On the Quantification of Dynamics in Reservoir Computing

David Verstraeten* and Benjamin Schrauwen

Department of Electronics and Information Systems (ELIS), Ghent University
Sint Pietersnieuwstraat 41, 9000 Ghent, Belgium
David.Verstraeten@UGent.be

Abstract. Reservoir Computing (RC) offers a computationally efficient and well performing technique for using the temporal processing power of Recurrent Neural Networks (RNNs), while avoiding the traditional long training times and stability problems. The method is both simple and elegant: a random RNN (called the *reservoir)* is constructed using only a few global parameters to tune the dynamics into a desirable regime, and the dynamic response of the reservoir is used to train a simple linear regression function called the readout function - the reservoir itself remains untrained. This technique has shown some experimentally very convincing results on a variety of tasks, but a thorough understanding of the importance of the dynamics for the performance is still lacking. This contribution aims to extend this understanding, by presenting a more sophisticated extension on the traditional way of characterizing the reservoir dynamics, by using the dynamic profile of the Jacobian of the reservoir instead of static, *a priori* measures such as the standard spectral radius. We show that this measure gives a more accurate description of the reservoir dynamics, and can serve as predictor for the performance. Additionally, due to the theoretical background from dynamical systems theory, this measure offers some insight into the underlying mechanisms of RC.

1 Introduction: Reservoir Computing

Reservoir Computing (RC) [15], an idea that was originally independently introduced as Echo State Networks (ESN) [5] and Liquid State Machines (LSM) [7] has grown in the last few years into a research subfield that has attracted quite some attention. This is likely due to the attractive properties of the method: it can be used to solve temporal learning tasks without extensive parameter tuning or long training times and is easy to use and understand.

Reservoir Computing relies on the dynamic response of an excitable, (usually) nonlinear medium - the *reservoir* - to a one- or multidimensional input signal. The state of the system - which is in effect a nonlinear transformation with fading memory of the input - is then used as input for a linear regression function, which can be trained using any of the available online or offline training methods for

* Corresponding author.

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 985–994, 2009.

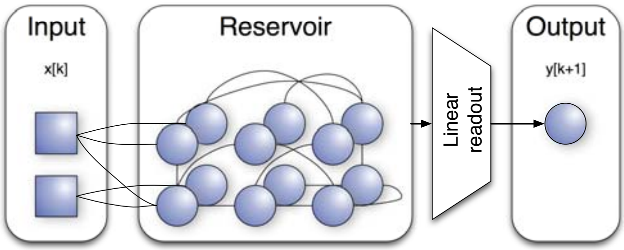

Fig. 1. Schematic view of an RC system, with the input signals (left) driving the reservoir, which is then used as input for the linear readout that in turn extracts the output

linear classifiers or regressors (see Fig. 1). In a sense, the functionality of the reservoir can be interpreted as being a random, temporal and nonlinear kernel [11], which performs a temporal-to-spatial transformation. This transformation boosts the computational capabilities of the subsequent linear classifier, enabling it to solve problems it would not be able to without the reservoir. Thus, in RC, two difficult tasks needed for solving temporal classification or regression problems are elegantly separated, namely the nonlinear dynamic preprocessing, and the training of the actual classifier.

In practice, the methodology to construct and train an ESN system can be summarized as follows:

- Construct a recurrent neural network consisting of N nodes with sigmoid (tanh) nonlinearities. The weights of the $N \times N$ reservoir weight matrix $\mathbf{W_{res}}$ and an $M \times N$ input weight matrix $\mathbf{W_{in}}$ are drawn from a random distribution (e.g. a gaussian distribution), or a from discrete set. Rescale $\mathbf{W_{res}}$ globally, such that the spectral radius of $\mathbf{W_{res}}$ is set to the desired value. The spectral radius of a matrix is its largest absolute eigenvalue, and denoted as $\rho(\mathbf{W_{res}})$. The rationale behind the rescaling is explained later. Rescale W_{in} with a constant value, the *input scale factor* (usually around 1).
- Simulate the network by driving it with an external (possibly multidimensional) input signal $\mathbf{u}[k]$. The network state at time k is denoted as $\mathbf{s}[k]$. The network is simulated recursively, in a timestep based way, as follows: $\mathbf{s}[k+1] = f(\mathbf{W_{res}}\mathbf{s}[k] + \mathbf{W_{in}}\mathbf{u}[k])$.
- Compute the output weights by least squares regression on the matrix $\mathbf{A}$ - which is a concatenation of all vectors $\mathbf{s}[k]$ - using the desired output matrix $\mathbf{o}$ as the right-hand side. I.e., compute the matrix $\mathbf{W_{out}}$ that satisfies the following equation: $\mathbf{W_{out}} = \min_{\mathbf{W}} \|\mathbf{AW} - \mathbf{o}\|^2$ .In practice, this can be done in a single step by using the Moore-Penrose generalized matrix inverse [10], or *pseudo-inverse* $\mathbf{A}^\dagger$ of the matrix $\mathbf{A}$, which is defined as : $\mathbf{A}^\dagger = (\mathbf{A^T A})^{-1}\mathbf{A^T}$, as follows: $\mathbf{W_{out}} = \mathbf{A}^\dagger \mathbf{o} = (\mathbf{A^T A})^{-1}\mathbf{A^T o}$.

– Simulate the network on the test set in the same way as above, and compute the output as follows: $\hat{\mathbf{o}}[k] = \mathbf{W_{out}}\mathbf{s}[k]$.
– Evaluate the performance based on the difference between the RC output $\hat{\mathbf{o}}[k]$ and the target output $\mathbf{o}[k]$.

While RC as a research area is rooted in neural network research and is still mainly active there, its ideas are extendable to other fields. The basic idea of using a nonlinear dynamic system to act as a complex preprocessing filter for the linear readout is very powerful, and can be ported to other research areas. More exotic incarnations of RC have already been described in literature, which include (ranked in order of 'deviation from the standard RNN') : bandpass reservoirs [12,18], Cellular Nonlinear Network (CNN) reservoirs [16] and a reservoir built from coupled nano-photonic nonlinear components [13].

2 Disadvantages of Static Reservoir Measures

As was mentioned in the brief description of the ESN methodology, the spectral radius is an important parameter that controls the dynamic regime of the reservoir. It amounts to a global scaling of the eigenvalues of the connection matrix. From a system theoretic point of view, this can be interpreted as follows: for a small-signal approximation (i.e. the state of the reservoir remains near the zero fix-point), the reservoir can be approximated as a linear time-invariant, discrete-time system:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k]$$
$$\mathbf{y}[k+1] = \mathbf{C}\mathbf{x}[k+1] + \mathbf{D}\mathbf{u}[k+1]$$

where $\mathbf{x}[k]$ represents the state of the reservoir (the vector of neuron activations) at time k, and $\mathbf{u}[k]$ and $\mathbf{y}[k]$ represent the input and output to the system, respectively. The matrix $\mathbf{A}$ contains the internal weights of the reservoir ($\mathbf{W}_{res}$ from above), the B matrix contains the input-to-reservoir weights ($\mathbf{W}_{in}$ from above), and $\mathbf{C}$ and $\mathbf{D}$ contain the (trained) reservoir-to-output ($\mathbf{W}_{out}$ from above) and input-to-output weights respectively (the latter is usually left zero).

It follows from linear system theory [2] that if the matrix A has all singular values smaller than 1[1], it is definitely stable, while if any absolute eigenvalue (i.e. spectral radius) is larger than 1, the system (i.e. the reservoir) will surely be unstable in the sense that it will deviate unboundedly from the fixed point when started from a non-zero state. However, the reservoirs of the ESN type (and the reservoirs that we will consider in this contribution), have a squashing tanh() nonlinearity, that counteracts this unbounded growth - which means that the norm of the state vector of the reservoir will always remain bounded. This nonlinearity also means that the spectral radius as a stability measure looses its significance when the system deviates from an ϵ-region around the zero state.

[1] This implies that the maximal gain in any direction in state space is smaller than one, and the system is always contracting.

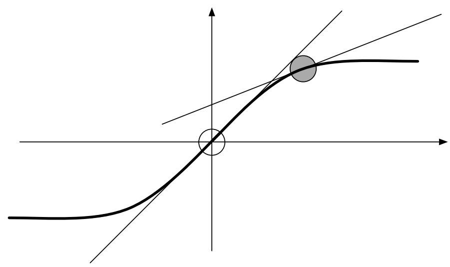

Fig. 2. The gain of the sigmoid nonlinearity is largest around the origin (white circle). Once the neuron is driven by an external signal or a constant bias, the working point shifts up or downward (gray circle) and the local gain decreases, resulting in a less dynamically excitable reservoir.

This means that it is possible that reservoirs with a spectral radius larger than one do possess the echo state property when driven by an external input - this was e.g. proven experimentally in [14]. Once the system is driven by an external input or if a constant bias is fed to the nodes, the operating point of all neurons shifts along the nonlinearity and the effective local gain (i.e. the slope of the tangent in the operating point) becomes smaller (see Fig. 2).

In [5], the linear approximation described above is used to derive some mathematically founded guidelines for constructing weight matrices for suitable reservoirs. The suitability of a reservoir in this case is mainly identified through the presence of the so-called *echo state property*, which roughly states that the state of the reservoir is only determined by the inputs from a sufficiently long time in the past, and that the initial state of the reservoir eventually gets washed out. Several bounds for this property have been described in literature:

- A reservoir whose weight matrix $\mathbf{W}_{res}$ has a largest singular value (LSV) (denoted as $\overline{\sigma}(\mathbf{W}_{res})$) smaller than one, is guaranteed to have the echo state property. However, in practice this guideline is of little use since these reservoirs are not dynamically rich enough to perform well.
- A reservoir whose weight matrix has a spectral radius (SR) - i.e. a largest absolute eigenvalue - larger than one is guaranteed *not* to have the echo state property. So, $\rho(\mathbf{W}_{res}) \leq 1$ is a necessary condition for the echo state property. While the spectral radius criterium is not a sufficient condition, in practice it is used as a guideline for constructing good reservoirs for many problems.
- In [1], a tighter bound on the echo state property than $\overline{\sigma}(\mathbf{W}_{res}) < 1$ was presented. A Euclidean weighted matrix norm $\|\mathbf{W}\|_{\mathbf{D}} = \left\|\mathbf{D}\mathbf{W}\mathbf{D}^{-1}\right\|_2 = \overline{\sigma}(\mathbf{D}\mathbf{W}\mathbf{D}^{-1})$ was introduced, and it turns out that for a certain class of structured weighting matrices $\mathbf{D}_\delta$, the relation $\rho(\mathbf{W}) < \inf_{\mathbf{D}_\delta} \left\|\mathbf{D}_\delta\mathbf{W}\mathbf{D}_\delta^{-1}\right\| < \overline{\sigma}(\mathbf{W})$ holds. The center term $\inf_{\mathbf{D}_\delta} \left\|\mathbf{D}_\delta\mathbf{W}\mathbf{D}_\delta^{-1}\right\|$ is called the structured singular value μ_{SSV}, a quantity widely used in robust control theory. It turns out that μ_{SSV} offers a bound on the echo state property (namely, $\mu_{SSV} < 1$)

that is less conservative than the standard $\overline{\sigma}(\mathbf{W}_{res}) < 1$. However, while this new bound is an improvement over the standard LSV and SR bounds, it is computationally quite demanding to evaluate (21 seconds for a reservoir of 500 nodes, versus .6 seconds to compute the spectral radius).

All the quantities described above are static measures that only take the internal reservoir weight matrix into account and disregard other factors such as input scaling, bias or dynamic range of the input signals - factors that are equally important in defining the dynamic properties of the system. Clearly, an accurate way of quantifying the dynamics of the reservoir, evaluated in the current working point of the reservoir, would be very useful. This notion is explored further in the next section.

3 Quantifying Dynamic Properties of Reservoirs

The readout of the RC system is quite unsophisticated in terms of computational power: it is both linear and memoryless. While these properties enable the application of easy and optimal training algorithms, this also means that complex temporal problems cannot be solved by the linear readout alone. Thus, the functionality of the reservoir is twofold: it should perform a suitably nonlinear transformation of the input so that the discriminating power of the linear readout gets boosted, and it should also offer a fading memory of past inputs to the readout. In some ways, these two functions are contrary to each other: it was shown theoretically that linear networks and even long delay lines have the largest memory of past inputs [4,17,3], but many non-trivial tasks require at least some form of nonlinear behaviour, which reduces the memory of the network. Thus, a good reservoir should ideally find the optimal trade-off between these two opposing goals. This is closely linked to the dynamic regime of the reservoir.

The dynamic properties of the reservoir at a given point in time are determined by a couple of factors: the reservoir weight matrix (this was discussed in the previous section), an optional bias, the nonlinearity of the nodes and the external input that drives the reservoir. These factors determine the operating regime of the reservoir, and as such the local gain of the system at any given time. Here, we will discuss two (related) tools for quantifying these dynamic properties, namely the local Lyapunov exponent (LLE) and the Jacobian of the reservoir. The Jacobian $\mathbf{J}_f$ of a map $f(s)$ is given by:

$$\mathbf{J}_f(\mathbf{s}) = \begin{pmatrix} \frac{\partial f_1}{\partial s_1}(\mathbf{s}) \cdots \frac{\partial f_1}{\partial s_n}(\mathbf{s}) \\ \frac{\partial f_n}{\partial s_1}(\mathbf{s}) \cdots \frac{\partial f_n}{\partial s_n}(\mathbf{s}) \end{pmatrix},$$

where $s = [s_1 s_2 \ldots s_n]$ is the vector of activation values of the neurons in the reservoir, and f is the nonlinearity of the nodes - in this case a tanh(). The matrix contains the local derivative of every state value w.r.t. every other state value. In the case of a tanh() reservoir, this simplifies to:

$$\mathbf{J}_f(\mathbf{s}) = \mathrm{diag}[1 - s_1^2[k], 1 - s_2^2[k], \ldots, 1 - s_n^2[k]]\mathbf{W}_{res},$$

where the notation diag[] signifies a diagonal matrix with the given values on the diagonal, and $\mathbf{W}$ is the weight matrix of the reservoir. From this, the kth LLE $\widetilde{\lambda}_k$ can then be approximated as: $\widetilde{\lambda}_k = \log\left(\prod_{n=1}^{N}(r_k)^{1/n}\right)$ with N the number of timesteps in the trajectory that is considered, and r_k the kth eigenvalue of $\mathbf{J}_f$. The LLE offers a local estimation of the predictability or excitability of a dynamic system around a certain point in state space. It is only an approximation of the true Lyapunov spectrum for two reasons: first of all, we only consider a finite trajectory while the definition of the Lyapunov exponent requires that $N \rightarrow \infty$, and secondly the system under consideration is driven by an external input signal. However, we argue that this spectrum can still offer a valid quantification of the local dynamic properties of the reservoir.

In [15], the relationship between the mean of the maximum of the local Lyapunov spectrum and the performance of the reservoir was studied, and it was found that for a given task, the optimal performance of a reservoir was consistently attained for the same value of the maximal LLE. While this finding was useful from a theoretical point of view because it offered a more refined measure of the reservoir dynamics that the stationary measures mentioned in the previous section, it does not supply a practical means for choosing the reservoir dynamics or offers insight into the meaning of this metric.

Closer inspection of the *complete* local Lyapunov spectrum reveals another, and in some ways more useful phenomenon. Figure 3 shows a plot of the mean over time of all LLEs as the spectral radius of the reservoir is varied from .1 to 3 and the reservoir is driven by noise (which is the input for the NARMA task, see below). The plot shows that the maximal exponent increases monotonically (as was shown previously in [15]), but also that the *minimal* exponent reaches a maximum for a spectral radius of 1, and then decreases again. Thus, the bundle of LLEs becomes narrower and then broader again as the spectral radius of the reservoir weight matrix is increased. More importantly, the maximum of the minimal lyapunov exponent is a good predictor for the optimal performance of the system. In the next section, we will present some more elaborate experimental results and discuss the implications of this phenomenon.

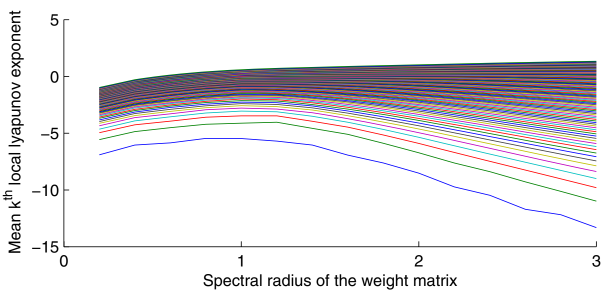

Fig. 3. The full mean (over time) local lyapunov spectrum for a reservoir of 100 nodes for the NARMA task

4 Experiments and Discussion

The maximal LE is - for autonomous systems - an indicator of chaotic behaviour: if it is larger than zero the system is said to be chaotic, meaning that perturbations from a trajectory are amplified exponentially in at least one direction. At first sight no such interpretation exists for the minimal LE - it simply quantifies the direction of minimal expansion of the system. However, closer inspection reveals that a more informative interpretation is possible by inspecting the Jacobian matrix itself.

We start with the following remark: when evaluating the Jacobian around the origin in state space (zero fixpoint, i.e. $\mathbf{s} = 0$), it reduces to the weight matrix $\mathbf{W}$ of the reservoir, and its largest eigenvalue is precisely the spectral radius of the reservoir. Therefore, the eigenvalue spectrum of the Jacobian can be seen as a dynamic extension of the static eigenvalue spectrum of the weight matrix (which was the subject of previous work on dynamics in reservoirs, e.g. [9]). Moreover, the lyapunov spectrum at a single point in time, given by $\log(\text{eig}(\mathbf{J}_f^T \mathbf{J}_f))$, is equal to the log of the singular value spectrum of the Jacobian itself[2] . Following this line of reasoning, we measured the minimal singular value (SV) of the Jacobian [3] and computed its mean over time as we vary the spectral radius of the reservoir weight matrix, and the scaling factor of the input matrix. We then compared this measure with the performance on two tasks:

- The Mackey-Glass timeseries prediction. This mildly chaotic timeseries (with delay parameter $\tau = 17$) is a common benchmark and RC systems have shown very good performance on this task [6]. The RC system was trained to do one-step ahead prediction on a training timeseries of 4000 timesteps, and was then used to autonomously generate the signal by feeding its own prediction back as input into the reservoir. The performance is evaluated as the first timestep when the divergence (expressed as the absolute error) between the predicted and target signal exceeds 0.1.
- Modelling a 30th order Nonlinear AutoRegressive Moving Average (NARMA) system. Here, the input $u[k]$ to the network is a random signal sampled from a uniform distribution in $[0, .5]$, and the target output is given by $y[k+1] = 0.2y[k] + 0.04y[k](\sum_{i=0}^{29} y[k-i]) + 1.5u[k-29]u[k] + 0.001$. The performance is measured with the normalized root mean square error (NRMSE).

Figure 4 shows the mean maximal LLE, the mean minimal singular value of the Jacobian, and the score on both tasks, as the spectral radius and scaling factor of the input matrix are swept within the plausible range $[.1, 2]$ with steps of .1 (every point in the plots represents the average over twenty different reservoir instantiations). The top plots show the same measure that was introduced in [15]. This measure clearly does not capture all necessary dynamic properties of the

[2] In general, for a matrix $\mathbf{M}$, the squares of its singular values are equal to the eigenvalues of $\mathbf{M}^T\mathbf{M}$.

[3] At every 50th timestep for computational reasons, but this provides sufficient accuracy.

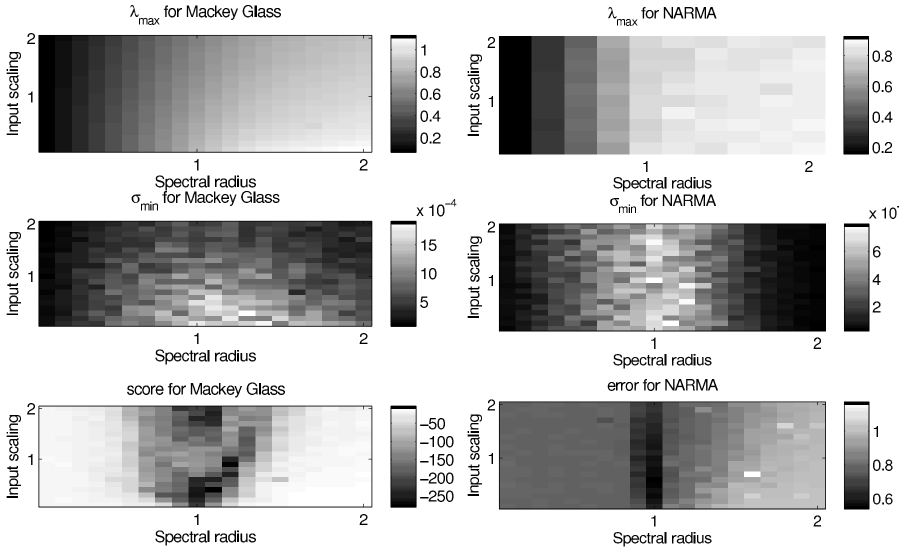

Fig. 4. The top plots show the maximal LLE, the middle plots show the minimal SV and the bottom plots show the performance for the Mackey-Glass prediction (left) and NARMA (right) task. Note that for the Mackey-Glass performance plot, higher is better while for NARMA lower is better.

reservoir, since it increases monotonically with the spectral radius, and the input scaling has hardly any influence. The middle plots on the other hand - which show the minimal SV σ_m - offer a much more nuanced image. The minimal SV σ_m varies with both the spectral radius and the input scaling - which indicates that it captures the changing dynamical properties of the reservoir as a function of the scaling parameters quite well. Moreover, the area of optimal performance (bottom plots) coincides quite nicely with the areas where σ_m is highest. Thus, σ_m is a more accurate predictor of performance than both the largest LLE and the spectral radius.

The interpretation of σ_m of the Jacobian is at first sight not trivial: it simply qualifies the minimal gain of the system in any direction in state space. However, σ_m can be written as the ratio between the norm $\|\mathbf{J}_f\|$ and the condition number $\kappa(\mathbf{J}_f)$ of the Jacobian:$\sigma_m = \frac{\|\mathbf{J}_f\|_2}{\kappa(\mathbf{J}_f)}$, since $\sigma_m^{-1} = \left\|\mathbf{J}_f^{-1}\right\|$ and $\kappa(\mathbf{J}_f) = \left\|\mathbf{J}_f^{-1}\right\| \|\mathbf{J}_f\|$ and where $\|\cdot\|_2$ denotes the l_2 norm.

This relation yields an interesting interpretation. In the field of robotics (which borrows substantially from dynamical system theory), both the condition number and the norm of the Jacobian are widely used measures for quantifying the dynamic behaviour of e.g. robotic manipulators [8]. In particular, the norm of the Jacobian is a measure of the maximal gain of the system in any direction, while the condition number is used to quantify the dexterity of the robot arm or the closeness to a singular position (where the robot looses one or more degrees of freedom due to constraints on the joints) - large condition numbers are an indication of low dexterity. When we transpose this interpretation to the reservoir, we can see that the maximization of σ_m is in fact a joint optimization of:

- the maximal gain of the system, thus ensuring good excitability and separation of the input signals in state space, and
- minimization of the condition number, which means that the dynamical system is far from singularity and has many degrees of freedom.

These two quantities are in opposition: if the gain of the reservoir is too high, the nodes will start to saturate and the expressive power of the nonlinearity decreases, which means that the reservoir is constrained to a lower-dimensional subspace of the state space. If it is too low, the reservoir does not separate the input signals enough. This trade-off is clearly present in the measure presented here.

One disadvantage of this measure is that it does not apply to linear reservoirs - a maximization of the minimal SV of the jacobian (which is then just the reservoir weight matrix) results in unbounded weights.

5 Conclusions

While RC often achieves impressive performance on many tasks, the tuning of the parameters that control the dynamics of the reservoir is still a matter of expertise and manual experimentation, which is partly due to a lack of measures for accurately quantifying the dynamics of the reservoir. We have presented a novel metric for measuring the dynamical properties of the reservoir, and have shown that it is a more accurate predictor of performance than previously published measures. Moreover, we have given an interpretation of the measure that offers more insight into the functionality of the reservoir, showing that a trade-off is made between the excitability of the reservoir and its ‘degrees of freedom’ in state space.

Reservoir Computing has originated in the field of neural networks, but has since been extended to other, more generic implementations. For these more exotic reservoirs (such as reservoirs built from nano-photonic components [13]) especially, the standard tuning parameters such as spectral radius become meaningless. The measure introduced in this contribution can fill this void and offers a useful method for tuning and quantifying the dynamics of these novel reservoir implementations.

References

1. Buehner, M., Young, P.: A tighter bound for the echo state property. IEEE Transactions on Neural Networks 17(3), 820–824 (2006)
2. Csaki, F.: State-space methods for control systems. Akademiai Kiado (1977)
3. Ganguli, S., Huh, D., Sompolinsky, H.: Memory traces in dynamical systems. Proceedings of the National Academy of Sciences 105(48), 18970 (2008)

4. Jaeger, H.: Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology (2001)
5. Jaeger, H.: Adaptive nonlinear system identification with echo state networks. In: Advances in Neural Information Processing Systems, pp. 593–600 (2003)
6. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. Science 308, 78–80 (2004)
7. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)
8. Merlet, J.P.: Jacobian, manipulability, condition number, and accuracy of parallel robots. Journal of Mechanical Design 128, 199 (2006)
9. Ozturk, M.C., Xu, D., Principe, J.C.: Analysis and design of echo state networks. Neural Computation 19, 111–138 (2006)
10. Penrose, R.: A generalized inverse for matrices, vol. 51(1955), pp. 406–413 (1955)
11. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)
12. Siewert, U., Wustlich, W.: Echo-state networks with band-pass neurons: Towards generic time-scale-independent reservoir structures. Preliminary Report (October 2007)
13. Vandoorne, K., Dierckx, W., Schrauwen, B., Verstraeten, D., Baets, R., Bienstman, P., Campenhout, J.V.: Toward optical signal processing using photonic reservoir computing. Optics Express 16(15), 11182–11192 (2008)
14. Verstraeten, D., Schrauwen, B., Stroobandt, D.: Reservoir-based techniques for speech recognition. In: Proceedings of the World Conference on Computational Intelligence, pp. 1050–1053 (2006)
15. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt., D.: A unifying comparison of reservoir computing methods. Neural Networks 20, 391–403 (2007)
16. Verstraeten, D., Souza, S.X.-d., Schrauwen, B., Suykens, J., Stroobandt, D., Vandewalle, J.: Pattern classification with CNNs as reservoirs. In: Proceedings of the International Symposium on Nonlinear Theory and its Applications (NOLTA), vol. 9 (2008)
17. White, O.L., Lee, D.D., Sompolinsky, H.: Short-term memory in orthogonal neural networks. Neural Comput. Phys. Rev. Lett. 92, 148102 (2002)
18. Wyffels, F., Schrauwen, B., Verstraeten, D., Stroobandt, D.: Band-pass reservoir computing. In: Hou, Z., Zhang, N. (eds.) Proceedings of the International Joint Conference on Neural Networks, Hong Kong, June, pp. 3203–3208 (2008)