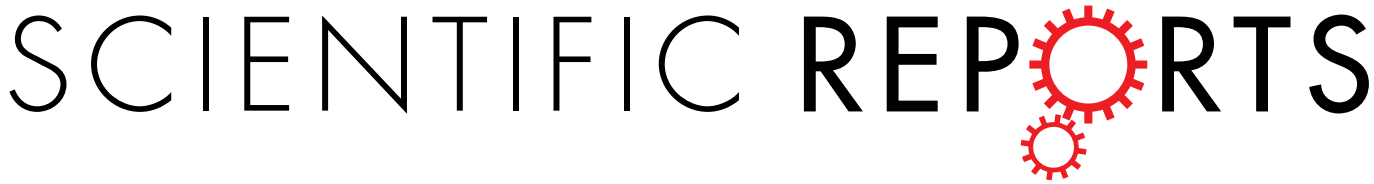

OPEN

Optimal nonlinear information processing capacity in delay-based reservoir computers

Received: 04 March 2015
Accepted: 03 July 2015
Published: 11 September 2015

Lyudmila Grigoryeva[1], Julie Henriques[1,2], Laurent Larger[3] & Juan-Pablo Ortega[4]

Reservoir computing is a recently introduced brain-inspired machine learning paradigm capable of excellent performances in the processing of empirical data. We focus in a particular kind of time-delay based reservoir computers that have been physically implemented using optical and electronic systems and have shown unprecedented data processing rates. Reservoir computing is well-known for the ease of the associated training scheme but also for the problematic sensitivity of its performance to architecture parameters. This article addresses the reservoir design problem, which remains the biggest challenge in the applicability of this information processing scheme. More specifically, we use the information available regarding the optimal reservoir working regimes to construct a functional link between the reservoir parameters and its performance. This function is used to explore various properties of the device and to choose the optimal reservoir architecture, thus replacing the tedious and time consuming parameter scannings used so far in the literature.

The increase in need for information processing capacity, as well as the physical limitations of the Turing or von Neumann machine methods implemented in most computational systems, have motivated the search for new brain-inspired solutions some of which present an outstanding potential. An important direction in this undertaking is based on the use of the intrinsic information processing abilities of dynamical systems[1] which opens the door to high performance physical realizations whose behavior is ruled by these structures[2,3].

The contributions in this paper take place in a specific implementation of this idea that is obtained as a melange of a recently introduced machine learning paradigm known under the name of **reservoir computing (RC)**[4–10] with a realization based on the sampling of the solution of a time-delay differential equation[11,12]. We refer to this combination as **time-delay reservoirs (TDRs)**. Physical implementations of this scheme carried out with dedicated hardware are already available and have shown excellent performances in the processing of empirical data: spoken digit recognition[13–17], the NARMA model identification task[11,18], continuation of chaotic time series, and volatility forecasting[19]. A recent example that shows the potential of this combination is the results in[17] where an optoelectronic implementation of a TDR is capable of achieving the lowest documented error in the speech recognition task at unprecedented speed in an experiment design in which digit and speaker recognition are carried out in parallel.

A major advantage of RC is the linearity of its training scheme. This choice makes its implementation easy when compared to more traditional machine learning approaches like recursive neural networks, which usually require the solution of convoluted and sometimes ill-defined optimization problems. In exchange, as it can be seen in most of the references quoted above, the system performance is not robust

[1]Laboratoire de Mathématiques de Besançon, UMR CNRS 6623, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon cedex. France. [2]Cegos Deployment. 11, rue Denis Papin. F-25000 Besançon. [3]FEMTO-ST, UMR CNRS 6174, Optics Department, Université de Franche-Comté, UFR des Sciences et Techniques. 15, Avenue des Montboucons. F-25000 Besançon cedex. France. [4]Centre National de la Recherche Scientifique, Laboratoire de Mathématiques de Besançon, UMR CNRS 6623, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon cedex. France. Correspondence and requests for materials should be addressed to J.-P.O. (email: Juan-Pablo.Ortega@univ-fcomte.fr)

with respect to the choice of the parameter values $\boldsymbol{\theta}$ of the nonlinear kernel used to construct the RC (see below). More specifically, small deviations from the optimal parameter values can seriously degrade the performance and moreover, the optimal parameters are highly dependent on the task at hand. This observation makes the kernel parameter optimization a very important step in the RC design and has motivated the introduction of alternative parallel-based architectures[19,20] to tackle this difficulty.

The main contribution of this paper is the introduction of an approximated model that, to our knowledge, provides the first rigorous analytical description of the delay-based RC performance. This powerful theoretical tool can be used to systematically study the delay-based RC properties and to replace the trial and error approach in the choice of architecture parameters by well structured optimization problems. This method simplifies enormously the implementation effort and sheds new light on the mechanisms that govern this information processing technique.

TDRs are based on the interaction of the time-dependent input signal $z(t) \in \mathbb{R}$ that we are interested in with the solution space of a time-delay differential equation of the form

$$\dot{x}(t) = -x(t) + f(x(t-\tau), I(t), \boldsymbol{\theta}), \tag{1}$$

where f is a nonlinear smooth function (we call it **nonlinear kernel**) that depends on the K parameters in the vector $\boldsymbol{\theta} \in \mathbb{R}^K$, $\tau > 0$ is the **delay**, $x(t) \in \mathbb{R}$, and $I(t) \in \mathbb{R}$ is obtained using a temporal multiplexing over the delay period of the input signal $z(t)$ that we explain later on. We note that, even though the differential equation takes values in the real line, its solution space is infinite dimensional since an entire function $x \in C^1([-\tau, 0], \mathbb{R})$ needs to be specified in order to initialize it. The choice of nonlinear kernel is determined by the intended physical implementation of the computing system; we focus on two parametric sets of kernels that have already been explored in the literature, namely, the Mackey-Glass[21] and the Ikeda[22] families. These kernels were used for reservoir computing purposes in the RC electronic and optic realizations in[14] and[15], respectively.

In order to visualize the TDR construction using a neural networks approach it is convenient, as in[12,14], to consider the Euler time-discretization of (1) with integration step $d := \tau/N$, namely,

$$\frac{x(t) - x(t-d)}{d} = -x(t) + f(x(t-\tau), I(t), \boldsymbol{\theta}). \tag{2}$$

The design starts with the choice of a number $N \in \mathbb{N}$ of **virtual neurons** and of an adapted **input mask** $\mathbf{c} \in \mathbb{R}^N$. Next, the input signal $z(t)$ at a given time t is multiplexed over the delay period by setting $\mathbf{I}(t) := \mathbf{c}z(t) \in \mathbb{R}^N$ (see Module A in Fig. 1). We then organize it, as well as the solutions of (2), in **neuron layers** $\mathbf{x}(t)$ parametrized by a discretized time $t \in \mathbb{Z}$ by setting

$$x_i(t) := x(t\tau - (N-i)d), \quad I_i(t) := I(t\tau - (N-i)d), \quad i \in \{1, \ldots, N\}, \quad t \in \mathbb{Z}, \tag{3}$$

where $x_i(t)$ and $I_i(t)$ stand for the ith-components of the vectors $\mathbf{x}(t)$ and $\mathbf{I}(t)$, respectively, with $t \in \mathbb{Z}$. We say that $x_i(t)$ is the ***i*th neuron value of the *t*th layer of the reservoir** and d is referred to as the **separation between neurons**. With this convention, the solutions of (2) are described by the following recursive relation:

$$\begin{aligned} x_i(t) &:= e^{-\xi} x_{i-1}(t) + (1 - e^{-\xi}) f(x_i(t-1), I_i(t), \boldsymbol{\theta}), \quad \text{with} \\ x_0(t) &:= x_N(t-1), \quad \text{and} \\ \xi &:= \log(1+d), \end{aligned} \tag{4}$$

that shows how, as depicted in Module B in Fig. 1, any neuron value is a convex linear combination of the previous neuron value in the same layer and a nonlinear function of both the same neuron value in the previous layer and the input. The weights of this combination are determined by the separation between neurons; when the distance d is small, the neuron value $x_i(t)$ is mainly influenced by the previous neuron value $x_{i-1}(t)$, while large distances between neurons give predominance to the previous layer and foster the input gain. The recursions (4) uniquely determine a smooth map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ that specifies the neuron values as a recursion on the neuron layers via an expression of the form

$$\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{I}(t), \boldsymbol{\theta}), \tag{5}$$

where F is constructed out of the nonlinear kernel map f that depends on the K parameters in the vector $\boldsymbol{\theta}$; F is referred to as the **reservoir map**.

The construction of the TDR computer is finalized by connecting, as in Module C of Fig. 1, the reservoir output to a linear readout $\mathbf{W}_{\text{out}} \in \mathbb{R}^N$ that is calibrated using a training sample by minimizing the associated task mean square error via a linear regression. We will refer to the Module B in Fig. 1 as the **reservoir** or the **time-delay reservoir** (TDR) and to the collection of the three modules as the **reservoir computer** (RC) or the **TDR computer**. A TDR based on the direct sampling of the solutions of (1) will

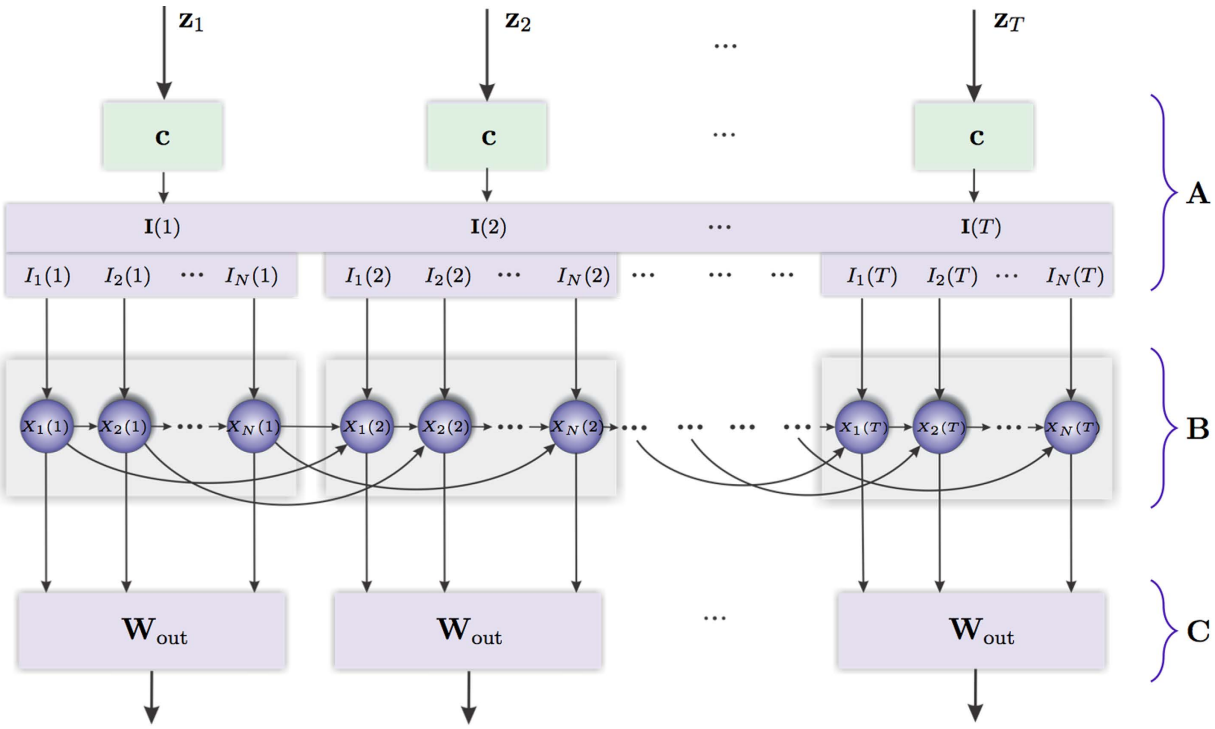

Figure 1. Neural diagram representing the architecture of the time-delay reservoir (TDR) and the three modules of the reservoir computer (RC): (**A**) is the input layer, (**B**) is the time-delay reservoir, and (**C**) is the readout layer.

be called a **continuous time TDR** and those based on the recursion (5) will be referred to as **discrete time TDRs**.

As we already mentioned, the performance of the RC for a given task is much dependent on the value of the kernel parameters $\boldsymbol{\theta}$ and, in some cases, on the entries of the input mask $\mathbf{c}$ used for signal multiplexing. When comparing RC to standard neural networks and thinking of it as a machine learning paradigm, the RC training phase can be assimilated to the determination of both the linear readout $\mathbf{W}_{\text{out}}$ (straightforward in this case using a linear regression) and the optimal parameters $\boldsymbol{\theta}$. Unlike the situation encountered in the neural networks context for which efficient training algorithms have been developed over the years (see[23] for a particularly good performing example), the optimal parameters $\boldsymbol{\theta}$ are usually determined in the RC context by trial and error or using computationally costly systematic scannings that are by far the biggest burden at the time of adapting the RC to a new task.

In this paper we construct an approximate model that we use to establish a functional link between the RC performance and the parameters $\boldsymbol{\theta}$ and the input mask values $\mathbf{c}$. Given a specific task, this explicit expression can be used to find appropriate parameter and mask values by solving a well structured and algorithmically convenient optimization problem that readily provides them.

The construction of this approximated formula is based on the observation that the optimal RC performance is always obtained when the TDR is working in a **stable unimodal regime**, that is, the reservoir is initialized at a stable equilibrium of the autonomous system ($I(t)=0$) associated to (1) and the mean and variance of the input signal $I(t)$ are designed using the input mask $\mathbf{c}$ so that the reservoir output remains around it and does not visit other stable equilibria or dynamical elements. In the next section we provide empirical and theoretical arguments for this claim. The performance measures that we consider in our study are the nonlinear memory capacities introduced in[24] as a generalization of the linear concept proposed in[25–28].

Results

Optimal performance: stability and unimodality. *Stability and the reservoir defining properties.* The estimations of the RC performance using the nonlinear memory capacity that we present later on, consist of approximating the reservoir by its partial linearization at the level of the delayed self feedback term and of respecting the nonlinearity in the input injection. This approach is only acceptable when the optimal dynamical regime that we are interested in, remains close to a given point. A natural candidate for such qualitative behavior could be obtained by initializing the reservoir at an asymptotically stable equilibrium of the autonomous system associated to (1) and by controlling the mean and the variance

of the input signal $I(t)$ so that the reservoir output remains close to it. This equilibrium point can be interpreted as an input bias that, for general RCs, is one of the most commonly optimized parameters.

There is both theoretical and empirical evidence that suggests that optimal performance is obtained for the RCs that we are interested in when working in a statistically stationary regime around a stable equilibrium. Indeed, one of the defining features of RC, namely the **echo state property**, is materialized for general RCs by enforcing that the spectral radius of the internal connectivity matrix of the reservoir is smaller than one[5,6,10], which is the critical stability value for a quiescent state of the network when operating autonomously (without external injected information). It is well-known that the translation of this condition for TDRs implies parameter settings that ensure the existence of a stable state of (1) when $I(t)$ is set to zero. This feature typically relates to gains of the feedback smaller than the Hopf threshold of the delay dynamics or, equivalently, to a sufficiently low feedback rate so that self-sustained oscillations are avoided.

Asymptotic stability is closely related with the so-called **fading memory property**[6,29]: the impact of any past injected input necessarily vanishes after a transient whose duration is typically of the order of the absolute value of the inverse of the smallest negative real part in the Lyapunov exponents. When the feedback gain is set too close to zero, the RC does not exhibit a long enough transient and thus presents an intrinsic memory that is too short to secure the self mixing of the temporal information necessary for its processing. On the other hand, if the feedback gain is set too close to the instability threshold, the input information flow requires too much time to vanish and hence the fading memory property is poorly satisfied. We recall the well known fact (see Section 8.2 in[29]) that the fading memory property can be realized by input-output systems generated by time-delay differential equations only when these exhibit a unique stable equilibrium.

In the context of recent successful physical realizations of RC, experimental parameters are systematically chosen so that the conditions described above are satisfied. Indeed, in[14,15] these conditions are ensured via a proper tuning of the gain of the delayed feedback function. This approach differs from the one in[17], where the conditions are met by choosing a laser injection current strictly smaller but close to the lasing threshold, as well as by using a moderate feedback, which prevents eventual self sustained external cavity mode oscillations. An additional important observation suggested by all these experimental setups is the need for a nonlinearity at the level of the input injection. In[14,15] this feature is obtained using a strong enough input signal amplitude and via the transformation associated to the nonlinear delayed feedback. In[17] the delayed feedback is linear but an external Mach-Zehnder modulator is used that implicitly provides a nonlinear transformation of the input signal as it is optically seeded through the nonlinear electro-optic modulation transfer function of the Mach-Zehnder.

We conclude by emphasizing that, even though optimal performance is attained when working in a statistically stationary regime around a stable equilibrium for the specific time-delay RCs that we are using, this is not a general feature that applies to all RCs or other network based computational paradigms. Indeed, recent works[30] prove the existence of RCs (random Boolean networks in the case of[30]) that exhibit optimal performance when operating in a chaotic regime; more generally, this fact has also been observed in certain recurrent neural networks[31].

Stability analysis of the time-delay reservoir. Due to the central role played by stability in our discussion, we now carefully analyze various sufficient conditions that ensure that the RC is functioning in a stable regime. All the statements that follow are carefully proved in the Supplementary Material section. Consider first an equilibrium $x_0 \in \mathbb{R}$ of the continuous time model (1) working in autonomous regime, that is, we set $I(t) = 0$. It can be shown using a Lyapunov-Krasovskiy-type analysis[32,33] that the asymptotic stability of x_0 is guaranteed whenever there exists an $\varepsilon > 0$ and a constant $|k_\varepsilon| < 1$ such that either

1. $f(x + x_0, 0, \boldsymbol{\theta}) \leq k_\varepsilon x + x_0$ for all $x \in (-\varepsilon, \varepsilon)$, or
2. $\frac{f(x+x_0,0,\boldsymbol{\theta})-x_0}{x} \leq k_\varepsilon$ for all $x \in (-\varepsilon, \varepsilon)$.

The first condition can be used to prove the stability of equilibria exhibited by TDRs created using concave (but not necessarily differentiable) nonlinear kernels. As to the second one, it shows that if f is differentiable at x_0 then this point is stable as long as $|\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1$, with $|\partial_x f(x_0, 0, \boldsymbol{\theta})$ the first derivative of the nonlinear kernel f in (1) with respect to the first argument at the point $(x_0, 0, \boldsymbol{\theta})$.

The stability study can also be carried out by working with the discrete-time approximation (5) of the TDR which is determined by the reservoir map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$. More specifically, it can be shown that $x_0 \in \mathbb{R}$ is an equilibrium of (1) if and only if $\mathbf{x}_0 := (x_0, \ldots, x_0)^T \in \mathbb{R}^N$ is a fixed point of (5). The asymptotic stability of this fixed point is ensured whenever the linearization $D_{\mathbf{x}}F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$, which is a $N \times N$ matrix that will be referred to as the **connectivity matrix**, has a spectral radius smaller than one. Since it is not possible to compute the eigenvalues of $D_{\mathbf{x}}F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ for an arbitrary number of neurons N, we are hence obliged to proceed by finding estimations for the Cauchy bound[34] of its characteristic polynomial or by bounding the spectral radius $\rho(D_{\mathbf{x}}F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}))$ using either a matrix norm or the Gershgorin discs[35]. An in-depth study of all these options showed that it is the use of the maximum row sum matrix norm $|||\cdot|||_\infty$ that yields the best stability bounds via the following statement:

$$\rho(D_{\mathbf{x}}F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})) \leq |||D_{\mathbf{x}}F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})|||_{\infty} 1 < \text{if and only if} \quad |\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1. \tag{6}$$

Notice that this remarkable result puts together the stability conditions for the continuous and discrete time systems.

As an example of application of these results, consider the Mackey-Glass nonlinear kernel[21]

$$f(x, I, \boldsymbol{\theta}) = \frac{\eta(x + \gamma I)}{1 + (x + \gamma I)^p}, \tag{7}$$

where the parameter $\boldsymbol{\theta} := (\gamma, \eta, p)$ is a three tuple of real values; γ is usually referred to as the **input gain** and η the **feedback gain**. When this prescription is used in (1) in the autonomous regime, that is, $I(t) = 0$, the associated dynamical system exhibits two families of equilibria x_0 parametrized by η, namely, $x_0 = 0$ and the roots of $x_0^p = \eta - 1$. For example, in the case $p = 2$, two distinct cases arise: when $\eta < 1$ there is a unique equilibrium at the origin which is stable as long as $\eta \in (-1, 1)$. When $\eta > 1$ two other equilibria appear at $x_0 = \pm(\eta - 1)^{1/2}$ which are stable whenever $\eta < 3$. These statements are proved in Corollary D.6 of the Supplementary Material. Analogous statements for the Ikeda kernel[22] $f(x, I, \boldsymbol{\theta}) = \eta \sin^2(x + \gamma I + \phi)$, $\boldsymbol{\theta} := (\eta, \gamma, \phi)$ can be found in Corollary D.7 of the Supplementary Material. A particularly convenient sufficient condition is $|\eta| \leq 1$ that simultaneously ensures stability and unimodality (existence of a single stable equilibrium).

Empirical evidence. In order to confirm these theoretical and experimental arguments, we have carried out several numerical simulations in which we studied the RC performance in terms of the dynamical regime of the reservoir at the time of carrying out various nonlinear memory tasks. More specifically, we construct a reservoir using the Ikeda nonlinear kernel with $N = 20$, $d = 0.2581$, $\eta = 1.2443$, $\gamma = 1.4762$, and $\phi = 0.1161$. The equilibria of the associated autonomous system are given by the points x_0 where the curves $y = x$ and $y = \eta \sin^2(x + \phi)$ intersect. With this parameter values, intersections take place at $x_0 = 0.0244$, $x_0 = 0.9075$, and $x_0 = 1.063$, which makes multi modality possible. As it can be shown with the results in the Supplementary Material section (see Corollary D.7), the first and the third equilibria are stable. In order to verify that the optimal performance is obtained when the RC operates in the neighborhood of a stable equilibrium, we study the normalized mean square error (NMSE) exhibited by a TDR initialized at $x_0 = 0.0244$ when we present to it a quadratic memory task. More specifically, we inject in a TDR an independent and identically normally distributed signal $z(t)$ with mean zero and variance 10^{-4} and we then train a linear readout $\mathbf{W}_{\text{out}}$ (obtained with a ridge penalization of $\lambda = 10^{-15}$) in order to recover the quadratic function $z(t-1)^2 + z(t-2)^2 + z(t-3)^2$ out of the reservoir output. The top left panel in Fig. 2 shows how the NMSE behaves as a function of the mean and the variance of the input mask **c**. It is clear that by modifying any of these two parameters we control how far the reservoir dynamics separates from the stable equilibrium, which we quantitatively evaluate in the two bottom panels by representing the RC performance in terms of the mean and the variance of the resulting reservoir output. Both panels depict how the injection of a signal slightly shifted in mean or with a sufficiently high variance results in reservoir outputs that separate from the stable equilibrium and in a severely degraded performance. An important factor in this deterioration seems to be the multi modality, that is, if the shifting in mean or the input signal variance are large enough then the reservoir output visits the stability basin of the other stable point placed at $x_0 = 1.063$; in the top right and bottom panels we have marked with red color the values for which bimodality has occurred so that the negative effect of this phenomenon is noticeable. In the Supplementary Material section we illustrate how the behavior that we just described is robust with respect to the choice of nonlinear kernel and is similar when the experiment is carried out using the Mackey-Glass function.

The approximating model and the nonlinear memory capacity of the reservoir computer. The findings just presented have major consequences in the theoretical tools available for the evaluation of the RC performance. Indeed, since we now know that optimal operation is attained when the TDR functions in a unimodal fashion around an asymptotically stable steady state, we can approximate it by its partial linearization with respect to the delayed self feedback term at that point and keeping the nonlinearity for the input injection. For statistically independent input signals of the type used to compute nonlinear memory capacities of the type introduced in[24], this approximation allows us to visualize the TDR as a N-dimensional (N is the number of neurons) vector autoregressive stochastic process of order one[36] (we denote it as VAR(1)) for which the value of the associated nonlinear memory capacities can be explicitly computed. As we elaborate later on in the discussion, the quality of this approximation at the time of evaluating the memory capacities of the original system is excellent and the resulting function can be hence used for RC optimization purposes regarding the nonlinear kernel parameter values $\boldsymbol{\theta}$ and the input mask **c**.

Consider a stable equilibrium $x_0 \in \mathbb{R}$ of the autonomous system associated to (1) or, equivalently, a stable fixed point of (5) of the form $\mathbf{x}_0 := (x_0, \ldots, x_0)^{\top} \in \mathbb{R}^N$. If we approximate (5) by its partial line-

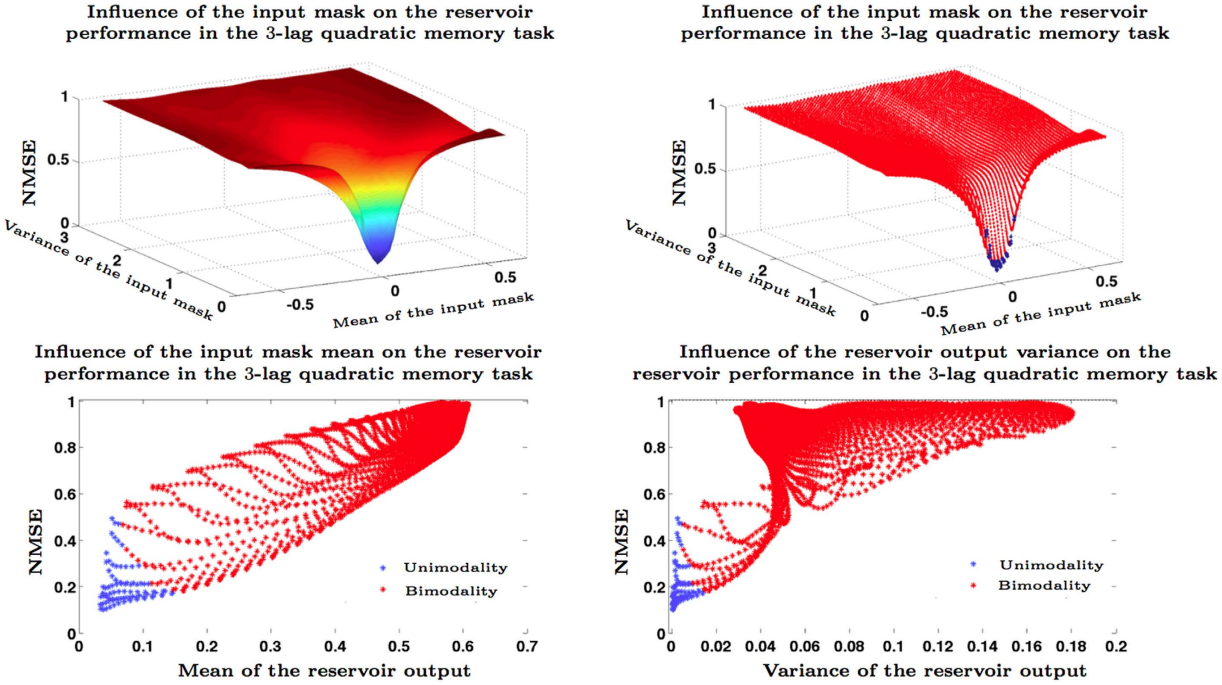

Figure 2. Behavior of the reservoir performance in a quadratic memory task as a function of the mean and the variance of the input mask. The modification of any of these two parameters influences how the reservoir dynamics separates from the stable equilibrium. The top panels show how the performance degrades very quickly as soon as the mean and the variance of the input mask (and hence of the input signal) separate from zero. The bottom panels depict the reservoir performance as a function of the various output means and variances obtained when changing the input means and variances. In the top right and bottom panels we have indicated with red markers the cases in which the reservoir visits the stability basin of a contiguous stable equilibrium hence showing how unimodality is associated to optimal performance.

arization at $\mathbf{x}_0$ with respect to the delayed self feedback and by the R-order Taylor series expansion of the functional that describes the signal injection, we obtain an expression of the form:

$$\mathbf{x}(t) = F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) + A(\mathbf{x}_0, \boldsymbol{\theta})(\mathbf{x}(t-1) - \mathbf{x}_0) + \boldsymbol{\varepsilon}(t), \tag{8}$$

where $A(\mathbf{x}_0, \boldsymbol{\theta}) := D_{\mathbf{x}} F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ is the linear connectivity matrix and $\varepsilon(t)$ is given by:

$$\varepsilon(t) = (1 - e^{-\xi})\left(q_R(z(t), c_1), q_R(z(t), c_1, c_2), \ldots, q_R(z(t), c_1, \ldots, c_N)\right)^{\top}, \tag{9}$$

with

$$q_R(z(t), c_1, \ldots, c_r) := \sum_{i=1}^{R} \frac{z(t)^i}{i!} (\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta}) \sum_{j=1}^{r} e^{-(r-j)\xi} c_j^i, \tag{10}$$

and $(\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta})$ is the ith order partial derivative of the nonlinear kernel f with respect to the second argument $I(t)$, evaluated at the point $(x_0, 0, \boldsymbol{\theta})$.

If we now use as input signal $z(t)$ independent and identically distributed random variables with mean 0 and variance σ_z^2 (we denote it by $\{z(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(0, \sigma_z^2)$) then the recursion (8) makes the reservoir layer dynamics $\{\mathbf{x}(t)\}_{t\in\mathbb{Z}}$ into a discrete time random process that, as we show in what follows, is the solution of a N-dimensional vector autoregressive model of order 1 (VAR(1)). Indeed, it is easy to see that the assumption $\{z(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(0, \sigma_z^2)$ implies that $\{\mathbf{I}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\mathbf{0}_N, \Sigma_I)$, with $\Sigma_I := \sigma_z^2 \mathbf{c}^{\top}\mathbf{c}$, and that $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{Z}}$ is a family of N-dimensional independent and identically distributed random variables with mean $\boldsymbol{\mu}_{\varepsilon}$ and covariance matrix Σ_{ε} given by the following expressions:

$$\boldsymbol{\mu}_{\varepsilon} = \mathrm{E}[\boldsymbol{\varepsilon}(t)] = (1 - e^{-\xi})\left(q_R(\mu_z, c_1), q_R(\mu_z, c_1, c_2), \ldots, q_R(\mu_z, c_1, \ldots, c_N)\right)^{\top}, \tag{11}$$

where the polynomial q_R is the same as in (10) and where we use the convention that the powers $\mu_z^i := \mathrm{E}[z(t)^i]$, for any $i \in \{1, \ldots, R\}$ and with $\mathrm{E}[\cdot]$ denoting the mathematical expectation. Additionally, $\Sigma_\varepsilon := \mathrm{E}[(\boldsymbol{\varepsilon}(t) - \boldsymbol{\mu}_\varepsilon)(\boldsymbol{\varepsilon}(t) - \boldsymbol{\mu}_\varepsilon)^\top]$ has entries determined by the relation:

$$(\Sigma_\varepsilon)_{ij} = (1 - e^{-\xi})^2 \Big(\big(q_R(\cdot, c_1, \ldots, c_i) \cdot q_R(\cdot, c_1, \ldots, c_j)\big)(\mu_z) - q_R(\mu_z, c_1, \ldots, c_i) q_R(\mu_z, c_1, \ldots, c_j) \Big), \tag{12}$$

where the first summand stands for the multiplication of the polynomials $q_R(\cdot, c_1, \ldots, c_i)$ and $q_R(\cdot, c_1, \ldots, c_j)$ and the subsequent evaluation of the resulting polynomial at μ_z, and the second one is made out of the multiplication of the evaluation of the two polynomials.

Using these observations, we can consider (8) as the prescription of a VAR(1) model driven by the independent noise $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{Z}}$. If the nonlinear kernel f satisfies the generic condition that the polynomial in u given by $\det(\mathbb{I}_N - A(\mathbf{x}_0, \boldsymbol{\theta})u)$, does not have roots in and on the complex unit circle, then (8) has a second order stationary solution[36] $\{\mathbf{x}(t)\}_{t\in\mathbb{Z}}$ with time-independent mean given by

$$\boldsymbol{\mu}_x = \mathrm{E}[\mathbf{x}(t)] = (\mathbb{I}_N - A(\mathbf{x}_0, \boldsymbol{\theta}))^{-1}(F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) - A(\mathbf{x}_0, \boldsymbol{\theta})\mathbf{x}_0 + \boldsymbol{\mu}_\varepsilon) \tag{13}$$

and an also time independent autocovariance function $\Gamma(k) := \mathrm{E}[(\mathbf{x}(t) - \boldsymbol{\mu}_x)(\mathbf{x}(t-k) - \boldsymbol{\mu}_x)^\top]$, $k \in \mathbb{Z}$, recursively determined the Yule-Walker equations (see[36] for a detailed presentation). Indeed, $\Gamma(0)$ is given by the vectorized equality:

$$\mathrm{vec}(\Gamma(0)) = (\mathbb{I}_{N^2} - A(\mathbf{x}_0, \boldsymbol{\theta}) \otimes A(\mathbf{x}_0, \boldsymbol{\theta}))^{-1}\mathrm{vec}(\Sigma_\varepsilon), \tag{14}$$

which determines the higher order autocovariances via the relation $\Gamma(k) = A(\mathbf{x}_0, \boldsymbol{\theta})\Gamma(k-1)$ and the identity $\Gamma(-k) = \Gamma(k)^\top$. As we explain in the following paragraphs, the moments (11), (13), and (14) are all that is needed in order to characterize the memory capacities of the RC.

A ***h*-lag memory task** is determined by a (in general nonlinear) function $H : \mathbb{R}^{h+1} \to \mathbb{R}$ that is used to generate a one-dimensional signal $y(t) := H(z(t), z(t-1), \ldots, z(t-h))$ out of the reservoir input. Given a TDR computer, the optimal linear readout $\mathbf{W}_{\mathrm{out}}$ adapted to the memory task H is given by the solution of a ridge linear regression problem with regularization parameter $\lambda \in \mathbb{R}$ (usually tuned during the training phase via cross-validation) in which the covariates are the neuron values corresponding to the reservoir output and the explained variables are the values $\{y(t)\}$ of the memory task function. The H-memory capacity $C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda)$ of the TDR computer under consideration characterized by a nonlinear kernel f with parameters $\boldsymbol{\theta}$, an input mask $\mathbf{c}$, and a regularizing ridge parameter λ is defined as one minus the normalized mean square error committed at the time of accomplishing the memory task H. When the reservoir is approximated by a VAR(1) process, then the corresponding ***H*-memory capacity** is given by

$$C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda) = \frac{\mathrm{Cov}(y(t), \mathbf{x}(t))^\top (\Gamma(0) + \lambda\mathbb{I}_N)^{-1}(\Gamma(0) + 2\lambda\mathbb{I}_N)(\Gamma(0) + \lambda\mathbb{I}_N)^{-1}\mathrm{Cov}(y(t), \mathbf{x}(t))}{\mathrm{var}(y(t))} \tag{15}$$

The developments leading to this expression are contained in the Supplementary Material section. It is easy to show that:

$$0 \leq C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda) \leq 1. \tag{16}$$

Notice that in order to evaluate (15) for a specific memory task, only $\mathrm{Cov}(y(t), \mathbf{x}(t))$ and $\mathrm{var}(y(t))$ need to be computed since the autocovariance $\Gamma(0)$ is fully determined by (14) once the reservoir and the equilibrium $\mathbf{x}_0$ around which we operate have been chosen. As an example, we provide the expressions corresponding to the two most basic information processing routines, namely the linear and the quadratic memory tasks. Details on how to obtain the following equalities are contained in the Supplementary Material section.

The *h*-lag linear memory task. Linear memory tasks are those associated to linear task functions $H : \mathbb{R}^{h+1} \to \mathbb{R}$, that is, if we denote $\mathbf{z}^h(t) := (z(t), z(t-1), \ldots, z(t-h))^\top$ and $\mathbf{L} \in \mathbb{R}^{h+1}$, we set $H(\mathbf{z}^h(t)) := \mathbf{L}^\top \mathbf{z}^h(t)$. Various computations included in the Supplementary Material section using the so called MA(∞) representation of the VAR(1) process show that $\mathrm{var}(y(t)) = \sigma_z^2 \|\mathbf{L}\|^2$, and $\mathrm{Cov}(y(t), x_i(t)) = (1 - e^{-\xi})\sum_{j=1}^{h+1}\sum_{s=1}^{N} L_j (A(\mathbf{x}_0, \boldsymbol{\theta})^{j-1})_{is} p_R(\mu_z, c_1, \ldots, c_s)$, $\quad i \in \{1, \ldots, N\}$, where the polynomial p_R on the variable x is defined by $p_R(x, c_1, \ldots, c_s) := x \cdot q_R(x, c_1, \ldots, c_s)$ and its evaluation in the previous formula follows the same convention as in (11).

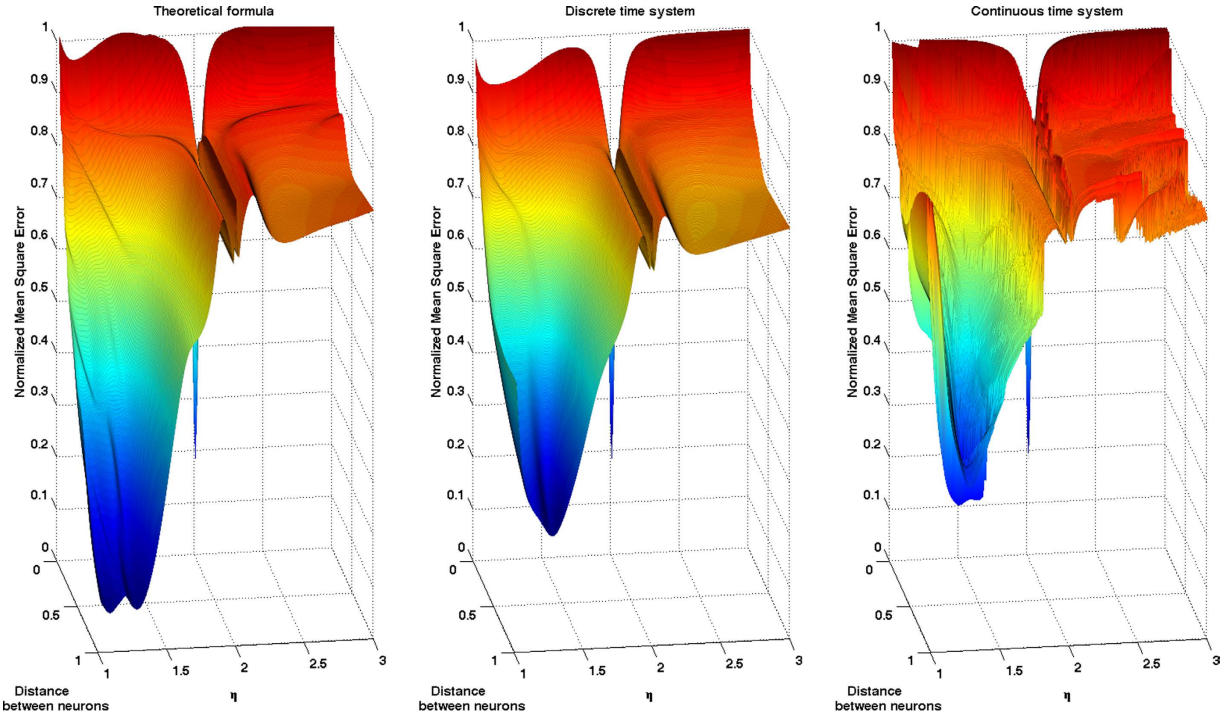

Figure 3. Error surfaces exhibited by a Mackey-Glass kernel based reservoir computer in a 6-lag quadratic memory task, as a function of the distance between neurons and the parameter η. The points in the surfaces of the middle and right panels are the result of Monte Carlo evaluations of the NMSE exhibited by the discrete and continuous time TDRs, respectively. The left panel was constructed using the formula (15) that is obtained as a result of modeling the reservoir with an approximating VAR(1) model. The computational convenience of the formula (15) can be visualized by noticing that each point in the middle and right panels took 37 and 41 seconds, respectively, to be estimated using a computer code written down in a high level programming language running on a single 2.53 GHz Intel i5 core; the same computation using (15) in the left panel took only 1.1 seconds.

The *h*-lag quadratic memory task. In this case we use a quadratic task function of the form

$$H(\mathbf{z}^h(t)) := \mathbf{z}^h(t)^\top Q \mathbf{z}^h(t) = \sum_{i=1}^{h+1}\sum_{j=1}^{h+1} Q_{ij} z(t-i+1) z(t-j+1), \tag{17}$$

for some symmetric $h+1$-dimensional matrix Q. If we define $y(t) := H(\mathbf{z}^h(t))$, we have that $\mathrm{var}(y(t)) = \left(\mu_z^4 - \sigma_z^4\right)\sum_{i=1}^{h+1} Q_{ii}^2 + 4\sigma_z^4 \sum_{i=1}^{h+1}\sum_{j>i}^{h+1} Q_{ij}^2$, and

$$\mathrm{Cov}(y(t), x_i(t)) = (1 - e^{-\xi}) \sum_{j=1}^{h+1}\sum_{r=1}^{N} Q_{jj} (A^{j-1})_{ir} \left(s_R(\mu_z, c_1, \ldots, c_r) - \sigma_z^2 q_R(\mu_z, c_1, \ldots, c_r)\right), \tag{18}$$

where the polynomial s_R on the variable x is defined as $s_R(x, c_1, \ldots, c_r) := x^2 \cdot q_R(x, c_1, \ldots, c_r)$.

Discussion

The possibility to approximate the TDR using a model of the type (8) opens the door to the theoretical treatment of many RC design related questions that so far were addressed using a trial and error approach. In particular, the availability of a closed form formula of the type (15) for the memory capacity of the RC is extremely convenient to determine the optimal reservoir architecture to carry out a given task. Nevertheless, it is obviously very important to assess the quality of the VAR(1) approximation underlying it and of the consequences that result from it. Indeed, we recall that the expression (15) was obtained via the partial linearization of the reservoir at a stable equilibrium in which it is initialized and kept in stationary operation. Despite the good theoretical and experimental reasons to proceed in this fashion provided above, we have confirmed their pertinence by explicitly comparing the reservoir memory capacity surfaces obtained empirically with those coming from the analytical expression (15). We have carried this comparison out for various tasks and have constructed the memory capacity surfaces as a function of different design parameters.

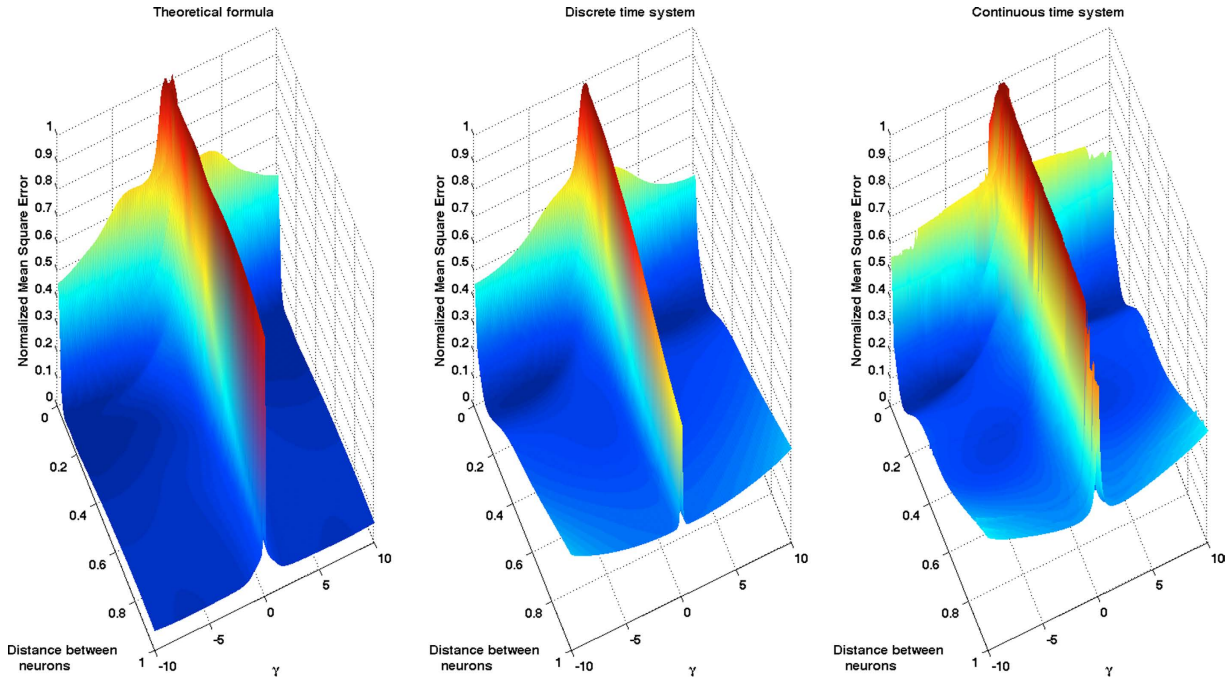

Figure 4. Error surfaces exhibited by a Mackey-Glass kernel based reservoir computer in a 3-lag quadratic memory task, as a function of the distance between neurons and the parameter γ. The points in the surfaces of the middle and right panels are the result of Monte Carlo evaluations of the NMSE exhibited by the discrete and continuous time TDRs, respectively. The left panel was constructed using the formula (15) that is obtained as a result of modeling the reservoir with an approximating VAR(1) model.

We first consider a RC constructed using the Mackey-Glass nonlinear kernel (7) with $p=2$, $\gamma=0.796$, and twenty neurons. We present to it the 6-lag quadratic memory task H corresponding to choosing in (17) a seven dimensional diagonal matrix Q with the diagonal entries given by the vector (0, 1, 1, 1, 1, 1, 1). The first element, corresponding to the 0-lag memory (quadratic nowcasting), is set to zero in order to keep the difficulty of the task high enough. We then vary the value d of the distance between neurons between 0 and 1 and the feedback gain parameter η between 1 and 3. As we already discussed, the TDR in autonomous regime exhibits for these parameter values two stable equilibria placed at $\pm(\eta-1)^{1/2}$; for this experiment we will always work with the positive equilibria by initializing the TDRs at those points. Fig. 3 represents the normalized mean square error (NMSE) surfaces (which amounts to one minus the capacity) obtained using three different approaches. The left panel was obtained using the formula (15) constructed with an eight-order Taylor expansion of the nonlinear kernel on the signal input ($R=8$ in (9)). The points in the surfaces of the middle and right panels are the result of Monte Carlo evaluations (using 50,000 occurrences each) of the NMSE exhibited by the discrete and continuous time TDRs, respectively. The time-evolution of the time-delay differential equation (continuous time model) was simulated using a Runge-Kutta fourth-order method with a discretization step equal to $d/5$. A quick inspection of Fig. 3 reveals the ability of (15) to accurately capture most of the details of the error surface and, most importantly, the location in parameter space where optimal performance is attained; it is very easy to visualize in this particular example how sensitive the magnitude of the error and the corresponding memory capacity are to the choice of parameters and how small in size the region in parameter space associated with acceptable operation performance may be.

In order to show that these statements are robust with respect to the choice of task and varying parameters, we have carried out a similar experiment with a RC in which we fix the feedback gain $\eta_0=1.0781$ and we vary the input gain γ and the distance between neurons d. The quadratic memory task is reduced this time to 3-lags. We emphasize that in this setup the stable operation point is always the same and equal to $(\eta_0-1)^{1/2}$. Fig. 4 shows how the performance of the memory capacity estimate (15) at the time of capturing the optimal parameter region is in this situation comparable to the results obtained for the 6-lag quadratic memory task represented in Fig. 3. We also point out that in this case there is a lower variability of the performance which, in our opinion, has to do with the fact that modifying the parameter γ adjusts the input gain but leaves unchanged the operation point. Additionally, the moderate difficulty of the task makes possible attaining lower optimal error rates with the same number of neurons. In order to ensure the robustness of these results with respect to the choice of nonlinear

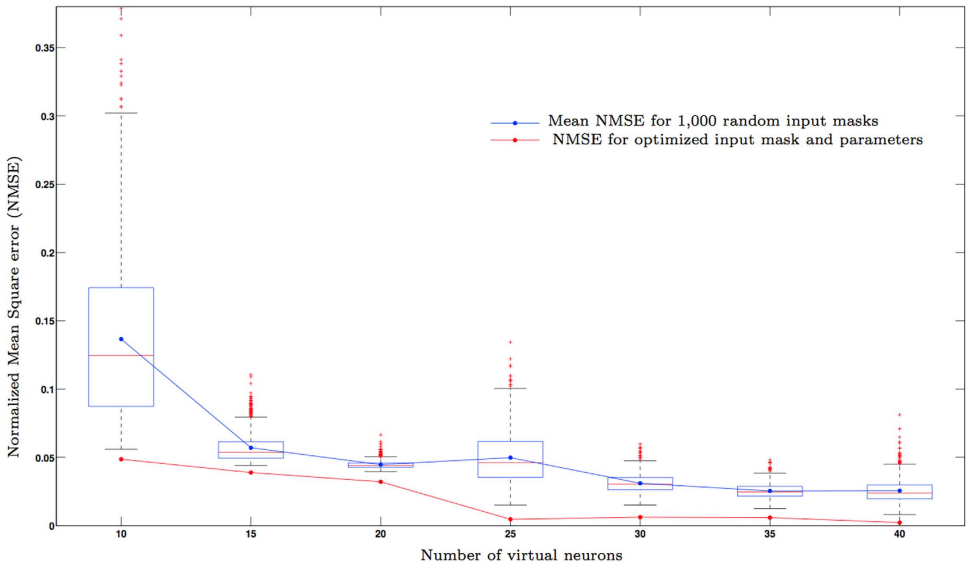

Figure 5. Influence of the mask optimization on the reservoir performance in the 3-lag quadratic memory task. The red line links the points that indicate the error committed by a RC with optimized parameters and mask. The box plots give information about the distribution of performances obtained with 1,000 input masks randomly picked (only reservoir parameters have been optimized). As it is customary, on each box, the central mark is the median and the edges of the box are the 25th and 75th percentiles (q_1 and q_3, respectively). The whiskers extend to the most extreme data points not considered outliers and outliers are plotted individually using red crosses. Points are drawn as outliers if they are larger than $q_3 + 1.5(q_3 - q_1)$ or smaller than $q_1 - 1.5(q_3 - q_1)$. The blue line links the points that indicate the mean NMSE committed when using the 1,000 different randomly picked masks.

kernel, we have included in the Supplementary Material section the results of a similar experiment carried out using the Ikeda prescription.

Once the adequacy of the memory capacity evaluation formula (15) has been established, we can use this result to investigate the influence of other architecture parameters in the reservoir performance. In Fig. 5 we depict the results of an experiment where we study the influence of the choice of input mask $\mathbf{c}$ in the performance of a Mackey-Glass kernel based reservoir in a 3-lag quadratic memory task. The figure shows, for each number of neurons, the performance obtained by a RC in which the reservoir parameters $\boldsymbol{\theta}$ and the input mask $\mathbf{c}$ have been chosen so that the memory capacity $C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda)$ in (15) is maximized; we have subsequently kept the optimal parameters $\boldsymbol{\theta}$ and we have randomly constructed one thousand input masks $\mathbf{c}$ with entries belonging to the interval $[-3, 3]$. The box plots in Fig. 5 give an idea of the distribution of the degraded performances with respect to the optimal mask for different numbers of virtual neurons.

In conclusion, the construction of approximating models for the reservoir as well as the availability of performance evaluation formulas like (15) based on it, constitute extremely valuable analytical tools whose existence should prove very beneficial in the fast and efficient extension and customization of RC type techniques to tasks far more sophisticated than the ones we considered in this paper. This specific point is the subject of ongoing research on which we will report in a forthcoming publication.

References

1. Crutchfield, J. P., Ditto, W. L. & Sinha, S. Introduction to focus issue: intrinsic and designed computation: information processing in dynamical systems-beyond the digital hegemony. *Chaos (Woodbury, N.Y.)* **20,** 037101 (2010).
2. Caulfield, H. J. & Dolev, S. Why future supercomputing requires optics. *Nature Photonics* **4,** 261–263 (2010).
3. Woods, D. & Naughton, T. J. Optical computing: Photonic neural networks. *Nature Physics* **8,** 257–259 (2012).
4. Jaeger, H. The 'echo state' approach to analysing and training recurrent neural networks. Tech. Rep., German National Research Center for Information Technology (2001).
5. Jaeger, H. & Haas, H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* **304,** 78–80 (2004).
6. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation* **14,** 2531–2560 (2002).
7. Maass, W. Liquid state machines: motivation, theory, and applications. In Barry Cooper, S. S. & Sorbi, A. (eds.) *Computability In Context: Computation and Logic in the Real World* chap. 8, 275–296 (2011).
8. Crook, N. Nonlinear transient computation. *Neurocomputing* **70,** 1167–1176 (2007).
9. Verstraeten, D., Schrauwen, B., DHaene, M. & Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Networks* **20,** 391–403 (2007).

10. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* **3,** 127–149 (2009).
11. Rodan, A. & Tino, P. Minimum complexity echo state network. *IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council* **22,** 131–44 (2011).
12. Gutiérrez, J. M., San-Martín, D., Ortín, S. & Pesquera, L. Simple reservoirs with chain topology based on a single time-delay nonlinear node. In *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 13–18 (2012).
13. Jaeger, H., Lukoševičius, M., Popovici, D. & Siewert, U. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks* **20,** 335–352 (2007).
14. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nature Communications* **2,** 468 (2011).
15. Larger, L. *et al.* Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Optics Express* **20,** 3241 (2012).
16. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Scientific reports* **2,** 287 (2012).
17. Brunner, D., Soriano, M. C., Mirasso, C. R. & Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications* **4** (2013).
18. Atiya, A. F. & Parlos, A. G. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council* **11,** 697–709 (2000).
19. Grigoryeva, L., Henriques, J., Larger, L. & Ortega, J.-P. Stochastic time series forecasting using time-delay reservoir computers: performance and universality. *Neural Networks* **55,** 59–71 (2014).
20. Ortin, S., Pesquera, L. & Gutiérrez, J. M. Memory and nonlinear mapping in reservoir computing with two uncoupled nonlinear delay nodes. In *Proceedings of the European Conference on Complex Systems* 895–899 (2012).
21. Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197,** 287–289 (1977).
22. Ikeda, K. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications* **30,** 257–261 (1979).
23. Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **70,** 489–501 (2006). URL http://www.sciencedirect.com/science/article/pii/S0925231206000385.
24. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Scientific reports* **2** (2012).
25. Jaeger, H. Short term memory in echo state networks. *Fraunhofer Institute for Autonomous Intelligent Systems. Technical Report*. **152** (2002).
26. White, O., Lee, D. & Sompolinsky, H. Short-Term Memory in Orthogonal Neural Networks. *Physical Review Letters* **92,** 148102 (2004).
27. Ganguli, S., Huh, D. & Sompolinsky, H. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America* **105,** 18970–5 (2008).
28. Hermans, M. & Schrauwen, B. Memory in linear recurrent neural networks in continuous time. *Neural networks : the official journal of the International Neural Network Society* **23,** 341–55 (2010).
29. Boyd, S. & Chua, L. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems* **32,** 1150–1161 (1985).
30. Snyder, D., Goudarzi, A. & Teuscher, C. Computational capabilities of random automata networks for reservoir computing. *Physical Review E* **87,** 042808 (2013). URL http://link.aps.org/doi/10.1103/PhysRevE.87.042808.
31. Büsing, L., Schrauwen, B. & Legenstein, R. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation* **22,** 1272–311 (2010). URL http://www.mitpressjournals.org/doi/abs/10.1162/neco.2009.01-09-947#.VUiMY1p16ao.
32. Krasovskiy, N. N. *Stability of Motion* (Stanford University Press, 1963).
33. Wu, M., He, Y. & She, J.-H. *Stability Analysis and Robust Control of Time-Delay Systems* (Springer, 2010).
34. Rahman, Q. I. & Schmeisser, G. *Analytic Theory of Polynomials* (Clarendon Press, Oxford, 2002).
35. Horn, R. A. & Johnson, C. R. *Matrix Analysis* (Cambridge University Press, 2013), second edn.
36. Lütkepohl, H. *New Introduction to Multiple Time Series Analysis* (Springer-Verlag, Berlin, 2005).

Acknowledgments

We acknowledge partial financial support of the Région de Franche-Comté (Convention 2013C-5493), the ANR "BIPHOPROC" project (ANR-14-OHRI-0002-02), the European project PHOCUS (FP7 Grant No. 240763), the Labex ACTION program (Contract No. ANR-11-LABX-01- 01), and Deployment S.L. L.G. acknowledges financial support from the Faculty for the Future Program of the Schlumberger Foundation.

Author Contributions

L.G., J.H., J.-P.O. and L.L. contributed to all sections of the paper and the supplementary material. All authors reviewed the manuscript.

Additional Information

Supplementary information accompanies this paper at http://www.nature.com/srep

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Grigoryeva, L. *et al.* Optimal nonlinear information processing capacity in delay-based reservoir computers. *Sci. Rep.* **5**, 12858; doi: 10.1038/srep12858 (2015).