

Memory in linear recurrent neural networks in continuous time

Michiel Hermans*, Benjamin Schrauwen

Department of Electronics and Information Systems, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium

Abstract

Reservoir Computing is a novel technique which employs recurrent neural networks while circumventing difficult training algorithms. A very recent trend in Reservoir Computing is the use of real physical dynamical systems as implementation platforms, rather than the customary digital emulations. Physical systems operate in continuous time, creating a fundamental difference with the classic discrete time definitions of Reservoir Computing. The specific goal of this paper is to study the memory properties of such systems, where we will limit ourselves to linear dynamics. We develop an analytical model which allows the calculation of the memory function for continuous time linear dynamical systems, which can be considered as networks of linear leaky integrator neurons. We then use this model to research memory properties for different types of reservoirs. We start with random connection matrices with a shifted eigenvalue spectrum, which perform very poorly. Next, we transform two specific reservoir types, which are known to give good performance in discrete time, to the continuous time domain. Reservoirs based on uniform spreading of connection matrix eigenvalues on the unit disk in discrete time give much better memory properties than reservoirs with random connection matrices, where reservoirs based on orthogonal connection matrices in discrete time are very robust against noise and their memory properties can be tuned. The overall results found in this work yield important insights in how to design networks for continuous time.

Key words:

Reservoir Computing, Continuous time, Memory function, Linear dynamics, Recurrent neural networks

1. Introduction

A significant body of research on neural networks focuses on recurrent neural networks. These networks have for instance been studied for their ability to store patterns (the so-called Hopfield networks (Hopfield, 1982)). More recently however, recurrent networks are being used to process temporal information; due to internal feedback, these networks have an intrinsic ability to retain information about past input for a certain time, which allows for the processing of signals which are explicitly coded in time. This property is often called ‘fading memory’, ‘short term memory’ or the ‘echo state property’. Though training algorithms for recurrent neural networks exist which can be quite successful (most notably *backpropagation through time* (Rumelhart et al., 1986)), some problems like slow convergence and limited applicability due to high computational costs remain (Hammer and Steil, 2002).

An alternative approach is generally known as *Reservoir Computing* (RC), discovered independently by Jaeger (Jaeger, 2001a) for analog hyperbolic tangent neurons, and by Maass (Maass et al., 2002) for spiking neurons, where the networks are called ‘Echo State Networks’ and ‘Liquid State Machines’ respectively. In these systems, the

“reservoir” is a randomly initiated recurrent neural network with fixed interconnection weights, excited by the input which needs to be processed. The readout mechanism consist of a single layer of linear nodes observing the reservoir nodes, which are usually trained using linear regression. This approach is very fast and easy to implement, and does not suffer from common problems in other training algorithms such as fading error gradients or slow convergence (check Prokhorov (2005) for a broader context of RC in schemes of training recurrent networks). Despite its recent introduction, RC has already proven to be quite powerful: many applications for RC have already been successfully implemented. For instance speech recognition (Verstraeten et al., 2006), robotics applications (Antonelo et al., 2008), and for certain tasks (predicting chaotic time series) it outperforms all other state-of-the-art techniques (Jaeger and Haas, 2004).

Due to its generic nature, RC is not limited to digital simulations of neural networks. Any high dimensional complex dynamical system with the right properties¹ can

¹These properties are still the matter of some debate, though it is generally accepted that good reservoirs should have input separability and fading memory (Maass et al., 2005; Maass and Markram, 2004) and its dynamics should be close to the regime shift from stable to chaotic (Bertschinger and Natschläger, 2004; Legenstein and Maass, 2007)

*Corresponding author. Tel.: +32 9 264 95 28.

Email address: Michiel.Hermans@ugent.be (Michiel Hermans)

serve as a reservoir. For instance the use of photonic components (Vandoorne et al., 2008) and the use of a liquid water surface (Fernando and Sojakka, 2003). As of yet unpublished research at TU Graz has been focusing on the computational abilities of nonlinear spring-mass-damper systems, with possible applications in robotics. It has even been suggested that certain bacterial organisms use their gene regulation network to perform RC for adapting to changing environmental conditions (Jones et al., 2007).

An important trend in RC is to use leaky integrators, which employ a low-pass filtering operation inside the neurons to increase the reservoir’s temporal processing abilities by increasing its memory (Jaeger et al., 2007), which has also been extended to band-pass filters (Schrauwen et al., 2007; wyffels et al., 2008). One of the most important empirical results in this field is that usually the timescale of the reservoir (which is determined by the pass-band of the filtering operation) has to be matched to the timescale of the signal in order to maximize performance (see for instance Schrauwen et al. (2007)). In spiking leaky integrate-and-fire neurons as well, low-pass filtering is performed when using exponential synapse models (Gerstner and Kistler, 2002). Here as well, the timescale of the synapses can be tuned to optimize performance on certain tasks (Hermans et al., 2008).

Compared to ‘classic’ RC, which is implemented in discrete time, all of the above examples have a continuous time element: most are physical systems which inherently operate in continuous time. When applying low-pass filtering in discrete time reservoirs, the equations which describe the evolution of the reservoir states can in fact be considered as a discrete time approximation of continuous time differential equations (Schrauwen et al., 2007).

As stated earlier, one of the most important properties a reservoir needs is fading memory. This paper will research this property for continuous time linear systems. Although RC in fact relies on nonlinearity for its computational power, a linear model serves as a first-order approximation which can still be approached with analytical methods and can give valuable insights in the dynamics of the system.

Little research on continuous time RC has yet been performed. A recent paper (Ganguli et al., 2008) studies short term memory in discrete time neural networks by using a criterion based on Fisher information, and it briefly mentions an extension of its theoretical frame to continuous time reservoirs. However, the method by which they evaluate memory does not depend on the nature of the signal which needs to be remembered, which - as we will show - is quite important in the criterion we use in this paper.

In order to study the temporal processing ability of linear first order systems, one needs to know how much information on past input is coded in the immediate spatial state of the system. This property is represented by the *memory function* (MF) (Jaeger, 2001b), which measures the ability to reconstruct the input signal from a past time

τ from the immediate state of the system. The classical definition of the MF is usually defined for discrete time step reservoirs, where the input consist of a signal $s(i)$, $i \in \mathbb{N}$ that takes on a random value from a normal distribution at each time step. The MF is then defined as

$$m(k) = \frac{\langle s(i-k)u_k(i) \rangle_i^2}{\sigma^2(s)\sigma^2(u_k)}, \quad (1)$$

in which $\langle \dots \rangle_i$ is the mean over all values i , and $u_k(i) = \mathbf{y}(k)^\top \mathbf{a}(i)$, the output trained to reconstruct the signal from k timesteps ago, with $\mathbf{y}(k)$ optimized output weights and $\mathbf{a}(i)$ the reservoir state at the i -th time step, and σ denotes standard deviation. The downside of this approach is that the signal used here is not necessarily a realistic signal that has to be processed. If we for instance define a continuous time equivalent for this type of signal by assuming that the time steps become infinitesimally small, this will become a signal which fluctuates infinitely fast and has an unbounded spectrum. This type of signal does not exist in continuous time because it would require infinite power. Throughout this paper we shall therefore assume that the continuous-time signal $s(t)$ is of a more realistic form which has a bounded spectrum and amplitude.

This paper is structured as follows. First, we propose a derivation for the MF which takes into account the nature of the input signal and the low pass filtering of the reservoir. First we will derive an analytical expression which allows for a fast and accurate computation of the MF. Next, we shall use this expression to semi-empirically analyze the memory of linear leaky-integrator reservoirs in continuous time for different parameters such as eigenvalue distribution, state noise and the reservoir timescale. The only limitation of our model is the condition that the connection matrix should be diagonalizable, which is true for almost all common reservoir setups².

2. Analytical derivation of the memory function

2.1. Reservoir states

In this part, we shall derive the MF for any diagonalizable connection matrix \mathbf{W} of size $N \times N$, where N is the number of neurons. To start our discussion, we will first consider the case where no noise is present in the system. The reservoir state $\mathbf{a}(t)$ then evolves according to

$$\dot{\mathbf{a}}(t) = \mathbf{W}\mathbf{a}(t) + \mathbf{v}s(t), \quad (2)$$

in which \mathbf{v} is a vector which describes the weights of input connections. The general solution to equation (2) in steady

²A notable exception is the *delay line*, which consists of a chain of neurons where input is fed only to the first neuron, and the last one does not give output to any other neurons. The delay line has been studied as an important academic example in White et al. (2004) and Ganguli et al. (2008)

state regime (i.e., when the influence of initial conditions has faded) is given by

$$\mathbf{a}(t) = [\mathbf{u}(t)e^{\mathbf{W}t}\mathbf{v}] * s(t), \quad (3)$$

where $\mathbf{u}(t)$ is the Heaviside-step function. When we now assume that \mathbf{W} is diagonalizable, hence that $\mathbf{W} = \mathbf{C}\mathbf{D}\mathbf{C}^{-1}$ with \mathbf{D} a diagonal matrix with eigenvalues λ_i . Then we can rewrite the above equation elementwise as

$$\begin{aligned} a_i(t) &= \left[\mathbf{u}(t)\mathbf{C}e^{\mathbf{D}t}\underbrace{\mathbf{C}^{-1}\mathbf{v}}_{\mathbf{p}} \right]_i * s(t) \\ &= \mathbf{u}(t) \sum_{j=1}^N C_{ij}p_j \exp(\lambda_j t) * s(t) \\ &= [\mathbf{Tz}(t)]_i, \end{aligned}$$

where $T_{ij} = C_{ij}p_j$ and $z_i(t) = \mathbf{u}(t)\exp(\lambda_i t) * s(t)$. This means that any linear combination of reservoir states can be written as a linear combination of the output of filters with impulse response $\exp(\lambda_i t)$ operating on the input signal. These are all well known basic results from linear dynamical system theory as found in for instance Sontag (1998).

2.2. Modeling noise

Noise can appear in many different forms: one can inject noise into the neurons as extra input, noise can be superposed on the input signal, or the neurons can be inherently noisy, superposing noise on their output etc. In this article, we limit ourselves to the last option, which means we superpose a term $\sigma(\mathbf{a}')\sqrt{\epsilon}\mathbf{h}(t)$ on the reservoir states, where $\sigma(\mathbf{a}')$ is the mean over all neurons of the standard deviation of the states when no noise is present, ϵ is the signal-to-noise ratio (SNR), and $\mathbf{h}(t)$ is the noise signal with unit standard deviation and zero mean. Scaling the noise to the mean standard deviation of the reservoir states has the advantage that one does not need to account for internal amplification of the input signal.

If we further assume that this noise has a bandwidth which is far greater than the pass-bands of the neurons, it is easy to see that noise will propagate through the network only to a limited degree. Low-pass filtering of a signal comes down to taking its average value over an exponential window. If the noise fluctuates very fast compared to the time scale of the low-pass filtering operation, the output of this filtering will be very small in amplitude. Each neuron acts as a low pass filter or an integrator of some sort and hence the states of the neurons are assumed to filter out the noise on their input. We will therefore assume that no noise propagation exists in the network, which leads to the following reservoir states

$$\mathbf{a}(t) = \mathbf{Tz}(t) + \sigma(\mathbf{a}')\sqrt{\epsilon}\mathbf{h}(t). \quad (4)$$

Note that in any realistic scenario, noise superposed on the reservoir states will propagate through the network

to some degree, which means this model will only be applicable in some situations. However, this approximation allows us to form a direct link between eigenvalues of the correlation matrix of the reservoir states, and noise sensitivity (see 3.1).

Extending the analytical model used in this paper to fully account for noise propagation is in fact not very difficult, but since this introduces extra parameters (particularly noise spectral range) we choose this simpler model. There is another advantage to this model: if the reservoir is a physical system, reading out the reservoir state will have to be done by some sort of measurement, which is usually inherently noisy. Assuming non-propagating noise obviously also serves to model limitations on readout precision. When noise on the readout mechanism is much more intense than the noise propagating through the network, the above approximation will be valid a fortiori.

It should be noted that adding state noise which does not propagate is mathematically identical to ridge regression, a special case of Tikhonov regularisation (Tikhonov and Arsenin, 1977), a technique to regularize readout weights, which is commonly performed when training linear regressors to avoid overfitting (Wyffels et al., 2008).

2.3. Memory function

Optimal readout weights for reconstructing the signal at a time τ in the past can be found by solving the system $\mathbf{A}\mathbf{x} = \mathbf{y}$ (see for instance Jaeger (2001b)), where $\mathbf{x}(\tau) = \langle \mathbf{a}(t)s(t-\tau) \rangle_t$ and $\mathbf{A} = \langle \mathbf{a}(t)\mathbf{a}^T(t) \rangle_t$. This yields the following expression for the memory function

$$m(\tau) = \frac{\mathbf{x}^T(\tau)\mathbf{A}^{-1}\mathbf{x}(\tau)}{\sigma^2(s)}. \quad (5)$$

For simplicity, we assume $\sigma^2(s) = 1$ and a zero mean, although these are in no way necessary conditions. For continuous time, we define the operator $\langle \rangle_t$ as

$$\langle f(t) \rangle_t = \lim_{P \rightarrow \infty} \frac{1}{2P} \int_{-P}^P f(t) dt,$$

i.e the mean over time; we assume no time origin for the signal and reservoir states. Furthermore we assume that the signal is a stationary stochastic process. Using the above definitions, we can now write $\mathbf{x}(\tau)$ and \mathbf{A} in a more explicit form. We start with $\mathbf{x}(\tau)$.

$$\begin{aligned} [\mathbf{x}(\tau)]_i &= [\langle \mathbf{Tz}(t)s(t-\tau) \rangle_t + \sigma(\mathbf{a}')\sqrt{\epsilon} \langle \mathbf{h}(t)s(t-\tau) \rangle_t]_i \\ &= \sum_{j=0}^N T_{ij} \left\langle \int_0^\infty dt' s(t-t')e^{\lambda_j t'} s(t-\tau) \right\rangle_t \\ &= \sum_{j=0}^N T_{ij} \int_0^\infty dt' \exp(\lambda_j t') \underbrace{\langle s(t-t')s(t-\tau) \rangle_t}_{R(t'-\tau)} \\ &= \sum_{j=0}^N T_{ij} \underbrace{\int_0^\infty dt' \exp(\lambda_j t') R(t'-\tau)}_{b_j(\tau)}, \end{aligned}$$

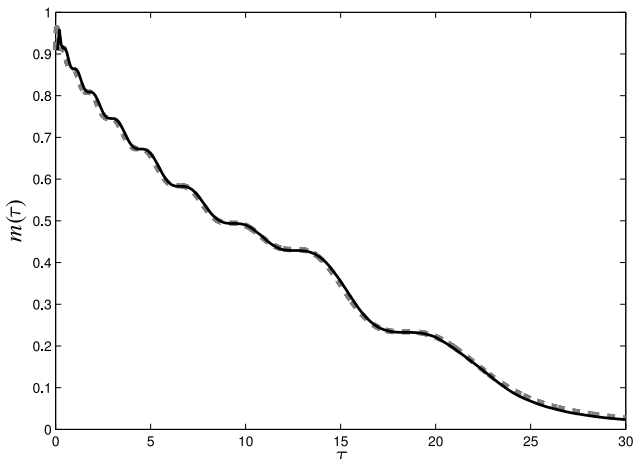


Figure 1: Comparison of the empirically determined MF versus the result obtained in equation (7). Grey dashed line is analytical prediction, black solid line empirical measurement. Setup of the experiment is described in the text.

where $R(t)$ is the autocorrelation function of the input signal. Writing this in matrix notation gives.

$$\mathbf{x}(\tau) = \mathbf{T}\mathbf{b}(\tau).$$

Using the same reasoning, we can calculate the correlation matrix \mathbf{A} . Notice that, since $\mathbf{a}(t) = \mathbf{T}\mathbf{z}(t) + \sqrt{\epsilon}\mathbf{h}(t)$, with $\mathbf{a}(t)$ real, and \mathbf{T} and $\mathbf{z}(t)$ generally complex, we can write $\mathbf{a}^\top(t) = \mathbf{a}^\dagger(t) = \mathbf{z}^\dagger(t)\mathbf{T}^\dagger + \sigma(\mathbf{a}')\sqrt{\epsilon}\mathbf{h}^\dagger(t)$, where \dagger stands for the Hermitian transpose. This allows us to calculate a Hermitian form for \mathbf{A} :

$$\begin{aligned} \mathbf{A} &= \mathbf{T} \underbrace{\langle \mathbf{z}(t)\mathbf{z}^\dagger(t) \rangle}_{\mathbf{B}} \mathbf{T}^\dagger + \epsilon\sigma^2(\mathbf{a}') \langle \mathbf{h}(t)\mathbf{h}^\dagger(t) \rangle \\ &= \mathbf{T}\mathbf{B}\mathbf{T}^\dagger + \epsilon\sigma^2(\mathbf{a}')\mathbf{I}. \end{aligned}$$

Here, \mathbf{B} is the correlation matrix for the responses of the filters $\exp(\lambda_i t)$, which can now be calculated the same way as the elements of $\mathbf{b}(\tau)$:

$$B_{ij} = \int_0^\infty dt \int_0^\infty dt' R(t-t') \exp(\lambda_i t) \exp(\lambda_j^* t').$$

The variances of the states of the individual neurons without noise are given by the diagonal elements of the correlation matrix, which yields $\sigma^2(\mathbf{a}') = N^{-1}\text{tr}(\mathbf{T}\mathbf{B}\mathbf{T}^\dagger)$. This number is also equal to the mean eigenvalue of the noiseless correlation matrix. When we denote these as γ_i , we can write $N^{-1}\text{tr}(\mathbf{T}\mathbf{B}\mathbf{T}^\dagger)$ as $\bar{\gamma}$ for short. Combining these equations finally leads to a useful expression for the MF:

$$m(\tau) = \mathbf{b}^\dagger(\tau)\mathbf{T}^\dagger (\mathbf{T}\mathbf{B}\mathbf{T}^\dagger + \bar{\gamma}\epsilon\mathbf{I})^{-1} \mathbf{T}\mathbf{b}(\tau). \quad (6)$$

This expression allows for a quick numerical evaluation of the MF for linear dynamical systems. When $\epsilon = 0$, the MF reduces to

$$m(\tau) = \mathbf{b}^\dagger(\tau)\mathbf{B}^{-1}\mathbf{b}(\tau), \quad (7)$$

which means that without noise, the MF depends solely on the eigenvalues of \mathbf{W} and not on the input vector or connection topology, and it remains invariant under a similarity transformation of \mathbf{W} . We can for instance replace \mathbf{W} with its diagonal form \mathbf{D} which reduces the network to a set of decoupled complex filters. If real elements are required, each complex pair of eigenvalues can be replaced by a 2×2 block on the diagonal of \mathbf{W} , resulting in a damped oscillator. Real eigenvalues simply represent disconnected low-pass filters.

To perform empirical studies on the MF, we shall assume an input autocorrelation function of the form

$$R(t) = \exp(-\alpha|t|),$$

which describes a signal which is limited in bandwidth by the finite autocorrelation length α , and where we define the signal timescale as α^{-1} . This serves as an analogy to the signal used in discrete networks and is quite common for many natural stochastic processes. Using this function, we can calculate the elements of \mathbf{B} and $\mathbf{b}(\tau)$:

$$B_{ij} = \frac{1}{(\alpha - \lambda_i)(\alpha - \lambda_j^*)} \left(1 - \frac{2\alpha}{\lambda_i + \lambda_j^*} \right) \quad (8)$$

$$b_i(\tau) = \frac{(\lambda_i - \alpha)e^{-\alpha\tau} + 2\alpha e^{\tau\lambda_i}}{(\alpha^2 - \lambda_i^2)}. \quad (9)$$

To validate these expressions, we compared their result with the empirically determined MF of a 10-neuron toy network. To approximate a continuous-time neural network in the simulation, we used discrete time steps of 1 ms in a reservoir with a timescale $\tau_R = 1$ s. Generating a signal which has $R(t) = \exp(-\alpha|t|)$ was done by creating a signal which was sampled from a Gaussian distribution each time step, and then low pass-filter this with timescale α^{-1} . The resulting signal can easily be proved to approximately have the desired autocorrelation function. As parameters we chose, $\alpha = 1$ Hz, $\zeta = 0.9$ (where ζ is the spectral radius of the matrix used to construct \mathbf{W} , as fully explained in logarithmic A.2), and the MF was numerically evaluated using formula (5). Simulation was performed for a duration of 2×10^4 s (2×10^7 time steps). No noise was imposed on the signal; the only limitation on accuracy was the finite simulation time step and numerical precision. The result is depicted in Figure 1, which shows a good correspondence between theory and experiment.

For practical purposes, we use the Moore-Penrose pseudoinverse for calculating the inverse of the correlation matrix in equation (6), as is common for practical purpose linear regression problems. In some cases (see Section 3.2), this matrix will be very ill-conditioned, and inverting it is bound to be quite inaccurate. In fact, inverting the correlation matrix is required to calculate optimal output weights for signal reconstruction, where one has to solve the system $\mathbf{A}\mathbf{x} = \mathbf{y}$ as stated before. The pseudoinverse has the advantage that it finds an optimal solution for this problem, even when limited by numerical precision.

As such, solutions given by the pseudoinverse give results which represent not necessarily exact solutions, but solutions which reflect practical numerical limitations on the MF.

2.4. Further definitions

2.4.1. Reservoir timescale

In this paragraph, we will define a few useful parameters to qualify reservoirs and their memory. First of all we introduce a parameter to describe the intrinsic timescale of a reservoir, which we shall call the *reservoir timescale* τ_R and define as

$$\tau_R = - \left(\frac{\text{tr}(\mathbf{W})}{N} \right)^{-1} = - \left(\frac{1}{N} \sum_{i=1}^N \lambda_i \right)^{-1}. \quad (10)$$

This definition is based on a reservoir in which all neurons act as low-pass filters with the same timescale τ_R , where all diagonal elements of \mathbf{W} are equal to $-\tau_R^{-1}$. Note that this does not imply that, for a network with a certain τ_R , all neurons have to act as low pass filters with the same timescale, which would imply all diagonal elements of \mathbf{W} would have to be equal.

It is easy to see that the shape of the MF depends on the ratio between the signal timescale and the reservoir timescale. We can define normalized eigenvalues $\kappa_i = \tau_R \lambda_i$, and a normalized time $\tau_\alpha = \alpha \tau$. We can then rewrite equations (8) and (9) as

$$B_{ij} = \frac{\tau_R^2}{(\alpha\tau_R - \kappa_i)(\alpha\tau_R - \kappa_j^*)} \left(1 - \frac{2\alpha\tau_R}{\kappa_i + \kappa_j^*} \right),$$

$$b_i(\tau_\alpha) = \tau_R \frac{(\kappa_i - \alpha\tau_R)e^{-\tau_\alpha} + 2\alpha\tau_R \exp\left(\tau_\alpha \frac{\kappa_i}{\alpha\tau_R}\right)}{\alpha^2\tau_R^2 - \kappa_i^2}.$$

Since $\bar{\gamma}$ scales with $\text{tr}(\mathbf{TB}\mathbf{T}^\dagger)$, the noise term can also be rewritten the same way as \mathbf{B} . The factors τ_R and τ_R^2 which emerge after the transformation are eliminated when calculating the actual MF, and as such one can see that $m(\tau_\alpha)$ solely depends on the normalized eigenvalue spectrum κ_i , and the ratio between reservoir and signal timescale $\alpha\tau_R$. For this reason we shall throughout the remainder of this paper assume $\alpha = 1$, as if working with time τ_α . The results can then easily be extended to any desirable timescale.

2.4.2. Memory capacity and quality

In discrete time reservoirs one defines the *memory capacity* as the sum over the MF:

$$MC = \sum_{k=0}^{\infty} m(k),$$

which can be proven to be always smaller or equal to N (Jaeger, 2001b). For continuous time, the obvious equivalent is the integral over the MF.

$$\mu_c = \int_0^{\infty} m(\tau) d\tau. \quad (11)$$

Note that this quantity, for real physical systems, would have the dimension of time. If the MF would be equal to one up to a certain time θ in the past and zero everywhere else, memory capacity would be equal to θ . In Section 2.5 we shall suggest an upper bound for memory capacity in continuous time which is very similar to the bound N . We also introduce a useful quantity which we denote as the *memory quality*, defined as

$$\mu_q(x) = \frac{1}{x} \int_0^x m(\tau) d\tau. \quad (12)$$

This measure is always smaller or equal to 1 and is a number which denotes the average MF up to a time x in the past which can be chosen at a value relevant for a certain task or type of reservoir. In this article we will mostly take $x = \mu_c$. In this case, the memory quality expresses the relative amount of memory which is actually present in a range equal to the memory capacity. In Section 3.4, we will study a special kind of reservoir which has a MF which abruptly drops to a much lower value at a certain time T_R in the past. For these reservoirs we shall therefore choose $x = T_R$.

μ_c as well as $\mu_q(x)$ can be straightforwardly calculated from equation (6). To do this, one needs to calculate the integrals over the crossproducts of the elements $b_i(\tau)$. Resulting formulas become quite complex, particularly for $\mu_q(x)$ and hence we shall omit these here.

2.5. Asymptotic memory capacity

An interesting limiting case for the memory capacity is for when τ_R goes to infinity and no noise is present. First of all, we look more closely at the definition of the memory capacity:

$$\begin{aligned} \mu_c &= \int_0^{\infty} m(\tau) d\tau \\ &= \sum_{i=1}^N \sum_{j=1}^N \int_0^{\infty} b_i^*(\tau) b_j(\tau) [\mathbf{B}^{-1}]_{ij} d\tau \\ &= \text{tr} \left(\mathbf{B}^{-1} \int_0^{\infty} \mathbf{b}(\tau) \mathbf{b}^\dagger(\tau) d\tau \right). \end{aligned}$$

With this definition and equations (8) and (9) we can calculate the asymptotic limit. To do this, we again define normalized eigenvalues $\kappa_i = \tau_R \lambda_i$, and normalize time on the reservoir timescale: $\theta = \tau/\tau_R$. Transforming the integration variable from τ to θ gives

$$\int_0^{\infty} b_i(\tau) b_j^*(\tau) d\tau = \tau_R \int_0^{\infty} b_i(\theta) b_j^*(\theta) d\theta.$$

Expanding $b_i(\theta)$ and taking the limit $\tau_R \rightarrow \infty$ gives

$$\begin{aligned} \lim_{\tau_R \rightarrow \infty} b_i(\theta) &= \lim_{\tau_R \rightarrow \infty} \frac{\left(\left(\frac{\kappa_i}{\tau_R} - \alpha \right) e^{-\alpha\tau_R\theta} + 2\alpha e^{\theta\kappa_i} \right)}{\left(\alpha^2 - \left(\frac{\kappa_i}{\tau_R} \right)^2 \right)} \\ &= \frac{2e^{\theta\kappa_i}}{\alpha}. \end{aligned}$$

This yields

$$\lim_{\tau_R \rightarrow \infty} \tau_R \int_0^\infty b_i(\theta) b_j^*(\theta) d\theta = \lim_{\tau_R \rightarrow \infty} \frac{-4\tau_R}{\alpha^2(\kappa_i + \kappa_j^*)}.$$

We can similarly apply the limit to the elements of \mathbf{B} , where we find

$$\lim_{\tau_R \rightarrow \infty} B_{ij} = \lim_{\tau_R \rightarrow \infty} \frac{-2\tau_R}{\alpha^2(\kappa_i + \kappa_j^*)}.$$

Finally, we can write the the memory capacity as

$$\lim_{\tau_R \rightarrow \infty} \mu_c = \frac{2}{\alpha} \text{tr}(\mathbf{I}) = \frac{2}{\alpha} N. \quad (13)$$

When $\tau_R \rightarrow \infty$, the MF will stretch on to infinity and consequently it will be infinitesimally close to zero, so at first sight the above calculation might not seem very useful. However, there are strong suggestions that this limit might in fact be an upper bound for the memory capacity of linear first order networks. It seems the memory capacity always rises monotonically with τ_R (see following sections), and reaches an asymptotic upper limit for very large τ_R . Furthermore, in Section 3.4, when researching a special kind of reservoirs where we can find approximate solutions for the memory capacity, we can also confirm that this number is the maximal value for memory capacity. So far, we have not been able to find mathematical proof that this in fact a true upper bound, and for now we will leave this as a conjecture to be proven or disproven in future research. Notice that this expression is, just like in discrete time networks, proportional to N and links memory capacity to signal statistics, suggesting that each neuron is capable to store a maximal amount of ‘‘information’’ equal to $2/\alpha$, just like in discrete time each single neuron is capable to store one time step of the input signal.

In the next sections, we shall investigate a few important reservoir types, using results which have been acquired for discrete time networks which we translate to the continuous time domain.

3. Memory and noise sensitivity

3.1. Noise and the correlation matrix

Before moving to empirical testing of different reservoir types, we derive a general expression which connects the noise sensitivity of the MF to a basis of orthogonal reservoir states and the eigenvalues of the correlation matrix \mathbf{A} . The operations performed in this section are highly similar to those in Principal Component Analysis (Pearson, 1901; Jolliffe, 2002), where we apply it on continuous time functions instead of the more common discrete data points.

$\mathbf{A} = \mathbf{TBT}^\dagger$ is a real symmetric positive-definite matrix³ and has an eigendecomposition such that $\mathbf{TBT}^\dagger =$

$\mathbf{U}\mathbf{\Gamma}\mathbf{U}^\dagger$, where \mathbf{U} is orthogonal, and $\mathbf{\Gamma}$ a diagonal matrix with only positive real eigenvalues γ_i . Notice that, since $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$, we can write

$$\begin{aligned} (\mathbf{TBT}^\dagger + \bar{\gamma}\epsilon\mathbf{I})^{-1} &= (\mathbf{U}\mathbf{\Gamma}\mathbf{U}^\dagger + \bar{\gamma}\epsilon\mathbf{U}\mathbf{U}^\dagger)^{-1} \\ &= \mathbf{U}^\dagger(\mathbf{\Gamma} + \bar{\gamma}\epsilon\mathbf{I})^{-1}\mathbf{U} \\ &= \mathbf{U}(\mathbf{\Gamma} + \bar{\gamma}\epsilon\mathbf{I})^{-1}\mathbf{U}^\dagger. \end{aligned}$$

This allows us to write the MF as follows

$$\begin{aligned} m(\tau) &= \underbrace{\mathbf{b}^\dagger(\tau)\mathbf{T}^\dagger\mathbf{U}}_{\boldsymbol{\psi}^\dagger(\tau)} (\mathbf{\Gamma} + \bar{\gamma}\epsilon\mathbf{I})^{-1} \underbrace{\mathbf{U}^\dagger\mathbf{T}\mathbf{b}(\tau)}_{\boldsymbol{\psi}(\tau)} \\ &= \sum_{i=1}^N \frac{\psi_i^2(\tau)}{\gamma_i + \bar{\gamma}\epsilon}. \end{aligned}$$

Notice that $\boldsymbol{\psi}(\tau)$ is strictly real. Defining $\beta_i(\tau) = \psi_i^2(\tau)\gamma_i^{-1}$, we can write this as

$$m(\tau) = \sum_{i=1}^N \frac{\beta_i(\tau)}{1 + \bar{\gamma}\gamma_i^{-1}\epsilon}. \quad (14)$$

This means that we can decompose the MF as a set of functions which are all real, positive and between zero and one. The order of the terms in this equation follows the order of the eigenvalues γ_i , with $i = 1$ corresponding to the largest of γ_i and as such in a declining order.

There is a clear interpretation of the functions $\beta_i(\tau)$. Suppose we wish to linearly transform the reservoir states $\mathbf{a}(t)$ in the absence of noise, so as to define an base of states $\hat{\mathbf{a}}(t)$ which have the property that $\langle \hat{\mathbf{a}}(t)\hat{\mathbf{a}}^\dagger(t) \rangle = \mathbf{I}$, i.e. a set of orthogonal (uncorrelated) states with unit standard deviation. The above procedure does in fact perform this transformation. When we implicitly define $\hat{\mathbf{a}}(t)$ as

$$\mathbf{a}(t) = \mathbf{U}\mathbf{\Gamma}^{1/2}\hat{\mathbf{a}}(t), \quad (15)$$

we can easily see that this yields the desired expression for $\langle \mathbf{a}(t)\mathbf{a}^\dagger(t) \rangle$. So we can define the orthogonal base states from this expression:

$$\hat{\mathbf{a}}(t) = \mathbf{\Gamma}^{-1/2}\mathbf{U}^\dagger\mathbf{a}(t), \quad (16)$$

which yields the desired correlation matrix. Looking at the original definition of the MF, one can then easily define the MF in terms of the base states $\hat{\mathbf{a}}(t)$, and in the absence of noise:

$$m(\tau) = \sum_{i=1}^N \langle s(t-\tau)\hat{a}_i(t) \rangle^2, \quad (17)$$

so that $\beta_i(\tau) = \langle s(t-\tau)\hat{a}_i(t) \rangle^2$.

Each of the terms in (14) has a clear dependence on its corresponding eigenvalue γ_i and its noise sensitivity. We can make a very rough estimate of the MF for a certain noise intensity by approximating $\epsilon\bar{\gamma}/\gamma_i$ as zero when $\epsilon < \gamma_i/\bar{\gamma}$, and $\epsilon\bar{\gamma}/\gamma_i = \infty$ when $\epsilon > \gamma_i/\bar{\gamma}$. This means that we only add up terms in (14) up to $i = k$ where $\epsilon < \gamma_k/\bar{\gamma}$ and $\epsilon > \gamma_{k+1}/\bar{\gamma}$. This estimation is inaccurate but gives a

³For random matrices, discussed in the next paragraph, some negative eigenvalues can be found, but these are quite probably due to errors originating from limited numerical precision.

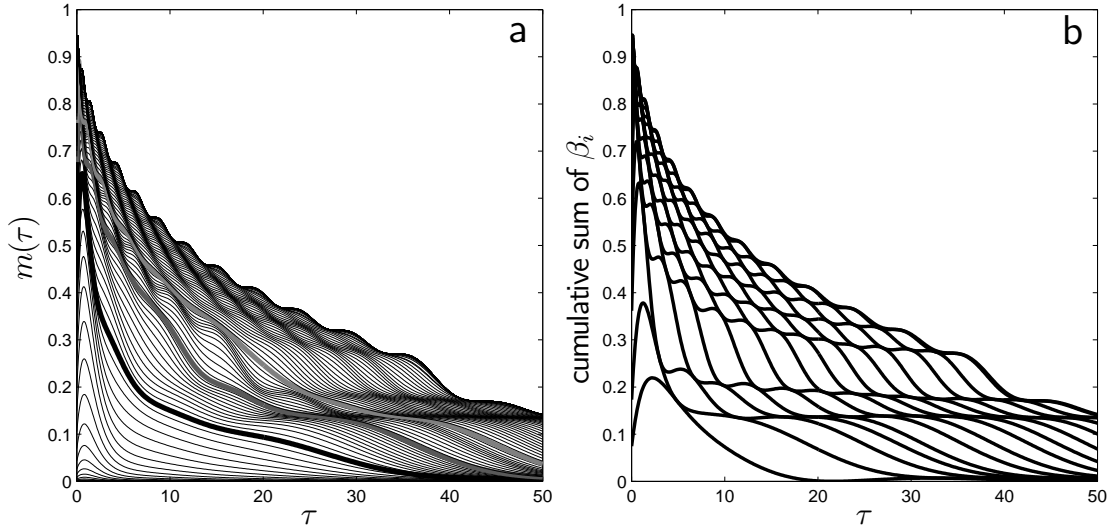


Figure 2: (a) Example of the MF for different values of ϵ of a 20-neuron network constructed as described in 3.2 with timescale $\tau_R = 2$, and $\zeta = 0.9$ (which is a spectral radius, see logarithmic A.2). The thick black line is for $\epsilon = 1$, the dark grey line for $\epsilon = 10^{-3}$ and the light grey line for $\epsilon = 10^{-6}$. Notice the strong sensitivity for noise: even for a signal-to-noise ratio of the order 10^{-6} , the MF is still not close to its asymptotic convergence to its ideal for $\epsilon = 0$ (highest line). (b) Cumulative sum of β -functions. The k -th line from the bottom up is the sum of the β -functions up to k . The last 4 β -functions were not added due to the fact that the smallest of the eigenvalues γ_i cannot be calculated accurately, resulting in irregular behavior

graphical interpretation of the β -functions in (14).

Figure 2 gives a depiction of the MF for different noise values on the right, and a cumulative sum of β -functions on the left. Notice the increasing number of oscillations on the β -functions, which gives rise to the final shape of the MF when no noise is present.

There is a very clear interpretation for this type of noise sensitivity. When writing equation (15) elementwise, one can see that the base states \hat{a}_j are each encoded in the reservoir state \mathbf{a} with a magnitude $\sqrt{\gamma_j}$:

$$a_i(t) = \sum_{j=1}^N U_{ij} \sqrt{\gamma_j} \hat{a}_j(t).$$

The smaller γ_j gets, the smaller the actual contribution of \hat{a}_j is to the reservoir states, and the more it will “drown” into the surrounding noise.

This result reflects facts which have been suggested by others as well. For instance, in Jaeger (2001b) it is mentioned that the memory capacity of neural networks is limited by the conditioning of the correlation matrix, where it was found that for discrete time reservoirs, the measured memory capacity always is slightly smaller than its theoretical value. Since the condition number of the correlation matrix is equal to the ratio between its highest and lowest singular values, which are equal to the eigenvalues because \mathbf{A} is a positive semi-definite matrix, this effect is directly linked to our result.

In Ozturk et al. (2006) a different approach is used where the goal is to maximize the entropy of the reservoir states in order to span the highest possible range of non-linear mappings of the input signal. It was found that -

in order to do this - the reservoir states should be as little correlated as possible. When the reservoir states are all uncorrelated, we essentially get the orthogonal base states $\hat{a}_i(t)$, where the eigenvalues of the correlation matrix are equal to the individual variances of the reservoir states, which means that they will not become excessively low.

3.2. Random reservoirs

In this section we will discuss the MF for reservoirs with connection matrices with random Gaussian elements that have unit standard deviation and zero mean (which we shall call ‘random network’ for short). In RC in discrete time, this is the most common way to build connection matrices, and here we will investigate a naive translation of this technique to the continuous time domain. For this, we will start from discrete time networks which employ a low-pass filtering operation, and convert this to a continuous time system by reducing the timestep to zero. The full process is described in Appendix A.2, and the resulting connection matrix is the initial random matrix with an eigenvalue spectrum shifted to the left half of the complex plane and scaled with τ_R^{-1} . The spectral radius η , as defined in Appendix A.2, was chosen at 0.9, but the MF does not depend critically on this value and it seems conclusions found here remain generally valid when $0.5 < \eta < 1$. When η becomes smaller than 0.5, the MF will quickly start to decrease in quality and capacity and will become more sensitive to noise.

We can use our analytical model to study the memory for these types of reservoirs. First, we shall define a criterion to optimize the MF of a given reservoir. For practical purposes, this will obviously depend on the task

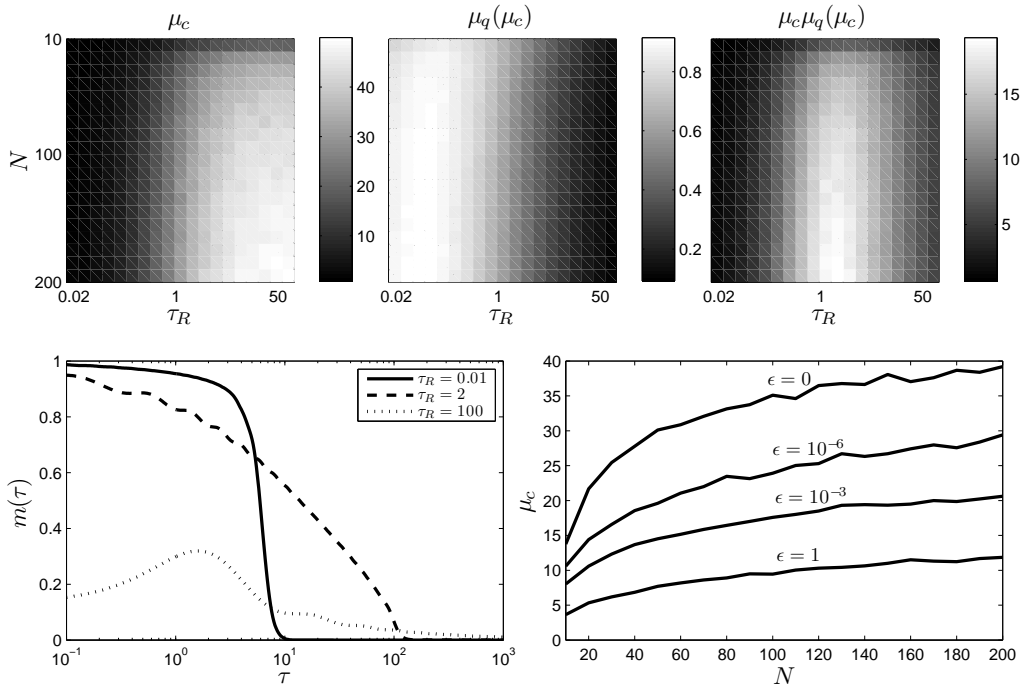


Figure 3: Top three panels: μ_c , $\mu_q(\mu_c)$ and $\mu_c\mu_q(\mu_c)$ in function of N and τ_R for random networks (τ_R shown on a logarithmic scale and $\epsilon = 0$). Bottom left panel: MF with respect to different values of τ_r . The argument τ is shown on a logarithmic scale, $N = 100$. Bottom right panel: memory capacity in function of N with respect to different noise levels. For this experiment, $\tau_R = 2$. All results in this figure have been found by averaging over 50 reservoir initializations.

which needs to be performed, but as a general criterion, we wish to find a balance between memory capacity and memory quality. As an overall objective function for optimum memory, we multiply the memory quality by the memory capacity. This number is equal to $\int_0^{\mu_c} m(\tau) d\tau$, and signifies how much memory is in fact present within the range $\{0 \dots \mu_c\}$.

We investigate the three measures μ_c , $\mu_q(\mu_c)$ and $\mu_c\mu_q(\mu_c)$ as functions of the reservoir timescale and neuron number N . Results are depicted in the top three panels of Figure 3. The highest memory quality is found at low τ_R (between 0.01 and 0.1), and memory capacity rises monotonically with τ_R . Optimal values for $\mu_c\mu_q(\mu_c)$ are found for $\tau_R \approx 2$ and this seems independent of the number of neurons. The shape of the MFs corresponding to high, low, and optimal τ_R are depicted in the bottom left panel of Figure 3. As one can expect, fast reservoirs will have very good memory, but only for a very short history. Slow reservoirs have generally very low MFs which extend very far. The optimal value for τ_R as found by our criterion tries to balance these two effects by producing a MF with a reasonable range and quality. Still, it is fairly low in a large part of its range, until it drops to about zero at $\tau = 100$.

The bottom right panel of Figure 3 shows the average memory capacity as a function of N for different noise levels and $\tau_R = 2$. Two things can be derived from this graph. First, it seems that μ_c does not grow linearly with N at

all, not even when $\epsilon = 0$. Secondly, as we have already seen in Figure 2, random reservoirs are very sensitive to noise. Both these effects are caused by the fact that the correlation matrices for these types of networks, are very ill-conditioned, with condition numbers which reach the order of 10^{18} for $N = 50$, and higher still for higher N . Most of the normalized eigenvalues $\gamma_i \bar{\gamma}^{-1}$ are extremely small and cause the high noise sensitivity. In fact, most numerical values of γ_i are so small compared to the highest eigenvalue, that for $N > 20$ it becomes virtually impossible to calculate accurate values, resulting in negative γ_i (which is definitely incorrect, since \mathbf{A} has to be positive semi-definite).

The relative number of numerically incalculable eigenvalues increases rapidly with N . This means that - theoretically - memory capacity may in fact grow linearly with N , but confirming this with our analytical model would require much higher numerical precision for inverting \mathbf{A} . Another illustration of this fact can be found when numerically trying to confirm equation 13. For this, we took $\tau_R = 10^5$ (much higher than α^{-1}) and calculated the according memory capacity with our analytical model for networks of 100 neurons. Over 50 trials, the average μ_c was equal to 48, approximately 4 times lower than the expected 200. Obviously, such high numerical precision requirements are undesirable and these results lead us to conclude that random reservoirs in continuous time are unsuitable for good memory storage.

3.3. Exponentially distributed eigenvalues

It has been suggested in Ozturk et al. (2006) that good performance for a wide variety of tasks for discrete time reservoirs with hyperbolic tangent nonlinearities, can be achieved when the eigenvalues of the connection matrix are spread uniformly over a disk on the complex plain with a spectral radius $\eta < 1$. The main motivation of this is the maximization of state entropy to span the highest possible range of nonlinear mappings of input data. Although we are not able to generalize our analytical model to nonlinear neurons, we can still investigate the memory of the linear continuous time equivalent of such reservoirs.

To find such an equivalent, we apply the inverse z -transform to the uniform distribution of eigenvalues for the discrete system. The entire process is explained in Appendix B.1. The end result of this transformation yields an exponential distribution of eigenvalues in a horizontal strip in the left half of the complex plain. The actual eigenvalues are selected using an algorithm (see Appendix A.3) which makes sure they are spread evenly over their distribution. This avoids clustering of eigenvalues, or areas where no eigenvalues are present. When using random number generators to generate eigenvalues, the resulting reservoirs will have memories which differ widely in quality: some very poor, some very good. Making sure eigenvalues were spread evenly over their distribution reduced this variance, and resulted in overall better memory properties. The actual construction of the connection matrices is explained in detail in Appendix A.1.

We performed the same tests as in Section 3.2 to qualify memory capacity, where again, $\alpha = 1$. Results are depicted in Figure 4. This time, the optimal reservoir timescale was found for $\tau_R = 6$ again virtually independent of N . Memory capacity again rises monotonically with τ_R . Also, the MF for optimal τ_R is of a higher overall quality than that found for random reservoirs, with a relatively high value over most of its range until a drop-off at $\tau = 100$. The bottom right panel shows memory capacity at the optimal reservoir timescale. Again, μ_c rises slower than linear with N , however, we find significantly better values than for random reservoirs. We can again test equation (13) using 100 neurons and $\tau_R = 10^5$, averaged over 50 reservoir initializations. This gives an average memory capacity of about 154, which is still below the theoretical value of 200, but the difference is not as dramatic as for random reservoirs.

Clearly, choosing exponentially distributed eigenvalues will give acceptable memory capacity and quality for most tasks. However, looking at the memory capacities at different noise levels, we can again see that they are quite sensitive to noise. Relative to the memory capacity when no noise is present, noise sensitivity is in fact similar to that of random reservoirs, however an absolute comparison shows that even when $\epsilon = 1$, memory capacity is comparable to that of noiseless random reservoirs. The eigenvalue spectrum of \mathbf{A} is indeed generally better than

that of random reservoirs: most eigenvalues can still be calculated accurately and remain in a reasonable range. However, with increasing N , most normalized eigenvalues slowly drop, and eventually, more and more become incalculable, explaining the slower-than-linear increase of μ_c in function of N .

3.4. Resonator reservoirs

The third type of reservoirs we will investigate, we will similarly derive from a result which has been achieved for discrete time. In White et al. (2004), it has been theorized that networks with orthogonal connection matrices have memory which is very robust under noisy conditions, and as such are optimal for memory storage. The eigenvalues of these matrices all lie on a circle on the complex plain. Again applying the z -transform yields eigenvalues which all lie on a vertical line, parallel to the imaginary axis (see Appendix B.2), which crosses the real axis at $-\tau_R^{-1}$. We choose the imaginary parts of the eigenvalues equidistantly, with a difference in angular frequency ω between two successive eigenvalues. We now redefine the index i for the eigenvalues going from $-(N-1)/2$ to $(N-1)/2$ rather than from 1 to N . This gives (see Appendix B.2)

$$\lambda_i = j\omega i - \frac{1}{\tau_R}, \quad (18)$$

where we use j as the symbol for the imaginary unit to avoid confusion with indices i or j .

When no noise is present, we can replace the reservoir by a set of disconnected filters characterized by the eigenvalues. The reservoir states are then given by

$$a_i(t) = \int_0^\infty \exp(j\omega i t') \underbrace{\exp(-t'/\tau_R) s(t-t')}_{s_W(t,t')}, \quad (19)$$

where $s_W(t, t')$ is what we will call the “windowed signal”, the signal at a time $t-t'$, multiplied by an exponential window function $\exp(-t'/\tau_R)$. One immediately notices that the reservoir states in the above equation are very similar to the first N Fourier coefficients of a discrete Fourier transform of s_W . Signal reconstruction can then be performed by constructing the Fourier series. This kind of reservoir is basically made of a set of damped resonators, which act as band-pass filters. Therefore, we shall simply call them *resonator reservoirs* for the remainder of this article.

The nature of resonator reservoirs allows us to make good analytical approximations for its memory properties. Although not too complicated, the necessary derivations are quite involved. To avoid breaking up the main text, we have placed all calculations in Appendix C, and we will limit ourselves to mentioning the main results here. First we will assume $\epsilon = 0$ and describe the properties of the MF and an approximation of its memory quality. Next we study the effects of noise.

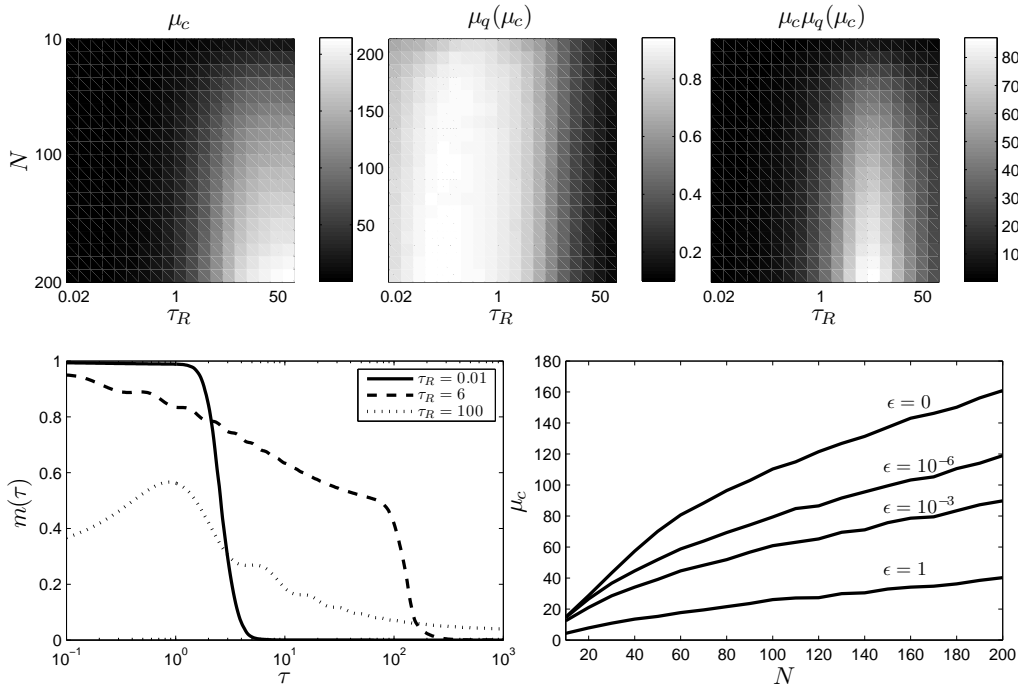


Figure 4: Top three panels: μ_c , $\mu_q(\mu_c)$ and $\mu_c\mu_q(\mu_c)$ in function of N and τ_R for exponentially distributed eigenvalue networks (τ_R shown on a logarithmic scale and $\epsilon = 0$). Bottom left panel: MF with respect to different values of τ_R . The argument τ is shown on a logarithmic scale, $N = 100$. Bottom right panel: memory capacity in function of N with respect to different noise levels. All results in this figure have been found by averaging over 50 reservoir initializations.

3.4.1. Tunability

A discrete Fourier transform is defined for a periodic function or a function within an interval. We can define this period as the longest resonance period present in the network, which we shall call the *reservoir period* $T_R = \frac{2\pi}{\omega}$. The first problem which becomes apparent is the fact that the exponential window stretches on asymptotically to infinity. When this window has dropped significantly at T_R , reconstruction of the signal further in the past is heavily limited by interference from the more recent part of the signal. As a consequence, $m(\tau)$ will drop off fast for $\tau > T_R$ and we will only have useful memory up to a time T_R in the past. If the exponential window stretches on far beyond T_R , signal reconstruction for $\tau < T_R$ is in the same way hindered by interference from the signal stretching beyond T_R in the past.

This result means that we can in fact tune the reservoir to have a MF which is concentrated in an interval $\tau \in \{0 \dots T_R\}$ simply by tuning ω . Choosing the reservoir timescale τ_R has to be such that interference from $\tau > T_R$ is negligible, but should still allow for good reconstruction of the signal for $\tau < T_R$.

In Appendix C.1 we find the general shape of the MF. It consists of a periodic function with period T_R , multiplied by a factor $\exp(-2\tau/\tau_R)$. This quantifies the above statement in the sense that the MF which stretches beyond T_R is a factor $\exp(-2T_R/\tau_R)$ smaller than the MF for $\tau < T_R$. A depiction of this property is presented in

Figure 7a.

3.4.2. Memory quality

Since it is obvious that good memory can only be achieved for up to a history T_R , we will use the memory quality $\mu_q(T_R)$ as the main criterion to qualify the memory of resonator reservoirs.

It is easy to discern the factors which will limit $\mu_q(T_R)$. Most importantly, signal reconstruction is limited by the finite number of Fourier coefficients which can be extracted from the reservoir. A full description of the signal within the reservoir period needs an infinite amount of Fourier coefficients. However, since the signal we consider here has a spectrum which drops asymptotically for high frequencies, a truncated Fourier series can already give a good approximation. The second effect we have to consider is the balance between signal interference from $\tau > T_R$, and the reconstruction within the reservoir period. Notice that the signal that can be reconstructed from the reservoir states is s_W , which then has to be divided by the window function to retrieve $s(t)$. If the window function is already very low when $\tau < T_R$, division will significantly amplify any errors.

In Appendix C.2, we derived an approximate solution for the memory quality which takes in account the mentioned effects. We find

$$\mu_q(T_R) = \frac{2}{\pi} \left[1 - e^{-2\frac{T_R}{\tau_R}} \right] \arctan \left(\frac{\pi N}{T_R(\alpha + \tau_R^{-1})} \right). \quad (20)$$

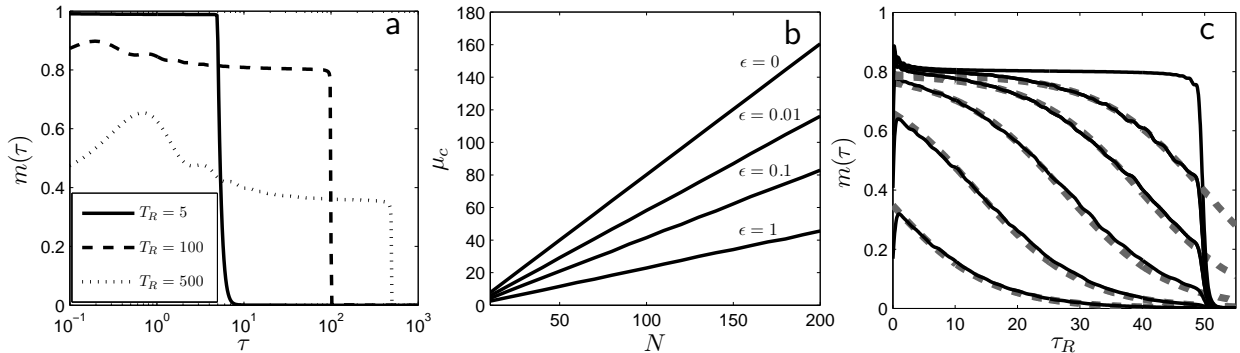


Figure 5: Properties of resonator reservoirs. (a) MF with respect to different values of T_R . τ shown on logarithmic scale, $N = 100$, and $\tau_R = 0.3T_R$. (b) Memory capacity in function of N for different values of noise. For this experiment, $T_R = N$, and again $\tau_R = 0.3T_R$. (c) Black lines: memory functions at different values of noise, from left bottom corner to the right, ϵ equals 10, 1, 0.1, 0.01, 0.001, and zero. Grey dashed lines are predicted curves as given by equation C.4, where τ_s has been determined simply by visual matching. $N = 50$, $T_R = 50$, $\tau_R = 0.3T_R$. All results in this figure have been found by averaging over 50 reservoir initializations.

This expression allows us to find an optimal solution for τ_R when no noise is present (either numerically or graphically, since this is a transcendental equation for τ_R), which is $\tau_R \approx 0.3T_R$. It also clearly expresses the relation between μ_c , N and T_R , which is depicted in Figure 7b. Basically, memory quality has to be sacrificed to increase T_R , as one would intuitively expect.

We can now confirm the limit situation from equation (13) analytically, where τ_R as well as T_R goes to infinity. We indeed find (see Appendix C.2.3) that $\mu_c = \frac{2N}{\alpha}$. This can be easily confirmed with the numerical evaluation of the analytical model. Using 100 neurons and $\tau_R = T_R = 10^5$ indeed yields a memory capacity of almost 200.

3.4.3. Effects of Noise

Resonator reservoirs are particularly robust against noise, as one can see in Figure 5b. Here, memory capacity grows in fact linearly with reservoir size. Also, the memory capacity remains very good at a realistic SNR of 10^{-2} . This reflects the fact that a Fourier decomposition is a very efficient way to decompose a signal. The condition number of the correlation matrix \mathbf{A} is not necessarily low and is for instance on average of the order 10^8 for $N = 100$. However, most of the normalized eigenvalues $\gamma_i \bar{\gamma}^{-1}$ are not small at all, and usually only few of the normalized eigenvalues are truly small, most are around the order 10^{-2} .

The shape of the MF under the influence of noise is depicted in Figure 5c. The cut-off at T_R remains, and with increasing noise levels the MF decays starting close to $\tau = T_R$. In Appendix C.2.4 we derive the shape of the MF under the influence of noise. We also find that with increasing noise level, this function will shift to the left within the range $\{0 \dots T_R\}$, as can be clearly seen in the figure. Unfortunately, we have not been able to fully quantify this phenomenon.

4. Discussion

One of the key features in RC is its intrinsic memory: it is able to store a certain amount of information on past inputs in its immediate spatial state. This property allows it to process sequences of data using a simple memoryless readout. This transient memory property can be characterized by the memory function, which quantifies how well the input signal from a certain time in the past can be reconstructed using only the current reservoir state.

Due to its generic character, RC is not limited to digital implementations and can be extended to physical systems, as in Fernando and Sojakka (2003); Jones et al. (2007); Vandoorne et al. (2008). We explored memory properties of linear first order systems in continuous time. We found an expression for the MF for all networks which have a diagonalizable connection matrix. This expression allows for fast numerical calculation of the MF or properties such as memory capacity.

We discovered that the MF for continuous time reservoirs is critically dependent on the nature of the signal it needs to remember. More specifically, the MF depends on the autocorrelation function of the input signal. We found what we expect strongly to be an upper bound on the memory capacity of linear first order dynamical system, which scales simply with the number of neurons N , which is equivalent to results in discrete time reservoirs, and would suggest that constructing reservoirs with sufficient memory capacity can simply be achieved by increasing the number of neurons. However, upon trying to substantiate this claim, it became immediately obvious that in realistic scenarios, the memory capacity is severely reduced by the conditioning of the correlation matrix and consequently it grows much slower than linear with N for reservoir types described in Section 3.2. This is caused by the inability to invert the correlation matrix of the reservoir states to great numerical accuracy.

We significantly improved the scaling of the memory

capacity with N by the transformation of discrete time reservoir dynamics to continuous time. Some important prescriptions for “good reservoirs” in discrete time have been made in past research: one which enriches the dynamics of nonlinear reservoirs (Ozturk et al., 2006), and one which makes the MF very robust against noise (White et al., 2004). For both these types of reservoirs we defined continuous time equivalents. First of all, the reservoirs described in Section 3.3 reached far higher memory capacity than reservoirs with random connection matrices, but still suffer from a significant noise sensitivity. Finally, resonator reservoirs (described in Section 3.4) - specifically built for good memory - outperform the other types in noise robustness significantly, and indeed show a linear increase in memory capacity with increasing N . Furthermore, memory depth of resonator reservoirs can be tuned such that memory is very good up to a certain time T_R in the past.

This work sheds light on memory properties of continuous time dynamical systems. Unfortunately, as is common in the field of RC, precise analytical modeling is restricted to the linear case, whereas RC derives its power from nonlinearities. As stated before, a linear first order dynamical system can only perform a linear filtering operation on its input. Therefore, future work should focus on testing the results found in this paper to dynamical systems with some nonlinearity and to investigate the computational power for the different kinds of reservoirs discussed in this work. Especially in the design of Echo State Networks with leaky integrator neurons, using an eigenvalue distribution similar to that used in Section 3.3 can be a valuable improvement over simply using random matrices. Another very interesting research direction is the inclusion of fixed delays into the network, as this is a common (and important) feature in many physical dynamical systems.

Acknowledgements

This work was partially funded by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), the Photonics@be Interuniversity Attraction Poles program (IAP 6/10), FWO Flanders project # G.0088.09, and # FP7-231267 (ORGANIC) of the European commission. We would also like to thank Professor Dr. Jan Van Campenhout for helpful suggestions.

Appendix

A. Construction of connection matrices

A.1. For a given eigenvalue distribution

Here we explain the process for building connection matrices with a given eigenvalue distribution. Though this is common knowledge, we include it here for completeness. To build a connection matrix \mathbf{W} for a given

eigenvalue distribution we start of with a diagonal matrix \mathbf{D} with the eigenvalues ordered by absolute value on the diagonal. Since our resulting connection matrix has to have real elements, all eigenvalues will be either real, or complex conjugated pairs. In the next step, we perform an orthogonal transformation to make this a real block-diagonal matrix. To do this, we construct a matrix \mathbf{O} . When D_{ii} is real, $O_{ii} = 1$. When D_{ii} and $D_{i+1,i+1}$ are a complex conjugated pair, we take the elements of the 2×2 block on the diagonal at the i -th and $i + 1$ -th row and column of \mathbf{O} as

$$[\mathbf{O}]_{\{i,i+1\}} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{j}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \end{pmatrix}.$$

All other elements of \mathbf{O} are zero. Finally, we can transform \mathbf{D} to a real block-diagonal form: $\mathbf{D}_b = \mathbf{O}\mathbf{D}\mathbf{O}^\dagger$. The resulting matrix is a block-diagonal matrix with the same structure as \mathbf{O} . All real eigenvalues remain in the same place as in \mathbf{D} , complex pairs of eigenvalues are replaced by a block with elements

$$[\mathbf{D}_b]_{\{i,i+1\}} = \begin{pmatrix} \Re(\lambda_i) & \Im(\lambda_i) \\ -\Im(\lambda_i) & \Re(\lambda_i) \end{pmatrix},$$

Which means one can also directly construct \mathbf{D}_b from the real and imaginary parts of the eigenvalues.

One can already use \mathbf{D}_b as a connection matrix, where it is clear that all single diagonal entries simply act as disconnected low-pass filters, and 2×2 blocks are associated with two interconnected neurons which together act as a damped resonator. A more general connection topology can be constructed by a similarity transform:

$$\mathbf{W} = \mathbf{C}\mathbf{D}_b\mathbf{C}^{-1},$$

where \mathbf{C} can be any nonsingular matrix with the eigenvectors of \mathbf{W} as its columns. For the connection matrices used for empirical testing throughout this paper we will choose \mathbf{C} with random Gaussian elements.

A.2. Random matrices

The update equation for discrete time linear neural networks is given by

$$\mathbf{a}_{i+1} = \mathbf{W}'\mathbf{a}_i + \mathbf{v}'s_i,$$

where the subindex states the time and not vector element positions. The spectral radius of \mathbf{W}' has to be smaller or equal to one to ensure stability. When we assume discrete time runs in steps of duration Δt , and we implement a low-pass filtering operation with timescale τ_R (which - we will see - approximately fits its previous definition), we can rewrite this equation as

$$\begin{aligned} \mathbf{a}_{i+1} &= \left(1 - \frac{\Delta t}{\tau_R}\right) \mathbf{a}_i + \frac{\Delta t}{\tau_R} \mathbf{W}'\mathbf{a}_i + \frac{\Delta t}{\tau_R} \mathbf{v}'s_i \\ &= \mathbf{a}_i + \frac{\Delta t}{\tau_R} (\mathbf{W}' - \mathbf{I})\mathbf{a}_i + \frac{\Delta t}{\tau_R} \mathbf{v}'s_i, \end{aligned}$$

based on Jaeger et al. (2007). When we rewrite this equation as a finite difference approximation of equation (2), we get

$$\begin{aligned} \frac{\mathbf{a}_{i+1} - \mathbf{a}_i}{\Delta t} &= \underbrace{\frac{1}{\tau_R}(\mathbf{W}' - \mathbf{I})}_{\mathbf{W}} \mathbf{a}_i + \underbrace{\frac{1}{\tau_R} \mathbf{v}'}_{\mathbf{v}} s_i \\ &= \mathbf{W} \mathbf{a}_i + \mathbf{v} s_i, \end{aligned}$$

Obviously, when $\Delta t \rightarrow 0$ this results in equation (2), which is the inverse of the Euler approximation for integrating a differential equation. This means that we can simply use any random matrix with a spectral radius $\eta \leq 1$, subtract the unit matrix and scale with τ_R^{-1} .

Random matrices (with elements independently drawn from a Gaussian distribution with zero mean and unit standard deviation) have their eigenvalue spectrum roughly spread out in a disk centered on the origin of the complex plain with radius \sqrt{N} (known as Girko's circular law (Girko, 1984)). We assume \mathbf{W}' is scaled on its spectral radius and we also assume that the mean of the eigenvalues is roughly equal to zero. One can then find that the eigenvalue spectrum of \mathbf{W} is simply that of \mathbf{W}' , shifted to the left on the complex plain by 1, and scaled to τ_R^{-1} . When looking at the eigenvalue decompositions $\mathbf{W} = \mathbf{C} \mathbf{D} \mathbf{C}^{-1}$ and $\mathbf{W}' = \mathbf{C}' \mathbf{D}' \mathbf{C}'^{-1}$ we can write

$$\begin{aligned} \mathbf{D} &= \frac{1}{\tau_R} \mathbf{C}^{-1} (\mathbf{W}' - \mathbf{I}) \mathbf{C} \\ &\Downarrow \\ \mathbf{D} + \frac{1}{\tau_R} \mathbf{I} &= \frac{1}{\tau_R} \mathbf{C}^{-1} \mathbf{W}' \mathbf{C}. \end{aligned}$$

Since the left hand side is diagonal, the right hand side is the unique eigenvalue decomposition of \mathbf{W}' and $\mathbf{C}' = \mathbf{C}$ and $\mathbf{D} = \tau_R^{-1} (\mathbf{D}' - \mathbf{I})$, which means the eigenvalues of \mathbf{W}' are shifted to the left.

Stability in continuous time for equation (2) is guaranteed when all eigenvalues of \mathbf{W} lie on the left half of the complex plain. Since all eigenvalues of \mathbf{W}' lie in the unit disk, which is then shifted to the left by 1 and scaled with a positive number τ_R^{-1} , we indeed get a stable system. The timescale τ_R fits its previous definition when the mean of the eigenvalues of \mathbf{W}' equals zero. This is of course generally not true, and for constructing matrices for testing in Section 3.2, we used a similar shifting of the eigenvalues of the initial random matrix, where we subtracted its initial mean value so that the spectrum is centered around zero, before we scale to the spectral radius η .

The above process can be summarized in one single equation. Starting from any random matrix \mathbf{W}_0 with eigenvalues λ_i^0 with mean value $\bar{\lambda}^0$, we can construct a connection matrix \mathbf{W} to be used in continuous time:

$$\mathbf{W} = \frac{1}{\tau_R} \left(\eta \frac{\mathbf{W}_0 - \bar{\lambda}^0 \mathbf{I}}{\max |\lambda_i^0 - \bar{\lambda}^0|} - \mathbf{I} \right)$$

A.3. Evenly distributed eigenvalues

One strategy to generate eigenvalues for an exponentially distributed spectrum, is to use a random number

generator with an exponential distribution for the real part, and one with a uniform distribution for the imaginary part. The problem that one faces with this strategy is that the eigenvalues will not necessarily be evenly spread; some places will be crowded, others empty. This gives a very large variance of the MFs; some performing very poor, others very good. Therefore, we shall use a simple algorithm that avoids clustering of eigenvalues. We start from the z -domain where we spread eigenvalues more or less evenly, and later transform them to the Laplace domain using the z -transform (see Appendix B).

The method used in Ozturk et al. (2006) to generate even distributions is based on Erdogmus et al. (2003), which uses an iterative method with entropy maximization as its end goal. We used a much simpler approach starting from a geometric point of view. We first define the upper half of the unit disk. The first eigenvalue of the discrete system is chosen randomly from this circle segment. Next, we define a circle of a certain radius ρ_h around this point, and make sure no other eigenvalues can be chosen within it. We repeat the process until we have defined $N/2$ points and then include their complex conjugates. We also make sure no eigenvalue is chosen with an imaginary part smaller than $\rho_h/2$, which avoids clustering with the complex conjugates. ρ_h has to be chosen small enough so that there will be enough space to have $N/2$ eigenvalues within the given area, but large enough to avoid clustering. Since the area cut out by each circle is proportional to ρ_h^2 , and the total area cut out by the circles is proportional to N , ρ_h will have to be proportional to $N^{-1/2}$. Rather than meticulously working out the necessary conditions for ρ_h , we eventually settled after some trial and error to choose $\rho_h = (1.7N)^{-1/2}$. An example of the resulting distribution of eigenvalues is given in Figure 6.

B. z -transform of eigenvalue spectrum

To transform a dynamical system in continuous time to discrete time, one has to use the z -transform (see for instance Jury (1964)). The transformation between the complex variable z in the z -domain and s in the Laplace domain is defined as

$$z = e^{sT_s},$$

where T_s is the sampling period by which the input signal is sampled. Discrete-time reservoirs can in fact be considered as a system where the input signal is sampled from a continuous signal at each time step. As such, we can transform this system to a continuous time equivalent by an inverse z -transform. Since reservoir dynamics are predominantly determined by the eigenvalues, we can use the above equation to find the Laplace-domain equivalent eigenvalues. Obviously, the sample period has no well defined meaning in this reasoning. However, when applying the transformation in the two following examples, one can quickly see that it is fact proportional to the reservoir timescale.

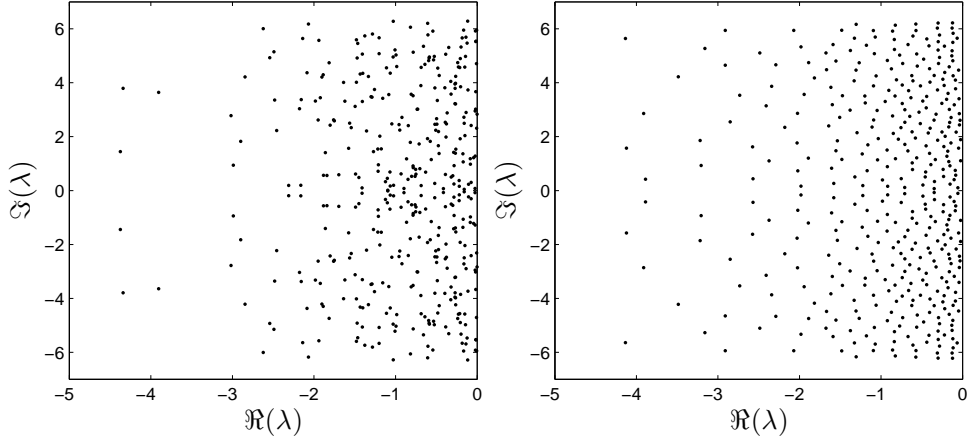


Figure 6: Examples of eigenvalue distributions using random number generators (left), or using the algorithm described in the text (right). $N = 500$, $\tau_R = 1$.

B.1. Laplace-eigenvalue distribution for a uniform distribution in the z -domain

Starting from a distribution in the z -domain, defined in Ozturk et al. (2006) as uniform over a disk with radius η , we can again use the z -transform to determine the distribution of eigenvalues in the Laplace domain. Using coordinates σ and ϖ which denote the real and imaginary part of s , we can define an infinitesimal patch of area $d\sigma d\varpi$. In the z -domain, we use coordinates ρ and ϕ for the radius and angle. A patch of area is here defined as $\rho d\rho d\phi$. The expected number of eigenvalues in this patch is proportional to its surface because of the uniform distribution. Using the relation $\rho = e^{\sigma T_s}$ we find $d\rho = T_s e^{\sigma T_s} d\sigma$. Together with $d\phi = d\varpi T_s$ we can finally write that the expected number of eigenvalues in the patch $d\sigma d\varpi$ is proportional to $T_s^2 e^{2\sigma T_s} d\sigma d\varpi$. Since ϕ goes from $-\pi$ to π , the distribution in function of ϖ will be uniform between $\varpi \in \{-\pi/T_s \dots \pi/T_s\}$, and zero outside this range. Equivalently, we find that for $\rho > \eta$, the distribution is zero, so in the Laplace-domain the distribution will be equal to zero for $\sigma < \ln(\rho)/T_s$. We can then finally write for the distribution of eigenvalues in the Laplace domain D_λ :

$$D_\lambda(\sigma, \varpi) \sim e^{2\sigma T_s} \text{u} \left(\frac{\ln(\rho)}{T_s} - \sigma \right) \text{rect} \left(\frac{T_s \varpi}{2\pi} \right),$$

where $\text{u}(x)$ is the unit step function, and $\text{rect}(x)$ is the rectangle function, equal to one when $x \in \{-1/2 \dots 1/2\}$, and zero everywhere else. Since we defined τ_R as the inverse of the mean real part of the eigenvalues, we can do the same here and use the above formula to find that $\tau_R^{-1} = (2T_s)^{-1} + \ln(\eta)$. For simplicity, we assume $\eta = 1$. This finally yields for the distribution D_λ :

$$D_\lambda(\sigma, \varpi) \sim e^{\sigma \tau_R} \text{u}(-\sigma) \text{rect} \left(\frac{\tau_R \varpi}{4\pi} \right). \quad (\text{B.1})$$

B.2. Resonator reservoirs

Here we base ourselves on a result found in White et al. (2004). In this paper it was found that optimal noise robustness for memory storage is found for reservoirs where

the eigenvalues of the connection matrix all lie on a circle on the complex plain centered on the origin, with a radius smaller than 1 (i.e. an orthogonal connection matrix). We assume that discrete-time eigenvalues, denoted by ξ_i can be written as

$$\xi_i = \eta \exp(2\pi j \theta_i),$$

where we use the symbol j as the imaginary unit to avoid confusion with indices i or j . Transformation of this system to the Laplace domain yields

$$\lambda_i = \frac{\ln(\eta) + 2\pi j \theta_i}{T_s},$$

which means all eigenvalues will lie on a line parallel to the imaginary axis which crosses the real axis at $\ln(\eta)/T_s = -\tau_R^{-1}$. We are free to choose the values T_s and η , which means we have control over the imaginary as well as the real part of the eigenvalues. We will choose the eigenvalues to lie equidistantly on this line spread between values $\omega N/2$ and $-\omega N/2$. This way, when choosing i as $i = -(N-1)/2 \dots (N-1)/2$, we can write the eigenvalues as

$$\lambda_i = j\omega i - \frac{1}{\tau_R}. \quad (\text{B.2})$$

C. Memory quality of resonator reservoirs

C.1. General shape of the memory function

We can draw conclusions concerning the shape of the MF of resonator reservoirs when we look at equations (7) and (9) which state that the MF consists of a set of cross-products of the elements $b_i(\tau)$. The timescale τ_R defines the exponential window as defined above and we assume that $\tau_R \gg \alpha^{-1}$, i.e. that the reservoir timescale is much longer than the signal fluctuations. This way, for $\tau \gg \alpha^{-1}$ we can neglect the term with $\exp(-\alpha\tau)$ in (9). The cross-products of the elements of $\mathbf{b}(\tau)$ can then be written as

$$b_i(\tau) b_j^*(\tau) \approx \exp \left(-2 \frac{\tau}{\tau_R} \right) \frac{\exp(j\omega(i-j)\tau)}{(\alpha^2 - \lambda_i^2)(\alpha^2 - \lambda_j^{*2})},$$

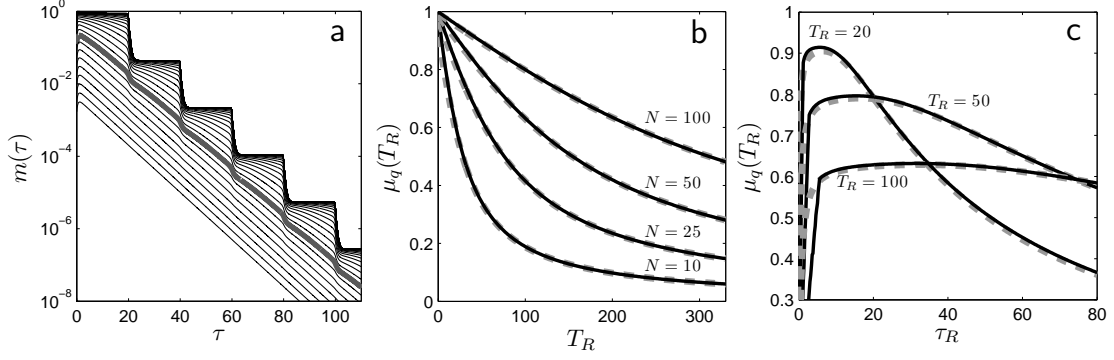


Figure 7: (a) Depiction of the MF of a reservoir with 50 neurons at different noise levels (the thick grey line is at $\epsilon = 1$). $T_R = 20$ and $\tau_R \approx 13.4$: chosen such that $\exp(-2T_R/\tau_R) = 1/20$. The y -axis is on a logarithmic scale to visualize its exponential decay. Notice the sudden drop-off at each multiple of T_R . (b) Depiction of $\mu_q(T_R)$ in function of T_R for different reservoir sizes. The grey dashed lines are the predictions made by equation (C.3), whereas the black lines are from the full analytical model. τ_R is chosen to be 0.3 times the reservoir period T_R , which is an optimal value found with equation (C.3). (c) The influence of signal interference: $\mu_q(T_R)$ in function of τ_R for different T_R , N is chosen at 50. Grey dashed lines are the theoretical prediction found by equation (C.3), black lines are the values found for the full analytical model.

which means that the MF consists of a factor which is periodic with maximum period T_R , and a factor which decays exponentially with decay period $\tau_R/2$. In Figure 7a is a depiction of the MF for a resonator reservoir which confirms this.

C.2. Approximation of the memory quality

Here we will derive the approximation for the memory quality $\mu_q(T_R)$ for resonator reservoirs. We will split the calculations up in two main parts. First, we investigate interference from the signal beyond the reservoir period, next we will account for the finite number of Fourier coefficients.

We will redefine the windowed signal $s_W(t, t')$ as being equal to $s(t - t') \exp(-t'/\tau_R)$ for $0 < t' < T_R$ and zero elsewhere. It is useful to state this as its full Fourier series:

$$s_W(t, t') = \sum_{i=-\infty}^{\infty} e^{j\omega i t'} a_i(t),$$

where

$$a_i(t) = \frac{1}{T_R} \int_0^{T_R} e^{j\omega i t'} e^{-\frac{t'}{\tau_R}} s(t - t') dt'.$$

Next we define $\tilde{s}_W(t - t')$ as the truncated Fourier series:

$$\tilde{s}_W(t, t') = \sum_{i=-(N-1)/2}^{(N-1)/2} e^{j\omega i t'} a_i(t).$$

Thirdly, we define the actually reconstructed windowed signal $\hat{s}_W(t, t')$, which is also a truncated Fourier series, but has coefficients which are defined by equation (19), i.e:

$$\hat{s}_W(t, t') = \sum_{i=-(N-1)/2}^{(N-1)/2} e^{j\omega i t'} a'_i(t),$$

with (adding the scaling factor T_R^{-1})

$$a'_i(t) = \frac{1}{T_R} \int_0^{\infty} e^{j\omega i t'} e^{-\frac{t'}{\tau_R}} s(t - t') dt'. \quad (\text{C.1})$$

Similar to $s_W(t, t')$ we define $\tilde{s}_W(t, t')$ and $\hat{s}_W(t, t')$ to be zero outside the interval $0 < t' < T_R$.

C.2.1. Signal interference

Looking at equation (C.1), we can divide the integration in equal intervals, i.e., we define

$$\int_0^{\infty} f(t) dt = \sum_{j=0}^{\infty} \int_0^{T_R} f(t + jT_R) dt.$$

Since $\frac{2\pi}{\omega} = T_R$, this yields

$$\begin{aligned} a'_i(t) &= \int_0^{\infty} e^{j\omega i t'} e^{-\frac{t'}{\tau_R}} s(t - t') dt' \\ &= \sum_{j=0}^{\infty} e^{-j\frac{T_R}{\tau_R}} \int_0^{T_R} e^{j\omega i t'} e^{-\frac{t'}{\tau_R}} s(t - t' - jT_R) dt' \\ &= \sum_{j=0}^{\infty} a_i(t - jT_R) e^{-j\frac{T_R}{\tau_R}}. \end{aligned}$$

We can then redefine $\hat{s}_W(t, t')$ as

$$\begin{aligned}
\widehat{s}_W(t, t') &= \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j\omega i t'} \sum_{j=0}^{\infty} a_i(t - jT_R) e^{-j\frac{T_R}{\tau_R}} \\
&= \sum_{j=0}^{\infty} e^{-j\frac{T_R}{\tau_R}} \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j\omega i t'} a_i(t - jT_R) \\
&= \sum_{j=0}^{\infty} e^{-j\frac{T_R}{\tau_R}} \tilde{s}_W(t - jT_R, t').
\end{aligned}$$

Finally, we can use this expression in the first step to finding the memory quality. Since the MF does not depend on the scaling of the reconstructed signal, we can write (replacing t' with τ)

$$m(\tau)_{[\tau \in \{0 \dots T_R\}]} = \frac{\langle s_W(t, \tau) \widehat{s}_W(t, \tau) \rangle_t^2}{\sigma^2(s_W(t, \tau)) \sigma^2(\widehat{s}_W(t, \tau))}.$$

The numerator is given by

$$\begin{aligned}
&\langle s_W(t, \tau) \widehat{s}_W(t, \tau) \rangle_t^2 \\
&= \sum_{j=0}^{\infty} e^{-j\frac{T_R}{\tau_R}} \langle s_W(t, \tau) \tilde{s}_W(t - jT_R, \tau) \rangle_t^2 \\
&\approx \langle s_W(t, \tau) \tilde{s}_W(t, \tau) \rangle_t^2,
\end{aligned}$$

since we can safely assume that the present signal will be virtually uncorrelated with the signal from a multiple of T_R in the past. The denominator can be worked out in a similar manner. Calculating the variance for $\widehat{s}_W(t, t')$, we find

$$\begin{aligned}
\sigma^2(\widehat{s}_w(t, \tau)) &= \lim_{P \rightarrow \infty} \frac{1}{2P} \int_{-P}^P \widehat{s}_W^2(t, \tau) dt \\
&= \lim_{P \rightarrow \infty} \frac{1}{2P} \int_{-P}^P \left[\sum_{j=0}^{\infty} e^{-j\frac{T_R}{\tau_R}} \tilde{s}_W(t - jT_R, \tau) \right]^2 dt
\end{aligned}$$

Again, we neglect correlation between the present signal and the signals which extend multiple times T_R in the past. As such we can rewrite this as

$$\begin{aligned}
\sigma^2(\widehat{s}_w(t, \tau)) &\approx \sum_{j=0}^{\infty} e^{-2j\frac{T_R}{\tau_R}} \lim_{P \rightarrow \infty} \frac{1}{2P} \int_{-P}^P \tilde{s}_W^2(t - jT_R, \tau) dt \\
&= \sigma^2(\tilde{s}_W(t, \tau)) \sum_{j=0}^{\infty} e^{-2j\frac{T_R}{\tau_R}} \\
&= \frac{\sigma^2(\tilde{s}_W(t, \tau))}{1 - e^{-2\frac{T_R}{\tau_R}}},
\end{aligned}$$

which finally leads us to the MF up to T_R in the past:

$$m(\tau)_{[\tau \in \{0 \dots T_R\}]} = \left[1 - e^{-2\frac{T_R}{\tau_R}} \right] \frac{\langle s_W(t, \tau) \tilde{s}_W(t, \tau) \rangle^2}{\sigma^2(s_W(t, \tau)) \sigma^2(\tilde{s}_W(t, \tau))}.$$

C.2.2. Consequences of truncated Fourier series

To calculate the memory quality, we will have to make further approximations:

1. We assume that $m(\tau)$ is nearly constant in the range $\tau \in \{0 \dots T_R\}$. This constant is equal to the memory quality $\mu_q(T_R)$. The assumption can be validated by looking at Figure 5.
2. We assume that $\langle s_W(t, \tau) \tilde{s}_W(t, \tau) \rangle$, $\sigma^2(s_W(t, \tau))$, and $\sigma^2(\tilde{s}_W(t, \tau))$ all evolve with τ as $\exp(-2\tau/\tau_R)$, multiplied by some constant value. This assumption is exactly true for $\sigma^2(s_W(t, \tau))$, and approximately for the others as long as τ_R is not too small relative to T_R .

Applying this approximation, we only need to find the relative proportions of $\langle s_W(t, \tau) \tilde{s}_W(t, \tau) \rangle$, $\sigma^2(s_W(t, \tau))$, and $\sigma^2(\tilde{s}_W(t, \tau))$ to find the actual memory quality. In order to do this, we integrate these expressions over τ in the reservoir period. We find

$$\begin{aligned}
&\int_0^{T_R} \langle s_W(t, \tau) \tilde{s}_W(t, \tau) \rangle_t d\tau \\
&= \int_0^{T_R} d\tau \left\langle \sum_{i=-\infty}^{\infty} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j\omega(i-j)\tau} a_i(t) a_j^*(t) \right\rangle_t \\
&= \sum_{i=-\infty}^{\infty} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \langle a_i(t) a_j^*(t) \rangle_t \int_0^{T_R} e^{j\omega(i-j)\tau} d\tau \\
&= \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \langle |a_i(t)|^2 \rangle_t.
\end{aligned}$$

Similarly, we find

$$\begin{aligned}
\int_0^{T_R} \sigma^2(s_W(t, \tau)) d\tau &= \sum_{i=-\infty}^{\infty} \langle |a_i(t)|^2 \rangle \\
\int_0^{T_R} \sigma^2(\tilde{s}_W(t, \tau)) d\tau &= \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \langle |a_i(t)|^2 \rangle.
\end{aligned}$$

This finally yields for the memory quality

$$\mu_q(T_R) \approx [1 - \exp(-2T_R/\tau_R)] \frac{\sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \langle |a_i(t)|^2 \rangle}{\sum_{i=-\infty}^{\infty} \langle |a_i(t)|^2 \rangle}. \quad (\text{C.2})$$

The terms $\langle |a_i(t)|^2 \rangle$ form the power spectrum of the windowed signal. We can use the Wiener-Khinchin theorem which states that the power spectrum of a signal is equal to the spectrum of its autocorrelation function, which we shall denote as $R_W(t)$. Since this is a discrete

spectrum, we have to assume the windowed function is periodic and calculate the autocorrelation function accordingly. Since we take the mean power spectrum over t , we shall take the mean over t for the autocorrelation function as well. This will allow us to incorporate the signal statistics $R(t) = \exp(-\alpha|t|)$. We can calculate

$$\begin{aligned}
& \langle R_W(t') \rangle_t \\
&= \left\langle \int_0^{T_R-t'} s_W(t, \tau) s_W(t, \tau + \theta) d\tau \right\rangle_t \\
&+ \left\langle \int_0^{t'} s_W(t, T_R - t' + \tau) s_W(t, \tau) d\tau \right\rangle_t \\
&= \int_0^{T_R-t'} e^{-\frac{2\tau+t'}{\tau_R}} \underbrace{\langle s(t-\tau) s(t-\tau+t') \rangle_t}_{R(t')} d\tau \\
&+ \int_0^{t'} e^{-\frac{T_R-t'+2\tau}{\tau_R}} \underbrace{\langle s(t-\tau) s(t-\tau+t'-T_R) \rangle_t}_{R(t'-T_R)} d\tau \\
&= \frac{\tau_R}{2} \left(e^{-t'(\alpha+\tau_R^{-1})} (1 - e^{(t'-T_R)\tau_R^{-1}}) \right) \\
&+ \frac{\tau_R}{2} \left(e^{(t'-T_R)(\alpha+\tau_R^{-1})} (1 - e^{-t'\tau_R^{-1}}) \right) \\
&\approx \frac{\tau_R}{2} \left(e^{-t'(\alpha+\tau_R^{-1})} + e^{(t'-T_R)(\alpha+\tau_R^{-1})} \right).
\end{aligned}$$

The discrete Fourier spectrum of this function can be calculated as

$$\langle |a_i(t)|^2 \rangle_t = \frac{1}{T_R} \int_0^{T_R} \exp(j\omega t') \langle R_W(t') \rangle_t dt',$$

which yields

$$\langle |a_i(t)|^2 \rangle_t \sim \frac{1}{\frac{T_R^2}{4\pi^2} (\alpha + \tau_R^{-1})^2 + i^2}.$$

The sums in equation (C.2) can be approximated by integrals:

$$\sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \langle |a_i(t)|^2 \rangle_t \sim \int_{-N/2}^{N/2} \frac{1}{\frac{T_R^2}{4\pi^2} (\alpha + \tau_R^{-1})^2 + q^2} dq,$$

and similar for the denominator. This finally yields for the memory quality

$$\mu_q(T_R) = \frac{2}{\pi} \left[1 - e^{-2\frac{T_R}{\tau_R}} \right] \arctan \left(\frac{\pi N}{T_R(\alpha + \tau_R^{-1})} \right). \quad (\text{C.3})$$

The validity for this approximation is pictured in Figure 7b and 7c. It appears this gives a good estimate for the memory quality as long as τ_R is not too small compared to T_R .

C.2.3. Asymptotic memory capacity

The limit situation $\tau_R \rightarrow \infty$ can now also be worked out. Notice that the assumptions we made in Section 2.5

for the normalized κ_i implies that κ_i remains finite, and so the imaginary parts to have to be finite. This means that, since $\lambda_i = \kappa_i/\tau_R$, the imaginary parts of the eigenvalues will go to zero as well, implying that T_R has to go to infinity together with τ_R for the derivation to remain valid. When we assume that the MF is flat in intervals $\{iT_R \cdots (i+1)T_R\}$, and using the results from (C.1), we can write

$$\mu_c = T_R \sum_{j=0}^{\infty} \mu_q(T_R) e^{-2j\frac{T_R}{\tau_R}} = T_R \frac{\mu_q(T_R)}{1 - e^{-2j\frac{T_R}{\tau_R}}},$$

and together with equation (C.3) this becomes

$$\mu_c = \frac{2}{\pi} T_R \arctan \left(\frac{\pi N}{T_R(\alpha + \tau_R^{-1})} \right).$$

When τ_R and T_R go to infinity, the limit is

$$\lim_{T_R, \tau_R \rightarrow \infty} \mu_c = \frac{2N}{\alpha},$$

confirming equation (13).

C.2.4. Influence of noise

When noise is added to the reservoirs states, the reconstructed windowed signal will consist of the noiseless windowed signal, plus a signal consisting of the Fourier transform of random entries (the noise). When we assume that the noise is generated by a stationary process, we can assume that the second signal has a constant variance in the range $\{0 \cdots T_R\}$. If we divide the reconstructed windowed signal through $\exp(-\tau/\tau_R)$ to reconstruct the original signal, the part of it caused by noise will increase exponentially with τ . If we write the reconstructed windowed signal without noise divided through $\exp(-\tau/\tau_R)$ as $\hat{s}(t-\tau)$ and the signal caused by noise as $c\sqrt{\epsilon}r(t)\exp(\tau/\tau_R)$ where c is a constant and $r(t)$ is the random signal scaled to unit variance. We can then write for the MF

$$\begin{aligned}
& m(\tau)_{[\tau \in \{0 \cdots T_R\}]} \\
&= \frac{\left\langle s(t-\tau) (\hat{s}(t-\tau) + \sqrt{\epsilon}cr(t)e^{\frac{\tau}{\tau_R}}) \right\rangle_t^2}{\sigma^2(s(t-\tau)) \sigma^2(s(t-\tau) + c\sqrt{\epsilon}r(t)e^{\frac{\tau}{\tau_R}})} \\
&= \frac{\mu_q(T_R)_{\epsilon=0}}{1 + \epsilon \underbrace{\frac{c^2}{\sigma^2(\hat{s}(t-\tau))}}_{c'^2} e^{2\frac{\tau}{\tau_R}}} \\
&= \frac{\mu_q(T_R)_{\epsilon=0}}{1 + c'^2 \epsilon e^{2\frac{\tau}{\tau_R}}},
\end{aligned}$$

where we again assume that the covariance and variances in the above expressions do not depend on τ . Unfortunately, calculating c' is far from trivial since it requires detailed knowledge of the optimal readout weights, which

also depend on ϵ . Nevertheless, an interesting observation can be made when writing the above expression as

$$\begin{aligned} m(\tau)_{[\tau \in \{0 \dots T_R\}]} &= \frac{\mu_q(T_R)_{\epsilon=0}}{1 + e^{\frac{2}{\tau_R} \tau + \ln(\epsilon c'^2)}} \\ &= \frac{\mu_q(T_R)_{\epsilon=0}}{1 + e^{\frac{2}{\tau_R}(\tau + \tau_S)}}, \end{aligned} \quad (\text{C.4})$$

where $\tau_S = \frac{\tau_R}{2} \ln(\epsilon c'^2)$. This is basically a reversed sigmoid curve (a Fermi function to be precise). If we consider τ_S to rise monotonically with ϵ , this means that increasing the noise level shifts the above function to the left without changing its shape. The actual MF then approximately fits a reversed Fermi function within the interval $\tau \in \{0 \dots T_R\}$. This behavior can be seen in Figure 5.

References

- Antonelo, E. A., Schrauwen, B., Stroobandt, D., 2008. Event detection and localization for small mobile robots using reservoir computing. *Neural Networks* 21, 862–871.
- Bertschinger, N., Natschläger, T., 2004. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation* 16 (7), 1413–1436.
- Erdogmus, D., Hild, K.E., I., Principe, J., 2003. Online entropy manipulation: stochastic information gradient. *Signal Processing Letters* 10 (8), 242–245.
- Fernando, C., Sojakka, S., 2003. Pattern recognition in a bucket. In: *Proceedings of the 7th European Conference on Artificial Life*. pp. 588–597.
- Ganguli, S., Huh, D., Sompolinsky, H., November 2008. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America* 105 (48), 18970–18975.
- Gerstner, W., Kistler, W., 2002. *Spiking Neuron Models*. Cambridge University Press.
- Girko, V. L., 1984. Circular law. *Theory of Probability and Its Applications* 29, 694–706.
- Hammer, B., Steil, J. J., 2002. Perspectives on learning with recurrent neural networks. In: *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*. pp. 357–369.
- Hermans, M., Schrauwen, B., Stroobandt, D., 2008. Biologically inspired features in spiking neural networks. In: *Proceedings of the 19th Annual Workshop on Circuits, Systems and Signal Processing*. pp. 328–334.
- Hopfield, J. J., Apr 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79 (8), 2554–2558.
- Jaeger, H., 2001a. The “echo state” approach to analysing and training recurrent neural networks. Tech. Rep. GMD Report 148, German National Research Center for Information Technology.
- Jaeger, H., 2001b. Short term memory in echo state networks. Tech. Rep. GMD Report 152, German National Research Center for Information Technology.
- Jaeger, H., Haas, H., April 2 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science* 308, 78–80.
- Jaeger, H., Lukosevicius, M., Popovici, D., 2007. Optimization and applications of echo state networks with leaky integrator neurons. *Neural Networks* 20, 335–352.
- Jolliffe, I. T., 2002. *Principal Component Analysis*. Springer.
- Jones, B., Stekel, D., Rowe, J., Fernando, C., 2007. Is there a liquid state machine in the bacterium *escherichia coli*? In: *IEEE Symposium on Artificial Life*. pp. 187–191.
- Jury, E. I., 1964. *Theory and Application of the Z-Transform Method*. Wiley, New York.
- Legenstein, R. A., Maass, W., 2007. Edge of chaos and prediction of computational performance for neural microcircuit models. *Neural Networks* 20 (3), 323–333.
- Maass, W., Legenstein, R. A., Bertschinger, N., 2005. Methods for estimating the computational power and generalization capability of neural microcircuits. In: Saul, L. K., Weiss, Y., Bottou, L. (Eds.), *Advances in Neural Information Processing Systems*. Vol. 17. MIT Press, pp. 865–872.
- Maass, W., Markram, H., 2004. On the computational power of recurrent circuits of spiking neurons. *Journal of Computer and System Sciences* 69 (4), 593–616.
- Maass, W., Natschläger, T., Markram, H., 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14 (11), 2531–2560.
- Ozturk, M. C., Xu, D., Principe, J. C., 2006. Analysis and design of echo state networks. *Neural Computation* 19, 111–138.
- Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 2 (6), 559–572.
- Prokhorov, D., 2005. Echo state networks: Appeal and challenges. In: *Proceedings of the International Joint Conference on Neural Networks*. pp. 1463–1466.
- Rumelhart, D., Hinton, G., Williams, R., 1986. *Learning internal representations by error propagation*. MIT Press, Cambridge, MA., Ch. 8.
- Schrauwen, B., Defour, J., Verstraeten, D., Van Campenhout, J., 2007. The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. pp. 471–479.
- Sontag, E., 1998. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, New York.
- Tikhonov, A. N., Arsenin, V. Y., 1977. *Solutions of Ill-Posed Problems*. Winston and Sons.
- Vandoorne, K., Dierckx, W., Schrauwen, B., Verstraeten, D., Baets, R., Bienstman, P., van Campenhout, J., 2008. Toward optical signal processing using photonic reservoir computing. *Optics Express* 16 (15), 11182–11192.
- Verstraeten, D., Schrauwen, B., Stroobandt, D., 2006. Reservoir-based techniques for speech recognition. In: *Proceedings of the World Conference on Computational Intelligence*. pp. 1050–1053.
- White, O. L., Lee, D. D., Sompolinsky, H., 2004. Short-term memory in orthogonal neural networks. *Physical Review Letters* 92 (14), 148102.
- wyffels, F., Schrauwen, B., Stroobandt, D., 2008a. Regularization methods for reservoir computing. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. pp. 808–817.
- wyffels, F., Schrauwen, B., Verstraeten, D., Stroobandt, D., 2008b. Band-pass reservoir computing. In: *Proceedings of the International Joint Conference on Neural Networks*. pp. 3203–3208.