Input-to-state representation in linear reservoirs dynamics
Verzelli, Pietro; Alippi, Cesare; Livi, Lorenzo; Tino, Peter

DOI:
10.1109/TNNLS.2021.3059389

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.


General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

• Users may freely distribute the URL that is used to identify this publication.
• Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
• Users may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
• Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 22. Nov. 2021
Reservoir computing is a popular approach to design recurrent neural networks, due to its training simplicity and approximation performance. The recurrent part of these networks is not trained (e.g., via gradient descent), making them appealing for analytical studies by a large community of researchers with backgrounds spanning from dynamical systems to neuroscience. However, even in the simple linear case, the working principle of these networks is not fully understood and their design is usually driven by heuristics. A novel analysis of the dynamics of such networks is proposed, which allows the investigator to express the state evolution using the controllability matrix. Such a matrix encodes salient characteristics of the network dynamics; in particular, its rank represents an input-independent measure of the memory capacity of the network. Using the proposed approach, it is possible to compare different reservoir architectures and explain why a cyclic topology achieves favourable results as verified by practitioners.

Keywords Reservoir Computing · Recurrent Neural Networks · Dynamical Systems

1 Introduction

Despite of being applied to a large variety of tasks, Recurrent Neural Networks (RNNs) are far from being fully understood and performance improvement is usually driven by heuristics. Understanding how the computation is conducted by the dynamics of the RNNs is an old question [1] which still remains unanswered, even though advances were recently achieved [2,3]. The introduction of gating mechanisms (such as LSTM [4] and GRU [5]) dramatically improved the performance of RNNs, but the use of complex architectures makes the theoretical analysis harder [6,7,8]. Even for simple networks, we currently lack a sound framework to describe how the signal history is encoded in the state. Another relevant issue is the memory-nonlinearity trade-off [9,10]. Memory capacity maximization does not necessarily imply performance (e.g., prediction) maximization [11]. In recent years, large efforts have been devoted to tackle these problems, by studying the dynamical systems behind RNNs [2,12,13,14,15].
Reservoir Computing (RC) is a computational paradigm developed independently by Jaeger [16, 17] (Echo State Networks (ESNs)) and Maas [18] (Liquid State Machines (LSMs)) and Tino [19] (Fractal Predicting Machines (FPMs)). The basic idea is to create a representation of the input signal using an untrained RNN, called the reservoir, and then to use a trainable readout layer to generate the network output. RC demonstrated its effectiveness in various tasks and has risen great interest in the physical computing community due to the underlying idea of natural computation. In particular, photonics [20] and neuromorphic computation [21] are commonly implemented using RC, but also a bucket of water [22] or road traffic [23] have been used as reservoirs. See [24] for a review.

The architecture simplicity makes RC prone to theoretical investigations [25, 26, 27, 15, 12, 28], which have mainly concentrated on questions about computational capabilities of whole classes of dynamical systems [20], while little has been understood about how specific setting of the dynamical system can influence the network computational property [15]. Some important theoretical results can be derived assuming that the reservoir dynamics are linear [12, 11, 15, 28, 29]. This assumption can be seen as a first-order approximation of the nonlinear system, sharing some – but not all – of its features [30].

In this work, we propose a novel analysis shedding light on how a linear RNN encodes input signals in its state. The analysis decouples the network state in two terms: the controllability matrix \( C \), which only depends on the reservoir and input weights vector, and the network encoded input \( s \), which depends on the reservoir weights and the input signal driving the system. Writing the system in this form, allows one to decouple the properties of the reservoir topology – encoded in \( C \) – from the specific input driving the network, encoded in \( s \). We analyze different reservoir topologies in terms of the nullspace of \( C \) and show that the rank of \( C \) is a measure for the richness of the representation of the input signal. More specifically, we show that the nullspace of \( C \) is linked to the memory forgetting capacity of the network. Based on these results, we demonstrate that a cyclic reservoir topology (forming a ring structure) is optimal. The claim is corroborated by empirical evidence.

The remaining of the paper is organized as follows. In Section 2 the RC paradigm is presented, along with the various reservoir topologies studied in this work. Section 3 develops a representation of the network output in terms of its controllability matrix. Section 4 is devoted to the analysis of the input representation in the network’s states, while Section 6 discusses the memory of the network in terms of the rank of the controllability matrix. In Section 6 some experiments are conducted to validate are claims. Finally, Section 7 draws the conclusion. The Appendices are dedicated to derivations.

2 Reservoir computing

RC was developed as a tool to explain the brain working principle [18] and as computational paradigm to avoid the complex and expensive training procedure of RNNs [16], e.g., based on backpropagation through time [31, 32]. RC reads the input signal through an input layer, resulting in an untrained representation of the input in the network’s state. A readout layer is then trained to produce the system output. Common tasks involve time-series prediction [33, 34], simulation of dynamical systems [35], and time-series classification [36].

Let \( x_k \in \mathbb{R}^n \) be the state of a reservoir of dimension \( n \) at time \( k \), \( W \in \mathbb{R}^{n \times n} \) be its reservoir connection matrix and \( w \in \mathbb{R}^n \) its input weights vector. When considering left-infinite signal, the time-index \( k \) runs from 0 to \( -\infty \), so that the driving input signal is \( u = (u_0, u_{-1}, u_{-2}, \ldots) \), \( u_{-k} \in \mathbb{R} \).

In the linear case, the reservoir evolves according to:

\[
x_k = W x_{k-1} + w u_k
\]

By recursively applying (1) we get

\[
x_0 = W x_{-1} + w u_0 = W^2 x_{-2} + W w u_{-1} + w u_0
\]

\[
= W^3 x_{-3} + W^2 w u_{-2} + W w u_{-1} + w u_0
\]

i.e.,

\[
x_0 = \sum_{k=0}^{\infty} W^k w u_{-k}
\]

One usually relies on a (trained) linear readout \( r \in \mathbb{R}^n \) to generate the system output \( y_{-k} \), so that at time 0

\[
y_0 = r \cdot x_0 = r \cdot \sum_{k=0}^{\infty} W^k w u_{-k}
\]
Figure 1: Schematic representation of a RC with a scalar input, a uni-dimensional output and $N = 8$ neurons in the reservoir. Fixed connections are drawn using solid lines while dashed lines represent learned connections.

In the seminal paper [16], training is based on a simple least-square regression. More sophisticated techniques where later introduced, including some forms of regularization [37] and online-training procedures [38]. For simplicity, we have only described the case in which the input and the output are uni-dimensional, but the proposed approach can be easily generalized to the multidimensional case. In Fig. 1 a schematic representation of the RC architecture is depicted.

Even though a reservoir layer does not require a proper training procedure, hyper-parameters tuning must be carried out in to improve performance. The most studied hyper-parameter is the Spectral Radius (SR) $\rho(W)$ [39, 40, 41] which is the largest eigenvalue in magnitude of $W$, related to the Maximum Singular Value (MSV) $\sigma_{\max}(W)$. Other hyper-parameters refer to the input and output scaling factors, the sparsity degree of $W$ and the input bias. Moreover, the update equation (1) is usually chosen to be non-linear, i.e., $x_k = \phi(Wx_{k-1} + w u_k)$ and different choices of the nonlinear transfer function $\phi$ can be explored [42].

Many studies were devoted to understand how these hyper-parameters affect the dynamics of the network and its computational capabilities. In particular, it appears that the hyper-parameter space can be divided into a region where the dynamics are “regular” (meaning that they are stable with respect to the inputs driving the system) and another one where they are “disordered” (meaning that they are unstable and do not provide a representation for the input) [43]. The narrow region separating these two regimes is known in the literature as Edge of Chaos or Edge of Criticality (EoC) [44, 45, 46] and appears to be common to a large variety of complex systems beyond RNNs [47, 48, 49].

In the following, we introduce different reservoir architectures that are commonly found in the literature.

### 2.1 Delay line

In a delay line each neuron is connected the subsequent one to form a chain-like architecture, so that the reservoir connection matrix $W_d$ reads:

$$W_{d,ij} = \delta_{i,j-1} \quad (6)$$

where $\delta$ is the Kronecker delta. Note that the last neuron in the chain is not connected to the first one. Moreover, the input weights vector is $w_d = (1, 0, 0, \ldots, 0)$, meaning that the input enters the network only through the first neuron of the chain. Mathematically, such a model setting corresponds to the $n$-th order AR model, which is a really popular and studied tool in time series analysis and system identification [50].
2.2 Cyclic reservoir

A reservoir is said to be cyclic when each neuron is connected to another one, in a way that they form a ring. The reservoir matrix of a cyclic reservoir has the form:

\[ W_{c,ij} = \delta_{i,j-1} \] (7)

where, with an abuse of notation, \( \delta_{0,-1} := \delta_{0,n-1} \). From the product of Kroenecker deltas, it follows that:

\[ W_{c,ij}^2 = \sum_k W_{c,ik} W_{c,kj} = \sum_k \delta_{i,k-1} \delta_{k,j-1} = \delta_{i,j-2} \] (8)

as well for higher powers.

2.3 Random reservoir

In a random reservoir the entries of the reservoir matrix \( W_r \) are independent random variables. In the sequel, we consider the generic \( ij \) component of the matrix to be drawn from a Gaussian distribution.

\[ W_{r,ij} \sim \mathcal{N} \left( 0, \frac{\rho^2}{n} \right) \] (9)

With this choice, the expected value for the SR \( \langle \rho(W_r) \rangle = \rho \) [13] and MSV \( \langle \sigma_{\text{max}}(W_r) \rangle = 2\rho \) [51].

2.4 Wigner reservoir

The diagonal elements are distributed as in (9), i.e., \( W_{w,ii} \sim \mathcal{N}(0, \rho_1^2/n) \), while off-diagonal elements follow:

\[ W_{w,ij} = W_{r,ji} \sim \mathcal{N} \left( 0, \frac{\rho_2^2}{n} \right), \quad i \neq j \] (10)

Wigner matrices are symmetric. In this work, we will always set \( 2\rho_1 = \rho_2 = \rho \). Notably, this leads to \( \langle \rho(W_t) \rangle = \langle \sigma_{\text{max}}(W_t) \rangle = \rho \).

3 Controllability matrix and network encoded input

Here we develop a representation for the network state evolution based on the Cayley-Hamilton (CH) theorem [52]. The CH theorem states that every real square matrix satisfies its own characteristic equation, implying that

\[ W^n = \varphi_{n-1} W^{n-1} + \varphi_{n-2} W^{n-2} + \cdots + \varphi_1 W + \varphi_0 I \] (11)
where the $\varphi_i$ are the negated coefficient of the characteristic polynomial (see Appendix A for details). Accordingly, any power of matrix $W$ can be written as a linear combination of the first $n - 1$ powers, where $n$ is the matrix order (and also the size of the reservoir):

$$W^k = \sum_{j=0}^{n-1} \varphi_j^{(k)} W^j$$

(12)

where the apex $k$ denotes the fact that the $n$ coefficients are expansion coefficients of the $k$-th power of $W$. In Appendix B we also show how the coefficients $\varphi_j^{(k)}$ can be written in terms of $\varphi_j$ in (11). By inserting (12) in (4), we obtain:

$$x_0 = \sum_{k=0}^{\infty} \sum_{j=0}^{n-1} \varphi_j^{(k)} W^j w u_{-k}$$

(13)

$$= \sum_{j=0}^{n-1} W^j w \sum_{k=0}^{\infty} \varphi_j^{(k)} u_{-k} = \sum_{j=0}^{n-1} W^j w s_j$$

(14)

where

$$s_j := \sum_{k=0}^{\infty} \varphi_j^{(k)} u_{-k}$$

(15)

is what we call the network encoded input. It is useful to interpret $s = (s_0, s_1, \ldots, s_{n-1})$ as a vector with $n$ components, which “encodes” the left-infinite input signal $u$ in the spatial representation provided by the network. In order for the $s_j$ term to exist, the sum in (15) must converge; we will discuss this issue in the next section. We emphasize the fact that the sum over $j$ (the dimensionality of our system) is a finite sum with $n$ terms, as opposed to the infinite sum over the $k$ (the time index).

Inspired by well-known tools from control theory [53], we define the controllability matrix of the reservoir as

$$C = \begin{bmatrix} w & W w & W^2 w & \cdots & W^{n-1} w \end{bmatrix}$$

(16)

Then, the state-update equation (4) becomes

$$x_0 = C s$$

(17)

and the output (5) can then be expressed as:

$$y_0 = r C s$$

(18)

i.e., the readout filters the input according to controllability matrix.

### 4 How the network encodes the input signal

From (18), we see that the possibility for the readout to produce the correct output (i.e., the output that solves the task at hand depends on two distinct elements: the controllability matrix $C$ (function of $W$ and $w$) and the network encoded input $s$ (which depends on $W$ and $u$).

In Appendix B we show that the coefficients $\varphi_i^{(k+1)}$ of (12) can be recursively expressed in terms of $\varphi_i^{(k)}$ as:

$$\begin{bmatrix} \varphi_0^{(k+1)} \\ \varphi_1^{(k+1)} \\ \vdots \\ \varphi_{n-1}^{(k+1)} \end{bmatrix} = M \begin{bmatrix} \varphi_0^{(k)} \\ \varphi_1^{(k)} \\ \vdots \\ \varphi_{n-1}^{(k)} \end{bmatrix}$$

(19)

where $M$ is the Frobenius companion matrix of $W$ (see Appendix B for details). Note that the characteristic polynomial of $M$ is that of $W$; as such, the two matrices share the same eigenvalues. Thus, the series (15) converges, for bounded inputs, when $W$ has a spectral radius smaller than 1. Note that most theoretical results (see e.g., [16, 26, 15]) require the MSV to be smaller than one, which is a stricter condition (as $SR \leq MSV$); it follows that our analysis can be applied to a larger class of reservoirs.
The $s$ vector can be written as:

\[
\begin{bmatrix}
  s_0 \\
  s_1 \\
  \vdots \\
  s_{n-2} \\
  s_{n-1}
\end{bmatrix} = \begin{bmatrix}
  \sum_{k=0}^{\infty} \phi_0^{(k)} u_{-k} \\
  \sum_{k=0}^{\infty} \phi_1^{(k)} u_{-k} \\
  \vdots \\
  \sum_{k=0}^{\infty} \phi_{n-2}^{(k)} u_{-k} \\
  \sum_{k=0}^{\infty} \phi_{n-1}^{(k)} u_{-k}
\end{bmatrix}
\]

(20)

In Appendix A, we show that for $k < n$, $\phi_j^{(k)} = \delta_{kj}$ holds. We also note that terms corresponding to time-step $k = n$ follow from (11). This means that (4) can be written as:

\[
\begin{bmatrix}
  s_0 \\
  s_1 \\
  \vdots \\
  s_{n-2} \\
  s_{n-1}
\end{bmatrix} = \begin{bmatrix}
  u_0 \\
  u_1 \\
  \vdots \\
  u_{-(n-2)} \\
  u_{-(n-1)}
\end{bmatrix} + \begin{bmatrix}
  u_{-n}\varphi_0 \\
  u_{-n}\varphi_1 \\
  \vdots \\
  u_{-n}\varphi_{n-2} \\
  u_{-n}\varphi_{n-1}
\end{bmatrix} + \begin{bmatrix}
  \sum_{k=n+1}^{\infty} \phi_0^{(k)} u_{-k} \\
  \sum_{k=n+1}^{\infty} \phi_1^{(k)} u_{-k} \\
  \vdots \\
  \sum_{k=n+1}^{\infty} \phi_{n-2}^{(k)} u_{-k} \\
  \sum_{k=n+1}^{\infty} \phi_{n-1}^{(k)} u_{-k}
\end{bmatrix}
\]

(21)

All other terms in (21) corresponding to time steps $k > n$ can be computed according to (12).

This procedure shows that, in general, the inputs from 0 to $n - 1$ steps back in time will always appear in their original form, and the cross-contribution starts only from $u_{-n}$ backwards in time. We will make use of this result to analytically examine the properties of different networks topologies. Moreover, by deriving the expression for the $\phi_j^{(k)}$, we can study how the network is able to recall its past inputs. In general, if the $\phi_j^{(k)} s$ are large then the network will not be able to recall the inputs, since the input $u_{-j}$ can only be read through $s_j = u_{-j} + \sum_{k=n}^{\infty} \phi_j^{(k)} u_{-k}$. It follows that having large expansion coefficients $\phi_j^{(k)}$ prevents the network from being able to recall its past inputs. We will show in the next section that, when we can derive an analytical expression for the $\phi_j^{(k)}$, it is possible to anticipate how the network recalls its past inputs. Note that, as implied by (18), for a linear network this is deeply related to its expressive power, since the network output is basically a linear combination of past inputs. The inputs accessibility to the readout is also due to $C$, which is a property of the network only, as it does not depend on any input signal. A detailed discussion about the relation between the network properties and the rank of the controllability matrix $C$ was recently presented in [54]. There, the authors prove that the memory capacity for linear reservoirs equals the rank of $C$. In the following, we analyze the different architectures described in Section 2 discussing the properties of their controllability matrices.

In the random reservoir case (Subsection 2.2), the property of $C$ can be studied by considering the expected values of the norm of its columns, which describes how the system accesses past inputs. Let us consider the $n$-by-$n$ matrix $W_r$ in (9) and a vector with $n$ components $w = \{w_j\} \sim N(0, \frac{1}{n})$. Since $w_j$ are generated independently, the expected value of the squared norm of the random vector $w$ is $\bar{l}(w) = n\langle w_j^2 \rangle$. We drop the $r$ in $W_r$ to simplify the notation. We now study $z := Ww$ and obtain:

\[
\langle z_i^2 \rangle = \langle (Ww)^2 \rangle = \langle \sum_j W_{ij} w_j \rangle^2 = n\langle W_{ij}^2 \rangle \langle w_j^2 \rangle
\]

(22)

where the last equality follows from the independence of the zero-mean entries of $W$ and $w$. Now, by the way we constructed $W$ and $w$, we see that $\langle W_{ij}^2 \rangle = \rho^2/n$ and that $\langle w_j^2 \rangle = 1/n$. This results in:

\[
\langle z_i^2 \rangle = n\langle W_{ij}^2 \rangle \langle w_j^2 \rangle = n\rho^2 \frac{1}{n} = \frac{\rho^2}{n}
\]

(23)

This means that $l(z) = n\langle z_i^2 \rangle = \rho^2$ and that the standard deviation is $\sqrt{\langle z_i^2 \rangle} = \frac{\rho}{\sqrt{n}}$.

From the above the first column of $C$ has euclidean norm $\|w\| = 1$, the second $\rho$, the third $\rho^2$, the last one $\rho^{(n-1)}$. 

\[\mbox{A preprint - February 15, 2021}\]
Since \( \rho \) must be smaller than 1, the components of last columns of \( \mathcal{C} \) shrink quickly. This fact explains the shading observed in the column of \( \mathcal{C} \) for the random case of Fig. 3 and Fig. 4. For the Wigner case the effect is emphasized by the correlations introduced by the symmetry of \( W_w \). The controllability matrix \( \mathcal{C} \) for the delay line and the cyclic reservoir can instead be described in exact terms (see Appendices C and D). A sample of each case in provided in Fig. 5 and Fig. 4 for \( n = 100 \) and \( n = 1000 \), respectively. For the delay line a complete analysis of the network output can be carried. As shown in Appendix C, we can write that:

\[
y_0 = r \cdot I \cdot s_0 = \sum_{i=0}^{n-1} r_i u_{-i}
\]

where \( I \) is the identity matrix. This is, as one would expect, simply a regression model of order \( n \).

For the cyclic reservoir architecture (Subsection 2.2) we show in Appendix B that defined the \( i \)-time permuted input weights vector as:

\[
w^{(i)} := W_c^{i} w
\]

then, the output of the cyclic reservoir at time zero \( y_0 \) is:

\[
y_0 = r \cdot \tilde{\mathcal{C}}_c \cdot \tilde{s}
\]

where

\[
\tilde{s}_j = \sum_{p=0}^{\infty} \rho^{j+pn} u_{-j+pn}
\]

\[
\tilde{\mathcal{C}}_c = \begin{bmatrix} w & w^{(1)} & w^{(2)} & \ldots & w^{(n-1)} \end{bmatrix}
\]

The fact that, as suggested in [27], \( w \) should be non-periodic for the network to work at its best, is now evident. In fact, if \( w \) is periodic, it means that some columns of \( \tilde{\mathcal{C}}_c \) are linearly related and, therefore, the rank degenerates, as supported by theoretical arguments in [15].

Note that \( u_{-j} \) is only readable through the term \( s_j = u_{-j} + \rho^n u_{-(j+n)} + \ldots \) and, in order to do that, it must hold that \( u_{-j} \gg \rho^n u_{-(j+n)} \). This may suggest to choose small spectral radii, but the smaller the spectral radius, the faster the decay of the memory, since \( \tilde{s}_j = \rho^j s_j \). This confirms previous intuitions [27], stating that by choosing a small spectral radius, the network preserves an accurate representation of recent inputs, at the expense of losing the ability to recall remote ones. Conversely, if one sets a large spectral radius (i.e., close to 1) the network will be able to (partially) recall inputs from the past, but its memory of more recent inputs will decrease.

5 The nullspace of \( \mathcal{C} \) and the network memory

By [18] one can understand how the rank of the controllability matrix \( \mathcal{C} \) is associated with the degrees of freedom (the effective number of parameters used by the model to solve the task at hand) that can be exploited by the readout (i.e., the “complexity” of the model).

Note from Figures 3 and 4 that the cyclic reservoir always has the highest rank of \( \mathcal{C} \), while the Wigner the lowest. The difference increases with the number of neurons.

The fact that \( \mathcal{C} \) is not full-rank is linked to the presence of the nullspace [3]. This means that there are some network encoded inputs \( s \) which are mapped to 0 by \( \mathcal{C} \) [17] and hence are indistinguishable by the readout perspective. In Fig. 5 we plot the rank of \( \mathcal{C} \) as a function of the reservoir dimension \( n \). In the experiments using Wigner and cyclic reservoirs, the spectral radius \( \rho \) and the maximum singular value \( \sigma_{\text{max}} \) coincide and their values are set to 0.995 (Fig. 5a) and 0.9 (Fig. 5b). For the random reservoir, \( \rho \) and \( \sigma_{\text{max}} \) are distinct, so we design an experiment where the spectral radius is fixed and another one where the maximum singular value is set (we remind the reader that \( \langle \rho \rangle = \frac{1}{n} \langle \sigma_{\text{max}} \rangle \)). But what is the shape of the basis of this nullspace? We show its basis in two cases (see Fig. 6). The controllability matrix obtained with a cyclic reservoir does not have a nullspace for such a value of \( \rho \), since \( \mathcal{C} \) is full-rank. Note that, in order to interpret each vector in Fig. 6 as a time series, one must consider the last inputs seen as the ones closer to the origin. Given this interpretation, we clearly see how the memory is linked to the rank of \( \mathcal{C} \): the reservoirs’ ability to recall past inputs depend on the rank of \( \mathcal{C} \) as inputs which only differ in the far-away past are mapped to the same final state.

Note that this is coherent with the findings in [15], since the \( Q \) defined in that work is simply \( Q = \mathcal{C}^\top \mathcal{C} \) and the number of motifs is related to the rank of \( Q \) (and so, of \( \mathcal{C} \)).

What we call the nullspace is practically the effective nullspace detected up to the numerical precision, computed using the Numpy dedicated function [55].
Wigner architectures appear to have a short memory. Their accuracy dramatically decreases as \( \tau \) grows. The behavior of the cyclic reservoir appears to be radically different. As described in [27], the performance does not decrease gradually, but remains almost constant for some time and then abruptly decreases. The drop in performance occurs when \( \tau = n \), where

\[
\text{Normalized Root Mean Squared Error (NRMSE)} := \sqrt{\frac{\sum_{k=t_0}^{T} (y_k - \hat{y}_k)^2}{\sum_{k=t_0}^{T} (y_k - \bar{y})^2}}
\]

Let us consider an example. Let \( s_1 \) and \( s_2 \) be two network encoded inputs, which differ only in the last \( n-m \) elements. Denote by \( x_0 \) the final state of the system after being fed by the signal \( s_t \). We can then write \( s_2 = s_1 + d \), where \( d \) encodes the difference between the two representations. We see that the first \( m \) elements of \( d \) are null. So, we can write:

\[
x_0^2 = C s_2 = C (s_1 + d) = C s_1 + C d = C s_1 + 0 = x_0^1
\]

since \( d \) lives in the nullspace of \( C \). This results in the network not being able to distinguish between the two signals.

### 6 Memory curves

In order to validate our theoretical claims, we designed networks asked to remember a random i.i.d. input. We generate inputs of length \( T \) and split them in a training set ranging in \((0, t_0)\) and a test set ranging in \((t_0, T)\). In the task under consideration, the network is trained to reproduce past input (a white noise signal) at a given past-horizon \( \tau \), so that \( y_k \equiv u_{k-\tau} \). The input signal \( u \) is chosen to be Gaussian i.i.d. white noise, \( u_k \sim \mathcal{N}(0, 1) \). With choice \( T = 1500 \) and \( t_0 = 1000 \), the training set contains \( L_{\text{train}} = 1000 \) sample and the test set \( L_{\text{test}} = 500 \). The readout is finally configured through a least means square procedure and its accuracy is then evaluated on the test set as \( \gamma = \max \{1 - \text{NRMSE}, 0\} \), where the Normalized Root Mean Squared Error (NRMSE):

\[
\text{NRMSE} := \sqrt{\frac{\sum_{k=t_0}^{T} (y_k - \hat{y}_k)^2}{\sum_{k=t_0}^{T} (y_k - \bar{y})^2}}
\]
Figure 5: Ranks of the controllability matrix $C$ as a function of the reservoir dimension $n$, for $\rho = 0.995$ (a) and $\rho = 0.9$ (b). Note the saturation of the delay line and the cyclic reservoir, which happens, because $\rho$ is not close enough to 1 for its powers to be numerically distinguishable from zero.

$n$ being the number of neurons in the reservoir. This is coherent with the theory we developed and with the findings in [56]. Note that in [57] Section 3] a similar shape for the memory curve is obtained by using an “almost unitary” reservoir matrix, where all singular values equal a constant $C < 1$. We note that also the cyclic reservoir $W_c$ shares this feature, explaining why the results are similar.

We comment on how the SR affects the performance: when the SR is close to one, the accuracy in the reconstruction of $u(t - \tau)$ is lower for recent inputs samples (i.e., smaller $\tau$) but higher for distant in time ones. In other words, choosing a large $\rho$ allows the network to better remember the distant past, at the price of compromising its ability to remember
the recent one. This is a direct implication of (27) and (21): a larger SR amplifies the contribution of past inputs over the more recent ones, since the input reproducibility property is controlled by $\rho^{j+pn}$ (see Appendix B for more details).

We investigate the impact of the spectral radius on memory capacity in Figure 8, where the accuracy $\gamma$ of the three architectures in recalling a past input (at various $\tau$) is plotted as function of the SR. According to our predictions, a larger SR is required to correctly recall inputs that are further in past (but for which $\tau < N$), since the SR controls the magnitude of the $\phi_j^{(k)}$, i.e., the permanence of $u_k$ on the state. We notice that the Random and the Wigner architectures show a similar behavior, with the former displaying a superior performance than the latter. Instead, the Cyclic network has the same behavior as the SR increases, but displays an abrupt fall as it approaches 1. This is coherent with the theory we developed, since for $\rho \approx 1$ the powers of $\rho$ in (27) do not shrink towards zero fast enough to forget remote inputs with the consequence that the network state will be an unreadable superposition of all the past outputs.

7 Conclusions

In this paper, we proposed a methodology for explaining how linear reservoirs encode inputs in their internal states. Theoretic findings allow to express the system state in terms of the controllability matrix $C$ and the network encoded input $s$. The matrix properties of $C$ allow us to compare different connectivity patterns for the reservoir in a quantitative way by. Results show that reservoirs with a cyclic topology give the richest possible encoding of input signals, yet they also offer one the most parsimonious reservoir parametrization. To the best of our knowledge, our contribution pioneers the rigorous study on how specific coupling patterns for the recurrent layer and individual setting of the dynamical
system influence computational properties (e.g., memory), providing deeper insights about phenomena that so far have been observed only empirically in the literature.

A Cayley-Hamilton Theorem

The CH Theorem allows one to describe the $n$-th power of a matrix in term of the first $n-1$-powers (including the zero power, which is the identity).

Let $W \in \mathbb{R}^{n \times n}$ be a square matrix. Its characteristic polynomial is defined as:

$$\det(\lambda I - W) = 0 \quad \rightarrow \quad \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0 = 0 \quad (31)$$

where $\lambda$ is an eigenvalue of $W$ and the $\alpha_k$ are the coefficients of the characteristic polynomial.

**Theorem 1 (Cayley-Hamilton)** Every real square matrix satisfies its characteristic equation

$$W^n + \alpha_{n-1}W^{n-1} + \cdots + \alpha_1W + \alpha_0 I = 0. \quad (32)$$

Accordingly, it is possible to show that the $n$-th power of the matrix can be represented as a linear combination of its lower powers:

$$W^n = -\alpha_{n-1}W^{n-1} - \cdots - \alpha_1W - \alpha_0 I \quad (33)$$
Figure 8: Accuracy in remembering an i.i.d. past-input as function of the spectral radius. All the networks have $N = 100$ neurons. Plotted values are averages over 10 different realizations, with the shaded area accounting for a standard deviation: a side effect of plotting data in this way is that the value may be negative even if the accuracy is defined as a positive quantity.

For matrix $W$, Theorem 1 states:

$$W^n = \varphi_{n-1} W^{n-1} + \varphi_{n-2} W^{n-2} + \cdots + \varphi_1 W + \varphi_0 I$$

(34)

Here, $W$ is an $n \times n$ matrix, $I$ is the $n \times n$ identity matrix, and $\varphi_k = -\alpha_k$ are the negated coefficients of the characteristic polynomial of $W$. It holds true that

$$W^m = \phi_{n-1}^{(m)} W^{n-1} + \phi_{n-2}^{(m)} W^{n-2} + \cdots + \phi_1^{(m)} W + \phi_0^{(m)} I$$

(35)

implying that any power $m \geq n$ of $W$ can be specified by $W$ and scalars $(\phi_{n-1}^{(m)}, \ldots, \phi_0^{(m)})$. The apexes denote the fact that the $n$ coefficients are those proper of the $m$-th power for the $\phi_j^{(m)}$ coefficients. Note that, for $m < n$, we have $(\phi_{n-1}^{(i)}, \ldots, \phi_0^{(i)}) = (0, \ldots, 0, 1, 0, \ldots, 0)$, where the only non-zero term is the $m$-th one, i.e.,

$$\phi_j^{(m)} = \delta_{mj} \quad \text{for} \quad m < n$$

(36)

Moreover, note that for $m = n$, $(\phi_{n-1}^{(m)}, \ldots, \phi_0^{(m)}) = (\varphi_{n-1}, \ldots, \varphi_0)$. 

12
For each $m \geq n$, we can derive the scalars in recursive way by noting that:

$$W^{m+1} = W^m W$$

$$= (\phi_{n-1}^{(m)} W^{n-1} + \phi_{n-2}^{(m)} W^{n-2} + \cdots + \phi_1^{(m)} W + \phi_0^{(m)} I) W$$

$$= \phi_{n-1}^{(m)} (\varphi_1 W^{n-1} + \varphi_2 W^{n-2} + \cdots + \varphi_1 W + \varphi_0 I)$$

$$+ \phi_{n-2}^{(m)} W^{n-1} + \cdots + \phi_1^{(m)} W^2 + \phi_0^{(m)} W$$

$$= \varphi_{n-1}^{(m)} (\varphi_1 W^{n-1} + \varphi_2 W^{n-2} + \cdots + \varphi_1 W + \varphi_0 I)$$

$$+ \phi_{n-1}^{(m)} W^{n-1} + \cdots + \phi_1^{(m)} W^2 + \phi_0^{(m)} W$$

$$+ \cdots + (\varphi_1 \phi_{n-1}^{(m)} + \phi_0^{(m)}) W + (\varphi_0 \phi_{n-1}^{(m)}) I$$

$$= \varphi_{n-3}^{(m)} + \cdots + \varphi_1^{(m)} W^2 + \phi_0^{(m)} W$$

which implies

$$\begin{align*}
\phi_0^{(m+1)} &= \varphi_0 \phi_{n-1}^{(m)} \\
\phi_1^{(m+1)} &= \varphi_1 \phi_{n-1}^{(m)} + \phi_0^{(m)} \\
\phi_{n-2}^{(m+1)} &= \varphi_{n-2} \phi_{n-1}^{(m)} + \phi_{n-3}^{(m)} \\
\phi_{n-1}^{(m+1)} &= \varphi_{n-1} \phi_{n-1}^{(m)} + \phi_{n-2}^{(m)} \\
\end{align*}$$

Eq. 42 can be thought as a linear system:

$$\begin{bmatrix}
\phi_0^{(m+1)} \\
\phi_1^{(m+1)} \\
\vdots \\
\phi_{n-2}^{(m+1)} \\
\phi_{n-1}^{(m+1)}
\end{bmatrix}
= M
\begin{bmatrix}
\phi_0^{(m)} \\
\phi_1^{(m)} \\
\vdots \\
\phi_{n-2}^{(m)} \\
\phi_{n-1}^{(m)}
\end{bmatrix}$$

where $M$ is defined as:

$$M =
\begin{bmatrix}
0 & \cdots & \cdots & \cdots & 0 & \varphi_0 \\
1 & 0 & \cdots & 0 & \varphi_1 \\
0 & \cdots & \cdots & \cdots & 0 & \varphi_{n-2} \\
0 & 0 & \cdots & 1 & \varphi_{n-1}
\end{bmatrix}$$

Note that the characteristic polynomial of $M$ is equal to the one of $W$, so that they also share the same eigenvalues. In fact, $M$ is also know as the Frobenius companion matrix of $W$.

B  The network encoded input

The possibility for the readout to produce the correct output for the task at hand depends on two distinct elements: the controllability matrix $C$ (which depends on $W$ and $w$) and the $s$ vector (which depends on both $W$ and the signal $u$). Here we show how $s$ is obtained from $u$.

Under the assumption of bounded inputs $u_{-k} \in [-U, U], \forall k$, we see that

$$|s_j| = \left| \sum_{k=0}^{\infty} \phi_j^{(k)} u_{-k} \right| \leq U \sum_{k=0}^{\infty} \phi_j^{(k)}$$

allowing us to focus on the properties of the $\phi_j^{(k)}$. 

13
These terms are the element of \( s \), which we rewrite as:

\[
\begin{bmatrix}
  s_0 \\
  s_1 \\
  \vdots \\
  s_{n-2} \\
  s_{n-1}
\end{bmatrix} =
\begin{bmatrix}
  \sum_{k=0}^{\infty} \phi_0^{(k)} u_{-k} \\
  \sum_{k=0}^{\infty} \phi_1^{(k)} u_{-k} \\
  \vdots \\
  \sum_{k=0}^{\infty} \phi_{n-2}^{(k)} u_{-k} \\
  \sum_{k=0}^{\infty} \phi_{n-1}^{(k)} u_{-k}
\end{bmatrix}
\]

(45)

In Appendix A we show that for \( k < n \), \( \phi_j^{(k)} = \delta_{kj} \) (Eq. 36). This implies that the first \( n - 1 \) time steps are simply the inputs:

\[
\begin{bmatrix}
  s_0 \\
  s_1 \\
  \vdots \\
  s_{n-2} \\
  s_{n-1}
\end{bmatrix} =
\begin{bmatrix}
  u_0 + \sum_{k=n}^{\infty} \phi_0^{(k)} u_{-k} \\
  u_{-1} + \sum_{k=n}^{\infty} \phi_1^{(k)} u_{-k} \\
  \vdots \\
  u_{-(n-2)} + \sum_{k=n}^{\infty} \phi_{n-2}^{(k)} u_{-k} \\
  u_{-(n-1)} + \sum_{k=n}^{\infty} \phi_{n-1}^{(k)} u_{-k}
\end{bmatrix}
\]

(46)

Then, we observe that the terms corresponding to time-step \( k = n \) follow from Eq. 34

\[
\begin{bmatrix}
  s_0 \\
  s_1 \\
  \vdots \\
  s_{n-2} \\
  s_{n-1}
\end{bmatrix} =
\begin{bmatrix}
  u_0 \\
  u_{-1} \\
  \vdots \\
  u_{-(n-2)} \\
  u_{-(n-1)}
\end{bmatrix} +
\begin{bmatrix}
  \sum_{k=n+1}^{\infty} \phi_0^{(k)} u_{-k} \\
  \sum_{k=n+1}^{\infty} \phi_1^{(k)} u_{-k} \\
  \vdots \\
  \sum_{k=n+1}^{\infty} \phi_{n-2}^{(k)} u_{-k} \\
  \sum_{k=n+1}^{\infty} \phi_{n-1}^{(k)} u_{-k}
\end{bmatrix}
\]

(47)

successive terms corresponding to time steps \( k > n \) can be computed by using (42). This procedure shows that, in general, the inputs from 0 to \( n - 1 \) time steps in the past will always appear in their original form, and the “mixing” will begin starting from the \( n \)-th time step in the past.

C Delay line

It is easy to see that, applying \( W_d \) to a vector \( \mathbf{v} = (v_1, v_2, \ldots, v_n) \) results in a vector

\[
\mathbf{v}' := W_{d:1} W_{d:2} \ldots W_{d:n} \mathbf{v} = (0, v_1, \ldots, v_{n-1})
\]

and because of the associativity of the matrix product, we see that applying \( W_{d:1} \) to a vector \( k \) times results in permuting the vector \( k \) times and the substituting the first \( k \) elements with the same number of 0s. So, the controllability matrix for the delay line is:

\[
C_d = \begin{bmatrix} w_d & W_d w_d & \ldots & W_d^{n-1} w_d \end{bmatrix}
\]

(48)

which would be a lower diagonal matrix for a generic \( \mathbf{v} \) but for \( w_d = (1, 0, \ldots, 0) \) is just the identity.

Now, consider the fact that

\[
W_d^n = 0
\]

(49)
The CH theorem implies that any higher power will be null as well. So we simply have:

\[ s_0 = u_0 \]
\[ s_1 = u_1 \]

and so on, because all the \( \phi_j^{(m)} \) for \( m > n \) are null. If we define \( s_d := (u_0, u_{-1}, u_{-2}, \ldots, u_{-(n-1)}) \):

\[ y_0 = r \cdot c_d \cdot s_d = r \cdot I \cdot s_d = \sum_{i=0}^{n-1} r_i u_{-i} \quad (50) \]

which, as expected, is simply a regressive model of order \( n \).

### D Cyclic reservoirs

The characteristic polynomial of \( W_c \) is \( \lambda^n = 1 \) so that the CH Theorem implies:

\[ W_c^n = I \quad (51) \]

Meaning that, for all \( m > n \),

\[ W_c^m = \sum_{j=0}^{n-1} \phi_j^{(m)} W_c^j = W_c^\mu \quad (52) \]

where \( \mu := m \mod n \). Note that, in general:

\[ (aW_c)^m = a^m W_c^\mu \quad (53) \]

So, if in our reservoir we fix \( W = \rho W_c \) (where \( \rho \) is a parameter controlling the spectral radius) we obtain a number of simplifications. First of all, the elements of \( s \) assume a regular form. For example:

\[ s_0 = u_0 + \rho^n u_{-n} + \rho^{2n} u_{-2n} + \ldots \]
\[ s_1 = u_{-1} + \rho^n u_{-(n+1)} + \rho^{2n} u_{-(2n+1)} + \ldots \]

so that their general form is

\[ s_j = \sum_{k=0}^{\infty} \phi_j^{(k)} u_{-k} = \sum_{p=0}^{\infty} \rho^{pn} u_{-j+pn} \quad (54) \]

Moreover, the controllability matrix \( C_c \) assumes a simple form. If we define the \( i \)-time permuted input weight vector as:

\[ w^{(i)} := W_c^i w \quad (55) \]

we obtain:

\[ C_c = [w \quad \rho w^{(1)} \quad \rho^2 w^{(2)} \ldots \quad \rho^{n-1} w^{(n-1)}] \quad (56) \]

so that:

\[ y_0 = (r_0, r_1, \ldots, r_{n-1}) C_c \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix} \quad (57) \]

The output can be written in compact form by defining:

\[ \tilde{s}_j = \sum_{p=0}^{\infty} \rho^{j+pn} u_{-j+pn} \quad (58) \]

\[ \tilde{C}_c = [w \quad w^{(1)} \quad w^{(2)} \ldots w^{(n-1)}] \quad (59) \]

so that, finally:

\[ y_0 = r \tilde{C}_c \tilde{s} \quad (60) \]
Acknowledgment

LL gratefully acknowledges partial support of the Canada Research Chairs program.

References


