Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals

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

Abstract

This paper addresses the reservoir design problem in the context of delay-based reservoir computers for multidimensional input signals, parallel architectures, and real-time multitasking. First, an approximating reservoir model is presented in those frameworks that provides an explicit functional link between the reservoir architecture and its performance in the execution of a specific task. Second, the inference properties of the ridge regression estimator in the multivariate context are used to assess the impact of finite sample training on the decrease of the reservoir capacity. Finally, an empirical study is conducted that shows the adequacy of the theoretical results with the empirical performances exhibited by various reservoir architectures in the execution of several nonlinear tasks with multidimensional inputs. Our results confirm the robustness properties of the parallel reservoir architecture with respect to task misspecification and parameter choice that had already been documented in the literature.

Key Words: Reservoir computing, echo state networks, liquid state machines, time-delay reservoir, parallel computing, memory capacity, multidimensional signals processing, Big Data.

1 Introduction

The recent and fast development of numerous massive data acquisition technologies results in a considerable growth of the data volumes that are stored and that need to be processed in the context of many human activities. The variability, complexity, and volume of this information have motivated the appearance of the generic term *Big Data*, which is mainly used to refer to datasets whose features make the traditional data processing approaches inadequate. This relatively new concept calls for the development of specialized tools for data preprocessing, analysis, transferring, and visualization, as well as for novel data mining and machine learning techniques in order to tackle specific computational tasks.

In this context, there is a recent but already well established paradigm for neural computation known by the name of **reservoir computing (RC)** [Jaeg 01, Jaeg 04, Maas 02, Maas 11, Croo 07, Vers 07,

[1]Department of Mathematics and Statistics. Universität Konstanz. Box 146. D-78457 Konstanz. Germany. Lyudmila.Grigoryeva@uni-konstanz.de

[2]CHRU Besançon. 2 place Saint Jacques. F-25000 Besançon. jhenriques@chu-besancon.fr

[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. Laurent.Larger@univ-fcomte.fr

[4]Corresponding author. Universität Sankt Gallen, Faculty of Mathematics and Statistics, Bodanstrasse 6, CH-9000 Sankt Gallen, Switzerland, and Centre National de la Recherche Scientifique (CNRS), France. Juan-Pablo.Ortega@univ-fcomte.fr

Luko 09] (also referred to as *Echo State Networks* and *Liquid State Machines*), that has already shown a significant potential in successfully confronting some of the challenges that we just described.

This brain-inspired machine learning methodology exhibits several competitive advantages with respect to more traditional approaches. First, the supervised learning scheme associated to it is extremely simple. Second, some implementations of the RC paradigm are based on the computational capacities of certain dynamical systems [Crut 10] that open the door to physical realizations that have already been built using dedicated hardware (see, for instance, [Jaeg 07, Atiy 00, Appe 11, Roda 11, Larg 12, Paqu 12]) and that, recently, have shown unprecedented information processing speeds [Brun 13]. Our work takes place in the context of a specific type of RCs called **time-delay reservoirs (TDRs)** that are constructed via the sampling of the solutions of time-delay differential equations.

Despite the outstanding empirical performances of TDRs described in the above listed references and the convenience of their associated learning scheme, a well-known important drawback is that these devices show a certain lack of structural task universality. More specifically, each task presented to a TDR requires that the TDR parameters and, more generally, its architecture are tuned in order to achieve optimal performance or, equivalently, small deviations from the optimal parameter values can seriously degrade the reservoir performance. The optimal parameters have been traditionally found by trial and error or by running costly numerical scannings for each task. More recently, in [Grig 15] we introduced a method to overcome this difficulty by providing a functional link between the RC parameters and its performance with respect to a given task and that can be used to accurately determine the optimal reservoir architecture by solving a well structured optimization problem; this feature simplifies enormously the implementation effort and sheds new light on the mechanisms that govern this information processing technique.

This paper builds on the techniques introduced in [Grig 15] and extends those results in the following directions:

(i) The memory capacity formulas in [Grig 15] are generalized to **multidimensional input signals** and we provide capacity estimations for the simultaneous execution of several memory tasks. This feature, sometimes referred to as **real-time multitasking** [Maas 11] is usually presented as one of the most prominent computational advantages of RC.

(ii) We provide memory capacity estimations for **parallel arrays of reservoir computers**. This reservoir architecture has been introduced in [Orti 12, Grig 14] and has been empirically shown to exhibit improved robustness properties with respect to the dependence of the optimal reservoir parameters on the task presented to the device and also with respect to task misspecification.

(iii) We carry out an in-depth study of the ridge regression estimator in the multivariate context in order to assess **the impact of the finiteness of the training sample on the decrease of reservoir capacity**. More specifically, when the teaching signal used to train the RC has finite size, the faulty estimation of the RC readout layer (see Section 2.2) introduces an error that adds to the characteristic error associated to the RC scheme and that we explicitly quantify. The resulting formula can be used in passing to determine, for a given training sample, the value of the ridge regularization strength that minimizes the training error. The linear character of the readout scheme is a defining feature of RC that, apart from its simplicity, offers as an advantage the possibility of using statistical inference in order to assess the quality of the training, something that is in general impossible when using other machine learning strategies like standard neural networks.

(iv) We conduct an empirical study that shows the adequacy of our theoretical results with the empirical performances exhibited by TDRs in the execution of various nonlinear tasks with multidimensional inputs. Additionally, using the approximating model, we confirm the robustness properties of the

parallel reservoir architecture with respect to task misspecification and parameter choice that had already been documented in [Grig 14].

The paper is organized as follows: Section 2 recalls the general setup for time-delay reservoir computing, as well as the notions of characteristic error and memory capacity in the multitasking setup. Section 3 constitutes the core of the paper and addresses the points **(i)** through **(iii)** listed in the previous paragraphs. The results in that section are presented in a nutshell in order to make their use as accessible as possible with a minimum amount of prerequisites; all the details regarding the models that lead to them can be found in the appendices in Section 6. The empirical study described in point **(iv)** is contained in Section 4. Section 5 concludes the paper

Acknowledgments: We thank two anonymous referees for their comments that have significantly improved the paper. We also thank Serge Massar for questions and comments in relation to Proposition 6.2. We acknowledge partial financial support of the Région de Franche-Comté (Convention 2013C-5493), the European project PHOCUS (FP7 Grant No. 240763), the ANR "BIPHOPROC" project (ANR-14-OHRI-0002-02), and Deployment S.L. LG acknowledges financial support from the Faculty for the Future Program of the Schlumberger Foundation.

2 Reservoir computing and time-delay reservoirs: notation and preliminaries

In this section we introduce the notation that we use in the paper, we briefly recall the general setup for time-delay reservoirs (TDRs), and provide various preliminary concepts that are needed in the following sections.

2.1 Notation

Column vectors are denoted by bold lower or upper case symbol like $\mathbf{v}$ or $\mathbf{V}$. We write $\mathbf{v}^\top$ to indicate the transpose of $\mathbf{v}$. Given a vector $\mathbf{v} \in \mathbb{R}^n$, we denote its entries by v_i, with $i \in \{1, \ldots, n\}$; we also write $\mathbf{v} = (v_i)_{i\in\{1,\ldots,n\}}$. The symbols $\mathbf{i}_n$ and $\mathbf{0}_n$ stand for the vectors of length n consisting of ones and zeros, respectively. We denote by $\mathbb{M}_{n,m}$ the space of real $n \times m$ matrices with $m, n \in \mathbb{N}$. When $n = m$, we use the symbol $\mathbb{M}_n$ to refer to the space of square matrices of order n. Given a matrix $A \in \mathbb{M}_{n,m}$, we denote its components by A_{ij} and we write $A = (A_{ij})$, with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots m\}$. If A and B are two matrices with the same number of rows, we denote by $(A||B)$ the matrix resulting from their vertical concatenation. We write $\mathbb{I}_n$ and $\mathbb{O}_n$ to denote the identity matrix and the zero matrix of dimension n, respectively. We use $\mathbb{S}_n$ to indicate the subspace $\mathbb{S}_n \subset \mathbb{M}_n$ of symmetric matrices, that is, $\mathbb{S}_n = \left\{A \in \mathbb{M}_n \mid A^\top = A\right\}$. Given a matrix $A \in \mathbb{M}_{n,m}$, we denote by vec the operator that transforms A into a vector of length nm by stacking all its columns, namely,

$$\mathrm{vec} : \mathbb{M}_{n,m} \longrightarrow \mathbb{R}^{nm}, \quad \mathrm{vec}\,(A) = (A_{11}, \ldots, A_{n1}, \ldots, A_{1m}, \ldots, A_{nm})^\top, \quad A \in \mathbb{M}_{n,m}.$$

When A is symmetric, we denote by vech the operator that stacks the elements on and below the main diagonal of A into a vector of length $N := \frac{1}{2}n\,(n+1)$, that is,

$$\mathrm{vech} : \mathbb{S}_n \longrightarrow \mathbb{R}^N, \quad \mathrm{vech}\,(A) = (A_{11}, \ldots, A_{n1}, A_{22}, \ldots, A_{n2}, \ldots, A_{nn})^\top, \quad A \in \mathbb{S}_n.$$

Let $N := \frac{1}{2}n\,(n+1)$. We denote by $L_n \in \mathbb{M}_{N,n^2}$ and by $D_n \in \mathbb{M}_{n^2,N}$ the elimination and the duplication matrices [Lutk 05], respectively. These matrices satisfy that:

$$\mathrm{vech}\,(A) = L_n \mathrm{vec}\,(A), \quad \text{and} \quad \mathrm{vec}\,(A) = D_n \mathrm{vech}\,(A). \tag{2.1}$$

Consider $A \in \mathbb{S}_n$, $\mathbf{v} = \text{vech}(A) \in \mathbb{R}^N$, and $S = \{(i,j)\,|i,j \in \{1,\ldots,n\}, i \geq j\}$. Let $\sigma : S \longrightarrow \{1,\ldots,N\}$ be the operator that assigns to the position of the entry (i,j), $i \geq j$, of the matrix A the position of the corresponding element of $\mathbf{v}$ in the vech representation. We refer to the inverse of this operator as $\sigma^{-1} : \{1,\ldots,N\} \longrightarrow S$. The symbol $||A||_{\text{Frob}}$ denotes the Frobenius norm of $A \in \mathbb{M}_{m,n}$ defined as $\|A\|^2_{\text{Frob}} := \text{trace}\left(A^T A\right)$ [Meye 00]. Finally, the symbols $\text{E}[\cdot]$ and $\text{Cov}(\cdot,\cdot)$ denote the mathematical expectation and the covariance, respectively.

2.2 The general setup for time-delay reservoir (TDR) computing

The functional time-delay differential equations used for TDR computing. The time-delay reservoirs studied in this paper are constructed by sampling the solutions of time-delay differential equations of the form

$$\dot{x}(t) = -x(t) + f(x(t-\tau), I(t), \boldsymbol{\theta}), \tag{2.2}$$

where f is a nonlinear smooth function that will be referred to as **nonlinear kernel**, $\boldsymbol{\theta} \in \mathbb{R}^K$ is a vector that contains the parameters of the nonlinear kernel, $\tau > 0$ is the **delay**, $x(t) \in \mathbb{R}$, and $I(t) \in \mathbb{R}$ is an external forcing that makes (2.2) non-autonomous and that in our construction will be used as an inlet into the system for the signal that needs to be processed. We emphasize that the solution space of equation (2.2) 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 nonlinear kernel f is chosen based on the concrete physical implementation of the computing system that is envisioned. We consider two specific parametric sets of kernels that have already been explored in the literature, namely:

(i) The **Mackey-Glass** [Mack 77] nonlinear kernel:

$$f(x, I, \boldsymbol{\theta}) = \frac{\eta\,(x + \gamma I)}{1 + (x + \gamma I)^p}, \quad \boldsymbol{\theta} := (\gamma, \eta, p) \in \mathbb{R}^3, \tag{2.3}$$

which is used in electronics-based RC implementations [Appe 11].

(ii) The **Ikeda** [Iked 79] nonlinear kernel

$$f(x, I, \boldsymbol{\theta}) = \eta \sin^2(x + \gamma I + \phi), \quad \boldsymbol{\theta} := (\eta, \gamma, \phi) \in \mathbb{R}^3, \tag{2.4}$$

associated to an optical RC implementation [Larg 12].

For these specific choices of nonlinear kernel, the parameters γ and η are usually referred to as the **input** and **feedback gains**, respectively.

Continuous and discrete-time approaches to multidimensional TDR computing. We briefly recall the design of a TDR using the solutions of (2.2). The following constructions are discussed in detail in [Grig 15]. TDRs are based on the sampling of the solutions of (2.2) when driven by an input forcing obtained out of the signal that needs to be processed. More specifically, let $\mathbf{z}(t) \in \mathbb{R}^n$, $t \in \mathbb{Z}$, be an n-dimensional discrete-time **input signal**. This signal is, first, time and dimensionally multiplexed over a delay period by using an **input mask** $\mathbf{C} \in \mathbb{M}_{N,n}$ and by setting $\mathbf{I}(t) := \mathbf{C}\mathbf{z}(t)$, $t \in \mathbb{Z}$, where N is a design parameter called the **number of neurons** of each **reservoir layer**. The resulting discrete-time N-dimensional signal $\mathbf{I}(t) \in \mathbb{R}^N$ is called **input forcing**.

We construct the TDR as a collection of **neuron values** $x_i(t)$ organized in **layers** $\mathbf{x}(t) \in \mathbb{R}^N$ (also referred to as **reservoir output**) of $N \in \mathbb{N}$ of **virtual neurons** each, parameterized by $t \in \mathbb{Z}$. The value $x_i(t)$, referred to as the i**th neuron value** of the tth layer $\mathbf{x}(t)$ of the reservoir.

In the **continuous time TDR** case, the reservoir output is obtained by sampling a solution $x(t)$ of (2.2) by setting

$$x_i(t) := x(t\tau - (N-i)d), \quad i \in \{1,\ldots,N\}, \quad t \in \mathbb{Z}, \tag{2.5}$$

where $d := \tau/N$ is referred to as the **separation between neurons**. The solution $x(t)$ has been obtained by using an external forcing $I(s)$ in (2.2) constructed out of the input forcing $\mathbf{I}(t)$ as follows: given $s \in \mathbb{R}$, let $t \in \mathbb{Z}$ and $i \in \{2, \ldots, N\}$ be the unique values such that $s \in (t\tau-(N-i-1)d, t\tau-(N-i)d]$ and that we use to define the external forcing as $I(s) := (\mathbf{I}(t))_i$.

The **discrete-time** TDR is constructed via the Euler time-discretization of (2.2) with an integration step of $d := \tau/N$. In this case, the neuron values are determined by the following recursions:

$$x_i(t) := e^{-\xi} x_{i-1}(t) + (1 - e^{-\xi}) f(x_i(t-1), (\mathbf{I}(t))_i, \boldsymbol{\theta}), \tag{2.6}$$

with $x_0(t) := x_N(t-1)$, $\xi := \log(1+d)$, and $i \in \{1, \ldots, N\}$. In this case, the recursions (2.6) uniquely determine a smooth map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ referred to as the **reservoir map** that specifies the neuron values of a given tth layer as a recursion on the neuron values of the preceding layer $t-1$ via an expression of the form

$$\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{I}(t), \boldsymbol{\theta}). \tag{2.7}$$

The TDR memory capacity for real-time multitasking. In this paper we study the performance of TDRs at the time of simultaneously performing several memory tasks (see Figure 1). This means that we will evaluate the ability of the TDR to reproduce a prescribed multidimensional nonlinear function of the input signal

$$\begin{array}{rccc} H: & \mathbb{R}^{(h+1)n} & \longrightarrow & \mathbb{R}^q \\ & \mathrm{vec}(\mathbf{z}(t), \ldots, \mathbf{z}(t-h)) & \longmapsto & \mathbf{y}(t), \end{array} \tag{2.8}$$

that we will call q**-dimensional** h**-lag memory task** for the n-dimensional input signal $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$.

In the RC context, this task is performed by using a finite size realization of the input signal $\{\mathbf{z}(-h+1), \ldots, \mathbf{z}(T)\}$ that is used to construct a q-dimensional **teaching signal** $\{\mathbf{y}(1), \ldots, \mathbf{y}(T)\}$ by setting $\mathbf{y}(t) := H(\mathrm{vec}(\mathbf{z}(t), \ldots, \mathbf{z}(t-h)))$. The teaching signal is subsequently used to determine a pair $(W_{\mathrm{out}}, \boldsymbol{a}_{\mathrm{out}}) \in \mathbb{M}_{N,q} \times \mathbb{R}^q$ that performs the memory task as an affine combination of the reservoir outputs. The optimal pair $(W_{\mathrm{out}}, \boldsymbol{a}_{\mathrm{out}})$ is obtained with a ridge regression that minimizes the regularized mean square error (MSE), that is,

$$\begin{aligned} (W_{\mathrm{out}}, \boldsymbol{a}_{\mathrm{out}}) &= \underset{W\in\mathbb{M}_{N,q}, \boldsymbol{a}\in\mathbb{R}^q}{\arg\min} \left(\mathrm{trace}\left(\mathrm{E}\left[(W^\top \cdot \mathbf{x}(t) + \boldsymbol{a} - \mathbf{y}(t))^\top (W^\top \cdot \mathbf{x}(t) + \boldsymbol{a} - \mathbf{y}(t))\right]\right) + \lambda \|W\|^2_{\mathrm{Frob}}\right) \\ &=: \underset{W\in\mathbb{M}_{N,q}, \boldsymbol{a}\in\mathbb{R}^q}{\arg\min} \left(\mathrm{MSE}(W, \boldsymbol{a}) + \lambda \|W\|^2_{\mathrm{Frob}}\right). \end{aligned} \tag{2.9}$$

The optimal pair $(W_{\mathrm{out}}, \boldsymbol{a}_{\mathrm{out}})$ that solves the ridge regression problem (2.9) is referred to as the **readout layer**. The ridge regularization strength parameter $\lambda \in \mathbb{R}$ is traditionally tuned during the training phase via cross validation; in the next section we provide a result that can be used, once the training sample has been chosen, to determine beforehand the value of the parameter λ that minimizes the training error. The explicit solution of the optimization problem (2.9) (see [Grig 15] for the details) is given by

$$W_{\mathrm{out}} = (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} \mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t)), \tag{2.10}$$

$$\boldsymbol{a}_{\mathrm{out}} = \boldsymbol{\mu}_y - W_{\mathrm{out}}^\top \boldsymbol{\mu}_x, \tag{2.11}$$

where $\boldsymbol{\mu}_x := \mathrm{E}[\mathbf{x}(t)] \in \mathbb{R}^N$, $\boldsymbol{\mu}_y := \mathrm{E}[\mathbf{y}(t)] \in \mathbb{R}^q$, $\Gamma(0) := \mathrm{Cov}(\mathbf{x}(t), \mathbf{x}(t)) \in \mathbb{S}_N$, and $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t)) \in \mathbb{M}_{N,q}$. Stationarity hypotheses are assumed on the teaching signal and the reservoir output so that the first and second order moments that we just listed are time-independent. The error committed by the reservoir when accomplishing the task H with the optimal readout will be referred to as its **characteristic error** and is given by the expression

$$\mathrm{MSE}(W_{\mathrm{out}}) = \mathrm{trace}(\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t)) - W_{\mathrm{out}}^\top (\Gamma(0) + 2\lambda \mathbb{I}_N) W_{\mathrm{out}}), \tag{2.12}$$

that can be encoded under the form of a **memory capacity** $C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda)$ with values between zero and one that depends on the task H that is being tackled, the input mask $\mathbf{C}$, the reservoir parameters $\boldsymbol{\theta}$ and the regularization strength λ:

$$C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda) := 1 - \frac{\mathrm{MSE}(W_{\mathrm{out}})}{\mathrm{trace}\left(\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))\right)} = \frac{\mathrm{trace}(W_{\mathrm{out}}^{\top}(\Gamma(0) + 2\lambda \mathbb{I}_N) W_{\mathrm{out}})}{\mathrm{trace}\left(\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))\right)}, \tag{2.13}$$

where W_{out} is provided by the solution in (2.10). In order to evaluate (2.13) for a specific memory task, the expressions of $\Gamma(0)$, $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, and $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ need to be computed. The matrix $\Gamma(0)$ depends exclusively on the input signal and the reservoir architecture but $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ and $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$ are related to the specific memory task H at hand. The computation of (2.13) is in general very complicated and that is why in [Grig 15] we introduced a simplified reservoir model that allowed us to efficiently evaluate this expression for one-dimensional statistically independent input signals and memory tasks. The extension of this theoretical tool to a multidimensional setup and to parallel architectures is one of the main goals of this paper.

Finally, there are situations in which the moments $\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_y$, $\Gamma(0)$, and $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, necessary to compute the readout layer $(W_{\mathrm{out}}, \boldsymbol{a}_{\mathrm{out}})$ using the equations (2.10)-(2.11), are obtained directly out of finite sample realizations of the teaching signal and of the reservoir output. The use in that context of finite sample empirical estimators carries in its wake an additional error that adds up to the characteristic error (2.12) and that we study later on in Section 3.2.

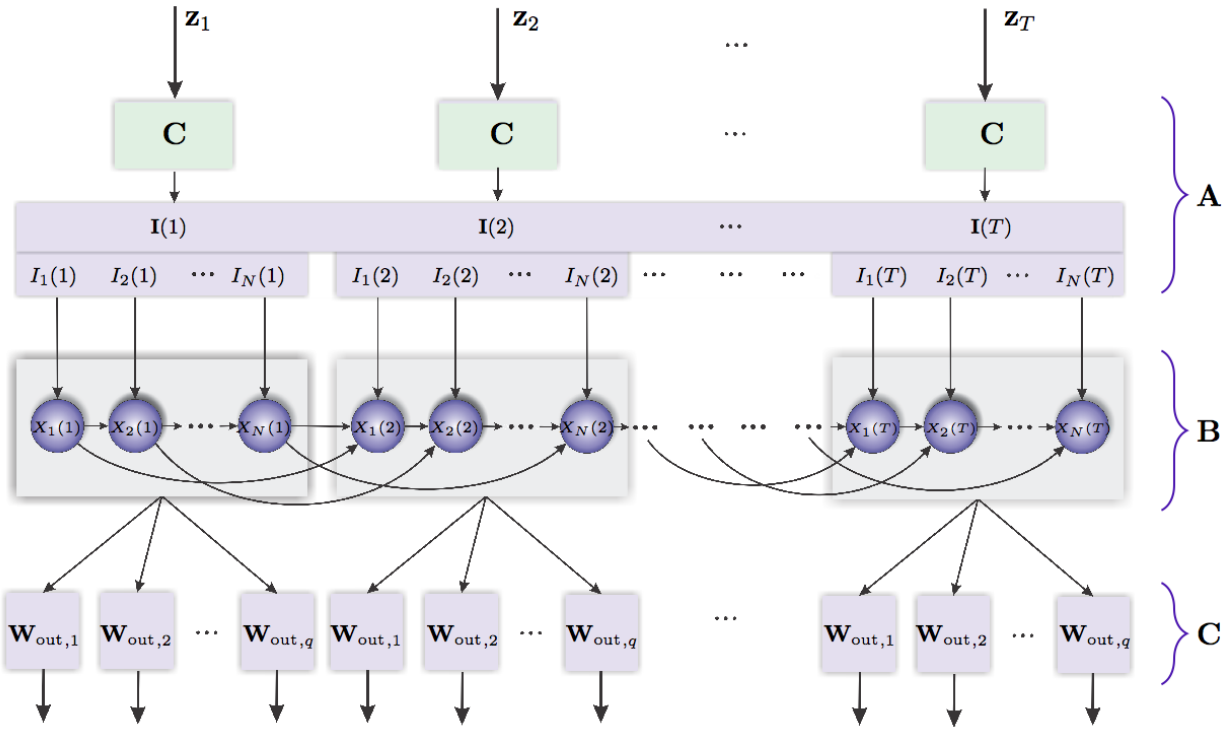

Figure 1: Diagram representing the architecture of a TDR reservoir with a multitask readout. The module A is the input layer, B is a neural diagram representing the discrete-time reservoir dynamics implied by equation (2.6), and C is the multitask readout layer in which each column of the matrix W_{out} accomplishes a different task based on the same reservoir output.

3 Memory capacities for parallel RC architectures and for multidimensional input signals and tasks

This section is the core of the paper and provides estimates for the memory capacity of a TDR in the presence of multidimensional input signals, in the execution of several simultaneous memory tasks, as well as for parallel reservoir architectures. These estimations are based on a generalization of the reservoir model proposed in [Grig 15] that allows the explicit computation of the different elements that constitute the capacity formula (2.13) and makes hence accessible its evaluation.

In the second subsection we study the dependence of the reservoir performance on the length of the teaching signal and on the regularization strength parameter. In particular we formulate an expression that provides an estimation of the added error that is committed in memory tasks when the training is incomplete due to the finiteness of the training sample. This expression can be used, once the training sample has been chosen, to determine beforehand the value of the ridge regularization strength parameter λ in (2.9) that minimizes the training error, thus avoiding costly cross validation procedures.

The following paragraphs provide all these quantitative results in a nutshell in order to make their use as accessible as possible. The reader interested in the details of the models that lead to them is referred to Section 6.

3.1 TDR memory capacities for multidimensional input signals and tasks

In the reservoir computing setup introduced in the previous section, we saw that the memory capacity of such device for a given task H is given by (see (2.10) and (2.13))

$$C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda) = \frac{\operatorname{trace}(\operatorname{Cov}(\mathbf{x}(t), \mathbf{y}(t))^\top (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} (\Gamma(0) + 2\lambda \mathbb{I}_N)(\Gamma(0) + \lambda \mathbb{I}_N)^{-1} \operatorname{Cov}(\mathbf{x}(t), \mathbf{y}(t)))}{\operatorname{trace}\left(\operatorname{Cov}(\mathbf{y}(t), \mathbf{y}(t))\right)}. \tag{3.1}$$

We emphasize that, in order to compute (3.1), the values of $\Gamma(0) := \operatorname{Cov}(\mathbf{x}(t), \mathbf{x}(t))$, $\operatorname{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, and $\operatorname{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ are needed which, for the original reservoir system are, in general, not available. The approximated reservoir model introduced in [Grig 15] and that we generalize in the appendix in Section 6.1 provides an estimation for these three ingredients and hence makes the computation of C_H possible.

Estimation of the memory capacity of single operating TDRs. As we already pointed out in Section 2.2, the value of the matrix $\Gamma(0) := \operatorname{Cov}(\mathbf{x}(t), \mathbf{x}(t))$ depends exclusively on the input signal and the reservoir architecture, while $\operatorname{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, and $\operatorname{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ depend on the specific memory task at hand. In this section we focus on the estimation of $\Gamma(0)$; Section 6.3 in the Appendix shows how to compute the matrices $\operatorname{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, and $\operatorname{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ using the reservoir model in linear and quadratic memory tasks cases. The following result provides an estimate of the memory capacity of a TDR.

Proposition 3.1 *Consider a TDR characterized by a reservoir map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ as in (2.7) and suppose that it operates around a stable fixed point $\mathbf{x}_0 = x_0 \mathbf{i}_N \in \mathbb{R}^N$ of the autonomous system associated to it, that is, $F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) = \mathbf{x}_0$. Suppose that one of the following two conditions holds:*

(i) *the variance of the input signal $\{\mathbf{z}(t)\}$ is sufficiently small,*

(ii) *the variance of the input signal $\{\mathbf{z}(t)\}$ is finite and the norm of the input mask $\mathbf{C}$ is sufficiently small.*

Then, the memory capacity $C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda)$ *of the reservoir for a task* H *can be approximated by the expression* (3.1)*, where* $\Gamma(0)$ *is determined by the equality*

$$\mathrm{vech}(\Gamma(0)) = (\mathbb{I}_{N'} - L_N(A(\mathbf{x}_0, \boldsymbol{\theta}) \otimes A(\mathbf{x}_0, \boldsymbol{\theta}) D_N))^{-1} \mathrm{vech}(\Sigma_\varepsilon),$$

where $N' := \dfrac{1}{2}N(N+1)$ *and* $L_N \in \mathbb{M}_{N', N^2}$, $D_N \in \mathbb{M}_{N^2, N'}$ *are the elimination and the duplication matrices, respectively, and* vech *is the operator introduced in Section 2.1. The matrix* $A(\mathbf{x}_0, \boldsymbol{\theta})$ *is the so called* **connectivity matrix** *of the reservoir at the point* $\mathbf{x}_0$*:*

$$A(\mathbf{x}_0, \boldsymbol{\theta}) := \begin{pmatrix} \Phi & 0 & \dots & 0 & e^{-\xi} \\ e^{-\xi}\Phi & \Phi & \dots & 0 & e^{-2\xi} \\ e^{-2\xi}\Phi & e^{-\xi}\Phi & \dots & 0 & e^{-3\xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ e^{-(N-1)\xi}\Phi & e^{-(N-2)\xi}\Phi & \dots & e^{-\xi}\Phi & \Phi + e^{-N\xi} \end{pmatrix}, \tag{3.2}$$

where $\Phi := (1 - e^{-\xi})\partial_x f(x_0, 0, \boldsymbol{\theta})$ *and* $\partial_x f(x_0, 0, \boldsymbol{\theta})$ *is the first derivative of the nonlinear kernel* f *in* (2.2) *with respect to the first argument and computed at the point* $(x_0, 0, \boldsymbol{\theta})$*. Finally,* Σ_ε *is a symmetric matrix (see* (6.8) *for its explicit expression) whose entries are affine functions of the higher-order moments of the input signal* $\{\mathbf{z}(t)\}$ *(see* (6.7) *for the definition), whose existence and time-independence we assume. The matrices* $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$ *and* $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ *in* (3.1) *depend on the memory task at hand and can be explicitly computed on a case by case basis using the reservoir model (see Section 6.3 in the Appendix for linear and quadratic examples).*

Estimation of the memory capacity of parallel arrays of TDRs. We now consider the parallel time-delay reservoir architecture that was introduced in [Orti 12, Grig 14]. This reservoir design has shown very satisfactory robustness properties with respect to model misspecification and parameter choice (see Section 4 for an explanation and for additional empirical evidence). The basic idea of this approach consists in presenting the input signal to a parallel array of p reservoirs, each of them operating with, in principle, different parameter values $\boldsymbol{\theta}^{(j)}$ and around different stable fixed points $\mathbf{x}_0^{(j)} \in \mathbb{R}^N$, $j \in \{1, \dots, p\}$, of the associated autonomous systems. The concatenation of the outputs of these reservoirs is then used to construct a single readout layer via a ridge regression. Figure 2 provides a diagram representing the parallel reservoir computing architecture. The following result provides an estimate of the memory capacity of a parallel array of TDRs.

Proposition 3.2 *Consider a parallel array of* p *TDRs, each with* N_j *neurons and operating with parameter values* $\boldsymbol{\theta}^{(j)}$ *around stable fixed points* $\mathbf{x}_0^{(j)} \in \mathbb{R}^N$, $j \in \{1, \dots, p\}$*, of the associated autonomous systems. Let* $N^* := N_1 + \cdots + N_p$, $\boldsymbol{\Theta} := (\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(p)})$*, and* $\mathbf{X}_0 := (\mathbf{x}_0^{(1)}, \dots, \mathbf{x}_0^{(p)}) \in \mathbb{R}^{N^*}$*. Suppose that one of the following two conditions holds:*

(i) *the variance of the input signal* $\{\mathbf{z}(t)\}$ *is sufficiently small,*

(ii) *the variance of the input signal* $\{\mathbf{z}(t)\}$ *is finite and the norms of the input masks* $\mathbf{C}^{(j)}$ *are sufficiently small.*

Then, the memory capacity $C_H(\boldsymbol{\Theta}, \{\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(p)}\}, \lambda)$ *of the reservoir for a task* H *can be approximated by the expression* (3.1)*, where* $\Gamma(0)$ *is determined by the equality*

$$\mathrm{vech}(\Gamma(0)) = (\mathbb{I}_{N^{*\prime}} - L_{N^*}(A(\mathbf{X}_0, \boldsymbol{\Theta}) \otimes A(\mathbf{X}_0, \boldsymbol{\Theta}))D_{N^*})^{-1} \mathrm{vech}(\Sigma_\varepsilon^{(\mathbf{X}_0, \boldsymbol{\Theta})})$$

where $N^{*\prime} := \dfrac{1}{2}N^*(N^*+1)$ *and* $L_{N^*} \in \mathbb{M}_{N^{*\prime}, N^{*2}}$, $D_{N^*} \in \mathbb{M}_{N^{*2}, N^{*\prime}}$ *are the elimination and the duplication matrices, respectively, and* vech *is the operator introduced in Section 2.1. The symbol* $A(\mathbf{X}_0, \boldsymbol{\Theta})$

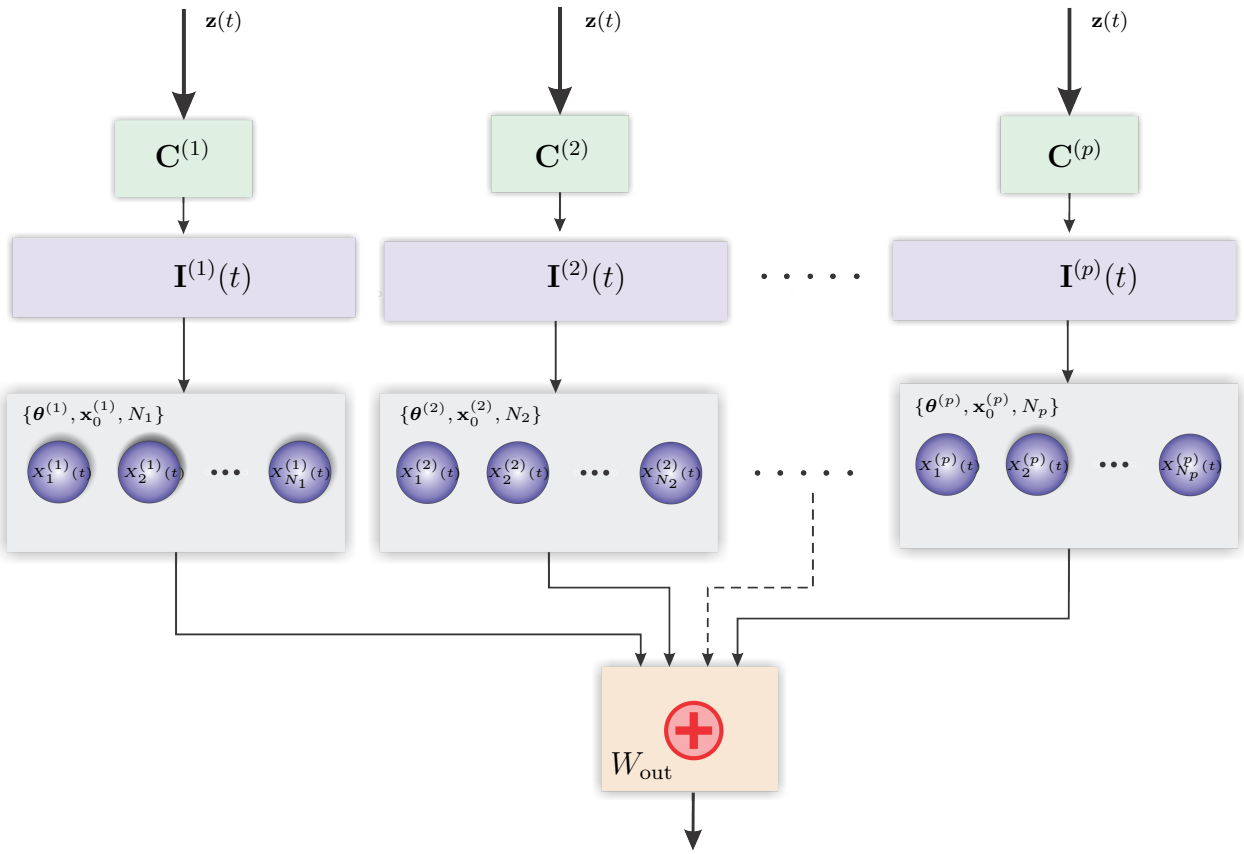

Figure 2: Diagram representing the architecture of a parallel reservoir computer.

denotes a block-diagonal matrix whose diagonal blocks are the connectivity matrices of each of the reservoirs at the stable equilibria $\mathbf{x}_0^{(j)}$, $j \in \{1, \ldots, p\}$. *Finally,* $\Sigma_\varepsilon^{(\mathbf{X}_0, \boldsymbol{\Theta})}$ *is a symmetric matrix (see* (6.24)-(6.25) *for its explicit expression) whose entries are affine functions of the higher-order moments of the input signal* $\{\mathbf{z}(t)\}$ *(see* (6.7) *for the definition), whose existence and time-independence we assume. The matrices* $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$ *and* $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ *in* (3.1) *depend on the memory task at hand and can be explicitly computed on a case by case basis using the proposed approximated reservoir model (see Section 6.3 in the Appendix for linear and quadratic memory tasks examples).*

3.2 The impact of the teaching signal size on the reservoir performance

In the preceding section we evaluated the reservoir characteristic error or, equivalently, its capacity, in terms of second order moments of the input signal and the reservoir output. There are situations in which those moments are obtained directly out of finite sample realizations of the teaching signal and of the reservoir output using empirical estimators. That approach introduces an estimation error in the readout layer $(W_{\text{out}}, \boldsymbol{a}_{\text{out}})$ that adds to the characteristic error (2.12). The quantification of that error is the main goal of this section.

First, since this error depends on the value of the regularizing constant λ, the pair $(W_\lambda, \boldsymbol{a}_\lambda)$ will denote in what follows the optimal readout layer $(W_{\text{out}}, \boldsymbol{a}_{\text{out}})$ given by (2.10)-(2.11) for a fixed value of the parameter λ; the particular case $\lambda = 0$ will be simply written as $W := W_0$. With this notation, the

characteristic error in (2.12) can be rewritten as

$$\mathrm{MSE}_{\mathrm{char},\lambda} = \mathrm{trace}(\mathrm{Cov}\,(\mathbf{y}(t),\mathbf{y}(t)) - W_\lambda^\top(\Gamma(0)+2\lambda\mathbb{I}_N)W_\lambda), \tag{3.3}$$

and we emphasize that it corresponds to the error committed by the reservoir when the readout layer $(W_\lambda, \boldsymbol{a}_\lambda)$ has been perfectly estimated.

Consider now $\{\mathbf{x}(1),\mathbf{x}(2),\ldots,\mathbf{x}(T)\}$ and $\{\mathbf{y}(1),\mathbf{y}(2),\ldots,\mathbf{y}(T)\}$ samples of size T of the reservoir output $\{\mathbf{x}(t)\}_{t\in\mathbb{N}}$ and the teaching signal $\{\mathbf{y}(t)\}_{t\in\mathbb{N}}$ processes. We concatenate horizontally these observations and we obtain the matrices $X := (\mathbf{x}(1)||\mathbf{x}(2)||\ldots||\mathbf{x}(T)) \in \mathbb{M}_{N,T}$ and $Y := (\mathbf{y}(1)||\mathbf{y}(2)||\ldots||\mathbf{y}(T)) \in \mathbb{M}_{q,T}$. We now quantify the loss of memory capacity caused by using in the RC not the optimal readout layer $(W_\lambda, \boldsymbol{a}_\lambda)$ given by (2.10)-(2.11) but an empirical estimation $(\widehat{W}_\lambda, \widehat{\boldsymbol{a}}_\lambda)$ based on X and Y. More specifically, for a fixed λ and samples X and Y, we produce an estimation $(\widehat{W}_\lambda, \widehat{\boldsymbol{a}}_\lambda)$ of $(W_\lambda, \boldsymbol{a}_\lambda)$ by using the sample based estimators

$$\widehat{\boldsymbol{\mu}}_x := \frac{1}{T}X\mathbf{i}_T, \qquad \widehat{\boldsymbol{\mu}}_y := \frac{1}{T}Y\mathbf{i}_T, \qquad \widehat{\Gamma}(0) := \frac{1}{T}XX^\top - \widehat{\boldsymbol{\mu}}_x\widehat{\boldsymbol{\mu}}_x^\top = \frac{1}{T}XAX^\top, \tag{3.4}$$

$$\widehat{\mathrm{Cov}\,(\mathbf{y}(t),\mathbf{y}(t))} = \frac{1}{T}YY^\top - \widehat{\boldsymbol{\mu}}_y\widehat{\boldsymbol{\mu}}_y^\top = \frac{1}{T}YAY^\top, \tag{3.5}$$

$$\widehat{\mathrm{Cov}\,(\mathbf{x}(t),\mathbf{y}(t))} = \frac{1}{T}XY^\top - \widehat{\boldsymbol{\mu}}_x\widehat{\boldsymbol{\mu}}_y^\top = \frac{1}{T}XAY^\top, \quad \text{with} \quad A := \mathbb{I}_T - \frac{1}{T}\mathbf{i}_T\mathbf{i}_T^\top, \tag{3.6}$$

in (2.10)-(2.11) which yield:

$$\widehat{W}_\lambda = (\widehat{\Gamma}(0)+\lambda\mathbb{I}_N)^{-1}\widehat{\mathrm{Cov}\,(\mathbf{x}(t),\mathbf{y}(t))} = (XAX^\top+\lambda T\mathbb{I}_N)^{-1}XAY^\top, \tag{3.7}$$

$$\widehat{\boldsymbol{a}}_\lambda = \widehat{\boldsymbol{\mu}}_y - \widehat{W}_\lambda^\top\widehat{\boldsymbol{\mu}}_x = \frac{1}{T}(Y-\widehat{W}_\lambda^\top X)\mathbf{i}_T, \tag{3.8}$$

and determine a finite sample ridge regression estimator. The **total mean square training error** committed by the reservoir computer when using arbitrary training pairs (X,Y) of length T and the empirically estimated readout layers $(\widehat{W}_\lambda, \widehat{\boldsymbol{a}}_\lambda)$ associated to them is defined as

$$\begin{aligned}\mathrm{MSE}_{\mathrm{total},\lambda} &:= \frac{1}{T}\mathrm{trace}\left(\mathrm{E}\left[\left(Y-\widehat{\boldsymbol{a}}_\lambda\mathbf{i}_T^\top-\widehat{W}_\lambda^\top X\right)^\top\left(Y-\widehat{\boldsymbol{a}}_\lambda\mathbf{i}_T^\top-\widehat{W}_\lambda^\top X\right)\right]\right)\\ &= \frac{1}{T}\mathrm{trace}\left(\mathrm{E}\left[\left(Y-\widehat{\mathcal{W}}_\lambda^\top\mathcal{X}\right)^\top\left(Y-\widehat{\mathcal{W}}_\lambda^\top\mathcal{X}\right)\right]\right),\end{aligned} \tag{3.9}$$

where

$$\widehat{\mathcal{W}}_\lambda := \left(\widehat{\boldsymbol{a}}_\lambda \mid \widehat{W}_\lambda^\top\right)^\top \quad \text{and} \quad \mathcal{X} := \left(\mathbf{i}_T \mid X^\top\right)^\top. \tag{3.10}$$

The following proposition, whose proof can be found in Appendix 6.4.2, uses the properties of the ridge estimator to quantitatively evaluate the total mean square reservoir error in (3.9), conditional on a reservoir output X. This statement requires certain technical stationarity and independence assumptions that are spelled out in Appendix 6.4.

Proposition 3.3 *Given a reservoir output X of size T, the total mean square reservoir training error conditional on X and for any teaching signal Y, is given by*

$$\mathrm{MSE}_{\mathrm{total},\lambda} \mid X = \mathrm{trace}\,(\Sigma) + \frac{1}{T}\mathrm{trace}\left[\mathrm{trace}\,(\Sigma)\left(R_\lambda\mathcal{X}A\mathcal{X}^\top\left(R_\lambda\mathcal{X}\mathcal{X}^\top-2\mathbb{I}_{N+1}\right)\right)+\lambda^2T^2R_\lambda\mathcal{W}\mathcal{W}^\top R_\lambda\mathcal{X}\mathcal{X}^\top\right], \tag{3.11}$$

where N is the number of neurons of the reservoir, $\mathcal{X}$ is defined in (3.10), and $\mathcal{W} := \left(\boldsymbol{a}||W^\top\right)^\top$ with $W := \Gamma(0)^{-1}\mathrm{Cov}\,(\mathbf{x}(t),\mathbf{y}(t))$ and $\boldsymbol{a} := \boldsymbol{\mu}_y - W^\top\boldsymbol{\mu}_x$. Finally, $R_\lambda := (\mathcal{X}A\mathcal{X}^\top+\lambda T\mathbb{I}_{N+1})^{-1}$ and $\Sigma := \mathrm{Cov}\,(\mathbf{y}(t),\mathbf{y}(t)) - W^\top\Gamma(0)W$.

Practical use of the total training error formula (3.11). The total training error formula (3.11) can be implemented in practice by using the approximation of the ingredients necessary to evaluate it (namely $\Gamma(0)$, $\mathrm{Cov}\,(\mathbf{y}(t),\mathbf{y}(t))$, and $\mathrm{Cov}\,(\mathbf{x}(t),\mathbf{y}(t))$) provided in the propositions 3.1 and 3.2. Another possibility to proceed consists of using the empirical estimators provided in (3.4)-(3.6).

An important application of the error formula (3.11) is the possibility to determine the optimal value of the regularization strength λ without going through costly cross validation procedures. Indeed, once a training sample has been fixed and a reservoir output X is available, an optimal value of λ can be determined by minimizing (3.11) or, equivalently, by picking the values λ that make its gradient vanish. A straightforward computation using the fact R_λ commutes with $\mathcal{X}A\mathcal{X}^\top$ shows that the optimal values λ are characterized by the roots of the equation:

$$\mathrm{trace}\Big[\mathrm{trace}(\Sigma)\left(2R_\lambda^2\mathcal{X}A\mathcal{X}^\top\left(\mathbb{I}_{N+1}-R_\lambda\mathcal{X}\mathcal{X}^\top\right)\right) + \lambda T R_\lambda\left(2\mathcal{W}\mathcal{W}^\top-\lambda T R_\lambda\mathcal{W}\mathcal{W}^\top-\lambda T\mathcal{W}\mathcal{W}^\top R_\lambda\right)\cdot R_\lambda\mathcal{X}\mathcal{X}^\top\Big]=0.$$

We finally point out that (3.11) characterizes only the total training error in situations in which the autocovariance $\Gamma(0)$ is regular, that is, it has no zero eigenvalues. An extension to the singular case in which the testing or generalization error [Hast 13] is also computed has been carried out in [Grig 16].

Ergodicity and the convergence of the total training error to the characteristic error. A result presented later on in the appendix (see Proposition 6.2) shows that the variance of the estimator $(\widehat{W}_\lambda,\widehat{\boldsymbol{a}}_\lambda)$ of $(W_\lambda,\boldsymbol{a}_\lambda)$ tends to zero as the sample size tends to infinity. This means that the estimation of the readout layer is perfect in the large sample size limit, which leads us to expect that the total errors $\mathrm{MSE}_{\mathrm{total},\lambda}\mid X$ converge to the characteristic error $\mathrm{MSE}_{\mathrm{char},\lambda}$ in that asymptotic regime. This fact can be rigorously proved under the hypothesis of ergodicity for second moments (see page 47 in [Hami 94]) of the joint process $\{(\mathbf{x}(t),\mathbf{y}(t))\}_{t\in\mathbb{N}}$, in which case

$$\mathrm{MSE}_{\mathrm{total},\lambda}\mid X\overset{T\to\infty}{\longrightarrow}\mathrm{MSE}_{\mathrm{char},\lambda}.$$

This statement is the subject of Proposition 6.6 in the Appendix 6.4.

4 Empirical study

The reservoir model for multidimensional computing and parallel architectures

The goal of this section is twofold. First, we assess the ability of the results introduced in Proposition 3.1 for the multidimensional setup to produce good estimates of the memory capacity of the original reservoir, both for the continuous and the discrete-time configurations. Our study shows that there is a good match between the performances of the model and of the actual reservoir and hence indicates that the explicit expression of the reservoir capacity coming from the model can be used to find, for a given multidimensional task with multidimensional input signal, parameters and input masks for the original system that optimize its performance.

Second, we use the parallel reservoir performance estimates in Proposition 3.2 in order to verify various universality properties of this reservoir architecture that were already documented in [Grig 14]. Universality refers to the ability of the reservoir to perform well for a variety of tasks with a given set of parameters that are kept fixed and that have been optimized for a single one (task misspecification) or, alternatively, when the optimal performance for a given task is not very sensitive to the reservoir parameters. This property is of paramount importance at the time of implementing the simultaneous execution of several tasks. This feature, sometimes referred to as **real-time multitasking** [Maas 11], is usually presented as one of the most prominent computational advantages of RC.

Evaluation of the TDR performance in the processing of multidimensional signals.

In the first empirical experiment we present a quadratic memory task to an individually operating TDR. More explicitly, we inject in the reservoir a three dimensional independent input signal $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$, $\mathbf{z}(t)\in\mathbb{R}^3$ with mean zero and covariance matrix Σ_z, and we study its ability to reconstruct the signal $y(t)=\sum_{h=0}^{3}\sum_{i,j=1}^{3}z_i(t-h)z_j(t-h)\in\mathbb{R}$. In the terminology introduced later on in the Appendix 6.3, this exercise amounts to a quadratic task characterized by the vector $Q\in\mathbb{R}^{78}$ defined by $Q:=(\mathrm{vec}(Q^*))^\top D_{12}$, with D_{12} the duplication matrix in dimension twelve and $Q^*\in\mathbb{S}_{12}$ a block diagonal matrix with four matrices $\mathbf{i}_3\mathbf{i}_3^\top$ as its diagonal blocks. Indeed, if $\mathbf{z}^3(t):=(\mathbf{z}(t),\mathbf{z}(t-1),\mathbf{z}(t-2),\mathbf{z}(t-3))$, it is clear that

$$\begin{aligned} y(t) &= \mathbf{z}^3(t)^\top Q^*\mathbf{z}^3(t)=\mathrm{trace}\left(\mathbf{z}^3(t)^\top Q^*\mathbf{z}^3(t)\right)=(\mathrm{vec}(Q^{*\top}))^\top\left(\mathbf{z}^3(t)\otimes\mathbb{I}_{12}\right)\mathrm{vec}\left(\mathbf{z}^3(t)\right)\\ &= (\mathrm{vec}(Q^*))^\top\mathrm{vec}\left(\mathbf{z}^3(t)\mathbf{z}^3(t)^\top\right)=(\mathrm{vec}(Q^*))^\top D_{12}\mathrm{vech}\left(\mathbf{z}^3(t)\mathbf{z}^3(t)^\top\right)=Q\cdot\mathrm{vech}\left(\mathbf{z}^3(t)\mathbf{z}^3(t)^\top\right). \end{aligned}$$

In order to tackle this multidimensional task we use two twenty neuron TDRs constructed using the Mackey-Glass (2.3) (with $p=2$) and the Ikeda (2.4) nonlinear kernels. In Figures 3 and 4 we depict the error surfaces exhibited by both RCs in discrete and continuous time as a function of the distance between neurons and the feedback gain η and using fixed input gains γ whose values are indicated in the legends. Those error surfaces are computed using Monte Carlo simulations. At the same time we compute the error surfaces produced by the corresponding reservoir model and by evaluating the explicit capacity formula in that case.

The resulting figures exhibit a remarkable similarity that had already been observed for scalar input signals in [Grig 15]. More importantly, the figures show the ability of the theoretical formula based on the reservoir model to locate the regions in parameter space for which the reservoir capacity has local maxima and that can be used as preliminary estimates in the search for the parameter values of the original system for which capacity reaches a global maximum.

The distinction between local and global extrema should be considered very carefully at the time of using the model because what is a global maximum in performance for the model can be just a local one for the original system. The examples described in Figures 3 and 4 illustrate this point very well. Indeed, in the Mackey-Glass kernel case in Figure 3, the error function of the continuous time system exhibits a local minimum in the (η,d) plane for the value $(1.24,0.58)$ (30.81% associated error) and a global minimum for $(2,0.02)$ (27.71% associated error); the reservoir model exhibits minima in nearby points ($(1.2,0.88)$ with 17.48% associated error and $(2,0.03)$ for a 29.53% error) but what was a local minimum for the original system is a global one for the model and vice versa. In the Ikeda kernel case in Figure 4, the local (respectively, global) minimum of the error function of the continuous time system coincides approximately with the local (respectively, global) minimum of the model. Indeed, the original system exhibits a global (respectively, local) minimum in the (η,d) plane for the value $(0.84,0.03)$ with 15.88% associated error (respectively, for $(0.64,0.97)$, with 26.95% associated error); the model exhibits a global (respectively, local) minimum in the (η,d) plane for the value $(0.84,0.03)$ with 9.93% associated error (respectively, for $(0.52,0.98)$, with 10.67% associated error). Note that in both cases, the reservoir performances at the optimal values are lower than those corresponding to the model; we emphasize that, as we have noticed in unreported simulations, this is not always the case.

Robustness properties of the parallel reservoir architecture.

We now we use the reservoir performance estimates in Proposition 3.2 to study the robustness properties of the parallel architecture with respect to parameter choice and misspecification task.

(i) Parallel TDR configurations and robustness with respect to the choice of reservoir parameters. An interesting feature of parallel TDR architectures that was observed in [Grig 14] is that optimal performance has a reduced sensitivity to the choice of reservoir parameters when compared

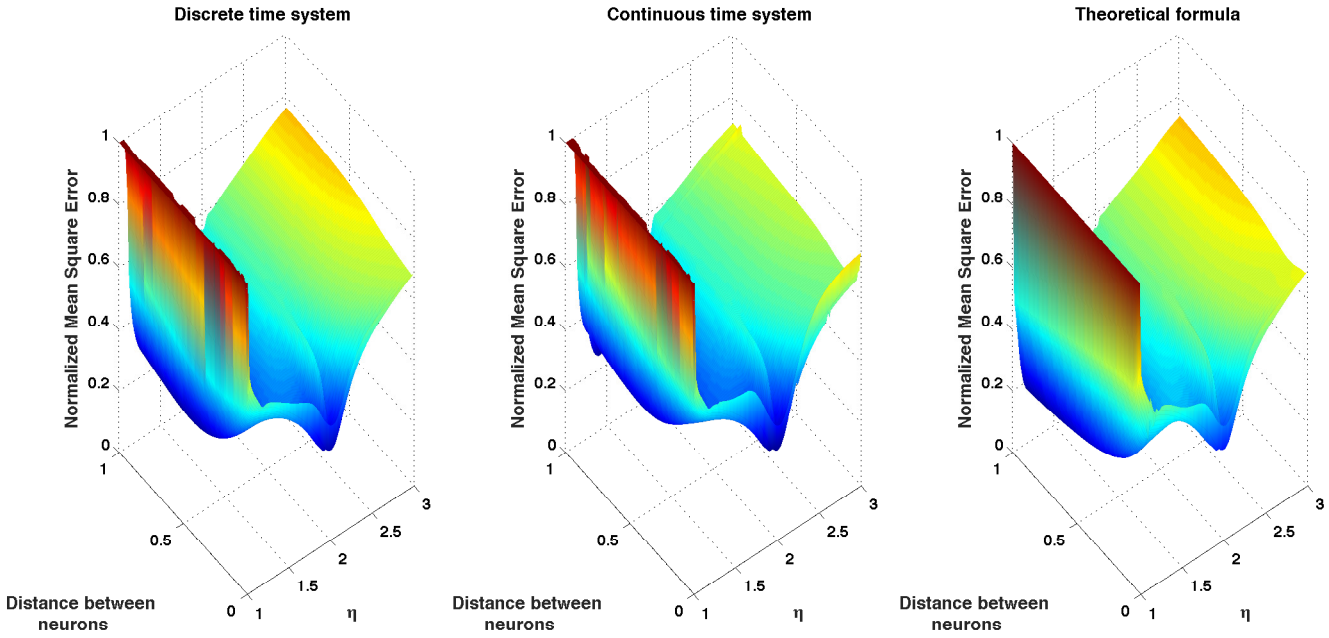

Figure 3: Normalized mean square error surfaces exhibited by an individually operating TDR constructed using a nonlinear Mackey-Glass kernel (2.3) with $p = 2$ performing a 3-lag quadratic memory task on a 3-dimensional independent mean zero input signal with covariance matrix Σ_z given by $\text{vech}(\Sigma_z) = (0.0016, 0.0012, 0.0008, 0.0017, 0.0002, 0.0018)$. In these figures the input gain $\gamma = 0.6163$ is kept constant. The values of the input mask $\mathbf{C} \in \mathbb{M}_{20,3}$ are chosen randomly using a uniform distribution in the interval $[-1, 1]$. The left and middle panels show the error surfaces exhibited by the discrete and continuous time reservoirs computed via Monte Carlo simulations using 50,000 points for both the training and the testing. The right panel shows the error produced by the reservoir model and computed evaluating the explicit capacity formula (3.1).

to that of individually operating reservoirs. In order to provide additional evidence of this fact, we have constructed parallel pools of 2, 5, 10, and 20 parallel Mackey-Glass-based TDRs with $p = 2$ and we present to them a 9-lag quadratic memory task is $y(t) = \sum_{i=0}^{9} z(t-i)^2$ that, in the terminology introduced later on in Appendix 6.3 corresponds to the quadratic task characterized by the vector $Q = (\text{vec}(\mathbb{I}_{10}))^{\top} D_{10}$, with D_{10} the duplication matrix in dimension ten. For each of these parallel TDR architectures, as well as for an individually operating TDR, we will vary the number of the constituting neurons from 20 to 100. For each of these resulting configurations we randomly draw 1000 sets of input masks with entries uniformly distributed in the interval $[-3, 3]$ and reservoir parameters and distance between neurons d, also uniformly distributed in the intervals $\eta \in [1, 3]$, $\gamma \in [-3, 3]$, and $d \in (0, 1)$.

Figure 5 provides the box plots corresponding to the distributions of normalized mean squared errors obtained with each configuration by making the input masks and reservoir parameter values randomly vary, all of them computed using the capacity formulas associated to the parallel reservoir model (6.30). This figure provides striking evidence of the facts that first, the parallel architecture performs on average better (even though the optimal performance may be attained just by a single reservoir) and, more importantly, that this performance is not sensitive to the choice of reservoir parameters and input mask. Indeed the negligeable variance in the box plots associated to the 20 parallel reservoirs architecture means that the performance for the task considered is good regardless the reservoir parameter sets and mask used to achieve it.

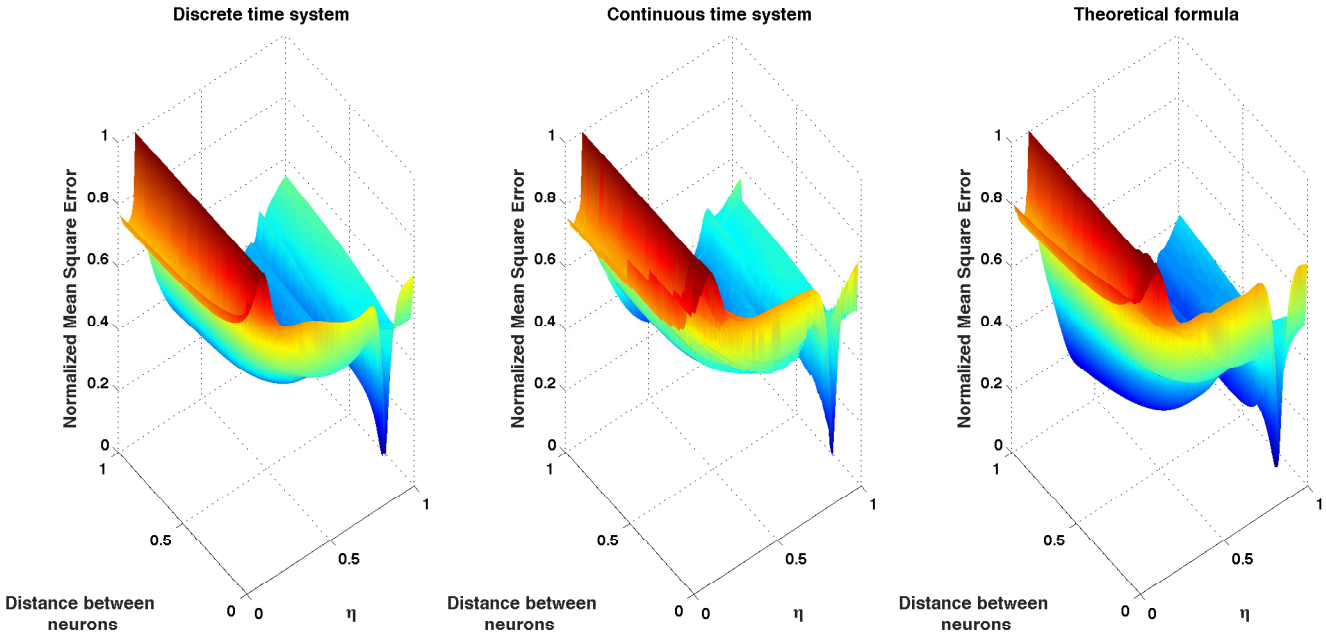

Figure 4: Normalized mean square error surfaces exhibited by an individually operating TDR constructed using a nonlinear Ikeda kernel (2.4) performing a 3-lag quadratic memory task on a 3-dimensional independent mean zero input signal with covariance matrix Σ_z given by $\text{vech}(\Sigma_z) = (0.005, 0.0046, 0.0041, 0.0042, 0.0037, 0.004)$. In these figures the input gain $\gamma = 0.3724$ and the phase shift $\phi = 0.7356$ are kept constant. The values of the input mask $\mathbf{C} \in \mathbb{M}_{20,3}$ are chosen randomly using a uniform distribution on the interval $[-1, 1]$. The left and middle panels show the error surfaces exhibited by the discrete and continuous time reservoirs computed via Monte Carlo simulations using 50,000 points for both the training and the testing. The right panel shows the error produced by the reservoir model and computed evaluating the explicit capacity formula (3.1).

(ii) Parallel TDR configurations and robustness with respect to memory task misspecification. The term task misspecification refers to the effect observed when the reservoir is optimized for a given task, the corresponding parameters are kept, and the reservoir capacity is measured for a different task. Robustness with respect to task misspecification, that is, a low sensitivity of the optimal parameters values to the task at hand is a measure of the universality properties of the device.

In the next experience we see how, given a specific memory task and given a parallel array of TDRs or an individually operating one that have been optimized for that particular task, the performance of the different configurations degrades when the task is modified but the reservoir parameters are left unchanged. This is what we call memory task misspecification. We use again parallel pools of 1, 2, 5, 10, and 20 Mackey-Glass-based TDRs with neurons ranging from 10 to 100. For each of those configurations, we choose parameters that optimize their performance with respect to the 3-lag quadratic memory task specified by the matrix $Q = (\text{vec}(\mathbb{I}_4))^\top D_4$ and with respect to a one-dimensional independent input signal with mean zero and variance 0.0001. Once the optimal parameters for each configuration have been found using the nonlinear capacity function based on the reservoir model (6.30), we fix them and we subsequently expose the corresponding reservoirs to random memory tasks of different specifications, namely, 1000 different 9-lag quadratic tasks of the form $Q = (\text{vec}(Q_{10}^*))^\top D_{10}$, where $Q^* \in \mathbb{M}_{10}$ is a randomly generated diagonal matrix with entries drawn from the uniform distribution on the interval $[-10, 10]$. Figure 6 contains the box plots corresponding to the performance distributions of the different configurations. In this case, parallel architectures offer an improvement of capacity and robustness when

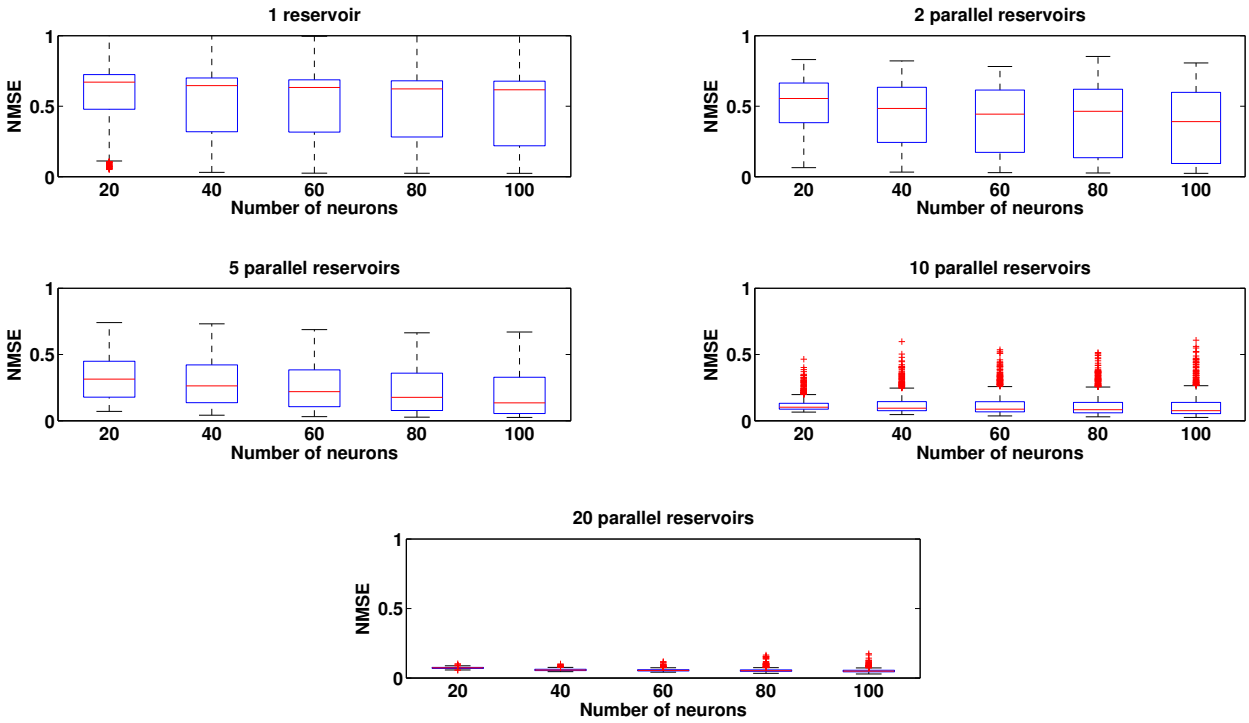

Figure 5: Normalized error distributions for parallel arrays of Mackey-Glass-based TDRs (with $p = 2$) in a 9-lag quadratic memory task when the kernel parameters and the input mask are varied randomly.

compared to the single reservoir design. The most visible improvement is obtained in this case when using a parallel pool of two reservoirs.

5 Conclusions

In this paper we provided quantitative results that allow the evaluation of the memory capacity of several reservoir computing (RC) architectures constructed via the sampling of the solutions of time-delay differential equations and which are referred to as time-delay reservoirs (TDRs).

More explicitly, we generalize the reservoir model introduced in [Grig 15] in order to supply ready-to-use approximate formulas for the memory capacity of TDRs in the framework of multidimensional input signals and real-time multitasking, known to be one of the most prominent computational advantages of RC. Additionally, we have extended these results to provide estimates on the memory capacity of parallel arrays of reservoir computers, a reservoir architecture that has been empirically shown to exhibit improved information processing performances.

We have also quantitatively studied the impact of finite sample training on the decrease of reservoir capacity and we have provided a formula that evaluates the reservoir error in the presence of an imperfect training carried out using a finite size training sample. The resulting formula can be used in passing to determine, for a given training sample, the value of the ridge regularization strength that optimizes the reservoir performance.

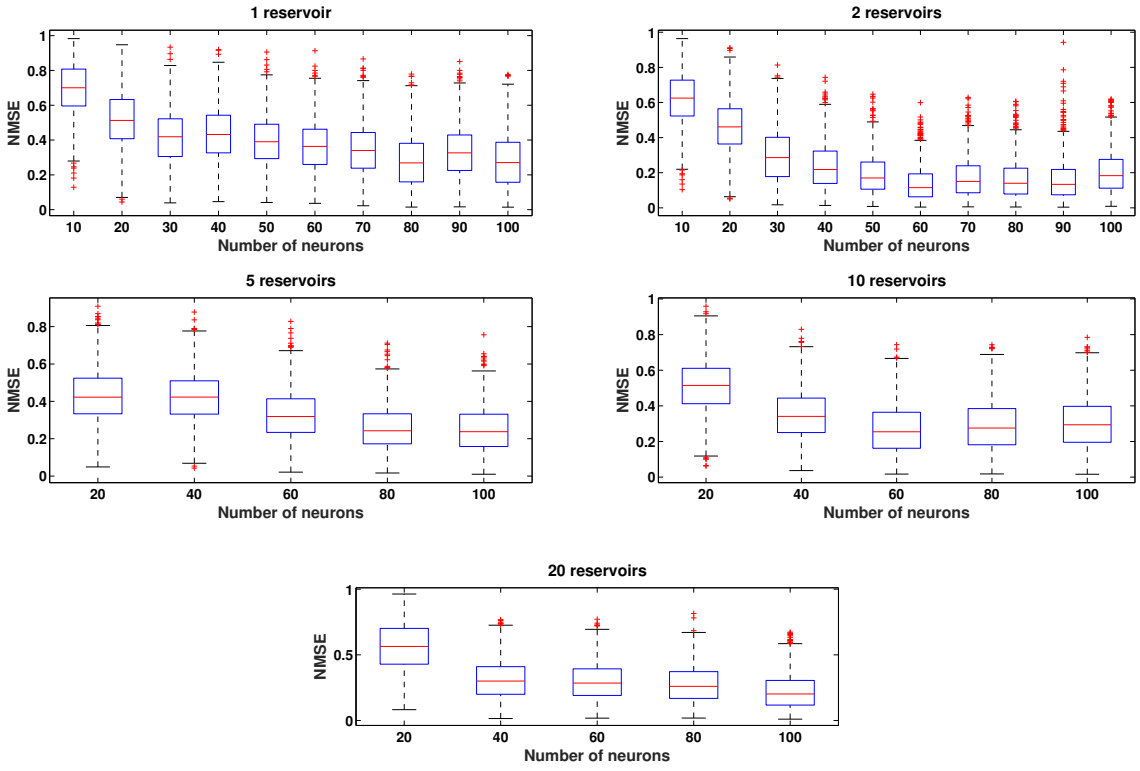

Figure 6: Capacity robustness of parallel arrays of Mackey-Glass kernel based TDRs ($p = 2$) with respect to task misspecification. The box plots report the distribution of normalized mean square errors committed in the execution of 1000 randomly generated 9-lag diagonal quadratic memory tasks by different TDR configurations that had been initially optimized for a 3-lag quadratic memory task.

The paper concludes with an empirical study in which we have shown the adequacy of our theoretical results with the empirical performances exhibited by TDRs in the execution of various nonlinear tasks with multidimensional inputs and where we confirmed, using the approximating model, the robustness properties of the parallel reservoir architecture with respect to task misspecification and parameter choice that had already been previously documented in the literature.

6 Appendices

The following appendices provide details on the results presented in Section 3 and, in particular, on the reservoir models that justify the propositions 3.1 and 3.2. Those results are based on generalizations of the reservoir model proposed in [Grig 15] that accommodate the treatment of multidimensional input signals and the execution of several simultaneous memory tasks, as well as parallel reservoir architectures. The main virtue of the reservoir model is that it allows for the explicit computation of the different elements that constitute the capacity formula (2.13) making hence accessible its evaluation.

The last appendix contains detailed proofs of the properties of the finite sample ridge estimator

that are needed to conclude the results presented in Section 6.4 on the dependence of the reservoir performance on the length of the teaching signal and the regularization strength parameter.

6.1 The reservoir model for multidimensional input signals

The reservoir model introduced in [Grig 15] is based on the observation that optimal reservoir performance is frequently attained when the reservoir is functioning in a neighborhood of an asymptotically stable equilibrium of the autonomous system associated to (2.2). This feature suggests the possibility of approximating the reservoir by its partial linearization at that stable fixed point with respect to the delayed self feedback but keeping the nonlinearity at the level of the input signal injection. This observation motivated in [Grig 15] an in-depth study of the stability properties of the equilibria x_0 of the time-delay differential equation (2.2) and of the corresponding fixed points $\mathbf{x}_0 = x_0\mathbf{i}_N \in \mathbb{R}^N$ of the discrete-time approximation (2.7), both considered in the autonomous regime, that is, when $I(t) = 0$ in (2.2) and $\mathbf{I}(t) = \mathbf{0}_N$ in (2.7), respectively. In particular, it was shown that (see Corollary D.5 and Theorem D.10 in the Supplementary Material in [Grig 15]) that $|\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1$ is a sufficient condition for the asymptotic stability of $x_0 \in \mathbb{R}$ and $\mathbf{x}_0 = x_0\mathbf{i}_N \in \mathbb{R}^N$ in the continuous and discrete-time cases, respectively, which given a particular kernel f allows for the identification of specific regions in parameter space in which stability is guaranteed (see Corollaries D.6 and D.7 of the Supplementary Material in [Grig 15] for the Mackey-Glass and Ikeda kernel cases).

Consider now a discrete-time TDR described by a reservoir map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ as in (2.7). Let $\mathbf{x}_0 \in \mathbb{R}^N$ be a stable fixed point of the autonomous system associated to (2.7), that is, $F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) = \mathbf{x}_0$. In order to write down the approximate reservoir model as in [Grig 15] we start by approximating (2.7) by its partial linearization at $\mathbf{x}_0$ with respect to the delayed self feedback and by the Rth-order Taylor series expansion on the input forcing $\mathbf{I}(t) \in \mathbb{R}^N$. We obtain the following expression:

$$\mathbf{x}(t) = F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) + A(\mathbf{x}_0, \boldsymbol{\theta})(\mathbf{x}(t-1) - \mathbf{x}_0) + \boldsymbol{\varepsilon}(t), \tag{6.1}$$

where $F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ is the reservoir map evaluated at the point $(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ and $A(\mathbf{x}_0, \boldsymbol{\theta}) := D_{\mathbf{x}} F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ is the first derivative of F with respect to its first argument, computed at the point $(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$. The vector $\boldsymbol{\varepsilon}(t) \in \mathbb{R}^N$, $t \in \mathbb{Z}$, in (6.1) is obtained out of the Taylor series expansion of $F(\mathbf{x}(t), \mathbf{I}(t), \boldsymbol{\theta})$ in (2.7) on $\mathbf{I}(t)$ up to some fixed order $R \in \mathbb{N}$. For each $r \in \{1, \ldots, N\}$ its rth component can be written as

$$\varepsilon_r(t) = (1 - e^{-\xi}) \sum_{i=1}^{R} \frac{1}{i!} (\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta}) \sum_{j=1}^{r} e^{-(r-j)\xi} I_j(t)^i, \tag{6.2}$$

where $(\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta})$ is the ith order partial derivative of the nonlinear reservoir kernel map f in (2.2) with respect to the second argument $I(t)$ computed at the point $(x_0, 0, \boldsymbol{\theta})$. Finally, $A(\mathbf{x}_0, \boldsymbol{\theta})$ is called the **connectivity matrix** of the reservoir at the point $\mathbf{x}_0$ and has the following explicit form

$$A(\mathbf{x}_0, \boldsymbol{\theta}) = \begin{pmatrix} \Phi & 0 & \ldots & 0 & e^{-\xi} \\ e^{-\xi}\Phi & \Phi & \ldots & 0 & e^{-2\xi} \\ e^{-2\xi}\Phi & e^{-\xi}\Phi & \ldots & 0 & e^{-3\xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ e^{-(N-1)\xi}\Phi & e^{-(N-2)\xi}\Phi & \ldots & e^{-\xi}\Phi & \Phi + e^{-N\xi} \end{pmatrix}, \tag{6.3}$$

where $\Phi := (1 - e^{-\xi})\partial_x f(x_0, 0, \boldsymbol{\theta})$ and $\partial_x f(x_0, 0, \boldsymbol{\theta})$ is the first derivative of the nonlinear kernel f in (2.2) with respect to the first argument and computed at the point $(x_0, 0, \boldsymbol{\theta})$.

Suppose now that the input signal is a collection of n-dimensional independent and identically distributed random variables $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}} \sim \text{IID}(\mathbf{0}_n, \Sigma_z)$, $\Sigma_z \in \mathbb{S}_n^+$, and that we take as input mask the

matrix $\mathbf{C} \in \mathbb{M}_{N,n}$. Since for each $t \in \mathbb{Z}$ the input forcing $\mathbf{I}(t) \in \mathbb{R}^N$ is constructed via the assignment $\mathbf{I}(t) := \mathbf{C}\mathbf{z}(t)$ we have that $\{\mathbf{I}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\mathbf{0}_N, \Sigma_I)$, with $\Sigma_I := \mathbf{C}\Sigma_z\mathbf{C}^\top$. It follows then that $I_j(t) = \sum_{k=1}^n \mathbf{C}_{jk} z_k(t)$ which substituted in (6.2) yields that

$$\boldsymbol{\varepsilon}(t) = (1-e^{-\xi}) \begin{pmatrix} V_R\left(\mathbf{z}(t), \{\mathbf{C}_{1,\cdot}\}, x_0, \boldsymbol{\theta}\right) \\ V_R\left(\mathbf{z}(t), \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,2\}}, x_0, \boldsymbol{\theta}\right) \\ \vdots \\ V_R\left(\mathbf{z}(t), \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,N\}}, x_0, \boldsymbol{\theta}\right) \end{pmatrix}, \tag{6.4}$$

with the polynomials

$$V_R\left(\mathbf{z}(t), \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,r\}}, x_0, \boldsymbol{\theta}\right) := \sum_{i=1}^R (\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta}) \sum_{j=1}^r e^{-(r-j)\xi} \sum_{k_1+\cdots+k_n=i} \frac{1}{k_1!\ldots k_n!} \prod_{s=1}^n \mathbf{C}_{js}^{k_s} \cdot \prod_{s=1}^n z_s(t)^{k_s}. \tag{6.5}$$

The symbol $\{\mathbf{C}_{j,\cdot}\}$, $j \in \{1,\ldots N\}$, denotes the set of all the entries in the jth row of the input mask matrix $\mathbf{C}$. The assumption $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\mathbf{0}_n, \Sigma_z)$ implies that $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{Z}}$ is also a family of N-dimensional independent and identically distributed random variables with mean $\boldsymbol{\mu}_\varepsilon$ given by

$$(\boldsymbol{\mu}_\varepsilon)_r = (1-e^{-\xi}) \sum_{i=1}^R (\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta}) \sum_{j=1}^r e^{-(r-j)\xi} \sum_{k_1+\cdots+k_n=i} \frac{1}{k_1!\ldots k_n!} \prod_{s=1}^n \mathbf{C}_{js}^{k_s} \cdot \mu_{k_1,\ldots,k_n}(\mathbf{z}), \tag{6.6}$$

where

$$\mu_{k_1,\ldots,k_n}(\mathbf{z}(t)) := \mathrm{E}\left[\prod_{s=1}^n z_s(t)^{k_s}\right] \tag{6.7}$$

denotes a higher-order moment of $\mathbf{z}(t) \in \mathbb{R}^n$ whose existence and time-independence we assume for values $k_1, \ldots, k_n$ such that $k_1+\cdots+k_n \le 2R$. Additionally, the covariance matrix $\Sigma_\varepsilon := \mathrm{E}\left[(\boldsymbol{\varepsilon}(t) - \boldsymbol{\mu}_\varepsilon)(\boldsymbol{\varepsilon}(t) - \boldsymbol{\mu}_\varepsilon)^\top\right]$ has entries determined by the relation:

$$\begin{aligned}(\Sigma_\varepsilon)_{rs} = (1-e^{-\xi})^2 \mathrm{E}\left[V_R\left(\mathbf{z}(t), \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,r\}}, x_0, \boldsymbol{\theta}\right) \cdot V_R\left(\mathbf{z}(t), \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,s\}}, x_0, \boldsymbol{\theta}\right)\right] \\ - (\boldsymbol{\mu}_{\boldsymbol{\varepsilon}})_r (\boldsymbol{\mu}_{\boldsymbol{\varepsilon}})_s, \quad r,s \in \{1,\ldots,N\},\end{aligned} \tag{6.8}$$

where the first summand is computed by first multiplying the polynomials $V_R\left(\mathbf{z}, \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,r\}}, x_0, \boldsymbol{\theta}\right)$ and $V_R\left(\mathbf{z}, \{\mathbf{C}_{j,\cdot}\}_{j\in\{1,\ldots,s\}}, x_0, \boldsymbol{\theta}\right)$ on the variable $\mathbf{z} \in \mathbb{R}^n$ and subsequently evaluating the resulting polynomial according to the following convention: any monomial of the form $az_1^{k_1}\cdots z_n^{k_n}$ is replaced by $a\mu_{k_1,\ldots,k_n}(\mathbf{z})$.

A particular case in which the higher-order moments (6.7) can be readily computed is when $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IN}(\mathbf{0}_n, \Sigma_z)$, that is, $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ follows an n-dimensional multivariate normal distribution. Indeed, following [Holm 88, Tria 03], let $k_1, \ldots, k_n \in \mathbb{N}$ be n nonzero natural numbers such that $K := k_1+k_2+\cdots+k_n$ and let $\mathbf{z}^{\mathcal{K}} := \left(z_1\mathbf{i}_{k_1}^\top, z_2\mathbf{i}_{k_2}^\top, \ldots, z_n\mathbf{i}_{k_n}^\top\right)^\top \in \mathbb{R}^K$. The vector $\mathbf{z}^{\mathcal{K}} \in \mathbb{R}^K$ is Gaussian with zero mean and covariance matrix $\Sigma_z^{\mathcal{K}} \in \mathbb{S}_K$ given by $(\Sigma_z^{\mathcal{K}})_{ij} = \mathrm{Cov}(z_i^{\mathcal{K}}, z_j^{\mathcal{K}})$, for any $i,j \in \{1,\ldots,K\}$, that is, $\mathbf{z}^{\mathcal{K}} \sim \mathrm{IN}(\mathbf{0}_K, \Sigma_z^{\mathcal{K}})$. Then, using Theorem 1 in [Tria 03], we can write that

$$\mu_{k_1,\ldots,k_n}(\mathbf{z}) = \begin{cases} Hf(\Sigma_z^{\mathcal{K}}), & \text{when } K = 2l, \quad l \in \mathbb{N}, \\ \\ 0, & \text{otherwise,} \end{cases} \tag{6.9}$$

where the symbol $Hf(\Sigma_z^{\mathcal{K}})$ denotes the hafnian of the covariance matrix $\Sigma_z^{\mathcal{K}}$ of order $2l$, $l \in \mathbb{N}$, defined by

$$Hf(\Sigma_z^{\mathcal{K}}) := \sum_{I,J} (\Sigma_z^{\mathcal{K}})_{i'_1 j'_1} (\Sigma_z^{\mathcal{K}})_{i'_2 j'_2} \cdots (\Sigma_z^{\mathcal{K}})_{i'_l j'_l}, \tag{6.10}$$

where the sum is running over all the possible decompositions of $\{1, 2, \ldots, 2l = K\}$ into disjoint subsets I, J of the form $I = \{i'_1, \ldots, i'_l\}$, $J = \{j'_1, \ldots, j'_l\}$, such that $i'_1 < \cdots < i'_l$, $j'_1 < \cdots < j'_l$, and $i'_w < j'_w$, for each $w \in \{1, \ldots, l\}$.

We now proceed as in [Grig 15] and consider (6.1) as a VAR(1) model [Lutk 05] driven by the independent noise $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{Z}}$. If we assume that the nonlinear kernel f satisfies the stability condition $|\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1$, then the proof of Theorem D.10 in [Grig 15] shows that the spectral radius $\rho(A(\mathbf{x}_0, \boldsymbol{\theta})) < 1$, which implies in turn that (6.1) has a unique causal and second order stationary solution [Lutk 05, Proposition 2.1] $\{\mathbf{x}(t)\}_{t\in\mathbb{Z}}$ with time-independent mean

$$\boldsymbol{\mu}_x = (I_N - A(\mathbf{x}_0, \boldsymbol{\theta}))^{-1}(F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) - A(\mathbf{x}_0, \boldsymbol{\theta})\mathbf{x}_0 + \boldsymbol{\mu}_\varepsilon). \tag{6.11}$$

The model (6.1) can hence be rewritten in mean-adjusted form as

$$\mathbf{x}(t) - \boldsymbol{\mu}_x = A(\mathbf{x}_0, \boldsymbol{\theta})(\mathbf{x}(t-1) - \boldsymbol{\mu}_x) + (\boldsymbol{\varepsilon}(t) - \boldsymbol{\mu}_\varepsilon). \tag{6.12}$$

Additionally, the autocovariance function $\Gamma(k) := \mathrm{E}\left[(\mathbf{x}(t) - \boldsymbol{\mu}_x)(\mathbf{x}(t-k) - \boldsymbol{\mu}_x)^\top\right]$ at lag $k \in \mathbb{Z}$ is determined by the Yule-Walker equations [Lutk 05], which have the following solutions in vectorized form:

$$\mathrm{vech}(\Gamma(0)) = (\mathbb{I}_{N'} - L_N(A(\mathbf{x}_0, \boldsymbol{\theta}) \otimes A(\mathbf{x}_0, \boldsymbol{\theta})D_N))^{-1}\mathrm{vech}(\Sigma_\varepsilon), \tag{6.13}$$

$$\Gamma(k) = A(\mathbf{x}_0, \boldsymbol{\theta})\Gamma(k-1) \text{ with } \Gamma(-k) = \Gamma(k)^\top, \tag{6.14}$$

where $N' := \frac{1}{2}N(N+1)$ and $L_N \in \mathbb{M}_{N',N^2}$, $D_N \in \mathbb{M}_{N^2,N'}$ are the elimination and the duplication matrices, respectively. We recall that, as it is stated in Proposition 3.1, the autocovariance function $\Gamma(0)$ is one of the key components of the capacity formula (2.13) that we intend to explicitly evaluate.

6.2 The reservoir model for parallel TDRs with multidimensional input signals

In this appendix we provide the details about the generalization of the reservoir model to the parallel time-delay reservoir architecture that was introduced in [Orti 12, Grig 14]. This reservoir design has shown very satisfactory robustness properties with respect to model misspecification and parameter choice. The basic idea on which this approach is built consists of presenting the input signal to a parallel array of reservoirs, each of them running with different parameter values. As it is shown in Figure 2, the concatenation of the outputs of these reservoirs is then used to construct a single readout layer via a ridge regression.

Consider a parallel array of p time-delay reservoirs. For each $j \in \{1, \ldots, p\}$, the jth time-delay reservoir is based on a time-delay differential equation like (2.2) and has an associated nonlinear kernel $f^{(j)}$ that depends on the parameters vector $\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{K_j}$ and the time-delay $\tau_j > 0$, namely

$$\dot{x}(t) = -x(t) + f^{(j)}(x(t-\tau_j), I(t), \boldsymbol{\theta}^{(j)}). \tag{6.15}$$

Let $N_j \in \mathbb{N}$ be the number of the virtual neurons of the jth reservoir and let $d_j := \tau_j/N_j$ be the corresponding separation between neurons. Let $N^* := \sum_{j=1}^p N_j$ and $K^* := \sum_{j=1}^p K_j$ be the total

number of virtual neurons and the total number of parameters of the parallel array, respectively. The discrete-time description of the parallel array of p TDRs with total number of neurons N^* is obtained by Euler time-discretizing each of the differential equations (6.15) with integration step d_j and by organizing the solutions in neural layers described by the following recursions

$$x_i^{(j)}(t) := e^{-\xi^{(j)}} x_{i-1}^{(j)}(t) + (1 - e^{-\xi^{(j)}}) f^{(j)}(x_i^{(j)}(t-1), (\mathbf{I}^{(j)}(t))_i, \boldsymbol{\theta}^{(j)}), \quad i \in \{1, \ldots, N_j\},\ j \in \{1, \ldots, p\} \tag{6.16}$$

with $\xi^{(j)} := \log(1 + d_j)$ and using the convention $x_0^{(j)}(t) := x_{N_j}^{(j)}(t-1)$. The different p input forcings $\mathbf{I}^{(j)}(t) \in \mathbb{R}^{N_j}$, $j \in \{1, \ldots, p\}$, are created out of the n-dimensional input signal $\{\mathbf{z}(t)\}_{t \in \mathbb{Z}}$ by using p input masks $\mathbf{C}^{(j)} \in \mathbb{M}_{N_j, n}$ and by setting $\mathbf{I}^{(j)}(t) := \mathbf{C}^{(j)}\mathbf{z}(t)$.

Consider now $\mathbf{X}(t) = \left(\mathbf{x}^{(1)}(t), \ldots, \mathbf{x}^{(p)}(t)\right) \in \mathbb{R}^{N^*}$, where $\mathbf{x}^{(j)}(t) \in \mathbb{R}^{N_j}$, $j \in \{1, \ldots, p\}$, is the neuron layer at time t corresponding to the jth individual TDR. As in the case of the individually operating time-delay reservoir in Section 6.1, the recursions (6.16) uniquely determine reservoir maps $F^{(j)} : \mathbb{R}^{N_j} \times \mathbb{R}^{N_j} \times \mathbb{R}^{K_j} \to \mathbb{R}^{N_j}$, constructed out of the associated nonlinear kernels $f^{(j)}$, $j \in \{1, \ldots, p\}$ that can be put together to determine the map

$$\begin{pmatrix} \mathbf{x}^{(1)}(t) \\ \mathbf{x}^{(2)}(t) \\ \vdots \\ \mathbf{x}^{(p)}(t) \end{pmatrix} = \begin{pmatrix} F^{(1)}(\mathbf{x}^{(1)}(t-1), \mathbf{I}^{(1)}(t), \boldsymbol{\theta}^{(1)}) \\ F^{(2)}(\mathbf{x}^{(2)}(t-1), \mathbf{I}^{(2)}(t), \boldsymbol{\theta}^{(2)}) \\ \vdots \\ F^{(p)}(\mathbf{x}^{(p)}(t-1), \mathbf{I}^{(p)}(t), \boldsymbol{\theta}^{(p)}) \end{pmatrix}. \tag{6.17}$$

that can be rewritten as

$$\mathbf{X}(t) = F(\mathbf{X}(t-1), \mathbf{I}(t), \boldsymbol{\Theta}), \tag{6.18}$$

where $F : \mathbb{R}^{N^*} \times \mathbb{R}^{N^*} \times \mathbb{R}^{K^*} \to \mathbb{R}^{N^*}$ is referred to as the **parallel reservoir map**, $\mathbf{I}(t) := \left(\mathbf{I}^{(1)}(t), \ldots, \mathbf{I}^{(p)}(t)\right)$, and $\boldsymbol{\Theta} := (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(p)}) \in \mathbb{R}^{K^*}$ is the vector containing all the parameters of the parallel array of TDRs. Parallel TDRs based on the recursion (6.18) are referred to in the sequel as **discrete-time parallel TDRs**.

We now generalize to the parallel context the reservoir model that we described in detail in Section 6.1. We start by choosing p stable equibria $x_0^{(j)}$ of the dynamical systems (6.15) or, equivalently, p fixed points of the form $\mathbf{x}_0^{(j)} := x_0^{(j)} \mathbf{i}_{N_j}$ of each of the reservoir maps in (6.17). These fixed points determine a fixed point $\mathbf{X}_0 := (\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(p)}) \in \mathbb{R}^{N^*}$ of the parallel array in (6.18). Now, as in Section 6.1 we partially linearize (6.18) at $\mathbf{X}_0$ and use a higher Rth-order Taylor series expansion on the forcing $\mathbf{I}(t) \in \mathbb{R}^{N^*}$. Analogously to the single reservoir case, we obtain that

$$\mathbf{X}(t) = F(\mathbf{X}_0, \mathbf{0}_{N^*}, \boldsymbol{\Theta}) + A(\mathbf{X}_0, \boldsymbol{\Theta})(\mathbf{X}(t-1) - \mathbf{X}_0) + \boldsymbol{\varepsilon}^{(\mathbf{X}_0, \boldsymbol{\Theta})}(t), \tag{6.19}$$

where $A(\mathbf{X}_0, \boldsymbol{\Theta}) := D_{\mathbf{X}} F(\mathbf{X}_0, \mathbf{0}_{N^*}, \boldsymbol{\Theta})$ is the **parallel reservoir connectivity matrix**, which is the first derivative of F with respect to its first argument and evaluated at the point $(\mathbf{X}_0, \boldsymbol{\Theta})$, and

$$\boldsymbol{\varepsilon}(t)^{(\mathbf{X}_0, \boldsymbol{\Theta})} := \begin{pmatrix} \boldsymbol{\varepsilon}(t)^{(\mathbf{x}_0^{(1)}, \boldsymbol{\theta}^{(1)})} \\ \boldsymbol{\varepsilon}(t)^{(\mathbf{x}_0^{(2)}, \boldsymbol{\theta}^{(2)})} \\ \vdots \\ \boldsymbol{\varepsilon}(t)^{(\mathbf{x}_0^{(p)}, \boldsymbol{\theta}^{(p)})} \end{pmatrix} \in \mathbb{R}^{N^*} \tag{6.20}$$

with

$$\boldsymbol{\varepsilon}(t)^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})} := (1-e^{-\xi^{(j)}}) \begin{pmatrix} V_R^{(j)}\left(\mathbf{z}(t), \left\{\mathbf{C}_{1,\cdot}^{(j)}\right\}, x_0^{(j)}, \boldsymbol{\theta}^{(j)}\right) \\ V_R^{(j)}\left(\mathbf{z}(t), \left\{\mathbf{C}_{i,\cdot}^{(j)}\right\}_{i\in\{1,2\}}, x_0^{(j)}, \boldsymbol{\theta}^{(j)}\right) \\ \vdots \\ V_R^{(j)}\left(\mathbf{z}(t), \left\{\mathbf{C}_{i,\cdot}^{(j)}\right\}_{i\in\{1,\ldots,N_j\}}, x_0^{(j)}, \boldsymbol{\theta}^{(j)}\right) \end{pmatrix} \in \mathbb{R}^{N_j}, \;\; j\in\{1,\ldots,p\}, \tag{6.21}$$

where the polynomials $V_R^{(j)}$ are defined as in (6.5). The assumption that the input signal $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ is a family of n-dimensional independent and identically distributed random variables implies that the same property holds for the family $\left\{\boldsymbol{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})}(t)\right\}_{t\in\mathbb{Z}}$ of N^*-dimensional random variables in (6.19), namely, that $\left\{\boldsymbol{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})}(t)\right\}_{t\in\mathbb{Z}} \sim \text{IID}(\boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})}, \Sigma_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})})$. The mean $\boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})}$ can be written as

$$\boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})} = \begin{pmatrix} \boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)})} \\ \boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(2)},\boldsymbol{\theta}^{(2)})} \\ \vdots \\ \boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)})} \end{pmatrix}, \tag{6.22}$$

where $\boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})} := E\left[\boldsymbol{\varepsilon}(t)^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})}\right] \in \mathbb{R}^{N_j}$, $j\in\{1,\ldots,p\}$, whose components are determined by an expression of the form (6.6), that is,

$$\begin{aligned} \left(\boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})}\right)_r =& (1-e^{-\xi^{(j)}}) \sum_{i=1}^{R} (\partial_I^{(i)} f^{(j)})(x_0^{(j)}, 0, \boldsymbol{\theta}^{(j)}) \sum_{j'=1}^{r} e^{-(r-j')\xi^{(j)}} \\ & \times \sum_{k_1+\cdots+k_n=i} \frac{1}{k_1!\ldots k_n!} \prod_{s=1}^{n} (\mathbf{C}_{j's}^{(j)})^{k_s} \cdot \mu_{k_1,\ldots,k_n}(\mathbf{z}), \;\; r\in\{1,\ldots,N_j\}. \end{aligned} \tag{6.23}$$

Additionally, the covariance matrix $\Sigma_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})} := \mathrm{E}\left[(\boldsymbol{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})}(t) - \boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})})(\boldsymbol{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})}(t) - \boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})})^\top\right]$ can be written as

$$\Sigma_\varepsilon^{(\mathbf{X}_0,\boldsymbol{\Theta})} = \begin{pmatrix} \Sigma_\varepsilon^{(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)}),(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)})} & \cdots & \Sigma_\varepsilon^{(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)}),(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)})} \\ \vdots & \ddots & \vdots \\ \Sigma_\varepsilon^{(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)}),(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)})} & \cdots & \Sigma_\varepsilon^{(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)}),(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)})} \end{pmatrix}, \tag{6.24}$$

where each block $\Sigma_\varepsilon^{(\mathbf{x}_0^{(i)},\boldsymbol{\theta}^{(i)}),(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})} \in \mathbb{M}_{N_i,N_j}$, $i,j\in\{1,\ldots,p\}$ represents the covariance between the innovation components that drive the ith and the jth time-delay reservoirs, respectively, and which has entries determined by:

$$\begin{aligned} (\Sigma_\varepsilon^{(\mathbf{x}_0^{(i)},\boldsymbol{\theta}^{(i)}),(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})})_{rs} =& (1-e^{-\xi^{(i)}})(1-e^{-\xi^{(j)}}) \mathrm{E}\Big[V_R^{(i)}\left(\mathbf{z}(t), \left\{\mathbf{C}_{j',\cdot}^{(i)}\right\}_{j'\in\{1,\ldots,r\}}, x_0^{(i)}, \boldsymbol{\theta}^{(i)}\right) \\ & \times V_R^{(i)}\left(\mathbf{z}(t), \left\{\mathbf{C}_{j',\cdot}^{(j)}\right\}_{j'\in\{1,\ldots,s\}}, x_0^{(j)}, \boldsymbol{\theta}^{(j)}\right)\Big] \\ & - (\boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(i)},\boldsymbol{\theta}^{(i)})})_r (\boldsymbol{\mu}_\varepsilon^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})})_s, \;\; r\in\{1,\ldots,N_i\}, \;\; s\in\{1,\ldots,N_j\}, \end{aligned} \tag{6.25}$$

where the first summand is computed using the same approach that we described in expression (6.8).

The connectivity matrix can be easily written in terms of the connectivity matrices of each of the reservoirs that make up the parallel pool as

$$A(\mathbf{X}_0,\mathbf{\Theta}) := D_{\mathbf{X}}F(\mathbf{X}_0,\mathbf{0}_{N^*}\mathbf{\Theta}) = \begin{pmatrix} A^{(1)}(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)}) & \mathbb{O}_{N_1,N_2} & \cdots & \mathbb{O}_{N_1,N_p} \\ \mathbb{O}_{N_2,N_1} & A^{(2)}(\mathbf{x}_0^{(2)},\boldsymbol{\theta}^{(2)}) & \cdots & \mathbb{O}_{N_2,N_p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{O}_{N_p,N_1} & \mathbb{O}_{N_p,N_2} & \cdots & A^{(p)}(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)}) \end{pmatrix}, \tag{6.26}$$

where for each $j \in \{1,\ldots,p\}$ the matrix $A^{(j)}(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)}) := D_{\mathbf{x}}F^{(j)}(\mathbf{x}_0^{(j)},\mathbf{0}_{N_j},\boldsymbol{\theta}^{(j)})$ is the connectivity matrix of the jth TDR, determined as the first derivative of the corresponding jth reservoir map $F^{(j)}$ with respect to its first argument, evaluated at the point $(\mathbf{x}_0^{(j)},\mathbf{0}_{N_j},\boldsymbol{\theta}^{(j)})$. Each of those individual connectivity matrices $A^{(j)}(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})$ has the explicit form provided in (6.3). Moreover, if for each of the individual equilibria $x_0^{(j)}$ used in the construction we require the stability condition $|\partial_x f^{(j)}(x_0^{(j)},0,\boldsymbol{\theta})| < 1$, then by the proof of Theorem D.10 in [Grig 15] we have that the spectral radii $\rho(A^{(j)}(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})) < 1$ and, consequently

$$\rho(A(\mathbf{X}_0,\mathbf{\Theta})) < 1. \tag{6.27}$$

In these conditions, (6.19) determines a VAR(1) model driven by the noise $\left\{\boldsymbol{\varepsilon}^{(\mathbf{X}_0,\mathbf{\Theta})}(t)\right\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\mathbf{\Theta})},\Sigma_{\varepsilon}^{(\mathbf{X}_0,\mathbf{\Theta})})$ and that has a unique causal and second order stationary solution $\{\mathbf{X}(t)\}_{t\in\mathbb{Z}}$ with time-independent mean

$$\boldsymbol{\mu}_X^{(\mathbf{X}_0,\mathbf{\Theta})} = \begin{pmatrix} \boldsymbol{\mu}_x^{(\mathbf{x}_0^{(1)},\boldsymbol{\theta}^{(1)})} \\ \boldsymbol{\mu}_x^{(\mathbf{x}_0^{(2)},\boldsymbol{\theta}^{(2)})} \\ \vdots \\ \boldsymbol{\mu}_x^{(\mathbf{x}_0^{(p)},\boldsymbol{\theta}^{(p)})} \end{pmatrix}, \tag{6.28}$$

with

$$\boldsymbol{\mu}_x^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})} = (\mathbb{I}_{N_j} - A^{(j)}(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)}))^{-1}(F^{(j)}(\mathbf{x}_0^{(j)},\mathbf{0}_{N_j},\boldsymbol{\theta}^{(j)}) - A^{(j)}(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})\mathbf{x}_0^{(j)} + \boldsymbol{\mu}_{\varepsilon}^{(\mathbf{x}_0^{(j)},\boldsymbol{\theta}^{(j)})}). \tag{6.29}$$

This allows us to write the parallel reservoir model (6.19) in mean-adjusted form as

$$\mathbf{X}(t) - \boldsymbol{\mu}_X^{(\mathbf{X}_0,\mathbf{\Theta})} = A(\mathbf{X}_0,\mathbf{\Theta})(\mathbf{X}(t-1) - \boldsymbol{\mu}_X^{(\mathbf{X}_0,\mathbf{\Theta})}) + (\boldsymbol{\varepsilon}(t)^{(\mathbf{X}_0,\mathbf{\Theta})} - \boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\mathbf{\Theta})}). \tag{6.30}$$

Finally, the autocovariance function $\Gamma(k) := \mathrm{E}[(\mathbf{X}(t) - \boldsymbol{\mu}_X^{(\mathbf{X}_0,\mathbf{\Theta})})(\mathbf{X}(t-k) - \boldsymbol{\mu}_X^{(\mathbf{X}_0,\mathbf{\Theta})})^{\top}]$ of $\{\mathbf{X}(t)\}_{t\in\mathbb{Z}}$ at lag $k \in \mathbb{Z}$ is determined by the Yule-Walker equations [Lutk 05] whose solutions in vectorized form are:

$$\mathrm{vech}(\Gamma(0)) = (\mathbb{I}_{N^{*\prime}} - L_{N^*}(A(\mathbf{X}_0,\mathbf{\Theta}) \otimes A(\mathbf{X}_0,\mathbf{\Theta}))D_{N^*})^{-1}\,\mathrm{vech}(\Sigma_{\varepsilon}^{(\mathbf{X}_0,\mathbf{\Theta})}), \tag{6.31}$$

$$\Gamma(k) = A(\mathbf{X}_0,\mathbf{\Theta})\Gamma(k-1) \;\text{ with }\; \Gamma(-k) = \Gamma(k)^{\top}, \tag{6.32}$$

where $N^{*\prime} := \dfrac{1}{2}N^*(N^*+1)$ and $L_{N^*} \in \mathbb{M}_{N^{*\prime},N^{*2}}$, $D_{N^*} \in \mathbb{M}_{N^{*2},N^{*\prime}}$ are the elimination and the duplication matrices, respectively. We recall that, as it is stated in Proposition 3.2, the autocovariance function $\Gamma(0)$ is one of the key components of the capacity formula (2.13) that we intend to explicitly evaluate.

6.3 The reservoir model and the computation of teaching covariances for linear and quadratic memory tasks

As we already pointed out in Section 3.1, the computation of the reservoir memory capacity $C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda)$ (see (3.1)) requires the evaluation of the following three ingredients: $\Gamma(0) := \mathrm{Cov}(\mathbf{x}(t), \mathbf{x}(t))$, $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$, and $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$. In the preceding two appendices we showed how the reservoir models (6.12) and (6.30) provide approximated expressions of $\Gamma(0)$ via the Yule-Walker equations associated to the corresponding VAR(1) model.

In what follows we show how to use those models, as well as their dynamical features in order to explicitly compute the covariances $\mathrm{Cov}(\mathbf{x}(t), \mathbf{y}(t))$ and $\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))$ associated to two different memory tasks. As we already explained in (2.8), a memory task is determined by a map

$$\begin{array}{rccc} H: & \mathbb{R}^{(h+1)n} & \longrightarrow & \mathbb{R}^q \\ & \mathrm{vec}(\mathbf{z}(t), \ldots, \mathbf{z}(t-h)) & \mapsto & \mathbf{y}(t), \end{array} \tag{6.33}$$

with $q, h \in \mathbb{N}$, that is made out of q different real valued functions of the input signal, h time steps into the past. The reservoir memory capacity $C_H(\boldsymbol{\theta}, \mathbf{C}, \lambda)$ associated to H measures the ability of the reservoir with parameters $\boldsymbol{\theta}$ to recover that function after being trained using a teaching signal.

In the next paragraphs we place ourselves in the context of a parallel array of p time-delay reservoirs as in Figure 2, with collective nonlinear kernel parameters $\boldsymbol{\Theta} \in \mathbb{R}^{K^*}$ and operating in the neighbourhood of a stable fixed point $\mathbf{X}_0 \in \mathbb{R}^{N^*}$, with N^* and K^* the total number of neurons and the total number of parameters of the array, respectively. The input signal $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ is assumed to be a family of n-dimensional independent and identically distributed random variables with mean zero and covariance matrix $\Sigma_z \in \mathbb{S}_n^+$, that is $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\mathbf{0}_n, \Sigma_z)$. In these conditions we will provide explicit expressions for $\mathrm{Cov}(\mathbf{X}(t), \mathbf{y}(t))$ and $\mathrm{trace}\,(\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t)))$ for the two particular cases in which H are linear and quadratic functions..

Linear memory task. Consider the q-dimensional h-lag linear memory task function $H : \mathbb{R}^{(h+1)n} \longrightarrow \mathbb{R}^q$ determined by the assignment $H(\mathbf{z}^h(t)) := L^\top \mathbf{z}^h(t) =: \mathbf{y}(t)$, where $\mathbf{z}^h(t) = \mathrm{vec}(\mathbf{z}(t), \mathbf{z}(t-1), \ldots, \mathbf{z}(t-h)) \in \mathbb{R}^{(h+1)n}$ and $L \in \mathbb{M}_{(h+1)n,q}$. We now compute in this case $\mathrm{Cov}(\mathbf{X}(t), \mathbf{y}(t))$ and $\mathrm{trace}\,(\mathrm{Cov}(\mathbf{y}(t), \mathbf{y}(t))) = \mathrm{Cov}(\mathbf{y}(t)^\top, \mathbf{y}(t)^\top)$, that are required for the memory capacity evaluation.
(i) We start with $\mathrm{Cov}(\mathbf{y}(t)^\top, \mathbf{y}(t)^\top)$ and write

$$\begin{aligned} \mathrm{Cov}(\mathbf{y}(t)^\top, \mathbf{y}(t)^\top) &= \mathrm{E}\left[\mathbf{y}(t)^\top \mathbf{y}(t)\right] - \mathrm{E}\left[\mathbf{y}(t)^\top\right] \mathrm{E}\left[\mathbf{y}(t)\right] \\ &= \mathrm{E}\left[\mathbf{z}^h(t)^\top L L^\top \mathbf{z}^h(t)\right] = \mathrm{trace}\left(L L^\top \mathrm{E}[\mathbf{z}^h(t)\mathbf{z}^h(t)^\top]\right) = \mathrm{trace}(L L^\top \Sigma_{z^h}), \end{aligned} \tag{6.34}$$

where the covariance matrix $\Sigma_{z^h} \in \mathbb{S}_{(h+1)n\times(h+1)n}$ has the form

$$\Sigma_{z^h} = \begin{pmatrix} \Sigma_z & \cdots & \mathbb{O}_n \\ \vdots & \ddots & \vdots \\ \mathbb{O}_n & \cdots & \Sigma_z \end{pmatrix}.$$

(ii) We now compute $\mathrm{Cov}(\mathbf{X}(t), \mathbf{y}(t))$. As we already pointed out, the stability condition on the fixed point $\mathbf{X}_0$ implies that the unique stationary solution of the VAR(1) model (6.30) admits a MA(∞) representation of the form:

$$\mathbf{X}(t) - \boldsymbol{\mu}_X^{(\mathbf{X}_0, \boldsymbol{\Theta})} = \sum_{j=0}^{\infty} \Psi_j \boldsymbol{\rho}(t-j), \tag{6.35}$$

with $\Psi_j \in \mathbb{M}_{N^*}$ and $\boldsymbol{\rho}(t) := \boldsymbol{\varepsilon}(t)^{(\mathbf{X}_0, \boldsymbol{\Theta})} - \boldsymbol{\mu}_\varepsilon^{(\mathbf{X}_0, \boldsymbol{\Theta})}$. Then, for any $i \in \{1, \ldots, N^*\}$ and $j \in \{1, \ldots, q\}$ we

write

$$\begin{aligned}\mathrm{Cov}(X_i(t),y_j(t)) &= \sum_{k=0}^{\infty}\mathrm{Cov}((\Psi_k\boldsymbol{\rho}(t-k))_i,(L^\top\cdot\mathbf{z}^h(t))_j) = \sum_{k=0}^{\infty}\sum_{u=1}^{N^*}\sum_{v=1}^{(h+1)n}(\Psi_k)_{iu}L_{vj}\mathrm{E}[\rho_u(t-k)(\mathbf{z}^h(t))_v]\\ &= \sum_{k=0}^{\infty}\sum_{u=1}^{N^*}\sum_{v=1}^{n}\sum_{s=1}^{h+1}(\Psi_k)_{iu}L_{v\cdot s,j}\mathrm{E}[(\varepsilon_u(t-k)^{(\mathbf{X}_0,\boldsymbol{\Theta})}-(\boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})})_u)z_v(t-s+1)]\\ &= \sum_{u=1}^{N^*}\sum_{v=1}^{n}\sum_{s=0}^{h}(\Psi_s)_{iu}L_{v\cdot s,j}\mathrm{E}[\varepsilon_u(t-s)^{(\mathbf{X}_0,\boldsymbol{\Theta})}z_v(t-s)],\end{aligned}\tag{6.36}$$

where the vector $\boldsymbol{\varepsilon}(t)^{(\mathbf{X}_0,\boldsymbol{\Theta})}$ is provided in (6.20)-(6.21) and the expectations $\mathrm{E}[\varepsilon_u(t-s)^{(\mathbf{X}_0,\boldsymbol{\Theta})}z_v(t-s)]$ are computed by multiplying the V_R polynomial corresponding to $\varepsilon_u(t)^{(\mathbf{X}_0,\boldsymbol{\Theta})}$ by the monomial $z_v(t)$; the resulting polynomial is the evaluated on the higher order moments of $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ using the same rule that we stated after (6.8).

Quadratic memory task. Consider now the q-dimensional h-lag quadratic memory task function $H:\mathbb{R}^{(h+1)n}\longrightarrow\mathbb{R}^q$ determined by the assignment $H(\mathbf{z}^h(t)):=Q\cdot\mathrm{vech}(\mathbf{z}^h(t)\cdot\mathbf{z}^h(t)^\top)=:\mathbf{y}(t)$, where $\mathbf{z}^h(t)=\mathrm{vec}(\mathbf{z}(t),\mathbf{z}(t-1),\ldots,\mathbf{z}(t-h))\in\mathbb{R}^{(h+1)n}$, $Q\in\mathbb{M}_{q,q^*}$ and $q^*:=\dfrac{1}{2}(h+1)n((h+1)n+1)$. We now provide explicit expressions for $\mathrm{Cov}(\mathbf{X}(t),\mathbf{y}(t))$ and $\mathrm{Cov}(\mathbf{y}(t)^\top,\mathbf{y}(t)^\top)$ in this case, that are required in order to evaluate the corresponding memory capacity.

(i) We start with $\mathrm{Cov}(\mathbf{y}(t)^\top,\mathbf{y}(t)^\top)$. Let $M^h:=\mathbf{z}^h(t)\mathbf{z}^h(t)^\top$ and write

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}(t)^\top,\mathbf{y}(t)^\top) &= \mathrm{E}\left[\mathbf{y}(t)^\top\mathbf{y}(t)\right]-\mathrm{E}\left[\mathbf{y}(t)^\top\right]\mathrm{E}\left[\mathbf{y}(t)\right]\\ &= \mathrm{E}[(\mathrm{vech}(\mathbf{z}^h(t)\mathbf{z}^h(t)^\top))^\top Q^\top Q(\mathrm{vech}(\mathbf{z}^h(t)\mathbf{z}^h(t)^\top))]-\mathrm{E}\left[\mathbf{y}(t)^\top\right]\mathrm{E}\left[\mathbf{y}(t)\right]\\ &= \mathrm{trace}(Q^\top Q\cdot\mathrm{E}[\mathrm{vech}(M^h)(\mathrm{vech}(M^h))^\top])-\mathrm{E}\left[\mathbf{y}(t)^\top\right]\mathrm{E}\left[\mathbf{y}(t)\right].\end{aligned}\tag{6.37}$$

Notice now that for any $i,j\in\{1,\ldots,(h+1)n\}$ there exist $l_i,l_j\in\{0,\ldots,h\}$ and $m_i,m_j\in\{0,\ldots,n\}$ such that $i=l_ip+m_i$, $j=l_jp+m_j$ and hence

$$M^h_{ij}=(\mathbf{z}^h(t)\mathbf{z}^h(t)^\top)_{ij}=z_{m_i}(t-l_i)z_{m_j}(t-l_j).\tag{6.38}$$

Consequently, for any $i,j\in\{1,\ldots,(h+1)n\}$

$$\begin{aligned}\mathrm{E}[\mathrm{vech}(M^h)\mathrm{vech}(M^h)^\top_{ij}] &= \mathrm{E}[\mathrm{vech}(M^h)_i\mathrm{vech}(M^h)_j]=\mathrm{E}[M^h_{\sigma^{-1}(i)}M^h_{\sigma^{-1}(j)}]\\ &= \mathrm{E}[z_{m_{r(i)}}(t-l_{r(i)})z_{m_{s(i)}}(t-l_{s(i)})z_{m_{u(j)}}(t-l_{u(j)})z_{m_{v(j)}}(t-l_{v(j)})],\end{aligned}\tag{6.39}$$

where the operator σ^{-1} assigns to the index of the position of an element in $\mathrm{vech}(M^h)$ the two indices corresponding to its position in the matrix $M^h\in\mathbb{S}_{q^*}$. In this expression $(r(i),s(i)):=\sigma^{-1}(i)$, $(u(j),v(j)):=\sigma^{-1}(j)$ and $r(i)=l_{r(i)}p+m_{r(i)}$, $s(i)=l_{s(i)}p+m_{s(i)}$, $u(j)=l_{u(j)}p+m_{u(j)}$, $v(j)=l_{v(j)}p+m_{v(j)}$ with $l_{r(i)},l_{s(i)},l_{u(j)},l_{v(j)}\in\{0,\ldots,h\}$ and $m_{r(i)},m_{s(i)},m_{u(j)},m_{v(j)}\in\{1,\ldots,p\}$. Additionally, using this notation the following relation holds true

$$\mathrm{E}\left[y_k(t)\right]=\sum_{i=1}^{q^*}Q_{kj}\mathrm{E}[(\mathrm{vech}(M^h))_i]=\sum_{j=1}^{q^*}Q_{kj}\mathrm{E}[z_{m_{r(i)}}(t-l_{r(i)})z_{m_{s(i)}}(t-l_{s(i)})].\tag{6.40}$$

We hence now can derive the expression for (6.37) as

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}(t)^\top,\mathbf{y}(t)^\top) = \sum_{i=1}^{q^*}\sum_{k=1}^{q}Q_{ki}\Big(&\sum_{j=1}^{q^*}Q_{kj}\cdot\mathrm{E}[z_{m_{r(i)}}(t-l_{r(i)})z_{m_{s(i)}}(t-l_{s(i)})z_{m_{u(j)}}(t-l_{u(j)})z_{m_{v(j)}}(t-l_{v(j)})]\\ &-Q_{ki}\mathrm{E}[z_{m_{r(i)}}(t-l_{r(i)})z_{m_{s(i)}}(t-l_{s(i)})]^2\Big),\end{aligned}\tag{6.41}$$

where we used the same notation as in (6.39) and (6.40).
(ii) $\mathrm{Cov}(\mathbf{X}(t),\mathbf{y}(t))$: for any $i\in\{1,\ldots,N^*\}$, $j\in\{1,\ldots,q\}$ we have

$$\begin{aligned}
\mathrm{Cov}(X_i(t),y_j(t)) &= \sum_{k=0}^{\infty}\mathrm{Cov}((\Psi_k\boldsymbol{\rho}(t-k))_i,(Q\cdot\mathrm{vech}(M^h))_j)\\
&=\sum_{k=0}^{\infty}\sum_{u=1}^{N^*}\sum_{v=1}^{q^*}(\Psi_k)_{iu}Q_{jv}\mathrm{E}[\rho_u(t-k)(\mathrm{vech}(M^h))_v]\\
&=\sum_{k=0}^{\infty}\sum_{u=1}^{N^*}\sum_{v=1}^{q^*}(\Psi_k)_{iu}Q_{jv}\mathrm{E}[(\varepsilon_u(t-k)^{(\mathbf{X}_0,\boldsymbol{\Theta})}-(\boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})})_u)(\mathrm{vech}(M^h))_v]\\
&=\sum_{k=0}^{\infty}\sum_{u=1}^{N^*}\sum_{v=1}^{q^*}(\Psi_k)_{iu}Q_{jv}\{\mathrm{E}[(\varepsilon_u(t-k)^{(\mathbf{X}_0,\boldsymbol{\Theta})}-(\boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})})_u)z_{m_{r(v)}}(t-l_{m_{r(v)}})z_{m_{s(v)}}(t-l_{m_{s(v)}})]\}\\
&=\sum_{k=0}^{h}\sum_{u=1}^{N^*}\sum_{v=1}^{q^*}(\Psi_k)_{iu}Q_{jv}\{\mathrm{E}[\varepsilon_u(t-k)^{(\mathbf{X}_0,\boldsymbol{\Theta})}\cdot z_{m_{r(v)}}(t-l_{m_{r(v)}})z_{m_{s(v)}}(t-l_{m_{s(v)}})]\\
&\quad-(\boldsymbol{\mu}_{\varepsilon}^{(\mathbf{X}_0,\boldsymbol{\Theta})})_u\mathrm{E}[z_{m_{r(v)}}(t-l_{m_{r(v)}})z_{m_{s(v)}}(t-l_{m_{s(v)}})]\},
\end{aligned}\tag{6.42}$$

where the vector $\boldsymbol{\varepsilon}(t)^{(\mathbf{X}_0,\boldsymbol{\Theta})}$ is provided in (6.20)-(6.21) and where we used the same notation as in (6.39) and (6.40).

6.4 Training errors with estimated parameters

In this appendix we prove the Proposition 3.3 that quantifies the training reservoir error committed when using readout layers that have been estimated using finite samples. Since that result is a consequence of the properties of the ridge estimator, we will start by stating and proving a few facts about it in the matrix context that are needed later on.

Consider the regression model

$$\mathbf{y}(t)=\boldsymbol{a}+W^{\top}\mathbf{x}(t)+\boldsymbol{\varepsilon}(t),\quad\{\boldsymbol{\varepsilon}(t)\}\sim\mathrm{IN}(\mathbf{0}_q,\Sigma_{\varepsilon}^q),\quad t\in\mathbb{N},\tag{6.43}$$

where $\mathbf{x}(t)\in\mathbb{R}^N$, $\mathbf{y}(t)\in\mathbb{R}^q$, and $\boldsymbol{\varepsilon}(t)\in\mathbb{R}^q$, $t\in\mathbb{N}$, are random vectors defined on a given probability space $(\Omega,\mathcal{F},\mathbb{P})$. The element $\boldsymbol{a}\in\mathbb{R}^q$ is a constant vector intercept and $W\in\mathbb{M}_{N,q}$ is the regression matrix. As we indicate in (6.43), we assume that $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{N}}$ is a family of q-dimensional independent and normally distributed random vectors with mean $\mathbf{0}_q$ and covariance matrix $\Sigma_{\varepsilon}^q\in\mathbb{S}_q$. Additionally, all along this section we will work under the following important assumptions:

(A1) $\mathbf{x}(t)\in\mathbb{R}^N$ is a random vector independent of $\boldsymbol{\varepsilon}(s)\in\mathbb{R}^q$ for any $s\leq t$, $s,t\in\mathbb{N}$.

(A2) The joint process $\{(\mathbf{x}(t),\mathbf{y}(t))\}_{t\in\mathbb{N}}$ is second-order stationary. This assumption implies, in particular, that the second-order moments $\Gamma(0):=\mathrm{Cov}(\mathbf{x}(t),\mathbf{x}(t))$, $\mathrm{Cov}(\mathbf{x}(t),\mathbf{y}(t))$, and $\mathrm{Cov}(\mathbf{y}(t),\mathbf{y}(t))$ exist and are time-independent.

Given that any regression model (6.43) with intercept can be rewritten as

$$\mathbf{y}(t)=\overline{W}^{\top}\overline{\mathbf{x}}(t)+\boldsymbol{\varepsilon}(t),\quad\{\boldsymbol{\varepsilon}(t)\}\sim\mathrm{IN}(\mathbf{0}_q,\Sigma_{\varepsilon}^q),\quad t\in\mathbb{N},\tag{6.44}$$

with $\overline{W}:=(\boldsymbol{a}\mid W^{\top})^{\top}$ and $\overline{\mathbf{x}}(t):=(1\mid\mathbf{x}(t)^{\top})^{\top}$, we can focus in what follows, without loss of generality, exclusively on regression models without intercept:

$$\mathbf{y}(t)=W^{\top}\mathbf{x}(t)+\boldsymbol{\varepsilon}(t),\quad\{\boldsymbol{\varepsilon}(t)\}\sim\mathrm{IN}(\mathbf{0}_q,\Sigma_{\varepsilon}^q),\quad t\in\mathbb{N}.\tag{6.45}$$

Definition 6.1 *Consider a regression model as in (6.45). The* **ridge regularized version** W_λ *of* W *with* **regularization strength** $\lambda \in \mathbb{R}^+$ *is the solution of the optimization problem*

$$W_\lambda = \underset{W\in\mathbb{M}_{N,q}}{\arg\min} \left(\text{trace}\left(\text{E}\left[(W^\top\cdot\mathbf{x}(t)-\mathbf{y}(t))^\top(W^\top\cdot\mathbf{x}(t)-\mathbf{y}(t))\right]\right)+\lambda\|W\|^2_{\text{Frob}}\right). \tag{6.46}$$

When assumptions **(A1)** *and* **(A2)** *hold, the solution of (6.46) is given by the constant matrix:*

$$W_\lambda = \left(\text{Cov}(\mathbf{x}(t),\mathbf{x}(t))+\lambda\mathbb{I}_N\right)^{-1}\text{Cov}(\mathbf{x}(t),\mathbf{y}(t)). \tag{6.47}$$

Note that the ridge regularized version of W with zero regularization strength coincides with W, that is, $W_0 = W$. Moreover, it is easy to verify that

$$W_\lambda = \left(\text{Cov}(\mathbf{x}(t),\mathbf{x}(t))+\lambda\mathbb{I}_N\right)^{-1}\text{Cov}(\mathbf{x}(t),\mathbf{x}(t))W, \tag{6.48}$$

or, equivalently,

$$W_\lambda - W = -\lambda\left(\text{Cov}(\mathbf{x}(t),\mathbf{x}(t))+\lambda\mathbb{I}_N\right)^{-1}W. \tag{6.49}$$

6.4.1 Properties of the ridge estimator

We now study the distribution properties of the empirical ridge estimator $\widehat{W}_\lambda$ of W_λ introduced in (3.7) for the regression model (6.45) under the assumptions **(A1)** and **(A2)**. To that end, we will consider a *fixed* finite sample realization $X \in \mathbb{M}_{N,T}$ of $\{\mathbf{x}(t)\}_{t\in\mathbb{N}}$ of length T (we use the same notation as in Section 3.2) and we will determine the distribution of the different ridge estimates $\widehat{W}_\lambda$ of W_λ obtained when *arbitrary* realizations $E \in \mathbb{M}_{q,T}$ and $Y \in \mathbb{M}_{q,T}$ of length T of $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{N}}$ and $\{\mathbf{y}(t)\}_{t\in\mathbb{N}}$ are considered.

We start by noticing that if $\{\boldsymbol{\varepsilon}(t)\} \sim \text{IN}(\mathbf{0}_q, \Sigma^q_\varepsilon)$ then $E \sim \text{MN}(\mathbb{O}_{q,T}, \Sigma^q_\varepsilon, \mathbb{I}_T)$, where the symbol MN stands for the matrix normal distribution (see [Gupt 00] for definitions and properties). We will hence be considering realizations of (6.45) of the form

$$Y = W^\top X + E, \quad X \in \mathbb{M}_{N,T}, \quad E \sim \text{MN}(\mathbb{O}_{q,T}, \Sigma^q_\varepsilon, \mathbb{I}_T). \tag{6.50}$$

Proposition 6.2 *Consider the regression model*

$$\mathbf{y}(t) = W^\top\mathbf{x}(t)+\boldsymbol{\varepsilon}(t), \;\; \{\boldsymbol{\varepsilon}(t)\}\sim\text{IN}(\mathbf{0}_q,\Sigma^q_\varepsilon), \;\; t\in\mathbb{N}, \tag{6.51}$$

that satisfies the assumptions **(A1)** *and* **(A2)**. *Let* $X \in \mathbb{M}_{N,T}$ *be a fixed finite sample realization of* $\{\mathbf{x}(t)\}_{t\in\mathbb{N}}$ *of length* T. *Then, the ridge estimator* $\widehat{W}_\lambda := (XAX^\top+\lambda T\mathbb{I}_N)^{-1}XAY^\top$ *conditional on* X *is matrix normally distributed as*

$$\left(\widehat{W}_\lambda - W\right)\Big|_X \sim \text{MN}\left(-\lambda T R_\lambda W, R_\lambda XAX^\top R_\lambda, \Sigma^q_\varepsilon\right), \tag{6.52}$$

where $A := \mathbb{I}_T - \dfrac{1}{T}\mathbf{i}_T\mathbf{i}_T^\top$ *and* $R_\lambda := (XAX^\top+\lambda T\mathbb{I}_N)^{-1}$.

Proof. We start by recalling the following result that appears as Theorem 2.3.10 in [Gupt 00].

Lemma 6.3 *Let* $Z \sim \text{MN}_{n,m}(M_Z, U, V)$ *and let* $B \in \mathbb{M}_{p,n}$ *and* $C \in \mathbb{M}_{m,q}$ *be matrices of ranks* $p \le n$ *and* $m \le q$, *respectively. Then,* $BZC \sim \text{MN}_{p,q}(BM_ZC, BUB^\top, CVC^\top)$. *Moreover,*

$$\text{E}\left[ZZ^\top\right] = \text{trace}\,(V)\,U + M_Z M_Z^\top, \tag{6.53}$$

$$\text{E}\left[Z^\top Z\right] = \text{trace}\,(U)\,V + M_Z^\top M_Z. \tag{6.54}$$

We now notice that by (6.48) and (6.50), we have that

$$\widehat{W}_\lambda = R_\lambda X A Y^\top = R_\lambda X A X^\top W + R_\lambda X A E^\top.$$

Given that $XAX^\top = R_\lambda^{-1} - \lambda T \mathbb{I}_N$, this identity can be rewritten as,

$$\widehat{W}_\lambda - W + \lambda T R_\lambda W = R_\lambda X A E^\top.$$

Now, as $E^\top \sim \mathrm{MN}(\mathbb{O}_{T,q}, \mathbb{I}_T, \Sigma_\varepsilon^q)$, by Lemma 6.3, we conclude that

$$\left(\widehat{W}_\lambda - W + \lambda T R_\lambda W\right)\Big|_X \sim \mathrm{MN}\left(\mathbb{O}_{N,q}, R_\lambda X A A^\top X^\top R_\lambda, \Sigma_\varepsilon^q\right).$$

The statement in (6.52) follows from noticing that $AA^\top = A$. ■

6.4.2 Proof of Proposition 3.3

The proof of this result is a consequence of the following statement in which we consider a regression model without intercept. The result can be extended in a straightforward manner in order to accommodate the presence of an intercept by using the observation in (6.44).

Proposition 6.4 *Consider a regression model as in* (6.51) *that satisfies the assumptions* **(A1)** *and* **(A2)**. *Let* $X \in \mathbb{M}_{N,T}$ *be a fixed finite sample realization of* $\{\mathbf{x}(t)\}_{t\in\mathbb{N}}$ *of length* T. *The total mean square error* $\mathrm{MSE}_{\mathrm{total},\lambda} \mid X$ *conditional on* X *committed when using empirical estimates of* $\widehat{W}_\lambda$ *based on finite sample realizations of the form* (6.50) *is defined by*

$$\mathrm{MSE}_{\mathrm{total},\lambda} \mid X := \frac{1}{T}\mathrm{trace}\left[\mathrm{E}\left[\left(Y - \widehat{W}_\lambda^\top X\right)^\top \left(Y - \widehat{W}_\lambda^\top X\right)\Big| X\right]\right].$$

Then,

$$\mathrm{MSE}_{\mathrm{total},\lambda} \mid X = \mathrm{trace}\left(\Sigma_\varepsilon^q\right) + \frac{1}{T}\mathrm{trace}\left[\mathrm{trace}\left(\Sigma_\varepsilon^q\right)\left(R_\lambda X A X^\top \left(R_\lambda X X^\top - 2\mathbb{I}_N\right)\right) + \lambda^2 T^2 R_\lambda W W^\top R_\lambda X X^\top\right]. \tag{6.55}$$

Proof. Let $M_\lambda := \widehat{W}_\lambda - W$. The total error can be rewritten in terms of this random variable as

$$\begin{aligned}
\mathrm{MSE}_{\mathrm{total},\lambda} \mid X :&= \frac{1}{T}\mathrm{trace}\left[\mathrm{E}\left[\left(Y - \widehat{W}_\lambda^\top X\right)^\top \left(Y - \widehat{W}_\lambda^\top X\right)\Big| X\right]\right] \\
&= \frac{1}{T}\mathrm{trace}\left[\mathrm{E}\left[\left(E - M_\lambda^\top X\right)^\top \left(E - M_\lambda^\top X\right)\Big| X\right]\right] \\
&= \frac{1}{T}\mathrm{trace}\left[\mathrm{E}\left[EE^\top\right]\right] + \frac{1}{T}\mathrm{trace}\left[\mathrm{E}\left[X^\top M_\lambda M_\lambda^\top X \mid X\right]\right] - \frac{2}{T}\mathrm{trace}\left[\mathrm{E}\left[E^\top M_\lambda^\top X \mid X\right]\right] \\
&= \mathrm{trace}\left(\Sigma_\varepsilon^q\right) + \frac{1}{T}\mathrm{trace}\left[M_\lambda M_\lambda^\top \mid X\right] X X^\top - \frac{2}{T}\mathrm{trace}\left[\mathrm{E}\left[X^\top M_\lambda E \mid X\right]\right].
\end{aligned} \tag{6.56}$$

We now study separately the last two summands of (6.56). First,

$$\begin{aligned}
\mathrm{trace}\left[\mathrm{E}\left[X^\top M_\lambda E \mid X\right]\right] &= \mathrm{trace}\left[\mathrm{E}\left[\left(\widehat{W}_\lambda - W\right) E \mid X\right]\right] = \mathrm{trace}\left[X^\top\left(\mathrm{E}\left[\widehat{W}_\lambda E \mid X\right] - \mathrm{E}\left[WE \mid X\right]\right)\right] \\
&= \mathrm{trace}\left[X^\top R_\lambda X A \cdot \mathrm{E}\left[\left(X^\top W E + E^\top E\right) \mid X\right]\right] = \mathrm{trace}\left(\Sigma_\varepsilon^q\right)\mathrm{trace}\left(X^\top R_\lambda X A\right).
\end{aligned} \tag{6.57}$$

Regarding the second summand, we note that by Proposition 6.2 and the equality (6.53)

$$\mathrm{trace}\left[M_\lambda M_\lambda^\top \mid X\right] X X^\top = \mathrm{trace}\left(\Sigma_\varepsilon^q\right) R_\lambda X A X^\top R_\lambda X X^\top + \lambda^2 T^2 R_\lambda W W^\top R_\lambda X X^\top. \tag{6.58}$$

The substitution of (6.57) and (6.58) in (6.56) yield (6.55), as required. ■

6.4.3 The large sample limit in the presence of the ergodicity hypothesis

We start by recalling a definition that will be used in the following paragraphs. Additional details about it can be found in [Hami 94].

Definition 6.5 *A covariance-stationarity vector process* $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ *is said to be ergodic for second moments if for all* $j \in \mathbb{Z}$*:*

$$\frac{1}{T-j}\sum_{t=j+1}^{T}(\mathbf{z}(t)-\mathrm{E}[\mathbf{z}(t)])(\mathbf{z}(t-j)-\mathrm{E}[\mathbf{z}(t)])^{\top} \xrightarrow[T\to\infty]{\text{dist}} \Gamma(j), \tag{6.59}$$

where Γ *denotes the autocovariance function of* $\{\mathbf{z}(t)\}_{t\in\mathbb{Z}}$ *and the symbol* $\xrightarrow[T\to\infty]{\text{dist}}$ *denotes convergence in distribution.*

We start by noting that the ergodicity for second moments of the joint process $\{(\mathbf{x}(t),\mathbf{y}(t))\}_{t\in\mathbb{N}}$ implies, using the notation used so far, that

$$\frac{1}{T}XAX^{\top} \xrightarrow[T\to\infty]{\text{dist}} \Gamma(0)=\mathrm{Cov}(\mathbf{x}(t),\mathbf{x}(t)), \tag{6.60}$$

$$TR_{\lambda}=\left(\frac{1}{T}XAX^{\top}+\lambda\mathbb{I}_N\right)^{-1} \xrightarrow[T\to\infty]{\text{dist}} \left(\Gamma(0)+\lambda\mathbb{I}_N\right)^{-1}, \tag{6.61}$$

$$\frac{XX^{\top}}{T} \xrightarrow[T\to\infty]{\text{dist}} \Gamma(0)+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^{\top}. \tag{6.62}$$

Using these relations, we can easily conclude using (6.52) in Proposition 6.2) that the variance of the estimator $(\widehat{W}_{\lambda},\widehat{\boldsymbol{a}}_{\lambda})$ of $(W_{\lambda},\boldsymbol{a}_{\lambda})$ tends to zero as the sample size tends to infinity. This fact is proved in the following proposition, which is stated, without loss of generality in view of the observation in (6.44), for a regression model without intercept.

Proposition 6.6 *Consider a regression model as in* (6.51) *that satisfies the assumptions* **(A1)** *and* **(A2)** *and where additionally, the joint process* $\{(\mathbf{x}(t),\mathbf{y}(t))\}_{t\in\mathbb{N}}$ *is ergodic for second moments. Let* $\{X_i\}_{i\in\mathbb{N}}$ *be a sequence of nested realizations of* $\{\mathbf{x}(t)\}_{t\in\mathbb{N}}$ *of length* i*, that is,* $X_i \in \mathbb{M}_{N,i}$ *and that for each* $i \in \mathbb{N}$*, the matrix* X_{i+1} *is obtained from* X_i *by adding a new column on the right. Under these hypotheses*

$$\mathrm{MSE}_{\mathrm{total},\lambda} \mid X_i \xrightarrow[i\to\infty]{\text{dist}} \mathrm{MSE}_{\mathrm{char},\lambda}. \tag{6.63}$$

Proof. We start by noting that, using the conventions introduced in this section (no intercept), the characteristic error $\mathrm{MSE}_{\mathrm{char},\lambda}$ in (2.12) and in (3.3) (that considers an intercept) can be easily rewritten as

$$\mathrm{MSE}_{\mathrm{char},\lambda}=\mathrm{trace}\left(\Sigma_{\varepsilon}^{q}\right)+\mathrm{trace}\left[\left(W-W_{\lambda}\right)\left(W-W_{\lambda}\right)^{\top}\left(\Gamma(0)+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^{\top}\right)\right]. \tag{6.64}$$

Now, by expression (6.55) that quantifies the total errors $\mathrm{MSE}_{\mathrm{total},\lambda} \mid X_i$, we have:

$$\mathrm{MSE}_{\mathrm{total},\lambda} \mid X_i = \mathrm{trace}\,(\Sigma_\varepsilon^q) + \frac{1}{i}\mathrm{trace}\Big[\mathrm{trace}\,(\Sigma_\varepsilon^q)\left(R_{\lambda,\,i}X_iA_iX_i{}^\top\left(R_{\lambda,\,i}X_iX_i{}^\top - 2\mathbb{I}_N\right)\right)$$

$$+\lambda^2 i^2 R_{\lambda,\,i}WW^\top R_{\lambda,\,i}X_iX_i{}^\top\Big] = \mathrm{trace}\,(\Sigma_\varepsilon^q) + \frac{1}{i}\mathrm{trace}\,(\Sigma_\varepsilon^q)\,\mathrm{trace}\left(\underbrace{iR_{\lambda,\,i}}_{\textbf{(a)}}\underbrace{\frac{X_iA_iX_i{}^\top}{i}}_{\textbf{(b)}}\left(\underbrace{iR_{\lambda,\,i}}_{\textbf{(c)}}\underbrace{\frac{X_iX_i{}^\top}{i}}_{\textbf{(d)}} - 2\mathbb{I}_N\right)\right)$$

$$+\lambda^2\mathrm{trace}\left(\underbrace{iR_{\lambda,\,i}}_{\textbf{(e)}}WW^\top\underbrace{iR_{\lambda,\,i}}_{\textbf{(f)}}\underbrace{\frac{XX^\top}{i}}_{\textbf{(g)}}\right), \quad (6.65)$$

with $A_i = \mathbb{I}_i - \frac{1}{i}\mathbf{i}_i\mathbf{i}_i^\top$ and $R_{\lambda,i} = \left(X_iA_iX_i^\top + \lambda i\mathbb{I}_i\right)^{-1}$.

The ergodicity hypothesis and the relations (6.60)-(6.62) imply that the parts **(a)**, **(c)**, **(e)**, and **(f)** in the previous expression converge to $(\Gamma(0)+\lambda\mathbb{I}_N)^{-1}$, **(b)** converges to $\Gamma(0)$, and **(d)** and **(g)** converge to $\Gamma(0)+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^\top$. Consequently, the second summand in (6.65) converges to zero due to the $1/i$ in its front and the third one to

$$\mathrm{trace}\left[\lambda^2\left(\Gamma(0)+\lambda\mathbb{I}_N\right)^{-1}WW^\top\left(\Gamma(0)+\lambda\mathbb{I}_N\right)^{-1}\left(\Gamma(0)+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^\top\right)\right]$$

$$= \mathrm{trace}\left[\left(W-W_\lambda\right)\left(W-W_\lambda\right)^\top\left(\Gamma(0)+\boldsymbol{\mu}_x\boldsymbol{\mu}_x^\top\right)\right],$$

where the last equality is a consequence of (6.49). This relation, together with (6.64) proves the statement (6.63). ■

References

[Appe 11] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. "Information processing using a single dynamical node as complex system". *Nature Communications*, Vol. 2, p. 468, Jan. 2011.

[Atiy 00] A. F. Atiya and A. G. Parlos. "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*, Vol. 11, No. 3, pp. 697–709, Jan. 2000.

[Brun 13] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer. "Parallel photonic information processing at gigabyte per second data rates using transient states". *Nature Communications*, Vol. 4, No. 1364, 2013.

[Croo 07] N. Crook. "Nonlinear transient computation". *Neurocomputing*, Vol. 70, pp. 1167–1176, 2007.

[Crut 10] J. P. Crutchfield, W. L. Ditto, and S. Sinha. "Introduction to focus issue: intrinsic and designed computation: information processing in dynamical systems-beyond the digital hegemony". *Chaos (Woodbury, N.Y.)*, Vol. 20, No. 3, p. 037101, Sep. 2010.

[Grig 14] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. "Stochastic time series forecasting using time-delay reservoir computers: performance and universality". *Neural Networks*, Vol. 55, pp. 59–71, 2014.

[Grig 15] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. "Optimal nonlinear information processing capacity in delay-based reservoir computers". *Scientific Reports*, Vol. 5, No. 12858, pp. 1–11, 2015.

[Grig 16] L. Grigoryeva and J.-P. Ortega. "Multidimensional ridge regression: generalization error with estimated parameters". *Preprint*, 2016.

[Gupt 00] A. Gupta and D. Nagar. *Matrix Variate Distributions*. Chapman and Hall/CRC, 2000.

[Hami 94] J. D. Hamilton. *Time series analysis*. Princeton University Press, Princeton, NJ, 1994.

[Hast 13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, second Ed., 2013.

[Holm 88] B. Holmquist. "Moments and cumulants of the multivariate normal distribution". *Stochastic Analysis and Applications*, Vol. 6, No. 3, pp. 273–278, Jan. 1988.

[Iked 79] K. Ikeda. "Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system". *Optics Communications*, Vol. 30, No. 2, pp. 257–261, Aug. 1979.

[Jaeg 01] H. Jaeger. "The 'echo state' approach to analysing and training recurrent neural networks". Tech. Rep., German National Research Center for Information Technology, 2001.

[Jaeg 04] H. Jaeger and H. Haas. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". *Science*, Vol. 304, No. 5667, pp. 78–80, 2004.

[Jaeg 07] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. "Optimization and applications of echo state networks with leaky-integrator neurons". *Neural Networks*, Vol. 20, No. 3, pp. 335–352, 2007.

[Larg 12] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing". *Optics Express*, Vol. 20, No. 3, p. 3241, Jan. 2012.

[Luko 09] M. Lukoševičius and H. Jaeger. "Reservoir computing approaches to recurrent neural network training". *Computer Science Review*, Vol. 3, No. 3, pp. 127–149, 2009.

[Lutk 05] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, 2005.

[Maas 02] W. Maass, T. Natschläger, and H. Markram. "Real-time computing without stable states: a new framework for neural computation based on perturbations". *Neural Computation*, Vol. 14, pp. 2531–2560, 2002.

[Maas 11] W. Maass. "Liquid state machines: motivation, theory, and applications". In: S. S. Barry Cooper and A. Sorbi, Eds., *Computability In Context: Computation and Logic in the Real World*, Chap. 8, pp. 275–296, 2011.

[Mack 77] M. C. Mackey and L. Glass. "Oscillation and chaos in physiological control systems". *Science*, Vol. 197, pp. 287–289, 1977.

[Meye 00] C. Meyer. *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*. Society for Industrial and Applied Mathematics, 2000.

[Orti 12] S. Ortin, L. Pesquera, and J. M. Gutiérrez. "Memory and nonlinear mapping in reservoir computing with two uncoupled nonlinear delay nodes". In: *Proceedings of the European Conference on Complex Systems*, pp. 895–899, 2012.

[Paqu 12] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. "Optoelectronic reservoir computing". *Scientific reports*, Vol. 2, p. 287, Jan. 2012.

[Roda 11] A. Rodan and P. Tino. "Minimum complexity echo state network.". *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, Vol. 22, No. 1, pp. 131–44, Jan. 2011.

[Tria 03] K. Triantafyllopoulos. "On the central moments of the multidimensional Gaussian distribution". *The Mathematical Scientist*, Vol. 28, pp. 125–128, 2003.

[Vers 07] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. "An experimental unification of reservoir computing methods". *Neural Networks*, Vol. 20, pp. 391–403, 2007.