Simple Deterministically Constructed Cycle Reservoirs with Regular Jumps

Ali Rodan[1] **and Peter Tiňo**[1]

[1]School of computer Science, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (email: a.a.rodan, P.Tino@cs.bham.ac.uk).

Keywords: Reservoir Computing, Echo State Networks, Simple Recurrent Neural Networks, Memory Capability.

Abstract A new class of state-space models, reservoir models, with a fixed state transition structure (the "reservoir") and an adaptable readout from the state space has recently emerged as a way for time series processing/modelling. Echo State Network (ESN) is one of the simplest, yet powerful, reservoir models. ESN models are generally constructed in a randomized manner. In our previous study (Rodan & Tino, 2011) we showed that a very simple, cyclic, deterministically generated reservoirs can yield performance competitive with standard ESN. In this contribution we extend our previous study in three aspects: **1)** We introduce a novel simple deterministic reservoir model, Cycle Reservoir with Jumps (CRJ), with highly constrained weight values, that has superior performance to standard ESN on a variety of temporal tasks of different origin and characteristics. **2)** We elaborate on the possible link between reservoir characterizations, such as eigenvalue distribution of the reservoir matrix or pseudo-Lyapunov exponent of the input-driven reservoir dynamics, and the model performance. It has been suggested that a uniform coverage of the unit disk by such eigenvalues can lead to superior model performances. We show that despite highly constrained eigenvalue distribution, CRJ consistently outperform ESN (that have much more uniform eigenvalue

coverage of the unit disk). Also, unlike in the case of ESN, pseudo-Lyapunov exponents of the selected 'optimal' CRJ models are consistently negative. **3)** We present a new framework for determining short term memory capacity of linear reservoir models to a high degree of precision. Using the framework we study the effect of shortcut connections in the CRJ reservoir topology on its memory capacity.

1 Introduction

Reservoir Computing (RC) (Lukosevicius & Jaeger, 2009) is a new class of state-space models based on a fixed randomly constructed state transition mapping (realized through so-called *reservoir*) and an adaptable (usually linear) *readout* mapping from the reservoir. Echo State Networks (ESNs) (Jaeger, 2001), Liquid State Machines (LSMs) (Maass et al., 2002) and the back-propagation decorrelation neural network (Steil, 2004) are examples of popular RC methods. For a comprehensive review of RC see (Lukosevicius & Jaeger, 2009).

In this contribution we concentrate on Echo State Networks, one of the simplest, yet effective forms of reservoir computing. Briefly, ESN is a recurrent neural network with a non-trainable sparse recurrent part (reservoir) and a simple linear readout. Typically, the reservoir connection weights, as well as the input weights are randomly generated. The reservoir weights are then scaled so that the spectral radius of the reservoir's weight matrix W is < 1. This ensures a sufficient condition for the *"Echo State Property"* (ESP): the reservoir state is an *"echo"* of the entire input history. ESN has been successfully applied in time-series prediction tasks (Jaeger & Hass, 2004), speech recognition (Skowronski & Harris, 2006), noise modelling (Jaeger & Hass, 2004), dynamic pattern classification (Jaeger, 2002b), reinforcement learning (Bush & Anderson, 2005), and in language modelling (Tong et al., 2007).

A variety of extensions/modifications of the classical ESN can be found in the literature, e.g. intrinsic plasticity (Schrauwen et al., 2008b; Steil, 2007), refined training algorithms (Jaeger & Hass, 2004), leaky-integrator reservoir units (Jaeger et al., 2007a), support vector machine (Schmidhuber et al., 2007), filter neurons with delay&sum readout (Holzmann & Hauser, 2009), pruning connections within the reservoir (Dutoit et al., 2009) etc. There have also been attempts to impose specialized interconnection topolo-

gies on the reservoir, e.g. hierarchical reservoirs (Jaeger, 2007), small-world reservoirs (Deng & Zhang, 2007) and decoupled sub-reservoirs (Xue et al., 2007).

There are still problems preventing ESN to become a widely accepted tool, e.g. poorly understood reservoir properties (Xue et al., 2007), reservoir specification requires numerous trails and even luck (Xue et al., 2007), random connectivity and weight structure of the reservoir is unlikely to be optimal (Ozturk et al., 2007). Typically, in order to construct a reservoir model one needs to specify the reservoir size, sparsity of reservoir and input connections, scaling of input and reservoir weights.

Simple reservoir topologies have been proposed as alternative to the randomized ESN reservoir - e.g. 'feedforward' reservoirs with tap delay connections (Cernansky & Makula, 2005), reservoir with diagonal weight matrix (self-loops) (Fette & Eggert, 2005) and cycle topology of reservoir connections (Rodan & Tino, 2011). The Simple Cycle Reservoir (SCR) introduced in (Rodan & Tino, 2011) achieved comparable performances to 'standard' ESN on a variety of data sets of different origin and memory structure. We also proved that the memory capacity of linear SCR can be made arbitrarily close to the proven optimal value (for any recurrent neural network of the ESN form).

In this paper we extend the cycle reservoir of (Rodan & Tino, 2011) with a regular structure of shortcuts (Jumps) - *Cycle Reservoir with Jumps* (CRJ). In the spirit of SCR we keep the reservoir construction simple and deterministic. Yet, it will be shown that such an extremely simple regular architecture can significantly outperform both SCR and standard randomized ESN models. Prompted by these results, we investigate some well known reservoir characterizations, such as eigenvalue distribution of the reservoir matrix, pseudo-Lyapunov exponent of the input-driven reservoir dynamics, or memory capacity and their relation to the ESN performance.

The paper is organized as follows. Section 2 gives a brief overview of Echo state network design and training. In Section 3 we present our proposed model - CRJ. Experimental results are presented and discussed in Sections 4 and 5, respectively. Section 6 investigates three reservoir characterizations (eigen-spectrum of the reservoir weight matrix, short term memory capacity and pseudo-Lyapunov exponent) in the context of reservoir models studied in this paper. Finally, the work is concluded in section 7.

2 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: $s(n) = (s_1(t), ..., s_K(t))^T$, $x(t) = (x_1(t), ..., x_N(t))^T$, 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.

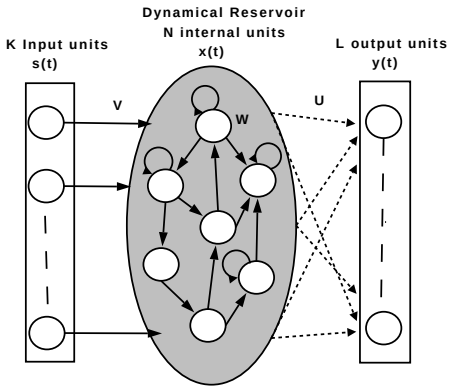

Figure 1: Echo State Network (ESN) Architecture

The internal units are updated according to[1]:

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

where f is the reservoir activation function (typically tanh or some other sigmoidal function); $z(t+1)$ is an optional uniform i.i.d. noise. The linear readout is computed as[2]:

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

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 echo state property, the reservoir connection matrix W is typically scaled as $W \leftarrow \alpha W/|\lambda_{max}|$,

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

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

where $|\lambda_{max}|$ is the spectral radius[3] of W and $0 < \alpha < 1$ is a scaling parameter[4] (Jaeger, 2002b).

ESN memoryless readout can be trained both offline (Batch) and online by minimizing a given loss function. In most cases we evaluate the model performance via Normalized Mean Square Error (NMSE):

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

where $\hat{y}(t)$ is the readout output, $y(t)$ is the desired output (target), $\|.\|$ denotes the Euclidean norm and $\langle \cdot \rangle$ denotes the empirical mean.

In the *offline (Batch) training* mode one first runs the network on the training set, and subsequently computes the output weights that minimize the NMSE. In summary, the following steps are performed:

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 the remaining network states $x(t)$ into a matrix X.
3. The target values from the training set are collected in a vector y.
4. The output unit weights are computed using ridge regression:

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

   where I is the identity matrix and $\rho > 0$ is a regularization factor determined on a hold-out validation set .

Standard recursive algorithms, such as Recursive Least Squares (RLS), for NMSE minimization can be used in *online readout training*. In RLS, after the initial washout period the output weights U are recursively updated at every time step t:

$$k(t) = \frac{P(t-1)\ x(t)}{x^T(t)\ P(t-1)\ x(t) + \gamma} \tag{5}$$

$$P(t) = \gamma^{-1}(P(t-1) - k(t)\ x^T(t)\ P(t-1)) \tag{6}$$

$$U(t) = U(t-1) + k(t)\ [y(t) - \hat{y}(t)] \tag{7}$$

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

[4] Such scaling corresponds to the sufficient condition for ESP only. For necessary condition, maximum singular value would need to be used.

where k stands for the innovation vector; y and $\hat{y}$ correspond to the desired and calculated (readout) output unit activities; P is the error covariance matrix initialized with large diagonal values. 'Forgetting parameter' $0 < \gamma < 1$ is usually set to a value close to 1.0. In this work γ is set on a hold-out validation set.

3 Cycle Reservoir with Jumps

In (Rodan & Tino, 2011) we proposed a Simple Cycle Reservoir (SCR) with performance competitive to that of standard ESN. Unlike ESN, the construction of SCR model is completely deterministic and extremely simple. All cyclic reservoir weights have the same value; all input connections also have the same absolute value. Viewing reservoir interconnection topology as a graph, the SCR has a small degree of local clustering and a large average path length. In contrast, ESN (a kind of random network) has small degree of local clustering and small average path length. It has been argued that reservoirs should ideally have small clustering degree (sparse reservoirs) (Jaeger & Hass, 2004) so that the dynamic information flow through the reservoir nodes is not 'too cluttered'. Also a small average path length, while having longer individual paths within the reservoir, can allow for representation of a variety of dynamical time scales. We propose a Cycle Reservoir with Jumps (CRJ) which, compared with SCR leads to slightly higher degree of local clustering while achieving much smaller average path length.

The CRJ model has a fixed simple regular topology: the reservoir nodes are connected in a uni-directional cycle (as in SCR) with bi-directional shortcuts (jumps) (Fig. 2). All cycle connections have the same weight $r_c > 0$ and likewise all jumps share the same weight $r_j > 0$. In other words, non-zero elements of W are:

- the 'lower' sub-diagonal $W_{i+1,i} = r_c$, for $i = 1...N-1$,
- the 'upper-right corner' $W_{1,N} = r_c$ and
- the jump entries r_j. Consider the jump size $1 < \ell < \lfloor N/2 \rfloor$. If $(N \mod \ell) = 0$, then there are N/ℓ jumps, the first jump being from unit 1 to unit $1+\ell$, the last one from unit $N+1-\ell$ to unit 1 (see Figure 2 (A)). If $(N \mod \ell) \neq 0$, then there are $\lfloor N/\ell \rfloor$ jumps, the last jump ending in unit $N+1-(N \mod \ell)$ (see Figure 2 (B)). In such cases, we also consider extending the reservoir size by κ

units ($1 \leq \kappa < \ell$), such that $N \mod (N + \kappa) = 0$. The jumps are bi-directional sharing the same connection weight r_j.

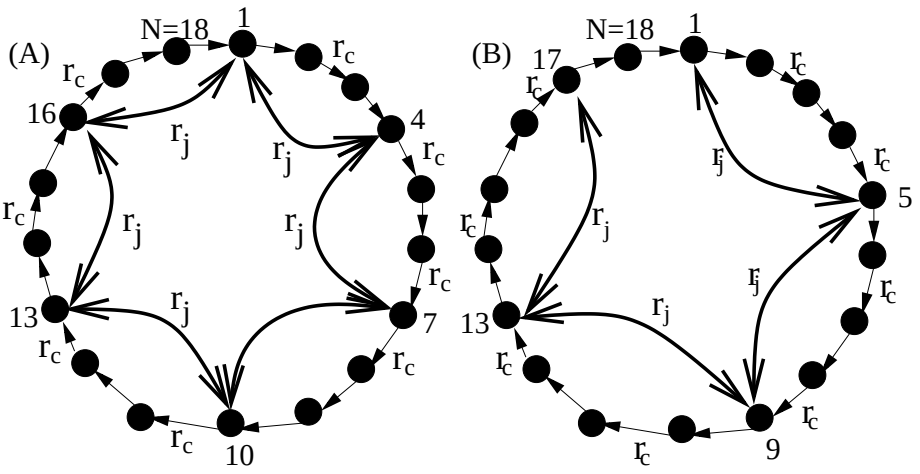

Figure 2: An Example of CRJ Reservoir Architecture with $N = 18$ Units and Jump Size $\ell = 3$ (A) and $\ell = 4$ (B).

As with the SCR model, in the CRJ model we use full input-to-reservoir connectivity with the same absolute value $v > 0$ of the connection weight. It is shown in (Rodan & Tino, 2011) that an aperiodic character of signs of the input weights in $V = (V_1, V_2, ..., V_K)$ is essential for the SCR model. Unlike in (Rodan & Tino, 2011), in this contribution we use the same method for obtaining the input weight signs, universally across all data sets. In particular, the input signs are determined from decimal expansion $d_0.d_1d_2d_3...$ of an irrational number - in our case π. The first N decimal digits $d_1, d_2, ..., d_N$ are thresholded at 4.5, i.e. 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. The values v, r_c, and r_j are chosen on the validation set.

4 Experiments

In this section we test and compare our simple CRJ reservoir topology with standard ESN and SCR on a variety of timeseries tasks widely used in the ESN literature and covering a wide spectrum of memory structure (Schrauwen et al., 2008b; Cernansky &

Tino, 2008; Jaeger, 2001, 2002a, 2003; Jaeger & Hass, 2004; Verstraeten et al., 2007; Steil, 2007).

4.1 Experimental Setup

For each data set and each model class (ESN, SCR, and CRJ) we picked on the validation set a model representative to be evaluated on the test set. The readout mapping was fitted both using offline (Ridge Regression) and online (RLS) training. Then, based on validation set performance, the offline or online trained readout was selected and tested on the test set.

For RLS training we add noise to the internal reservoir activations where the noise is optimized for each dataset and each reservoir size using validation set (Wyffels et al., 2008). For SCR architecture the model representative is defined by the absolute input weight value $v \in (0, 1]$ and the reservoir cycle connection weight $r_c \in (0, 1]$. For the CRJ architecture the model representative is defined by the absolute input weight value $v \in (0, 1]$, the reservoir cycle connection weight $r_c \in (0, 1]$, the jump size $1 < \ell < \lfloor N/2 \rfloor$ and the jump weight $r_j \in (0, 1]$. For the ESN architecture, the model representative is specified by the reservoir sparsity, spectral radius λ of the reservoir weight matrix, input weight connectivity and input weight range $[-a, a]$. We present the results for three reservoir sizes $N = 100, 200, 300$.

For ESN we calculated out-of sample (test set) performance measures over 10 simulation runs (presented as mean and StDev). The selected SCR and CRJ representatives are evaluated out-of-sample only once, since their construction is completely deterministic. The only exception is the speech recognition experiment - due to limited test set size, following (Verstraeten et al., 2007), a 10-fold cross-validation was performed (and paired t-test was used to assess statistical significance of the result).

Details of the experimental setup, including ranges for cross-validation based grid search on free-parameters, are presented in Table 1. Detailed parameter settings of the selected model representatives can be found in the Appendix.

Table 1: Summary of the Experimental Setup. Grid Search Ranges are Specified in MATLAB Notation, i.e. $[s : d : e]$ Denotes a Series of Numbers Starting from s, Increased by Increments of d, Until the Ceiling e is Reached.

Reservoir topologies	ESN, SCR and CRJ
Readout learning	RLS with dynamic noise injection, Ridge Regression
Reservoir matrix	ESN (random weights with spectral radius α in $[0.05 : 0.05 : 1]$, and connectivity con in $[0.05 : 0.05 : 0.5]$) CRJ and SCR (r_c in $[0.05 : 0.05 : 1]$, r_j in $[0.05 : 0.05 : 1]$)
jump size	$1 < \ell < \lfloor N/2 \rfloor$, where N is the reservoir size.
reservoir size	N in $[100 : 100 : 300]$
input scale	v (for SCR and CRJ) and a (for ESN) from $[0.01 : 0.005 : 1]$
input sign generation	SCR and CRJ: thresholded decimal expansion of π
readout regularization	reservoir noise size (RLS), regularization factor (ridge regression) 10^q, $q = [-15 : 0.25 : 0]$

4.2 Experimental Tasks and Results

System Identification

As a System Identification task, we considered a NARMA system of order 10 (Atiya & Parlos, 2000) given by eq. (8).

$$y(t+1) = 0.3y(t) + 0.05y(t)\sum_{i=0}^{9} y(t-i) + 1.5s(t-9)s(t) + 0.1, \qquad (8)$$

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 $[0, 0.5]$. The current output depends on both the input and the previous outputs. In general, modelling this system is difficult due to the non-linearity and possibly long memory. The networks were trained on system identification task to output $y(t)$ based on $s(t)$. The input $s(t)$ and target data $y(t)$ are shifted by -0.5 and scaled by 2 as in (Steil, 2007). *NARMA* sequence has a length of 8000 items where the first 2000 were used for training, the following 5000 for validation, and the remaining 2000 for testing. The first 200 values from the training, validation and test sequences were used as the initial washout period.

The results are presented in Table 2. Even though SCR is slightly inferior to the standard ESN construction, the simple addition of regular shortcuts (jumps) to the SCR leads to a superior performance of CRJ topology.

Table 2: Test Set NMSE Results of ESN, SCR, and CRJ Reservoir Models on the 10th Order NARMA System. Reservoir Nodes with $tanh$ Transfer Function were Used.

reservoir model	$N = 100$	$N = 200$	$N = 300$
ESN	0.0788 (0.00937)	0.0531 (0.00198)	0.0246 (0.00142)
SCR	0.0868	0.0621	0.0383
CRJ	**0.0619**	**0.0196**	**0.0130**

Time Series Prediction

The Santa Fe Laser dataset (Jaeger et al., 2007a) is a cross-cut through periodic to chaotic intensity pulsations of a real laser. The task was to predict the next value $y(t+1)$. The dataset contains 9000 values, the first 2000 values were used for training, the next 5000 for validation, and the remaining 2000 values was used for testing the models. The first 200 values from training, validation and testing sequences were used as the initial washout period.

The results are shown in Table 3. Again, ESN and SCR are almost on-par, with SCR slightly inferior. However, the CRJ topology can outperform the other architectures by a large margin.

Table 3: Test Set NMSE Results of ESN, SCR, and CRJ Reservoir Models on the Santa Fe Laser Dataset. Reservoir Nodes with $tanh$ Transfer Function were Used.

reservoir model	$N = 100$	$N = 200$	$N = 300$
ESN	0.0128 (0.00371)	0.0108 (0.00149)	0.00895 (0.00169)
SCR	0.0139	0.0112	0.0106
CRJ	**0.00921**	**0.00673**	**0.00662**

Speech Recognition

For this task we used the Isolated Digits dataset[5]. It 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. Because of the limited test set size, 10-fold cross-validation was performed (Verstraeten et al., 2007) and paired t-test was used to assess whether the perceived differences in model performance are statistically significant. The Lyon Passive Ear model (Lyon, May 1982) 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 (Schrauwen et al., 2007) 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 modelling capabilities of different reservoir models on high-dimensional (86 input channels) time series. The results confirming superior performance of the simple CRJ model are shown in Table 4. For reservoir size $N = 100$ the CRJ model is significantly superior to ESN at the confidence level 96%. For reservoirs with $N = 200$ and $N = 300$ neurons CRJ beats ESN at significance levels greater than 99%.

Table 4: WER Results of ESN, SCR, and CRJ Models on the *Isolated Digits* (Speech Recognition) Task. Reservoir Nodes with $tanh$ Transfer Function f were Used.

reservoir model	$N = 100$	$N = 200$	$N = 300$
ESN	0.0296 (0.0063)	0.0138 (0.0042)	0.0092 (0.0037)
SCR	0.0329 (0.0031)	0.0156 (0.0035)	0.0081 (0.0022)
CRJ	**0.0281 (0.0032)**	**0.0117 (0.0029)**	**0.0046 (0.0021)**

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

Memory and Non-Linear Mapping Task

The last task, used in Verstraeten et al. (2010), is a generalization of the delay XOR-task used in (Schrauwen et al., 2008a). It allows one to systematically study two characteristics of reservoir topologies: memory and the capacity to process non-linearities in the input time series. The memory is controlled by the delay d of the output, and the 'degree of non-linearity' is determined by a parameter $p > 0$. The input signal $s(t)$ contains uncorrelated values from a uniform distribution over the interval $[-0.8, 0.8]$. The task is to reconstruct a delayed and non-linear version of the input signal:

$$y_{p,d}(t) = sign[\beta(t-d)] \cdot |\beta(t-d)|^p, \tag{9}$$

where $\beta(t-d)$ is the product of two delayed successive inputs,

$$\beta(t-d) = s(t-d) \cdot s(t-d-1).$$

The sign and absolute values are introduced to assure a symmetric output even in the case of even powers (Verstraeten et al., 2010). Following (Verstraeten et al., 2010), we considered delays $d = 1, ..., 15$ and powers $p = 1, ..., 10$ with a total of 150 output signals $y_{p,d}$ (realized as 150 readout nodes). The main purpose of this experiment is to test whether a single reservoir can have rich enough pool of internal representations of the driving input stream so as to cater for the wide variety of of outputs derived from the input for a range of delay and non-linearity parameters.

We used time series of length 8000, where a new time series was generated in each of 10 runs. The first 2000 items were used for training, the next 3000 for validation, and the remaining 3000 for testing the models. The first 200 values from training, validation and test sequences were used as the initial washout period. As in (Verstraeten et al., 2010), we used reservoirs of size 100 nodes.

Figure 3 illustrates the NMSE performance for ESN (A) , SCR (B) and CRJ (C). Shown are contour plots across the two degrees of freedom – the delay d and the non-linearity parameter p. We also show difference plots between the respective NMSE values: ESN - SCR(D), ESN - CRJ (E) and SCR - CRJ (F). When the task becomes harder (non-linearity and delay increase - upper-right corner of the contour plots) the performance of the simple reservoir constructions, SCR and CRJ, is superior to that of standard ESN. Interestingly, the simple reservoirs seem to outperform ESN by the

largest margin for moderate delays and weak non-linearity (small values of p). We do not have a clear explanation to offer but note that our later studies in section 6.2 show that, compared with ESN, the SCR and CRJ topologies have a potential for greater memory capacity. This seems to be reflected most strongly if the series is characterized by weak non-linearity.

5 Discussion

The experimental results clearly demonstrate that our very simple deterministic reservoir constructions have a potential to significantly outperform standard ESN randomized reservoirs. We propose that instead of relying on unnecessary stochastic elements in reservoir construction, one can obtain superior (and sometimes superior by a large margin) performance by employing the simple regular unidirectional circular topology with bi-directional jumps with fixed cycle and jump weights. However, it is still not clear exactly what aspects of dynamic representations in the reservoirs are of importance and why. In later sections we concentrate on three features of reservoirs - eigenspectrum of the reservoir weight matrix, (pseudo) Lyapunov exponent of the input-driven reservoir dynamics and short term memory capacity - and discuss their relation (or lack of) to the reservoir performance on temporal tasks.

Besides the symmetric bi-directional regular jumps we considered uni-directional jumps (both in the direction and in the opposite direction to the main reservoir cycle), as well as jumps not originating/ending in a regular grid of 'hub-like' nodes[6]. In all cases, compared with our regular CRJ topology, the performance was slightly worse. Of course, when allowing for two different weight values in the bidirectional jumps (one for forward, one for backward jumps), the performance improved slightly but not significantly over CRJ.

Our framework can be extended to more complex regular hierarchical reservoir constructions. For example, we can start with a regular structure of relatively short 'lower level' jumps in the style of CRJ topology. Then another layer of longer jumps over the shorter ones can be introduced etc. We refer to this architecture as *Cycle Reservoir with Hierarchical Jumps* (CRHJ). Figure 4 illustrates this idea on a 3-level hierarchy of

[6] For example, when a jump lands in unit n, the next jump originates in unit $n+1$ etc.

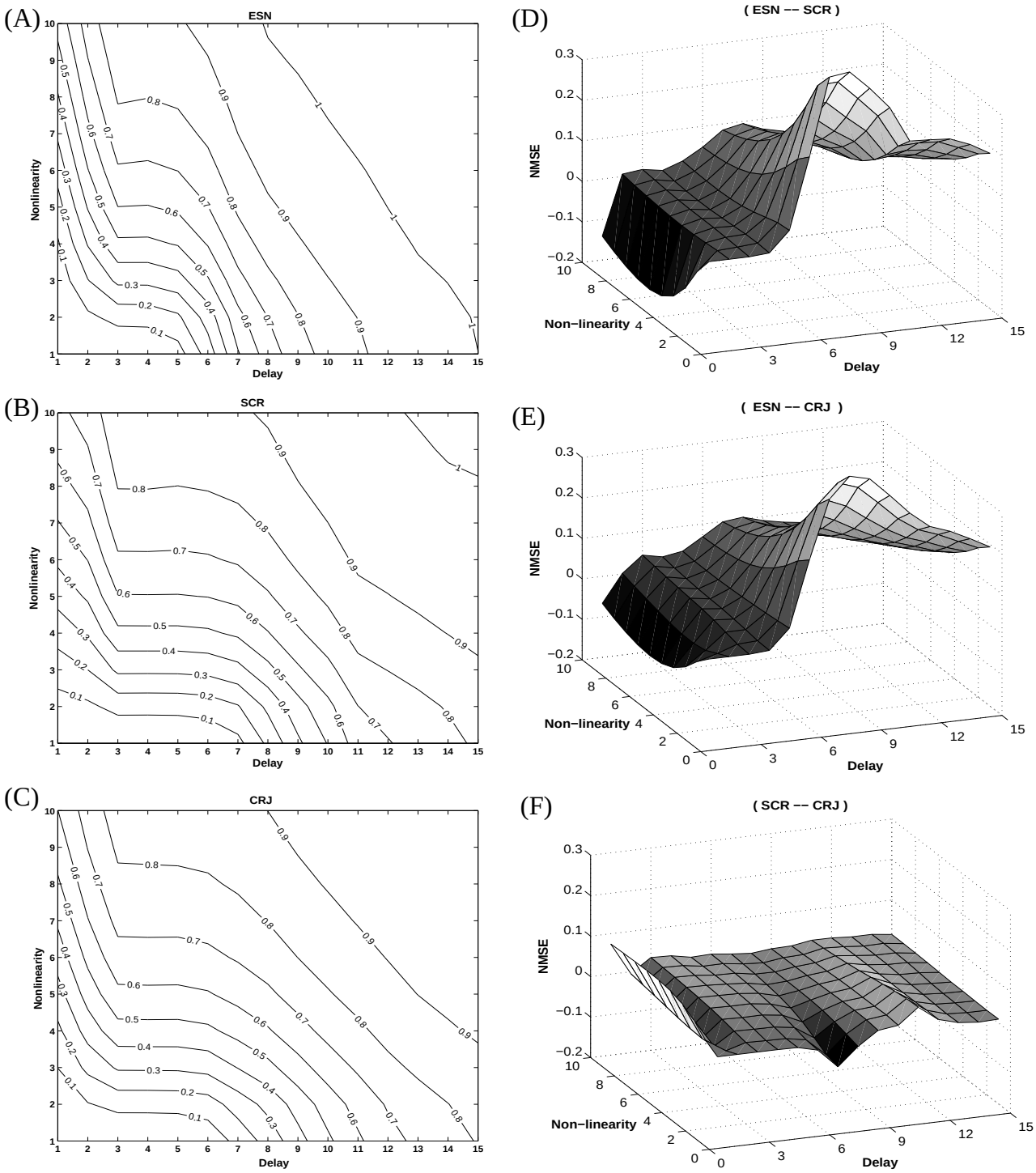

Figure 3: Memory and Non-Linear Mapping Task. Shown are NMSE Values for ESN (A), SCR (B) and CRJ (C). We also Show Difference Plots Between the Respective NMSE Values: ESN - SCR (D), ESN - CRJ (E) and SCR - CRJ (F).

jumps. As before, the cycle weights are denoted by r_c. The lowest level jump weights are denoted by r_{j_1}, the highest by r_{j_3}. On each hierarchy level, the jump weight has a single fixed value.

Table 5: Test Set NMSE Results of Deterministic CRHJ Reservoir Model on the Santa Fe Laser Dataset and NARMA System. Reservoir Nodes with $tanh$ Transfer Function were Used.

Dataset	$N = 100$	$N = 200$	$N = 300$
laser	0.00743	0.00594	0.00581
NARMA	0.0662	0.0182	0.0133

As an illustrative example, in Table 5 we show test set results for 3-level jump hierarchies with jump sizes 4, 8 and 16. We used the same jump sizes for both laser and NARMA data sets. The weights $r_c, r_{j_1}, r_{j_2}, r_{j_3} \in [0.05, 1)$ were found on the validation set. In most cases the performance of reservoirs with hierarchical jump structure slightly improves over the CRJ topology (see Tables 2 and 3). However, such more complex reservoir constructions, albeit deterministic, diverge from the spirit of the simple SCR and CRJ constructions. The potential number of free parameters (jump sizes, jump weights) grows and the simple validation set search strategy can quickly become infeasible.

The CRHJ structure differs from hierarchically structured randomized reservoir models proposed in the RC community (Jaeger, 2007; Triefenbach et al., 2010), where the reservoir structures are obtained by connecting[7] different smaller reservoirs constructed in a randomized manner.

Our CRJ reservoirs can also be related to the work of Deng & Zhang (2007) where massive reservoirs are constructed in a randomized manner so that they exhibit small-world and scale-free properties of complex networks. We refer to this model as the small world network reservoir (SWNR). We trained such SWNR architecture[8] on the laser and NARMA datasets, since for reasonable results the SWNR model needed to be of larger size, we conducted the comparative experiments with reservoirs of size

[7] possibly through trained connections

[8] We are thankful to the authors of (Deng & Zhang, 2007) for providing us with their code.

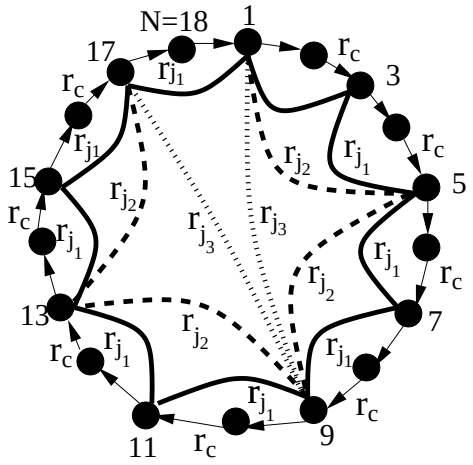

Figure 4: Reservoir Architecture of Cycle Reservoir with Hierarchical Jumps (CRHJ) with Three Hierarchical Levels. Reservoir Size $N = 18$, and the Jump Sizes are $\ell = 2$ for Level 1 , $\ell = 4$ for Level 2, and $\ell = 8$ for Level 3.

$N = 500$. The results (across 10 randomized SWNR model construction runs) for laser and NARMA data sets are presented in Table 6 . The performance was always inferior to our simple deterministically constructed CRJ reservoir. Detailed parameter settings of the selected model representatives can be found in the Appendix.

Finally, we mention that in the context of this paper, the work done in the complex network community, relating dynamics of large networks with different degrees of constrained interconnection topology between nodes, may be of interest. For example, Watts & Strogatz (1998) consider collective dynamics of networks with interconnection structure controlled from completely regular (each node on a ring connects to its k nearest neighbors), through "small-world" (for each node, with some probability p links to the nearest neighbors are rewired to any randomly chosen node on the ring), to completely random (p=1). However, such studies address different issues from those we are concerned with in this paper: first, our reservoirs are input-driven; second, our interconnection construction is completely deterministic and regular; and third, the dynamics of CRJ is given through affine functions in every node, put through a saturation sigmoid-type activation functions.

Table 6: Test Set NMSE Results of ESN, SWNR, Deterministic SCR and Deterministic CRJ reservoir Model on the Santa Fe Laser Dataset and NARMA System. Reservoir Size $N = 500$ and Reservoir Nodes with $tanh$ Transfer Function were Used.

Dataset	ESN	SWNR	SCR	CRJ
laser	0.00724 (0.00278)	0.00551 (0.00176)	0.00816	0.00512
NARMA	0.0104 (0.0020)	0.052 (0.0089)	0.0216	0.0081

6 Reservoir Characterizations

There has been a stream of research work trying to find useful characterizations of reservoirs that would correlate well with the reservoir performance on a number of tasks. For example, Legenstein & Maass (2007) introduce a 'kernel' measure of separability of different reservoir states requiring different output values. Since linear readouts are used, the separability measure can be calculated based on the rank of the reservoir design matrix[9]. In the same vein, in Bertschinger & Natschlager (2004) it is suggested that if a reservoir model is to be useful for computations on input time-series, it should have the "separation property" - different input time series which produce different outputs should have different reservoir representations. When linear readouts are used, this typically translates to 'significantly' different states. Moreover, it is desirable that the separation (distance between reservoir states) increases with the difference of the input signals.

In what follows we examine three other reservoir characterizations suggested in the literature, namely - eigenspectrum of the reservoir weight matrix Ozturk et al. (2007), (pseudo) Lyapunov exponent of the input-driven reservoir dynamics Verstraeten et al. (2007) and short term memory capacity Jaeger (2002b).

6.1 EigenSpectra of Dynamic Reservoirs

Several studies have attempted to link eigenvalue distribution of the ESN reservoir matrix W with the reservoir model's performance. First, in order to satisfy necessary

[9] reservoir states resulting from driving the reservoir with different input streams

condition for echo state property, the eigenvalues of W need to lie inside the unit circle. Ozturk, Xu and Principe (Ozturk et al., 2007) proposed that the distribution of reservoir activations should have high entropy. It is suggested that the linearized ESN designed with the recurrent weight matrix having the eigenvalues uniformly distributed inside the unit circle creates such an activation distribution (compared to other ESNs with random internal connection weight matrices). In such cases, the system dynamics will include uniform coverage of time constants (related to the uniform distribution of the poles) (Ozturk et al., 2007). However, empirical comparison of this type of reservoir with the standard ESN is still lacking (Lukosevicius & Jaeger, 2009).

It has been also suggested that sparsity of reservoir interconnections (non-zero entries in W) is a desirable property (Jaeger & Hass, 2004). On the other hand, Zhang & Wang (2008) argue that sparsely and fully connected reservoirs in ESN have the same limit eigenvalue distribution inside the unit circle. Furthermore, the requirement that the reservoir weight matrix be scaled so that the eigenvalues of W lie inside the unit circle has been criticized in (Verstraeten et al., 2006), where the experiments show that scaling with a large spectral radius seemed to be required for some tasks. On the other hand, smaller eigenvalue spread is necessarily for stable online training of the readout (Jaeger, 2005).

Our experimental results show that the simple CRJ and regular hierarchical CRHJ reservoirs outperform standard randomized ESN models on a wide variety of tasks. However, the eigenvalue spectra of our regularly and deterministically constructed reservoirs are much more constrained than those of the standard ESN models. Figure 5 shows eigenvalue distribution of representatives of the four model classes - ESN, SCR, CRJ, and CRHJ - fitted on the isolated digits dataset in the speech recognition task. Clearly the coverage of the unit circle by the ESN eigenvalues is much greater than in the case of the three regular deterministic reservoir constructions. While the ESN eigenvalues cover the unit sphere ‘almost uniformly’, the SCR, CRJ, and CRHJ eigenvalues are limited to a circular structure inside the unit disk. The eigenvalues of SCR must lie on a circle by definition. On the other hand, the eigenvalue structure of CRJ and CRHJ can be more varied. However, the eigenvalue distributions of CRJ and CRHJ reservoirs selected on datasets used in this study were all highly constrained following an approximately circular structure. This poses a question as to what aspects of eigen-

value distribution of the reservoir matrix are relevant for a particular class of problems. We suspect that the non-linear nature of the non-autonomous reservoir dynamics may be a stumbling block in our efforts to link linearized autonomous behavior of reservoirs with their modelling potential as non-linear non-autonomous systems. Deeper investigation of this issue is beyond the scope and intentions of this study, and it is a matter of future research.

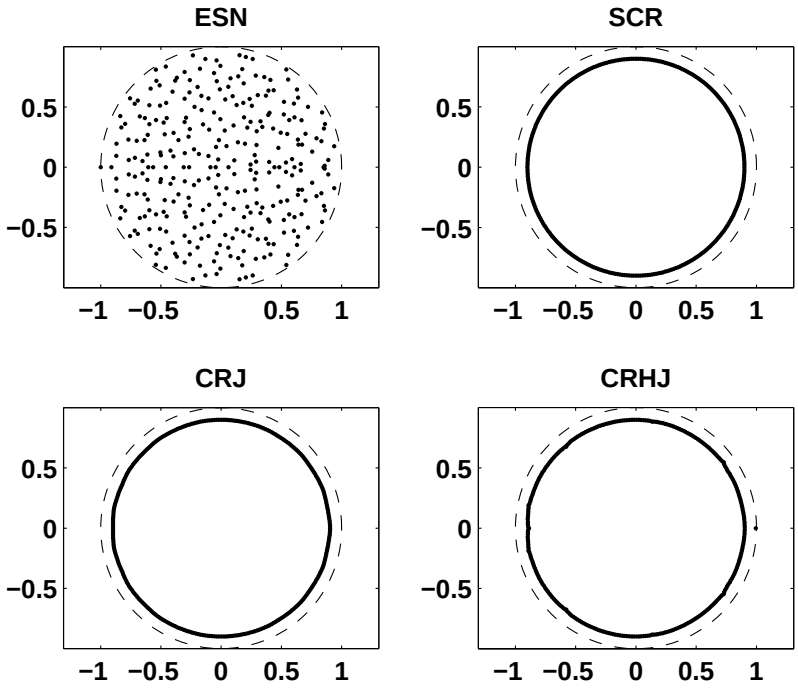

Figure 5: Eigenvalue Distribution for ESN, SCR, CRJ and CRHJ Reservoirs of $N = 300$ Neurons Selected on the Isolated Digits Dataset in the Speech Recognition Task (and Hence Used to Report Results in Table 4).

6.2 Memory Capacity

Another attempt at characterization of dynamic reservoirs is in terms of their (short-term) memory capacity (MC) (Jaeger, 2002a). It quantifies the ability of recurrent network architectures to encode past events in their state space so that past items in an i.i.d. input stream can be recovered (at least to certain degree). Consider a univariate stationary input signal $s(t)$ driving the network at the input layer. For a given delay k,

we construct a 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 $y(t)$:

$$MC_k = \frac{Cov^2(s(t-k), y(t))}{Var(s(t))\, Var(y(t))}, \tag{10}$$

where Cov denotes the covariance and Var the variance operators. The short term memory (STM) capacity is then given by (Jaeger, 2002a):

$$MC = \sum_{k=1}^{\infty} MC_k. \tag{11}$$

Traditionally, memory capacity has been estimated numerically by generating long input streams of i.i.d data and training different readouts for different delays k from 1 up to some upper bound k_{max}. Typically, due to short-term memory of reservoir models, k_{max} is of order 10^2. We will later show that such empirical estimations of MC_k, even for linear reservoirs, are inaccurate, especially for larger values of k.

Jaeger (2002a) proved that for *any* recurrent neural network with N recurrent neurons, under the assumption of i.i.d. input stream, MC cannot exceed N. We proved (Rodan & Tino, 2011) (under the assumption of zero-mean i.i.d. input stream) that MC of linear SCR architecture with N reservoir units can be made arbitrarily close to N. In particular, $MC = N - (1 - r^{2N})$, where $r \in (0, 1)$ is the single weight value for all connections in the cyclic reservoir. In order to study the memory capacity structure of linear SCR and the influence of additional shortcuts in CRJ, we first present a novel way of estimation of MC_k directly from the reservoir matrix.

Direct Memory Capacity Estimation for Linear Reservoirs

Given a (one side infinite) i.i.d. zero-mean real-valued input stream $s(..t) = ...\ s(t-3)\ s(t-2)\ s(t-1)\ s(t)$ emitted by a source P, the state (at time t) of the linear reservoir with reservoir weight matrix W and input vector V is

$$x(t) = \sum_{\ell=0}^{\infty} s(t-\ell)\, W^{\ell}\, V$$

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

$$U = R^{-1}\, p^{(k)}, \tag{12}$$

where

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

is the covariance matrix of reservoir activations and

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

The covariance matrix can be evaluated as

$$\begin{aligned} R &= E_{P(s(..t))}\left[\left(\sum_{\ell=0}^{\infty} s(t-\ell)\, W^\ell\, V\right)\cdot\left(\sum_{q=0}^{\infty} s(t-q)\, W^q\, V\right)^T\right] \\ &= E_{P(s(..t))}\left[\sum_{\ell,q=0}^{\infty} s(t-\ell)\, s(t-q)\, W^\ell\, VV^T\, (W^q)^T\right] \\ &= \sum_{\ell,q=0}^{\infty} E_{P(s(..t))}[s(t-\ell)\, s(t-q)]\, W^\ell\, VV^T\, (W^T)^q \\ &= \sigma^2 \sum_{\ell=0}^{\infty} W^\ell\, VV^T\, (W^T)^\ell, \end{aligned} \tag{13}$$

where σ^2 is the variance of the i.i.d. input stream.

Analogously,

$$\begin{aligned} p^{(k)} &= E_{P(s(..t))}\left[\sum_{\ell=0}^{\infty} s(t-\ell)\, s(t-k)\, W^\ell\, V\right] \\ &= \sum_{\ell=0}^{\infty} E_{P(s(..t))}[s(t-\ell)\, s(t-k)]\, W^\ell\, V \\ &= \sigma^2\, W^k\, V. \end{aligned} \tag{14}$$

Provided R is full rank, by (12), (13) and (14), the optimal readout vector $U^{(k)}$ for delay $k \geq 1$ reads

$$U^{(k)} = G^{-1}\, W^k\, V, \tag{15}$$

where

$$G = \sum_{\ell=0}^{\infty} W^\ell\, VV^T\, (W^T)^\ell. \tag{16}$$

The optimal 'recall' output at time t is then

$$\begin{aligned} y(t) &= x^T(t)\, U^{(k)} \\ &= \sum_{\ell=0}^{\infty} s(t-\ell)\, V^T\, (W^\ell)^T\, G^{-1}\, W^k\, V, \end{aligned} \tag{17}$$

yielding

$$\begin{aligned} Cov(s(t-k), y(t)) &= \sum_{\ell=0}^{\infty} E_{P(s(..t))}[s(t-\ell)\, s(t-k)]\, V^T\, (W^\ell)^T\, G^{-1}\, W^k\, V \\ &= \sigma^2\, V^T\, (W^k)^T\, G^{-1}\, W^k\, V. \end{aligned} \tag{18}$$

Since for the optimal recall output $Cov(s(t-k), y(t)) = Var(y(t))$ (Jaeger, 2002a; Rodan & Tino, 2011), we have

$$MC_k = V^T\, (W^k)^T\, G^{-1}\, W^k\, V. \tag{19}$$

Two observations can be made at this point. First, as proved by Jaeger Jaeger (2002a), MC_k constitute a decreasing sequence in $k \geq 1$. From (19) it is clear that MC_k scale as $\|W\|^{-2k}$, where $\|W\| < 1$ is a matrix norm of W. Second, denote the image of the input weight vector V through k-fold application of the reservoir operator W by $V^{(k)}$, i.e. $V^{(k)} = W^k\, V$. Then the matrix $G = \sum_{\ell=0}^{\infty} V^{(\ell)}\, (V^{(\ell)})^T$ can be considered a scaled 'covariance' matrix of the iterated images of V under the reservoir mapping. In this interpretation, MC_k is nothing but the squared 'Mahalanobis norm' of $V^{(k)}$ under such covariance structure,

$$\begin{aligned} MC_k &= (V^{(k)})^T\, G^{-1}\, V^{(k)} \\ &= \|V^{(k)}\|^2_{G^{-1}}. \end{aligned} \tag{20}$$

We will use the derived expressions to approximate the memory capacity of different kinds of (linear) reservoirs to a much greater degree of precision than that obtained through the usual empirical application of the definition in (10) - first generate a long series of i.i.d. inputs and drive with it the reservoir; then train the readout to recover the inputs delayed by k time steps; finish by numerically estimating the statistical moments in (10) using the target values (delayed inputs) and their estimates provided at ESN output.

We will approximate $G = \sum_{\ell=0}^{\infty} V^{(\ell)} \, (V^{(\ell)})^T$ by a finite expansion of the first L terms

$$\hat{G}(L) = \sum_{\ell=0}^{L} V^{(\ell)} \, (V^{(\ell)})^T. \tag{21}$$

We have

$$\begin{aligned} \|V^{(\ell)}\|_2 &\leq \|W^\ell\|_F \cdot \|V\|_2 \\ &\leq \sqrt{N} \cdot \|W^\ell\|_2 \cdot \|V\|_2 \\ &\leq \sqrt{N} \cdot \|W\|_2^\ell \cdot \|V\|_2 \\ &= \sqrt{N} \cdot (\sigma_{max}(W))^\ell \cdot \|V\|_2, \end{aligned} \tag{22}$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ is the (induced) L_2 and Frobenius norm, respectively, and $\sigma_{max}(W)$ is the largest singular value of W. Furthermore,

$$\begin{aligned} \|V^{(\ell)} \, (V^{(\ell)})^T\|_2 &= \|V^{(\ell)}\|_2^2 \\ &\leq N \cdot (\sigma_{max}(W))^{2\ell} \cdot \|V\|_2^2, \end{aligned}$$

and so, given a small $\epsilon > 0$, we can solve for the number of terms $L(\epsilon)$ in the approximation (21) of G so that the norm of contributions $V^{(\ell)} \, (V^{(\ell)})^T$, $\ell > L(\epsilon)$, is less than ϵ. Since $\sigma_{max}(W) < 1$,

$$\begin{aligned} \| \sum_{\ell=L(\epsilon)}^{\infty} V^{(\ell)} \, (V^{(\ell)})^T\|_2 &\leq \sum_{\ell=L(\epsilon)}^{\infty} \|V^{(\ell)} \, (V^{(\ell)})^T\|_2 \\ &\leq N \, \|V\|_2^2 \sum_{\ell=L(\epsilon)}^{\infty} (\sigma_{max}(W))^{2\ell} \\ &= N \, \|V\|_2^2 \, \frac{(\sigma_{max}(W))^{2L(\epsilon)}}{1 - (\sigma_{max}(W))^2}, \end{aligned} \tag{23}$$

we have that for

$$L(\epsilon) > \frac{1}{2} \, \frac{\log \frac{\epsilon \, (1 - (\sigma_{max}(W))^2)}{N \, \|V\|_2^2}}{\log \sigma_{max}(W))}, \tag{24}$$

it holds

$$\| \sum_{\ell=L(\epsilon)}^{\infty} V^{(\ell)} \, (V^{(\ell)})^T\|_2 \leq \epsilon,$$

and so with $L(\epsilon)$ terms in (21), G can be approximated in norm up to a term $< \epsilon$.

The Effect of Shortcuts in CRJ on Memory Capacity

In (Rodan & Tino, 2011) we proved that the 'k-step recall' memory capacity MC_k for the SCR with reservoir weight $r \in (0,1)$ is equal to

$$MC_k = r^{2k}\,(1-r^{2N})\,\zeta_{k \bmod N},$$

where $\zeta_j = r^{-2j}$, $j = 0,1,2,...,N-1$. It follows that for $k \geq 1$,

$$\begin{aligned} MC_k &= r^{2k}\,(1-r^{2N})\,r^{-2\,(k \bmod N)} \\ &= (1-r^{2N})\,r^{2\,[k-(k \bmod N)]} \\ &= (1-r^{2N})\,r^{2N\,(k \operatorname{div} N)}, \end{aligned} \tag{25}$$

where div represents integer division. Hence, for linear cyclic reservoirs with reservoir weight $0 < r < 1$, MC_k is a non-increasing piecewise constant function of k, with blocks of constant value

$$MC_{qN+j} = (1-r^{2N})\,r^{2Nq}, \quad q \geq 0, \quad j \in \{0,1,...,N-1\}. \tag{26}$$

In order to study the effect of reservoir topologies on the contributions MC_k to the memory capacity MC, we first selected three model class representatives (on the validation set) with $N = 50$ linear unit reservoirs on the system identification task (10-th order NARMA), one representative for each of the model classes ESN, SCR and CRJ (jump length 4). Linear and non-linear reservoirs of size 50 had similar performance levels on the NARMA task. To make the MC_k plots directly comparable, we then re-scaled the reservoir matrices W to a common spectral radius $\rho \in (0,1)$. In other words, we are interested in differences in the profile of MC_k for different reservoir types, as k varies. Of course, for smaller spectral radii, the MC contributions will be smaller, but the principal differences can be unveiled only if the same spectral radius is imposed on all reservoir structures.

The memory capacity of the reservoir models was estimated through estimation of MC_k, $k = 1,2,...,200$, in two ways:

1. *Empirical Estimation:* The i.i.d. input stream consisted of 9000 values sampled from the uniform distribution on $[-0.5, 0.5]$. The first 4000 values were used for training, the next 2000 for validation (setting the regularization parameter of

Ridge regression in readout training), and the remaining 3000 values was used for testing the models (prediction of the delayed input values). After obtaining the test outputs, the memory capacity contributions MC_k were estimated according to (10). This process was repeated 10 times (10 runs), in each run a new input series has been generated. Final MC_k estimates were obtained as averages of the MC_k estimated across the 10 runs. This represents the standard approach to MC estimation proposed by Jaeger (2002a) and used in the ESN literature (Fette & Eggert, 2005; Ozturk et al., 2007; Verstraeten et al., 2007; Steil, 2007).

2. *Theoretical Estimation:* The MC contributions MC_k were calculated from (19), with G approximated as in (21). The number of terms L has been determined according to (24), where the precision parameter ϵ was set to $\epsilon = 10^{-60}$.

Figures 6(A) and (B) present theoretical and empirical estimates, respectively, of MC_k for $\rho = 0.8$. Analogously, Figures 6(C) and (D) show theoretical and empirical estimates of MC_k for $\rho = 0.9$. The direct theoretical estimation (Figures 6(A,C)) is much more precise than the empirical estimates (Figures 6(B,D)). Note the clear step-wise behavior of MC_k for SCR predicted by the theory (eq. (26)). As predicted, the step size is $N = 50$. In contrast, the empirical estimations of MC_k can infer the first step at $k = 50$, but lack precision thereafter (for $k > 50$). Interestingly, SCR topology can keep information about the last $N - 1$ i.i.d. inputs to a high level of precision ($MC_k = 1 - r^{2N}$, $k = 1, 2, ..., N - 1$), but then loses the capacity to memorize inputs more distant in the past in a discontinuous manner (jump at $k = N = 50$). This behavior of MC_k for SCR is described analytically by eq. (26). In contrast, as a consequence of 'cross-talk' effects introduced by jumps in CRJ, the MC contributions MC_k start to rapidly decrease earlier than at $k = N$, but the reservoir can keep the information about some of the later inputs better than in the case of SCR (roughly for $50 \leq k \leq 60$). In the case of ESN, the MC_k values decrease more rapidly than in the case of both SCR and CRJ. Using the standard empirical estimation of MC_k, such a detailed behavior of memory capacity contributions would not be detectable. To demonstrate the potential of our method, we show in Figures 7(A,B) theoretically determined graphs of MC_k for delays up to $k = 400$ using $\rho = 0.8$ (A) and $\rho = 0.9$ (B).

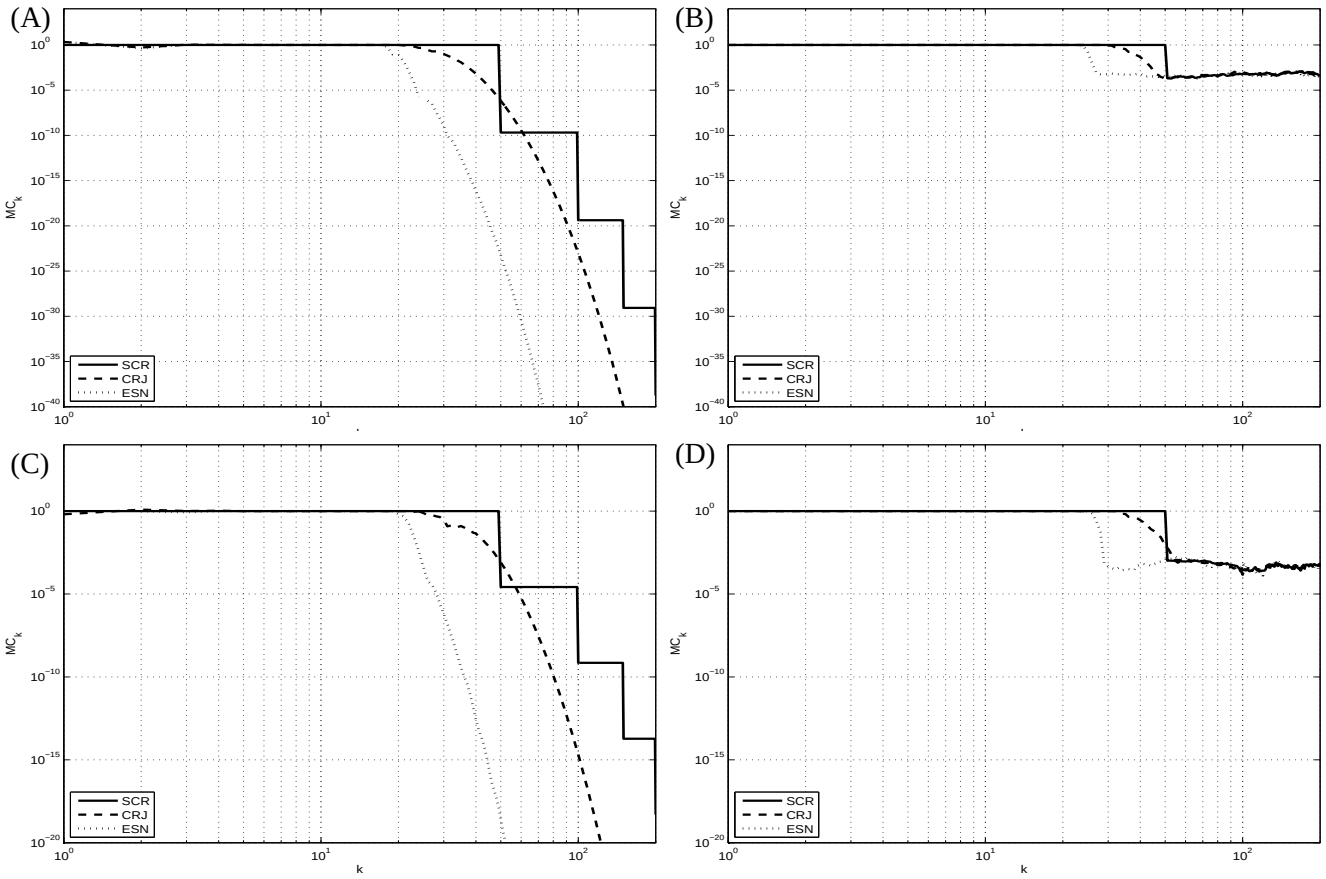

Figure 6: Theoretical (A,C) and Empirical (B,D) k-Delay MC of ESN (dotted line), SCR (solid line), and CRJ (dashed line) for Delays $k = 1, ..., 200$. The Graphs of MC_k are Shown for $\rho = 0.8$ (A,B) and $\rho = 0.9$ (C,D).

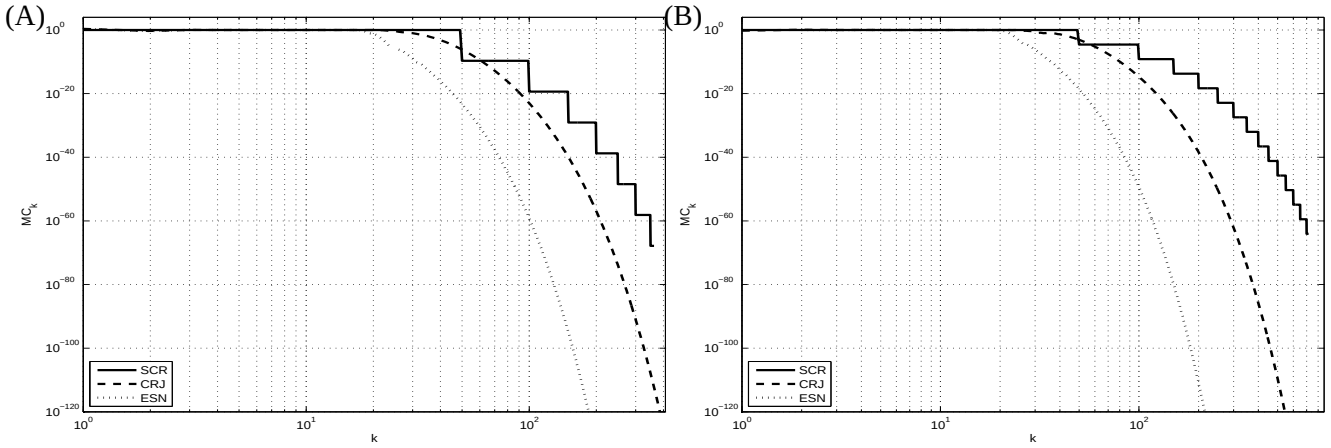

Figure 7: Theoretical k-Delay MC of ESN (dotted line), SCR (solid line), and CRJ (dashed line) for Delays $k = 1, ..., 400$. The Graphs of MC_k are Shown for $\rho = 0.8$ (A) and $\rho = 0.9$ (B).

6.3 Lyapunov Exponent

Verstraeten et al. (2007) suggest to extend numerical calculation of the well known Lyapunov exponent characterization of (ergodic) autonomous dynamical systems to input-driven systems. The same idea occurred previously in the context of recurrent neural networks for processing symbolic streams (Tabor, 2001) While the reservoir is driven by a particular input sequence, at each time step the local dynamics is linearized around the current state and the Lyapunov spectrum is calculated. The largest exponents thus collected are then used to produce an estimate of the average exponential divergence rate of nearby trajectories along the input-driven reservoir trajectory. Even though for input-driven systems this is only a heuristic measure[10], it nevertheless proved useful in suggesting the 'optimal' reservoir configuration across several tasks (Verstraeten et al., 2007). Indeed, in our experiments the selected ESN configurations in the laser, NARMA and speech recognition tasks all lead to pseudo-Lyapunov exponents ranging from 0.35 to 0.5. As in (Verstraeten et al., 2007), the exponents are positive, suggesting local exponential divergence along the sampled reservoir trajectories, and hence locally expanding systems (at least in one direction). For our simple reservoir architectures, SCR and CRJ, the selected configurations across the data sets also lead to similar pseudo-Lyapunov exponents, but this time in the negative range. For example the CRJ exponents ranged from -0.4 to -0.25. All exponents for the selected architectures of both SCR and CRJ were negative, implying contractive dynamics.

To study the pseudo-Lyapunov exponents of the selected reservoir architectures along the lines of (Verstraeten et al., 2007), for each data set, the reservoir matrix of each selected model representative from ESN, SCR and CRJ was rescaled so that the spectral radius ranged from 0.1 to 2. The resulting pseudo-Lyapunov exponents are shown in Figure 8 for the NARMA (A), laser (B), and speech (C) data sets. The vertical lines denote the spectral radii of the selected 'optimal' model representatives and black markers show the corresponding exponents. Interestingly, for all data sets, the pseudo-Lyapunov exponent lines of ESN are consistently on top of the SCR ones, which in turn are on top of those of CRJ. This ranking holds also for the selected model repre-

[10] Deep results of autonomous systems theory e.g. linking positive Lyapunov exponents to topological entropy (Pesin Theorem) no longer apply, nor do apply traditional notions of 'chaos' and 'order' developed in the context of autonomous systems.

sentatives on different tasks. Our results show that a reservoir model can have superior performance without expanding dynamics. In fact, in our experiments the CRJ reservoir achieved the best results while having on average contractive dynamics along the sampled trajectories and the least pseudo-Lyapunov exponent.

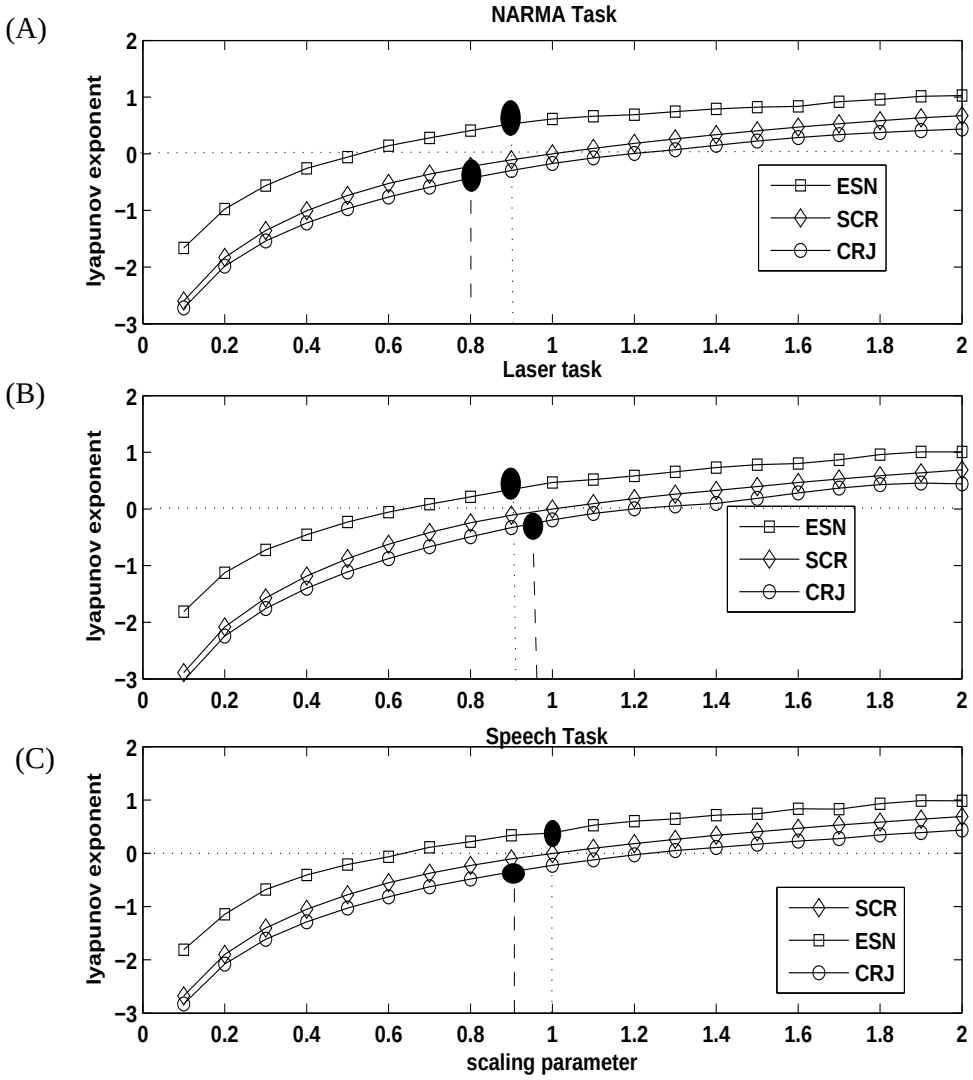

Figure 8: Pseudo-Lyapunov Exponents for ESN, SCR, and CRJ on the NARMA (A), Laser (B), and Speech Recognition (C) Tasks. The Vertical Lines Denote the Spectral Radii of the Selected 'Optimal' Model Representatives and Black Markers Show the Corresponding Exponents.

7 Conclusion

A large variety of reservoir computing models have been proposed, differing in reservoir generation and readout formulation (Lukosevicius & Jaeger, 2009). Echo state networks (ESN) (Jaeger, 2001) typically have a linear readout and a reservoir formed by a fixed recurrent neural network type dynamics. *Liquid state machines* (LSM) (Maass et al., 2002) have also mostly linear readout and the reservoirs are driven by the dynamics of a set of coupled spiking neuron models. *Fractal prediction machines* (FPM) (Tino & Dorffner, 2001) for processing symbolic sequences have fixed affine state transitions and the readout is constructed as a collection of multinomial distributions over next symbols. Continuously adaptable reservoirs were suggested by Steil (2007). Many other forms of reservoirs can be found in the literature (e.g. (Jones et al., 2007; Deng & Zhang, 2007; Dockendorf et al., 2009; Bush & Anderson, 2005; Ishii et al., 2004; Schmidhuber et al., 2007; Ajdari Rad et al., 2008)). However, exactly what aspects of reservoirs are responsible for their often reported superior modelling capabilities (Jaeger, 2001, 2002a,b; Jaeger & Hass, 2004; Maass et al., 2004; Tong et al., 2007) is still unclear.

Traditionally, reservoirs have been constructed in a randomized manner. Moreover, there have been several attempts to address the question of what exactly is a 'good' reservoir for a given application (Hausler et al., 2003; Ozturk et al., 2007). In our previous study (Rodan & Tino, 2011) we considered a very simple deterministically constructed cyclic reservoir (SCR). Besides eliminating the problem of non-transparency and trail-and-error construction of standard randomized ESN, the simple deterministically constructed SCR topologies were shown to yield comparable results to ESN on a variety of temporal tasks. In this paper we extended this study in several aspects:

1. We introduced a novel simple deterministic reservoir model, Cycle Reservoir with Jumps (CRJ) with highly constrained weight values, that has superior performance to standard ESN on four temporal tasks of different origin and characteristics.

2. We studied the effect of eigenvalue distribution of the reservoir matrix on the model performance. It has been suggested that a uniform coverage of the unit disk by such eigenvalues can lead to superior model performances. We showed that

this is not necessarily so. Despite having highly constrained eigenvalue distribution the CRJ consistently outperformed ESN with much more uniform eigenvalue coverage of the unit disk.

3. We presented a new framework for determining short term memory capacity MC of linear reservoir models to a high degree of precision. Using the framework we showed the effect of shortcut connections in the CRJ reservoir topology on its memory capacity. Due to cross-talk effects introduced by the jumps in CRJ, the MC contributions start to rapidly decrease earlier than in the case of SCR, but unlike in SCR, the decrease in MC_k in CRJ is gradual, enabling the reservoir to keep more information about some of the later inputs.

4. Through the study of pseudo-Lyapunov exponents we showed that even though (unlike ESN) the simple CRJ reservoirs have (average) contractive dynamics, they achieved consistently the best performance. This poses a interesting open question as to whether and in what contexts the "edge-of-chaos" hypothesis can be applied to reservoir computations.

We believe that if given a choice whether to construct a model in a randomized or completely deterministic manner, having guarantees of 'similar' performance levels, it is more advisable to go for the latter. Besides the advantages mentioned above, in our framework the important elements of the model structure have a chance to emerge. For example, we show that even though simple unidirectional cycle with fixed weight (SCR model) is already competitive, adding regular bidirectional shortcuts (of the same weight) originating and ending in few higher-clustering coefficient nodes (CRJ model), brings potentially huge performance improvements (and sometimes significantly beats ESN). Such an insight could not be obtained using traditional randomized reservoir generation. This opens new research questions as to exactly why such a jump modification has this effect. Such focused research program would not originate from studies consistently using randomized reservoir constructions. On the other hand, using randomized reservoir construction can have beneficial effects on model evaluation - in contrast to deterministically constructed reservoirs, one may need a smaller pool of different tasks to get the same statistical significance[11].

[11] We thank the anonymous reviewer for pointing this out.

Compared with traditional ESN, specific reformulations of reservoir models can often achieve improved performances (Steil, 2007; Xue et al., 2007; Deng & Zhang, 2007), at the price of even less transparent models and less interpretable dynamical organization. We propose that in order to quantify the benefit of the potentially complex current or future reservoir formulations, such models should be compared with our simple, deterministically constructed CRJ model that, as shown in this study, has a potential to significantly outperform the traditional ESN. Furthermore, it seems that characterizations of reservoirs in terms of memory capacity, eigenvalue decomposition of the reservoir weight matrix or pseudo-Lyapunov exponents cannot easily capture what makes reservoirs great temporal modelling tools. Reservoirs are non-linear non-autonomous dynamical systems that are difficult to characterize by linearization techniques (eigenspectrum), or methods not directly representing task-related useful temporal structure in the input driving stream (memory capacity). Theory and practice of deep reservoir characterizations that can be directly linked to their performance is an open problem and a matter for our future study.

Acknowledgments

We thank the anonymous reviewers for many helpful comments. This work has been supported by the UK BBSRC grant RRAE14541 and EU FP7 iSense grant (contract no. INSFO-ICT-270428).

Appendix

In this appendix we show detailed parameter settings of the selected model representatives in our experiments. Details of parameter values of models used in section 4.2 are provided in Table 7. Table 8 reports parameters for models used in the comparison experiment with SWNR (section 5). Finally, we report parameter values of the selected hierarchical extension (CRHJ) of the CRJ model in Table 9 (section 5).

Table 7: Parameter Values for the Selected ESN, SCR and CRJ Model Representatives with Reservoirs of N units.

Dataset	ESN	SCR	CRJ
laser $N = 200$	$con = 0.2$, $\lambda = 0.95$, $a = 1$	$v = 0.85$, $r_c = 0.7$	$v = 0.9$, $r_c = 0.7$, $r_j = 0.4$, $\ell = 5$
NARMA $N = 200$	$con = 0.15$, $\lambda = 0.85$, $a = 0.1$	$v = 0.05$, $r_c = 0.8$	$v = 0.05$, $r_c = 0.7$, $r_j = 0.5$, $\ell = 5$
speech $N = 200$	$con = 0.4$, $\lambda = 0.95$, $a = 1$	$v = 1$, $r_c = 0.95$	$v = 1$, $r_c = 0.9$, $r_j = 0.4$, $\ell = 13$
memory and nonlinear mapping task $N = 100$	$con = 0.2$, $\lambda = 0.95$, $a = 0.05$	$v = 0.025$, $r_c = 0.7$	$v = 0.025$, $r_c = 0.8$, $r_j = 0.3$, $\ell = 24$

Table 8: Parameter Values for the Selected ESN, SWNR, SCR and CRJ Model Representatives (Reservoir Size $N = 500$).

Dataset	ESN	SWNR	SCR	CRJ
laser	$con = 0.15$, $\lambda = 0.9$, $a = 1$	$\lambda = 5.5$, $a = 1$	$v = 0.7$, $r_c = 0.75$	$v = 0.7$, $r_c = 0.75$, $r_j = 0.15$, $\ell = 10$
NARMA	$con = 0.2$, $\lambda = 0.95$, $a = 0.1$	$\lambda = 2$, $a = 0.2$	$v = 0.05$, $r_c = 0.8$	$v = 0.1$, $r_c = 0.8$, $r_j = 0.5$, $\ell = 21$

Table 9: Parameter Values for the Selected CRHJ Model Representative (Reservoir Size $N = 100$).

Dataset	CRHJ
NARMA	$v = 0.05$, $r_c = 0.6$, $r_{j_1} = 0.05$, $r_{j_2} = 0.4$, $r_{j_3} = 0.25$
laser	$v = 1$, $r_c = 1$, $r_{j_1} = 0.55$, $r_{j_2} = 0.4$, $r_{j_3} = 0.1$

References

Ajdari Rad, A., Jalili, M., & Hasler, M. (2008). Reservoir optimization in recurrent neural networks using kronecker kernels. In *IEEE ISCAS*.

Atiya, A., & Parlos, A. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11, 697–709.

Bush, K., & Anderson, C. (2005). Modeling reward functions for incomplete state representations via echo state networks. In *Proceedings of the International Joint Conference on Neural Networks, Montreal, Quebec*.

Bsing, L.,Schrauwen, B., & Legenstein, R.A. (2010). Connectivity, Dynamics, and Memory in Reservoir Computing with Binary and Analog Neurons. *Neural Computation*, 22(5), 1272–1311.

Bertschinger,N. & Natschlager, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7), 1413–1436.

Cernansky, M., & Makula, M. (2005). Feed-forward echo state networks. In *In Proceedings of the IEEE International Joint Conference on Neural Networks,(IJCNN 2005)*,1479-1482.

Cernansky, M., & Tino, P. (2008). Predictive modelling with echo state networks. In *Proceedings of the 18th international conference on Artificial Neural Networks*, 778-787.

Deng, Z., & Zhang, Y. (2007). Collective behavior of a small-world recurrent neural system with scale-free distribution. *IEEE Transactions on Neural Networks*, 18(5), 1364–75.

Dockendorf, K., Park, I., Ping, H., Principe, J.C., & DeMarse, T. (2009). Liquid state machines and cultured cortical networks: The separation property. *Biosystems*, 95(2), 90–97.

Dutoit, X., Schrauwen, B., Van Campenhout, J., Stroobandt, D., Van Brussel, H., & Nuttin, M. (2009). Pruning and regularization in reservoir computing. *Neurocomputing*, 72, 1534–1546.

Fette, G., & Eggert, J. (2005). Short term memory and pattern matching with simple echo state networks. In *In Proc. of ICANN*, 1318.

Holzmann, G., & Hauser, H. (2009). Echo state networks with filter neurons and a delay and sum readout. *Neural Networks*, 32(2), 244–256.

Hausler, S., Markram, M., & Maass, W. (2003). 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)*, 8(4), 39–50.

Ishii, K., van der Zant, T., Becanovic, V., & Ploger, P. (2004). Identification of motion with echo state network. In *In Proceedings of the OCEANS 2004 MTS/IEEE -TECHNO-OCEAN Conference*, volume 3, pages 1205-1210.

Jaeger, H. (2001). The ”echo state” approach to analysing and training recurrent neural networks. Technical report gmd report 148, German National Research Center for Information Technology.

Jaeger, H. (2002a). Short term memory in echo state networks. Technical report gmd report 152, German National Research Center for Information Technology.

Jaeger, H. (2003). Adaptive nonlinear systems identification with echo state network. *Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA.*, 15, 593–600.

Jaeger, H. (2005). Reservoir riddles: Suggestions for echo state network research. In *In: Proceedings of International Joint Conference on Neural Networks IJCNN 2005*, Montreal, Canada. (1460-1462).

Jaeger, H. (2002b). A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the ”echo state network” approach. Technical report gmd report 159, German National Research Center for Information Technology.

Jaeger, H. (2007). Discovering multiscale dynamical features with hierarchical echo state networks. Technical report, Jacobs University technical report Nr. 10.

Jaeger, H., & Hass, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, 304, 78–80.

Jaeger, H., Lukosevicius, M., Popovici, D., & Siewert, U. (2007a). Optimisation and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3), 335–352.

Jaeger, H., Maass, W., & Principe, J.C. (2007b). Special issue. *Neural Networks*, 20.

Legenstein, R., & Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3), 323–334.

Jones, B., Stekel, D., Rowe, J., & Fernando, C. (2007). 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.

Lukosevicius, M. & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127–149.

Lyon, R.F. (1982). A computational model of filtering, detection and compression in the cochlea. In *In Proceedings of the IEEE ICASSP*, pages 1282-1285.

Maass, W., Natschlager, T., & Markram, H. (2002). Real-time computing without sTable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531–2560.

Maass, W., Natschlager, T., & Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology*, 98(4-6), 315–330.

Ozturk, M. C., Xu, D., & Principe, J.C. (2007). Analysis and design of echo state network. *Neural Computation*, 19(1), 111–138.

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

Rodan, A. & Tino, P. (2011). Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, 22(1), 131–144. 2011.

Schmidhuber, J., Wierstra, D., Gagliolo, M., & Gomez, F. (2007). Training recurrent networks by evolino. *Neural Computation*, 19, 757–779.

Schrauwen, B., Defour, J., Verstraeten, D. & Van Campenhout, J.M. (2007). 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.

Schrauwen, B., Buesing, L., & Legenstein, R. A. (2008a). On computational power and the order-chaos phase transition in reservoir computing. In *Neural Information Processing Systems (NIPS)*, 425-1432.

Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J., & Stroobandt, D. (2008b). Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7-9), 1159–1171.

Skowronski, M.D., & Harris, J.G. (2006). 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, 3153-3156.

Steil, J. (2007). Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. *Neural Networks*, 20, 353–364.

Steil, J. (2004). Backpropagation-decorrelation: Recurrent learning with o(n) complexity. *In Proc. of the International Joint Conference on Neural Networks, IJCNN '04*, volume 2, 843-848.

Tabor, W. (2002). The Value of Symbolic Computation. *Ecological Psychology*, 14(1-2), 21–51.

Tino, P. & Dorffner, G. (2001). Predicting the future of discrete sequences from fractal representations of the past. *Machine Learning*, 45(2), 187–218.

Tong, M. H., Bicket, A.D. , Christiansen, E.M., & Cottrell, G.W. (2007). Learning grammatical structure with echo state network. *Neural Networks*, 20, 424–432.

Triefenbach, F., Jalalvand, A., Schrauwen, B., & Martens, J. (2010). Phoneme recognition with large hierarchical reservoirs. In *Advances in Neural Information Processing Systems*, Vol. 23, pp.9.

Verstraeten, D., Schrauwen, B., D'Haene, M. , & Stroobandt, D. (2006). The unified reservoir computing concept and its digital hardware implementations. *In Proc. LATSIS*, pages 139-140.

Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20, 391–403.

Verstraeten, D., Dambre, J., Dutoit, X. & Schrauwen, B. (2010). Memory versus nonlinearity in reservoirs. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE Press.

Watts, D.J. & Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393 (6684): 409–10.1998.

Wyffels, F., Schrauwen, B., & Stroobandt, D. (2008). Stable output feedback in reservoir computing using ridge regression. In *Proceedings of the 18th international conference on Artificial Neural Networks*, pp.808-817.

Xue, Y. Yang, L. & Haykin, S. (2007). Decoupled echo state networks with lateral inhibition. *Neural Networks*, 20, 365–376.

Zhang, B., & Wang, Y. (2008). Echo state networks with decoupled reservoir states. In *18th IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2008)*.