Minimum Complexity Echo State Network

Ali Rodan, *Student Member, IEEE,* and Peter Tiňo

***Abstract*—Reservoir computing (RC) refers to a new class of state-space models with a fixed state transition structure (the *"reservoir"*) and an adaptable readout form the state space. The reservoir is supposed to be sufficiently complex so as to capture a large number of features of the input stream that can be exploited by the reservoir-to-output readout mapping. The field of RC has been growing rapidly with many successful applications. However, RC has been criticized for not being principled enough. Reservoir construction is largely driven by a series of randomized model building stages, with both researchers and practitioners having to rely on a series of trials and errors. To initialize a systematic study of the field, we concentrate on one of the most popular classes of reservoir computing methods - Echo State Network (ESN) - and ask: What is the minimal complexity of reservoir construction for obtaining competitive models and what is the memory capacity of such simplified reservoirs? On a number of widely used time series benchmarks of different origin and characteristics, as well as by conducting a theoretical analysis we show: A simple *deterministically* constructed cycle reservoir is comparable to the standard echo state network methodology. The (short term) memory capacity of linear cyclic reservoirs can be made arbitrarily close to the proved optimal value.**

***Index Terms*—Reservoir computing, Echo state networks, Simple recurrent neural networks, Memory capability, Time series prediction**

I. Introduction

RECENTLY there has been an outburst of research activity in the field of reservoir computing (RC) [1]. RC models are dynamical models for processing time series that make a conceptual separation of the temporal data processing into two parts: 1) representation of temporal structure in the input stream through a non-adaptable dynamic *"reservoir"*, and 2) a memoryless easy-to-adapt *readout* from the reservoir. For a comprehensive recent review of RC see [2]. Perhaps the simplest form of the RC model is the Echo State Network (ESN) [3]–[6]. Roughly speaking, ESN is a recurrent neural network with a non-trainable sparse recurrent part (reservoir) and a simple linear readout. Connection weights in the ESN reservoir, as well as the input weights are randomly generated. The reservoir weights are scaled so as to ensure the *"Echo State Property"* (ESP): the reservoir state is an *"echo"* of the entire input history. Typically, spectral radius of the reservoir's weight matrix W is made < 1[1]. ESN has been successfully applied in time-series prediction tasks [6], speech recognition [7], noise modeling [6], dynamic pattern classification [5], reinforcement learning [8], and in language modeling [9].

Many extensions of the classical ESN have been suggested in the literature, e.g. intrinsic plasticity [10], [11], decoupled reservoirs [12], refined training algorithms [6], leaky-integrator reservoir units [13], support vector machine [14], filter neurons with delay&sum readout [15] etc. However, there are still serious problems preventing ESN to become a widely accepted tool: **1)** There are properties of the reservoir that are poorly understood [12], **2)** specification of the reservoir and input connections require numerous trails and even luck [12], **3)** strategies to select different reservoirs for different applications have not been devised [16], **4)** imposing a constraint on spectral radius of the reservoir matrix is a weak tool to properly set the reservoir parameters [16], **5)** the random connectivity and weight structure of the reservoir is unlikely to be optimal and does not give a clear insight into the reservoir dynamics organization [16]. Indeed, it is not surprising that part of the scientific community is skeptical about ESNs being used for practical applications [17].

Typical model construction decisions that an ESN user must make include: setting the reservoir size; setting the sparsity of the reservoir and input connections; setting the ranges for random input and reservoir weights; and setting the reservoir matrix scaling parameter α. The dynamical part of the ESN responsible for input stream coding is treated as a black box which is unsatisfactory from both theoretical and empirical standpoints. First, it is difficult to put a finger on what it actually is in the reservoir's dynamical organization that makes ESN so successful. Second, the user is required to tune parameters whose function is not well understood. In this paper we would like to clarify by systematic investigation the reservoir construction, namely we show that in fact a very simple ESN organization is sufficient to obtain performances comparable to those of the classical ESN. We argue that for a variety of tasks it is sufficient to consider: **1)** a simple fixed non-random reservoir topology with full connectivity from inputs to the reservoir , **2)** a single fixed absolute weight value r for all reservoir connections and **3)** a single weight value v for input connections, with (deterministically generated) aperiodic pattern of input signs.

In contrast to the complex trial-and-error ESN construction, our approach leaves the user with only two free parameters to be set, r and v. This not only considerably simplifies the ESN construction, but also enables a more thorough theoretical analysis of the reservoir properties. The doors can be open for a wider acceptance of the ESN methodology amongst both practitioners and theoreticians working in the field of time series modeling/prediction. In addition, our simple deterministically constructed reservoir models can serve as useful baselines in future reservoir computing studies. The paper is organized as follows. Section II gives an overview of Echo state network design and training. In Section III we present our simplified reservoir topologies. Experimental results are presented in Section IV. We analyze both theoretically and empirically the short term memory capacity (MC) of our

The authors are with the School of Computer Science, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (e-mail: a.a.rodan, P.Tino@cs.bham.ac.uk).

[1] Note that this is not the necessary and sufficient condition for ESP

simple reservoir in Section V. Finally, our work is discussed and concluded in Sections VI and VII, respectively.

II. Echo State Networks

Echo state network is a recurrent discrete-time neural network with K input units, N internal (reservoir) units, and L output units. The activation of the input, internal, and output units at time step t are denoted by: $s(t) = (s_1(t), ..., s_K(t))^T$, $x(t) = (x_1(t), ..., x_N(t))^T$, and $y(t) = (y_1(t), ..., y_L(t))^T$ respectively. The connections between the input units and the internal units are given by an $N \times K$ weight matrix V, connections between the internal units are collected in an $N \times N$ weight matrix W, and connections from internal units to output units are given in $L \times N$ weight matrix U.

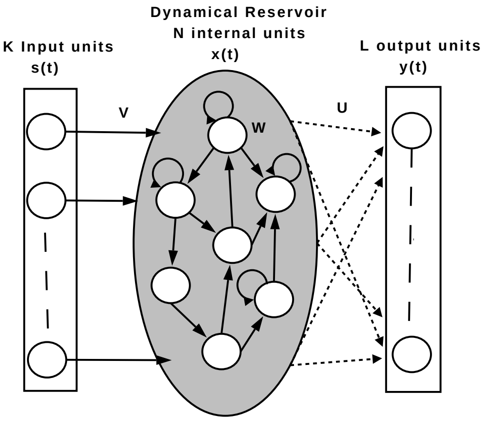

Fig. 1. Echo state network (ESN) Architecture

The internal units are updated according to[2]:

$$x(t+1) = f(Vs(t+1) + Wx(t)), \tag{1}$$

where f is the reservoir activation function (typically $tanh$ or some other sigmoidal function). The linear readout is computed as[3]:

$$y(t+1) = Ux(t+1). \tag{2}$$

Elements of W and V are fixed prior to training with random values drawn from a uniform distribution over a (typically) symmetric interval. To account for ESP, the reservoir connection matrix W is typically scaled as $W \leftarrow \alpha W/|\lambda_{max}|$, where $|\lambda_{max}|$ is the spectral radius[4] of W and $0 < \alpha < 1$ is a scaling parameter [5].

ESN memoryless readout can be trained both offline (Batch) and online by minimizing any suitable loss function. We use the Normalized Mean Square Error (NMSE) to train and evaluate the models:

$$NMSE = \frac{\langle \|\hat{y}(t) - y(t)\|^2 \rangle}{\langle \|y(t) - \langle y(t) \rangle \|^2 \rangle}, \tag{3}$$

where $\hat{y}(t)$ is the readout output, $y(t)$ is the desired output (target), $\|.\|$ denotes the Euclidean norm and $< \cdot >$ denotes the empirical mean. To train the model in off-line mode, we **1)** Initialize W with a scaling parameter $\alpha < 1$ and run the ESN on the training set; **2)** dismiss data from initial $washout$ period and collect remaining network states $x(t)$ row-wise into a matrix X[5] and **3)** calculate the readout weights using e.g. ridge regression [18]:

$$U = (X^T X + \lambda^2 I)^{-1} X^T y, \tag{4}$$

where I is the identity matrix, y a vector of the target values, and $\lambda > 0$ is a regularization factor.

III. Simple Echo state network reservoirs

To simplify the reservoir construction, we propose several easy structured topology templates and we compare them to those of the classical ESN. We consider both *linear reservoirs* that consist of neurons with identity activation function, as well as *non-linear reservoirs* consisting of neurons with the commonly used tangent hyperbolic (tanh) activation function. Linear reservoirs are fast to simulate but often lead to inferior performance when compared to non-linear ones [19].

A. Reservoir Topology

Besides the classical ESN reservoir introduced in the last section Figure. 1 , we consider the following three reservoir templates (model classes) with fixed topologies Figure. 2:

- *Delay Line Reservoir (DLR)* - composed of units organized in a line. Only elements on the lower sub-diagonal of the reservoir matrix W have non-zero values $W_{i+1,i} = r$ for $i = 1...N-1$, where r is the weight of all the feedforward connections.
- *DLR with feedback connections (DLRB)* - the same structure as DLR but each reservoir unit is also connected to the preceding neuron. Nonzero elements of W are on the lower $W_{i+1,i} = r$ and upper $W_{i,i+1} = b$ sub-diagonals, where b is the weight of all the feedback connections.
- *Simple Cycle Reservoir (SCR)* - units organized in a cycle. Nonzero elements of W are on the lower sub-diagonal $W_{i+1,i} = r$ and at the upper-right corner $W_{1,N} = r$.

B. Input Weight Structure

The input layer is fully connected to the reservoir. For ESN the input weights are (as usual) generated randomly from a uniform distribution over an interval $[-a, a]$. In case of simple reservoirs (DLR, DLRB and SCR), all input connections have the same absolute weight value $v > 0$; the sign of each input weight is determined randomly by a random draw from Bernoulli distribution of mean $1/2$ (unbiased coin). The values v and a are chosen on the validation set.

[2] There are no feedback connections from the output to the reservoir and no direct connections from the input to the output.

[3] The reservoir activation vector is extended with a fixed element accounting for the bias term.

[4] The largest among the absolute values of the eigenvalues of W.

[5] In case of direct input-output connections, the matrix X collects inputs $s(t)$ as well.

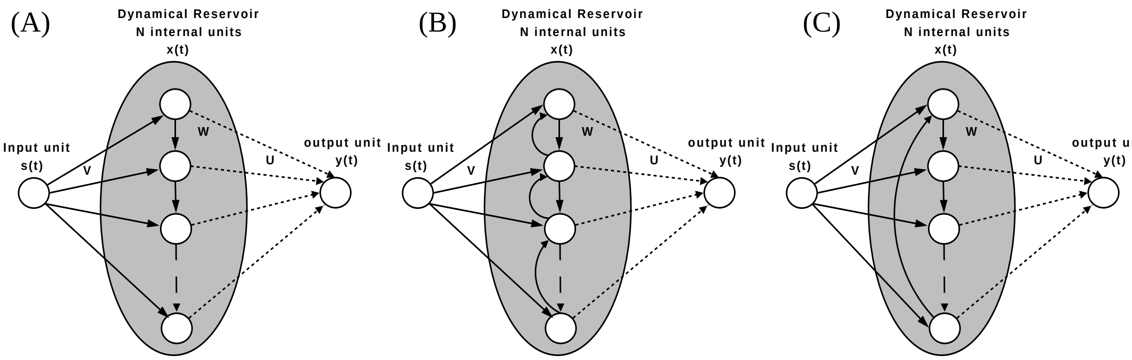

Fig. 2. (A) Delay Line Reservoir (DLR). (B) Delay Line Reservoir with feedback connections (DLRB). (C) Simple Cycle Reservoir (SCR).

IV. Experiments

A. Datasets

We use a range of timeseries covering a wide spectrum of memory structure and widely used in the ESN literature [3], [4], [6], [10], [11], [19]–[21]. For each data set, we denote the length of the training, validation and test sequences by L_{trn}, L_{val} and L_{tst}, respectively. The first L_v values from training, validation and test sequences are used as the initial washout period.

1) NARMA System: The Non-linear Auto-Regressive Moving Average (*NARMA*) system is a discrete time system. This system was introduced in [22]. The current output depends on both the input and the previous output. In general, modeling this system is difficult, due to the non-linearity and possibly long memory.

- *fixed order NARMA time series: NARMA* systems of order $O = 10, 20$ given by equations 5, and 6, respectively.

$$y(t+1) = 0.3y(t) + 0.05y(t)\sum_{i=0}^{9} y(t-i) + 1.5s(t-9)s(t) + 0.1, \tag{5}$$

$$\begin{aligned} y(t+1) = &\tanh(0.3y(t) + 0.05y(t)\sum_{i=0}^{19} y(t-i) \\ &+ 1.5s(t-19)s(t) + 0.01), \end{aligned} \tag{6}$$

where $y(t)$ is the system output at time t, $s(t)$ is the system input at time t (an i.i.d stream of values generated uniformly from an interval $[0, 0.5]$) [21], [22].

-random 10th order NARMA time series: This system is generated by:

$$y(t+1) = \tanh(\alpha y(t) + \beta y(t)\sum_{i=0}^{9} y(t-i) + \gamma s(t-9)s(t) + \varphi), \tag{7}$$

where α, β, γ and φ are assigned random values taken from $\pm 50\%$ interval around their original values in eq. (5) [21]. Since the system is not stable, we used a non-linear saturation function $tanh$ [21]. The input $s(t)$ and target data $y(t)$ are shifted by -0.5 and scaled by 2 as in [10]. The networks were trained on system identification task to output $y(t)$ based on $s(t)$, with $L_{trn} = 2000$, $L_{val} = 3000$, $L_{tst} = 3000$ and $L_v = 200$.

2) Laser Dataset: The Santa Fe Laser dataset [13] is a cross-cut through periodic to chaotic intensity pulsations of a real laser. The task is to predict the next laser activation $y(t+1)$, given the values up to time t; $L_{trn} = 2000$, $L_{val} = 3000$, $L_{tst} = 3000$ and $L_v = 200$.

3) Hénon Map: Hénon Map dataset [23] is generated by:

$$y(t) = 1 - 1.4y(t-1)^2 + 0.3y(t-2) + z(t), \tag{8}$$

where $y(t)$ is the system output at time t, $z(t)$ is a normal white noise with standard deviation of 0.05 [24]. We used $L_{trn} = 2000$, $L_{val} = 3000$, $L_{tst} = 3000$ and $L_v = 200$. The dataset is shifted by -0.5 and scaled by 2. Again, the task is to predict the next value $y(t+1)$, given the values up to time t.

4) Non-linear Communication Channel: The dataset was created as follows [6]: First, an i.i.d. sequence $d(t)$ of symbols transmitted through the channel is generated by randomly choosing values from $\{-3, -1, 1, 3\}$ (uniform distribution). Then, $d(t)$ values are used to form a sequence $q(t)$ through a linear filter

$$\begin{aligned} q(t) = &0.08d(t+2) - 0.12d(t+1) + d(t) + 0.18d(t-1) \\ &- 0.1d(t-2) + 0.09d(t-3) - 0.05d(t-4) \\ &+ 0.04d(t-5) + 0.03d(t-6) + 0.01d(t-7). \end{aligned} \tag{9}$$

Finally, a non-linear transformation is applied to $q(n)$ to produce the signal $s(n)$:

$$s(t) = q(t) + 0.0036q(t)^2 - 0.11q(t)^3. \tag{10}$$

Following [6], the input $s(t)$ signal was shifted +30. The task is to output $d(t-2)$ when $s(t)$ is presented at the network input. $L_{trn} = 2000$, $L_{val} = 3000$, $L_{tst} = 3000$ and $L_v = 200$.

5) IPIX Radar: The sequence (used in [12]) contains 2000 values with $L_{trn} = 800$, $L_{val} = 500$, $L_{tst} = 700$ and $L_v = 100$. The target signal is the sea clutter data (the radar backscatter from an ocean surface). The task was to predict $y(t+1)$ and $y(t+5)$ (1 and 5 step ahead prediction) when $y(t)$ is presented at the network input.

6) Sunspot series: The dataset (obtained from [25]) contains 3100 sunspots numbers from Jan 1749 to April 2007, where $L_{trn} = 1600$, $L_{val} = 500$, $L_{tst} = 1000$ and $L_v = 100$. The task was to predict the next value $y(t+1)$ based on the history of y up to time t.

7) Non-linear System with Observational Noise: This system was studied in [26] in the context of Bayesian Sequential State estimation. The data is generated by:

$$s(t) = 0.5s(t-1)+25\frac{s(t-1)}{1+s^2(t-1)}+8\cos(1.2(t-1))+w(t), \tag{11}$$

$$y(t) = \frac{s^2(t)}{20} + v(t), \tag{12}$$

where the initial condition is $s(0) = 0.1$; $w(t)$ and $v(t)$ are zero-mean Gaussian noise terms with variances taken from $\{1, 10\}$, i.e. $(\sigma_w^2, \sigma_v^2) \in \{1, 10\}^2$. $L_{trn} = 2000$, $L_{val} = 3000$, $L_{tst} = 3000$ and $L_v = 200$. The task was to predict the value $y(t + 5)$, given the values from $t - 5$ up to time t presented at the network input.

8) Isolated Digits: This dataset[6] is a subset of the TI46 dataset which contains 500 spoken *Isolated Digits* (zero to nine), where each digit is spoken 10 times by 5 female speakers. These 500 digits are randomly split into training ($N_{trn} = 250$) and test ($N_{tst} = 250$) sets. Because of the limited amount of data, model selection was performed using 10-fold cross-validation on the training set. The Lyon Passive Ear model [27] is used to convert the spoken digits into 86 frequency channels. Following the ESN literature using this dataset, the model performance will be evaluated using the Word Error Rate (WER), which is the number of incorrect classified words divided by the total number of presented words. The 10 output classifiers are trained to output 1 if the corresponding digit is uttered and -1 otherwise. Following [28] the temporal mean over complete sample of each spoken digit is calculated for the 10 output classifiers. The Winner-Take-All (WTA) methodology is then applied to estimate the spoken digit's identity. We use this data set to demonstrate the modeling capabilities of different reservoir models on high-dimensional (86 input channels) time series.

B. Training

We trained a classical ESN, as well as SCR, DLR, and DLRB models (with linear and tanh reservoir nodes) on the time series described above with the NMSE to be minimized. The model fitting was done using ridge regression[7], where the regularization factor λ was tuned per reservoir and per dataset on the validation set. For each model we calculate the average NMSE[8] over 10 simulation runs. Our experiments are organized along four degrees of freedom: 1) reservoir topology; 2) reservoir activation function; 3) input weight structure; 4) reservoir size.

C. Results

For each data set and each model class (ESN, DLR, DLRB, SCR) we picked on the validation set a model representative to be evaluated on the test set. Ten randomizations of each model representative were then tested on the test set. For the DLR, DLRB and SCR architectures the model representatives are defined by the input weight value v and the reservoir weight r (for DLRB network we also need to specify the value b of the feedback connection). The randomization was performed solely by randomly generating the signs for individual input weights[9], the reservoir itself was intact. For the ESN architecture, the model representative is specified by input weight scaling, reservoir sparsity and spectral radius of the weight matrix. For each model setting (e.g. for ESN - input weight scaling, reservoir sparsity and spectral radius), we generate 10 randomized models and calculate their average validation set performance. The best performing model setting on the validation set is then used to generate another set of 10 randomized models that are fitted on the training set and subsequently tested on the test set.

For some data sets the performance of linear reservoirs was consistently inferior to that of non-linear ones. Due to space limitations, in such cases the performance of linear reservoirs is not reported. Linear reservoirs are explicitely mentioned only when they achieve competitive (or even better) results than their non-linear counterparts.

Figures 3, 4 and 5(A) show the average test set NMSE (across ten randomizations) achieved by the selected model representatives. Figure 3 presents results for the four model classes using non-linear reservoir on the *laser*, *Hénon Map* and *Non-linear Communication Channel* datasets. On those time series, the test NMSE for linear reservoirs were of an order of magnitude worse than the NMSE achieved by the non-linear ones. While the ESN architecture slightly outperforms the simplified reservoirs on the *laser* and *Hénon Map* time series, for the *Non-linear Communication Channel* the best performing architecture is the simple delay line network (DLR). The SCR reservoir is consistently the second-best performing architecture. Even though the differences between NMSE are in most cases statistically significant, from the practical point of view, they are minute. Note that the *Non-linear Communication Channel* can be modeled rather well with a simple Markovian delay line reservoir and no complex ESN reservoir structure is needed. Non-linearity in the reservoir activation and the reservoir size seem to be two important factors for successful learning on those three datasets.

Figure 4 presents results for the four model classes on the three *NARMA* time series, namely fixed *NARMA* of order 10, 20 and random *NARMA* of order 10. The performance of linear reservoirs do not improve with increasing reservoir size. Interestingly, within the studied reservoir range (50-200), linear reservoirs beat the non-linear ones on *20-th order*

[6] obtained from http://snn.elis.ugent.be/rctoolbox

[7] We also tried other forms of offline and online readout training, such as wiener-hopf methodology (e.g. [16]), pseudoinverse solution (e.g [3]), singular value decomposition (e.g. [20]) and Recursive Least square (e.g. [21]). Ridge regression lead to the best results. We are thankful to the anonymous referee for suggesting the inclusion of ridge regression in our repertoire of batch training methods.

[8] word error Rate (WER) in the case of *Isolated Digits* dataset

[9] Strictly speaking we randomly generated the signs for input weights and input biases. However, as usual in the neural network literature, the bias terms can be represented as input weights from a constant input +1.

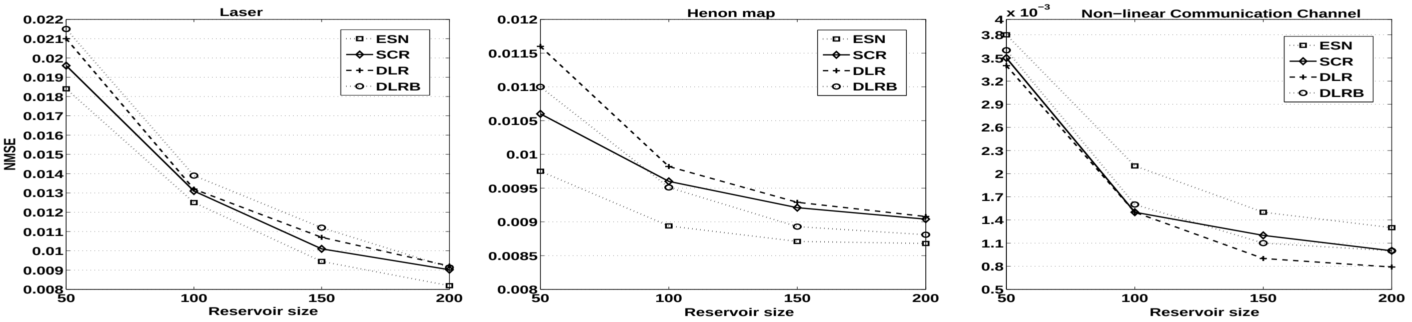

Fig. 3. Test set performance of ESN, SCR, DLR, and DLRB topologies with $tanh$ transfer function on the *laser*, *Hénon Map*, and *Non-linear Communication Channel* datasets.

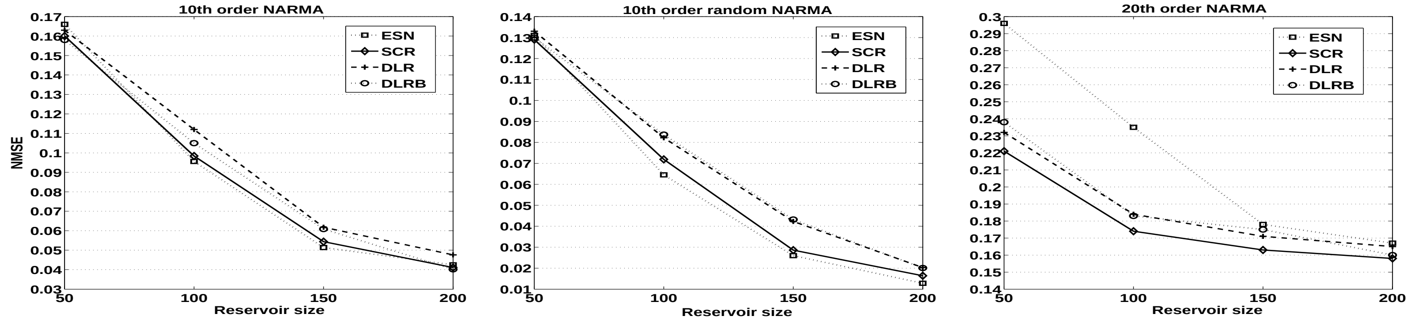

Fig. 4. Test set performance of ESN, SCR, DLR, and DLRB topologies with $tanh$ transfer function on *10th-order, random 10th-order* and *20th-order NARMA* datasets.

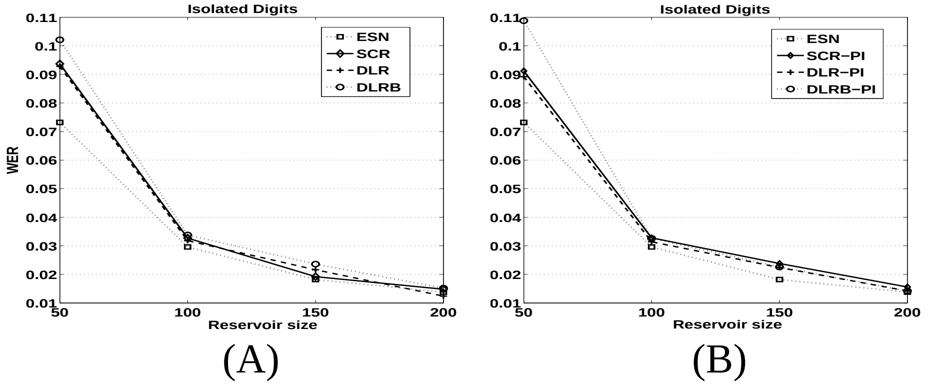

Fig. 5. Test set performance of ESN, SCR, DLR, and DLRB topologies on the *Isolated Digits* (speech recognition) task using two ways of generating input connection sign patterns; using initial digits of π (A), and random generation (i.i.d. Bernoulli distribution with mean 1/2) (B). Reservoir nodes with $tanh$ transfer function f were used.

TABLE I

Mean NMSE for ESN, DLR, DLRB, and SCR across 10 simulation runs (standard deviations in parenthesis) and SCR topologies with deterministic input sign generation on the *IPIX Radar* and *Sunspot* series. The results are reported for prediction horizon ν and models with non-linear reservoirs of size $N = 80$ (*IPIX Radar*) and linear reservoirs with $N = 200$ nodes (*Sunspot series*).

Dataset	ν	ESN	DLR	DLRB	SCR	SCR-PI	SCR-EX	SCR-Log
IPIX Radar	1	0.00115 (2.48E-05)	0.00112 (2.03E-05)	0.00110 (2.74E-05)	0.00109 (1.59E-05)	0.00109	0.00109	**0.00108**
	5	0.0301 (8.11E-04)	0.0293 (3.50E-04)	0.0296 (5.63E-04)	**0.0291** (3.20E-04)	0.0299	0.0299	0.0297
Sunspot	1	0.1042 (8.33E-5)	**0.1039** (9.19E-05)	0.1040 (7.68E-05)	**0.1039** (5.91E-05)	0.1063	0.1065	0.1059

TABLE II
NMSE for ESN, DLR, DLRB, and SCR across 10 simulation runs (standard deviations in parenthesis) and SCR topologies with deterministic input sign generation on the *Non-linear System with Observational Noise* data set. Reservoirs had $N = 100$ internal nodes with $tanh$ transfer function f.

var w	var v	ESN	DLR	DLRB	SCR	SCR-PI	SCR-EX	SCR-Log
1	1	0.4910 (0.0208)	0.4959 (0.0202)	0.4998 (0.0210)	**0.4867** (0.0201)	0.5011	0.5094	0.5087
10	1	0.7815 (0.00873)	0.7782 (0.00822)	0.7797 (0.00631)	**0.7757** (0.00582)	0.7910	0.7902	0.7940
1	10	0.7940 (0.0121)	0.7671 (0.00945)	0.7789 (0.00732)	0.7655 (0.00548)	0.7671	**0.7612**	0.7615
10	10	0.9243 (0.00931)	0.9047 (0.00863)	0.9112 (0.00918)	0.9034 (0.00722)	0.8986	0.8969	**0.8965**

NARMA[10]. For all *NARMA* series, the SCR network is either the best performing architecture or is not worse than the best performing architecture in a statistically significant manner. Note that *NARMA* time series constitute one of the most important and widely used benchmark datasets used in the echo state network literature (e.g. [3], [4], [6], [10], [11], [19]–[21]).

The results for the high-dimensional data set *Isolated Digits* are presented in figure 5(A). Except for the reservoir size 50, the performances of all studied reservoir models are statistically the same (see table IV in [Appendix A]). When compared to ESN, the simplified reservoir models seem to work equally well on this high dimensional input series.

For *IPIX Radar*, *Sunspot Series* and *Non-linear System with Observational Noise* the results are presented in tables I and II, respectively. On these data sets, the ESN performance did not always monotonically improve with the increasing reservoir size. That is why for each data set we determined the best performing ESN reservoir size on the validation set ($N = 80$, $N = 200$, $N = 100$ for *IPIX Radar*, *Sunspot* Series and *Non-linear System with Observational Noise*, respectively). The performance of the other model classes (DLR, DLRB and SCR) with those reservoir sizes was then compared to that of ESN. In line with most RC studies using the *Sunspot* data set (e.g. [29]), we found that linear reservoirs were on par[11] with the non-linear ones. For all three data sets, the SCR architecture perform slightly better than standard ESN, even though the differences are in most cases not statistically significant.

Ganguli, Huh and Sompolinsky [30] quantified and theoretically analyzed memory capacity of non-autonomous linear dynamical systems (corrupted by a Gaussian state noise) using Fisher information between the state distributions at distant times. They found out that the optimal Fisher memory is achieved for so called non-normal networks with DLR or DLRB topologies and derived the optimal input weight vector for those linear reservoir architectures. We tried setting the input weights to the theoretically derived values, but the performance did not improve over our simple strategy of randomly picked signs of input weights followed by model selection on the validation set. Of course, the optimal input weight considerations of [30] hold for linear reservoir models only. Furthermore, according to [30], the linear SCR belongs to the class of so called normal networks which are shown to be inferior to the non-normal ones. Interestingly enough, in our experiments, the performance of linear SCR was not worse than that of non-normal networks.

D. Further Simplifications of Input Weight Structure

The only random element of the SCR architecture is the distribution of the input weight signs. We found out that any attempt to impose a regular pattern on the input weight signs (e.g. a periodic structure of the form $+-+-...$, or $+--+--...$ etc.) lead to performance deterioration. Interestingly enough, it appears to be sufficient to relate the sign pattern to a single *deterministically* generated aperiodic sequence. Any simple pseudo-random generation of signs with a fixed seed is fine. Such sign patterns worked universally well across all benchmark data sets used in this study. For demonstration, we generated the universal input sign patterns in two ways:

1) the input signs are determined from decimal expansion $d_0.d_1d_2d_3...$ of irrational numbers (in our case π (**PI**) and e (**EX**)). The first N decimal digits $d_1, d_2, ..., d_N$ are thresholded at 4.5, e.g. if $0 \leq d_n \leq 4$ and $5 \leq d_n \leq 9$, then the n-th input connection sign (linking the input to the n-th reservoir unit) will be $-$ and $+$, respectively,
2) (**Log**) - the input signs are determined by the first N iterates in binary symbolic dynamics of the logistic map $f(x) = 4x(1-x)$ in a chaotic regime (initial condition was 0.33, generating partition for symbolic dynamics with cut-value at 1/2).

The results shown in figures 6 (*NARMA*, *laser*, *Hénon Map* and *Non-linear Communication Channel* data sets), 5(B) (*Isolated Digits*), and tables I and II (*IPIX Radar*, *Sunspot*, and *Non-linear System with Observational Noise*), indicate that comparable performances of our SCR topology can be obtained without any stochasticity in the input weight generation by consistent use of the same sign generating algorithm across a variety of data sets. Detailed results are presented in table V [Appendix A].

We tried to use these simple deterministic input sign generation strategy for the other simplified reservoir models (DLR and DLRB). The results were consistent with our findings for the SCR. We also tried to simplify the input weight structure by connecting the input to a single reservoir unit only. However, this simplification either did not improve, or deteriorated the model performance.

[10] The situation changes for larger reservoir sizes. For example, non-linear ESN and SCR reservoirs of size 800 lead to the average NMSE of 0.0468 (std 0.0087) and 0.0926 (std 0.0039), respectively.

[11] and sometimes better (within the range of reservoir sizes considered in our experiments)

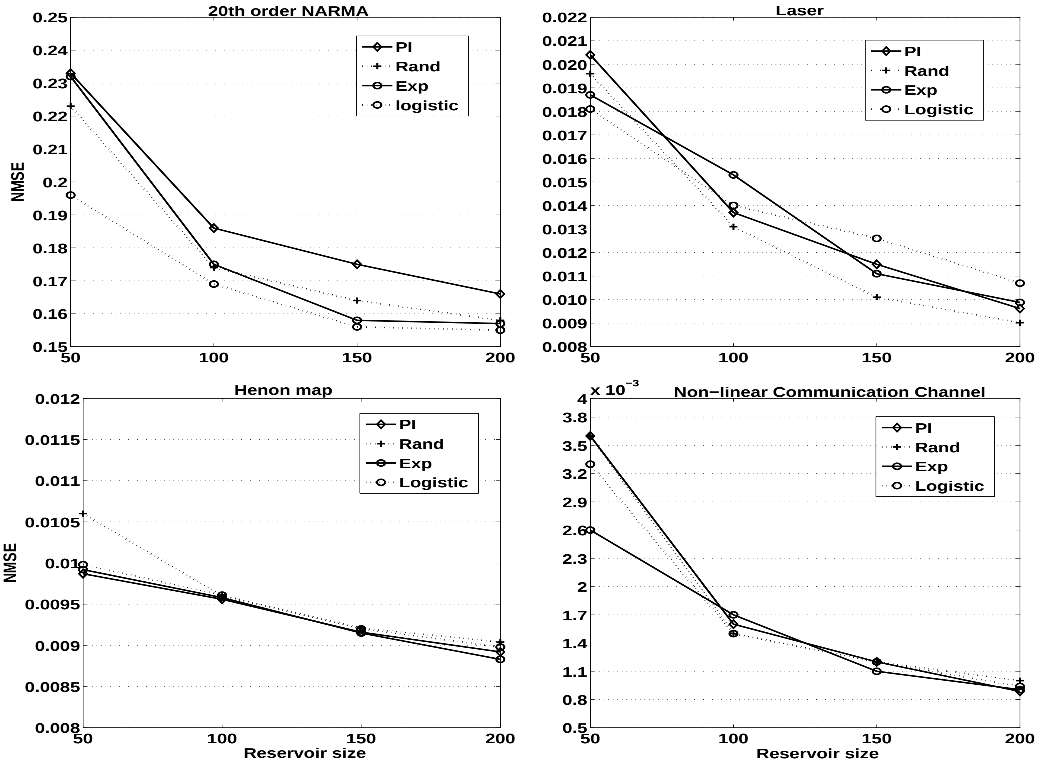

Fig. 6. Test set performance of SCR topology using four different ways of generating pseudo-randomized sign patterns; using initial digits of π , and Exp ; logistic map trajectory, and random generation (i.i.d. Bernoulli distribution with mean 1/2). The result are reported for *20th NARMA*, *laser*, *Hénon Map*, and *Non-linear Communication Channel* datasets. Reservoir nodes with $tanh$ transfer function f were used.

E. Sensitivity Analysis

We tested sensitivity of the model performance on 5-step ahead prediction with respect to variations in the (construction) parameters[12]. The reservoir size is $N = 100$ for 10th order *NARMA* data set. In the case of ESN we varied the input scaling, as well as the spectral radius and connectivity of the reservoir matrix. In figure 7(A), we show how the performance depends on the spectral radius and connectivity of the reservoir matrix. The input scaling is kept fixed at the optimal value determined on the validation set. Performance variation with respect to changes in input scaling (while connectivity and spectral radius are kept fixed at their optimal values) are reported in table III. For the SCR and DLR models figures 7(C,D), illustrate the performance sensitivity with respect to changes in the only two free parameters - the input and reservoir weights v and r, respectively. In the case of DLRB model, figures 7(B), present the performance sensitivity with respect to changes in the reservoir weights r and b, while keeping the input weight fixed to the optimal value[13].

We performed the same analysis on *Laser* and *IPIX Radar* data sets and obtained similar stability patterns. In general, all the studied reservoir models show robustness with respect to small (construction) parameter fluctuations around the optimal parameter setting.

TABLE III
BEST CONNECTIVITY AND SPECTRAL RADIUS FOR ESN WITH DIFFERENT INPUT SCALING FOR *10th order NARMA* DATASET.

Data set	Inp	Con	Spec	NMSE
10th order *NARMA*	0.05	0.18	0.85	0.1387 (0.0101)
	0.1	0.18	0.85	0.1075 (0.0093)
	0.5	0.18	0.85	0.2315 (0.0239)
	1	0.18	0.85	0.6072 (0.0459)

V. SHORT TERM MEMORY CAPACITY OF SCR ARCHITECTURE

In his report [4] Jaeger quantified the inherent capacity of recurrent network architectures to represent past events through a measure correlating the past events in an i.i.d. input stream with the network output. In particular, assume that the network is driven by a univariate stationary input signal $s(t)$. For a given delay k, we consider the network with optimal parameters for the task of outputting $s(t-k)$ after seeing the input stream $...s(t-1)s(t)$ up to time t. The goodness of fit is measured in terms of the squared correlation coefficient between the desired output (input signal delayed by k time steps) and the observed network output $y(t)$:

$$MC_k = \frac{Cov^2(s(t-k), y(t))}{Var(s(t))\ Var(y(t))}, \tag{13}$$

where Cov denotes the covariance and Var the variance operators. The short term memory (STM) capacity is then

[12] We are thankful to the anonymous reviewer for making the suggestion

[13] Note that figures 7(A-C/D) are not directly comparable since the model parameters that get varied are different for each model (e.g. connectivity and spectral radius for ESN vs. input and reservoir weights for SCR). In this sense, only figures 7(C) and (D) can be compared directly.

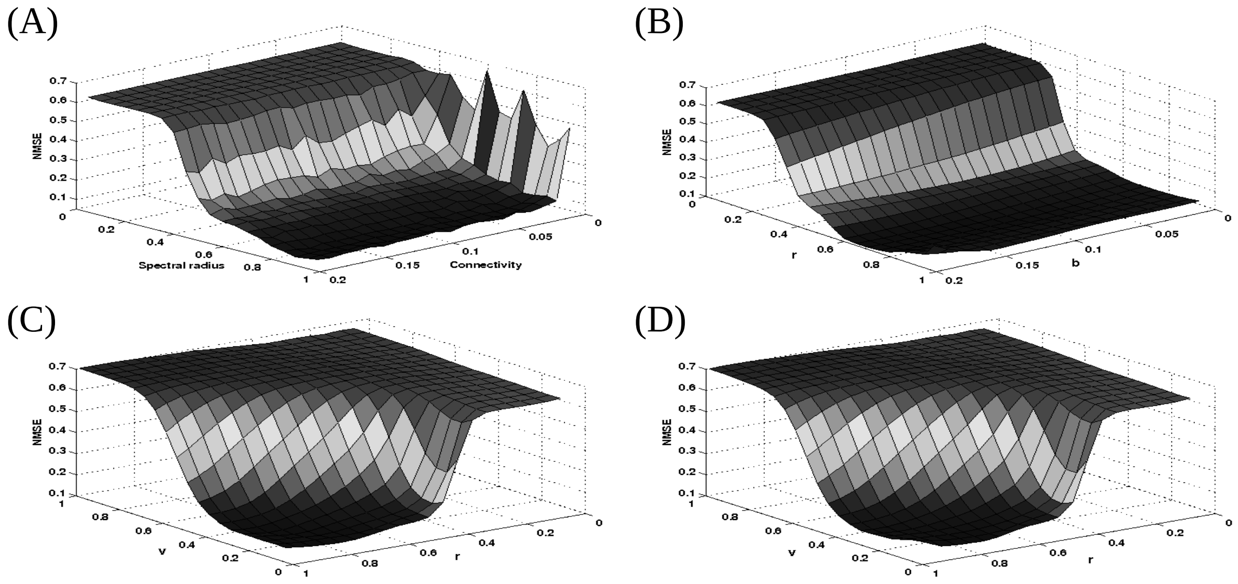

Fig. 7. Sensitivity of ESN (A), DLRB (B), DLR (C), and SCR (D) topologies on the *10th order NARMA* dataset. The input sign patterns for SCR, DLR, and DLRB non-linear reservoirs were generated using initial digits of π.

given by [4]:

$$MC = \sum_{k=1}^{\infty} MC_k. \qquad (14)$$

Jaeger [4] proved that for *any* recurrent neural network with N recurrent neurons, under the assumption of i.i.d. input stream, the STM capacity cannot exceed N. We prove (under the assumption of zero-mean i.i.d. input stream) that the STM capacity of linear SCR architecture with N reservoir units can be made arbitrarily close to N.

Since there is a single input (univariate time series), the input matrix V is an N-dimensional vector $V = (V_1, V_2, ..., V_N)^T$. Consider a vector rotation operator rot_1 that cyclically rotates vectors by 1 place to the right, e.g. $\mathrm{rot}_1(V) = (V_N, V_1, V_2, ..., V_{N-1})^T$. For $k \geq 1$, the k-fold application of rot_1 is denoted by rot_k. The $N \times N$ matrix with k-th column equal to $\mathrm{rot}_k(V)$ is denoted by Ω, e.g. $\Omega = (\mathrm{rot}_1(V), \mathrm{rot}_2(V), ..., \mathrm{rot}_N(V))$.

Theorem 1: Consider a linear SCR network with reservoir weight $0 < r < 1$ and an input weight vector V such that the matrix Ω is regular. Then the SCR network memory capacity is equal to

$$MC = N - (1 - r^{2N}).$$

The proof can be found in [Appendix B, C].

We empirically evaluated the short-term memory capacity (MC) of ESN and our three simplified topologies. The networks were trained to memorize the inputs delayed by $k = 1, 2, ..., 40$. We used one input node, 20 linear reservoir nodes, and 40 output nodes (one for each k). The input consisted of random values sampled from a uniform distribution in the range [-0.5, 0.5]. The input weights for ESN and our simplified topologies have the same absolute value 0.5 with randomly selected signs. The elements of the recurrent weight matrix are set to 0 (80% of weights), 0.47 (10% of weights), or -0.47 (10% of weights), with 0.2 reservoir weights connection fraction and spectral radius $\lambda = 0.9$ [16]. DLR and SCR weight r was fixed and set to the value $r = 0.5$. For DLRB $r = 0.5$ and $b = 0.05$. The output weights were computed using pseudo-inverse solution. The empirically determined MC values for ESN, DLR, DLRB and SCR models were (averaged over 10 simulation runs, standard dev. in parenthesis) 18.25 (1.46), 19.44 (0.89), 18.42 (0.96) and 19.48 (1.29), respectively. Note that the empirical MC values for linear SCR are in good agreement with the theoretical value of $20 - (1 - 0.5^{40}) \approx 19$.

VI. Discussion

A large number of models designed for time series processing, forecasting or modeling follows a *state-space formulation*. At each time step t, all *'relevant'* information in the driving stream processed by the model up to time t is represented in the form of a *state* (at time t). The model output depends on the past values of the driving series and is implemented as a function of the state - the so-called *read-out* function. The state space can take many different forms, e.g. a finite set, a countably infinite set, an interval etc. A crucial aspect of state-space model formulations is an imposition that the state at time $t + 1$ can be determined in a recursive manner from the state at time t and the current element in the driving series (*state transition* function). Depending on the application domain, numerous variations on the state space structure, as well as the state-transition/readout function formulations have been proposed.

One direction of research into a data-driven state space model construction imposes a state space structure (e.g. an N-dimensional interval) and a semi-parametric formulation of both the state-transition and readout functions. The parameter fitting is then driven by a cost functional $\mathcal{E}$ measuring the appropriateness of alternative parameter settings for the given task. Recurrent neural networks are examples of this type of approach [22]. If $\mathcal{E}$ is differentiable, one can employ the gradient of $\mathcal{E}$ in the parameter fitting process. However, there is a well known problem associated with parameter fitting in the state-transition function [31]: briefly, in order to 'latch' an important piece of past information for the future use, the state-transition dynamics should have an attractive set. In the neighborhood of such a set the derivatives vanish and hence cannot be propagated through time in order to reliably bifurcate into a useful latching set.

A class of approaches referred to as *reservoir computing* try to avoid this problem by fixing the state-transition function - only the readout is fitted to the data [2], [32]. The state space with the associated state transition structure is called the *reservoir*. The reservoir is supposed to be sufficiently complex so as to capture a large number of features of the input stream that can potentially be exploited by the readout.

The reservoir computing models differ in how the fixed reservoir is constructed and what form the readout takes. For example, *echo state networks* (ESN) [3] typically have a linear readout and a reservoir formed by a fixed recurrent neural network type dynamics. *Liquid state machines* (LSM) [33] also mostly have a linear readout and the reservoirs are driven by the dynamics of a set of coupled spiking neuron models. *Fractal prediction machines* (FPM) [34] have been suggested for processing symbolic sequences. Their reservoir dynamics is driven by fixed affine state transitions over an N-dimensional interval. The readout is constructed as a collection of multinomial distributions over next symbols. Many other (sometimes quite exotic) reservoir formulations have been suggested (e.g. [11], [35]–[37]).

The field of reservoir computing has been growing rapidly with dedicated special sessions at conferences and special issues of journals [38]. Reservoir computing has been successfully applied in many practical applications [3]–[6], [9], [39]. However, reservoir computing is sometimes criticized for not being principled enough [17]. There have been several attempts to address the question of what exactly is a 'good' reservoir for a given application [16], [40], but no coherent theory has yet emerged. The largely black box character of reservoirs prevents us from performing a deeper theoretical investigation of the dynamical properties of successful reservoirs. Reservoir construction is often driven by a series of (more-or-less) randomized model building stages, with both the researchers and practitioners having to rely on a series of trials and errors. Sometimes reservoirs have been evolved in a costly and difficult to analyze evolutionary computation setting [8], [14], [41], [42].

In an attempt to initialize a systematic study of the field, we have concentrated on three research questions: **1)** What is the minimal complexity of the reservoir topology and parametrization so that performance levels comparable to those of standard reservoir computing models, such as ESN, can be recovered? **2)** What degree of randomness (if any) is needed to construct competitive reservoirs? **3)** If simple competitive reservoirs constructed in a completely deterministic manner exist, how do they compare in terms of memory capacity with established models such as recurrent neural networks?

On a number of widely used time series benchmarks of different origin and characteristics, as well as by conducting a theoretical analysis we have shown: **1)** A very simple cycle topology of reservoir is often sufficient for obtaining performances comparable to those of ESN. Except for the *NARMA* datasets, non-linear reservoirs were needed. **2)** Competitive reservoirs can be constructed in a completely deterministic manner: The reservoir connections all have the same weight value. The input connections have the same absolute value with sign distribution following one of the universal deterministic aperiodic patterns. **3)** The memory capacity of linear cyclic reservoirs with a single reservoir weight value r can be made to differ arbitrarily close from the proved optimal value of N, where N is the reservoir size. In particular, given an arbitrarily small $\epsilon \in (0, 1)$, for

$$r = (1 - \epsilon)^{\frac{1}{2N}},$$

the memory capacity of the cyclic reservoir is $N - \epsilon$.

Even though the theoretical analysis of the SCR has been done for the linear reservoir case, the requirement that all cyclic rotations of the input vector need to be linearly independent seems to apply to the non-linear case as well. Indeed, under the restriction that all input connections have the same absolute weight value, the linear independence condition translates to the requirement that the input sign vector follows an aperiodic pattern. Of course, from this point of view, a simple standard basis pattern (+1,-1,-1,...,-1) is sufficient. Interestingly enough, we found out that the best performance levels were obtained when the input sign pattern contained roughly equal number of positive and negative signs. At the moment we have no satisfactory explanation for this phenomenon and we leave it as an open question for future research.

Jaeger argues [4] that if the vectors $W^i V$, $i = 1, 2, ..., N$, are linearly independent, then the memory capacity MC of linear reservoir with N units is N. Note that for the SCR reservoir

$$\mathrm{rot}_k(V) = \frac{W^k V}{r^k}, \quad k = 1, 2, ..., N,$$

and so the condition that $W^i V$, $i = 1, 2, ..., N$, are linearly independent directly translates into the requirement that the matrix Ω is regular. As $r \to 1$, the MC of SCR indeed approaches the optimal memory capacity N. According to Theorem 1, the MC measure depends on the spectral radius of W (in our case, r). Interestingly enough, in the verification experiments of [4] with a reservoir of size $N = 20$ and reservoir matrix of spectral radius 0.98, the empirically obtained MC value was 19.2. Jaeger commented that a conclusive analysis of the disproportion between the theoretical and empirical values of MC was not possible, however, he suggested that the disproportion may be due to numerical errors, as the condition number of the reservoir weight matrix W was about 50. Using our result, $MC = N - (1 - r^{2N})$ with $N = 20$ and $r = 0.98$ yields $MC = 19.4$. It is certainly true that for smaller spectral radius values, the empirically estimated MC values of linear reservoirs decrease, as verified in several studies (e.g. [19]), and this may indeed be at least partially due to numerical problems in calculating higher powers of W. Moreover, empirical estimates of MC tend to fluctuate rather strongly, depending on the actual i.i.d. driving stream used in the estimation (see e.g. [16]). Even though Theorem 1 suggests that the spectral radius of W should have an influence on the MC value for linear reservoirs, its influence becomes negligible for large reservoirs, since (provided Ω is regular) the MC of SCR is provably bounded within the interval $(N-1, N)$.

Memory capacity MC of a reservoir is a representative member from the class of reservoir measures that quantify the amount of information that can be preserved in the reservoir about the past. For example, Ganguli, Huh and Sompolinsky [30] proposed a different (but related) quantification of memory capacity for linear reservoirs (corrupted by a Gaussian state noise). They evaluated the Fisher information between the reservoir activation distributions at distant times. Their analysis shows that the optimal Fisher memory is achieved for the reservoir topologies corresponding e.g. to our DLR or DLRB reservoir organizations. Based on the Fisher memory theory, the optimal input weight vector for those linear reservoir architectures was derived. Interestingly enough, when we tried setting the input weights to the theoretically derived values, the performance in our experiments did not improve over our simple strategy for obtaining the input weights. While in the setting of [30], the memory measure does not depend on the distribution of the source generating the input stream, the MC measure of [4] is heavily dependent on the generating source. For the case of i.i.d. source (where no dependencies between the time series elements can be exploited by the reservoir) the memory capacity $MC = N-1$ can be achieved by a very simple model: DLR reservoir with unit weight $r = 1$, one input connection with weight 1 connecting the input with the 1st reservoir unit, and for $k = 1, 2, ..., N-1$ one output connection of weight 1 connecting the $(k+1)$-th reservoir unit with the output. The linear SCR, on the other hand, can get arbitrarily close to the theoretical limit $MC = N$. In cases of non i.i.d. sources, the temporal dependencies in the input stream can increase the memory capacity beyond the reservoir size N [4]. The simple nature of our SRC reservoir can enable a systematic study of the MC measure for different kinds of input stream sources and this is a matter for our future research.

Compared with traditional ESN, recent extensions and reformulations of reservoir models often achieved improved performances [11], [12], [36], at the price of even less transparent models and less interpretable dynamical organization. We stress that the main purpose of this study is not a construction of yet another reservoir model achieving an (incremental or more substantial) improvement over the competitors on the benchmark data sets. Instead, we would like to propose as simplified as possible reservoir construction, without any stochastic component, that while competitive with *standard* ESN, yields transparent models, more amenable to theoretical analysis than the reservoir models proposed in the literature so far. Such reservoir models can potentially help us to answer the question just what is it in the organization of the non-autonomous reservoir dynamics that leads to often impressive performances of reservoir computation. Our simple deterministic SCR model can be used as a a useful baseline in future reservoir computation studies. It is the level of improvement over the SCR baseline that has a potential to truly unveil the performance gains achieved by the more (and sometimes much more) complex model constructions.

VII. Conclusion

Reservoir computing learning machines are state-space models with fixed state transition structure (the ‘reservoir’) and an adaptable readout form the state space. The reservoir is supposed to be sufficiently complex so as to capture a large number of features of the input stream that can be exploited by the reservoir-to-output readout mapping. Even though the field of reservoir computing has been growing rapidly with many successful applications, both researchers and practitioners have to rely on a series of trials and errors.

To initialize a systematic study of the field, we have concentrated on three research issues:

1) What is the minimal complexity of the reservoir topology and parametrization so that performance levels comparable to those of standard reservoir computing models, such as ESN, can be recovered?
2) What degree of randomness (if any) is needed to construct competitive reservoirs?
3) If simple competitive reservoirs constructed in a completely deterministic manner exist, how do they compare in terms of memory capacity with established models such as recurrent neural networks?

On a number of widely used time series benchmarks of different origin and characteristics, as well as by conducting a theoretical analysis we have shown:

1) A simple cycle reservoir topology is often sufficient for obtaining performances comparable to those of ESN.
2) Competitive reservoirs can be constructed in a completely deterministic manner.
3) The memory capacity of simple linear cyclic reservoirs can be made to be arbitrarily close to the proved optimal MC value.

References

[1] M. Lukosevicius and H. Jaeger, “Overview of reservoir recipes,” School of Engineering and Science, Jacobs University, Technical Report No. 11, 2007.

[2] ——, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3(3), pp. 127–149, 2009.

[3] H. Jaeger, “The ”echo state” approach to analysing and training recurrent neural networks,” German National Research Center for Information Technology, Technical Report GMD report 148, 2001.

[4] ——, “Short term memory in echo state networks,” German National Research Center for Information Technology, Technical Report GMD report 152, 2002.

[5] ——, “A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the ”echo state network” approach,” German National Research Center for Information Technology, Technical Report GMD report 159, 2002.

[6] H. Jaeger and H. Hass, “Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication,” *Science*, vol. 304, pp. 78–80, 2004.

[7] M. Skowronski and J. Harris, “Minimum mean squared error time series classification using an echo state network prediction model,” in *IEEE International Symposium on Circuits and Systems, Island of Kos, Greece, pp. 3153-3156*, 2006.

[8] K. Bush and C. Anderson, “Modeling reward functions for incomplete state representations via echo state networks,” in *Proceedings of the International Joint Conference on Neural Networks, Montreal, Quebec*, July 2005.

[9] M. H. Tong, A. Bicket, E. Christiansen, and G. Cottrell, “Learning grammatical structure with echo state network,” *Neural Networks*, vol. 20, pp. 424–432, 2007.

[10] B. Schrauwen, M. Wardermann, D. Verstraeten, J. Steil, and D. Stroobandt, “Improving reservoirs using intrinsic plasticity,” *Neurocomputing*, vol. 71(7-9), pp. 1159–1171, 2008.

[11] J. Steil, “Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning,” *Neural Networks*, vol. 20, pp. 353–364, 2007.

[12] Y. Xue, L. Yang, and S. Haykin, “Decoupled echo state networks with lateral inhibition,” *Neural Networks*, vol. 20, pp. 365–376, 2007.

[13] H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert, “Optimisation and applications of echo state networks with leaky-integrator neurons,” *Neural Networks*, vol. 20(3), pp. 335–352, 2007.

[14] J. Schmidhuber, D. Wierstra, M. Gagliolo, and F. Gomez, “Training recurrent networks by evolino,” *Neural Computation*, vol. 19, pp. 757–779, 2007.

[15] H. G. and H. Hauser, “Echo state networks with filter neurons and a delay and sum readout,” *Neural Networks*, vol. 32(2), pp. 244–256, 2009.

[16] M. C. Ozturk, D. Xu, and J. Principe, “Analysis and design of echo state network,” *Neural Computation*, vol. 19(1), pp. 111–138, 2007.

[17] D. Prokhorov, “Echo state networks: appeal and challenges,” in *In Proc. of International Joint Conference on Neural Networks (pp. 1463-1466). Montreal, Canada.*, 2005.

[18] F. Wyffels, B. Schrauwen, , and D. Stroobandt, “Stable output feedback in reservoir computing using ridge regression,” in *Proceedings of the 18th international conference on Artificial Neural Networks, pp.808-817, Lecture Notes in Computer Science, LNCS 5163, Springer-Verlag*, 2008.

[19] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods,” *Neural Networks*, vol. 20, pp. 391–403, 2007.

[20] M. Cernansky and P. Tino, “Predictive modelling with echo state networks,” in *Proceedings of the 18th international conference on Artificial Neural Networks, (eds) V. Kurkova, R. Neruda, J. Koutnik. pp. 778-787, Lecture Notes in Computer Science, LNCS 5163, Springer-Verlag*, 2008.

[21] H. Jaeger, “Adaptive nonlinear systems identification with echo state network,” *Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA.*, vol. 15, pp. 593–600, 2003.

[22] A. F. Atiya and A. G. Parlos, “New results on recurrent network training: Unifying the algorithms and accelerating convergence,” *IEEE Transactions on Neural Networks*, vol. 11, pp. 697–709, 2000.

[23] M. Henon, “A two-dimensional mapping with a strange attractor,” *Comm. Math. Phys.*, vol. 50, pp. 69–77, 1976.

[24] M. Slutzky, P. Cvitanovic, and D. Mogul, “Manipulating epileptiform bursting in the rat hippocampus using chaos control and adaptive techniques,” *IEEE transactions on bio-medical engineering*, vol. 50(5), pp. 559–570, 2003.

[25] *Sunspot numbers.* National Geophysical Data Center (NGDC), 2007.

[26] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in *IEE Proc.-Radar, Sonar Navig., vol. 140, pp. 107-113,*, Apr. 1993.

[27] R. Lyon, “A computational model of filtering, detection and compression in the cochlea,” in *In Proceedings of the IEEE ICASSP, pages 1282-1285*, May 1982.

[28] B. Schrauwen, J. Defour, D. Verstraeten, and J. Van Campenhout, “The introduction of time-scales in reservoir computing, applied to isolated digits recognition.” in *In Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN 2007), volume 4668 of LNCS, pages 471-479. Springer*, 2007.

[29] F. Schwenker and A. Labib, “Echo state networks and neural network ensembles to predict sunspots activity,” in *ESANN 2009 proceedings, European Symposium on Artificial Neural Networks -Advances in Computational Intelligence and Learning, Bruges (Belgium)*, 2009.

[30] S. Ganguli, D. Huh, and H. Sompolinsky, “Memory traces in dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 105, pp. 18 970–18 975, 2008.

[31] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5(2), pp. 157–166, 1994.

[32] B. Schrauwen, D. Verstraeten, and J. Campenhout, “An overview of reservoir computing: theory, applications and implementations,” in *ESANN’2007 proceedings - European Symposium on Artificial Neural Networks, Bruges, Belgium, 471-482*, 2007.

[33] W. Maass, T. Natschlager, and H. Markram, “Real-time computing without stable states: a new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14(11), pp. 2531–2560, 2002.

[34] P. Tino and G. Dorffner, “Predicting the future of discrete sequences from fractal representations of the past,” *Machine Learning*, vol. 45(2), pp. 187–218, 2001.

[35] B. Jones, D. Stekel, J. Rowe, and C. Fernando, “Is there a liquid state machine in the bacterium escherichia coli?” in *In Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-Alife), pages 187-191.*, 2007.

[36] Z. Deng and Y. Zhang, “Collective behavior of a small-world recurrent neural system with scale-free distribution,” *IEEE Transactions on Neural Networks*, vol. 18(5), pp. 1364–1375, 2007.

[37] K. Dockendorf, I. Park, H. Ping, J. Principe, and T. DeMarse, “Liquid state machines and cultured cortical networks: The separation property,” *Biosystems*, vol. 95(2), pp. 90–97, 2009.

[38] H. Jaeger, W. Maass, and J. Principe, “Special issue,” *Neural Networks*, vol. 20, 2007.

[39] W. Mass, T. Natschlager, and H. Markram, “Fading memory and kernel properties of generic cortical microcircuit models,” *Journal of Physiology*, vol. 98(4-6), pp. 315–330, 2004.

[40] S. Hausler, M. Markram, and W. Maass, “Perspectives of the high-dimensional dynamics of neural microcircuits from the point of view of low-dimensional readouts,” *Complexity (Special Issue on Complex Adaptive Systems)*, vol. 8(4), pp. 39–50, 2003.

[41] K. Ishii, T. van der Zant, V. Becanovic, and P. Ploger, “Identification of motion with echo state network,” in *In Proceedings of the OCEANS 2004 MTS/IEEE -TECHNO-OCEAN Conference, volume 3, pages 1205-1210*, 2004.

[42] A. Ajdari Rad, M. Jalili, and M. Hasler, “Reservoir optimization in recurrent neural networks using kronecker kernels,” in *In IEEE ISCAS*, 2008.

Appendix A
Detailed Results

Detailed results including standard deviations across repeated experiments (as described in section IV) are shown in tables IV and V.

Appendix B
Notation and auxiliary results

We consider an ESN with linear reservoir endowed with cycle topology (SCR). The reservoir weight is denoted by r. Since we consider a single input, the input matrix V is an N-dimensional vector $V_{1..N} = (V_1, V_2, ..., V_N)^T$. By $V_{N..1}$ we denote the ‘reverse’ of $V_{1..N}$, e.g. $V_{N..1} = (V_N, V_{N-1}, ..., V_2, V_1)^T$. Consider a vector rotation operator rot_1 that cyclically rotates vectors by 1 place to the right, e.g. given a vector $a = (a_1, a_2, ..., a_n)^T$, $\mathrm{rot}_1(a) = (a_n, a_1, a_2, ..., a_{n-1})^T$. For $k \geq 0$, the k-fold application of rot_1 is denoted by[14] rot_k.

The $N \times N$ matrix with k-th column equal to $\mathrm{rot}_k(V_{N..1})$ is denoted by Ω , e.g.

$$\Omega = (\mathrm{rot}_1(V_{N..1}), \mathrm{rot}_2(V_{N..1}), ..., \mathrm{rot}_N(V_{N..1})).$$

We will need a diagonal matrix with diagonal elements $1, r, r^2, ..., r^{N-1}$:

$$\Gamma = \mathrm{diag}(1, r, r^2, ..., r^{N-1}).$$

Furthermore, we will denote the matrix $\Omega^T\ \Gamma^2\ \Omega$ by A,

$$A = \Omega^T\ \Gamma^2\ \Omega$$

[14] rot_0 is the identity mapping.

TABLE IV
TEST SET PERFORMANCE OF ESN, SCR, DLR, AND DLRB TOPOLOGIES ON DIFFERENT DATASETS FOR INTERNAL NODES WITH $tanh$ TRANSFER FUNCTION.

Data set	reservoir size	ESN	DLR	DLRB	SCR
10th order *NARMA*	50	0.166 (0.0171)	0.163 (0.0138)	**0.158** (0.0152)	0.160 (0.0134)
	100	**0.0956** (0.0159)	0.112(0.0116)	0.105 (0.0131)	0.0983 (0.0156)
	150	**0.0514** (0.00818)	0.0618 (0.00771)	0.0609 (0.00787)	0.0544 (0.00793)
	200	0.0425 (0.0166)	0.0476 (0.0104)	**0.0402** (0.0110)	0.0411 (0.0148)
10th order random *NARMA*	50	0.131 (0.0165)	0.133 (0.0132)	0.130 (0.00743)	**0.129** (0.0111)
	100	**0.0645** (0.0107)	0.0822 (0.00536)	0.0837 (0.00881)	0.0719 (0.00501)
	150	**0.0260** (0.0105)	0.0423 (0.00872)	0.0432 (0.00933)	0.0286 (0.00752)
	200	**0.0128** (0.00518)	0.0203 (0.00536)	0.0201 (0.00334)	0.0164 (0.00412)
20th order *NARMA*	50	0.297 (0.0563)	0.232 (0.0577)	0.238 (0.0507)	**0.221** (0.0456)
	100	0.235 (0.0416)	0.184 (0.0283)	0.183 (0.0196)	**0.174** (0.0407)
	150	0.178 (0.0169)	0.171 (0.0152)	0.175 (0.0137)	**0.163** (0.0127)
	200	0.167 (0.0164)	0.165 (0.0158)	0.160 (0.0153)	**0.158** (0.0121)
laser	50	**0.0184** (0.00231)	0.0210 (0.00229)	0.0215 (0.00428)	0.0196 (0.00219)
	100	**0.0125** (0.00117)	0.0132 (0.00116)	0.0139 (0.00121)	0.0131 (0.00105)
	150	**0.00945** (0.00101)	0.0107 (0.00114)	0.0112 (0.00100)	0.0101 (0.00109)
	200	**0.00819** (5.237E-04)	0.00921 (9.122E-04)	0.00913 (9.367E-04)	0.00902 (6.153E-04))
Hénon Map	50	**0.00975** (0.000110)	0.0116 (0.000214)	0.0110 (0.000341)	0.0106 (0.000185)
	100	**0.00894** (0.000122)	0.00982 (0.000143)	0.00951 (0.000120)	0.00960 (0.000124)
	150	**0.00871** (4.988E-05)	0.00929 (6.260E-05)	0.00893 (6.191E-05)	0.00921 (5.101E-05)
	200	**0.00868** (8.704E-05)	0.00908 (9.115E-05)	0.00881 (9.151E-05)	0.00904 (9.250E-05)
Non-linear communication channel	50	0.0038 (4.06E-4)	**0.0034** (2.27E-4)	0.0036 (2.26E-4)	0.0035 (2.55E-4)
	100	0.0021 (4.42E-4)	**0.0015** (1.09E-4)	0.0016 (1.07E-4)	**0.0015** (1.23E-4)
	150	0.0015 (4.01E-4)	**0.0011** (1.12E-4)	**0.0011** (1.08E-4)	0.0012 (1.23E-4)
	200	0.0013 (1.71E-4)	**0.00099** (6.42E-5)	0.0010 (7.41E-5)	0.0010 (7.28E-5)
Isolated Digits	50	**0.0732** (0.0193)	0.0928 (0.0177)	0.1021 (0.0204)	0.0937 (0.0175)
	100	**0.0296** (0.0063)	0.0318 (0.0037)	0.0338 (0.0085)	0.0327 (0.0058)
	150	**0.0182** (0.0062)	0.0216 (0.0052)	0.0236 (0.0050)	0.0192 (0.0037)
	200	0.0138 (0.0042)	**0.0124** (0.0042)	0.0152 (0.0038)	0.0148 (0.0050)

TABLE V
TEST SET PERFORMANCE OF SCR TOPOLOGY ON DIFFERENT DATASETS USING THREE DIFFERENT WAYS OF GENERATING PSEUDO-RANDOMIZED INPUT SIGN PATTERNS: INITIAL DIGITS OF π AND Exp; SYMBOLIC DYNAMICS OF LOGISTIC MAP.

Data set	reservoir size	ESN	SCR-PI	SCR-Ex	SCR-Log
20th order *NARMA*	50	0.297 (0.0563)	0.233 (0.0153)	0.232 (0.0175)	**0.196** (0.0138)
	100	0.235 (0.0416)	0.186 (0.0166)	0.175 (0.0136)	**0.169** (0.0172)
	150	0.178 (0.0169)	0.175 (0.00855)	0.158 (0.0103)	**0.156** (0.00892)
	200	0.167 (0.0164)	0.166 (0.00792)	0.157 (0.00695)	**0.155** (0.00837)
laser	50	0.0184 (0.00231)	0.0204	0.0187	**0.0181**
	100	**0.0125** (0.00117)	0.0137	0.0153	0.0140
	150	**0.00945** (0.00101)	0.0115	0.0111	0.0126
	200	**0.00819** (5.237E-04)	0.00962	0.00988	0.0107
Hénon Map	50	**0.00975** (0.000110)	0.00986	0.00992	0.00998
	100	**0.00894** (0.000122)	0.00956	0.00985	0.00961
	150	**0.00871** (4.988E-05)	0.00917	0.00915	0.00920
	200	**0.00868** (8.704E-05)	0.00892	0.00883	0.00898
Non-linear communication channel	50	0.0038 (4.06E-4)	0.0036 (1.82E-04)	**0.0026** (6.23E-05)	0.0033 (1.09E-04)
	100	0.0021 (4.42E-4)	0.0016 (7.96E-05)	0.0017 (1.04E-04)	**0.0015** (8.85E-5)
	150	0.0015 (4.01E-4)	0.0012 (7.12E-05)	**0.0011** (6.10E-05)	0.0012 (4.56E-05)
	200	0.0013 (1.71E-4)	**0.00088** (2.55E-05)	0.00090 (3.05E-05)	0.00093 (3.33E-05)

and (provided A is invertible)

$$\begin{aligned} & (\mathrm{rot}_k(V_{1..N}))^T\ A^{-1}\ \mathrm{rot}_k(V_{1..N}), \quad k \geq 0, \\ = \quad & (\mathrm{rot}_{k(\mathrm{mod})N}(V_{1..N}))^T\ A^{-1}\ \mathrm{rot}_{k(\mathrm{mod})N}(V_{1..N}), \end{aligned}$$

by ζ_k.

Lemma 1: If Ω is a regular matrix, then $\zeta_N = 1$ and $\zeta_k = r^{-2k}$, $k = 1, 2, ..., N-1$.

Proof: Denote the standard basis vector $(1, 0, 0, ..., 0)^T$ in $\Re^N$ by e_1.It holds:

$$\mathrm{rot}_k(V_{1..N}) = \Omega^T \mathrm{rot}_k(e_1), \quad k = 1, 2, ..., N-1.$$

This can be easily shown, as $\Omega^T \mathrm{rot}_k(e_1)$ selects the $(k+1)$st column of Ω^T ($(k+1)$st row of Ω), which is formed by $(k+1)$st elements of vectors $\mathrm{rot}_1(V_{N..1})$, $\mathrm{rot}_2(V_{N..1})$, ..., $\mathrm{rot}_N(V_{N..1})$. This vector is equal to the k-th rotation of $V_{1..N}$.

It follows that for $k = 1, 2, ..., N-1$,

$$(\mathrm{rot}_k(V_{1..N}))^T \Omega^{-1} = (\mathrm{rot}_k(e_1))^T$$

and so

$$\begin{aligned} \zeta_k &= (\mathrm{rot}_k(V_{1..N}))^T\ A^{-1}\ \mathrm{rot}_k(V_{1..N}) \\ &= (\mathrm{rot}_k(V_{1..N}))^T\ \Omega^{-1}\ \Gamma^{-2}\ (\Omega^{-1})^T\ \mathrm{rot}_k(V_{1..N}) \\ &= (\mathrm{rot}_k(e_1))^T\ \Gamma^{-2}\ \mathrm{rot}_k(e_1). \\ &= r^{-2k}. \end{aligned}$$

APPENDIX C
PROOF OF THEOREM 1

Given an i.i.d. zero-mean real-valued input stream $s(..t) = ...\, s(t-2)\ s(t-1)\ s(t)$ emitted by a source P , the activations of the reservoir units at time t are given by

$$\begin{aligned}
x_1(t) &= V_1\ s(t) + r\ V_N\ s(t-1) + r^2\ V_{N-1}\ s(t-2)\\
&+ r^3\ V_{N-2}\ s(t-3) + ... + r^{N-1}\ V_2\ s(t-(N-1))\\
&+ r^N\ V_1\ s(t-N) + r^{N+1}\ V_N\ s(t-(N+1)) + ...\\
&+ r^{2N-1}\ V_2\ s(t-(2N-1)) + r^{2N}\ V_1\ s(t-2N)\\
&+ r^{2N+1}\ V_N\ s(t-(2N+1)) + ...
\end{aligned}$$

$$\begin{aligned}
x_2(t) &= V_2\ s(t) + r\ V_1\ s(t-1) + r^2\ V_N\ s(t-2)\\
&+ r^3\ V_{N-1}\ s(t-3) + ... + r^{N-1}\ V_3\ s(t-(N-1))\\
&+ r^N\ V_2\ s(t-N) + r^{N+1}\ V_1\ s(t-(N+1)) + ...\\
&+ r^{2N-1}\ V_3\ s(t-(2N-1)) + r^{2N}\ V_2\ s(t-2N)\\
&+ r^{2N+1}\ V_1\ s(t-(2N+1)) + r^{2N+2}\ V_N\ s(t-(2N+2))\\
&+ ...
\end{aligned}$$

$$\begin{aligned}
x_N(t) &= V_N\ s(t) + r\ V_{N-1}\ s(t-1)\\
&+ r^2\ V_{N-2}\ s(t-2) + ...\\
&+ r^{N-1}\ V_1\ s(t-(N-1)) + r^N\ V_N\ s(t-N)\\
&+ r^{N+1}\ V_{N-1}\ s(t-(N+1)) + ...\\
&+ r^{2N-1}\ V_1\ s(t-(2N-1)) + r^{2N}\ V_N\ s(t-2N)\\
&+ r^{2N+1}\ V_{N-1}\ s(t-(2N+1))\\
&+ r^{2N+2}\ V_{N-2}\ s(t-(2N+2)) + ...
\end{aligned}$$

For the task of recalling the input from k time steps back, the optimal least-squares readout vector U is given by

$$U = R^{-1}\ p_k, \tag{15}$$

where

$$R = E_{P(s(..t))}[x(t)\ x^T(t)]$$

is the covariance matrix of reservoir activations and

$$p_k = E_{P(s(..t))}[x(t)\ s(t-k)].$$

The covariance matrix R can be obtained in an analytical form. For example, because of the zero-mean and i.i.d. nature of the source P , the element $R_{1,2}$ can be evaluated as follows:

$$\begin{aligned}
R_{1,2} &= E_{P(s(..t))}[x(t)x^T(t)]\\
&= E[\ V_1\ V_2\ s^2(t) + r^2\ V_N\ V_1\ s^2(t-1)\\
&+ r^4\ V_{N-1}\ V_N\ s^2(t-2) + .. + r^{2(N-1)}\ V_2\ V_3\ s^2(t-(N-1))\\
&+ r^{2N}\ V_1\ V_2\ s^2(t-N) + r^{2(N+1)}\ V_N\ V_1\ s^2(t-(N+1))\\
&+ ... + r^{2(2N-1)}\ V_2\ V_3\ s^2(t-(2N-1))\\
&+ r^{4N}\ V_1\ V_2\ s^2(t-2N) + ...\ ]\\
&= V_1\ V_2\ Var[s(t)] + r^2\ V_N\ V_1\ Var[s(t-1)]\\
&+ r^4\ V_{N-1}\ V_N\ Var[s(t-2)] + ...\\
&...\ + r^{2N}\ V_1\ V_2\ Var[s(t-N)] + ...\\
&= \sigma^2\ (V_1V_2 + r^2V_NV_1 + r^4V_{N-1}V_N + ...\\
&...\ + r^{2(N-1)}V_2V_3 + r^{2N}V_1V_2 + ...)\\
&= \sigma^2\ (V_1V_2 + r^2V_NV_1 + r^4V_{N-1}V_N + ...\\
&...\ + r^{2(N-1)}V_2V_3)\ \sum_{j=0}^{\infty} r^{2Nj}.
\end{aligned} \tag{16}$$

where σ^2 is the variance of P. The expression (16) for $R_{1,2}$ can be written in a compact form as

$$R_{1,2} = \frac{\sigma^2}{1-r^{2N}}\ (\mathrm{rot}_1(V_{N..1}))^T\ \Gamma^2\ \mathrm{rot}_2(V_{N..1}). \tag{17}$$

In general,

$$R_{i,j} = \frac{\sigma^2}{1-r^{2N}}\ (\mathrm{rot}_i(V_{N..1}))^T\ \Gamma^2\ \mathrm{rot}_j(V_{N..1}),\quad i,j = 1,2,...,N, \tag{18}$$

and

$$\begin{aligned}
R &= \frac{\sigma^2}{1-r^{2N}}\ \Omega^T\ \Gamma^2\ \Omega\\
&= \frac{\sigma^2}{1-r^{2N}}\ A.
\end{aligned} \tag{19}$$

By analogous arguments,

$$p_k = r^k\ \sigma^2\ \mathrm{rot}_k(V_{1..N}). \tag{20}$$

Hence, the optimal readout vector reads (see (15)):

$$U = (1-r^{2N})\ r^k\ A^{-1}\ \mathrm{rot}_k(V_{1..N}). \tag{21}$$

The ESN output at time t is

$$\begin{aligned}
y(t) &= x(t)^T\ U\\
&= (1-r^{2N})\ r^k\ x(t)^T\ A^{-1}\ \mathrm{rot}_k(V_{1..N}).
\end{aligned}$$

Covariance of the ESN output with the target can be evaluated as:

$$\begin{aligned}
Cov(y(t), s(t-k)) &= (1-r^{2N})\ r^k\ Cov(x(t)^T, s(t-k))\\
&\times\ \ A^{-1}\ \mathrm{rot}_k(V_{1..N})\\
&= r^{2k}\ (1-r^{2N})\ \sigma^2\ (\mathrm{rot}_k(V_{1..N}))^T\\
&\times\ \ A^{-1}\ \mathrm{rot}_k(V_{1..N})\\
&= r^{2k}\ (1-r^{2N})\ \sigma^2\ \zeta_k.
\end{aligned}$$

Variance of the ESN output is determined as:

$$\begin{aligned}
Var(y(t)) &= U^T\ E[x(t)\ x(t)^T]\ U\\
&= U^T\ R\ U\\
&= p_k^T\ R^{-1}\ p_k\\
&= r^{2k}\ (\sigma^2)^2\ (\mathrm{rot}_k(V_{1..N}))^T\ R^{-1}\ \mathrm{rot}_k(V_{1..N})\\
&= Cov(y(t), s(t-k)).
\end{aligned}$$

We can now calculate the squared correlation coefficient between the desired output (input signal delayed by k time steps) and the network output $y(n)$:

$$\begin{aligned}
MC_k &= \frac{Cov^2(s(t-k), y(t))}{Var(s(t))\ Var(y(t))}\\
&= \frac{Var(y(t))}{\sigma^2}\\
&= r^{2k}\ (1-r^{2N})\ \zeta_k.
\end{aligned}$$

The memory capacity of the ESN is given by

$$MC = MC_{\geq 0} - MC_0,$$

where

$$\begin{aligned}
MC_{\geq 0} &= \sum_{k=0}^{\infty} MC_k\\
&= (1-r^{2N})\ \left[\sum_{k=0}^{N-1} r^{2k}\ \zeta_k + \sum_{k=N}^{2N-1} r^{2k}\ \zeta_k + \sum_{k=2N}^{3N-1} r^{2k}\ \zeta_k + ...\right]\\
&= (1-r^{2N})\ \left[\sum_{k=0}^{N-1} r^{2k}\ \zeta_k\right]\ \left[\sum_{k=0}^{\infty} r^{2k}\right]\\
&= \sum_{k=0}^{N-1} r^{2k}\ \zeta_k.
\end{aligned}$$

Hence,

$$
\begin{aligned}
MC &= \left[\sum_{k=0}^{N-1} r^{2k}\ \zeta_k\right] - (1-r^{2N})\zeta_0 \\
&= \zeta_0\ [1-(1-r^{2N})]\ + \sum_{k=1}^{N-1} r^{2k}\ \zeta_k \\
&= \zeta_0\ r^{2N}\ + \sum_{k=1}^{N-1} r^{2k}\ \zeta_k \\
&= \zeta_N\ r^{2N}\ + \sum_{k=1}^{N-1} r^{2k}\ \zeta_k \\
&= \sum_{k=1}^{N} r^{2k}\ \zeta_k .
\end{aligned}
$$

By lemma 1, $r^{2k}\ \zeta_k = 1$ for $k = 1, 2, ..., N-1,$ and $r^{2N}\ \zeta_N = r^{2N}$. It follows that $MC = N - 1 + r^{2N}$.

PLACE PHOTO HERE

Ali Rodan received the BSc and MSc degrees in Computer Science from Princess Sumaya University for Technology, Amman, Jordan, in 2004, and from Oxford Brookes University, Oxford, U.K, in 2005, respectively. He is currently working towards the Ph.D. degree in Computer Science at the University of Birmingham, UK. His research interests include Recurrent Neural Networks, Support Vector Machines, Reservoir Computing, and Data Mining.

PLACE PHOTO HERE

Peter Tiňo received the M.Sc. degree from the Slovak University of Technology, Bratislava, Slovakia, in 1988 and the Ph.D. degree from the Slovak Academy of Sciences, Bratislava, Slovakia, in 1997.

He was a Fullbright Fellow at the NEC Research Institute in Princeton, NJ, USA, from 1994 to 1995. He was a Postdoctoral Fellow at the Austrian Research Institute for AI in Vienna, Austria, from 1997 to 2000, and a Research Associate at the Aston University, UK, from 2000 to 2003. He is with the School of Computer Science, the University of Birmingham, UK, since 2003 and is currently a a senior lecturer. He is on the editorial board of several journals. His main research interests include probabilistic modeling and visualization of structured data, statistical pattern recognition, dynamical systems, evolutionary computation, and fractal analysis.

Dr. Tiňo was awarded the Fullbright Fellowship in 1994 and the UK-Hong Kong Fellowship for Excellence in 2008. He was awarded the Outstanding Paper of the Year for IEEE Transactions on Neural Networks with T. Lin, B.G. Horne, and C.L. Giles in 1998 for the work on recurrent neural networks. He won the 2002 Best Paper Award at the International Conference on Artificial Neural Networks with B. Hammer. In 2010 the paper he co-authored with S.Y. Chong and X. Yao won the 2011 IEEE Computational Intelligence Society Outstanding IEEE Transactions on Evolutionary Computation Paper Award.