physical neuron evaluated at  $N_\theta$  periodic times. To input the data to these virtual neurons, we multiply each value by a time series of length  $N_\theta$  consisting of ones and minus ones and send the resulting time series to the single device. The output of the reservoir is determined by the resulting state of the device at each of the  $N_\theta$  times for each element of the input data string. This output is multiplied by the output weight matrix to give the results. Training consists of determining the optimum set of output weights, which can be found through straightforward linear algebra.

The key computational concept supporting the reservoir computing approach is a nonlinear dynamical transformation of the processed information, i.e. sending the input data to a new space, in which simple linear algebra gives the read-out of the results<sup>26</sup>. In this work, the non-linear transformation is the purpose of our spin-torque nano oscillator represented by the function STNO( $\cdot$ ) in Eq. (1). The information is then encoded and injected into this nonlinear dynamical system after flattening the data and multiplying each element of the flattened  $\mathbf{X}_\sigma$  by a random binary mask  $\mathbf{M} \in \mathbb{R}^{N_\theta \times N_f}$  of 1's and  $-1$ 's. This binary masks starts the time-multiplexing technique as the value times the mask gives the input to each virtual neuron. As a result, the mask distributes the frequency content of each time step  $\tau$  of the input data into a fixed neural network layer (the reservoir) of  $N_\theta$  nodes. To summarise, Eq. (1) shows the details of our Reservoir Computing implementation:

$$\mathbf{X}_\sigma \rightarrow \mathbf{V}_\sigma: \begin{cases} \mathbf{x}_\sigma = \text{flatten}(\mathbf{M}\mathbf{X}_\sigma) & x_{i,\sigma} = \text{flatten}\left(\sum_f M_{\theta f} X_{f,\sigma}\right) \text{ with } i \in [1.. N_\theta N_f], \\ \mathbf{v}_\sigma = \text{STNO}(\mathbf{x}_\sigma) & v_{i,\sigma} = \text{STNO}(x_{i,\sigma}), \\ \mathbf{V}_\sigma = \text{reshape}(\mathbf{v}_\sigma) & V_{\theta\tau,\sigma} = \text{reshape}(v_{i,\sigma}), \end{cases} \quad (1)$$

where  $\text{flatten}(\cdot)$  takes a  $m$  by  $n$  matrix as input and outputs a vector of length  $mn$  and  $\text{reshape}(\cdot)$  does the reverse operation, i.e. takes a vector of length  $mn$  as input and outputs a  $m$  by  $n$  matrix.

Training (learning) is performed using a simple linear classifier. In this work, good performance is achieved using the Moore-Penrose pseudo-inverse after building the weight matrix  $\mathbf{W}$  optimisation problem with a subset of  $N_{\text{train}}$  digits  $\mathbf{W}[\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_{N_{\text{train}}}] = [\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_{N_{\text{train}}}]$ :

$$\mathbf{W} = [\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_{N_{\text{train}}}] [\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_{N_{\text{train}}}]^{-1}. \quad (2)$$

No regularisation technique is used. The testing (recognition) step is then achieved using the computed weights  $\mathbf{W}$  applied to the complementary (unseen) subset of  $N_{\text{test}}$  digits:

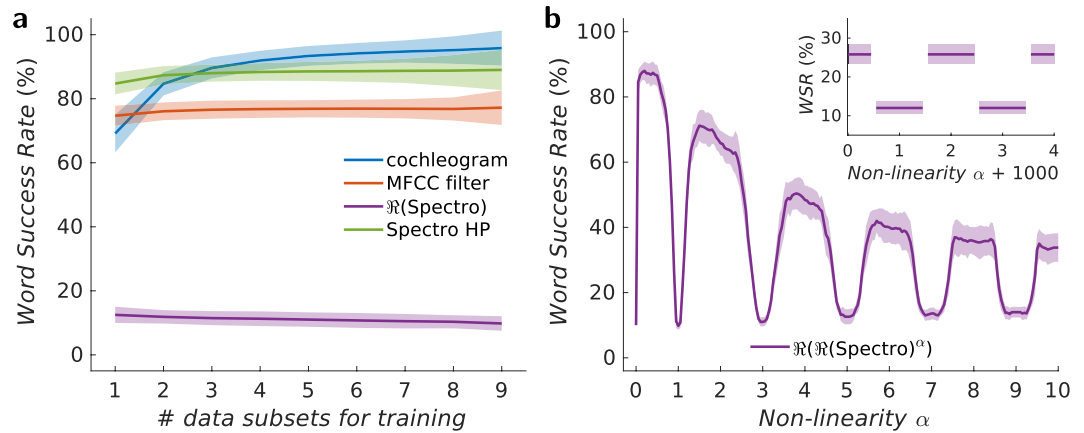
$$\begin{aligned} \hat{\mathbf{T}}_\sigma &= \mathbf{W}\mathbf{V}_\sigma & \hat{T}_{d\tau,\sigma} &= \sum_\theta W_{d\theta} V_{\theta\tau,\sigma} \\ \hat{\mathbf{t}}_\sigma &= \text{mean}_\tau(\hat{\mathbf{T}}_\sigma) & \hat{t}_{d,\sigma} &= \frac{1}{N_\tau} \sum_{\tau=1}^{N_\tau} \hat{T}_{d\tau,\sigma} \end{aligned} \quad (3)$$

The estimator for a specific digit is given by  $\hat{d}_\sigma = \text{argmax}(\hat{\mathbf{t}}_\sigma)$  (this corresponds to the Winner-Takes-All strategy, adequate for the present task). Digit  $\sigma$  is well recognised when  $\hat{d}_\sigma = \text{argmax}(\mathbf{t}_\sigma)$  and the main performance estimator used in this work is the Word Success Rate (WSR) and corresponds to the percentage of well recognised digits over the total number of digits to recognise ( $N_{\text{test}}$ ). Another common performance estimator, useful for identifying overfitting issues, is the Mean Squared Error (MSE):  $\text{MSE}(\hat{\mathbf{t}}_\sigma) = \mathbb{E}[(\hat{\mathbf{t}}_\sigma - \mathbf{t}_\sigma)^2]$ .

In all cases, the training and testing sets do not overlap. To avoid any learning bias while selecting samples randomly from the database when choosing the training and testing sets, we organise the 500 input files into 10 subsets of 50 files. Each subset contains one utterance of each digit pronounced by each of the five speakers. A random selection would produce an over-representation of some speakers and an under-representation of others. We take  $N$  utterance subsets (50 audio files, one for each digit and each speaker) for training (total training set size  $N \times 50$ ), and  $10 - N$  utterance subsets for testing (total testing set size  $(10 - N) \times 50$ ). To minimise the fluctuations that occur in the results due to random choices between the training and testing sets, we employ a cross-validation technique and therefore average over all possible choices. That is, when  $N$  utterance subsets are used for training, we average over  $10!/[N!(10 - N)!]$  possible ways to choose the training and testing sets. This procedure also allows us to determine a width to the distribution of individual outcomes indicated by the shaded regions in Fig. 2. All word success rate results reported in the paper are cross-validated test results.

In reservoir computing, training is fast and always converges due to the basic linear algebra algorithms. This behaviour stands in contrast to standard recurrent neural-network approaches for which learning can be time consuming and does not necessarily converge to the desired solution. In reservoir computing, the learning process only modifies the read-out weights whereas in other types of recurrent neural network it modifies the weights in all the other constituent layers in the neural network under complex feed-forward/back-propagation algorithms.

The contribution of the frequency filtering and the reservoir computing, respectively, are then analysed separately. In order to evaluate the impact of the frequency filtering on the input separation capability, a linear



**Figure 2.** Spoken digit recognition for filtered inputs. **(a)** Spoken digit cross-validated test recognition rates as a function of the number of data subsets  $N$  used for training (total size of the training set  $5 \times 10 \times N$ ) of the filtered input (without neural network) corresponding to four different methods: cochleogram, MFCC filter, Spectro HP and linear spectrogram ( $\alpha = 1$ ). **(b)** Spoken digit recognition as a function of non-linear coefficient for spectrogram methods (Inset: Word success rate for large non-linear coefficient values from 1000 to 1004). Here, 9 data subsets (90% of the database) are used for training our reservoir computing model and the remaining subset (10% of the database) is used to perform the recognition task. The shaded region corresponds to the uncertainty of the recognition rate, here the standard deviation).

classifier is trained directly on the different frequency channels. The classification results with both influence of the frequency filtering and the reservoir are computed by injecting the filtered input in a neural network composed of  $N_\theta$  interconnected neurons. Here, we use  $N_\theta = 400$  input neurons that are connected to all of the frequency channels for each time interval  $\tau$ , Fig. 1c, as this number allowed reaching maximum test accuracy. In the framework of reservoir computing, these fixed connections have random weights. To reach high classification rate, 400 neurons are sufficient<sup>6,8</sup>. The features of the magnetic neurons that we consider are specified in section IV. A linear classifier is trained to map the neuron outputs to the desired results. The contribution of the reservoir to the ultimate success is extracted from the results by subtracting the success rate found using only the frequency filtering methods.

**Acoustic filter: role of non-linearity.** First, we compute the digit recognition rate as a function of the number of utterances used in training for the cochleogram and the MFCC methods as shown in Fig. 2(a). The recognition rate increases with the number of trained utterances and then saturates in the case of the cochlear model. It remains almost constant for the MFCC model. Both filters achieve a high recognition rate. In particular, the cochlear model is an excellent acoustic feature extractor with recognition rates up to 95.8% (for 9 trained utterances) whereas the MFCC filter is less powerful, reaching recognition rates up to 77.2%.

These filters are commonly used for speech recognition tasks, because of their similarity to audio signal processing in biological ears, which perform complex frequency decompositions with high non-linearities. Both MFCC and cochlear methods use non-linearities to transform the audio data. For MFCC the transformed representation corresponds to the log-energy of the Mel frequency filter output<sup>9</sup>. In the cochleogram approach, the main non-linear ingredient corresponds to a set of interconnected automatic gain controls<sup>12,13</sup>. The successful separation of the data achieved by these filtering methods appears to be mainly due to the non-linear character of the transformation with a moderate influence of the kind of non-linearity (similarly to reservoirs that can have different kinds of non-linearity that work).

To establish the critical role of the non-linearity contained in the filtering methods to recognition performance, we start by investigating the separation achieved by a very simple linear spectrogram filter. This filter is based on standard Fourier transforms of the audio input over finite time windows. The Fourier transform is a linear operation that outputs a real and an imaginary part. We consider only the real part in the following in order to avoid introducing non-linearities by computing the norm. After the Fourier transforms,  $\mathbf{Z}_\sigma$  is the matrix of the real parts of the spectrogram with dimension  $N_f \times N_\tau$  where  $N_f$  is the number of frequency channels and  $N_\tau$  is the number of time steps, which depends on the particular digit. We normalise the data,  $\mathbf{X}_\sigma = \mathbf{Z}_\sigma / \max(|\mathbf{Z}_\sigma|)$  and  $X_{f,\sigma} \in [-1, 1]$  for  $f \in \{1..N_f\}$  and  $\tau \in \{1..N_\tau\}$ . The normalisation is crucial to ensure that there exists at least one  $X_{f,\sigma}$  that is equal either to 1 or to  $-1$  for each  $\mathbf{X}_\sigma$  when non-linearities are introduced into the transform.

To study the influence of a non-linear transformation on the normalised input data  $\mathbf{X}_\sigma$ , we choose to apply a point-wise operation, namely the exponent  $\alpha \in \mathbb{R}$ , giving rise to the transformation on each element of  $X_{f,\sigma}^{\text{filter}} \rightarrow (X_{f,\sigma}^{\text{filter}})^\alpha$ . The impact of the non-linear exponent  $\alpha$  on the recognition rate is shown in Fig. 2(b). The recognition rate oscillates strongly as a function of the non-linear exponent and decreases for large  $\alpha$ . Some particular values of the recognition rate can be easily understood. For  $\alpha = 0: \forall i$  and  $j, X_{f,\sigma} = 1$ , and it becomes impossible to discriminate between different digits  $\mathbf{X}_\sigma$  and the success rate is equal to 10% (random choice). As  $\alpha$  approaches zero, the success rate decreases drastically and drops to 10%. For such exponents, all inputs get



mapped to the same output making data separation impossible. For  $\alpha = 1$  the real part of the spectrogram corresponds to a linear transformation of the input data, thus there is no non-linear data separation and the word recognition rate  $\approx 10\%$  (random choice).

The evolution shown in Fig. 2(b) can be understood by decomposing the exponent  $\alpha$  into an integer part  $n \in \mathbb{N}$  and a real part  $\varepsilon \in \mathbb{R}$  around  $n$  ( $\varepsilon \in ]-0.5, 0.5]$ ):  $\alpha = n + \varepsilon$ . For  $X_{fr,\sigma}^- < 0$ ,  $X_{fr,\sigma}^- \rightarrow (X_{fr,\sigma}^-)^{n+\varepsilon} = |X_{fr,\sigma}^-|^{n+\varepsilon} (-1)^n (\cos(\pi\varepsilon) + i \sin(\pi\varepsilon))$  and for  $X_{fr,\sigma}^+ \geq 0$ ,  $X_{fr,\sigma}^+ \rightarrow (X_{fr,\sigma}^+)^{n+\varepsilon} = |X_{fr,\sigma}^+|^{n+\varepsilon}$ . For simplicity, we choose to consider only the real part of the data obtained after applying the non-linearity, so  $\mathbf{R}_\sigma = X_{fr,\sigma} \rightarrow \Re(X_{fr,\sigma}^{n+\varepsilon})$ :

$$\mathbf{R}_\sigma = \begin{cases} \Re((X_{fr,\sigma}^-)^{n+\varepsilon}) = |X_{fr,\sigma}^-|^{n+\varepsilon} (-1)^n \cos(\pi\varepsilon) \\ \Re((X_{fr,\sigma}^+)^{n+\varepsilon}) = |X_{fr,\sigma}^+|^{n+\varepsilon} \end{cases} \tag{4}$$

From Eq. (4), for  $X_{fr,\sigma}^- < 0$  there is an additional factor  $(-1)^n \cos(\pi\varepsilon)$  compared to  $X_{fr,\sigma}^+ > 0$ . Consider the particular case where  $\varepsilon = 0$ , then in the case of values of  $X_{fr,\sigma}^-$  that were initially negative, the values  $R_{fr,\sigma}$  have the sign  $(-1)^n$ . So, depending on the parity of  $n$ , there are two possibilities. If  $n$  is even, there is at least one value  $R_{fr,\sigma}$  in  $\mathbf{R}_\sigma$  equal to 1 for each  $\mathbf{R}_\sigma$  (digit in the database). If  $n$  is odd, the  $\mathbf{R}_\sigma$  digits originating from an input  $\mathbf{X}_\sigma$  where at least one  $X_{fr,\sigma}^- = -1$  have a corresponding  $R_{fr,\sigma} = -1$  (at least one  $R_{fr,\sigma} = 1$  otherwise). Therefore, the oscillating behaviour of the success rate shown in Fig. 2b is related to what happens to the negative input data as shown in Eq. (4).

The poorer performance for the recognition task for odd  $n$  comes from the fact that the phase from the Fourier transform is essentially arbitrary. When  $n$  is even, the important elements of  $\mathbf{R}_\sigma$  are always positive, but for odd  $n$  they are sometimes positive and sometimes negative. The greater variation in the latter case makes it essentially impossible for the neural network to connect the input data to the appropriate output. This behaviour is most easily seen in the limit that  $n$  becomes large as shown in the inset of Fig. 2b.

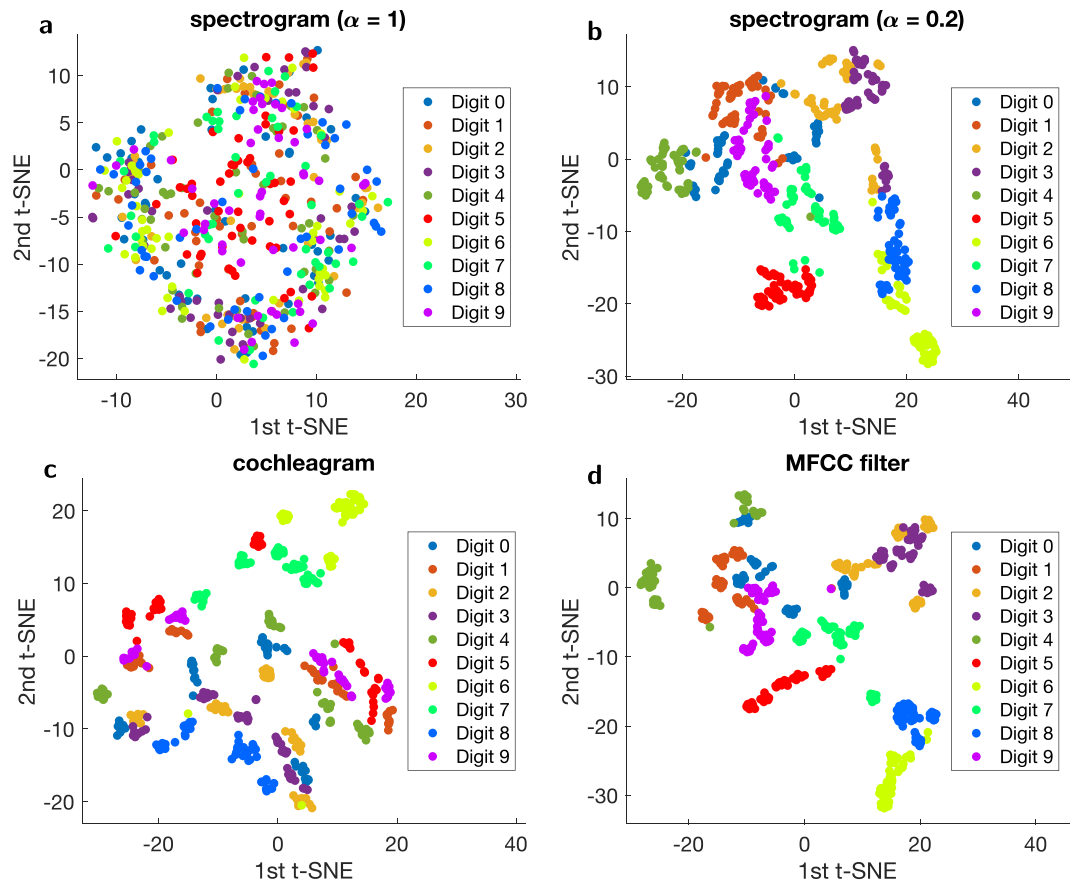
From Eq. (4) we can evaluate the effect of our non-linear transformation for  $n \rightarrow \infty$ :  $\lim_{n \rightarrow \infty} |X_{fr,\sigma}|^n = 0$  for  $|X_{fr,\sigma}| < 1$  and  $\forall n |X_{fr,\sigma}|^n = 1$  when  $|X_{fr,\sigma}| = 1$ . In practice, due to the numerical truncation on a computer, for  $n \gg 100$  and  $|X_{fr,\sigma}| < 1$ ,  $|X_{fr,\sigma}|^n = 0$ . So, for very large  $n$ , the resulting vector  $\mathbf{R}_\sigma$  contains only zeros and at least one element that is equal to 1 or  $-1$  after the non-linear transformation.

There are 500 digits in our spoken digit database and for very large odd values of  $n$ , there are 253  $\mathbf{R}_\sigma$  vectors with one  $R_{fr,\sigma} = 1$  and 247 with one  $R_{fr,\sigma} = -1$ . For each of these vectors, all other elements are mapped to zero. For large even values of  $n$ , all the 500  $\mathbf{R}_\sigma$  contain one  $R_{fr,\sigma} = 1$  with all others equal to zero. In this large-exponent limit, the classification task is simple to understand. The non-linearity selects the largest magnitude frequency/time component from the transformed audio file. If this component is constant between speakers, the digit can be identified. For large values of  $n$ , there are 2 different success rate values depending on the parity of  $n$ . As shown in the inset of Fig. 2b, for large values of  $\alpha$  ( $\alpha > 1000$ ), the success rate behaviour tends to a square function alternating between very low values (12%) around odd values of  $\alpha$ , i.e.  $\alpha \in [2n + 0.5, 2n + 1.5]$ , and a slightly higher value (25.8%) around even values of  $\alpha$ , i.e. for  $\alpha \in [2n - 0.5, 2n + 0.5]$ , where  $n \in \mathbb{N}$  (for large values of  $\alpha$ , when  $\alpha = n + 0.5$ , the success rate is not defined). The difference arises because of the random phases that arise from the Fourier transform. For even  $n$ , the phases are irrelevant, but for odd  $n$ , sometimes the value gets mapped to one and sometimes to minus one, making it much more difficult to classify.

Overall, as shown in Fig. 2(b), for a wide range of values of  $\alpha$ , the non-linearity drastically improves the recognition rate. In particular, the recognition rate is very high for low exponents  $\alpha$ . An optimum non-linearity is reached for  $\alpha = 0.2$  providing the highest recognition rate of 88%, which is comparable to those obtained for the cochleagram.

We use a *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) technique<sup>27</sup> to represent our  $N_f$  channels data in a 2D plot (see Fig. 3) in order to visualise how the data separation occurs and understand the recognition capacity of the different filtering methods. *t*-SNE is a nonlinear dimensionality reduction technique used for embedding high-dimensional data into a low-dimensional space of two or three dimensions. During the data reduction, the probability of two vectors to be neighbours is conserved, allowing visualisation of the structure in the data. Each digit is represented by coloured dots for all data points of the utterances. For instance, Fig. 3a shows that for the spectrogram with  $\alpha = 1$  (linear) for which the recognition rate is about 10% (random choice), there is no data separation as all the coloured points seem to be randomly distributed. In particular the digits of a same class do not form separated clusters. On the other hand, for the spectrogram with  $\alpha = 0.2$  (optimal non-linearity), data separation can clearly be seen in Fig. 3b correlating with the better recognition rate of 88% compared to the linear spectrogram ( $\alpha = 1$ ). Furthermore, *t*-SNE shows data separation capability for both the cochleagram and the MFCC filter. As shown in Fig. 3c,d, well defined clusters corresponding to the spoken digits appear and corroborate the high recognition rates exhibited by the two filtering methods.

To summarise this section, we show that the non-linear transformation applied to the input data by the MFCC filter and the cochleagram plays a similar role as the non-linear nodes in the reservoir neural network prior to the linear classifier. We highlight that these stand-alone feature extractors perform data separation due to their internal non-linear transformations. We indeed obtain recognition performance that are close to what is found with these approaches by adding a simple non-linear transformation to the individual elements of the conventional spectrogram. Depending on the non-linearity, the recognition rate can strongly vary from around 10% to 95.8%.



**Figure 3.** 2D representation of the two  $t$ -SNE components for: (a) the spectrogram with  $\alpha = 1$ , (b) the spectrogram with  $\alpha = 0.2$ , (c) the cochleagram, and (d) the MFCC filtering methods.

**Neural network: Reservoir computing based on non-linear oscillators.** Having shown that non-linear filtering methods can by themselves achieve high recognition rates, we turn to evaluating the gain in overall performance provided by a reservoir neural network taking as inputs the output of these acoustic filters. We implement the reservoir with a single non-linear oscillator<sup>6</sup>. In this approach, recurrent chains of non-linear transformations occur in time instead of space. The loss of parallelism is compensated by time-multiplexing, in turn requiring that the input be preprocessed. To do that, each point of interval  $\tau$  in Fig. 1(b) is multiplied by a random binary matrix (of dimensions  $N_f \times N_\theta$ ) to induce transient behaviour. This transformation is linear and does not affect the final recognition rate. Each point of the input audio file is converted in a binary sequence of duration  $\tau$  composed of  $N_\theta = 400$  points separated by time steps  $\theta$ . The time step  $\theta$  is set shorter than the relaxation time of the oscillator to keep the oscillator in the transient regime and generate temporal cascades at each sequence  $\tau$  of the pre-processed input.

We have developed a simple model based on a non-linear magnetic oscillator<sup>28</sup> taking into account the main ingredients for neuromorphic computing: non-linearity (square root dependence of the amplitude on the input current) and memory (relaxation time of the oscillator between two different output voltage levels). The dynamics of the evolution of the oscillator output microwave voltage  $v_i^{\text{osc}}$  as a function of the input voltage  $v_i^{\text{in}}$  at time step  $i$  can be solved numerically<sup>16</sup>:

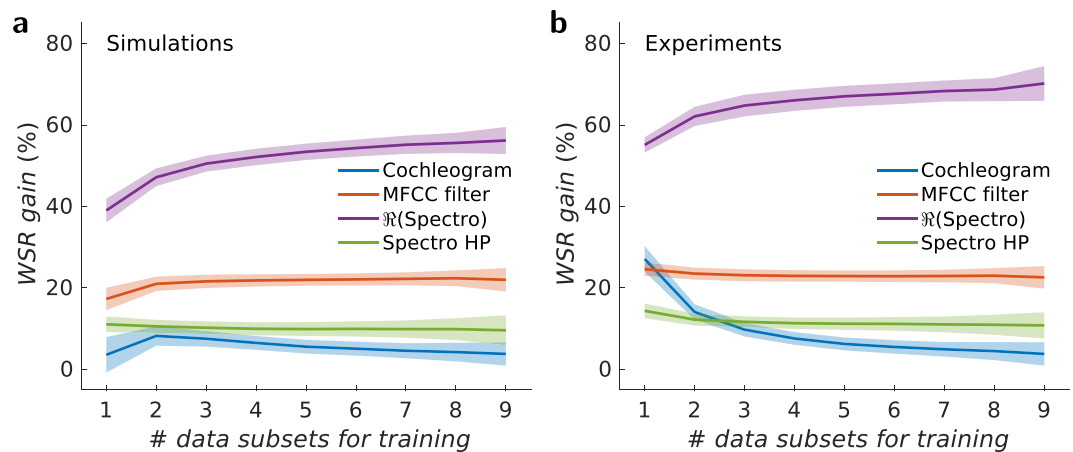
$$v_i^{\text{osc}} = v_i^\infty (1 - e^{-\Delta t/T_{\text{relax}}}) + v_{i-1}^{\text{osc}} \cdot e^{-\Delta t/T_{\text{relax}}}, \quad (5)$$

where  $T_{\text{relax}}$  is the relaxation time towards the asymptotic value  $v_i^\infty$  given by<sup>29</sup>:

$$v_i^\infty = c \sqrt{I_{\text{DC}} - v_i^{\text{in}}/R - I_c}, \quad (6)$$

with  $c$ , a constant related to the initial bias condition, i.e. the initial emitted voltage of oscillator,  $R$  the DC resistance of the oscillator and  $I_c$  the threshold current above which auto-oscillations can occur. In order to simulate the oscillator response to a time varying input ( $V_i^{\text{in}}$ ), we solve Eq. (5) numerically with the following parameters:  $\Delta t = 5$  ns,  $V_i^{\text{in}}/R = \pm 3$  mA,  $I_{\text{DC}} = 6$  mA,  $I_c = 4.9$  mA,  $T_{\text{relax}} = 410$  ns. These parameters, which constitute hyperparameters of our system, are extracted from experiments as reported elsewhere<sup>8</sup>.

Even if the recognition rate by the non-linear filters (MFCC and cochleagram) is already high, there is still room for improvement with the inclusion of a recurrent neural network. The increase in the recognition rate



**Figure 4.** Spoken digit recognition for a neural network. **(a)** Spoken digit gain on cross-validated test recognition rates as a function of the number of subsets  $N$  used during training for a non-linear oscillator modelled with Eqs. (2 and 3) and **(b)** experimental spin torque nano-oscillator driven by spin polarised current. The coloured region corresponds to the uncertainty of the recognition rate, here twice the standard deviation.

induced by the emulated non-linear oscillator is shown in Fig. 4a. We determine the increase in recognition rate due to the neural network by subtracting from the total recognition rate the contribution from acoustic filters previously calculated in Fig. 2(a) and normalising the result with the total recognition rate. The gain provided by the non-linear oscillator is low for the non-linear filters (training over 9 data subsets): 3.8% for the cochlear method, 9.6% for the Spectro HP method, and 22% for MFCC method. The increase is small because the total recognition rate (filter + network) is close to a perfect success rate: up to 99.6%, 98.6%, and 99.2% for the cochleogram, Spectro HP, and the MFCC filter respectively. On the other hand, the neural network drastically improves the recognition gain up to 55.6% (for 9 data subsets trained) but the final recognition rate (filter + neural network) is not as good, around 65.2%. The simulations have been obtained for a specific neural network based on the non-linear dynamics of an oscillator with time multiplexing in the framework of reservoir computing. As mentioned earlier, we choose this particular framework because it is frequently used for hardware implementations. However, these conclusions hold for very general types of (spatial or temporal) neural networks and learning processes because the limitation of the gain in recognition in the case of MFCC, Spectro HP, and cochlear filters is not due to the neural network but to the already excellent separation properties of the filtering.

We compare these simulations to the behaviour of an experimental non-linear oscillator. In particular, we choose a magnetic nano-oscillator that was recently demonstrated to be an excellent building block for neuromorphic computing<sup>8,16,30</sup>. This kind of oscillator is small (nanoscale), performs low power computing, has a high signal to noise ratio for high reliable computation, and allows a tunable non-linearity through the spin transfer torque mechanism. Our nanoscale oscillators are circular magnetic tunnel junctions, with a 6 nm thick FeB free layer and a diameter of 375 nm. The magnetisation in the FeB layer has a vortex structure as its ground state for these dimensions. In a small region called the vortex core, the elsewhere in-plane-curling magnetisation points out of the plane. Under dc current injection, the core of the vortex steadily gyrates around the centre of the dot with a frequency in the range 250 MHz to 400 MHz. Vortex dynamics driven by spin-torque are well-understood, well-controlled and have been shown to be particularly stable (more details can be found elsewhere<sup>31</sup>).

The experimental implementation of the spoken digit recognition task is described in ref.<sup>8</sup>. The preprocessed input signal (filtered digits with time multiplexing) is generated and sent to the sample using an arbitrary waveform generator. Then, the microwave voltage across the magnetic tunnel junction is measured by a real time oscilloscope and fast oscillations are observed. The amplitude of oscillator response is obtained by inserting a microwave diode between the sample and the oscilloscope and is processed as the output signal. The oscillation amplitude is robust to noise thanks to the confinement provided by the counteracting torques exerted by the injected current and the magnetic damping. In addition, the voltage amplitude is highly non-linear as a function of the injected current. The current depends on the voltage amplitude similarly to our simulated oscillator (square root dependence) in Eq. (3). Furthermore, the amplitude of the oscillator voltage intrinsically depends on past inputs when the time step  $\theta$  is shorter than the relaxation time of the magnetic nano-oscillator. Therefore, this single nano-device has the two most crucial properties of neurons: non-linearity and memory.

Tables 1 and 2 show the word success rates, as well as mean squared error obtained by simulations and experimentally. The gain on the spoken digit recognition for the different acoustic filters induced by the experimental magnetic nano-oscillator is shown in Fig. 4(b). There is very good agreement between the experimental results and the simulations. When 9 data sets are used while doing the training process the gain is 3.8% for the cochleogram, 10.8% for the Spectro HP filter, 22% for the MFCC filter, and 70.4% for linear spectrogram. We see in Tables 1 and 2, as well as Fig. 4 that, for some cases, the magnetic nano-oscillator exhibits slightly higher recognition gain than the simulations even though the latter neglects the intrinsic noise. We believe the better performance is mainly due to the higher complexity in the dynamics of the magnetic nano-oscillators, including a relaxation time that varies with current. Finally, the observation shows that overfitting effects are quite minimal.

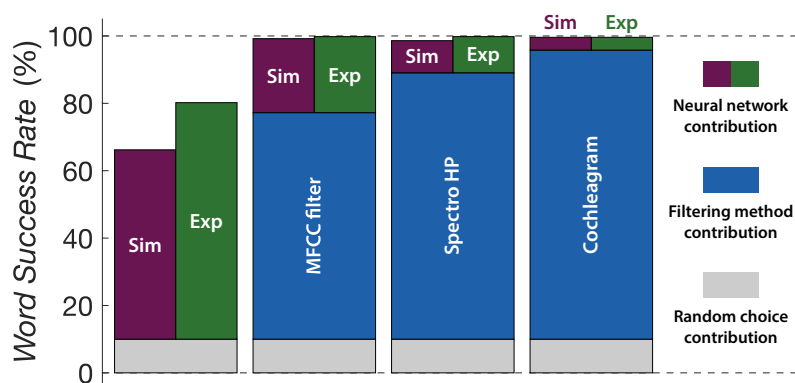


Filter	Training		Testing		Overfitting
	WSR (std)	MSE (std)	WSR (std)	MSE (std)	$MSE^{\text{test}}/MSE^{\text{train}}$
Cochleagram	100.0% (0.0%)	0.0133 ( $1.1 \cdot 10^{-4}$ )	99.6% (0.8%)	0.0193 ( $2.3 \cdot 10^{-3}$ )	1.451
MFCC	99.8% (0.1%)	0.0259 ( $1.1 \cdot 10^{-4}$ )	99.2% (1.0%)	0.0283 ( $8.2 \cdot 10^{-4}$ )	1.093
$\Re$ (Spectro)	75.7% (0.6%)	0.0708 ( $1.1 \cdot 10^{-4}$ )	65.2% (4.2%)	0.0738 ( $7.3 \cdot 10^{-4}$ )	1.042
Spectro HP	99.2% (0.2%)	0.0362 ( $9.9 \cdot 10^{-5}$ )	98.6% (1.9%)	0.0384 ( $6.5 \cdot 10^{-4}$ )	1.061

**Table 1.** Results for a simulated STNO neural network with  $N = 400$  nodes.

Filter	Training		Testing		Overfitting
	WSR (std)	MSE (std)	WSR (std)	MSE (std)	$MSE^{\text{test}}/MSE^{\text{train}}$
Cochleagram	100.0% (0.0%)	0.0192 ( $1.1 \cdot 10^{-4}$ )	99.6% (0.8%)	0.0222 ( $1.3 \cdot 10^{-3}$ )	1.156
MFCC	99.8% (0.1%)	0.0262 ( $1.4 \cdot 10^{-4}$ )	99.8% (0.6%)	0.0274 ( $1.0 \cdot 10^{-3}$ )	1.046
$\Re$ (Spectro)	90.2% (0.4%)	0.0699 ( $1.5 \cdot 10^{-4}$ )	80.0% (4.1%)	0.0726 ( $1.5 \cdot 10^{-3}$ )	1.039
Spectro HP	100.0% (0.0%)	0.0327 ( $7.1 \cdot 10^{-5}$ )	99.8% (0.6%)	0.0344 ( $7.7 \cdot 10^{-4}$ )	1.052

**Table 2.** Results for an experimental STNO neural network with  $N = 400$  nodes.



**Figure 5.** Contributions to the spoken digit cross-validated test recognition rate. Random choice level is shown in grey, the filtering methods in blue, and the neural network under the reservoir computing approach in purple and green for the simulations and experiments, respectively. Here, 9 data subsets (90% of the database) are used for training our reservoir computing model and the remaining subset (10% of the database) is used to perform the recognition task.

Some overfitting with respect to the mean squared error (MSE) can be seen when using cochleagram filters. This is due to the fact that in this situation, the role of the reservoir is quite minimal, whereas it uses the same number of parameters as in the other situations. However, the overfitting occurs in the MSE does not make the overall performance for the word success rate (WSR) on the test set significantly worse.

The different contributions to the spoken digit recognition task are summarised in Fig. 5 for the case in which nine utterances are used during the learning step and one during the recognition. The random choice level is 10% and is shown in grey. The contribution of the filtering methods is shown in blue (not visible in the case of the linear spectrogram,  $\alpha = 1$ ). Figure 5 also shows the net contribution of a neural network, in our case under the reservoir computing approach, to the spoken digit task. The simulated version of our neural network, i.e. using the simulated dynamics of the spin-torque vortex oscillators is shown in purple, while the results for the experimental magnetic nano-oscillators are shown in green. The main contribution to the spoken recognition task brought by the neural network happens when there is a lot of work to perform, i.e. when starting from the random choice level (linear spectrogram). Nevertheless, when our neural network is coupled with well performing stand-alone feature extraction techniques like the cochleagram or the MFCC filter, it is capable of bringing the recognition rate level to state-of-the-art values (overall WSR of 99.8% for the MFCC and the Spectro HP filters + experimental spin-torque vortex oscillator).

**More challenging spoken digit database.** The TI-46 database is based on clean audio waveforms from a limited number of speakers. We describe above how this database is rather limited for testing a neural network when combined with substantive preprocessing. We are able to test our nano-oscillator based reservoir computing approach by limiting the preprocessing to a basic linear preprocessing filter (linear spectrogram) and demonstrate its effectiveness. However, we can test the combination of effective preprocessing with reservoir computing by using a larger data set with a broader set of speakers and a variety of background noise types and levels. In this

SNR (dB)	Subway	Babble	Car	Exhibition	AVG
clean	95.15 (+56.34)	94.96 (+54.65)	90.98 (+45.88)	90.73 (+45.94)	92.96 (+50.70)
20	83.58 (+49.62)	90.70 (+57.75)	85.88 (+50.59)	74.90 (+43.63)	83.77 (+50.40)
15	79.48 (+47.39)	85.66 (+54.65)	83.92 (+51.76)	68.73 (+44.79)	79.45 (+49.65)
10	64.93 (+39.18)	79.46 (+55.82)	72.16 (+46.28)	57.92 (+36.30)	68.62 (+44.40)
AVG	80.79 (+48.13)	87.70 (+55.72)	83.24 (+48.63)	73.07 (+42.67)	81.20 (+48.79)

**Table 3.** Word success rate (in percent) for a simulated STNO neural network with  $N = 2000$  nodes after filtering the inputs with the MFCC filter combined with the reservoir. These results are for a reservoir trained with all noise levels and types and then tested for each part of the test set with the different noise levels and types. Each entry gives the overall word recognition rate and in parentheses, the gain achieved by adding the reservoir to the preprocessing filter. The first column gives the noise level for different parts of the test set and the top row gives the noise type. The bottom row gives the results over all levels of noise for each noise type and the right column gives the results for each noise level over all types of noise. The bottom right entry is the average result over the whole test set.

SNR (dB)	Subway	Babble	Car	Exhibition	AVG
clean	88.81 (+23.88)	87.21 (+24.81)	89.02 (+29.02)	91.51 (+25.87)	89.14 (+25.90)
20	67.54 (+23.51)	69.77 (+20.16)	69.02 (+26.67)	56.76 (+22.01)	65.77 (+23.09)
15	70.15 (+27.61)	63.57 (+16.96)	68.63 (+23.14)	59.46 (+28.57)	65.45 (+24.07)
10	55.97 (+18.66)	53.49 (+14.73)	56.86 (+21.17)	53.28 (+21.62)	54.90 (+19.05)
AVG	70.62 (+23.42)	68.51 (+19.17)	70.88 (+25.00)	65.25 (+24.52)	68.82 (+23.02)

**Table 4.** Word success rate (in percent) for a simulated STNO neural network with  $N = 2000$  nodes after filtering the inputs with the cochlear filter combined with the reservoir. These results are for a reservoir trained with all noise levels and types and then tested for each part of the test set with the different noise levels and types. Each entry gives the overall word recognition rate and in parentheses, the gain achieved by adding the reservoir to the preprocessing filter. The first column gives the noise level for different parts of the test set and the top row gives the noise type. The bottom row gives the results over all levels of noise for each noise type and the right column gives the results for each noise level over all types of noise. The bottom right entry is the average result over the whole test set.

section, we simulate the performance of our theoretical reservoir computing implementation (simulated STNO neural network) on the AURORA-2 database<sup>32,33</sup>.

The AURORA-2 database provides the data for the task of recognising digits taken from the TIDigits database<sup>34</sup> in noise and channel distorted environments (artificially corrupted). To simulate noisy telephony environments, the clean utterances are first down-sampled to 8 kHz, and then additive and convolutional noise is added. The AURORA-2 database has both clean and multi-condition training and test sets. Each type of noise is added into a subset of clean speech utterances, with seven different levels of signal-to-noise ratios (SNRs). This process generates seven subgroups of test sets for a specified noise type, with clean (infinite signal-to-noise ratios) and signal-to-noise ratios of 20, 15, 10, 5, 0, and  $-5$  dB.

We simulate the preprocessing and reservoir computing on all available isolated spoken digits from 0 to 9 in the training and testA datasets from 214 female and male speakers. The training dataset contains 2196 clean digits and several subsets of noisy (corrupted) digits with 4 different noise types (subway, babble, car, and exhibition hall noise) at different signal-to-noise ratios. During the training of our model, we select the 2196 clean digits and the corrupted digits with SNR = 20 dB (451 digits), SNR = 15 dB (444 digits), and SNR = 10 dB (430 digits). With this dataset, we train the model in mixed conditions.

The recognition (testing) is performed on testA dataset containing 4 types of added noise at different signal-to-noise ratios. The different noise types are the same as in the training set (unlike for the testB and testC subsets of AURORA-2 that we do not use). The testA dataset contains a subset of 1040 clean digits and several subsets containing each 1040 digits corrupted with subway, babble, car, and exhibition hall noise. We choose the following 4 subsets: clean and with corrupted with SNR = 20 dB, 15 dB, and 10 dB. To summarise, training is performed on 3521 digits and testing on 1040 unseen digits from the same categories as the training set. The test set contains 22.8% of the total number of digits (1040/4561).

Tables 3 and 4 give the simulated results for spoken digit recognition using the nano-oscillator based reservoir computing approach combined with the two filtering methods, MFCC, and cochlear, respectively. The results are given in word success rate (%). In parenthesis, we give the gain compared to the baseline (control test without the nano-oscillator based reservoir). Not surprisingly, the results are not as good as the results presented above for the TI-46 database. While the preprocessing filters do much worse without the reservoir than they do on the TI-46 database, they still do much better than linear preprocessing. In all cases, inclusion of the reservoir substantially improves the success rate compared to the baseline.

Comparison of Tables 3 and 4 shows that the MFCC filter is more robust to noise than the cochlear filter and gives better results in most cases. Interestingly, it does worse in almost all cases without the reservoir, but appears to allow the reservoir to make much larger improvements in the success rate. The average word success rate of the

testing set containing only clean digits is 92.96% for the MFCC filter. The training was performed on clean and corrupted digits (mixed conditions). The improvement over the baseline is 50.70% (given in parentheses) implying that the baseline value is 42.26% (=92.96% – 50.70%). When the cochlear filter is used, the average gain brought by the neural network is +25.90% when testing clean digits and is about two times smaller than for the MFCC filter. The same holds when noisy conditions are tested. The overall gain is +48.79% (+23.02%) for an overall average recognition rate of 81.20% (68.82%) when the input is preprocessed with the MFCC filter (cochlear filter). The baseline is lower for the MFCC filter than for the cochlear filter but as the gain is much larger (about 2 times), the absolute performance of our neural network is larger in noisy conditions for the MFCC filter. We suspect that similar results would hold for the class of MFCC-like filters, some of which are even more robust against the inclusion of noise.

There are many differences between the simulations performed on the TI-46 and AURORA-2 databases. For the TI-46 database, there are only 5 female speakers uttering each digit 10 times. Training is performed on some utterances and recognition is performed on the others and the success rate is the average success rate over all combinations. For the AURORA-2 database, there are 214 speakers, half of them are female and half are male, and they typically utter each digit twice. In contrast to TI-46, the speakers in the training set are different from the speakers in the testing set. Even without added noise, the test is much more difficult and involves almost 9 times more digits than for TI-46 (4561 digits in AURORA-2 vs 500 digits in TI-46).

Our results without the reservoir are consistent with previous results. Cochlear filtering achieves approximately a 60% word success rate (around 40% for MFCC) by itself on clean isolated digits of AURORA-2 database<sup>25</sup>. We obtain 63.24% (89.14–25.90%, see the last column of the first line of Table 4) for the cochlear filter and 42.26% (92.96–50.70%, see the last column of the first line of Table 3) for the MFCC filter.

## Conclusion

We test different frequency filtering methods as stand alone feature extractors. Training a linear classifier on the  $R_{\sigma}$  vectors for the classic TI-46 spoken digit data base, both the cochleagram and the MFCC filter give high identification rates without further processing. On the other hand, the real part of a linear spectrogram does not separate the inputs of different digit classes. Non-linearly transforming the spectrogram, gives similar results to the cochleagram and MFCC filters, stressing that the separation found for the MFCC and cochlear classifiers is due to the presence of non-linearity, with a minor effect due to the particular type of non-linear transformation.

In a second part, a non-linear oscillator is added to process the filtered input. The gain in word recognition due to the non-linear oscillator is computed for each filtering method. The non-linear oscillator is simulated and found to be in excellent agreement with experimental results with magnetic nano-oscillators. For the non-linear methods MFCC, Spectro HP, and cochleagram, the gain of word recognition is small, despite a nearly perfect word recognition. On the other hand, for the linear spectrogram, the gain of word recognition is much higher even if the final word recognition is maximum 80%.

An important lesson is that when evaluating hardware systems with speech recognition tasks, the final word recognition rate should be interpreted with caution. If a very efficient filtering is used to preprocess the input, the hardware system may not be adding much performance. A hardware system only adds something if it provides improved word recognition. It should be noted that the use of more complicated datasets, such as the proprietary spoken digits dataset used in<sup>21,25</sup>, or the inclusion of babble noise in the dataset would lead to significantly different results. The takeaway of our work is that, in order to test and compare hardware systems, using a linear spectrogram eases the interpretation of the results, because it does not introduce any separation of the input prior the hardware system. Furthermore, we show that a simple but powerful transformation like our Spectro HP filter starting from a simple spectrogram achieves state-of-the-art results (simulations and experimentally) without applying any specific acoustic filter that mimics the human auditory system (like the cochleagram or the MFCC filter).

Testing on the AURORA-2 dataset reveals that under noisy conditions the cochlear filter performs better by itself than the MFCC filter but the gain brought by the neural network is two times better on average for the MFCC filter and the overall word success rate is higher for the MFCC filter than for the cochlear filter.

## Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 14 May 2019; Accepted: 19 December 2019;

Published online: 15 January 2020

## References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436 (2015).
2. Mead, C. Neuromorphic electronic systems. *Proc. IEEE* **78**, 1629–1636, <https://doi.org/10.1109/5.58356> (1990).
3. Mead, C. & Ismail, M. (eds) *Analog VLSI implementation of neural systems*, vol. 80 (Springer Science & Business Media, 2012).
4. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560, <https://doi.org/10.1162/089976602760407955> (2002).
5. Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80, <https://doi.org/10.1126/science.1091277> (2004).
6. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
7. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Sci. Reports* **2**, 287 (2012).
8. Torrejon, J. *et al.* Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428 (2017).
9. Davis, S. & Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoust. Speech, Signal Process.* **28**, 357–366, <https://doi.org/10.1109/TASSP.1980.1163420> (1980).
10. Verstraeten, D., Schrauwen, B., Stroobandt, D. & Van Campenhout, J. Isolated word recognition with the liquid state machine: a case study. *Inf. Process. Lett.* **95**, 521–528, <https://doi.org/10.1016/j.ipl.2005.05.019>, Applications of Spiking Neural Networks (2005).
11. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **29**, 82–97 (2012).

12. Lyon, R. A computational model of filtering, detection, and compression in the cochlea. In *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, 1282–1285, <https://doi.org/10.1109/ICASSP.1982.1171644> (1982).
13. Slaney, M. *Lyon's Cochlear Model*. Apple Computer, Inc., Cupertino, CA, apple computer technical report#13 edn. (1988).
14. Brunner, D., Soriano, M. C., Mirasso, C. R. & Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
15. Vandoorne, K. *et al.* Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, 3541 (2014).
16. Riou, M. *et al.* Neuromorphic computing through time-multiplexing with a spin-torque nano-oscillator. *Electron Devices Meet. (IEDM), 2017 IEEE Int.* 36.3.1–36.3.4, <https://doi.org/10.1109/IEDM.2017.8268505> (2017).
17. Larger, L. *et al.* Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**, 3241–3249, <https://doi.org/10.1364/OE.20.003241> (2012).
18. Dejonckheere, A. *et al.* All-optical reservoir computer based on saturation of absorption. *Opt. Express* **22**, 10868–10881, <https://doi.org/10.1364/OE.22.010868> (2014).
19. Vinckier, Q. *et al.* High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica* **2**, 438–446, <https://doi.org/10.1364/OPTICA.2.000438> (2015).
20. Brunner, D. *et al.* Tutorial: Photonic neural networks in delay systems. *J. Appl. Phys.* **124**, 152004, <https://doi.org/10.1063/1.5042342> (2018).
21. Penkovsky, B., Larger, L. & Brunner, D. Efficient design of hardware-enabled reservoir computing in fpgas. *J. Appl. Phys.* **124**, 162101, <https://doi.org/10.1063/1.5039826> (2018).
22. Dupont, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783–22795, <https://doi.org/10.1364/OE.20.022783> (2012).
23. Martinenghi, R., Rybalko, S., Jacquot, M., Chembo, Y. K. & Larger, L. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **108**, 244101, <https://doi.org/10.1103/PhysRevLett.108.244101> (2012).
24. Romain Martinenghi, M. J. Y. K. C. L. L., Antonio Baylon Fuentes. Towards optoelectronic architectures for integrated neuromorphic computers. *Proc. SPIE OPTO* 8989, <https://doi.org/10.1117/12.2038347> (2014).
25. Larger, L. *et al.* High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Phys. Rev. X* **7**, 011015, <https://doi.org/10.1103/PhysRevX.7.011015> (2017).
26. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149, <https://doi.org/10.1016/j.cosrev.2009.03.005> (2009).
27. van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *J. machine learning research* **9**, 2579–2605 (2008).
28. Slavina, A. N. & Tiberkevich, V. S. Nonlinear auto-oscillator theory of microwave generation by spin-polarized current. *IEEE Transactions on Magn.* **45**, 1875–1918, <https://doi.org/10.1109/TMAG.2008.2009935> (2009).
29. Grimaldi, E. *et al.* Response to noise of a vortex based spin transfer nano-oscillator. *Phys. Rev. B* **89**, 104404, <https://doi.org/10.1103/PhysRevB.89.104404> (2014).
30. Romera, M. *et al.* Vowel recognition with four coupled spin-torque nano-oscillators. *Nature* **563**, 230–234, <https://doi.org/10.1038/s41586-018-0632-y> (2018).
31. Tsunegi, S. *et al.* High emission power and q factor in spin torque vortex oscillator consisting of feb free layer. *Appl. Phys. Express* **7**, 063009 (2014).
32. Hirsch, H. & Pearce, D. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings of ISCA ITRW ASR2000 on Automatic Speech recognition: Challenges for the next Millennium* (Paris, France, 2000).
33. *ELRA catalogue* (<http://catalog.elra.info>), *AURORA Project Database*, v2.0, ISLRN: 977-457-139-304-2, ELRA ID: AURORA/CD0002.
34. Leonard, R. & Doddington, G. *TIDigits*. (Linguistic Data Consortium, Philadelphia, 1993).

## Acknowledgements

F.A.A. is a Research Fellow of the F.R.S.-FNRS. This work was supported by the European Research Council ERC under Grant bioSPINspired 682955.

## Author contributions

The study was designed by F.A.A., J.G. and M.D.S., samples were optimised and fabricated by S.T. and K.Y., experiments were performed by M.R. and J.T., and numerical studies were realised by F.A.A. F.A.A. wrote the core of the manuscript and all authors contributed to the text as well as to the analysis of the results.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to F.A.A.

**Reprints and permissions information** is available at [www.nature.com/reprints](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 license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license 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 license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020