Performance boost of time-delay reservoir computing by non-resonant clock cycle

Florian Stelzer [a,b,*], André Röhm [c,d], Kathy Lüdge [c], Serhiy Yanchuk [a]

[a] *Institute of Mathematics, Technische Universität Berlin, D-10623, Germany*
[b] *Department of Mathematics, Humboldt-Universität zu Berlin, D-12489, Germany*
[c] *Institute of Theoretical Physics, Technische Universität Berlin, D-10623, Germany*
[d] *Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (CSIC-UIB), Campus Universitat de les Illes Balears, E-07122 Palma de Mallorca, Spain*

A B S T R A C T

The time-delay-based reservoir computing setup has seen tremendous success in both experiment and simulation. It allows for the construction of large neuromorphic computing systems with only few components. However, until now the interplay of the different timescales has not been investigated thoroughly. In this manuscript, we investigate the effects of a mismatch between the time-delay and the clock cycle for a general model. Typically, these two time scales are considered to be equal. Here we show that the case of equal or resonant time-delay and clock cycle could be actively detrimental and leads to an increase of the approximation error of the reservoir. In particular, we can show that non-resonant ratios of these time scales have maximal memory capacities. We achieve this by translating the periodically driven delay-dynamical system into an equivalent network. Networks that originate from a system with resonant delay-times and clock cycles fail to utilize all of their degrees of freedom, which causes the degradation of their performance.

1. Introduction

Reservoir computing is a machine learning method, which was introduced independently by both Jaeger (2001) as a mathematical framework and by Maass, Natschläger, and Markam (2002) from a biologically inspired background. It fundamentally differs from many other machine learning concepts and is particularly interesting due to its easy integration into hardware, especially photonics (Brunner et al., 2018; Van der Sande, Brunner, & Soriano, 2017). With the help of the reservoir computing paradigm, the naturally occurring computational power of almost any dynamical or physical system can be exploited. It is particularly valuable for solving the class of time-dependent problems, which is usually more difficult to address with artificial neural network-based approaches. A time-dependent problem requires to estimate a target signal $(y(t))_{t \in \mathbb{T}}$ which depends non-trivially on an input signal $(u(t))_{t \in \mathbb{T}}$, the set of times $\mathbb{T}$ may be continuous or discrete. This class of problems contains, in particular, speech recognition or time series prediction (Jaeger & Haas, 2004; Verstraeten, Schrauwen, & Stroobandt, 2006; Verstraeten,

Schrauwen, Stroobandt, & Van Campenhout, 2005), and also has great promise for error correction in optical data transmission (Argyris, Bueno, & Fischer, 2018). Furthermore, reservoir computing can be used to study fundamental properties of dynamical systems in a completely novel way (Pathak, Hunt, Girvan, Lu, & Ott, 2018), enabling new ways of characterizing physical systems.

The main idea of reservoir computing is simple, yet powerful: A dynamical system, the *reservoir*, is driven by an input $u(t)$. The state of the reservoir is described by a variable $x(t)$, which can be high- or even infinite-dimensional. A linear *readout mapping* $x(t) \mapsto \hat{y}(t)$ provides an output. While the parameters of the reservoir itself remain fixed at all times, the coefficients of the linear mapping $x(t) \mapsto \hat{y}(t)$ are subject to adaptation, i.e. the readout can be trained.

Reservoir computing is a supervised machine learning method. An input $u(t)$ and the corresponding target function $y(t)$ are given as a training example. Then optimal output weights, i.e. coefficients of the linear mapping $x(t) \mapsto \hat{y}(t)$, are determined such that $\hat{y}(t)$ approximates the target $y(t)$. This is analogous to a conventional artificial neural network where only the last layer is trained. The goal of this procedure is to approximate the mapping $u(t) \mapsto y(t)$ via $u(t) \mapsto x(t) \mapsto \hat{y}(t)$ such that it not only reproduces the target function for the given training input but also provides meaningful results for other, in certain sense similar, inputs. From a nonlinear dynamics perspective, the

* Corresponding author at: Department of Mathematics, Humboldt-Universität zu Berlin, D-12489, Germany.

E-mail addresses: stelzer@math.tu-berlin.de (F. Stelzer), aroehm@mailbox.tu-berlin.de (A. Röhm), kathy.luedge@tu-berlin.de (K. Lüdge), yanchuk@math.tu-berlin.de (S. Yanchuk).

readout mapping $x(t) \mapsto \hat{y}(t)$ is a linear combination of different degrees of freedom of the system.

The reservoir must fulfill several criteria to exhibit good computational properties: First, it must be able to carry information of multiple past input states, i.e. have memory. References Jaeger (2002) and Lymburn et al. (2019) study how a reservoir can store information about past inputs. Second, the system should contain a non-linearity to allow for non-trivial data processing. Jaeger (2001) proposed to use a recurrent neural network with random connections as reservoir. In this case the reservoir x has a high-dimensional state space by design, as the dimension equals the number of nodes. Such systems are in general able to store a large amount of information. Moreover, the recurrent structure of the network ensures that information about past input states remains for a number of time steps and fades slowly. Jaeger compared the presence of past inputs in the state of x to echoes. For this reason he called his proposed reservoir system *echo state network*.

In recent years, the field of reservoir computing has profited from experimental approaches that use a continuous time-delay dynamical system as reservoir (Appeltant et al., 2011). While hybrid network-delay systems have also been proposed (Röhm & Lüdge, 2018), typically, only a single dynamical nonlinear system is employed and connected to a long delay loop; i.e. as opposed to a network-based approach only a single active element is needed for a delay-based reservoir. Here, the complexity is induced by the inherent large phase space dimension of the dynamical system with time-delay (Erneux, 2009; Erneux, Javaloyes, Wolfrum, & Yanchuk, 2017; Yanchuk & Giacomelli, 2017). The main advantage of using a delay system over Jaeger's echo state approach (Jaeger, 2001) is that one can physically implement the delay system with analogue hardware at relatively low costs—either electronically (Appeltant et al., 2011) or even optically (Brunner, Soriano, Mirasso, & Fischer, 2011; Larger et al., 2012).

Two main time scales exist in such delay-based reservoir systems: the delay-time τ given by the physical length of the feedback line, and a clock cycle τ' given by the input speed. In this paper, we show that the non-trivial cases of mismatched delay-time and clock cycle possess better reservoir computing properties. We explain this by studying the corresponding equivalent network, where we can show that non-resonant ratios of τ' to τ have maximal memory capacities.

The paper is organized as follows: In Section 2 the general model of a reservoir computer based on a single delay-differential system is introduced. We refer to this method as *time-delay reservoir computing* (TDRC). Section 3 shows numerical simulations and the effect of mismatching clock cycles and delay-times. Section 4 derives a representation of the TDRC system with mismatching clock cycle and delay-time as an equivalent echo state network. Section 5 presents a direct calculation of the memory capacity. Section 6 derives a semi-analytic explanation for the observed decreased memory capacity for resonant τ' to τ ratios and provides an intuitive interpretation. All results are summarized in Section 7.

2. Time-delay reservoir computing

In this section we describe the reservoir computing system based on a delay equation. Its choice is inspired by the publications (Appeltant et al., 2011; Brunner et al., 2011), where it was experimentally implemented using analogue hardware. In comparison to the general reservoir computing scheme $u(t) \mapsto x(t) \mapsto \hat{y}(t)$ described above, an additional 'preprocessing' step is added to transform the input u in an appropriate way before being sent to the reservoir. This is particularly necessary when

the input u is discrete and the reservoir x is time-continuous. In the following, we describe the resulting chain of transformations

$$u \overset{\text{(I)}}{\mapsto} J(t) \overset{\text{(II)}}{\mapsto} x(t) \overset{\text{(III)}}{\mapsto} \hat{y}(t) \tag{1}$$

in detail.

2.1. Step (I): preprocessing of the input

Since the reservoir is implemented with the physical experiment in mind (e.g. semiconductor laser), its state variable $x(t)$ is time-continuous. However, the input data is discrete in typical applications of TDRC (Appeltant et al., 2011; Brunner et al., 2011; Röhm & Lüdge, 2018). For this reason the preprocessing function $u \mapsto J(t)$ translates the discrete input u into a continuous function $J(t)$.

We consider a discrete input sequence $(u(k))_{k \in \mathbb{N}_0}$, where $u(k) \in \mathbb{R}$ is one-dimensional, however, the method can be extended to multi-dimensional inputs. The important parameters that define the preprocessing are the clock cycle $\tau' > 0$, number of virtual nodes $N \in \mathbb{N}$ and the resulting time per virtual node $\theta := \tau'/N$. In Ref. Appeltant et al. (2011) the parameter value for the clock cycle was chosen $\tau' = \tau$ and the optimal values for θ are shown to be in the interval $[0.1, 1]$. Therefore, we will consider the parameter τ' of the order of the delay τ and θ within the interval $[0.1, 1]$ or even larger. In fact the exact value of θ, or correspondingly, N, does not qualitatively influence the phenomenon of the error increase by rational τ'/τ (as we show in the Appendix).

First, a function $\bar{u}(t)$ is defined as step function

$$\bar{u}(t) := u(k), \quad t \in [k\tau', (k+1)\tau'), \ k \in \mathbb{N}_0 \tag{2}$$

with step length τ'. Using the indicator function

$$\Pi_M(t) = \begin{cases} 1, & t \in M, \\ 0, & t \notin M, \end{cases} \tag{3}$$

the definition of $\bar{u}$ can be equivalently written as

$$\bar{u}(t) = \sum_{k \in \mathbb{N}_0} u(k) \Pi_{[k\tau', (k+1)\tau')}(t). \tag{4}$$

Secondly, $\bar{u}(t)$ is multiplied by the τ'-periodic mask

$$\mathcal{M}(t) = \sum_{n=1}^{N} w_n \Pi_{[(n-1)\theta, n\theta)}(t \bmod \tau'), \tag{5}$$

which is piecewise constant with step length $\theta = \tau'/N$ and values w_n. Multiple options for the choice of a mask function are compared in Ref. Kuriki, Nakayama, Takano, and Uchida (2018). The final preprocessed input signal $J(t)$ is

$$J(t) := \mathcal{M}(t)\,\bar{u}(t)$$
$$= \sum_{\substack{k \in \mathbb{N}_0 \\ 1 \le n \le N}} w_n u(k) \Pi_{[k\tau'+(n-1)\theta, k\tau'+n\theta)}(t). \tag{6}$$

It is a piecewise constant function with values

$$J_{k,n} := w_n u(k) \tag{7}$$

on the intervals $[k\tau' + (n-1)\theta, k\tau' + n\theta)$. The details of the preprocessing are illustrated in Fig. 1.

For further analysis, it is convenient to denote the 'mask'-vector W^{mask} and the input vector J_k as follows:

$$W^{\text{mask}} := (w_1, \ldots, w_N)^T, \quad J_k = W^{\text{mask}} u(k). \tag{8}$$
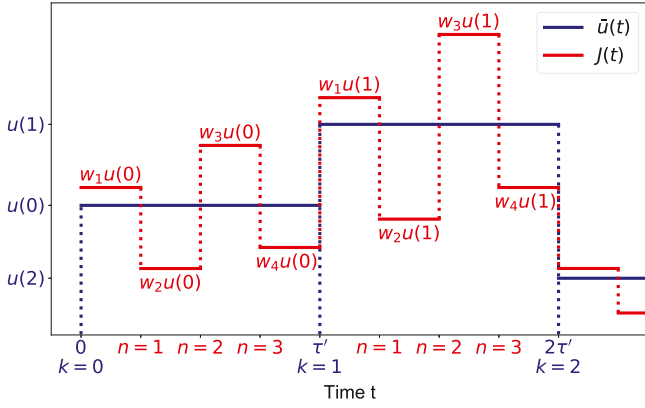
Fig. 1. Schematic representation of the preprocessing step: the discrete input sequence $u(0)$, $u(1)$, ... defines the function $\bar{u}(t)$ (blue), which is multiplied by a τ'-periodic mask function $\mathcal{M}(t)$ to obtain the preprocessed input $J(t)$ (red). Here the length of the mask vector is $N = 4$. The resulting function $J(t)$ enters the reservoir equation (9). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Step (II): reservoir

Inspired by the previous works (Appeltant et al., 2011), we study the reservoir given by the delay-differential equation

$$\frac{dx}{dt}(t) = -x(t) + f[x(t-\tau) + \gamma J(t)], \tag{9}$$

where $\tau > 0$ is the delay-time, $\gamma > 0$ is the input strength, $f : \mathbb{R} \to \mathbb{R}$ the activation function, and $J(t)$ the preprocessed input function. For a given preprocessed input $J(t)$, the reservoir variable $x(t)$ is computed by solving the delay differential equation (9) with an initial history function $x_0(s)$, $s \in [-\tau, 0]$. In order for the reservoir to be consistent, the reservoir state $x(t)$ for sufficiently large $t > 0$ should depend only on the input J and be independent on the initial state $x_0(s)$. In other words, identical reservoirs which are driven by the same input but have different initial states will approximate each other asymptotically. In the literature about reservoir computing this property is often referred to as *echo state property* or *fading memory property* (Jaeger, 2001). In the literature about dynamical systems, the phenomenon is called *generalized synchronization* (Pikovsky, Rosenblum, & Kurths, 2001; Rulkov, Sushchik, Tsimring, & Abarbanel, 1995) or *asymptotic stability* (Hale & Verduyn Lunel, 1993; Smith, 2010).

2.3. Step (III): readout

The continuous reservoir variable $x(t)$ needs to be discretized for the output. For this, the dynamical system is read out every θ time units. Because every small time window $[k\tau' + (n-1)\theta, k\tau' + n\theta)$ is fed with its own input $J_{k,n}$, these time windows are often seen as 'virtual nodes', and the whole delay system as a 'virtual network' (Appeltant et al., 2011; Schumacher, Toutounji, & Pipa, 2013b). We discretize the reservoir variable correspondingly:

$$X(k) := \begin{pmatrix} X_1(k) \\ \vdots \\ X_N(k) \end{pmatrix} := \begin{pmatrix} x((k-1)\tau' + \theta) \\ x((k-1)\tau' + 2\theta) \\ \vdots \\ x((k-1)\tau' + N\theta) \end{pmatrix}. \tag{10}$$

In fact, $X(k)$ is the vector containing the N-point discretization of the variable $x(t)$ on the interval $((k-1)\tau', k\tau']$. The output $\hat{y} = (\hat{y}(k))_{k \in \mathbb{N}_0}$ of the machine learning system is defined as

$$\hat{y}(k) = W^{out}X(k) + c, \tag{11}$$

where W^{out} is an N-dimensional row vector and $c \in \mathbb{R}$ is a scalar bias. The output weight variables W^{out} and c are to be adjusted

in the training process and are chosen by linear regression for reservoir computers (Jaeger, 2001).

3. Effect of the mismatch between delay and clock cycle times

When TDRC was first introduced, the clock cycle τ' for the preprocessing of the input mask was chosen to be equal to the delay τ (Appeltant et al., 2011; Brunner et al., 2011). In this case one can easily find an 'equivalent network' which is a discrete approximation of the reservoir system. See the supplementary material of Appeltant et al. (2011) or Ref. Schumacher, Toutounji, and Pipa (2013a) for an example. However, recent numerical observations show, that the performance may be improved if one sets $\tau' \neq \tau$. The earliest example of this can be arguably found in Ref. Paquot et al. (2012).

We use the NARMA-10 task (Atiya & Parlos, 2000), the memory capacity (MC) (Jaeger, 2002), the time series prediction task for the chaotic Lorenz system and the Santa Fe time-series prediction task (Weigend & Gershenfeld, 1993) to measure the performance of a simple TDRC to illustrate the role of the clock cycle τ'. These are typical benchmark tasks and we refer to Appendices C–E for a detailed explanation.

Moreover, we use four different functions for the activation function f in Eq. (9),

linear:	$f(x) = \alpha x,$	(12)
Mackey-Glass:	$f(x) = \alpha \dfrac{x}{1 + x^p},$	(13)
hyperbolic tangent:	$f(x) = \alpha \tanh(x),$	(14)
quadratic sin:	$f(x) = \alpha \sin^2(x + \varphi),$	(15)

where we use the parameter values $p = 1$ and $\varphi = 0.5$. The activation factor is chosen $\alpha = 0.9$ for all cases. In fact, the precise choice of the parameter α does not influence qualitatively the phenomenon which we consider, see Appendix F for more details. The other parameters are set to $\tau = 80$, $N = 50$, and $\gamma = 0.02$. For our tests we choose 100 different input weight vectors W^{mask} with independently $\mathcal{U}(-1, 1)$-distributed entries w_n. For each input weight vector we train the system with 50 000 training time steps and a Tikhonov regularization with parameter $\beta = 10^{-8}$. The training starts after an initial period of 200 000 inputs which is necessary to ensure that the reservoirs initial state does not influence the results. Only then do we record the system state for the next 20 000 inputs, with which we facilitate the training. The only exception are the simulations for the Santa Fe time-series prediction task. Since the Santa Fe dataset contains only about 9000 data points, we choose a period of 4000 initial inputs, 4000 training steps and 1000 test steps. All parameters for the simulations are summarized in Table 1. The memory capacity is evaluated up to length 300. Values that are close to 0 are not included in the sum. We estimate this overfitting threshold dynamically during the run, by using uncorrelated random target variables.

The top panel of Fig. 2 shows the results of simulations for the NARMA-10 task for the four different activation functions (12)–(15). The figure shows clear peaks of the error for certain values of the clock cycle τ'. These peaks are located close to low-order resonances with the delay-time $\tau = 80$ and fulfill the relation $a\tau \approx b\tau'$ for small $a, b \in \mathbb{N}$. In fact, the peaks are located slightly above the resonant τ' values. Note, that *tanh* and the linear function have very similar values. The lower panel of Fig. 2 depicts the memory capacity and reveals at least part of the reason for this: The total memory capacity for these resonant clock cycles decreases dramatically.

To explore if the observed effect is related to the choice of task, we plot the performance for four different tasks in Fig. 3. In each
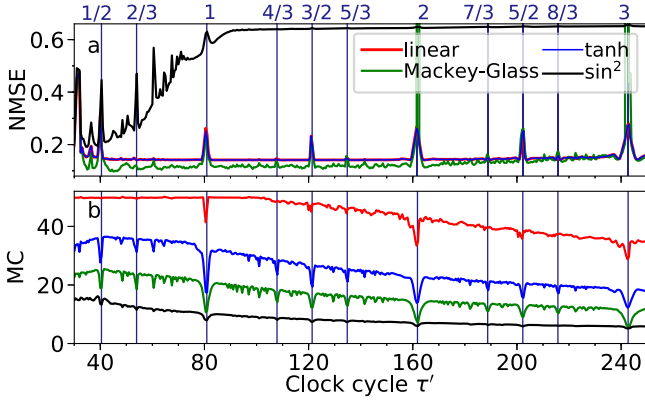
Fig. 2. Normalized mean square error (NMSE) of four different TDRCs given in Eq. (9) with nonlinearities of Eqs. (12)–(15) for the NARMA-10 task (top) and the total memory capacity MC_{tot} (bottom). Clearly visible are error peaks for values of τ' that are close to integer multiples of τ and low-order resonances. The presence of the peaks does not depend on the type of the activation function. Vertical lines denote the resonant values of τ to τ' as indicated on the top axis. The parameters for the simulations are listed in Table 1.
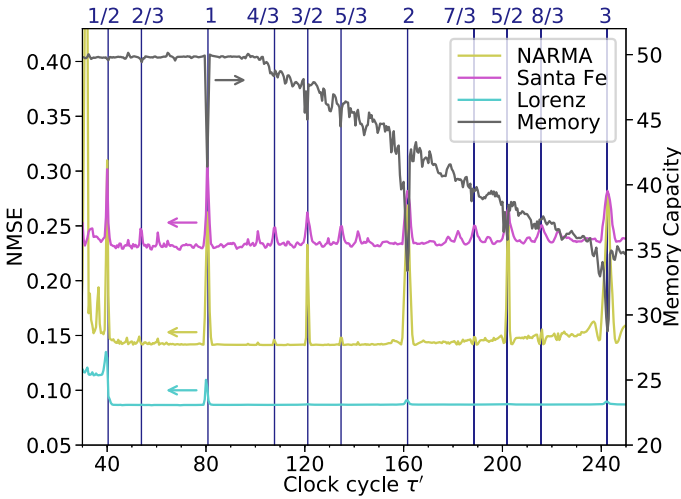

Fig. 3. Normalized mean square error (NMSE) for the NARMA-10 task, the Santa Fe time-series task and the Lorenz task (left axis) and the total memory capacity MC_{tot} (right axis) of the TDRC of Eq. (9) with linear activation function Eq. (12). For all tasks error peaks (resp. memory capacity drops) are visible for values of τ' that are close to integer multiples of τ and low-order resonances. Vertical lines denote the resonant values of τ to τ' as indicated on the top axis. The parameters for the simulations are listed in Table 1.

case the linear activation (12) was used. One can see that the error peaks (resp. performance drops) are visible in all tested tasks.

The rest of this paper is devoted to the explanation of this phenomenon. Thereby we focus on the linear activation (12). Lacking a nonlinearity, this is not the optimal choice for the TDRC, the resonance effects that we are interested in seem to be general and independent of f as shown in Fig. 2, where the performance for the activation functions (12)–(15) is computed. Furthermore, this simplification will allow us to deduce analytical results. In particular, we will be able to show the reason for the drop of the linear memory capacity explicitly.

4. Approximation by a network

In order to explain the degradation of the memory capacity for the resonant clock cycles, we present the time delay-system reservoir as an equivalent network. Similar procedure was done

Table 1
Parameters for the numerical computation of the NMSE for the NARMA-10, Lorenz and Santa Fe task and the memory capacity of the system.

System parameters	
Delay-time τ	80
Input scaling γ	0.02
Number of nodes N	50
Activation factor α	0.9
Range of clock cycle τ'	[30, 235]
Mackey-glass exponent p	1
Phase shift φ in activation function (15)	0.5
Simulation parameters	
Regularization parameter β	10^{-8}
Number of runs (with random mask)	100
Number of initial steps per run:	
∘ for Santa Fe task	4000
∘ for all other task	200 000
Number of training steps per run:	
∘ for Santa Fe task	4000
∘ for all other task	50 000
Number of test steps per run:	
∘ for Santa Fe task	1000
∘ for all other task	20 000
Time step of 4th order Runge–Kutta	0.01

in Appeltant et al. (2011) for a TDRC with $\tau' = \tau$ and arbitrary activation function. Here we present a derivation of an equivalent network for the case $\tau' \neq \tau$, where the results of Appeltant et al. (2011) cannot be applied. However, we have to restrict it to a linear activation function in order to obtain an explicit network representation.

An alternative way to describe the dependence of $X(k+1)$ on $X(k)$ and the input for the case $\tau' \neq \tau$ is presented in Larger et al. (2017), where the authors use an integral formula instead of a network formulation. This integral formula includes the case of a nonlinear activation function. For our explanation of the observed memory capacity drops, the network representation presented in this section is, however, indispensable.

Since a detailed derivation is technical, we move it to Appendix A and present the main results in this section.

As follows from Appendix A, the TDRC dynamics can be approximated by the discrete mapping

$$\tilde{X}(k+1) = B\tilde{X}(k) + (1 - e^{-\theta})F\left[A_q\tilde{X}(k+1-\ell) + A_{-(N-q)}\tilde{X}(k-\ell) + \gamma J_k\right], \quad (16)$$

where

$$\tilde{X}(k) := A_0^{-1}X(k) \quad (17)$$

and $X(k)$ is the discretized vector of the reservoir defined in Eq. (10). Let us define and explain further notations used in the mapping (16). The matrix

$$A_0 := \begin{pmatrix} 1 & 0 & \dots & 0 \\ e^{-\theta} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ e^{-(N-1)\theta} & \dots & e^{-\theta} & 1 \end{pmatrix} \quad (18)$$

is the classical coupling matrix of an equivalent network for TDRC with $\tau' = \tau$ (Appeltant et al., 2011). Moreover,

$$\ell := \left\lfloor \frac{m}{N} \right\rfloor, \quad q := m \bmod N,$$

$$m := \left\lceil \frac{\tau}{\theta} \right\rceil = \left\lceil \frac{\tau}{\tau'}N \right\rceil, \quad \theta = \frac{\tau'}{N}, \quad (19)$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the floor and the ceiling function, which we need to employ to allow delay-times τ that are not an integer

multiple of the time per virtual node θ. These quantities can be interpreted as follows: m is the number of virtual nodes that are needed to cover a τ-interval, q is a measure of the misalignment between τ and τ', and ℓ is roughly the ratio between the delay-time τ and the clock cycle τ'. For $\ell = 0$, the delay τ is shorter than the clock cycle τ', and it is similar to or larger than the clock cycle for $\ell \geq 1$. The matrices A_q and $A_{-(N-q)}$ are shifted versions of the matrix A_0. They are defined as follows:

$$
A_q := \begin{pmatrix}
0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & & & & & & \vdots \\
0 & & & & & & \vdots \\
1 & 0 & & & & & \vdots \\
e^{-\theta} & 1 & \ddots & & & & \vdots \\
\vdots & \ddots & \ddots & 0 & & & \vdots \\
e^{-(N-1-q)\theta} & \cdots & e^{-\theta} & 1 & 0 & \cdots & 0
\end{pmatrix} \tag{20}
$$

is obtained by a downwards shift of A_0 by q rows and

$$
A_{-(N-q)} :=
$$
$$
\begin{pmatrix}
e^{-(N-q)\theta} & \cdots & e^{-2\theta} & e^{-\theta} & 1 & 0 & \cdots & 0 \\
\vdots & & & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & 0 \\
e^{-(N-1)\theta} & \cdots & \cdots & \cdots & \cdots & e^{-2\theta} & e^{-\theta} & 1 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 0 \\
\vdots & & & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0
\end{pmatrix} \tag{21}
$$

is obtained by the upwards shift of A_0 by $N-q$ rows. Furthermore

$$
B := \begin{pmatrix}
e^{-N\theta} & \cdots & e^{-\theta} \\
0 & \cdots & 0 \\
\vdots & & \vdots \\
0 & \cdots & 0
\end{pmatrix}, \tag{22}
$$

$$
F \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} := \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix}, \tag{23}
$$

and J_k is defined as the input vector (8).

The mapping (16) generalizes previous results from Appeltant et al. (2011) and Schumacher et al. (2013b). If the clock cycle τ' satisfies $\tau' \in [\tau, \tau+\theta)$, the description coincides with the classical case $\tau' = \tau$ and the approximate equation (16) yields the same mapping

$$
\tilde{X}(k+1) = B\tilde{X}(k) + (1 - e^{-\theta})F[A_0\tilde{X}(k) + \gamma J_k] \tag{24}
$$

as presented in Appeltant et al. (2011) because then $\ell = 1$, $q = 0$, and $A_{-(N-q)} = 0$.

Analytical approaches for the nonlinear system (16) are challenging. To simplify, we will study the effect of different clock cycles τ' with the help of a linear activation function $f(x) = \alpha x$, where α is a scalar. Then Eq. (16) can be written as

$$
\tilde{X}(k+1) = B\tilde{X}(k) + \nu\alpha[A_q\tilde{X}(k+1-\ell) \\
+ A_{-(N-q)}\tilde{X}(k-\ell) + \gamma W^{\text{mask}}u(k)] \tag{25}
$$

by plugging in $W^{\text{mask}}u(k)$ for J_k and by writing $\nu := 1 - e^{-\theta}$ for the sake of shortness.

System (16) possesses the following properties: in the case $\tau' \geq \tau + \theta$, we have $\ell = 0$, and hence, Eq. (16) is in general

an implicit map. This is the physical case of a delay shorter than the clock cycle, which means that the feedback from some of the virtual nodes will act on other virtual nodes within the same cycle. However, for the linear activation function in Eq. (25) and by (17) we obtain for the case $\ell = 0$ the explicit map

$$
X(k+1) = AX(k) + W^{\text{in}}u(k), \tag{26}
$$

where

$$
A := A_0(\text{Id} - \nu\alpha A_q)^{-1}(B + \nu\alpha A_{-(N-q)})A_0^{-1} \tag{27}
$$

is a matrix that describes the coupling and local dynamics of the virtual network and

$$
W^{\text{in}} := \nu\alpha\gamma A_0(\text{Id} - \nu\alpha A_q)^{-1}W^{\text{mask}} \tag{28}
$$

is the generalized input matrix.

Eq. (26) is the main result of this section. It shows that the TDRC system can be modeled by an equivalent network for $\tau' \geq \tau + \theta$ if its activation function f is linear. In previous publications (Appeltant et al., 2011) such a network representation was derived for the case $\tau' = \tau$, however, in that case also for nonlinear activation functions.

An important observation from Eq. (26) is that the spectral radius of the matrix A must be smaller than one, otherwise the system (26) will not be asymptotically stable. We achieve this by choosing appropriate parameter $\alpha = 0.9$ (see Fig. 7 in Appendix F).

Eq. (26) allows us to calculate directly some figures of merit in the following. We first use it to explain the drops in the memory capacity in Fig. 2 for resonant delays. One important aspect to note, is that the basic shape of Eq. (26) does not change with τ'. Rather, a changing of the clock cycle leads to a change of the evolution matrices A and W^{in} of the equivalent network. The obtained system (26) can be equivalently considered as a specific echo state network.

5. Direct calculation of memory capacity

One can find an estimation for the memory capacity of a reservoir computing system by solving the system numerically and let it perform the memory task. But there are also analytic methods for some cases. In this section we explain how to calculate analytically the memory capacity of the linear echo state network (26) which corresponds to the case $\tau' \geq \tau + \theta$.

Memory capacity was originally defined by Jaeger in Jaeger (2002). In the following, we use a slightly modified formulation. Let the elements $u(k)$ of the input sequence be independently $\mathcal{N}(0, 1)$-distributed. Jaeger introduced the quantity MC_d which indicates how well the output $\hat{y}(k)$ of an ESN may approximate the input value $u(k - d)$ which was fed into the reservoir d time steps earlier. The memory capacity for a recall of d time steps in the past is defined by

$$
\text{MC}_d := \max_{W^{\text{out}}} \left(1 - \frac{E[(W^{\text{out}}X(k+d) - u(k))^2]}{\text{var}(u(k))} \right), \tag{29}
$$

where E denotes the expectation value and we require the initial state $X(0)$ of the reservoir to be stationary distributed in order to ensure that this definition is consistent, i.e. that the distribution of $X(k)$ does not depend on k. Since the spectral radius of A is less than one, the stationary distribution exists. In such a case, the memory capacity (29) with stationary distributed $X(0)$ can be equivalently written as

$$
\text{MC}_d := \max_{W^{\text{out}}} \left(1 - \frac{E[(W^{\text{out}}X(d) - u(0))^2]}{\text{var}(u(0))} \right) \\
= 1 - \min_{W^{\text{out}}} E[(W^{\text{out}}X(d) - u(0))^2]. \tag{30}
$$

Note that $u(0) \sim \mathcal{N}(0, 1)$ means that we can drop the term $\mathrm{var}(u(0))$ in (30).

The total memory capacity MC is defined as the sum of all d-step memory capacities

$$\mathrm{MC} := \sum_{d=1}^{\infty} \mathrm{MC}_d. \tag{31}$$

In the following we denote the optimal output weight vector for (30) by W_d^{out}. Let Σ be the covariance matrix of the stationary distribution of the reservoir. Jaeger (2002) noted that, if Σ is invertible, one can apply the Wiener–Hopf equation (Haykin, 1995) to find

$$W_d^{\mathrm{out}} = (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}} \Sigma^{-1}. \tag{32}$$

For details we refer to Appendix B. Using this optimal value W_d^{out}, the memory capacity (30) can be calculated as

$$\mathrm{MC}_d = (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}} \Sigma^{-1} A^{d-1} W^{\mathrm{in}}, \tag{33}$$

where we have used the relations

$$\begin{aligned}
\mathrm{E}[W_d^{\mathrm{out}}X(d)u(0)] &= \mathrm{cov}(W_d^{\mathrm{out}}X(d), u(0)) \\
&= (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}} \Sigma^{-1} A^{d-1} W^{\mathrm{in}}
\end{aligned} \tag{34}$$

and

$$\begin{aligned}
\mathrm{E}[(W_d^{\mathrm{out}}X(d))^2] &= \mathrm{var}(W_d^{\mathrm{out}}X(d)) \\
&= (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}} \Sigma^{-1} A^{d-1} W^{\mathrm{in}}.
\end{aligned} \tag{35}$$

So once the covariance matrix of the reservoir X is invertible, one can directly calculate the memory capacity. The stationary distribution of system (26) with standard normal distributed input elements $u(k)$ is a multivariate normal distribution with mean zero and covariance matrix

$$\Sigma = \sum_{j=0}^{\infty} A^j W^{\mathrm{in}} (W^{\mathrm{in}})^{\mathrm{T}} A^{j\mathrm{T}}. \tag{36}$$

We refer to Appendix B for a derivation. It is worth to comment on the structure of the matrix Σ. We note that the summands $A^j W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}}A^{j\mathrm{T}}$ in (36) are rank one symmetric matrices with the norm $\|A^j W^{\mathrm{in}}\|^2$. Since the spectral radius of A is less than one, this norm converges to zero as $j \to \infty$ and there is only a finite number of terms in (36) which can make numerically significant contribution to the rank of Σ. In addition, as we will see in the following, these rank-one matrices may have almost coinciding eigenspaces. As a result, the matrix Σ is in general numerically not invertible. Since our approach for the derivation of Eq. (33) relies on the invertibility of Σ, we cannot simply replace Σ^{-1} by a pseudo-inverse. In order to obtain an invertible covariance matrix, we need to perturb the stochastic process (26). We choose a small number $\sigma_\eta > 0$ and let $\eta(k) \sim \mathcal{N}(0, \mathrm{Id})$ be a sequence of independent multivariate normal distributed random variables. The stochastic process

$$X(k+1) = AX(k) + W^{\mathrm{in}}u(k) + \sigma_\eta \eta(k), \tag{37}$$

has the stationary distribution $\mathcal{N}(0, \Sigma_\eta)$ where the covariance matrix given by

$$\Sigma_\eta = \sum_{j=0}^{\infty} A^j (W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}} + \sigma_\eta \mathrm{Id}) A^{j\mathrm{T}} \tag{38}$$

is invertible.

6. Explanation for memory capacity gaps

Using the expressions (31) and (33) for the memory capacity obtained in Section 5, we provide an explanation for the loss of the memory capacity when τ'/τ is close to rational numbers with small denominator. The explanation is based on the structure of the covariance matrix Σ_η given by Eq. (38) and the corresponding expression for the memory capacity, which we repeat here for convenience

$$\begin{aligned}
&\mathrm{MC} = \sum_{d=1}^{\infty} \mathrm{MC}_d, \\
&\mathrm{MC}_d = (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}} \Sigma_\eta^{-1} A^{d-1} W^{\mathrm{in}},
\end{aligned} \tag{39}$$

where

$$\begin{aligned}
\Sigma_\eta &:= \sum_{j=0}^{\infty} (\Pi_j + \sigma_\eta A^j A^{j\mathrm{T}}), \\
\Pi_j &:= A^j W^{\mathrm{in}} (W^{\mathrm{in}})^{\mathrm{T}} A^{j\mathrm{T}}.
\end{aligned} \tag{40}$$

Our further strategy is as follows:

(i) Firstly, we remark that the norms of the individual terms in the sum (40) are converging to zero due to the convergence of the series. Hence, only the first finitely many terms play an important role. For instance, for our previously chosen parameters in Fig. 2, the terms with $j \gtrsim 30$ do not make a large contribution and can be neglected. In the following we denote the approximate number of significant terms by j_n.

(ii) We show that the largest eigenvalue of the jth term in (40) can be approximated by $\|A^j W^{\mathrm{in}}\|^2$ with the corresponding eigenvector $A^j W^{\mathrm{in}}$.

(iii) We show that the memory capacity is high, i.e. $\mathrm{MC}_d \approx 1$ for $d \leq j_n$, when the eigenvectors $A^j W^{\mathrm{in}}$ corresponding to the first relevant terms in the sum (40) are orthogonal.

(iv) Using our setup, we show numerically that the lower order resonances $\tau'/\tau \approx a/b$, where $a, b \in \mathbb{N}$ and b is small, lead to the alignment of the eigenvectors $A^j W^{\mathrm{in}}$, and hence, to the loss of the memory capacity. The small shift from the exact resonance values is explained by the standard drift property of delay systems.

(v) Finally we give an intuitive explanation of the obtained orthogonality conditions.

(i) *Convergence of the series* (40). The series (40) can be considered as the Neumann series $(\mathrm{Id} - T)^{-1} = \sum_{j=0}^{\infty} T^j$, where $T\mathcal{X} := A\mathcal{X}A^{\mathrm{T}}$, applied to the matrix $W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}} + \sigma_\eta \mathrm{Id}$. A sufficient condition for the convergence of such a series is that $\|T^k\|_{\mathrm{op}} := \sup_{\|\mathcal{X}\|=1} \|T^k \mathcal{X}\| < 1$ for some $k > 0$, where $\| \cdot \|$ is some matrix norm. Moreover, $\|T^k \mathcal{X}\| = \|A^k \mathcal{X} A^{k\mathrm{T}}\| \leq \|A^k\| \|\mathcal{X}\| \|A^{k\mathrm{T}}\| = \|A^k\|^2$ when $\|\mathcal{X}\| = 1$. Since the spectral radius of A is smaller than one, Gelfand's formula implies that there is a number $k > 0$ such that $\|A^k\| < 1$, and hence, the sufficient condition $\|T^k\|_{\mathrm{op}} < 1$ for the convergence of the series is satisfied.

(ii) *Estimating the largest eigenvalues and eigenvectors of the jth term in* (40). Consider at first the term with $j = 0$: $W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}} + \sigma_\eta \mathrm{Id}$. The largest eigenvalue of this matrix is $\|W^{\mathrm{in}}\|_2^2 + \sigma_\eta$ and the corresponding eigenvector is W^{in} as can be easily checked by the direct calculation

$$[W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}} + \sigma_\eta \mathrm{Id}]W^{\mathrm{in}} = (\|W^{\mathrm{in}}\|_2^2 + \sigma_\eta)W^{\mathrm{in}}. \tag{41}$$

For all other eigenvectors v, which are orthogonal to W^{in} due to the symmetry of the matrix, the corresponding eigenvalues are σ_η because

$$[W^{\mathrm{in}}(W^{\mathrm{in}})^{\mathrm{T}} + \sigma_\eta \mathrm{Id}]v = W^{\mathrm{in}}\langle W^{\mathrm{in}}, v\rangle + \sigma_\eta v = \sigma_\eta v. \tag{42}$$

These eigenvalues are by definition small, since σ_η is a small perturbation.

We can also find approximations of the eigenvectors and eigenvalues for the higher order terms $\Pi_j + \sigma_\eta A^j A^{j\mathrm{T}}, j > 0$. Namely,
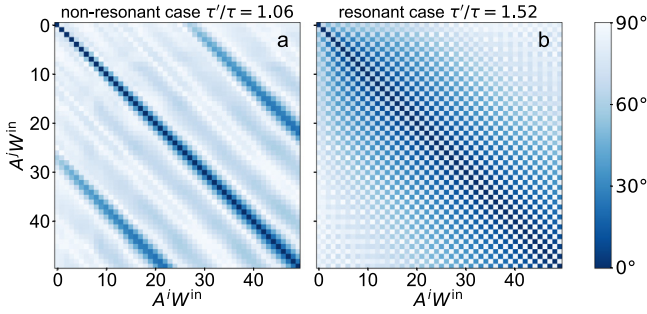
Fig. 4. The angles between $A^j W^{in}$ and $A^i W^{in}$, $i, j = 1, \ldots, 50$ are plotted in color, measuring, in particular, the orthogonality of the vectors $A^j W^{in}$ with different j. Panels (a) and (b) correspond to different ratios of τ'/τ: (a) $\tau'/\tau = 1.06$; (b) $\tau'/\tau = 1.52 \gtrsim 3/2$. The former is off-resonant, while the later is close to the 3/2 resonance, in particular when the drift property of the DDE system is taken into account. In (b) the vectors $A^j W^{in}$ point into the same direction for $j, j+2$, $j+4$, etc., i.e. after two time steps the input values $u(k)$ overlap in the state space of X and the memory capacity drops. In contrast, in the case $\tau'/\tau = 1.06$ (a) it takes almost 30 time steps before the input overlaps with past inputs in the network state. This explains the high memory capacity in this case, which is illustrated in Fig. 2.

for the unperturbed matrix Π_j, the largest eigenvalue is $\|A^j W^{in}\|^2$ and the corresponding eigenvector is $A^j W^{in}$ because

$$
\begin{aligned}
\Pi_j A^j W^{in} &= [A^j W^{in}(W^{in})^T A^{jT}]A^j W^{in} \\
&= A^j W^{in} \langle A^j W^{in}, A^j W^{in} \rangle \\
&= \|A^j W^{in}\|_2^2 A^j W^{in}.
\end{aligned} \tag{43}
$$

All other eigenvalues are zero. Since the largest eigenvalue of Π_j is geometrically and algebraically simple, it is continuous under the perturbation by $\sigma_\eta \mathrm{Id}$. Hence, the largest eigenvalue and the eigenvector of $\Pi_j + \sigma_\eta A^j A^{jT}$ are approximated by $\|A^j W^{in}\|^2$ and $A^j W^{in}$ with an error of order σ_η. All other eigenvalues are correspondingly small of order σ_η.

(iii) *The orthogonality of $A^j W^{in}$ leads to the high memory capacity.* Let j_n be the number of terms in (40) that are significant (see (i)), and let us assume that the eigenvectors $A^j W^{in}$, $j = 0, 1, \ldots, j_n$ are close to be orthogonal, i.e.

$$
\left| \langle A^j W^{in}, A^i W^{in} \rangle \right| \ll 1, \quad i, j = 0, 1, \ldots, j_n, \ j \neq i. \tag{44}
$$

As we will see in (iv), such an assumption is indeed reasonable in our setup. More precisely, one could consider (44) as $\left| \langle A^j W^{in}, A^i W^{in} \rangle \right| < \varepsilon$ introducing another small parameter $\varepsilon \ll 1$.

In case, when the orthogonality (44) holds, the largest eigenvalues of Σ_η and their corresponding eigenvectors can be approximated by $\|A^j W^{in}\|^2$ and $A^j W^{in}$, $j = 0, \ldots, j_n$. Indeed

$$
\begin{aligned}
\Sigma_\eta A^j W^{in} &= \sum_{k=0}^{\infty} (\Pi_k A^j W^{in} + \sigma_\eta A^k A^{kT} A^j W^{in}) \\
&= \|A^j W^{in}\|_2^2 A^j W^{in} + \mathcal{O}(\sigma_\eta) + \mathcal{O}(\varepsilon).
\end{aligned} \tag{45}
$$

In this case, the memory capacity can be calculated as follows:

$$
\begin{aligned}
MC_d &= (A^{d-1} W^{in})^T \Sigma_\eta^{-1} A^{d-1} W^{in} \\
&= (A^{d-1} W^{in})^T \frac{1}{\|A^{d-1} W^{in}\|_2^2} A^{d-1} W^{in} + \mathcal{O}(\sigma_\eta) + \mathcal{O}(\varepsilon) \\
&= 1 + \mathcal{O}(\sigma_\eta) + \mathcal{O}(\varepsilon)
\end{aligned} \tag{46}
$$

for $d \leq j_n$. Hence, the orthogonality of the vectors $A^j W^{in}$ with $A^i W^{in}$, $i \neq j$ guarantees a high memory capacity. We will present an intuitive explanation for this shortly.

(iv) *Resonances of τ' and τ lead to lower memory capacity.* The plots in Fig. 4 show $\langle A^j W^{in}, A^i W^{in} \rangle$ for different ratios τ'/τ. White

to light blue off-diagonal squares indicate that assumption (44) is satisfied, i.e. orthogonal or almost orthogonal vectors. Dark blue indicates a strong parallelism of the vectors. As can bee seen in the top panel of Fig. 4, the assumption (44) holds indeed for ratios τ'/τ which yield a good memory performance. Conversely, it is strongly violated for critical ratios $\tau'/\tau \gtrsim a/b$ with small denominator b, e.g. the center panel of Fig. 4.

We note that the critical ratios τ'/τ are slightly shifted from the exact resonant values 1, 3/2, etc. This shift is a manifestation of the 'drift' property of delay systems (Yanchuk & Giacomelli, 2017), which is caused by the fact that the effective round-trip of a signal in a delay system equals to the delay τ plus a finite processing time δ due to the integration (filtering). Therefore, the small shift of the error peaks is actually due to a resonance between the clock cycle τ' and $\tau + \delta$. This fact can be seen also in the structure of the coupling matrix A of the equivalent network.

(v) *Intuitive explanations.* There is an additional intuitive understanding of the above derived formulas. Recall that the original system of the reservoir of Eq. (9) combines the delay term $x(t - \tau)$ and the input $J(t)$ additively. The approximated network formula for an equivalent network translated this into the matrix A, which describes the free dynamics of the network, and the driving term defined by W^{in}. The state of the network is given by an N-dimensional system, and thus can at most hold N orthogonal dimensions (Dambre, Verstraeten, Schrauwen, & Massar, 2012). Each summand of Σ_η can now be understood as an imprint of the driving term on the system after j time steps. For $j = 0$ the matrix $A^0 = \mathrm{Id}$, and thus the imprint is given by W^{in}, i.e. the information of the current step is stored in the nodes as given by the weights of the effective input weight vector W^{in}. In the next step, the system will get an additional input, but also evolve according to its local dynamics A. Thus, after one time step, the imprint has transformed into AW^{in}, i.e. the summand for $j = 1$ and $\sigma_\eta \to 0$. Now in every step, the information that is currently present in the network will be 'rotated' in the phase space of the network according to A, while a new input will be projected onto the direction of W^{in}. This holds in general, so that the jth summand of Σ_η of Eq. (38) $A^j W^{in}$ describes the linear imprint of the input j steps in the past.

The orthogonality condition of Eq. (44) then is the same concept as demanding that new information from the inputs should not overwrite the already present information. If $A^r \approx s\mathrm{Id}$ for some $s \in \mathbb{R}$, then the information that was stored from r steps in the past will be partially overwritten by the currently injected step and lost. Hence, ensuring that the orthogonality between $A^j W^{in}$ is fulfilled as much as possible will maximize the linear memory. For the case of resonant feedback, i.e. $\tau' = \tau$, this condition is not fulfilled. This is due to the fact, that A has a strong diagonal component for the resonant cases, i.e. virtual nodes are most strongly coupled to themselves. This is a simple consequence of the fact that for $\tau = \tau'$, virtual nodes return to the single real node at the same time that they are updated. Similarly, for higher resonant cases $b\tau' = a\tau$, A^b will in general have a strong diagonal part and thus the eigenvector $A^b W^{in}$ will not be orthogonal to $A^0 W^{in}$, and the information will be overwritten.

7. Discussion

In this paper we have shown a generalization of the frequently used time-delay reservoir computing for cases other than $\tau' = \tau$. We observed that a sudden increase in the computing error (NARMA-10, Lorenz, and Santa-Fe NRMSE) and a drop in the linear memory capacity (MC) can be seen for resonant cases of $b\tau' \approx a\tau$ with $a, b \in \mathbb{N}$ where b is small for different activation functions including the linear, tanh, $\sin^2$, and the Mackey-Glass function. We derived an equivalent network for the case $\tau' \geq$
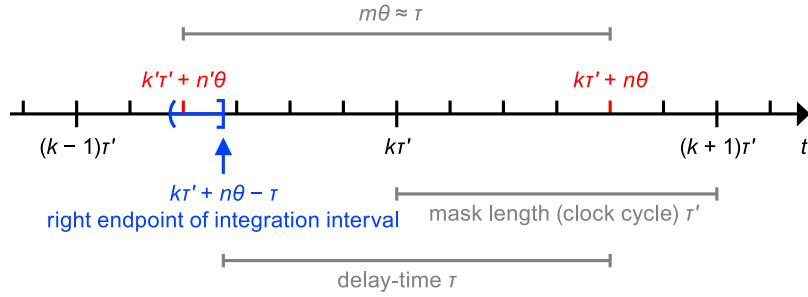
Fig. 5. The time interval over which the function x is integrated in Eq. (49) is highlighted in blue. As stated is Eq. (50), the point $k'\tau' + n'\theta$ must be chosen such that it lies within this interval. In Eq. (51) the value of x on the integration interval is approximated by the value of $x(k'\tau' + n'\theta)$. If the endpoints of the interval are grid points, $k'\tau' + n'\theta$ is chosen to be the right endpoint. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\tau + \theta$ which extends the previously studied case $\tau' = \tau$. Assuming a linear activation function $f(x) = \alpha x$, we can analytically solve the resulting implicit equations and obtain an expression for the total memory capacity MC. Here we find that the resulting memory capacity will be small for cases where τ and τ' are resonant because the information within the equivalent network will be overwritten by new inputs very quickly. Even though our analytics so far are only derived for the linear case, we expect these results to hold in more general situations, as numerical simulations with several different nonlinear activation functions in Figs. 2, 7, 8 indicate. More detailed analysis can be performed in future studies.

Appendix A. Derivation of equivalent networks

This section presents a detailed derivation of the ESN representation of TDRC systems. The derivation is structured as follows:

1. The delay system (9) is discretized such that the state of a virtual node $x(k\tau' + n\theta)$ depends on the state of its neighbor node $x(k\tau' + (n-1)\theta)$, the input $J_{k,n}$ and the state of a second node $x(k'\tau' + n'\theta)$ at the time $k' \leq k$. In order to do so, we approximate the integral of the continuous TDRC system on a small integration interval of length θ which covers the point $x(k'\tau' + n'\theta)$.
2. The formulas for n' and k' are derived.
3. The TDCR system can be written as a matrix equation. For this we use the same vectorization (10) of $x(t)$ as for the readout. An recurrent argument is employed to obtain the matrix equation.
4. It follows that the discretized TDRC system can be represented by an ESN if $\tau \leq \tau' - \theta$ and if the activation function f is linear.
5. For the sake of completeness, we formulate the equivalent ESN for the classical case $\tau' = \tau$, which was described in Appellant et al. (2011).

A.1. The delay reservoir system and discretization

Consider the delay-system (9), which we repeat here for convenience:

$$\dot{x}(t) = -x(t) + f[x(t - \tau) + \gamma J(t)], \tag{47}$$

where $\tau > 0$, $\gamma > 0$ and $f: \mathbb{R} \to \mathbb{R}$.

It follows that

$$e^{t-t_0}x(t) = x(t_0) + \int_{t_0}^{t} e^{s-t_0} f[x(s - \tau) + \gamma J(s)]\,ds \tag{48}$$

for $t \geq t_0$. Set $t_0 = k\tau' + (n-1)\theta$ and $t = k\tau' + n\theta$. Then

$$\begin{aligned} x(k\tau' + n\theta) &= e^{-\theta}x(k\tau' + (n-1)\theta) \\ &+ \int_0^\theta e^{s-\theta} f[x(k\tau' + (n-1)\theta + s - \tau) + \gamma J_{k,n}]\,ds, \end{aligned} \tag{49}$$

where $J_{k,n}$ is defined in (7). One option to discretize the system, is to approximate the function x by a step function with step length θ which is constant on the integration interval. One can find an appropriate step function by choosing $k'(k, n)$ and $n'(n)$ such that

$$k'\tau' + n'\theta \in (k\tau' + (n-1)\theta - \tau, k\tau' + n\theta - \tau] \tag{50}$$

and defining $x(t) \approx \tilde{x}(t) := x(k'\tau' + n'\theta)$ for $t \in (k\tau' + (n-1)\theta - \tau, k\tau' + n\theta - \tau]$. Then, one can replace x by $\tilde{x}$ in the integrand in Eq. (49). This yields

$$\begin{aligned} x(k\tau' &+ n\theta) \\ &\approx e^{-\theta}x(k\tau' + (n-1)\theta) \\ &+ \int_0^\theta e^{s-\theta} f[x(k'(k,n)\tau' + n'(n)\theta) + \gamma J_{k,n}]\,ds \\ &= e^{-\theta}x(k\tau' + (n-1)\theta) \\ &+ (1 - e^{-\theta})f[x(k'(k,n)\tau' + n'(n)\theta) + \gamma J_{k,n}]. \end{aligned} \tag{51}$$

A.2. The choice of k' and n'

The floor and the ceiling function are denoted by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$, respectively. One can choose k' and n' in the following way:

First, let $m \in \mathbb{Z}$, $m \geq 1$ be the unique number such that $\tau \in ((m-1)\theta, m\theta]$, i.e. $m = \lceil \tau/\theta \rceil$. Then

$$k'\tau' + n'\theta = k\tau' + n\theta - m\theta \tag{52}$$

as illustrated in Fig. 5. Now, the choice of n' follows directly from the restriction $n' \in \{1, \ldots, N\}$. It holds that

$$n'(n) = \begin{cases} (n - m) \mod N, & \text{if } N \nmid (n - m), \\ N, & \text{if } N \mid (n - m). \end{cases} \tag{53}$$

From this result follows that

$$(n-m)\theta = n'\theta + N\theta \cdot \begin{cases} \left\lfloor \frac{n-m}{N} \right\rfloor, & \text{if } N \nmid (n-m), \\ \frac{n-m}{N} - 1, & \text{if } N \mid (n-m), \end{cases}$$

$$= n'\theta + \left(\left\lceil \frac{n-m}{N} \right\rceil - 1 \right) \tau'. \tag{54}$$

Hence, Eq. (52) implies

$$k'(k,n) = k + \left\lceil \frac{n-m}{N} \right\rceil - 1. \tag{55}$$

Note that one has $k' = k$ as long as $n - m \in \{1, n-1\}$. If $n-m \in \{-N+1, \ldots, 0\}$, then $k' = k-1$. For $n-m \in \{-2N+1, \ldots, -N\}$ holds $k' = k-2$, etc.

A.3. Vectorization of the state space and a matrix equation for the discretized system

Define

$$X(k) := \begin{pmatrix} X_1(k) \\ \vdots \\ X_N(k) \end{pmatrix} := \begin{pmatrix} x((k-1)\tau' + \theta) \\ x((k-1)\tau' + 2\theta) \\ \vdots \\ x((k-1)\tau' + N\theta) \end{pmatrix} \tag{56}$$

and $\tilde{f} \equiv (1 - e^{-\theta})f$. From (51) follows

$$X_1(k+1) = x(k\tau' + \theta)$$
$$= e^{-\theta}x(k\tau') + \tilde{f}[x(k'(k,1)\tau' + n'(1)\theta) + \gamma J_{k,1}] \tag{57}$$
$$= e^{-\theta}X_N(k) + \tilde{f}[X_{n'(1)}(k'(k,1)+1) + \gamma J_{k,1}]$$

and repeated application of Eq. (51) yields

$$X_n(k+1)$$
$$= e^{-n\theta}X_N(k)$$
$$\quad + e^{-(n-1)\theta}\tilde{f}[X_{n'(1)}(k'(k,1)+1) + \gamma J_{k,1}]$$
$$\quad + e^{-(n-2)\theta}\tilde{f}[X_{n'(2)}(k'(k,2)+1) + \gamma J_{k,2}] \tag{58}$$
$$\quad \vdots$$
$$\quad + e^{-\theta}\tilde{f}[X_{n'(n-1)}(k'(k,n-1)+1) + \gamma J_{k,n-1}]$$
$$\quad + \tilde{f}[X_{n'(n)}(k'(k,n)+1) + \gamma J_{k,n}]$$

for $n \in \{2, \ldots, N\}$.

These equations can by rewritten as a matrix equation. Let

$$A_0 := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ e^{-\theta} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ e^{-(N-1)\theta} & \cdots & e^{-\theta} & 1 \end{pmatrix} \tag{59}$$

and

$$\tilde{F}\begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} := \begin{pmatrix} \tilde{f}(x_1) \\ \vdots \\ \tilde{f}(x_N) \end{pmatrix}. \tag{60}$$

Then

$$X(k+1) = A_0\tilde{F}\begin{pmatrix} X_{n'(1)}(k'(k,1)+1) + \gamma J_{k,1} \\ \vdots \\ X_{n'(N)}(k'(k,N)+1) + \gamma J_{k,N} \end{pmatrix}$$
$$+ \begin{pmatrix} e^{-\theta}X_N(k) \\ \vdots \\ e^{-N\theta}X_N(k) \end{pmatrix} \tag{61}$$

Let $\ell := \lfloor m/N \rfloor$ and $q := m \bmod N$, as defined in (19), i.e $m = \ell N + q$. By plugging this into Eq. (53) and noting that $1 \le n \le N$ and $0 \le q \le N-1$, one obtains

$$n'(n) = \begin{cases} n - q + N, & \text{for } n \le q, \\ n - q, & \text{for } n > q, \end{cases} \tag{62}$$

and by replacing m by $\ell N + q$ Eq. (55) follows

$$k'(k,n) = \begin{cases} k - \ell, & n > q, \\ k - \ell - 1, & n \le q. \end{cases} \tag{63}$$

Hence, the vector $(X_{n'(n)}(k'(k,n)+1))_{n=1,\ldots,N}$ can be written as follows:

$$\begin{pmatrix} X_{n'(1)}(k'(k,1)+1) \\ \vdots \\ X_{n'(N)}(k'(k,N)+1) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ X_1(k+1-\ell) \\ \vdots \\ X_{N-q}(k+1-\ell) \end{pmatrix}$$
$$+ \begin{pmatrix} X_{N-q+1}(k-\ell) \\ \vdots \\ X_N(k-\ell) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{64}$$

Thus, the map (61) can be written as

$$X(k+1) := A_0\tilde{F}[M_q X(k+1-\ell)$$
$$+ M_{-(N-q)}X(k-\ell) + \gamma J_k] + A_0 \begin{pmatrix} e^{-\theta}X_N(k) \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{65}$$

where the matrices $M_q = (\delta_{i,j+q})_{1 \le i,j \le N}$ and $M_{-(N-q)} = (\delta_{i,j-(N-q)})_{1 \le i,j \le N}$ are shift matrices.

The matrix A_0 is invertible and can be used to transform the system. Let $\tilde{X} := A_0^{-1}X$. Then

$$\tilde{X}(k+1) = B\tilde{X}(k) + \tilde{F}[A_q\tilde{X}(k+1-\ell)$$
$$+ A_{-(N-q)}\tilde{X}(k-\ell) + \gamma J_k], \tag{66}$$

where the matrix

$$A_q = M_q A_0 = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & & & \vdots \\ 0 & & & & & & \vdots \\ 1 & 0 & & & & & \vdots \\ e^{-\theta} & 1 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & 0 & & & \vdots \\ e^{-(N-1-q)\theta} & \cdots & e^{-\theta} & 1 & 0 & \cdots & 0 \end{pmatrix} \tag{67}$$

is obtained by a q rows downwards shift of A_0 and the matrix

$$
A_{-(N-q)} = M_{-(N-q)}A_0 =
\begin{pmatrix}
e^{-(N-q)\theta} & \cdots & e^{-2\theta} & e^{-\theta} & 1 & 0 & \cdots & 0 \\
\vdots & & & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & \ddots & \ddots & \ddots & 0 \\
e^{-(N-1)\theta} & \cdots & \cdots & \cdots & \cdots & e^{-2\theta} & e^{-\theta} & 1 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 0 \\
\vdots & & & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0
\end{pmatrix}
\tag{68}
$$

is obtained by an $N - q$ rows upwards shift of A_0 and

$$
B =
\begin{pmatrix}
e^{-N\theta} & \cdots & e^{-\theta} \\
0 & \cdots & 0 \\
\vdots & & \vdots \\
0 & \cdots & 0
\end{pmatrix}.
\tag{69}
$$

Eq. (69) for matrix B follows from Eq. (65). It must hold that

$$
\begin{pmatrix}
e^{-\theta}X_N(k) \\
0 \\
\vdots \\
0
\end{pmatrix}
= B\tilde{X}(k) = BA_0^{-1}X(k).
\tag{70}
$$

Hence,

$$
B =
\begin{pmatrix}
0 & \cdots & 0 & e^{-\theta} \\
0 & \cdots & \cdots & 0 \\
\vdots & & & \vdots \\
0 & \cdots & \cdots & 0
\end{pmatrix}
A_0.
\tag{71}
$$

A.4. An ESN representation of TDRC systems with suitable parameters

If $\tau \leq \tau' - \theta$, then $\ell = 0$. This follows from the definitions $\ell := \lfloor m/N \rfloor$ and $m = \lceil \tau/\theta \rceil$. Eq. (66) is in this case an implicit map:

$$
\tilde{X}(k+1) = B\tilde{X}(k) + \tilde{F}[A_q\tilde{X}(k+1) + A_{-(N-q)}\tilde{X}(k) + \gamma J_k].
\tag{72}
$$

However, for a linear activation function $f(x) = \alpha x$, where α is a scalar, holds $\tilde{f}(x) = (1 - e^{-\theta})\alpha x$ and hence one obtains the explicit linear map

$$
\tilde{X}(k+1) = (\mathrm{Id} - \nu\alpha A_q)^{-1}(B + \nu\alpha A_{-(N-q)})\tilde{X}(k) + \nu\alpha\gamma(\mathrm{Id} - \nu\alpha A_q)^{-1}J_k,
\tag{73}
$$

where $\nu := 1 - e^{-\theta}$. Since $\tilde{X} = A_0^{-1}X$ and $J_k = W^{\mathrm{mask}}u_k$, one can write this map in the original coordinates and in terms of the original input sequence

$$
X(k+1) = AX(k) + Wu(k),
\tag{74}
$$

where

$$
A := A_0(\mathrm{Id} - \nu\alpha A_q)^{-1}(B + \nu\alpha A_{-(N-q)})A_0^{-1}
\tag{75}
$$

and

$$
W^{\mathrm{in}} := \nu\alpha\gamma A_0(\mathrm{Id} - \nu\alpha A_q)^{-1}W^{\mathrm{mask}}.
\tag{76}
$$

The network matrix A is plotted in Fig. 6 for different parameters.
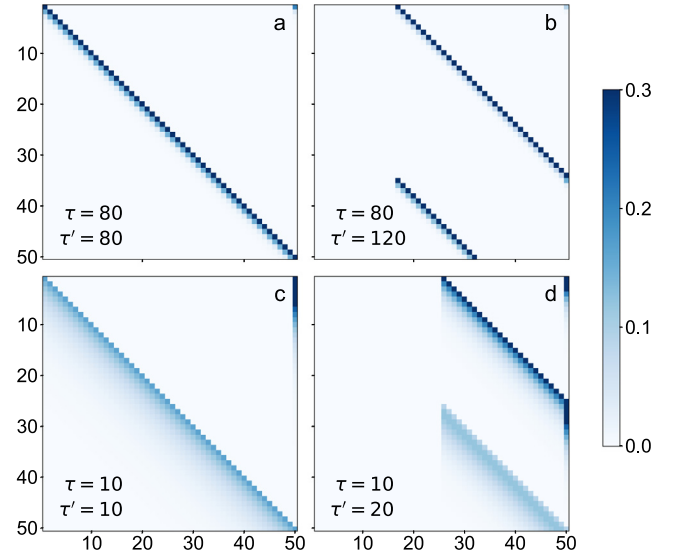

Fig. 6. Plot of the network matrix A, given by (75) resp. (80), for $N = 50$ and diverse values of τ and τ'. The connection weights are truncated at 0.3. The panels (a) and (c) show matrices for the classical case $\tau = \tau'$. (See Appendix A.5.) The panels (b) and (d) show matrices for the case $\tau \leq \tau' - \theta$ (Appendix A.4). These particular examples have resonant values of τ and τ', in fact we have (b) $\tau'/\tau = 1.5$ and (d) $\tau'/\tau = 2$. The strongest connection weights lie on diagonal lines which are shifted as the ratio of τ' and τ changes. The weights below these lines scale with the factor $e^{-n\theta}$, where n is the distance to the line. Since $\theta = \tau'/N$, the (off-diagonal) weights are larger in panels (c) and (d), were τ' is smaller.

A.5. The ESN representation of classical TDRC systems

The article (Appeltant et al., 2011) contains a description of an equivalent echo state network for TDRC systems with $\tau' = \tau$. This description is consistent with the case $\tau \in (\tau' - \theta, \tau']$ in the framework of our discretization. In this case,

$$
m = N, \quad \ell = 1, \quad q = 0,
\tag{77}
$$

and therefore, $A_q = A_0$ and $A_{-(N-q)}$ is the zero matrix. Thus, Eq. (66) simplifies to

$$
\tilde{X}(k+1) = B\tilde{X}(k) + \tilde{F}[A_0\tilde{X}(k) + \gamma J_k].
\tag{78}
$$

For a linear activation function $f(x) = \alpha x$ and $\tau \leq \tau' - \theta$, i.e. $\ell = 0$, the equivalent network written in the original coordinates is

$$
X(k+1) = AX(k) + Wu(k),
\tag{79}
$$

where

$$
A := A_0BA_0^{-1} + \nu\alpha A_0
\tag{80}
$$

and

$$
W^{\mathrm{in}} := \nu\alpha\gamma A_0 W^{\mathrm{mask}}.
\tag{81}
$$

Appendix B. Derivation of the memory capacity formula

We consider the linear echo state network

$$
X(k+1) = AX(k) + W^{\mathrm{in}}u(k),
\tag{82}
$$

where the input elements $u(k)$ are independently $\mathcal{N}(0, 1)$-distributed. In Section 5 we defined

$$
\mathrm{MC}_d = \max_{W^{\mathrm{out}}} \left(1 - \mathrm{E}[(W^{\mathrm{out}}X(k+d) - u(k))^2]\right)
\tag{83}
$$

and we claimed that

$$
W_d^{\mathrm{out}} = (A^{d-1}W^{\mathrm{in}})^{\mathrm{T}}\Sigma^{-1}
\tag{84}
$$

is the optimal argument for (83). In the following we show that W_d^{out} is indeed the optimal argument for (83).

Fig. 7. Influence of the parameter α of the linear model of Eq. (12) on the linear memory capacity. All other parameters are as in Table 1. The numerical simulations in the main part of this report were obtained with $\alpha = 0.9$.
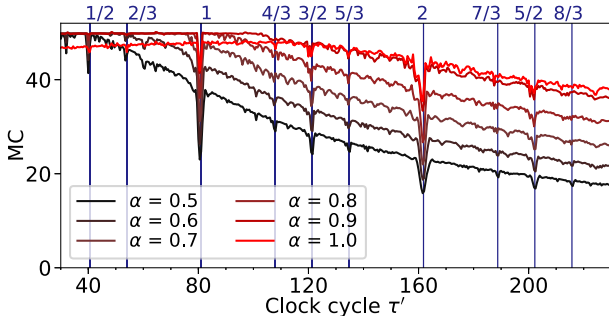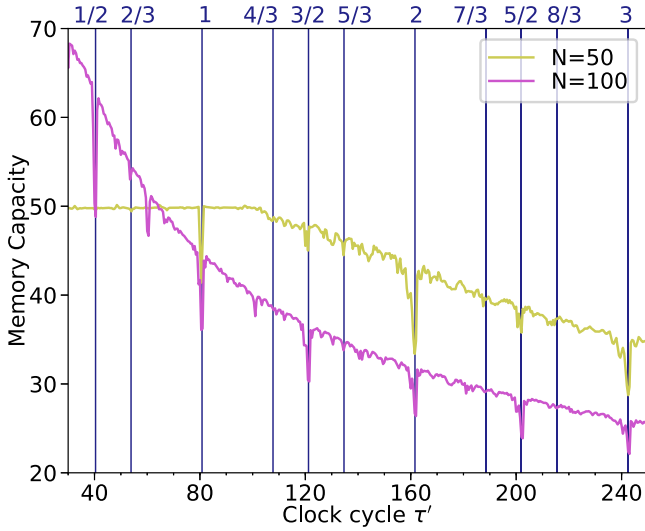

Fig. 8. Influence of the parameter N for the linear model of Eq. (12) on the linear memory capacity. All other parameters are as in Table 1. The numerical simulations in the main part of this report were obtained with $N = 50$.

In order to maximize (83), we need to minimize the mean square error

$$
\begin{aligned}
\text{MSE} &= \mathbb{E}[(W^{\text{out}}X(k+d) - u(k))^2] \\
&= \mathbb{E}[(W^{\text{out}}X(k+d))^2] + \mathbb{E}[u(k)^2] \\
&\quad - 2\mathbb{E}[W^{\text{out}}X(k+d)u(k)].
\end{aligned}
\tag{85}
$$

We know that $X(k) \sim \mathcal{N}(0, \Sigma)$ and hence

$$
W^{\text{out}}X(k) \sim \mathcal{N}(0, W^{\text{out}}\Sigma(W^{\text{out}})^{\text{T}}).
\tag{86}
$$

Note that $W^{\text{out}}\Sigma(W^{\text{out}})^{\text{T}}$ is a scalar because W^{out} is a row vector. Since the mean of $W^{\text{out}}X(k+d)$ is zero and $u(k) \sim \mathcal{N}(0, 1)$, we have

$$
\begin{aligned}
\mathbb{E}[(W^{\text{out}}X(k+d))^2] &= \text{var}(W^{\text{out}}X(k+d)) \\
&= W^{\text{out}}\Sigma(W^{\text{out}})^{\text{T}},
\end{aligned}
\tag{87}
$$

$$
\mathbb{E}[u(k)^2] = 1,
\tag{88}
$$

$$
\mathbb{E}[W^{\text{out}}X(k+d)u(k)] = \text{cov}(u(k), W^{\text{out}}X(k+d)).
\tag{89}
$$

Moreover,

$$
\begin{aligned}
&W^{\text{out}}X(k+d) \\
&= W^{\text{out}}\left(A^d X(k) + \sum_{j=0}^{d-1} A^j W^{\text{in}}u(k+d-1-j)\right)
\end{aligned}
\tag{90}
$$

and $u(k)$ is independent of $X(k)$. Therefore,

$$
\begin{aligned}
&\text{cov}(u(k), W^{\text{out}}X(k+d)) \\
&= \text{cov}(u(k), W^{\text{out}}A^{d-1}W^{\text{in}}u(k)) \\
&= W^{\text{out}}A^{d-1}W^{\text{in}}.
\end{aligned}
\tag{91}
$$

Thus, we obtain

$$
\text{MSE} = W^{\text{out}}\Sigma(W^{\text{out}})^{\text{T}} + 1 - 2W^{\text{out}}A^{d-1}W^{\text{in}}.
\tag{92}
$$

Since the mean square error is quadratic in the argument $W^{\text{out}} = (w_1^{\text{out}}, \ldots, w_N^{\text{out}})$, it has exactly one local minimum, which is the global minimum. A row vector W_d^{out} is the minimum argument if and only if

$$
\frac{\partial}{\partial w_n^{\text{out}}}\text{MSE}(W_d^{\text{out}}) = 0, \quad n = 1, \ldots, N.
\tag{93}
$$

For a quadratic form

$$
Q(v) = v^{\text{T}}Mv,
\tag{94}
$$

where $v \in \mathbb{R}^N$ and M is a symmetric matrix, the vector of the partial derivatives is given by

$$
\frac{\partial Q(v)}{\partial v} = 2v^{\text{T}}M.
\tag{95}
$$

Therefore,

$$
\frac{\partial \text{MSE}}{\partial W^{\text{out}}} = 2W^{\text{out}}\Sigma - 2(A^{d-1}W^{\text{in}})^{\text{T}}
\tag{96}
$$

and hence

$$
W_d^{\text{out}}\Sigma = (A^{d-1}W^{\text{in}})^{\text{T}}.
\tag{97}
$$

This formula is called Wiener–Hopf equation (Haykin, 1995). It follows that

$$
W_d^{\text{out}} = (A^{d-1}W^{\text{in}})^{\text{T}}\Sigma^{-1}.
\tag{98}
$$

Appendix C. The NARMA-10 benchmark

The 10th-order nonlinear autoregressive moving average (NARMA-10) task was introduced in Atiya and Parlos (2000) to evaluate the performance of machine learning methods on time series estimation. The NARMA-10 sequence $(y(k))_{k\geq0}$ is defined as follows: for an input sequence with independently $\mathcal{U}(0, 0.5)$-distributed elements $u(k)$, let

$$
y(0) = y(1) = \cdots = y(9) = 0
\tag{99}
$$

and

$$
\begin{aligned}
y(k+1) &= 0.3y(k) + 0.05y(k)\left(\sum_{j=0}^{9} y(k-j)\right) \\
&\quad + 1.5u(k-9)u(k) + 0.1
\end{aligned}
\tag{100}
$$

for $k \geq 9$.

In order to evaluate the performance of a reservoir computer, we choose sufficiently large numbers $k_0, K \in \mathbb{N}$ and we compare the output values $\hat{y}(k_0 + 1), \ldots, \hat{y}(k_0 + K)$ to the desired target values $y(k_0 + 1), \ldots, y(k_0 + K)$ by the normalized mean square error

$$
\text{NMSE} = \frac{1}{K}\sum_{k=k_0+1}^{k_0+K} \frac{(\hat{y}(k) - y(k))^2}{\text{var}(y)}.
\tag{101}
$$

Appendix D. The Lorenz benchmark

For the Lorenz task we use the three-dimensional Lorenz system:

$$
\begin{aligned}
\dot{\xi} &= 10(\upsilon - \zeta), \\
\dot{\upsilon} &= \xi(28 - \zeta) - \upsilon, \\
\dot{\zeta} &= \xi\upsilon - \frac{8}{3}\zeta.
\end{aligned}
\tag{102}
$$

We obtain a three-dimensional input sequence $u(k)$ of the reservoir by sampling with period 0.1 and normalization of all components, i.e.

$$
u(k) := \begin{pmatrix} \xi(k/10)/\mathrm{var}(\xi) \\ \upsilon(k/10)/\mathrm{var}(\upsilon) \\ \zeta(k/10)/\mathrm{var}(\zeta) \end{pmatrix}.
\tag{103}
$$

The task is a one-time-step-prediction task for the ξ-component, i.e. the target sequence is given by $y(k) := \xi((k+1)/10)/\mathrm{var}(\xi)$. For the evaluation we use the NMSE (101).

Appendix E. The Santa Fe benchmark

For the Santa Fe time-series prediction task we use a normalized version of the Santa Fe laser series (Weigend & Gershenfeld, 1993) as input. The target is to predict the next value of the series, i.e. as the Lorenz task, the Santa Fe task is a one-time-step-prediction task. For the evaluation we use the NMSE (101).

Appendix F. Parameters

Our choice of parameters does not significantly influence our results. In this section we present the numerical simulations that we used to verify this for α, θ resp. N.

Fig. 7 shows the memory capacity as a function of the clock cycle τ' for different α. The effect is visible for a large range of values. In the paper, we have used $\alpha = 0.9$, corresponding to the highest capacity out of the tested ones.

Fig. 8 shows the memory capacity as a function of the clock cycle τ' for two different numbers of virtual nodes $N = 50$ and $N = 100$. Since the time per virtual node is $\theta = \tau'/N$, it ranges between 0.6 and 5.0 in the case $N = 50$ and between 0.3 and 2.5 in the case $N = 100$. In both cases The effect is visible. In the paper, we have used $N = 50$.

References

Appeltant, L., Soriano, M., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., et al. (2011). Information processing using a single dynamical node as complex system. *Natural Communications*, [ISSN: 2041-1723] *2*(1), 468. http://dx.doi.org/10.1038/ncomms1476, URL http://www.nature.com/articles/ncomms1476.

Argyris, A., Bueno, J., & Fischer, I. (2018). Photonic machine learning implementation for signal recovery in optical communications. *Scientific Reports*, *8*(8487), 1–13. http://dx.doi.org/10.1038/s41598-018-26927-y.

Atiya, A. F., & Parlos, A. G. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, *11*(3).

Brunner, D., Penkovsky, B., Marquez, B. A., Jacquot, M., Fischer, I., & Larger, L. (2018). Tutorial: Photonic neural networks in delay systems. *Journal of Applied Physics*, *124*(15), 152004. http://dx.doi.org/10.1063/1.5042342.

Brunner, D., Soriano, M. C., Mirasso, C. R., & Fischer, I. (2011). Parallel photonic information processing at gigabyte per second data rates using transient states. *Natural Communications, 2*.

Dambre, J., Verstraeten, D., Schrauwen, B., & Massar, S. (2012). Information processing capacity of dynamical systems. *Scientific Reports, 2*, 514. http://dx.doi.org/10.1038/srep00514.

Erneux, T. (2009). *Surveys and Tutorials in the Applied Mathematical Sciences: vol. 3, Applied delay differential equations* (p. 204). Springer.

Erneux, T., Javaloyes, J., Wolfrum, M., & Yanchuk, S. (2017). Introduction to focus issue: Time-delay dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, [ISSN: 1054-1500] *27*(11), 114201. http://dx.doi.org/10.1063/1.5011354, URL http://aip.scitation.org/doi/10.1063/1.5011354.

Hale, J., & Verduyn Lunel, S. (1993). An introduction to functional differential equations. 99,

Haykin, S. (1995). *Adaptive filter theory* (3rd ed.). Prentice Hall.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks: Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep. 148.

Jaeger, H. (2002). Short term memory in echo state networks: Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep. 152.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science (80-.).*, [ISSN: 0036-8075] *304*(5667), 78–80. http://dx.doi.org/10.1126/SCIENCE.1091277.

Kuriki, Y., Nakayama, J., Takano, K., & Uchida, A. (2018). Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers. *Optics Express, 26*.

Larger, L., Baylón-Fuentes, A., Martinenghi, R., Udaltsov, V. S., Chembo, Y. K., & Jacquot, M. (2017). High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Physical Review X, 7*, 011015. http://dx.doi.org/10.1103/PhysRevX.7.011015, URL https://link.aps.org/doi/10.1103/PhysRevX.7.011015.

Larger, L., Soriano, M. C., Brunner, D., Appeltant, L., Gutierrez, J. M., Pesquera, L., et al. (2012). Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics Express*, *20*(3), 3241–3249. http://dx.doi.org/10.1364/OE.20.003241, URL http://www.opticsexpress.org/abstract.cfm?URI=oe-20-3-3241.

Lymburn, T., Khor, A., Stemler, T., Corrêa, D. C., Small, M., & Jüngling, T. (2019). Consistency in echo-state networks. *Chaos, 29*.

Maass, W., Natschläger, T., & Markam, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation, 14*(11), 2531–2560.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Optoelectronic reservoir computing. *Scientific Reports*, *2*(287), http://dx.doi.org/10.1038/srep00287.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., & Ott, E. (2018). Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, *120*(2), 24102. http://dx.doi.org/10.1103/physrevlett.120.024102.

Pikovsky, A., Rosenblum, M., & Kurths, J. (2001). *Synchronization: A universal concept in nonlinear science*. Cambridge University Press.

Röhm, A., & Lüdge, K. (2018). Multiplexed networks: reservoir computing with virtual and real nodes. *Journal of Physics Communications*, *2*, 85007. http://dx.doi.org/10.1088/2399-6528/aad56d.

Rulkov, N. F., Sushchik, M. M., Tsimring, L. S., & Abarbanel, H. D. I. (1995). Generalized synchronization of chaos in directionally coupled chaotic systems. *Physical Review E*, *51*, 980–994. http://dx.doi.org/10.1103/PhysRevE.51.980, URL https://link.aps.org/doi/10.1103/PhysRevE.51.980.

Van der Sande, G., Brunner, D., & Soriano, M. C. (2017). Advances in photonic reservoir computing. *Nanophotonics*, *6*(3), 561. http://dx.doi.org/10.1515/nanoph-2016-0132.

Schumacher, J., Toutounji, H., & Pipa, G. (2013a). An analytical approach to single node delay-coupled reservoir computing. In *Conference: 23rd international conference on artificial neural networks*. http://dx.doi.org/10.1007/978-3-642-40728-4_4.

Schumacher, J., Toutounji, H., & Pipa, G. (2013b). An analytical approach to single node delay-coupled reservoir computing. http://dx.doi.org/10.1007/978-3-642-40728-4_4.

Smith, H. (2010). *An introduction to delay differential equations with applications to the life sciences, Vol. 57*. http://dx.doi.org/10.1007/978-1-4419-7646-8.

Verstraeten, D., Schrauwen, B., & Stroobandt, D. (2006). Reservoir-based techniques for speech recognition. In *2006 IEEE Int. Jt. Conf. Neural Netw. Proc.* (pp. 1050–1053). IEEE, http://dx.doi.org/10.1109/IJCNN.2006.246804, URL http://ieeexplore.ieee.org/document/1716215/.

Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the Liquid State Machine: a case study. *Information Processing Letters*, [ISSN: 0020-0190] *95*(6), 521–528. http://dx.doi.org/10.1016/J.IPL.2005.05.019, URL https://www.sciencedirect.com/science/article/pii/S0020019005001523?via%3Dihub.

Weigend, A. S., & Gershenfeld, N. A. (1993). Results of the time series prediction competition at the santa fe institute. In *IEEE international conference on neural networks, Vol. 3* (pp. 1786–1793). http://dx.doi.org/10.1109/ICNN.1993.298828.

Yanchuk, S., & Giacomelli, G. (2017). Spatio-temporal phenomena in complex systems with time delays. *Journal of Physics A*, [ISSN: 1751-8113] *50*(10), 103001. http://dx.doi.org/10.1088/1751-8121/50/10/103001, URL http://stacks.iop.org/1751-8121/50/i=10/a=103001?key=crossref.f760c062e912b820ac69c9174ac61305.