



- (51) **International Patent Classification:**
G06N 3/04 (2006.01) *G06N 3/063* (2006.01)
- (21) **International Application Number:**
PCT/EP2016/081239
- (22) **International Filing Date:**
15 December 2016 (15.12.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
EP15200681.3 17 December 2015 (17.12.2015) EP
- (71) **Applicant:** **VRIJE UNIVERSITEIT BRUSSEL**
[BE/BE]; Pleinlaan 2, 1050 Brussel (BE).
- (72) **Inventors:** **KEUNINCKX, Lars**; Meidoornstraat 32, 2860 Sint-Katelijne-Waver (BE). **VAN DER SANDE, Guy**; Schaapbroekhof 10, 2820 Bonheiden (BE). **DANCKAERT, Jan**; Felix Sterckxstraat 53, 1020 Brussel (BE).
- (74) **Agents:** **VAN BLADEL, Marc** et al.; Hundelgemsesteenweg 1116, 9820 Merelbeke (BE).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

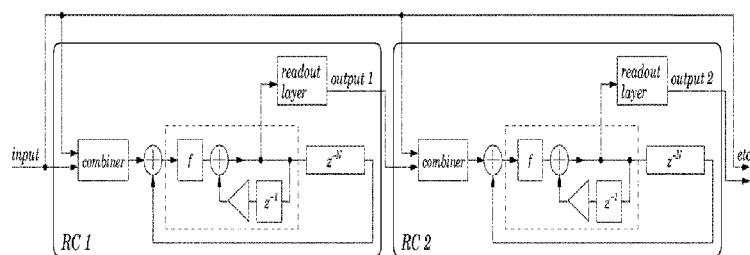
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

— with international search report (Art. 21(3))

(54) **Title:** RESERVOIR COMPUTING DEVICE**Fig. 4**

(57) **Abstract:** The present invention relates to a reservoir computing device comprising a cascade of at least two reservoir computing blocks, each reservoir computing block comprising - summing means for making a sum of a block input signal and a delayed signal output by the reservoir computing block, - a digital filter structure arranged for applying a non-linear function to said sum and for adding to a non-linear function output a filtered version of a preceding non-linear function output, yielding a digital filter structure output signal, - a delay line structure arranged for receiving as input the digital filter structure output signal and for outputting a delay line structure output signal to a readout layer, said delay line structure comprising a plurality of delay elements arranged for outputting the delayed signal being a delayed version of the digital filter structure output signal, whereby the block input signal is in the first reservoir computing block of the cascade a signal applied to the reservoir computing device and whereby in each subsequent reservoir computing block of the cascade the block input signal is, according to a given combining pattern, a delayed version of the signal applied to said reservoir computing device, the delay line structure output signal applied to said readout layer by the previous reservoir computing block in the cascade or a combination thereof.

Reservoir Computing Device

Field of the invention

[0001] The present invention is generally related to the field of machine learning and artificial neural networks. More specifically, it relates to a subclass of artificial neural networks known as echo state networks or reservoir computers.

Background of the invention

[0002] Reservoir computing is a paradigm recently introduced in machine learning, with the aim of simplifying the training of artificial neural networks (ANN). Within this context, a "reservoir" is a network of many randomly connected neurons or nodes. Each node performs some type of non-linear, in most cases time dependent, operation on its input signal. The input signal of the nodes either comes from the output of other nodes, an externally applied input or both. The reservoir usually comprises several hundred of nodes.

[0003] In a traditional ANN the connections between the nodes, called weights, are then adapted in such a way that a given class of input signals results in a given wanted output, i.e. the nodes and their interconnections solve a task. Since this problem is non-linear, there typically is no fast algorithm to find weights that optimally solve a task.

[0004] In reservoir computing (RC) the randomly chosen weights between the nodes of the reservoir are not altered once the network is assembled. Instead, the RC output is taken as a linear combination of the outputs of the nodes in the reservoir. Calculating the weights of this readout layer is a linear problem. Therefore these weights can be calculated by performing a least squares fit to the requested output. Mathematically this boils down to performing a pseudo-inverse operation on the matrix holding all neuron states of the reservoir over time, which is a well-founded and understood procedure.

[0005] To classify the input in two classes the operation of a reservoir computer can be understood as follows. The non-linearity present in the nodes projects the input signal to a phase space of a higher dimension than the input signal dimensionality. Typically, the input vectors may not be separable by a hyperplane in the lower dimensional input space, but are separable in the higher dimensional space. In that higher dimension the hyperplane separating both classes is described by the readout layer weights. If the linear combination of the node states equals a value above this hyperplane, the input is deemed to belong to one class, and if below, to the other.

[0006] This is illustrated in Fig.1 for a two input exclusive-or (XOR) operation. In the two dimensional input space (left) a hyperplane is a straight line. It is impossible to draw a single straight

line separating the inputs into two classes: those that result in a 'zero' output (circles) and those that lead to a 'one' output (hexagons). If the inputs are transformed into a higher dimensional space, for instance three-dimensional, a hyperplane dividing the inputs into two classes, hence, a 2D plane is readily found.

5 [0007] For a system identification task, the reservoir computer is trained such that it mimics a given system. The internal operation or composition of the system to be identified is typically unknown. This is called a 'black box' approach. In contrast, in a 'white box' approach the internal structure of the system to be identified is known, i.e. the system can be divided in well-known or (mathematically) well described blocks.

10 [0008] First a training input is applied to the system and the output is recorded. This forms an input/output training pair. The training input is then applied to the reservoir and the node states over time are recorded in a state matrix. The weights of the readout layer are calculated, using a pseudo-inverse of the state matrix, such that a linearly weighted combination of the node states results in an output as closely matched to the training output as possible. After this training phase,
15 the reservoir computer performs as the system to be identified, generalizing its response to previously unseen input signals.

[0009] The main obstacle for the practical and economically sound implementation of reservoir computers (and of any ANN of some size), is that each of the many nodes has to be built and connected to the other nodes. Especially for a photonic implementation this becomes problematic, as
20 combining light into a single non-linear node comprises splicing optical fibres.

[0010] It has been shown in the art that a reservoir comprising many non-linear nodes connected with random weights can be replaced by a single non-linear node with delayed feedback. Since delay in photonics is nothing more than a length of optical fibre, this approach radically simplifies the construction of a photonic reservoir computer.

25 [0011] In Fig.2 a diagram of a delay line based reservoir computer (DLRC) is shown. The input $x_{in}(t)$ is sampled every τ , which is also the duration of the delay in the feedback loop. Then the input is multiplied with a masking signal $m(t)$. This mask is repetitive with a period equal to τ and outputs in each period a sequence of N values randomly chosen from a certain set $\{m_1, \dots, m_L\}$ and spaced apart τ/N in time. This mask comprises the random connectivity between the neurons in the reservoir. The
30 simplest mask possibly consists of values drawn from a binary set $\{-1, +1\}$. The masked signal $J(t) = x_{in}(n\tau)m(t)$ is then applied to a non-linear time dependent node, which also receives input from the delayed feedback or echoes. The masked input is scaled by a factor γ and the feedback by a factor η to bias the non-linear node in the optimal dynamical regime. Both these factors depend on the task at hand as well as the dynamical behaviour of the non-linear node. Generally speaking the system must

be biased to the edge of stability (feedback η close to but below unity). In the absence of input, the "echoes" of the previous inputs then slowly fade while circulating and the system goes to the zero state. Thus the system shows fading memory, i.e. its dynamical behaviour is somewhat influenced by the past. Finding the optimal settings for these parameters is a non-linear problem which can be tackled by, for example, a gradient descent or scanning of the parameter space.

[0012] The output $y_{out}(n)$ is calculated as a linear combination of taps on the delay line:

$$y_{out}(n) = \sum_{i=1}^N w_i x[(n - i/N)\tau] \quad (1)$$

These taps are called 'virtual nodes' or 'virtual neurons'. The time interval between the virtual nodes in the delay line is $\Theta = \tau/N$.

[0013] The training of a single delay line reservoir computer with a single input, as found in the literature, is now explained. The goal of the training of reservoir computer is to determine the weights of the readout layer in such a way that a given input signal, or representative of a class of input signals, results in a desired output signal. This is done as follows. Let $x_1 \dots x_n$ be the training input samples and $y_1 \dots y_n$ be the output training samples. Each input sample x_i is masked (multiplied) with the fixed mask (suppose $m_1 \dots m_N$ is the mask). The masked signal $J_{i,k} = x_i * m_k$ for $k=1 \dots N$ is summed with the previous output of the delay line and applied to the non-linear node, as described above. N is also the number of 'virtual neurons'. The response of the non-linear node is stored in a matrix called the state matrix S , having entries $S_{i,k}$. The response of the non-linear node is also shifted back in the delay line.

[0014] The state matrix S thus has n rows with row index $i=1 \dots n$ and N columns with column index $k=1 \dots N$. Each column of S can be interpreted as a signal $S_{i=1 \dots n, k}$ of its own, with exactly n samples, obtained by some complex non-linear transformation. The output signal of the reservoir computer device y'_i , with $i=1 \dots n$, is formed by the readout layer, i.e. a linear combination of these N signals: $y'_i = \sum_{k=1}^N S_{i,k} w_k$, where the w_k ($k=1 \dots N$) denote the fixed weights (real numbers) of the output layer. Mathematically this is a projection of the N -dimensional function space back to the 1-dimensional function space of the input signal or output signal. In matrix notation this is : $\mathbf{S} \cdot \mathbf{w} = \mathbf{y}'$, where $\mathbf{w}$ and $\mathbf{y}'$ are interpreted as column vectors. For illustration reference is made to Fig.2.

[0015] The goal of the training is now to find optimal weights w_k such that the distance between an ideal or desired output signal y and the real output signal y' , is minimized. The desired output signal is a signal originating from a measurement of the black box system to be identified. The solution to this optimization depends on the distance (mathematically, a norm) used. Often, the Euclidian or L_2 norm is used. The weights w_k are chosen to minimize the sum of the squared error :

$$E = \sum_{i=1}^N (y_i - y'_i)^2$$

Minimizing E leads to the well-known least-squares problem, which is a linear problem, i.e. solvable by methods of linear algebra. The solution involves the use of the Moore-Penrose pseudo-inverse $S^\dagger = (S^T S)^{-1} S^T$, where S^T is the transpose of the state matrix, and $(.)^{-1}$ the matrix inverse. The optimal weights, that minimize the squared error between the wanted output y and the real output y' is given by $w = S^\dagger y$. This is, however, only one option to optimize the weights. As found in the literature, other norms can be used, leading to other ways of solving the problem. It is also notable that the state matrix S is expanded with a constant column (all ones-column), such that a constant can be added to y' . In such a way a DC bias can be added in case this would be useful.

[0016] Since the (linear) optimization of the weights can be done in numerically efficient ways, this training step is much faster than the training step found in other types of artificial neural networks, such as backpropagation. This is the main attractiveness of reservoir computing.

[0017] The structure of Fig.2 is not only well suited for photonic implementations. An electronic real-time implementation can be considered using the non-linearity present in a Mackey-Glass Delay Differential Equation (DDE)

$$\dot{x} = -x + f[\eta x(t - \tau) + \gamma J(t)] \quad (2)$$

in which the function f is defined as

$$f(x) = \frac{Cx}{1 + (Bx)^p} \quad (3)$$

with B and C constants and the exponent $p > 1$.

[0018] The electronic implementation comprises an analog circuit for the non-linear node, interfaced with digital delay. The core computation and non-linear transformation of the input signal to the higher dimensional phase space is performed by the analog non-linear node. The machine can then be given a task like e.g. real-time prediction of a chaotic time trace, given the previous samples.

[0019] When it comes to performing system identification tasks, especially those that simultaneously require a long machine memory and a highly non-linear transformation of the input, there remain some issues to be solved. A known solution for such tasks is to cascade two or more reservoir computers. However, this requires intermediate training signals. These intermediate training signals are only available given a priori knowledge of the internals of the unknown system. This obviously excludes a black box approach, wherein only the input and measured output of the system to be identified are available as a training pair.

[0020] The Ph.D. thesis *“Reservoir Computing Based on Delay-Dynamical Systems”* (L.Appeltant, Vrije Universiteit Brussel and Universitat de les Illes Balears, May 2012) discusses reservoir computing with a delayed feedback system comprising a single non-linear node and a delay line. The input signal is sequentialized in order to be fed to the reservoir. After processing the information in the reservoir and reading out the node states, they are fed to a linear training algorithm. The used training procedures are identical to the ones employed in traditional reservoir computing setups.

[0021] The paper *“Online learning navigation behaviors for small mobile robots with reservoir computing architectures”* (Antonelo et al., IEEE Trans. Neural Networks and Learning Systems, v, June ol.26, no.4, June 2014, pp.763-780) shows a hierarchical reservoir computer consisting of two series reservoir computers in a robotics application. Each reservoir computer is trained for different tasks and they operate at different timescales from one another. Different training data sets are needed thereby.

[0022] In *“Feedback control by online learning an inverse model”* (T. Waegeman et al., IEEE Trans. Neural Networks and Learning Systems, vol.23, no.10, August 2012, pp.1637-1648) an application of reservoir computing in system control is shown. The system to be controlled is called the ‘plant’. Two identical reservoir computing units (named “Network A” and “Network B”) are used in a scheme that allows online control of the plant without previous knowledge. The output of one of these reservoirs is not input into another reservoir. Both Network A and B receive the current actual state of the plant. In addition, Network A receives the past actual state as input, while Network B receives the desired plant output as input. The output of Network B is used as a training goal (and not as input to) Network A, in a scheme to mimic the inverse operation of the plant.

[0023] In *“Reservoir Computing”* (F.Grèzes, Ph.D. Thesis, City University of New York) a cascade of reservoirs is discussed that earlier was proposed by Treifenbach et al. in *“Phoneme recognition with large hierarchical reservoirs”* (Advances in Neural Information Processing systems, Neural Information Processing System Foundation, 2010). It is mentioned that Treifenbach reports having “observed a significant improvement of the correction error rate by 3-4%” and also that “additional layers beyond the second only provided minimal gain.” Only the output of the previous reservoir computing block is input into the next one in the cascade (reservoir block 2 and onwards in Figure 3.1 of the paper) and not the original or global input, as is input into the first reservoir computing block.

[0024] The paper *“Minimal Approach to neuro-inspired information processing”* (Soriano et al, Frontiers in Computational Neuroscience, vol.9, 68, June 2015) mentions that in case the simple architecture of a single dynamical node with delay is not sufficiently powerful or fast enough to

process complex signals, it can be combined with a hierarchical time-multiplexing structure or a combination of spatial- and time-multiplexing. This hierarchical time-multiplexing is to be construed as time-multiplexing of the signals applied to a single reservoir computer, with the aim of solving the technical difficulties encountered in achieving a high number of nodes in optical reservoir computer implementations.

[0025] There is a need for a solution for cascading two or more reservoir computing blocks wherein the limitations of the prior art approach, in particular with respect to the limited performance gains obtained by cascading reservoirs for system identification tasks, are overcome or avoided.

Summary of the invention

[0026] It is an object of embodiments of the present invention to provide for reservoir computing device suitable for system identification tasks while still allowing a so called black box approach.

[0027] The above objective is accomplished by the solution according to the present invention.

[0028] In a first aspect the invention relates to a reservoir computing device comprising a cascade of at least two reservoir computing blocks, each reservoir computing block comprising

- summing means for making a sum of a block input signal and a delayed signal output by the considered reservoir computing block,

- a digital filter structure arranged for applying a non-linear function to said sum and for adding to a non-linear function output a filtered version of a preceding non-linear function output, yielding a digital filter structure output signal,

- a delay line structure arranged for receiving as input the digital filter structure output signal and for outputting a delay line structure output signal to a readout layer, said delay line structure comprising a plurality of delay elements arranged for outputting the delayed signal being a delayed version of the digital filter structure output signal,

whereby the block input signal is in the first reservoir computing block of the cascade a signal applied to the reservoir computing device and whereby in each subsequent reservoir computing block of the cascade the block input signal is a delayed version of said signal applied to said reservoir computing device, the delay line structure output signal applied to said readout layer by the previous reservoir computing block in said cascade or a combination of said delayed version and said delay line structure output signal, whereby said delayed version, said delay line structure output signal and/or said combination are applied according to a given pattern.

[0029] From the second reservoir computing block onwards one can vary the input signal according to a given, prescribed pattern. The pattern can be any sequence formed with the output signal of the preceding block and a suitably delayed version of the signal used at the input of the reservoir computing device. In the pattern also a signal can be used which is a combination of the output signal of the preceding block and the signal applied at the very input of the reservoir computing device.

[0030] The proposed solution indeed allows for training the reservoir computing device comprising a cascade of a number of computing blocks, e.g. to perform an emulation of a device under test. Key feature is the specific structure of the reservoir computing device. By providing at the input of each subsequent computing block not only an output signal coming from the preceding computing block but also a suitably delayed version of the signal applied to the reservoir computing device, each block has at its disposal the best approximation of the desired output signal available so far and the originally applied signal. A best approximation is then determined for the reservoir computing block under consideration. Block after block is so optimized. No intermediate signals are needed anymore in the training. Although the invention is aimed at black box system identification, it is to be noted that it can also function in a white box approach.

[0031] In a preferred embodiment one or more blocks of the reservoir computing device are arranged for scaling the signal applied to the reservoir computing device with a first scaling factor and for scaling the delay line structure output signal applied to the readout layer by the previous reservoir computing block with a second scaling factor. This allows accounting for a difference in magnitude between the two signals applied to the reservoir computing block.

[0032] Preferably the signal applied to the reservoir computing device is a product of an external data signal with a masking function.

[0033] In one embodiment the digital filter structure comprises a first switch to guide the non-linear function output along either a first or a second of two parallel paths, each of the parallel paths having an own scaling factor.

[0034] Advantageously, the digital filter structure comprises a first order filter yielding the filtered version of the preceding non-linear function output.

[0035] In another embodiment the digital filter structure comprises a second switch to guide the filtered version of the preceding non-linear function output along either a first or a second of two further parallel paths, each of the further parallel paths having an own scaling factor.

[0036] In one embodiment the reservoir computing device comprises a third switch to guide the delayed signal along either a first or a second of two parallel feedback paths, each of the parallel feedback paths having an own feedback scaling factor.

[0037] Advantageously, the delay line structure output signal outputted to the readout layer is the digital filter structure output signal.

[0038] In a preferred embodiment the plurality of delay elements is part of a further digital filter and an output of the further digital filter is used as the delayed signal output.

5

[0039] In another aspect the invention relates to a method for emulating a device under test arranged for receiving an input signal and yielding an output signal. The method comprises

- modelling the device under test as a reservoir computing device comprising a cascade of at least two reservoir computing blocks as previously described,
- 10 - measuring a plurality of input/output pairs of input signals and output signals applied to said device under test, said input/output pairs forming a training set,
- training with said training set the device under test modelled as said reservoir computing device,
- determining from said training internal parameters of at least one of the digital filter
15 structure, the delay line structure and the readout layer of at least one of the reservoir computing blocks of the reservoir computing device,
- emulating the device under test using said determined internal parameters.

[0040] In an embodiment the device under test is a real time audio processing system.

[0041] In a more specific embodiment the device under test is a guitar amplifier.

20 **[0042]** In another use scenario the device under test is a hearing aid comprising a microphone which receives babble noise.

[0043] In other use scenarios the above-described method is used for classifying engine vibration patterns, for classifying heart sounds or for detecting sleep apnea.

25 **[0044]** For purposes of summarizing the invention and the advantages achieved over the prior art, certain objects and advantages of the invention have been described herein above. Of course, it is to be understood that not necessarily all such objects or advantages may be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention may be embodied or carried out in a manner that achieves or
30 optimizes one advantage or group of advantages as taught herein without necessarily achieving other objects or advantages as may be taught or suggested herein.

[0045] The above and other aspects of the invention will be apparent from and elucidated with reference to the embodiment(s) described hereinafter.

Brief description of the drawings

[0046] The invention will now be described further, by way of example, with reference to the accompanying drawings, wherein like reference numerals refer to like elements in the various figures.

[0047] Fig.1 illustrates a transformation to a higher phase space.

5 [0048] Fig.2 illustrates the structure of a delay line based reservoir computer as known in the art.

[0049] Fig.3 illustrates a time-discretized version of the scheme of Fig.2.

[0050] Fig.4 illustrates an embodiment of the reservoir computing device according to the invention.

10 [0051] Fig.5 illustrates a further embodiment of the invention.

[0052] Fig.6 illustrates a reservoir computing device according to the invention with a number of reservoir computing blocks.

[0053] Fig.7 illustrates another embodiment of the reservoir computing device according to the invention.

15 [0054] Fig.8 illustrates a portion of a guitar signal and resulting distorted amplifier signal.

[0055] Fig.9 illustrates the correspondence between the distortion of the emulated system obtained by means of the reservoir computing device and the real distortion measured with the guitar amplifier.

20 [0056] Fig.10 illustrates a use case where the reservoir computing device of the invention is applied to suppress babble noise.

[0057] Fig.11 illustrates the performance of the proposed scheme with some other forms of cascading reservoir computing blocks.

Detailed description of illustrative embodiments

25 [0058] The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto but only by the claims.

[0059] Furthermore, the terms first, second and the like in the description and in the claims, are used for distinguishing between similar elements and not necessarily for describing a sequence, either temporally, spatially, in ranking or in any other manner. It is to be understood that the terms so
30 used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

[0060] It is to be noticed that the term “comprising”, used in the claims, should not be interpreted as being restricted to the means listed thereafter; it does not exclude other elements or

steps. It is thus to be interpreted as specifying the presence of the stated features, integers, steps or components as referred to, but does not preclude the presence or addition of one or more other features, integers, steps or components, or groups thereof. Thus, the scope of the expression “a device comprising means A and B” should not be limited to devices consisting only of components A and B. It means that with respect to the present invention, the only relevant components of the device are A and B.

[0061] Reference throughout this specification to “one embodiment” or “an embodiment” means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases “in one embodiment” or “in an embodiment” in various places throughout this specification are not necessarily all referring to the same embodiment, but may. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner, as would be apparent to one of ordinary skill in the art from this disclosure, in one or more embodiments.

[0062] Similarly it should be appreciated that in the description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims following the detailed description are hereby expressly incorporated into this detailed description, with each claim standing on its own as a separate embodiment of this invention.

[0063] Furthermore, while some embodiments described herein include some but not other features included in other embodiments, combinations of features of different embodiments are meant to be within the scope of the invention, and form different embodiments, as would be understood by those in the art. For example, in the following claims, any of the claimed embodiments can be used in any combination.

[0064] It should be noted that the use of particular terminology when describing certain features or aspects of the invention should not be taken to imply that the terminology is being re-defined herein to be restricted to include any specific characteristics of the features or aspects of the invention with which that terminology is associated.

[0065] In the description provided herein, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In

other instances, well-known methods, structures and techniques have not been shown in detail in order not to obscure an understanding of this description.

[0066] The present invention is developed starting from the known scheme of Fig.2, which is first discretized and next provided with some additional features to make the resulting solution suitable for system identification tasks (like emulating a device under test) while still allowing a so called black box approach.

[0067] In a first step the digital structure represented in Fig.3 is considered. This discrete time delay line reservoir computing (DLRC) device is obtained by a time discretization of the mathematical expression of Equation 2. The analog non-linear node, responsible for the mixing of node states, is replaced by a first order Infinite Impulse Response (IIR) filter, such that no analog-to-digital and digital-to-analog conversion is needed anymore within the reservoir computing device itself. The clock speed is N times the input sampling speed. This structure is described by the following delay difference equation

$$x(n+1) = ax(n) + (1-a)f[\gamma J(n) + \eta x(n-N)] \quad (4)$$

with f denoting the non-linear function, $J(n)$ the masked input and $x(n)$ the reservoir states from which the output of the reservoir computing device are constructed. It can be shown that choosing node mixing parameter a as $a = \exp(-\Theta)$, with still $\Theta = \tau/N$, ensures the same node interaction topology as for the continuous time reservoir computing device.

[0068] In order to allow performing system identification tasks which require a lot of computational power it is proposed in this invention to use a cascade of two or more reservoir computing blocks with a structure as in Fig.4, wherein the structure of Fig.3 is modified to allow for two inputs. In Fig.4 an embodiment of a reservoir computing device according to the invention is shown with two reservoir computing blocks. Obviously in the first block of the cascade it is sufficient to have only one input to receive the signal applied to the reservoir computing device. The second block receives as one input a delayed version of the signal that is input to the first reservoir computing block and, as a second input, a signal output by the delay line structure of the preceding block, i.e. the first block in this particular example. In an embodiment with more than two reservoir computing blocks, any subsequent block also receives an appropriately delayed version of the signal applied to the first block and the delay line structure signal output by its preceding reservoir computing block.

[0069] Fig.4 also shows the basic structure of the reservoir computing blocks. To the signal received at the input of the reservoir computing block is added a feedback signal. The resulting sum

signal is applied to a digital filter structure performing a non-linear operation f . The output signal of the non-linear operation is in a filter combined (e.g., summed) with the previous non-linear function output or even an earlier non-linear function output, delayed over the appropriate number of samples. The result forms the output signal of the digital filter structure, which is next fed to a delay line structure. This delay line structure comprises a number of delay elements for creating a delayed version of the digital filter structure output signal. The delay line structure outputs a delay line structure output signal to a readout layer of the reservoir computing device. The delay line structure output signal is via the readout layer also fed to the subsequent block of the reservoir computing cascade.

[0070] As already mentioned, from the second block of the cascade onwards the input signal being used in the combiner is either the delay line structure output signal of the preceding reservoir computing block, which is applied via the readout layer, or an appropriately delayed version of the signal that was applied to the reservoir computing device. Switching between the two possible inputs occurs in an alternating way according to a given, prescribed pattern. In certain embodiments also a combination of both is used as an input.

[0071] In a preferred embodiment the input signal applied to the proposed reservoir computing device is itself a product of an external data signal with a masking function. This was already illustrated in Fig.2. By applying such a masking function the diversity of the state matrix is maximized, thereby allowing a better separation.

[0072] In a practical implementation a scaling factor γ is applied to the input signal and in the feedback path a feedback scaling factor η is used. Further the non-linear function output signal is scaled with a scaling factor $1-a$ and the signal in the feedback loop of the digital filter structure is scaled with a factor a .

[0073] In a preferred embodiment the basic scheme of Fig.4 is further modified as shown in Fig.5. The two inputs there each have a separate programmable input scaling factor γ_1 and γ_2 , respectively. This offers the benefit that inputs of considerably different amplitudes can be used. In case the two inputs always have a same order of magnitude a single input scaling factor would suffice. In the feedback part of the reservoir computing block a switch is provided that allows switching between two paths, which each have an own programmable feedback strength parameter (η_1 and η_2 , respectively). Further also for the non-linear function output signal a switch may be provided to switch between two possible mixing parameters $1-a_1$ and $1-a_2$ and in the feedback loop of the filtering structure switching between two parallel subpaths with parameters a_1 and a_2 , respectively, may be possible. A fixed and predetermined switching sequence $S(n)$ applied to the various switches in the scheme determines how both inputs are mixed in the delay lines. After the

switches have changed position, the inertia of the non-linear node ensures that for some time, its output is influenced by both inputs. The content of the delay line structure is then influenced by both inputs. For example, taking a switching sequence $S(n)$ consisting of $N=2$ 'ones' followed by $N=2$ 'zeros' implies the inputs have almost no interaction, except for those delay elements following but close to halfway the delay line. The highest interaction between both input streams is reached by using a switching sequence consisting of alternating ones and zeros: $S(n) = 1, 0, 1, 0, \dots$. Intermediate interaction is possible by sequences such as $S(n) = 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, \dots$

[0074] Furthermore digital filters can be added in the nodes of the delay line, in such a way that at each time step a different filter is addressed and fed back to the non-linear node. The purpose of these filters is twofold. Firstly, they expand the memory of the reservoir computing block independently of the choice of feedback strength to the non-linear node. Secondly, they allow emphasizing certain frequencies of interest in the input signal. The filters can have any of the known digital filter structures. In Fig.5 two pole filters have been drawn. In the echo state network literature this is known as (bandpass) filtered neurons (BPF).

[0075] As such, this structure is readily implementable on various electronic platforms. The various components of the reservoir computing blocks as shown e.g. in Fig.4 and Fig.5 and described above can advantageously be implemented using digital electronics. In preferred embodiments the reservoir computing device is implemented in FPGA or as a circuit block in a digital ASIC. For such embodiments the analog input signals are first digitized using an analog-to-digital converter. Subsequent processing is then done in the digital domain. The output of the readout layer of the final cascaded unit can then conveniently be transformed into the analog signal domain again via a digital-to-analog converter.

[0076] Alternatively, an implementation in software on a desktop computer or laptop or in an embedded system or otherwise programmable system can be envisaged. Such software embodiments can take on the form of an 'app' on a mobile device.

[0077] A block scheme of an embodiment with several reservoir computing blocks is shown in Fig.6. Just as in Fig.4, instead of only receiving the output of the previous RC block, each RC block receives as input also the original input in an appropriately delayed version : the original input given to the i -th RC block in the cascade is delayed over $i-1$ input samples. This is to avoid introducing a shift between related input and output samples.

[0078] It is a characterising feature of this invention that each reservoir computing block after the first does not only get an input from the preceding block but also the original input signal that was applied to the reservoir computing device as a whole. The preceding block yields the so far best approximation of the desired output signal. The desired output signal is a signal originating from

a measurement of the black box system to be identified. The current reservoir computing block is then capable of combining this so far best approximation with the original input signal to extract new information and obtain an even better approximation of the desired output. This approach is substantially different from prior art solutions having a cascade wherein the reservoir computing blocks only have one input, namely the output of the previous block (and for the first block the original input). The reservoir computing blocks then lose information after the readout layer, because the readout layer projects from a high-dimensional space to a lower dimensional space. The prior art way of cascading, where each block (except the first) receives the output of the previous block, can only be applied if the system to be identified can be physically interpreted and divided in sub-processes. Each reservoir computing block in the cascade can then emulate a single sub-process. This necessitates a training input-output signal pair per sub-process and impedes maintaining an overall black-box approach. In the present invention the need for underlying knowledge of the system to be identified is avoided and a single input-output training pair suffices.

[0079] The operation is understood as follows. Each RC block in the cascade is trained using the original training input as well as the approximation that is produced by the previous RC block. The desired output for the training is the original output of the training pair. For a given set of the parameters $(\gamma_1, \gamma_2, \eta_1, \eta_2, a_1, a_2)$, the (fast) linear training relies on the reservoir states induced by both inputs. The parameters of one RC block are optimized until no further improvement is found, before moving to the next RC block. This optimization can be done by parameter space scanning, gradient descent, genetic algorithm search or any other known optimization technique. As such, each of the following RC blocks receives information not available to the previous one, yet also has the original input signal to work from, resulting in a successive approximation of the system to be identified. This is the key advantage offered over the conventional cascaded approach. Only one input/output training pair and no intermediate signals are needed. Therefore this structure still allows for a black box approach, while at the same time offering inherently increased computational abilities over a single RC block.

[0080] For the dual input reservoir computer blocks forming the cascade, the determination of the weights follows the description given for reservoir computing blocks with a single input, as there is no difference, from a training point of view, between masked samples coming from the original global input or from the previous reservoir computer block output. Once the blocks of the reservoir computer device are trained, i.e. the weights are calculated so that an output is obtained close to the desired training output for the given training input. If the training signals were diverse enough, the machine will generalize to previously unseen data.

[0081] The performance of a delay line reservoir computer for a certain task also depends on the values of the input scalings $\gamma_{1,2}$ and the feedback scaling $\eta_{1,2}$, as well as the a-values in the non-linear node feedback and these need to be optimized too. The optimization of these machine parameters can be done by, for example, a gradient descent or a scan of a certain $(\eta_{1,2}, \gamma_{1,2}, a_{1,2})$ area.

5 At each set of parameter values $(\eta_{1,2}, \gamma_{1,2}, a_{1,2})$, the new optimal readout layer weights w are calculated by, for example, a least squares method as outlined above and the performance is determined. The parameters yielding the best performance, i.e. the least distance from the desired output, are then kept.

[0082] For the cascade, a best parameter set and corresponding weights are determined as
10 above, for each reservoir computing block, before moving on to the next block in the cascade. In this way, the cascade is trained from the first to the last reservoir computer block.

[0083] For optimal operation, a different mask can be used for every RC block in the cascade. These masks may be a permutation of one fixed mask, or chosen with completely different values. Fig.7 provides an illustration. Each block now has its own masking unit. The use of different masks for
15 every RC block in the cascade gives rise to different dynamical signals in the respective reservoir blocks. In that way, the readout layer of each block can extract new information. Note, however, that a different mask is not a priori necessary, but only an enhancement of the system proposed herein. Since each RC block already receives as input a different input signal, namely the estimate by the previous block, their dynamics will certainly differ. The choice for different masks in each RC block, as
20 opposed to a single fixed mask for every RC block, can help enrich the dynamics of the system as a whole, making it possible to extract more information in the readout layers.

[0084] The proposed cascaded structure can still be used in a white box reservoir computing cascade approach. In such an approach a priori knowledge of the internals of the system to be identified lead to parameter ranges within which to look for optimal operation of each RC block in the
25 cascade. In that case the training is performed using intermediate training signals modelling the different stages of the system to be identified.

[0085] A typical use case for the reservoir computing device according to the invention is as follows. An input signal x needs to be somehow related to or non-linearly transformed into a certain
30 output signal y . These signals may originate from an input-output relation measurement of a device under test (DUT) that needs to be emulated and for which there is no known mathematical or algorithmic description of the input-output relation. The DUT can be modelled using a reservoir computing device with a structure as described above. As already mentioned, this approach considers the DUT as a 'black box' system. The task then is to emulate the DUT, i.e. to mimic the input-output

relation that the DUT presents using only signals measurable from the DUT, without other information of the internal working of the DUT. This is achieved by training the reservoir computing device that models the DUT and deriving parameter values.

[0086] In another typical use case, the input signal may be a signal that is somehow contaminated and/or noisy/distorted and the desired output signal is the ideal uncontaminated signal without distortion and/or noise. Once a set of input/output training pairs is established, the output being the desired undistorted or noise-free signal, the reservoir computing device according to the invention that models the DUT, is trained as described earlier. If the set of training signal pairs is diverse enough (meaning these signals are exemplary of the task to be solved), the reservoir computing device will generalize this signal transforming behaviour to other, previously unseen data.

[0087] In order to make reservoir computing applicable to practical problems in real-time audio processing, a fully digital implementation as proposed here is desirable, given the abundance of digital computing hardware in the present day. Obviously, the sampling rate in such a digital implementation needs to be adapted to real-time audio. A minimum rate can be 8 kHz (as in telephony), while a standard rate may be 44.1 kHz.

[0088] As an application example related to real-time audio, the problem of the emulation of the non-linear distortion of guitar amplifiers is considered. In contrast to HiFi amplifiers, guitar amplifiers (or 'amps' for short) are not intended to faithfully reproduce an amplified version of the input signals. Instead they add linear and non-linear distortion, thereby colouring and shaping the sound. The severity of the distortion can go from adding light accents over a more crunchy sound to full-on clipping, depending on the music genre and artist. The sources of non-linear distortion within these amps include saturation in tube based amplifier stages, saturation of the magnetic core of the output transformer, supply voltage sinks caused by deliberately underdimensioned supply bypass capacitors and/or clipping introduced by diode clipper circuits. Indeed, the design of a "good" guitar amp is more art than science and the amplifier is at least as much responsible for the resulting sound as the guitar is. Furthermore, every brand and even type of amplifier has its own characteristic sound, often instantly recognizable by seasoned players.

[0089] Thus there is a problem of obtaining a range of sounds, while on a budget too limited to obtain many different amps. One technical solution for this is on the market today, namely the Kemper Profiling Amplifier. Its custom hardware is capable of learning and emulating the specific sound of an amp, but it does not offer a software solution that runs on other equipment.

[0090] The reservoir computing device presented above is applied to learn and then emulate in real time the distortion of a guitar amplifier, using a standard laptop as the computing platform. Reservoir computing has been used before for the purpose of emulating tube like distortion.

However, these attempts were not in real time and not starting from a real world amplifier, but rather from a model distortion.

[0091] A Behringer UCG102 Guitar-to-USB interface is used to record the guitar signal and the amplifier response. This basically is an external soundcard having a high-impedance input suited for electrical guitars, and a headphone output. A small board with switches and connectors was built to route the signals between the soundcard, the guitar and the amplifier in an easy fashion. No parts of the hardware used here are central to the reservoir computing device of the present invention, yet are only used for convenient signal routing. A Marshall Valvestate 10 is used, which is a small 10 W practice amplifier, with the "boost select" switch to on, as the system to be emulated. The amplifier response is measured at the line-out jack. Alternatively, a microphone can be placed in front of the amp speaker, as is often done in studio recordings.

[0092] By means of a software program the following steps are performed :

- send out a short sinewave to measure the loopthrough amplification factor of the soundcard,
- instruct the user to make the necessary adjustments on the amplifier to obtain a desired sound,
- record a few seconds of actual guitar play,
- excite the amplifier with the recorded guitar signal and simultaneously record the response of the amplifier.

[0093] The guitar signal and amplifier response are now used as an input/output training pair. For the training two seconds or 2×44100 samples of sound were used. Fig.8 shows a small portion of the guitar signal and resulting distorted amplifier signal, as measured using this setup.

[0094] A device with a cascade of 5 RC blocks is used, each having $N = 100$ neurons. The centre frequencies of the two pole filter bank are logarithmically spaced between 400 Hz and 19 kHz. The function in the non-linear node is chosen as $f(x) = x/(1+x^2)$. Each RC block was given a mask with values drawn at random from $\pm\{0.2, 0.6, 1.0\}$. The parameters and weights of the cascaded RC device are then optimized and saved in a 'patch' file. This patch file now contains all the settings needed to emulate the amp. A 'player' program then uses the patch to emulate the amplifier in real time. The emulated output is found at the headphone output of the soundcard. Fig.9 illustrates the good correspondence between the emulated and real distortion.

[0095] Another example application is the removal of babble noise from a hearing aid microphone signal. As the name suggests, babble noise consists of many simultaneous conversations going on in the background. This problem is also known as the 'cocktail party' effect. As known by those working in the field of hearing aids, speech intelligibility is highly influenced by the intensity and type of background noise. Babble noise can partially mask a speaker. Babble noise is particularly detrimental to speech intelligibility, because it is aperiodic. Therefore our brains cannot filter it out

very well. Since babble noise occupies the same bandwidth as the main speaker, linear filtering techniques are not useful. A conventional method to suppress babble and other types of noise uses the phase adjusted sum of two microphones in a technique reminiscent of antenna beamforming. The cascaded reservoir structure of the present invention can be used to nonlinearly filter the background noise and thus improve intelligibility of the main speaker. By way of example a noisy speech corpus database is considered. A reservoir computing device comprising a cascade of five RC blocks of one hundred virtual neurons trained with the regular time based training. The training input is a male voice uttering the sentence "*The sky that morning was clear and bright blue*" in the presence of babble noise, such that the signal-to-noise ratio is 0 dB. In the database the speech is recorded with a sample rate of 8 kHz. The target response is the same speech sample without the noise. The first 10000 samples are used for training and the system is then tested on the next 100000 samples. Results are shown in Fig.10. The noisy input signal is shown in Fig.10a and the clean target signal in Fig.10b. The response of the cascaded reservoir computing device is presented in Fig.10c. Visually, there is a resemblance between the target and response. Auditorily, there is a marked improvement in the speech intelligibility as the babble noise is severely reduced, while the speech remains largely intact.

[0096] To illustrate the strength of the proposed solution with a dual-input reservoir computing cascade, its performance is compared to that of some other forms of cascades. A cascade of five of dual-input reservoir computing blocks with dual inputs (i.e. output of the previous one and global input) as in the present invention is considered and compared to other cascades built using only the information found in the state of the art literature. Each block has one hundred filtered virtual neurons. The task at hand was the emulation of the distortion of a guitar amplifier (a Marshal V10, see Figs 8 and 9). The performance is measured as a normalized root mean squared error (NRMSE), so a lower obtained figure is better. Each dot in Fig.11 corresponds to an NRMSE obtained for training and testing of a reservoir computer in the cascade with a specific set of parameters $\gamma_{1,2}$, $\eta_{1,2}$, $a_{1,2}$. The parameters are in this example chosen by an algorithm called Random Optimization, but that does not matter for the further discussion.

[0097] For every reservoir computing block, twenty tries of the algorithm are performed. After every twenty iterations the best parameters for that computing block are kept and the output obtained by that block for those training parameters as well as the original or global input are transferred to the next computing block. Obviously, the first block only receives the global input. Then the training proceeds to the next block. Clearly, a successively better approximation is obtained. One thus reaches NRMSE = 0.106. At every stage, the performance improves significantly.

[0098] In order not to overload Fig.11, the best result obtained for each stage of the cascades is shown in the following. For comparison, the best result obtained for a single reservoir block with 500 filtered neurons (trained using the same algorithm) is shown as a dashed line marked 'A' (NRMSE = 0.220). A single reservoir computing block of 500 unfiltered virtual neurons marked by

The dashed line marked 'C' shows the NRMSE for a cascade of five single input blocks, again each having 100 unfiltered neurons. Here, all but the first block receive only the output of the previous RC and not the global input. This results in NRMSE = 0.180. As compared to the approach of the invention (the black dots), the incremental gains from adding stages to the cascade levels off. This is entirely in line with the results obtained by Treifenbach et al., referenced in the above-mentioned Grèzes paper. Other authors have come to the same conclusion regarding cascading reservoir computers, as this phenomenon seems to be task independent.

The dash-dot-dot line marked with 'D' shows the results for a cascade of five single input RCs with filtered neurons, resulting in NRMSE = 0.144, indicating the merit of filtered neurons in the invention.

Specifically, the performance is shown in Fig.11 of a cascade of five reservoir computer blocks. In this graph, the performance is measured in terms of the NRMSE (normalized root mean squared error), meaning the lower the figure the better the performance is.

Finally, the dashed line marked with 'E' shows the case of the cascade with dual input blocks, but with unfiltered virtual neurons, yielding NRMSE=0.126.

[0099] It is clear from Fig.11 that the dual-input approach combined with filtered virtual neurons shows significant improvement over other cascaded approaches for the task of black box system identification. It is to be noted that this task is of great practical importance in many fields.

[0100] The reservoir computing device (RCD) according to this invention may be targeted towards applications where signals that are 'audio-like' (time series) have to be processed/transformed or categorized.

[0101] The invention offers solutions to problems present in today's state-of-the-art audio systems, such as loudspeaker non-linear distortion compensation and non-linear room equalisation.

[0102] Another possible application is situational-aware volume control (for example, lowering music player volume when the sound of a nearby ambulance is detected, such that a user is informed of impending traffic danger). Another task for the reservoir computing device is the classification of engine vibration patterns, to provide early warning of imminent failure. This could both help to optimize service intervals and to decrease unexpected down time. Many users of hearing aids experience significant loss of speech intelligibility in the presence of babble noise (the so called

“cocktail party effect”). This kind of noise is impossible to filter out with conventional means, because it resides within the same bandwidth as the main speech that the user wants focus to. First results show that babble noise can be filtered with the proposed RC structure. Other applications in the medical sphere are detection of sleep apnea and classification of heart sounds. Self-administered tests controlled by portable handheld devices can both increase population coverage and decrease testing costs. Many of these novel applications demand either (or both) privacy and low-latency signal processing, such that cloud based solutions, where processing is outsourced to a large server, are likely to be problematic.

[0103] It is to be noted that alternative technologies, such as Deep Learning, are known to require very long training times (due to the backpropagation algorithm) as compared to reservoir computer training times (pseudoinverse methods).

[0104] While the invention has been illustrated and described in detail in the drawings and foregoing description, such illustration and description are to be considered illustrative or exemplary and not restrictive. The foregoing description details certain embodiments of the invention. It will be appreciated, however, that no matter how detailed the foregoing appears in text, the invention may be practiced in many ways. The invention is not limited to the disclosed embodiments.

[0105] Other variations to the disclosed embodiments can be understood and effected by those skilled in the art in practicing the claimed invention, from a study of the drawings, the disclosure and the appended claims. In the claims, the word “comprising” does not exclude other elements or steps, and the indefinite article “a” or “an” does not exclude a plurality. A single processor or other unit may fulfil the functions of several items recited in the claims. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage. A computer program may be stored/distributed on a suitable medium, such as an optical storage medium or a solid-state medium supplied together with or as part of other hardware, but may also be distributed in other forms, such as via the Internet or other wired or wireless telecommunication systems. Any reference signs in the claims should not be construed as limiting the scope.

Claims

1. Reservoir computing device comprising a cascade of at least two reservoir computing blocks, each reservoir computing block comprising
- 5 - summing means for making a sum of a block input signal and a delayed signal output by the reservoir computing block,
- a digital filter structure arranged for applying a non-linear function to said sum and for adding to a non-linear function output a filtered version of a preceding output of said non-linear function, yielding a digital filter structure output signal,
- 10 - a delay line structure arranged for receiving as input said digital filter structure output signal and for outputting a delay line structure output signal to a readout layer, said delay line structure comprising a plurality of delay elements arranged for outputting said delayed signal being a delayed version of said digital filter structure output signal,
- whereby said block input signal is in the first reservoir computing block of said cascade a signal
- 15 applied to said reservoir computing device and whereby in each subsequent reservoir computing block of said cascade said block input signal is a delayed version of said signal applied to said reservoir computing device, the delay line structure output signal applied to said readout layer by the previous reservoir computing block in said cascade or a combination of said delayed version and said delay line structure output signal, whereby said delayed version, said delay line structure output signal and/or
- 20 said combination are applied according to a given pattern.
2. Reservoir computing device as in claim 1, arranged for scaling said signal applied to said reservoir computing device with a first scaling factor and for scaling said delay line structure output signal applied to the readout layer by the previous reservoir computing block with a second scaling factor.
- 25 3. Reservoir computing device as in claim 1 or 2, wherein said signal applied to said reservoir computing device is a product of an external data signal with a masking function.
4. Reservoir computing device as in any of claims 1 to 3, wherein said digital filter structure comprises a first switch to guide said non-linear function output along either a first or a second of two parallel paths, each of said parallel paths having an own scaling factor.
- 30 5. Reservoir computing device as in any of the previous claims, wherein said digital filter structure comprises a first order filter yielding said filtered version of said preceding non-linear function output.
6. Reservoir computing device as in any of the previous claims, wherein said digital filter structure comprises a second switch to guide said filtered version of said preceding non-linear function output

along either a first or a second of two further parallel paths, each of said further parallel paths having an own scaling factor.

7. Reservoir computing device as in any of the previous claims, comprising a third switch to guide said delayed signal along either a first or a second of two parallel feedback paths, each of said parallel
5 feedback paths having an own feedback scaling factor.

8. Reservoir computing device as in any of the previous claims, wherein said delay line structure output signal outputted to said readout layer is said digital filter structure output signal.

9. Reservoir computing device as in any of the previous claims, wherein said plurality of delay elements is part of a further digital filter and an output of said further digital filter is used as said
10 delayed signal output.

10. Method for emulating a device under test arranged for receiving an input signal and yielding an output signal, said method comprising

- modelling said device under test as a reservoir computing device comprising a cascade of at least two reservoir computing blocks as in any of claims 1 to 9,
- 15 - measuring a plurality of input/output pairs of input signals and output signals applied to said device under test, said input/output pairs forming a training set,
- training with said training set said device under test modelled as said reservoir computing device,
- determining from said training internal parameters of at least one of said digital filter
20 structure, said delay line structure and said readout layer of at least one of said reservoir computing blocks of said reservoir computing device,
- emulating said device under test using said determined internal parameters.

11. Method for emulating a device under test as in claim 10, wherein said device under test is a real time audio processing system.

25 12. Method for emulating a device under test as in claim 11, wherein said device under test is a guitar amplifier.

13. Method for emulating a device under test as in claim 10 or 11, wherein said device under test is a hearing aid comprising a microphone which receives babble noise.

14. Use of the method as in claim 10, for classifying engine vibration patterns.

30 15. Use of the method as in claim 10, for classifying heart sounds.

16. Use of the method as in claim 10, for detecting sleep apnea.

1/6

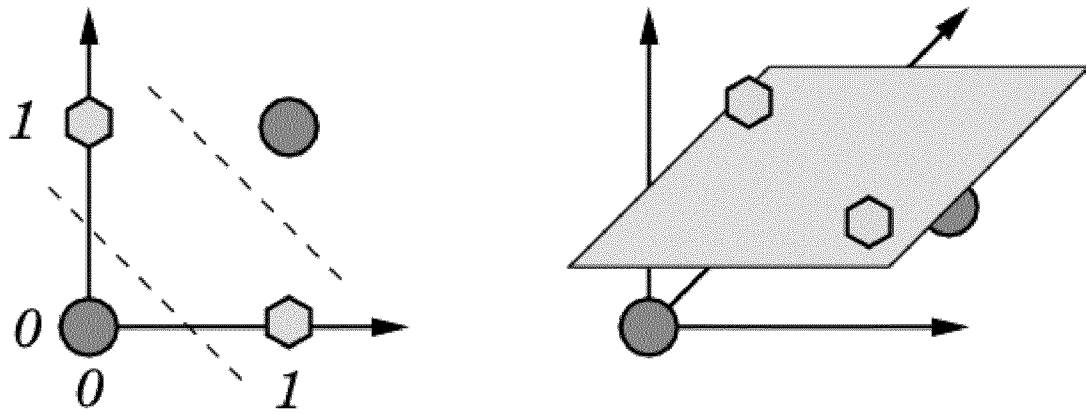


Fig.1

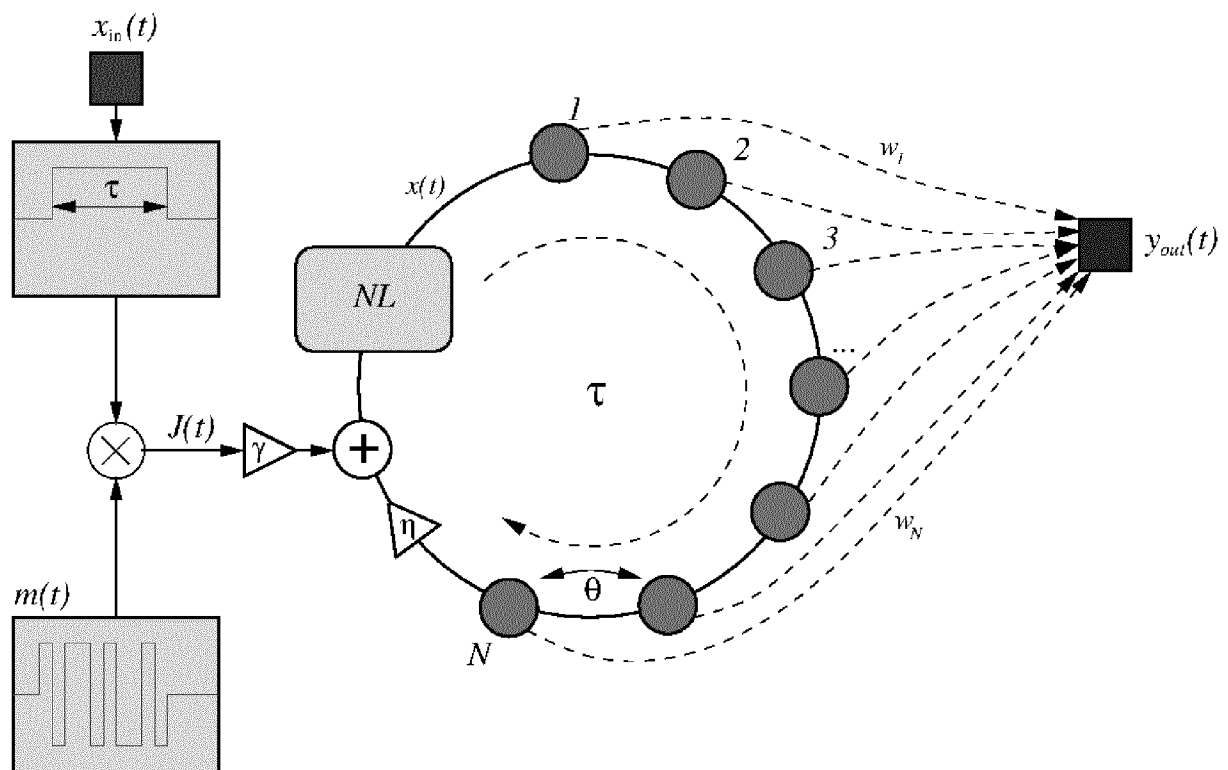


Fig.2

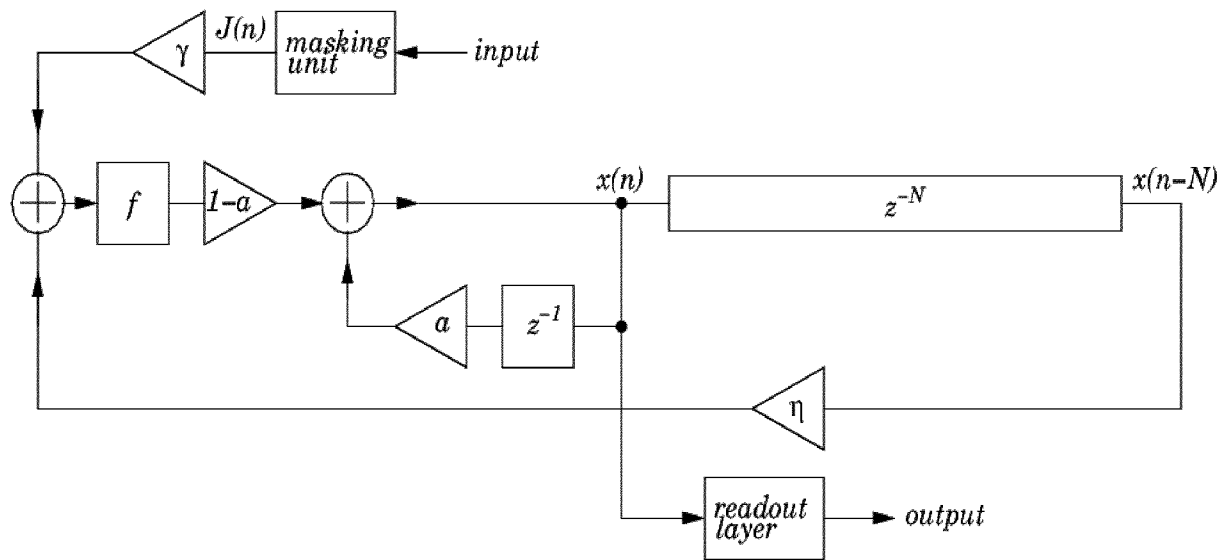


Fig.3

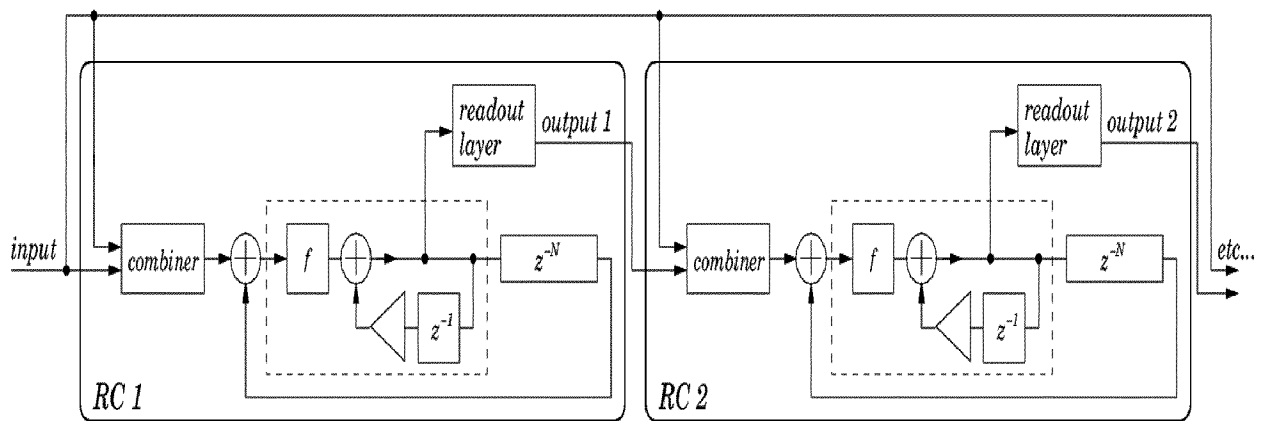


Fig.4

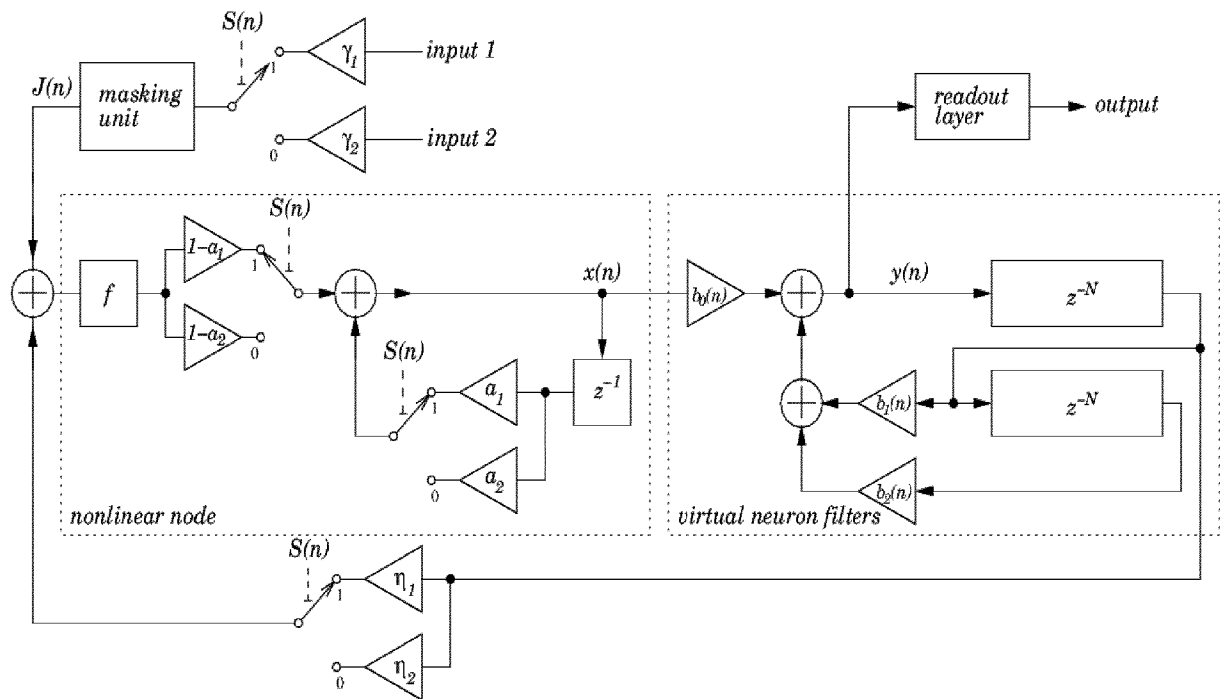


Fig.5

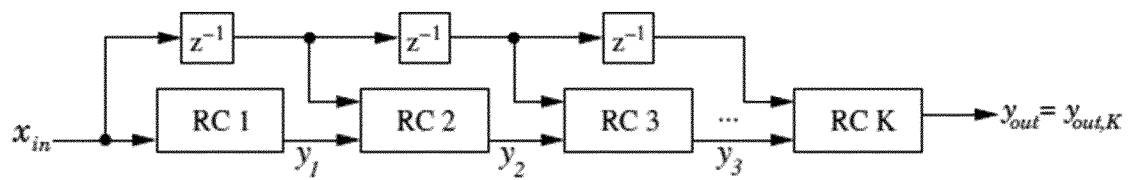


Fig.6

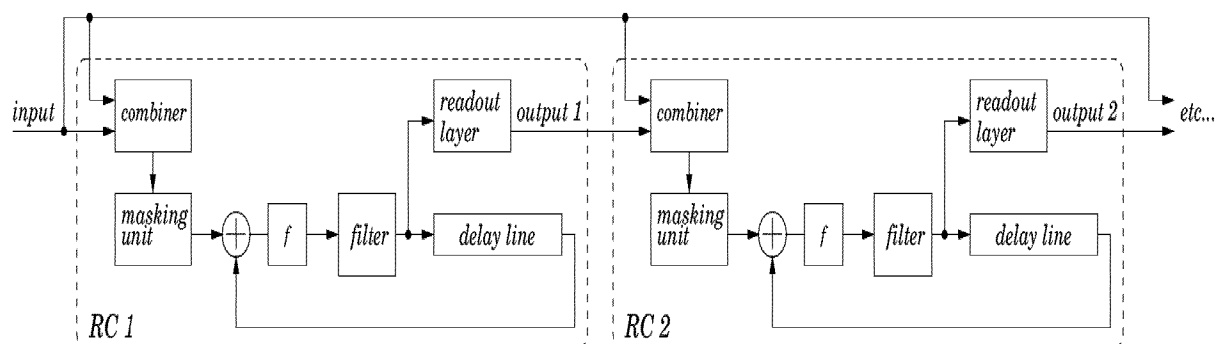


Fig.7

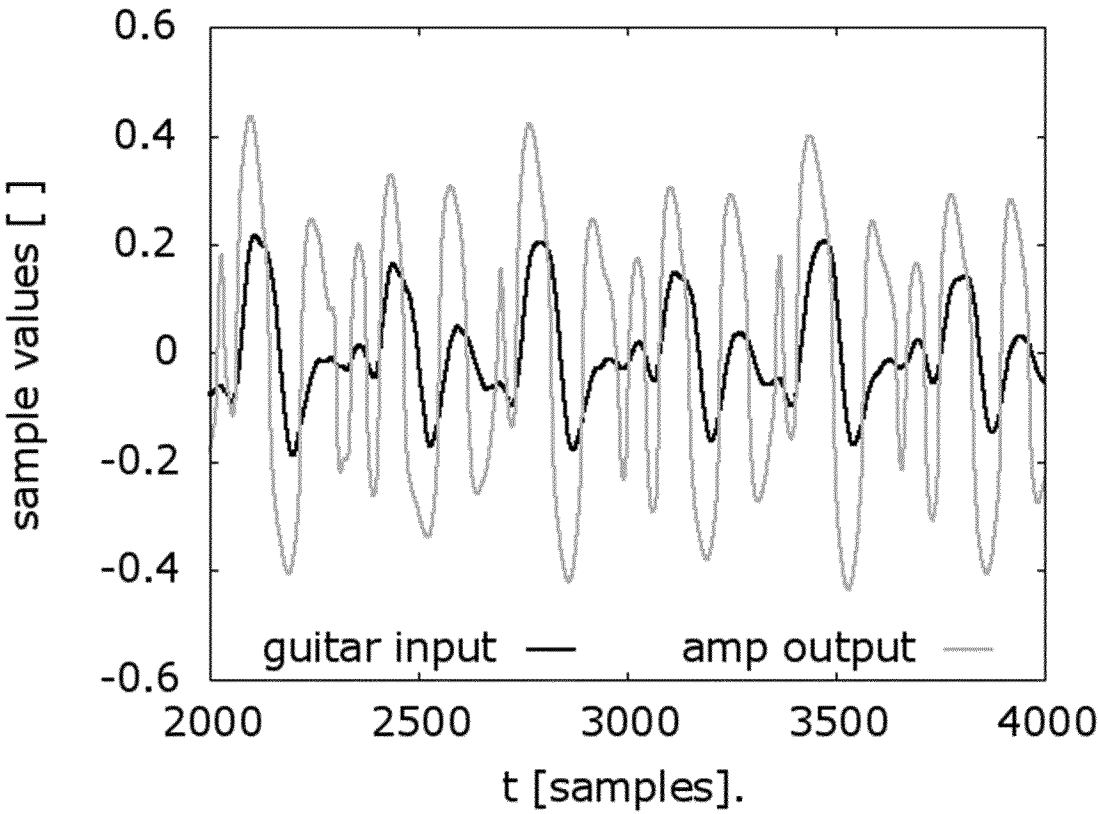


Fig.8

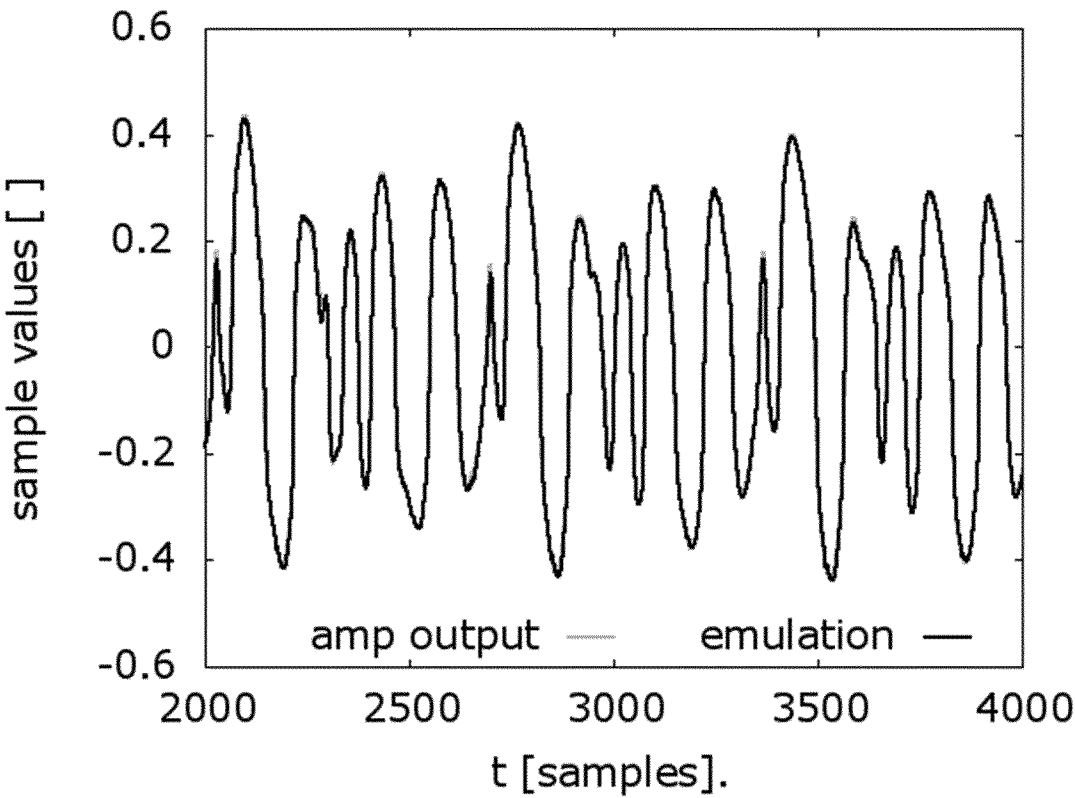
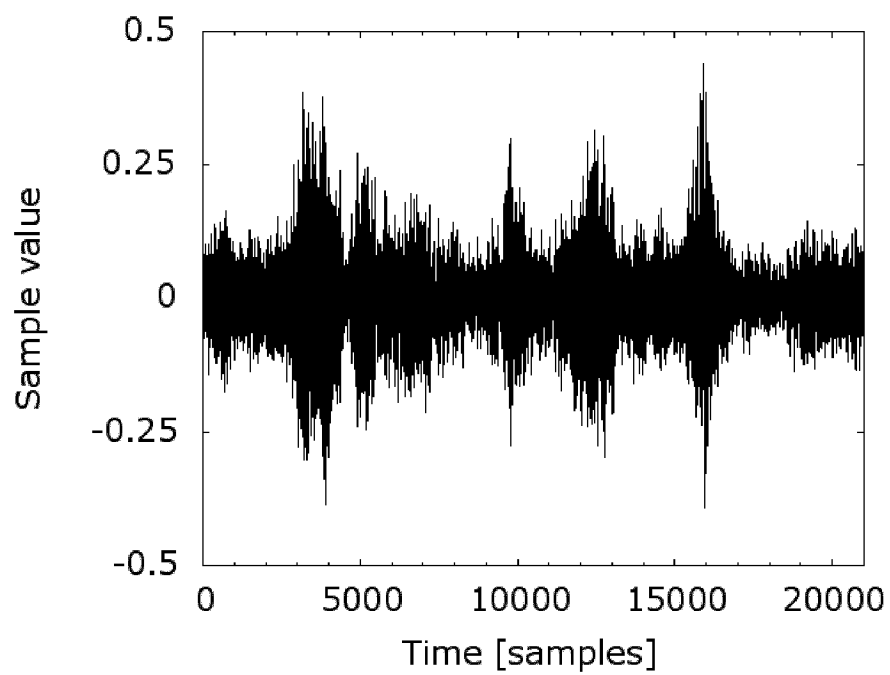
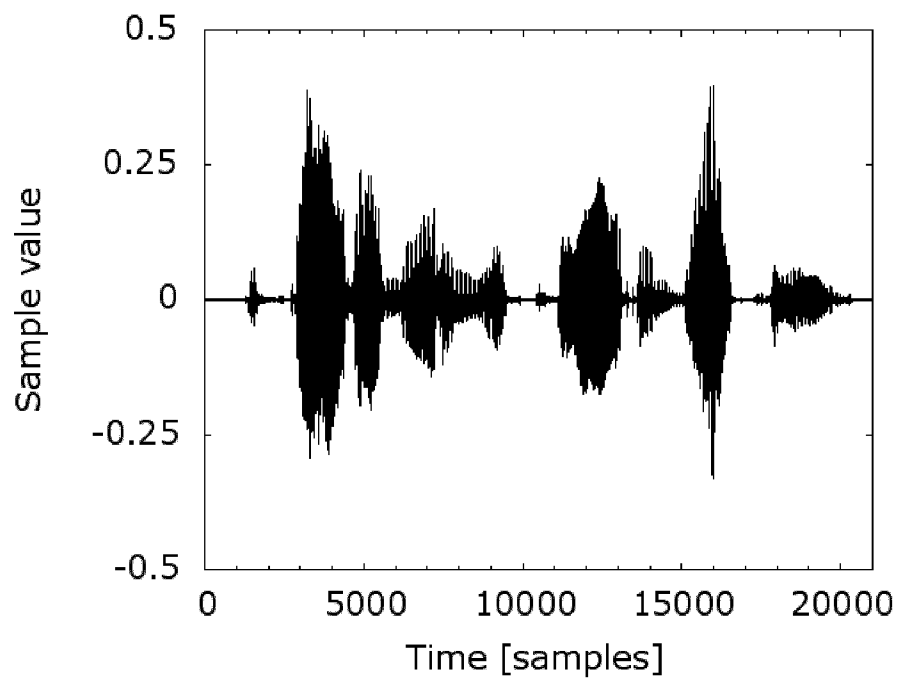
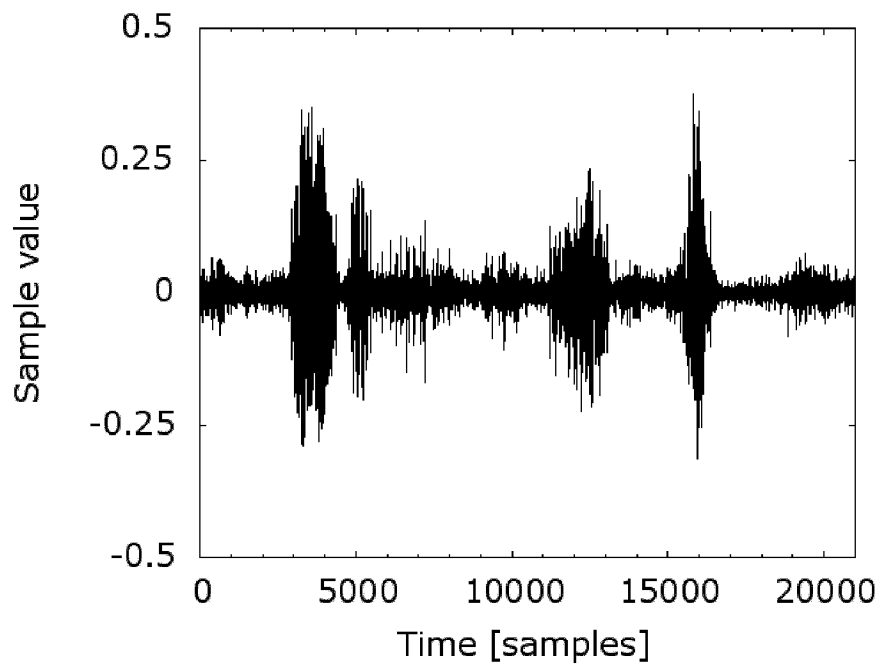
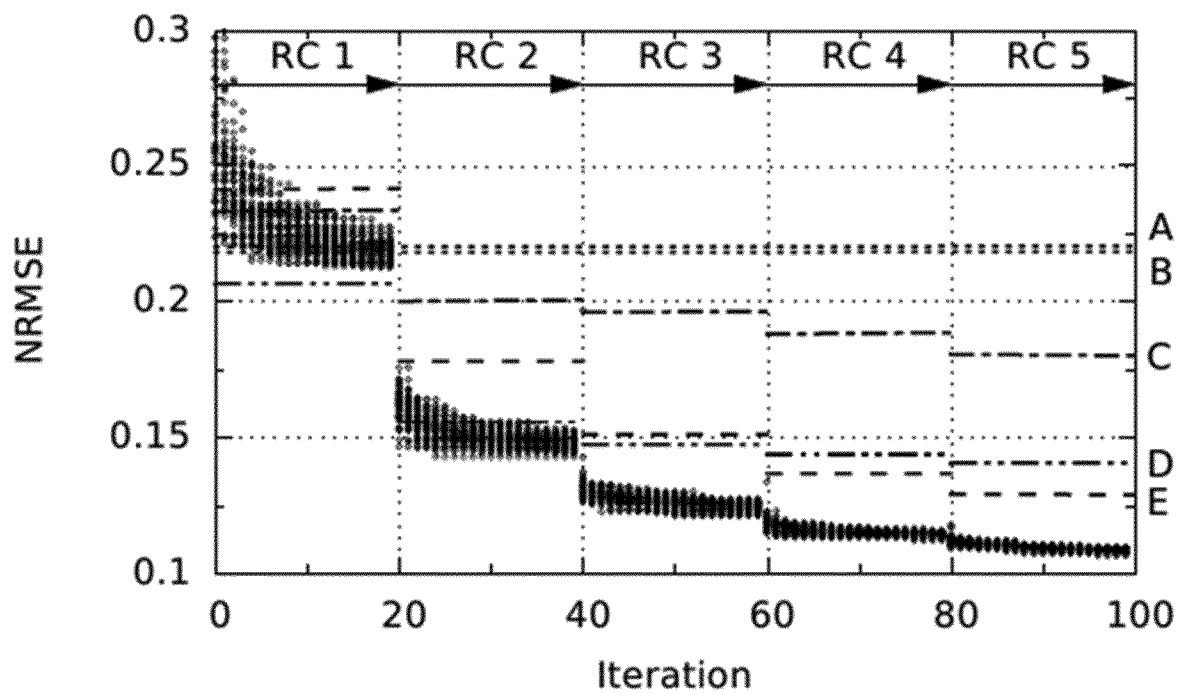


Fig.9

5/6

Fig.10aFig.10b

Fig.10cFig.11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/081239

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06N3/04 G06N3/063
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EP0-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	M. C. SORIANO ET AL: "Minimal approach to neuro-inspired information processing", FRONTIERS IN COMPUTATIONAL NEUROSCIENCE, vol. 9, 68, 2 June 2015 (2015-06-02), XP055281664, DOI: 10.3389/fncom.2015.00068 sections 1, 2 and 4	1-16
X	N. D. HAYNES ET AL: "Reservoir computing with a single time-delay autonomous Boolean node", PHYSICAL REVIEW E, vol. 91, 020801R, 23 February 2015 (2015-02-23), XP055281686, DOI: 10.1103/PhysRevE.91.020801 the whole document	1-16



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

10 March 2017

Date of mailing of the international search report

22/03/2017

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Douarche, Nicolas

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/081239

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	F. Grèzes: "Reservoir computing", Dissertation in Computer Science at Graduate Center of City University of New York, 4 September 2014 (2014-09-04), XP055281665, Retrieved from the Internet: URL: http://www.gc.cuny.edu/CUNY_GC/media/Computer-Science/Student Presentations/Felix Grezes/Second_Exam_Survey_Felix_Grezes_9_04_2014.pdf [retrieved on 2016-06-17] sections 3.3 and 3.4	1-16
X	L. APPELTANT: "Reservoir computing based on delay-dynamical systems", THESIS FOR PHD IN ENGINEERING SCIENCES AT UNIVERSITÉ LIBRE DE BRUXELLES, 22 May 2012 (2012-05-22), XP055281684, ISBN: 978-90-5718-120-7 chapters 2, 3, 5 and 6	1-16
X	E. A. ANTONELO, B. SCHRAUWEN: "Online learning navigation behaviors for small mobile robots with reservoir computing architectures", IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, vol. 26, no. 4, 3 June 2014 (2014-06-03), pages 763-780, XP011576090, DOI: 10.1109/TNNLS.2014.2323247 sections III and VI	1-16
X	T. WAEGEMAN ET AL: "Feedback control by online learning an inverse model", IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, vol. 23, no. 10, 15 August 2012 (2012-08-15), pages 1637-1648, XP011460816, DOI: 10.1109/TNNLS.2012.2208655 chapters II and III	1-16
A	F. TRIEFENBACH ET AL: "Acoustic modeling with hierarchical reservoirs", IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, vol. 21, no. 11, 29 August 2013 (2013-08-29), pages 2439-2450, XP011529573, DOI: 10.1109/TASL.2013.2280209 the whole document	1-16

-/--

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/081239

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	S. Ortín et al: "Memory and nonlinear mapping in reservoir computing with two uncoupled nonlinear delay nodes", The 2012 European Conference on Complex Systems (ECCS'12), 6 September 2012 (2012-09-06), XP055282100, Retrieved from the Internet: URL:http://www.ulb.ac.be/polytech/spg/ECCS2012/Presentations_ECCS_2012.zip [retrieved on 2016-06-17] the whole document -----	1-16