Neural Networks 36 (2012) 35–45

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Balanced echo state networks

Danil Koryakin, Johannes Lohmann, Martin V. Butz *

Cognitive Modeling, Department of Computer Science, University of Tübingen, Sand 14, 72076 Tübingen, Germany

ARTICLE INFO

Article history:
Received 28 September 2011
Received in revised form 14 June 2012
Accepted 14 August 2012

Keywords:
Echo state network
Output feedback
Multiple superimposed oscillator
Recurrent neural networks
Dynamic reservoir

ABSTRACT

This paper investigates the interaction between the driving output feedback and the internal reservoir dynamics in echo state networks (ESNs). The interplay is studied experimentally on the multiple superimposed oscillators (MSOs) benchmark. The experimental data reveals a dual effect of the output feedback strength on the network dynamics: it drives the dynamic reservoir but it can also block suitable reservoir dynamics. Moreover, the data shows that the reservoir size crucially co-determines the likelihood of generating an effective ESN. We show that dependent on the complexity of the MSO dynamics somewhat smaller networks can yield better performance. Optimizing the output feedback weight range and the network size is thus crucial for generating an effective ESN. With proper parameter choices, we show that it is possible to generate ESNs that approximate MSOs with several orders of magnitude smaller errors than those previously reported. We conclude that there appears to be still much more potential in ESNs than previously thought and sketch-out some promising future research directions.

1. Introduction

Reservoir computing is a relatively new approach to design and train recurrent neural networks. One of the most promising reservoir computing approaches is the echo state network (ESN) (Jaeger, 2001, 2003). ESNs offer some advantages over classic recurrent neural networks. Their biggest advantages are the training speed and the simplicity. Learning can be realized by any linear regression algorithm, providing optimal weights for the given ESN. ESNs were successfully applied for solving different problems ranging from static problems, such as pattern classification (Emmerich, Reinhart, & Steil, 2010), to various types of dynamic problems, such as modeling time-series (Jaeger, 2001). Related publications report high performance in real-world applications (Jaeger & Haas, 2004; Rodan & Tino, 2011; Verstraeten, Schrauwen, DHaene, & Stroobandt, 2007) as well as in commonly accepted benchmark problems (Jaeger, 2003; Jaeger, Lukosevicius, Popovici, & Siewert, 2007; Rodan & Tino, 2011; Verstraeten et al., 2007). However, there are also problems where these neural networks either do not work at all or lead to poor generalization when compared with other techniques. The multiple superimposed oscillator (MSO) is one of such difficult benchmark problems.

Particular properties of the MSO dynamics make it attractive to be modeled with ESNs. First of all, the dynamics represent a good benchmark for comparisons with various other algorithms. In contrast to other typical benchmarks, such as Mackey–Glass, the MSO dynamics are well-defined and do not depend on initial conditions or particular time scales to generate particular time series. Moreover, the MSO problem is challenging for ESNs. Some publications consider modeling MSO dynamics with standard ESNs, as initially proposed by Jaeger (2001) as nearly unsolvable. For example, in Xue, Yang, and Haykin (2007) and in Schmidhuber, Wierstra, Gagliolo, and Gomez (2007) it was claimed that standard ESNs are not able to model more complex MSO dynamics, hypothesizing that the inherent coupling of the neurons in the dynamic reservoir precludes the effective modeling of multiple target signals simultaneously.

Several approaches were proposed to solve this problem: some use the standard ESNs and directly aim at decoupling the reservoir neurons (Xue et al., 2007); others propose other neural network architectures, succeeding in modeling MSOs to varying degrees. Probably the most remarkable approaches with ESNs are presented in Holzmann and Hauser (2010) and Xue et al. (2007). Xue et al. (2007) proposed decoupled echo state networks (DESN), which consist of sub-reservoirs. A sub-reservoir is a group of neurons that are randomly connected to each other, as in the standard ESN proposed by Jaeger (2001). However, there are no direct connections between neurons of different sub-reservoirs but the sub-reservoirs are connected through a special decoupling mechanism, yielding good results on the MSO with two sine waves. In the networks presented by Holzmann and Hauser (2010), each reservoir neuron of the standard ESN was endowed with an additional infinite impulse response filter. Since the filter was parameterized differently for every reservoir neuron, a variety of

* Corresponding author. Tel.: +49 7071 29 7 0429.
E-mail address: martin.butz@uni-tuebingen.de (M.V. Butz).

0893-6080/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2012.08.008

echo states were generated in the dynamic reservoir, consequently yielding good results in the MSO with two, three, four, and five sine waves. Čerňanský and Tiňo (2008) investigated how variations of the reservoir topology influence the performance of ESNs. Among other benchmarks, they tested their alternative ESN and related architectures also on the MSO problem with two sine waves. Their results highlight that the ability of the reservoir to preserve a sufficient part of the input history is crucial for high performance. The same target dynamics were also successfully modeled by "Evolino" (Schmidhuber et al., 2007). The Evolino network consists of a number of memory cells, which are randomly connected to each other. Similar to the approach of Holzmann and Hauser (2010), every memory cell has a special predefined structure. However, memory cells are more complex as they consist of several neurons and can maintain an infinitely long internal activity. The weights of the connections between the memory cells were trained by means of evolutionary techniques.

Sometimes evolutionary algorithms were also used to evolve ESNs. For instance Roeschies and Igel (2010) applied an evolutionary algorithm to evolve standard ESNs. The authors proposed a number of operators that were used to find the most suitable dynamic reservoirs for modeling a particular time series. This approach was tested on the multiple superimposed oscillators with two, five and eight sine waves. In the case of two superimposed sine waves, Roeschies and Igel (2010) succeeded in evolving an ESN that yielded a very low test error with a reservoir of only three neurons. Unfortunately, the authors provided neither the structure of this network nor an analysis of how such a small network was able to model the target dynamics so precisely. The authors nonetheless emphasized that a proper choice of the spectral radius is crucial for accurately modeling MSO dynamics with ESNs.

In this work we show that standard ESNs can reach high accuracy when modeling MSOs with up to eight superimposed sine waves. We furthermore show that modeling MSOs successfully neither relies on a particular neural network architecture, nor does it require computationally rather expensive evolutionary techniques. However, a correct choice of the network parameters plays a crucial role in the achievable performance. In particular, we show that a simple optimization approach of the output feedback weights can enormously increase the modeling accuracy. These weights determine the extent of activity that flows back from the output layer into the dynamic reservoir. We show that the output feedback weights and possibly the input weights strongly pre-determine the reservoir dynamics available for computing the network output. While it had been suggested that these weights can be rather freely chosen without affecting the echo property significantly (Jaeger, 2002b), our experiments show that output feedback weight ranges need to be chosen with care. Moreover, we show that the probability of generating effective networks also depends on the chosen network size but remains nearly constant for different degrees of sparsity.

In the next section we give a short introduction to ESNs and the notation used herein. Section 3 presents an analysis of the interaction between the reservoir dynamics and the driving signals of the dynamic reservoir. The description of the experimental setup and the obtained results are presented in Section 4. Section 5 summarizes the best time series approximations obtained and compares them with related approaches. Section 6 summarizes the insights gained and concludes the paper.

2. ESN in a Nutshell

ESNs were initially proposed by Jaeger (2001). Their structure and a training approach are comprehensively described in a tutorial (Jaeger, 2002b). In the current section we reiterate the most important facts about them. ESNs are recurrent neural networks, which possess a number of features different from other types of recurrent neural networks. Special features of ESNs are a pre-wired dynamic reservoir and a very simple training procedure, which only optimizes the output weights.

The general structure of an ESN is shown in Fig. 1. The network has three groups of neurons: input neurons, output neurons, and reservoir neurons. Input neurons are optional. The interconnected reservoir neurons constitute the dynamic reservoir. The input neurons are connected to the dynamic reservoir and can be also directly connected to the network's output neurons. The output neurons can be connected to the reservoir via output feedback connections. Recurrent connections amongst the output neurons and connections between different output neurons are also possible.

The dynamic reservoir is the first key structure of an ESN ("RESERVOIR" in Fig. 1). It represents an environment where multiple continuous signals are unfolding over time. These signals are individual dynamics, which are combined to compute the network output. The dynamic reservoir is required to adhere to an *echo state property*, which ensures that any dynamic activity in the network attenuates over time. This is typically accomplished by first generating randomly weighted recurrent connections within the reservoir. The resulting weights in the weight matrix are then divided by the largest eigenvalue of the matrix and multiplied by λ^*, consequently yielding a matrix in which the largest absolute eigenvalue, i.e., the *spectral radius* equals λ^*. While the echo state property prevents the uncontrolled increase of the reservoir dynamics, the dynamics are also designed to maintain a particular dynamic activity over time serving as a tappable memory pool for generating the target output.

When input and output activities are fed into the dynamic reservoir, they can be considered as *external driving signals*. The signals excite the dynamic reservoir and are indispensable for memorization. The driving signals can come from the input neurons via the input connections (signal u(n) in Fig. 1) and from the output neurons via the feedback connections (signal $y(n-1)$ in Fig. 1). Seeing that the echo state property leads to activity attenuation over time, either input or output activities have to be induced into the dynamic reservoir. Which kind of connections should be used to drive the reservoir depends on the problem at hand.

The state of the dynamic reservoir at a certain point in time can be described by the activities of the reservoir neurons $\mathbf{x}$. In each iteration, these activities are updated as follows:

$$\mathbf{x}(n+1) = \mathbf{f}\left(\mathbf{W}_{IN}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}_{OFB}\mathbf{y}(n)\right), \quad (1)$$

where bold font letters denote vectors. In particular, $\mathbf{x}(n)$ encodes the activities of the reservoir neurons at time n, $\mathbf{u}(n)$ refers to the input activities, $\mathbf{y}(n)$ encodes the output activity, $\mathbf{f}$ is the component-wise transfer function (usually a sigmoid), and $\mathbf{W}_{IN}$, $\mathbf{W}$, and $\mathbf{W}_{OFB}$ denote the weight matrices for connections from the input neurons to the reservoir neurons, recurrent connections within the reservoir, and output feedback connections from the previous output activity back into the reservoir, respectively. As stated above, output feedback connections as well as input connections are optional, dependent on the time series problem tackled.

The second key structure of ESNs are the connections from the reservoir neurons to the output neurons. The output neurons may be considered as a readout layer for the dynamic reservoir. The connections to the output layer are responsible for translating the individual dynamics in the reservoir to the current output signal of the ESN. A learning algorithm determines the output weights by usually minimizing the summed squared differences of the network output when compared to the target dynamics. An advantage of ESNs is that they provide an opportunity to use

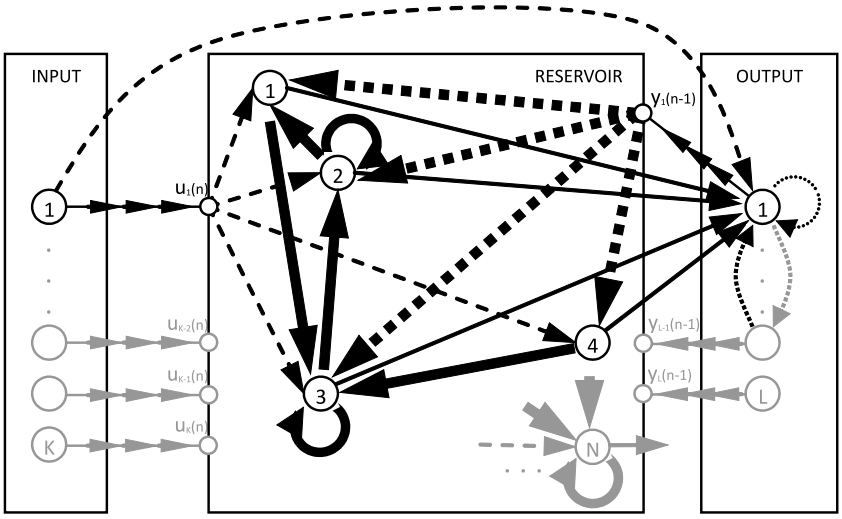

Fig. 1. General structure of an ESN.

any linear regression algorithm to find a matrix with the optimal output weights.

Let $\mathbf{W}_{OUT}$ denote the unique solution of the matrix equation

$$\mathbf{W}_{OUT} = \mathbf{M}^{-1}\mathbf{T}, \tag{2}$$

where $\mathbf{M}^{-1}$ is the inverse matrix of the reservoir activities over neurons and $\mathbf{T}$ is a matrix of target output values. Every row of these matrices contains the status of the dynamic reservoir and the desired output of the network at a certain time step of the training sequence, respectively. It is important to note that a certain number of initial steps are discarded during training (so-called "washout phase"). This is done to reduce the influence of arbitrarily chosen initial states of the dynamic reservoir on the solution of (2). Solving (2) is rather straightforward, for instance, by using the pseudoinverse. Further details about ESNs and the involved matrix computations can be found in the literature (Jaeger, 2002b).

3. Balanced network dynamics

In this section we put forward some theoretical considerations on how the input weights and the output feedback weights affect the performance of ESNs. We formalize the interaction between the reservoir dynamics and the output feedback at the level of a single reservoir neuron. To the best of our knowledge, the choice of the input and output feedback weights have not been deeply investigated. For example, in the tutorial on training ESNs (Jaeger, 2002b) only minor importance was given to these weights and only a short explanation of how these weights may influence the network output was provided.

The following quantitative analyses provides deeper insights into the computation of the states of the reservoir neurons. We discuss why a careful tuning of these weights is necessary to improve the performance of an ESN. Moreover, we define a balanced reservoir neuron as a neuron that experiences a similarly strong influence from both the reservoir neurons and the input and output neurons. Our analysis is confirmed by a detailed experimental analysis, which is provided in the subsequent section.

We consider an ESN without input neurons to analyze the interaction between the output feedback and the reservoir dynamics. An example of this neural network is shown in Fig. 2. In the analysis we refer to an arbitrary neuron of the network's reservoir by the letter *A*. The activation of neuron *A* is determined by a function that has two arguments: the activations of other reservoir neurons that are directly connected to *A* and the network output computed

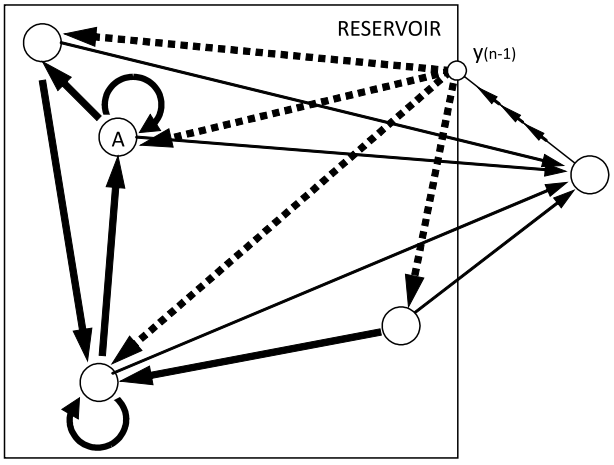

Fig. 2. An ESN whose activity is only driven by the output feedback connections.

in the last time step. For neuron *A*, the update function (1) has the form

$$x_A(n+1) = f\left(\sum_{i=1}^{N} w_{Ai}x_i(n) + w_{Ay}y(n)\right), \tag{3}$$

where the first summand denotes the contribution from the reservoir dynamics, while the second summand denotes the contribution from the output feedback. Both influence the state of each reservoir neuron. Usually ESNs have sparse reservoirs, where sparsity means that not all reservoir neurons are connected to each other directly. The weights of all network connections are random values taken uniformly from a certain zero-centered interval. Most of the time, this interval is chosen separately for the reservoir weights, the input weights, and the output feedback weights. The randomness of the weights supposes that some connections can have (close-to) zero weights. In the analysis below, we assume that neuron *A* is connected to all *N* other reservoir neurons and has a non-zero output feedback weight w_{Ay}, where we consider without loss of generality a single output feedback value *y*. For simplicity of notation in the text below, we drop the time step index, yielding

$$x_A \leftarrow f\left(\sum_{i=1}^{N} w_{Ai}x_i + w_{Ay}y\right). \tag{4}$$

We will show that the variety of states of a reservoir neuron depends on the relative strength of the external *driving signal*

compared to the internal *recurrent signal*. Driving signals are caused by output feedback connections and input connections, whereas recurrent signals are determined by the internal network dynamics. To define the two influences more formally for neuron A, we denote the contribution from the reservoir dynamics with

$$C^A_{RD} = \sum_{i=1}^{N} w_{Ai} x_i, \tag{5}$$

and the contribution from the output feedback with

$$C^A_{OFB} = w_{Ay} y. \tag{6}$$

We propose that a good balance between the two influences is established when

$$\sigma^2(C^A_{RD}) \approx \sigma^2(C^A_{OFB}), \tag{7}$$

where σ^2 is the (expected) variance of the respective contributions to a neuron's activity. The variances may be computed on a given sequence after discarding the initial transient. Assuming zero-centered activity distributions, the condition essentially specifies that output feedback and internal reservoir dynamics should both play equally strong roles in determining the unfolding internal reservoir activities.

The fulfillment of condition (7) depends on the target dynamics as well as on the parameters of the ESN. In a trained network, it may be assumed that the output signal y is very close to the target dynamics. Besides y, the feedback weight w_{Ay} as well as the internal weights and activities w_{Ai} and x_i determine the respective activity influences. Particularly the chosen weights strongly depend on the macro parameters used for the random generation of an ESN. These macro parameters are the spectral radius λ^*, the activation function of the reservoir neurons, and the interval for the random generation of the output feedback weights $[-W_{OFB}, +W_{OFB}]$.

We consider the contribution from the reservoir dynamics first. For the analysis, we assume a non-linear reservoir, whose neurons have the activation function TANH. Due to this activation function, activity values x_i are in $[-1, +1]$. To avoid the network from blowing up or saturating its internal dynamics, the spectral radius is usually set to values below one ($\lambda^* < 1$). Due to this setting, the weight values w_{Ai} will be on average in the range of λ^* — even if highly unbalanced initializations can violate this aspect significantly. To provide a high accuracy in the applications for modeling time-series, the reservoir neurons should operate without going into saturation. Thus, their activity should stay in a relatively small interval around zero. Under this condition, the contributions C^A_{RD} must be reduced by "pushing" the variances of the separate terms $(w_{Ai}x_{Ai})$ towards zero. Since both, w_{Ai} and x_{Ai} are small and zero-centered, the variance of the overall contribution C^A_{RD} of all other neurons in the reservoir will typically be small as well.

The second factor influencing the balance is the contribution from the output feedback. It depends on the activities of the output y and the output feedback weights w_{Ay}. According to the balance condition (7), the contribution from the output feedback should have a variance that is comparable to the one stemming from the reservoir neurons. One possibility to ensure this is to restrict the output feedback influence. The output feedback weight w_{Ay} is a random value taken from the interval $[-W_{OFB}, +W_{OFB}]$ for every reservoir neuron. Due to the importance of this interval throughout this paper and furthermore due to its zero-centered property, we reference this interval from now on as

$$W_{OFB} \Leftrightarrow [-W_{OFB}, +W_{OFB}]. \tag{8}$$

Depending on the target dynamics, which may be larger than 1, the feedback weights should compensate this influence ensuring a small external driving signal. An upper bound for the initialization interval to ensure an influence at least below one can thus be set by enforcing

$$W_{OFB} < \frac{1}{max(y^{target})}. \tag{9}$$

A particular choice of W_{OFB} should thus depend on the characteristics of the modeled dynamics. In many applications, including the modeling of MSOs, the output neurons have a linear activation function and the values of the target dynamics are usually not restricted to the interval $[-1, +1]$. In such cases, W_{OFB} should be significantly smaller than 1. Although the proposed inequality certainly determines only an upper bound on the range of the output feedback weights W_{OFB}, we will see below that sufficiently small W_{OFB} values are indeed crucial for increasing the likelihood of generating effective reservoir networks.

If W_{OFB} is not chosen small enough, however, the driving signal generated via the feedback connections is highly likely to overrule the internal dynamics. Thus, internal dynamic memory will be quickly overwritten, significantly reducing the memory capacity. As a rule-of thumb it can be hypothesized that the stronger the output feedback influence compared to the internal dynamics, the poorer the average ESN performance. Moreover, the negative effect of overly strong feedback connections can be expected to be especially visible at the extreme points of the target dynamics. On the other hand, W_{OFB} must not be too small, because the feedback signal would not be strong enough to excite the dynamic reservoir, resulting in progressively declining dynamics. This supposes that an optimization of the interval for initializing the output feedback weights may be valuable. Properly balanced, reservoir neurons have the highest probability of maintaining a large variety of reservoir dynamics for computing diverse network outputs.

Note that the presented considerations assume an ESN without input neurons. However, the above idea should also be generally valid for an ESN with both, output feedback and input neurons and even multiple of each. In these cases, the respective input and output feedback influences should be balanced with the internal reservoir dynamics and thus the balance condition (7) should be extended by the variance of the contribution from the input neurons, and would have the following form

$$\sigma^2(C^A_{RD}) \approx \sigma^2(C^A_{OFB}) \tag{10}$$

$$\sigma^2(C^A_{RD}) \approx \sigma^2(C^A_{IN}) \tag{11}$$

where $\sigma^2(C^A_{IN})$ denotes the variance of the contribution from the input neurons. Thus, in this case additional to the proper choice of W_{OFB} also the input weight range $[-W_{IN}, +W_{IN}]$ will need to be chosen suitably. Conditions (10) and (11) essentially suggests that neither the contribution from the input neurons nor the contribution from the output neurons should dominate the reservoir dynamics in order to make the reservoir dynamics flexibly available for computing network output.

To sum up, our considerations suggest that ESN performance crucially depends on a proper balance of the internal dynamics and the influence of external activities, mediated via the output feedback weights, the input weights, and the strengths of the input and target activities. Seeing that the external activities cannot be directly manipulated, the initialization of the output feedback and input weights are crucial. If they are set too high, they will dominate the dynamics unfolding in the reservoir and quickly overrule any other useful internal dynamics. The negative effect can be expected to be especially strong, if the majority of the reservoir neurons operates in the saturation regime, where the reservoir neurons act very restrictively on the variety of reservoir states available for computing the network output. In this case, the provided variety of states is even lower than the variety stemming from the feedback signal alone. A proper choice of the weight

initialization intervals is likely to reduce the number of unbalanced neurons, while increasing the number of balanced neurons, which are necessary to generate a high performing ESN. This puts a proper choice of the feedback weights and a proper choice of the input weights onto the same stage of importance as a correct choice of the spectral radius λ^*. In the following we experimentally verify the claims and theoretical considerations put forward in this section.

4. Experiments

The current section presents a number of experiments, which were conducted to scrutinize the impact of the output feedback weights on ESN performance focusing on the MSO benchmark. The experiments show that network performance varies highly significantly depending on the chosen output feedback weights. Additionally, the experiments reveal dependencies between the chosen output feedback weights and other network parameters, including the reservoir size and the connectivity of the dynamic reservoir.

4.1. Setup

The considered MSO time series data were generated by summing up several simple sine wave functions. Formally it is described by the following equation:

$$y(n) = \sum_{i=1}^{s} \sin(\alpha_i n), \tag{12}$$

where s denotes the number of sine waves, α_i the frequencies of the summed sine waves, and n specifies an integer index of the time step. In the following, we denote particular MSO dynamics as MSOx where x stands for the number of summed sine waves. Various publications have investigated the MSO problem with different numbers of sine waves. For example, in Xue et al. (2007) the approach was tested only on MSO2; up to eight sine waves were considered in Roeschies and Igel (2010); up to five sine waves were considered in Schmidhuber et al. (2007) and in Holzmann and Hauser (2010). In all of the mentioned studies the frequencies of the sine waves were taken from the same set: $\alpha_1 = 0.2$, $\alpha_2 = 0.311$, $\alpha_3 = 0.42$, $\alpha_4 = 0.51$, $\alpha_5 = 0.63$, $\alpha_6 = 0.74$, $\alpha_7 = 0.85$, and $\alpha_8 = 0.97$. The increasing values increase the period of the resulting signal and effectively prevent exact repetitions of $y(n)$ over the training and test sequences.

To maximize the generality of the conclusions put forward for MSOs in this work, we investigate the most commonly used dynamics in detail, which are MSO2, MSO3, MSO5, and MSO8. For MSO4, MSO6, and MSO7, we report results that were only optimized according to the basic schedule detailed below. The used lengths of the training and test sequences were 400 and 300 steps, respectively. The first 100 steps of the training sequence were used as the *washout phase*. They were considered neither for computing the optimal output weights based on (2) nor for computing the reported test errors. These sequence lengths were also chosen by Holzmann and Hauser (2010); Roeschies and Igel (2010); Schmidhuber et al. (2007). Xue et al. (2007) applied a training sequence with 700 steps.

The considered standard ESN used for the experiments is shown schematically in Fig. 2. Every network consisted of a dynamic reservoir and one linear output neuron without a self-recurrent connection. All reservoir neurons applied a non-linear TANH activation function. All neurons of the dynamic reservoir were connected to the output neuron. The output feedback connected the output neuron to all reservoir neurons. Since no input neurons were used, the output feedback was the only driving signal for the dynamic reservoir. The reservoir neurons were connected to each other choosing a particular sparsity. In fully connected reservoirs, all reservoir neurons were directly connected to each other with all connection weights being non-zero. More sparsely connected reservoirs were generated in two steps. First, a fully connected reservoir was generated. Next, a certain amount of randomly chosen weights of the reservoir connections were set to 0. The amount of the remaining reservoir connections corresponded to the chosen connectivity. In our experiments, we varied the connectivity while ensuring that the dynamic reservoir was still fully recurrently connected. A necessary condition for the reservoir existence is that $\lambda^* > 0$, that is, the largest eigenvalue is non-zero. If the connectivity was lower than this limit, the connections between the reservoir neurons are so sparse that the reservoir does not exist as a group of mutually connected neurons. Usually larger reservoirs (more neurons) tolerate a lower connectivity and still fulfill the requirement. In the generated ESNs all weights were initialized set uniformly randomly within a specific interval. We specify two different intervals for the initialization of the weights: one for the reservoir weights and one for the output feedback weights. The interval of the reservoir weights was $[-1, +1]$. After random initialization, their reservoir weights were scaled with the spectral radius, which was set to $\lambda^* = 0.8$. The initialization interval of the output feedback weights W_{OFB} is the most important parameter investigated below.

Additionally to the used structure of the ESNs, a suitable performance indicator and the number of runs were important as well. We used the normalized root mean square error (NRMSE), calculated as follows:

$$NRMSE = \sqrt{\frac{\sum_{t=0}^{T-1} (y(n) - o(n))^2}{T\sigma_y^2}}, \tag{13}$$

where T is the test sequence length, $o(n)$ is the network output at time step n, and σ_y^2 is the variance of the target dynamics $y(n)$. The test error was computed on the test sequence, where no teacher forcing was applied. The state of the dynamic reservoir was not reset after training, which made a washout after the training part unnecessary. Since the ESNs were generated randomly, the network performance was a stochastic value, which was represented by different statistics. Based on our investigations, the most representative statistic is the smallest NRMSE reached by an ESN among the other networks generated for the same parameter settings. Other possible statistics would be the cumulative statistics: mean NRMSE, median NRMSE and the error variance. We found that they are not expressive enough and even partially meaningless because for many parameter settings the resulting error distribution in the performance of the individual ESNs was far from being Gaussian. Two examples of such error distributions plotted as error histograms are shown in Fig. 3. As can be seen, the observed non-Gaussian error distributions consist of three "hills". While a group of "good" ESNs reproduces the target dynamics, "constant" ESNs yield the deviation from the mean as their performance error, and finally, a group of "outlier" networks become unstable and are thus not at all able to reproduce the target dynamics. Depending on the setup, the distribution within and across these three groups can be diverse. As a consequence, the average statistics are rather meaningless, since especially the mean will tend to report a value that lies in a region with very low distribution density. Also the error variance is rather meaningless since the obtained experiment data are not distributed around a highly populated center. Seeing that we are most interested in the group of "good" ESNs, the best NRMSE is the most suitable statistic for our analyses. The best NRMSE turned out to yield rather stable values for a large number of experimental runs. Nonetheless, in the

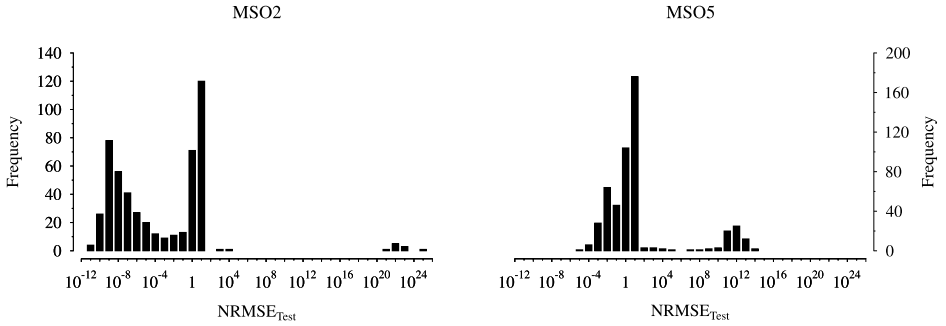

Fig. 3. Non-Gaussian error distributions of typical runs.

evaluation section below, we also report further error distribution results, similar to the ones presented in Fig. 3.

The randomness of the generated networks required a large number of experimental runs to compute the network performance. We generated 500 ESNs for each investigated setup and involved parameters. To verify that the derived statistics are sufficiently representative, we conducted a series of preliminary experiments with 500 runs and with 15,000 runs for the same parameter settings, where the respective error distributions did not reveal particularly strong differences.

In our experiments we tried to find out the unknown dependencies of the feedback weights on two network parameters: the reservoir size p_i and the reservoir connectivity c_i. For every considered parameter value combination p_i and c_i, we searched for the most suitable initialization interval W_{OFB} for the output feedback weights. For all reported optimization runs, we tested 24 possible intervals: $W_{OFB} \in \{10^{-15}, 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 5 \cdot 10^{-8}, 10^{-7}, 5 \cdot 10^{-7}, 10^{-6}, 5 \cdot 10^{-6}, 10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$. Thus, each best NRMSE reported is obtained by generating, optimizing, and testing $500 \cdot 24 = 12{,}000$ ESNs and reporting the lowest NRMSE produced by one of these networks during testing. This procedure can also be seen as a simple optimization procedure, which we formalize as follows:

$$\epsilon^* = \min_{j=1,\ldots,24} \left\{ \min_{r=1,\ldots,500} \{NRMSE_r(p_i, c_i, W_{j,OFB})\} \right\}, \quad (14)$$

$$W^*_{OFB} = \arg_{W_{j,OFB}} \min_{j=1,\ldots,24} \left\{ \min_{r=1,\ldots,500} \{NRMSE_r(p_i, c_i, W_{j,OFB})\} \right\}, \quad (15)$$

where ϵ^* denotes the best NRMSE as formalized, r is an iterator referring to the independent uniform random generation of a network, j iterates over the 24 intervals considered for W_{OFB} reported above, denoting a particular interval by $W_{j,OFB}$, and $\arg_{W_{j,OFB}}$ returns the $W_{j,OFB}$ that yielded the network with the lowest NRMSE.

4.2. Results

Here we present four sets of experiments, which highlight the importance of balancing the internal reservoir dynamics with the output feedback. The first set of experiments confirms that a maximally suitable output feedback range exists. The second and third set demonstrate how the output feedback weights interact with the reservoir size and the reservoir connectivity. The last set shows that an interval of the maximally suitable ranges of the output feedback weights can vary on a large scale depending on the chosen reservoir size.

4.2.1. Output feedback weights

We first show that there exists a range of output feedback weight ranges that yields the best performing networks with high probability. The used target time series is MSO3. The dynamic reservoir consisted of $p_i = 10$ neurons. The reservoir connectivity was set to $c_i = 40\%$. Fig. 4 shows the observed dependency between the network performance and the output feedback weight range W_{OFB}. As expected, best ESNs are generated for a particular range of W_{OFB} settings. Both too low as well as too high settings yield increasingly inaccurate performance. The results presented in Fig. 4 indicate continuously improving performance with decreasing output feedback weight ranges from $W_{OFB} = 10^{-2}$ to $W_{OFB} = 10^{-8}$. In the range of about $W_{OFB} = 10^{-8}$ to $W_{OFB} = 10^{-14}$ the best NRMSEs stay at approximately the same low level. Within this interval the output feedback and the reservoir dynamics appear sufficiently balanced.

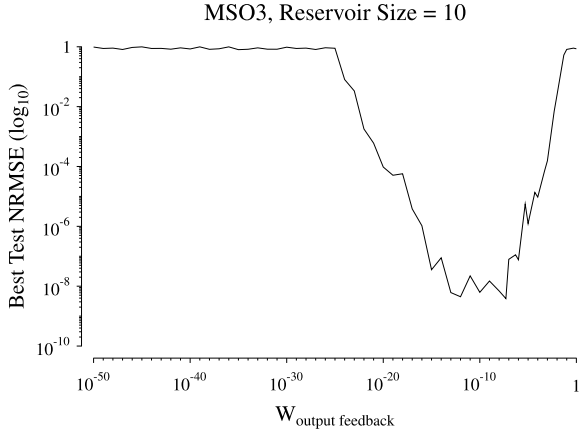

Fig. 4. Best NRMSE depending on the output feedback weight range W_{OFB}.

Table 1
Contributions C_{RD} and C_{OFB} for the increasing output feedback weights.

Interval W_{OFB}	$5 \cdot 10^{-8}$	0.005	0.5
NRMSE	$3.85 \cdot 10^{-9}$	$6.95 \cdot 10^{-3}$	0.901
Average $\sigma^2(C_{RD})$	$6.54 \cdot 10^{-13}$	$2.08 \cdot 10^{-3}$	7.19
Average $\sigma^2(C_{OFB})$	$4.82 \cdot 10^{-13}$	$6.16 \cdot 10^{-3}$	20.6
Max(C_{OFB}–C_{RD})	$6.31 \cdot 10^{-13}$	$1.11 \cdot 10^{-2}$	36.3

Table 1 shows average contributions C_{RD} and C_{OFB} obtained when averaging all reservoir neurons of the best ESNs for different initialization intervals W_{OFB}. The table presents data corresponding to three points on the curve in Fig. 4. These points are on the right slope of the curve. The best NRMSEs were obtained for $W_{OFB} = 5 \cdot 10^{-8}$. First deviations between ESN output and the target dynamics were observed with $W_{OFB} = 0.005$. For $W_{OFB} \geq 0.5$ the resulting ESNs failed to reproduce the target dynamics. The last row shows the maximum difference between the contributions C_{RD} and C_{OFB} among all reservoir neurons of the corresponding ESN. The table particularly shows that the variance in the reservoir dynamics increases with increasing values of W_{OFB}. The higher the feedback weights, the more external drive applies and thus the more activity

Table 2
Optimal reservoir sizes and corresponding output feedback weights.

Dynamics	Optimal size	Best W_{OFB}	Average $\sigma^2(C_{RD})$	Average $\sigma^2(C_{OFB})$
MSO2	5	10^{-10}	$6.48 \cdot 10^{-19}$	$2.02 \cdot 10^{-18}$
MSO3	7	10^{-8}	$1.96 \cdot 10^{-12}$	$1.78 \cdot 10^{-12}$
MSO5	11	10^{-7}	$2.28 \cdot 10^{-11}$	$2.47 \cdot 10^{-12}$
MSO8	68	$5 \cdot 10^{-5}$	$2.36 \cdot 10^{-6}$	$9.68 \cdot 10^{-7}$

is present in the network. With sufficiently low feedback weights, the variance in the feedback drops slightly below the variance induced by the reservoir connections, yielding the best performing networks. The low values of $\sigma^2(C_{RD})$ and $\sigma^2(C_{OFB})$ suggest that the network is operating in the linear regions of their sigmoids, which extends the memory capacity of the dynamic reservoir, as also suggested by Jaeger (2002a). This memory capacity extension is surely an additional factor that improves the ESN performance with high likelihood.

Below $W_{OFB} = 10^{-14}$, however, the driving effect of the output feedback on the reservoir decreases too much. Many dynamics in the network approach zero. Possibly also due to the double floating point precision restrictions, close-to-zero activities drop to zero and effectively die-out. Thus, the involved neurons do not contribute further to the unfolding of the reservoir dynamics, which causes the shrinkage of the dynamic reservoir capacity. ESNs that were generated with an output feedback weight range of less than $W_{OFB} = 10^{-26}$ yield the worst performance, because hardly any dynamics are induced into the network and those that are induced die-out quickly.

This set of experiments has shown that a proper choice of the output feedback weights is crucial for generating a well-performing ESN. Inappropriately chosen values were shown to prevent a proper reproduction of the target dynamics. According to the obtained results, the appropriately chosen feedback weights are rather small. Highest accuracy was obtained with weights ranging between $W_{OFB} = 10^{-14}$ and $W_{OFB} = 10^{-8}$. To the best of our knowledge, these values are significantly smaller than the usual feedback weight ranges considered in the literature. This may be the reason why it was partially believed that the standard ESNs were not able to model the MSO problem properly.

4.2.2. Reservoir size

Clearly, the reservoir size is of vital importance for the capabilities of an ESN. It influences the memory capacity and the variations in the reservoir dynamics to a large degree. In the following experiments, however, we show that the intuitive rule-of-thumb: "the larger the network the better the performance", does not necessarily hold in ESNs. Indeed, we show that there exists a maximally suitable reservoir size for the generation of the most accurate ESNs. We conducted experiments with a number of reservoir sizes starting with three neurons moving up to 250 neurons. Seeing that the interactions between the feedback weight ranges and the reservoir sizes are unknown, we determined the most suitable interval W_{OFB} according to the procedure specified in (15) for every reservoir size and report the corresponding best NRMSE based on (14). Other parameters and setups were chosen as specified above. The connectivity was set to 40%, except for the very small reservoirs with less than five neurons. To ensure that $\lambda^* > 0$, the connectivity of these reservoirs was slightly increased to 50% and 60%.

Fig. 5 shows the dependency of the network performance on the reservoir size. Each data point in the graphs essentially reports the best performance achieved in 12,000 independent runs. The figure shows that the smallest reservoir of three neurons was insufficient to model any of the taken dynamics. Larger reservoirs improved the performance where the start of the improvement depends on the complexity of the MSO problem. We call the point where

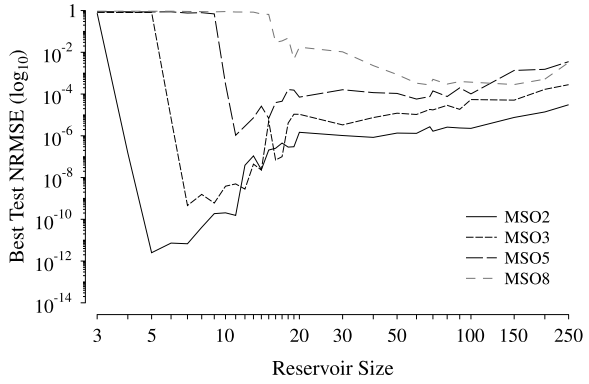

Fig. 5. Dependency of the best network performance on the chosen reservoir size.

performance starts improving the *corner reservoir size*. The error curve of MSO8 starts decreasing only at a reservoir size of 15, whereas the error curve of MSO2 starts decreasing already when the reservoir has four neurons. The rate of decrease is different in both cases. For MSO2 the best network with four reservoir neurons has a much smaller error than the best network with 15 reservoir neurons generated for MSO8. However, the most interesting phenomenon is the existence of a maximally suitable reservoir size, for which the best network yields the smallest error. This reservoir size can be seen as an optimal one at least for the parameters of our experiments.[1] For the target dynamics MSO2, MSO3, and MSO5, the maximally suitable size is only slightly larger than the corner reservoir size. The maximally suitable reservoir sizes and the corresponding intervals for initializing the output feedback weights are presented in Table 2. Additionally, the table shows average variances of the activity contributions from the reservoir dynamics and from the output feedback in the corresponding ESNs in the best runs.

For MSO8, the optimal reservoir size is not so clear-cut. All considered reservoir sizes between 60 and 150 neurons provided similar network performances. Among them the smallest error was obtained for a reservoir size of 68. Only for reservoir sizes above 150, network performance started deteriorating again. Note that particularly for the smaller MSO problems a sharp decrease in the test error of the best ESN can be observed. As shown in Fig. 5, with a few additional reservoir neurons the error decreases by several orders of magnitude! When the reservoir size is increased further above the optimal size, the performance of the best generated network starts slowly deteriorating again. This performance decrease depends on the complexity of the target dynamics: it is highest for MSO2 and lowest for MSO8.

To get a closer look at the performance of the ESNs generated for different reservoir sizes, we have drawn four error histograms for each considered target dynamic: one for the optimal reservoir size, one for a reservoir size smaller than the optimal one, one for a reservoirs size larger than the optimal one, and one for the largest considered reservoir size of 250 neurons. The considered reservoir sizes are summarized in Table 3. The error histograms are shown in Fig. 6.

[1] In the text below we will use the term "optimal" without repeating that optimal is meant with respect to the conditions chosen in our experiments.

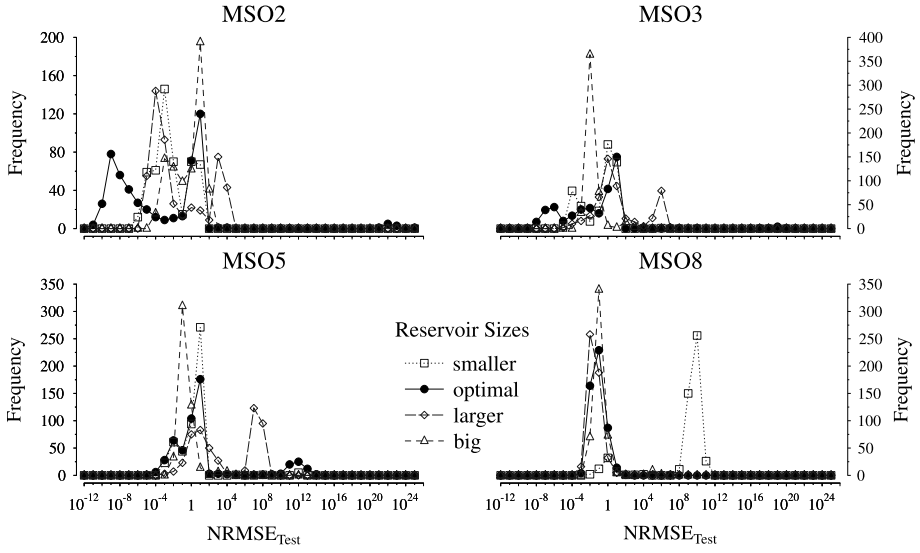

Fig. 6. Error distributions for MSO2, MSO3, MSO5, and MSO8.

Table 3
Reservoir sizes for drawing the error distributions.

Dynamics	Smaller size	Optimal size	Larger size	Largest size
MSO2	4	5	20	250
MSO3	6	7	30	250
MSO5	10	11	30	250
MSO8	40	68	100	250

The smaller and larger sizes were chosen relative to the optimal reservoir size. The plots show a significant difference between the error distributions obtained for the optimal and very large reservoir sizes. This difference is strongest for the simplest dynamics (MSO2). For MSO8, both distributions yield only minor differences. This confirms the observations taken from Fig. 5, where the difference between the network errors for the optimal and largest considered reservoir sizes is very large for MSO2, but rather small for MSO8.

Fig. 6 also shows that the distribution of the optimal reservoir size has the largest spread to the left in comparison to the other distributions in each figure. This confirms that an optimal reservoir size yields the highest probability of generating an ESN with a very high performance. The shapes of the error distributions are very important as they characterize the process of the random generation of the ESNs. For MSO2, MSO3, and MSO5 every distribution exhibits typically two or three "hills" (centers of mass) for all reservoir sizes, except for the largest one. The left "hill" corresponds to networks that approximate the dynamics of the MSO rather well. The center "hill" is produced by networks that yield flat approximations. The right "hill" is due to reproductions that become unstable. The size of the hill reflects the likelihood of reproducing the respective types of dynamics.

Fig. 7 shows examples of the output dynamics of ESNs that have different errors in MSO3, reflecting the three discussed "hills". First, an accurate reproduction of the dynamics can be seen. Second, a flat reproduction of the average value of the dynamics is shown. Third, an approximation that becomes unstable is displayed in the bottom panel.

As was shown in Fig. 5, every target dynamic investigated has a certain reservoir size that is sufficient for modeling the dynamics. According to our experiments, any smaller reservoir is highly unlikely to model the target dynamics accurately. The sufficient reservoir sizes are small for simpler dynamics like MSO2, MSO3, and MSO5. Larger reservoirs seem indispensable for more complex dynamics, such as MSO8.

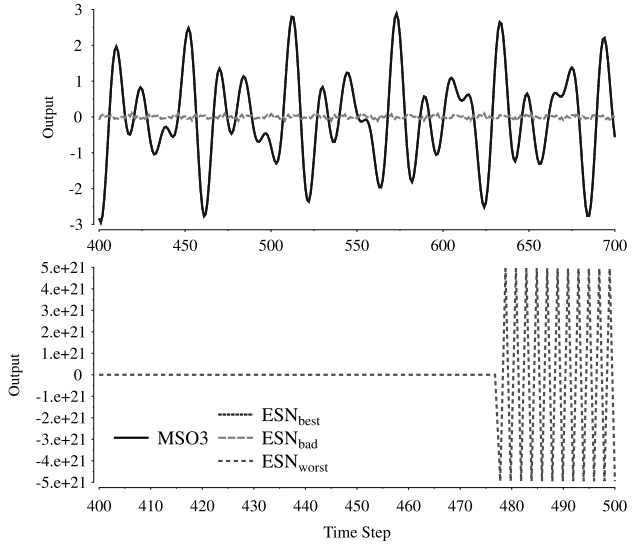

Fig. 7. Characteristic network output for the MSO3 problem. The upper panel displays the output of the best ESN and an ESN with an NRMSE of 1.0. The bottom panel shows an ESN with a very large error $4 \cdot 10^{21}$ (note the larger axis scaling).

The reservoir size does not only provide a variety of the dynamics but also determines the most suitable interval for initialization of the output feedback weights. As can be seen in Table 2, there is a dependency between the reservoir size and the most suitable output feedback weights: the larger the reservoir size, the larger the most suitable interval, indicating that larger feedback weights are needed to reach a balance of the reservoir neurons. It may be assumed that the larger the internal reservoir, the harder it is to induce output feedback dynamics into it; thus, the larger the necessary feedback weight ranges. However, larger feedback weight ranges may prevent the independent unfolding of internal dynamics — which is most likely the reason for the fact that larger networks do not necessarily reach equally good performance levels. Further experimental and theoretical work appears necessary to scrutinize the respective influences on the unfolding dynamics even further.

The current experiments nonetheless have shown that there exists an optimal reservoir size for a considered target dynamic. It was shown that the traditional perception – the larger the reservoir the better – does not hold. Moreover, the experiments

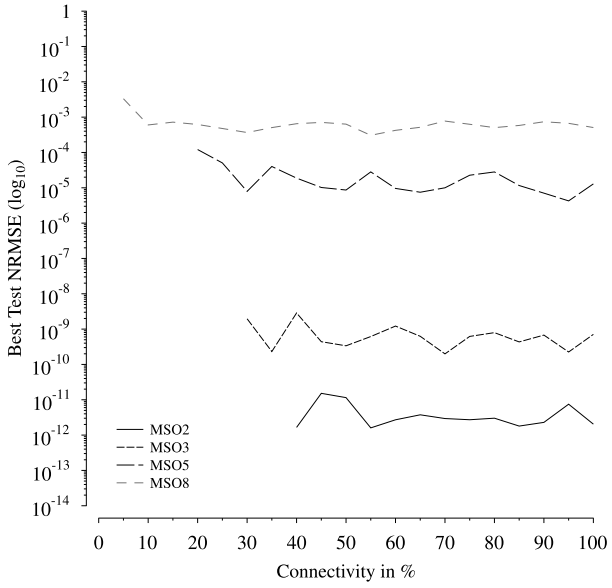

Fig. 8. ESN performance vs. reservoir connectivity.

again emphasize the crucial role of the output feedback weights: which output feedback weight ranges are most suitable does not not only depend on the target dynamics but also on the size of the reservoir.

4.2.3. Connectivity

So far we have shown that the initialization interval of the feedback weights as well as the reservoir size can strongly influence ESN performance. Another factor considered crucial in the literature is the connectivity of the dynamic reservoir. We thus conducted experiments with different reservoir connectivities. We report the best results obtained when permuting the W_{OFB} interval according to (15), as done above. Once again, the experiments were conducted for all four target dynamics MSO2, MSO3, MSO5, and MSO8. Networks generated for the same target dynamics had the same reservoir size, where the reservoir size was set to the experimentally-determined optimal ones (see Table 2). The largest connectivity considered was 100%, while the smallest connectivity was 40% for MSO2, 30% for MSO3, 20% for MSO5, and 5% for MSO8. These values were chosen on the basis of the reservoir size to satisfy that $\lambda^* > 0$ in all considered randomly generated ESNs. Except for the change in connectivity, the other settings and parameter values were kept as specified above.

Fig. 8 shows the dependency of the test error on the reservoir connectivity. Surprisingly, the best ESN performance stays approximately constant despite the changing connectivity. There is no particular connectivity where the generated networks reproduced the target dynamics especially precisely nor particularly poorly. All reservoirs had a sufficient memory capacity and a sufficient variety of echo states to reproduce the target dynamics with a similar precision. For the more complex dynamics MSO5 and MSO8, slightly worse reproduction accuracy was reached with very sparse reservoirs.

The obtained results suggest that the likelihood of generating a very high-performing ESN given a suitable network size is independent of the connectivity, as long as internal dynamics take place, that is, as long as $\lambda^* > 0$. Experiments with larger reservoir sizes and varying connectivity may demonstrate improvements of the ESN performance. Further experiments may be conducted in this respect. For the purpose of this paper, however, the main message is that the connectivity itself has nearly no influence on the best reachable performance in the MSO problems, given a favorable reservoir size and a suitable output feedback weight range W_{OFB} was chosen.

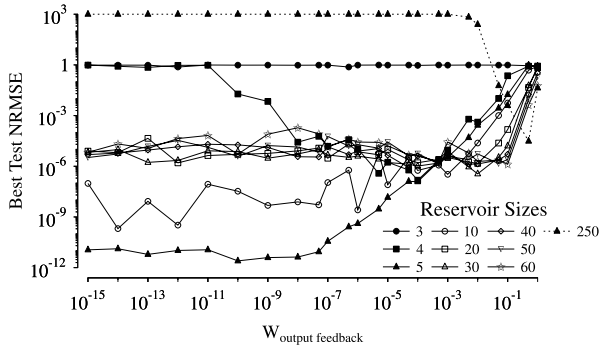

Fig. 9. Dependency of ESN performance on output feedback weight ranges for different reservoir sizes (MSO2).

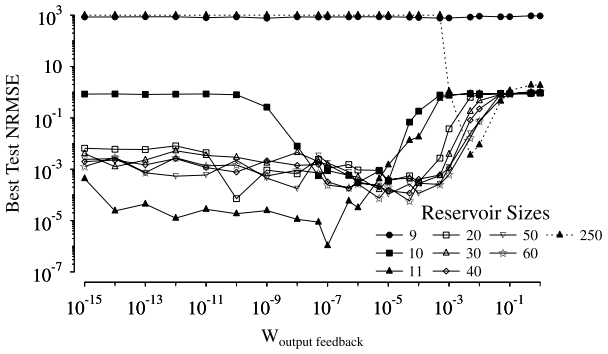

Fig. 10. Dependency of ESN performance on output feedback weight ranges for different reservoir sizes (MSO5).

4.2.4. Interaction of reservoir size and output feedback weight range

As a final performance evaluation, we were interested in the mutual dependence of the output feedback weight range W_{OFB} and the reservoir size N. The experiments were conducted on MSO2 and MSO5. The ESNs were generated and trained for every combination of the mentioned intervals of W_{OFB} and the reservoir sizes. Again, we report the best performance achieved with one out of 500 networks given a particular W_{OFB} and reservoir size N.

The results for MSO2 are shown in Fig. 9 and for MSO5 in Fig. 10. In both figures, almost all curves indicate smaller errors for at least a short range of W_{OFB}'s. Only networks with very small reservoirs were not able to reproduce the target dynamics — for MSO2, such reservoirs consisted of three or fewer neurons, for MSO5, nine or fewer neurons yielded insufficient reservoir capacities. According to the results obtained with larger reservoirs, it is possible to distinguish three different degrees of sensitivity. Networks with low sensitivity in the output feedback weights had either an optimal reservoir size or reservoirs that were by at most several dozens of neurons larger than the optimal one. These networks had a very small error on a wide range of output feedback weight ranges. However, the optimal reservoir size, nonetheless, yields errors that drop even further. Slightly smaller reservoirs quickly yield worse performances and were more sensitive to the choice of the output feedback weight range. The highest sensitivity on the choice of W_{OFB} was observed for the networks with very large reservoirs ($N = 250$). There is only a very small interval where the displayed curves go below 10^{-1} — an error level that indicates that the generated ESN was able to reproduce the target dynamics with an acceptable accuracy.

The obtained results clearly show that an inappropriate parameter initialization can prevent the generation of a powerful ESN. In most papers on ESNs, the generated reservoirs were large with sizes exceeding 100 neurons. The values of the output feedback weights were close to 1 for different target dynamics.

Table 4
Performance comparison of the best achieved performance when permutating the output feedback weight ranges, optimizing the reservoir size, and reporting the best network performance out of 500 randomly generated ones (column "our settings") with results reported in the literature; "Evolutionary" is from Roeschies and Igel (2010), "Evolino" by Schmidhuber et al. (2007), "DESN" from Xue et al. (2007), and "IIR ESN" by Holzmann and Hauser (2010).

Dynamics	Statistics	our settings	evolutionary	Evolino	DESN	IIR ESN
MSO2	Best NRMSE	$2.51 \cdot 10^{-12}$	$3.92 \cdot 10^{-8}$	$4.15 \cdot 10^{-3}$	–	$\approx 3 \cdot 10^{-9}$
	Best MSE	–	–	–	$3 \cdot 10^{-4}$	–
MSO3	Best NRMSE	$4.57 \cdot 10^{-10}$	–	$8.04 \cdot 10^{-3}$	–	$\approx 3 \cdot 10^{-7}$
MSO4	Best NRMSE	$5.72 \cdot 10^{-8}$	–	0.110	–	$\approx 10^{-5}$
MSO5	Best NRMSE	$1.06 \cdot 10^{-6}$	$2.54 \cdot 10^{-2}$	0.166	–	$\approx 8 \cdot 10^{-5}$
MSO6	Best NRMSE	$8.43 \cdot 10^{-5}$	–	–	–	–
MSO7	Best NRMSE	$1.01 \cdot 10^{-4}$	–	–	–	–
MSO8	Best NRMSE	$2.73 \cdot 10^{-4}$	$4.96 \cdot 10^{-3}$	–	–	–

For example, in Jaeger (2001) the output feedback weights were uniformly generated from the different intervals, where the smallest was $[-0.1, +0.1]$ and the largest was $[-4, +4]$. In the successful experiments with the non-standard IIR ESNs from Holzmann and Hauser (2010), the feedback weights were initialized from $[-0.1, +0.1]$ as well. Probably similar intervals were applied in other unsuccessful attempts to model MSO problems with ESNs. This is most likely one of the reasons why standard ESNs are considered unsuitable for modeling MSO dynamics.

4.3. Heuristic for generating a balanced ESN

The typical performance structure obtained given particular output feedback weight ranges W_{OFB} (cf. Figs. 9 and 10) suggests a rather straightforward heuristic to generate the most suitable ESN: Start with a small network and work your way up to larger networks until performance stops improving. For each considered network size, test the variations in the output feedback weight ranges specified in (14). Once performance stops improving, the network size will be close to optimal. If a larger increment in the tested network sizes was chosen, a more detailed local search may be necessary to determine the actual optimal network size. Starting with small networks has the advantage that the computational effort necessary to generate and test the networks is comparatively small. Seeing the rather steep drop from insufficient reservoir capacities to optimal reservoir sizes, this drop should be detectable in a reasonable amount of time. Taking this approach, we determined the best achievable performance for all MSOs tested. In the next section, we compare our results with previous findings from the literature.

5. Performance comparisons

Based on the above considerations, we determined approximately optimal reservoir sizes N and corresponding best accuracy values. The best performance values according to (14) are reported in Table 4 for all seven considered MSO problems (column "our settings"). We compare these results with other approaches that used the MSO dynamics as a benchmark. Two of them from Xue et al. (2007) and Holzmann and Hauser (2010) are extensions of standard ESNs. Another approach is the "Evolino" memory cell neural architecture presented in Schmidhuber et al. (2007). The approach from Roeschies and Igel (2010) used the same standard ESNs like those randomly generated in our work. However, in Roeschies and Igel (2010) the initial randomly structured ESNs were further optimized by an evolutionary algorithm. In Table 4 the modeling accuracy reached with the IIR filter ESNs is specified as an interval because in Holzmann and Hauser (2010) it was presented only graphically with plots. Note that Čerňanský and Tiňo (2008) reported MSE values as low as $1.04 \cdot 10^{-28}$ on the MSO2 with a simple tapped delay line and $1.3 \cdot 10^{-26}$ with an ESN with 400 hidden units. It remains unclear if these values beat our results, because the normalization and the square root limits comparability. Moreover, Čerňanský and Tiňo (2008) used a recursive connection in the output node, a reservoir of 400 neurons, and different lengths for train and test sequence. Nonetheless, for small MSOs it should be kept in mind that even simpler neural networks may yield highly satisfactory performance. For the case of evolutionary-optimized networks, usually the best generated network performance was reported averaged over several evolutionary runs. The 12,000 networks that we generated for a particular network size may be considered as only one optimization trial. As stated above, however, even more runs did not yield particularly different performance results. Thus, the performance results should be sufficiently comparable.

As Table 4 shows, the balancing of the reservoir neurons in the randomly generated ESNs provided the highest reproduction accuracy for all considered dynamics. Compared to the other approaches, it improves the accuracies by several orders of magnitude on the target dynamics. It is important to note that all compared approaches except the one from Xue et al. (2007) were applied to the same training and test sequences — 400 and 300 time steps, respectively. In Xue et al. (2007), the DESNs were trained under slightly more favorable conditions where the training sequence consisted of 700 time steps.

Apart from the very high accuracy, the balancing of the reservoir neurons also allowed the reduction of the memory requirement by optimizing the reservoir size. The most suitable reservoir sizes were determined to be five, seven, nine, eleven, 14, 18, and 68 reservoir neurons for the target dynamics MSO2, MSO3, MSO4, MSO5, MSO6, MSO7, and MSO8, respectively. The DESNs from Xue et al. (2007) had 400 neurons. In Holzmann and Hauser (2010), the IIR neuron ESNs had reservoirs with 100 neurons. In Schmidhuber et al. (2007) the memory cell networks had different amounts of neurons for different dynamics. The smallest neural network was used for MSO2 and had 60 neurons. The largest networks were used for MSO4 and for MSO5 and had 120 neurons. Only in Roeschies and Igel (2010), MSO2 was reported to be modeled very precisely with only three reservoir neurons. Unfortunately the authors did not give other parameters of that ESN. In our experiments, none of the generated networks with a three-neuron reservoir was able to reproduce the MSO2 dynamics.

Nonetheless, Table 4 shows huge performance gains through the balancing of the reservoir neurons and the choice of a favorable reservoir size in the MSO dynamics. It is likely that similar improvements of the accuracy can also be reached when modeling other dynamics that are combinations of multiple signals at different time scales.

6. Summary and conclusions

In the current article we have shown that it is vital to maintain a good balance between the driving input signal, which is induced into the reservoir network, and the recurrently unfolding internal dynamics for generating effectively-performing echo state

networks (ESNs) for the multiple superimposed oscillator (MSO) benchmark problems. Our experiments have shown that ESNs with output feedback connections, which carry the driving signal in this case, can reproduce the MSO dynamics much better than ever reported before. When choosing a favorable reservoir size and feedback error weight range, randomly generated ESNs have a high chance of generating highly accurate MSO dynamics. For several of the MSO benchmarks, we achieved NRMSEs that are several orders of magnitude smaller than the best ones reported in the literature. Two observations and consequent systematic variations have enabled us to achieve these results: The signals that drive the network dynamics (input and output feedback) need to be well-balanced with the internally unfolding reservoir dynamics; and the likelihood of generating suitable, internally unfolding, dynamics strongly depends on the size of the ESN neural reservoir.

Also from a theoretical perspective, we have proposed that the driving signals from the "outside" must not influence the internal dynamics too strongly. Thus, input flowing into the dynamic reservoir should not be too strong. In the case of the MSO problem, weight ranges in the order of 10^{-10}, for MSO2, to $5 \cdot 10^{-5}$, for MSO8, yielded the best results. Comparisons of the variances stemming from the activities in the internally propagated dynamics (5) and the output feedback dynamics (6) show that comparable variances yield the best performance. Thus, we proposed that the variances should be approximately balanced (7), (10) and (11). How such a balance can be achieved was sketched-out in Section 4.3.

While the common opinion is that larger recurrent neural networks are generally more likely to yield better performance, our results have shown the opposite. Larger reservoirs can actually affect the resulting ESN performance negatively! A similar observation was made elsewhere, stating that: "With only four neurons these nets will have markedly different dynamic properties across their random instantiations (which is not the case for larger ESNs where internetwork differences become insignificant with growing network size)" (Jaeger et al., 2007, p. 347). The authors of this paper conducted their experiments with the ESN with leaky-integrator neurons. Seeing the similarity between their observations and our experimental data, the characteristics of the randomly generated ESNs are to a certain degree independent of the reservoir neuron type — leaky-integrator or simple neuron. It appears that in larger networks dominant dynamics overrule other useful dynamics in an unfavorable way, somewhat averaging them out. It remains to be determined if sparsity may prevent this negative effect.

Our results in general indicate that ESNs can actually perform significantly better than believed previously. In many cases, reservoir sizes just above the overly small networks yielded the best performance. Moreover, small networks rely on small feedback weight ranges, larger ones need to have just the right feedback weight range.

We have also shown that the reservoir connectivity did not affect the resulting ESN performance in a systematic manner, given small networks and appropriate feedback weight ranges. However, sparsity may still be useful in larger networks or, of course, in other time series problems. Another ESN parameter, the spectral radius, was left out of the scope of the current work. An investigation of its impact on the performance of balanced ESNs and its relation to the other ESN parameters is pending. Also the impact of noise on the behavior of balanced ESNs, especially those with very small feedback weights, should be investigated further. Moreover, the conclusions derived from the MSO-based evaluations should be verified on other time series problems. Also dynamics that depend on additional input neurons should be considered. We expect that standard ESNs can reach similar performance gains when the strength of the driving signal is appropriately scaled.

In conclusion, while this paper has shown that ESNs can actually perform much better in the MSO benchmark problems than previously thought, we believe it also sets the stage and opens up potential for even further network optimizations. We have shown that a good reservoir size with a well-chosen feedback weight range can yield very good performance in otherwise randomly-initialized ESNs. Future research could work on preparing larger ESN for the reproduction of different but potentially similar dynamics. Sub-networks may be found that are maximally suitable to reproduce a particular dynamic. Overlaps in the identified sub-networks may then yield a similarity measure on the dynamics level. Thus, certainly also more evolutionary-based optimization techniques, similar to the ones utilized in Roeschies and Igel (2010) or in Schmidhuber et al. (2007), may very well be employed, starting however with much more favorable output feedback weight ranges and (sub-)network sizes. We hope that our analyses and the theoretical considerations put forward will enable the development of even more capable dynamic reservoirs.

Acknowledgements

Funding from the Emmy Noether program (German Research Foundation, DFG, BU1335/3-1) is acknowledged. The authors thank the Cognitive Modeling team for their comments and useful discussions.

References

Emmerich, C., Reinhart, R., & Steil, J. J. (2010). Recurrence enhances the spatial encoding of static inputs in reservoir networks. In K. Diamantaras, W. Duch, & L. Iliadis (Eds.), *Lecture notes in computer science*: *Vol. 6353. Artificial neural networks, ICANN 2010* (pp. 148–153). Berlin, Heidelberg: Springer Verlag.

Holzmann, G., & Hauser, H. (2010). Echo state networks with filter neurons and a delay & sum readout. *Neural Networks*, *23*, 244–256.

Jaeger, H. (2001) The "echo state" approach to analysing and training recurrent neural networks. *Technical report* GMD report 148, GMD — German National Research Institute for Computer Science.

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

Jaeger, H. (2002b) Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the echo state network approach. Technical report GMD report 159, Fraunhofer Institute for Autonomous Intelligent Systems (AIS).

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

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

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

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

Roeschies, B., & Igel, C. (2010). Structure optimization of reservoir networks. *Logic Journal of IGPL*, *18*, 635–669.

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

Čerňanský, M., & Tiňo, P. (2008). Predictive modeling with echo state networks. In *Lecture notes in computer science*, *Artificial neural networks — ICANN 2008* (pp. 778–787).

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

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