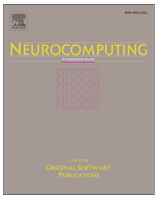
Neurocomputing 548 (2023) 126395

Contents lists available at ScienceDirect

Neurocomputing

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



Investigation of proper orthogonal decomposition for echo state networks

Jean Panaioti Jordanou [a],*, Eric Aislan Antonelo [a], Eduardo Camponogara [a], Eduardo Gildin [b]

[a] Department of Automation and Systems, Federal University of Santa Catarina, Florianopolis 88040-900, Santa Catarina, Brazil
[b] Harold Vance Department of Petroleum Engineering, Texas A&M University, College Station 77843-3116, TX, United States

ARTICLE INFO

Article history:
Received 2 December 2022
Revised 18 May 2023
Accepted 26 May 2023
Available online 1 June 2023
Communicated by Zidong Wang

Keywords:
Model order reduction
Reservoir computing
Echo state networks

ABSTRACT

Echo State Networks (ESN) are a type of Recurrent Neural Network that yields promising results in representing time series and nonlinear dynamic systems. Although they are equipped with a very efficient training procedure, Reservoir Computing strategies, such as the ESN, require high-order networks, i.e., many neurons, resulting in a large number of states that are magnitudes higher than the number of model inputs and outputs. A large number of states not only makes the time-step computation more costly but also may pose robustness issues, especially when applying ESNs to problems such as Model Predictive Control (MPC) and other optimal control problems. One way to circumvent this complexity issue is through Model Order Reduction strategies such as the Proper Orthogonal Decomposition (POD) and its variants (POD-DEIM), whereby we find an equivalent lower order representation to an already trained high dimension ESN. To this end, this work aims to investigate and analyze the performance of POD methods in Echo State Networks, evaluating their effectiveness through the Memory Capacity (MC) of the POD-reduced network compared to the original (full-order) ESN. We also perform experiments on two numerical case studies: a NARMA10 difference equation and an oil platform containing two wells and one riser. The results show that there is little loss of performance comparing the original ESN to a POD-reduced counterpart and that the performance of a POD-reduced ESN tends to be superior to a normal ESN of the same size. Also, the POD-reduced network achieves speedups of around 80% compared to the original ESN.

1. Introduction

Recurrent Neural Networks (RNN) are very relevant in applications related to modeling real-world phenomena when time-dependent data are available [1,2], and are considered universal approximators of dynamic systems. As RNNs are nonlinear, their training suffers from issues such as local minima, slow training, and the so-called "fading gradient" problem [3], which is a numerical problem inherent in Backpropagation Through Time (BPTT) [4], the algorithm used to calculate an RNN gradient. While some solutions focus on solving the fading gradient problem by changing the RNN structure, such as the Long Short-Term Memory (LSTM) network [5], or the gated recurrent unit [3], another flavor of RNN is worthy of attention: Reservoir Computing (RC). RC simplifies the learning by dividing the RNN into two parts: a high-dimensional recurrent nonlinear layer (the reservoir) with fixed, randomly generated weights and an adaptive readout output layer, which computes an instantaneous linear combination of the dynamic reservoir states [6]. The output-layer weights are trained through linear least squares, overcoming the problems related to nonlinear training and BPTT. *Reservoir Computing* became a unifying term for the frameworks of Liquid State Machines [7] and Echo State Networks (ESN) [6], both of which are methods for RNN training independently developed.

ESNs follow the general reasoning of Reservoir Computing: they adopt an architecture with a dynamic reservoir with fixed weights that projects the input to a high-dimensional space and a trainable static readout output layer. The dynamic reservoir needs to have many neurons [6] and the so-called Echo State property, which refers to the stability properties of the network. There are many successful applications of ESNs, such as: learning complex goal-directed robot behaviors [8], fuel cell lifetime prediction [9], wind speed prediction [10], medium voltage insulators classification [11], forecasting power system load using an ensemble deep ESN [12], power systems prediction with enhanced ESN that employ logistic mapping and bias dropout for reservoir weights generation [13], and prediction of the daily maximum temperature in the

* Corresponding author.

https://doi.org/10.1016/j.neucom.2023.126395
0925-2312/© 2023 Elsevier B.V. All rights reserved.

Melbourne airport with multi-reservoir ESN and an encoding and decoding scheme [14]. The large number of dynamic states in the reservoir is an essential characteristic, as the output, being a linear combination of them, can represent a more extensive repertoire of dynamics. However, using ESNs as dynamic models for problems such as optimization and MPC (Model Predictive Control) [15] may be an issue since the higher the number of states in the ESN is, the larger the optimization problem. Because the number of states in the ESN heavily dominates the number of inputs and outputs in such applications, a large reservoir size renders the optimization problem inherently larger and harder to solve.

As ESNs are high-dimensional, model order reduction methods can find equivalent ESN models with a considerably smaller number of states but which still keep the properties and performance of the original high-dimensional ESN. To that end, we count on Proper Orthogonal Decomposition (POD) [16], which applies Singular Value Decomposition (SVD) to find an optimal linear transformation that represents the state space of a large dynamical system in a more compact form. POD is already widely used to reduce the number of states of large dynamical models, especially phenomenological models such as a gas reservoir simulator [17] with tens of thousands of variables. However, POD has one disadvantage concerning nonlinear systems: although the method can reduce the number of states, it does not reduce the computation number of nonlinear functions. There are developments of interpolation methods, such as the Discrete Empirical Interpolation Method (DEIM) [16], to mitigate the issue by pivoting and approximating the nonlinear portion of the given model computation. Both POD and DEIM can find lower-dimensional networks that are equivalent to the original ESN and, thus, have the potential to alleviate the computational burden of simulations that depend on the size of the trained ESN.

The main objective of this work is to experiment with the use of POD and DEIM to obtain a reduced-order equivalent for an already-trained ESN. For such end, we apply the reduction given by POD in three different contexts: a Memory Capacity (MC) [18] evaluation experiment; a NARMA10 difference equation [19]; and a simulated oil platform containing two gas-lifted oil wells and one riser [20]. Additionally, we have shown results using DEIM-based reduction for the ESN in the first and last experiments mentioned above. We compare the performance of the reduced ESN to the original (non-reduced) ESN in the three experiments and another ESN with the same size as the reduced ESN in the MC and NARMA experiments. In this context, our main contributions are twofold: (1) we have developed efficient computational frameworks for implementing large echo-state network models in a variety of applications, which is achieved via model-order reduction (MOR) techniques; and (2) we have assessed the trade-offs between low-complexity reservoir models, resulting from the application of model order reduction (MOR), and the large baseline model in terms of numerical accuracy. The low-complexity models, despite their relatively small state-space dimensions, demonstrate comparable representation power to the large baseline model. As such, our work contributes to this nascent field of applications of MOR strategies to reservoir computing, which can potentially improve computational performance in modeling, control, and optimization. Specifically, the findings of our work are the following:

- The memory capacity of an ESN reduced by POD is generally higher than that of a non-reduced ESN of equivalent size. This difference in memory capacity is more significant as the desired ESN gets smaller in size.
- Given two echo state networks with the same number of states, the ESN obtained from POD reduction is likelier to perform better in a given task. This property is more evident and relevant when the desired reservoir is small.
- By employing a MOR method on ESNs, this work shows that small ESNs are robust and performant, improving their suitability for real-time or embedded applications with memory limitations.
- DEIM reduction alone for ESNs does not achieve satisfactory results compared to pure POD reductions.

In broader terms, the main implication of these findings is that a smaller version of an ESN, obtained by model order reduction, can achieve nearly equivalent behavior to the original (and larger) ESN, thus making dynamic reservoirs more compact. The new model can serve as a proxy model in optimization and predictive control, as an observer, and in other related tasks, addressing the issue of computational cost in a reservoir consisting of a large number of internal states (reservoir size), which can be orders of magnitude larger than the number of inputs and outputs.

This paper is organized as follows: Section 2 contains related works, Section 3 presents the Echo State Networks, Section 4 describes POD and DEIM, Section 5 reports on the case studies and experimental testing for the reduced ESN, and Section 7 concludes the work.

2. Related Work

In the following, we will discuss works in the literature that address the issue of reducing the model size in reservoir computing. One of them is [19], where they propose reducing the number of states by considering the output as a linear combination of the states at different instants in time, comparing to an original ESN through the Information Processing Capacity (IPC) metric, and also applying the proposal to a NARMA system and the generalized Hénon-map. The solution raises the effective number of states as a multiple of the delay or "drift-state" number utilized.

The architecture is very hardware-friendly, easing the computation compared to a standard ESN. Another example is the work [21], where they propose to employ the controllability matrix of the ESN as a means to find a so-called minimal ESN, which would be the ESN with the smallest reservoir that could reproduce the task at hand. They train the ESN for a particular task, obtain the controllability matrix at given points, and define its rank as a new candidate reservoir size. An extensive search procedure is then performed to find the optimal ESN at that size; however, there is no direct connection between the larger and the smaller ESN. In summary, the method in [21] proposes a useful way of finding a minimal reservoir for a task. In comparison, our work follows a different direction: reducing the size of the network through POD. Another work [22] proposes a different approach to reducing reservoir size, which calculates the correlation between each neuron and eliminates the reservoir neurons with the highest correlation.

The necessary large number of reservoir states in an ESN implies a complex computational model, therefore works such as [23] employ methods of so-called "network size reduction," which perform multi-objective optimization on the output weights and minimize not only the least-square error but also the number of non-zero elements in the output weights. Enforcing sparseness is ideal for simplifying computations with the ESN. Another work that follows this line of reasoning is [24], where they enforce a minimum complexity ESN by forcing the ESN reservoir to follow a deterministic form (i.e., a circular reservoir).

In [25], they propose to add the reservoir dimensionality reduction into the architecture via Principal Component Analysis (PCA) and calculate the output layer based on the PCA output instead of the reservoir states. They affirm that this enhances the dynamic properties of the resulting ESN concerning the system identified and improves the network generalization capabilities. Also, applying dimensionality reduction in the states renders the ESN a tool for dynamic system analysis. In this sense, our POD-ESN method is similar to PCA regarding obtaining the new state space but goes beyond [25] by embedding the reduction achieved in the reservoir's state update equation. In other words, the reservoir recurrent simulation is executed in the reduced state space with POD-ESN, which does not happen in [25].

Another approach of reduction in reservoir computing, not involving POD, is proposed in [26]. Their idea involves procedurally removing neurons according to the output weight value, which they curiously discovered that the network performance improves (given the Lorentz system as an application) by removing the neurons associated with large output weights. They thoroughly analyze the effect of removing different types of nodes in the ESN.

3. Echo State Networks (ESN)

An ESN is a type of recurrent neural network with useful characteristics for system identification [6], as it represents nonlinear dynamics well and the training consists in solving a linear least-squares problem of relatively low computational cost when compared to nonlinear optimization.

3.1. Model

Proposed in [27,28], the ESN is governed by the following discrete-time dynamic equations:

$$\mathbf{x}[k+1] = (1-\gamma)\mathbf{x}[k] + \gamma\mathbf{f}\left(\mathbf{W}_{\mathbf{r}}^{\mathbf{r}}\mathbf{x}[k] + \mathbf{W}_{\mathbf{i}}^{\mathbf{r}}\mathbf{u}[k] + \mathbf{W}_{\mathbf{b}}^{\mathbf{r}} + \mathbf{W}_{\mathbf{o}}^{\mathbf{r}}\mathbf{y}[k]\right) \quad (1)$$

$$\mathbf{y}[k+1] = \mathbf{W}_{\mathbf{r}}^{\mathbf{o}}\mathbf{x}[k+1], \quad (2)$$

where: the state of the reservoir neurons at time k is given by $\mathbf{x}[k]$; the current values of the input and output neurons are represented by $\mathbf{u}[k]$ and $\mathbf{y}[k]$, respectively; γ is called leak rate [6], which governs the percentage of the current state $\mathbf{x}[k]$ that is transferred into the next state $\mathbf{x}[k+1]$. The weights are represented in the notation $\mathbf{W}_{\mathbf{from}}^{\mathbf{to}}$, with "**b**", "**o**", "**r**", and "**i**" meaning the bias, output, reservoir, and input neurons, respectively; and $f = \tanh(\cdot)$ is an activation function widely used in the literature, also called a base function in system identification theory [1]. Fig. 1 depicts a standard architecture of an echo state network.

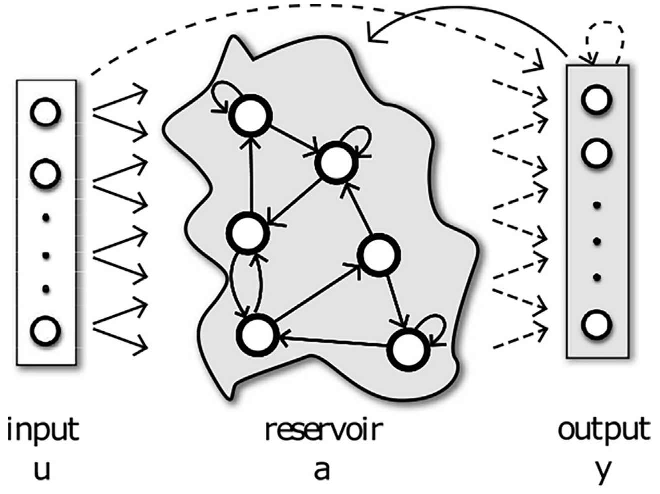


Fig. 1. Representation of an Echo State Network, one of the possible models in Reservoir Computing. Dashed connections (from Reservoir to Output Layer) are trainable, while solid connections are fixed and randomly initialized. This figure was obtained from [29].

The network has N neurons in the reservoir, which is the dimension of $\mathbf{x}[k]$ and is typically orders of magnitude higher than the number of network inputs. As long the network training uses regularization, N can be as large as needed, but at the expense of increased computation time to update the reservoir states as defined in (1). According to [18], the ESN with no output feedback connections (the output does not affect the state), which is given by $\mathbf{W}_{\mathbf{o}}^{\mathbf{r}}$, has a memory capacity (MC) bounded by the number of neurons in the reservoir ($MC \leqslant N$), assuming the use of linear output units.

The recurrent reservoir should possess the so-called Echo State Property (ESP) [28], i.e., a fading memory of its previous inputs, meaning that influences from past inputs on the reservoir states vanish with time. The ESP is guaranteed for reservoirs with $\tanh(\cdot)$ as the activation function, provided that the singular values of $\mathbf{W}_{\mathbf{r}}^{\mathbf{r}} < 1$. However, this condition limits the richness of the reservoir's dynamical qualities, which discourages its use in practice. Note that all connections going to the reservoir are randomly initialized, usually according to the following steps:

1. Every network weight is initialized from a normal distribution $\mathcal{N}(0,1)$.
2. $\mathbf{W}_{\mathbf{r}}^{\mathbf{r}}$ is scaled so that its spectral radius ρ (Eigenvalue with the largest module) characterizes a regime able to create reservoirs with rich dynamical capabilities. Setting $\rho < 1$ in practice often generates reservoirs with the ESP [6]. However, reservoirs with $\rho > 1$ can still have the ESP since the effective spectral radius may still be lower than 1 [30,31].
3. $\mathbf{W}_{\mathbf{i}}^{\mathbf{r}}$ and $\mathbf{W}_{\mathbf{b}}^{\mathbf{r}}$ are multiplied by scaling factors f_i^r and f_b^r, respectively, affecting the magnitude of the input.

These scaling parameters, ρ, f_i^r, and f_b^r are crucial in the learning performance of the network, having an impact on the nonlinear representation and memory capacity of the reservoir [32]. Also, low leak rates allow for higher memory capacity in reservoirs, while high leak rates favor quickly varying inputs and outputs. The settings of these parameters should be such that the generalization performance of the network (loss on a validation set) is enhanced.

3.2. Training

While in standard RNNs all weights are trained iteratively using gradient descent [4], ESNs restrict the training to the output layer $\mathbf{W}_{\mathbf{r}}^{\mathbf{o}}$. Because the echo state property does not emerge with output feedback $\mathbf{W}_{\mathbf{o}}^{\mathbf{r}}\mathbf{y}[k]$, this work favors reservoirs without feedback from the output, i.e., $\mathbf{W}_{\mathbf{o}}^{\mathbf{r}} = 0$. Also, the inputs do not interfere directly with the output, as systems with direct transmission are less smooth and more noise-sensitive. To train an ESN, the input data $\mathbf{u}[k]$ is arranged in a matrix $\mathbf{U}$ and the desired output $\mathbf{d}[k]$ in vector $\mathbf{D}$ over a simulation time, where each row $\mathbf{u}^T$ of $\mathbf{U}$ corresponds to a sample time k and its columns are related to the input units. For the sake of simplicity, we assume that there are multiple inputs and only one output. The rows of $\mathbf{U}$ are input into the network reservoir according to each sample time, thereby creating a state matrix $\mathbf{X}$ containing the resulting state sequence. Then, we apply the Ridge Regression algorithm [2] by using $\mathbf{X}$ as the input data matrix and $\mathbf{D}$ as the output data matrix or, in this case, a vector as we assumed single output. Ridge Regression results in solving the following linear system:

$$\left(\mathbf{X}^T\mathbf{X} - \lambda\mathbf{I}\right)\mathbf{W}_{\mathbf{r}}^{\mathbf{o}} = \mathbf{X}^T\mathbf{D}, \quad (3)$$

where λ is the Tikhonov regularization parameter, which penalizes the weight magnitude and avoids overfitting. There are also methods to apply least-squares training online [1], but this work does not use these algorithms.

4. Model Order Reduction

In this section, we propose Model Order Reduction (MOR) methods for reducing the reservoir dimensionality in ESNs, specifically the Proper Orthogonal Decomposition (POD) and the Discrete Empirical Interpolation Method (DEIM). We also propose a strategy for correcting the steady-state error introduced in ESNs by MOR methods.

4.1. Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition is a method to find a linear transformation [33] $\mathbf{T}$ for a given system that maps a high-dimensional state space into a reduced one, namely:

$$\mathbf{x} = \mathbf{T}\mathbf{z} \tag{4}$$

where $\mathbf{x}$ is a vector of dimension n and $\mathbf{z}$ is a vector of dimension $m \ll n$, so that $\mathbf{T} \in \mathbb{R}^{n\times m}$.

The transformation itself is akin to a similarity transformation, with the main difference being that $\mathbf{T}$ lacks an inverse for not being a square matrix. However, the $\mathbf{T}$ resulting from POD is orthonormal ($\mathbf{T}^T\mathbf{T} = \mathbf{I}$), so the transpose is used in place of an inverse.

To find $\mathbf{T}$, we gather snapshots of the states in a given dynamical system response, akin to gathering data in a machine learning problem. The columns of the snapshot matrix $\mathbf{X} \in \mathbb{R}^{n\times N}$ are the states $\mathbf{x}[k] \in \mathbb{R}^n$, where N is the number of snapshots such that $N \geqslant n$. Then, we wish to minimize the error induced by projecting the original state onto the reduced space and back, which leads to the following error function:

$$E(\mathbf{T}) = \sum_{k=1}^{N}\left(\mathbf{x}[k] - \mathbf{T}\underbrace{\mathbf{T}^T\mathbf{x}[k]}_{\mathbf{z}[k]}\right)^2 \tag{5}$$

The second term is $\mathbf{x}$ *projected* onto the reduced space of $\mathbf{z}$, and then *lifted* back. The optimal $\mathbf{T}$ is obtained through singular value decomposition (SVD) [34], decomposing $\mathbf{X}$ in the following form:

$$\mathbf{U_{svd}}\Sigma\mathbf{V}^T = \mathbf{X} \tag{6}$$

where $\mathbf{U_{svd}}$ contains the left singular vectors and has dimension $n \times n$, Σ contains the singular values and has dimension $n \times N$, with only n non-zero columns. We consider that Σ is sorted from the largest to the smallest singular value. POD does not use the right singular vector matrix $\mathbf{V}$.

The transformation $\mathbf{T}$ that minimizes $E(\mathbf{T})$ is found by concatenating the columns with the m largest corresponding singular values from $\mathbf{U_{svd}}$. We seek a truncation so that the reduced system energy is close to the original, measured by:

$$\epsilon = \sum_{j=1}^{m}\epsilon_j, \epsilon_j = \sigma_j / \sum_{i=1}^{n}\sigma_i \tag{7}$$

where ϵ is the total energy contribution of the singular values maintained in the reduced-order model, σ_j is the j^{th} highest singular value, ϵ_j is the energy contribution of that given singular value, and m is the reduced state dimension. The energy contribution of the remaining singular values in the reduction is a metric on how close the reduced-order model is to the original system regarding information. For this work, we measure the energy contribution of each singular value of the original signal and truncate $\mathbf{U_{svd}}$ to obtain $\mathbf{T}$ so that ϵ reaches a desired energy contribution value (*e.g.*, $\epsilon = 0.95$, so that the reduced system has 95% of the original system's energy). In other words, the reduced-order model carries ϵ information of the original system. After obtaining $\mathbf{T}$ for the dimension reduction through the process above, the reduced ESN dynamics can be expressed as follows:

$$\mathbf{z}[k+1] = (1-\gamma)\mathbf{z}[k] + \gamma\mathbf{T}^T\mathbf{f}\left(\mathbf{W_r^r}\mathbf{T}\mathbf{z}[k] + \mathbf{W_i^r}\mathbf{u}[k] + \mathbf{W_b^r}\right) \tag{8a}$$

$$\mathbf{y}[k+1] = \mathbf{W_r^o}\mathbf{T}\mathbf{z}[k+1], \tag{8b}$$

We can observe from the operation $\mathbf{T}^T\mathbf{f}(\cdot)$ that the reduced-order ESN does not reduce the number of computations by only performing POD on it. In fact, to compute the element-wise tanh, $\mathbf{T}$ brings the dimension back to the original state space size, which is to be reduced again with $\mathbf{T}^T$, increasing the number of computations. This computational increase is inherent in POD for nonlinear systems and will be dealt with by the method described in the next section.

4.2. Discrete Empirical Interpolation

The Discrete Empirical Interpolation Method (DEIM) is an approximation method to circumvent the POD computation issue [16], which consists of state projection and lifting operations to compute state transitions in the reduced-order model. The core idea of DEIM is to approximate the nonlinear term of a dynamic system as a polynomial interpolation that resembles the strategy employed in POD. Given the following discrete-time nonlinear system:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x} + \mathbf{f}(\mathbf{x}[k]), \tag{9}$$

where the nonlinear function is elementwise, meaning that

$$\mathbf{f} = (f(\mathbf{x}), f(\mathbf{x}), \ldots, f(\mathbf{x})) \tag{10}$$

for a given function f such as `tanh`. Notice that the system is divided into linear and nonlinear portions. Applying the POD ($\mathbf{x} = \mathbf{T}\mathbf{z}$) into such a system yields:

$$\mathbf{z}[k+1] = \mathbf{T}^T\mathbf{A}\mathbf{T}\mathbf{z}[k] + \mathbf{T}^T\mathbf{f}(\mathbf{T}\mathbf{z}[k]) \tag{11}$$

The nonlinear mapping $\mathbf{f}$ of the dynamic system can be approximated as follows:

$$\mathbf{P}^T\mathbf{f}(\mathbf{T}\mathbf{z}[k]) \approx \mathbf{P}^T\mathbf{U}\mathbf{c}[k] \tag{12}$$

where $\mathbf{U} \in \mathbb{R}^{n\times m}$, which is obtained from the same POD as $\mathbf{T}$, however with a different number m of singular vectors, with n being the number of states, and $\mathbf{P}$ is a pivoting matrix of the same dimension as $\mathbf{U}$. DEIM interprets that a linear combination, with basis $\mathbf{U}$ and the elements $\mathbf{c}[k]$ as function coefficients, approximates the elementwise function $\mathbf{f}$.

After obtaining $\mathbf{U}$ from $\mathbf{U_{svd}}$, we then obtain $\mathbf{P}$ with the following procedure [16]:

1. The index and value of the largest element of the first left-singular vector is stored in a list. $\mathbf{P}$ starts as a column matrix with the only non-zero element being the value 1 at the row corresponding to this index.
2. For each column $l \geqslant 2$ of the POD left-singular vectors (where $\widetilde{\mathbf{U}}_l$ is a matrix with the first $l-1$ columns of $\mathbf{U}$):
   (a) find $\mathbf{c}$ where $\left(\mathbf{P}^T\widetilde{\mathbf{U}}_l\right)\mathbf{c} = \mathbf{P}^T\mathbf{u}_l$, where $\mathbf{u}_l$ is the left-singular vector corresponding to the l^{th} column of $\mathbf{U}$.
   (b) Calculate $\mathbf{r} = \mathbf{u}_l - \widetilde{\mathbf{U}}_l\mathbf{c}$ and store the maximum absolute value and index of $\mathbf{r}$ in a list. Add a new column to $\mathbf{P}$ according to the obtained index.
3. Output: Pivoting matrix $\mathbf{P}$ according to the order dictated by the index list obtained.

This procedure guarantees that $\mathbf{P}^T\widetilde{\mathbf{U}}_l$ is always nonsingular; thus $\mathbf{c}$ is the unique solution to the linear system in step 2 [16]. Letting $\mathbf{U}$ be the matrix of left singular values obtained from the procedure, it follows from (12) that:

$$\mathbf{c}[k] = \left(\mathbf{P}^T\mathbf{U}\right)^{-1}\mathbf{P}^T\mathbf{f}(\mathbf{Tz}[k]) \tag{13}$$

The result from (13) leads to the DEIM function interpolation:

$$\hat{\mathbf{f}}(\mathbf{Tz}[k]) \approx \mathbf{U}\left(\mathbf{P}^T\mathbf{U}\right)^{-1}\mathbf{P}^T\mathbf{f}(\mathbf{Tz}[k]) \tag{14}$$

This function approximation has an ℓ_2 error bound of the following form [16]:

$$e_{\ell_2}(\mathbf{f}) \leqslant \|\left(\mathbf{P}^T\mathbf{U}\right)\|_2\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^T\right)\mathbf{f}(\mathbf{Tz}[k])\| \tag{15}$$

where, in turn:

$$\|\left(\mathbf{P}^T\mathbf{U}\right)\|_2 \leqslant \left(1 + \sqrt{2n}\right)^{m-1}\|\mathbf{u}_1\|_\infty^{-1} \tag{16}$$

with $\mathbf{u}_1$ being the first column of $\mathbf{U}$ and n being the number of original states.

The main advantage of DEIM is that, as $\mathbf{f}$ is an element-wise nonlinear function, the following equality holds:

$$\underbrace{\mathbf{U}\left(\mathbf{P}^T\mathbf{U}\right)^{-1}\mathbf{P}^T}_{\mathbf{T}_1\in\mathbb{R}^{n\times n}}\underbrace{\mathbf{f}(\mathbf{Tz}[k])}_{\mathbf{f}:\mathbb{R}^n\to\mathbb{R}^n}$$

$$= \underbrace{\mathbf{U}\left(\mathbf{P}^T\mathbf{U}\right)^{-1}}_{\mathbf{T}_2\in\mathbb{R}^{n\times m}}\underbrace{\mathbf{f}\left(\mathbf{P}^T\mathbf{Tz}[k]\right)}_{\mathbf{f}:\mathbb{R}^m\to\mathbb{R}^m} \tag{17}$$

The difference between the right-hand side and left-hand side of this equation is better seen in a compact form,

$$\mathbf{T}_1\mathbf{f}(\mathbf{Tz}[k]) = \mathbf{T}_2\mathbf{f}\left(\mathbf{P}^T\mathbf{Tz}[k]\right)$$

where $\mathbf{T_1}$ has n columns, which yields the same computation problem as the original Galerkin projection, whereas $\mathbf{T_2}$ has m columns, which is the reduced state space. This simple difference grants huge computational savings since the online calculations would be performed in terms of the reduced dimension $m, m \ll n$, which mitigates the computation issues regarding the POD method.

The DEIM-approximated reduced order ESN has the form obtained by applying DEIM from Eq. (17) into the already reduced ESN at (8):

$$\begin{aligned}\mathbf{z}[k+1] = {} & (1-\gamma)\mathbf{z}[k] \\ & + \gamma\mathbf{T}^T\mathbf{T_2}\mathbf{f}\left(\mathbf{P}^T\mathbf{W_r^r}\mathbf{Tz}[k] + \mathbf{P}^T\mathbf{W_i^r}\mathbf{u}[k] + \mathbf{P}^T\mathbf{W_b^r}\right)\end{aligned} \tag{18a}$$

$$\mathbf{y}[k+1] = \mathbf{W_r^o}\mathbf{Tz}[k+1], \tag{18b}$$

The property $\mathbf{P}^T\mathbf{f}(\cdot) = \mathbf{f}\left(\mathbf{P}^T\right)$ holds for elementwise operations, which justify the matrix placement in the DEIM reduced-order ESN.

4.3. Stability Loss in DEIM

According to [35], a contractive linear system is guaranteed to retain stability when applying POD for model order reduction; therefore, if the ESN is contractive, the POD-ESN is guaranteed to retain stability. However, DEIM has no such property. Assume an equilibrium point $\mathbf{x_{eq}}$ of the ESN, and a fixed input $\mathbf{u}$,

$$\mathbf{x_{eq}} = \mathbf{f}(\mathbf{W_r^r}\mathbf{x_{eq}} + \mathbf{W_i^r}\mathbf{u} + \mathbf{W_b^r}) \tag{19}$$

and its reduced mapping $\mathbf{z_{eq}} = \mathbf{T}^T\mathbf{x_{eq}}$. The Jacobian of the full and reduced order model are:

$$J(\mathbf{x_{eq}}) = (1-\gamma)\mathbf{I} + \gamma\mathbf{f}'(\mathbf{g}(\mathbf{x_{eq}}))\mathbf{W_r^r} \tag{20}$$

$$J(\mathbf{z_{eq}}) = (1-\gamma)\mathbf{I} + \gamma\mathbf{T}^T\mathbf{f}'(\mathbf{g}(\mathbf{Tz_{eq}}))\mathbf{W_r^r}\mathbf{T} \tag{21}$$

where:

$$\mathbf{g}(\mathbf{x}) = \mathbf{W_r^r}\mathbf{x} + \mathbf{W_i^r}\mathbf{u} + \mathbf{W_b^r} \tag{22}$$

Since $\mathbf{f}'$ is a diagonal matrix where each element belongs to the interval $(0, 1]$ for being the elementwise derivative of the tanh function, the stability of the ESN in both cases is governed by $\mathbf{W_r^r}$ at an equilibrium point. Also, as per [35], the POD reduction retains the stability of the ESN. Summing up, the original and reduced-order ESNs are stable provided that the spectral radius of $\mathbf{W_r^r}$ is smaller than 1.

With DEIM, however, the stability is not retained, as shown by calculating the Jacobian of an ESN reduced by both POD and DEIM:

$$\mathbf{J_{DEIM}}(\mathbf{z}) = (1-\gamma)\mathbf{I} + \gamma\mathbf{T}^T\mathbf{U}\left(\mathbf{P}^T\mathbf{U}\right)^{-1}\mathbf{f}'\left(\mathbf{P}^T\mathbf{g}(\mathbf{Tz})\right)\mathbf{P}^T\mathbf{W_r^r}\mathbf{T} \tag{23}$$

Notice that the term $\left(\mathbf{P}^T\mathbf{U}\right)^{-1}$ can amplify the Jacobian to the point that the ESN dynamic system has an unstable eigenvalue, despite POD-ESN being stable. This term represents the pivoting of the truncated singular vectors associated with DEIM.

5. Applications

This section presents results from experiments with reduced-order ESNs for three case studies, along with a preliminary analysis on the singular values of the ESN snapshots.

5.1. Preliminary Study: Energy contribution distribution in Echo State Networks

POD and DEIM originate from applying SVD into the ESN state response matrix, obtained from exciting the ESN's reservoir with an input signal. Thus, the SVD does not depend on the output layer. To test the influence of input signals into the singular values of the state snapshots, we initialize 20 different single-input ESN reservoirs and apply SVD into the snapshots of the response obtained from the reservoir, given as inputs with $10,000$ timesteps:

- A white noise following the normal distribution $\mathcal{N}(0, 1)$.
- Four different APRBS (Amplitude-modulated Pseudo-Random Binary Signal) random stair signals, defined by their minimum period, i.e., 10 timesteps, 100 timesteps, 500 timesteps, and $1,000$ timesteps.
- A concatenation in time of all the signals above.

The input signals for the experiments are shown in Fig. 2. Note that this discussion concerns only the state dynamics of the reservoir; therefore, it is neither dependent on the identified system nor on the output weights.

After exciting the ESN with the signals mentioned above, one at a time, we perform SVD of the resulting ESN state response snapshots and plot the energy contribution ϵ_j associated with each singular value, sorted from highest to lowest according to Eq. (7). All the reservoirs employed for this experiment are fully leaked ($\gamma = 1$), have 500 neurons, a spectral radius $\rho = 0.99$, and a value 0.1 for both input scaling and bias scaling.

Fig. 3 showcases the mean and standard deviation of the energy contribution of the 10 highest singular values for each state snapshot considering 20 randomly initialized reservoirs. We infer from this result that the singular values become more evenly distributed the higher the frequencies of the input signal are. As the white noise is a signal with heavy high-frequency information, we expect the ESN state response to having a more even energy contribution distribution among the singular values.

Meanwhile, the lower frequency signals have the energy contribution concentrated about the highest magnitude singular

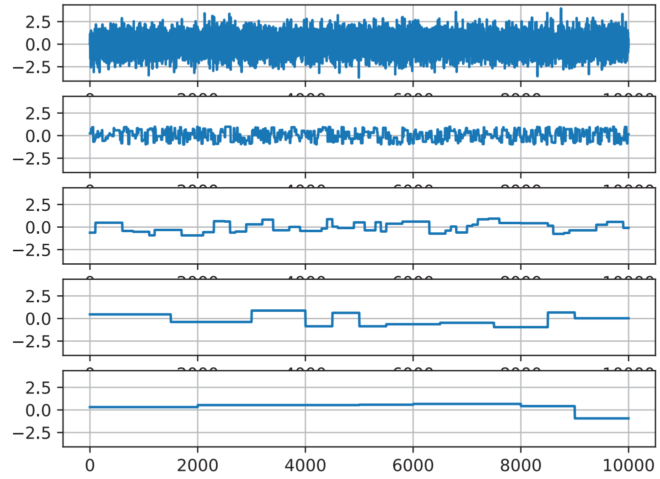


Fig. 2. One-dimensional input signals for the reservoir energy contribution distribution experiment. White noise (top), APRBS signals (usually used in identification tasks): with a minimum period of 10, 100, 500, and 1,000 timesteps, respectively, from second topmost plot to bottom.

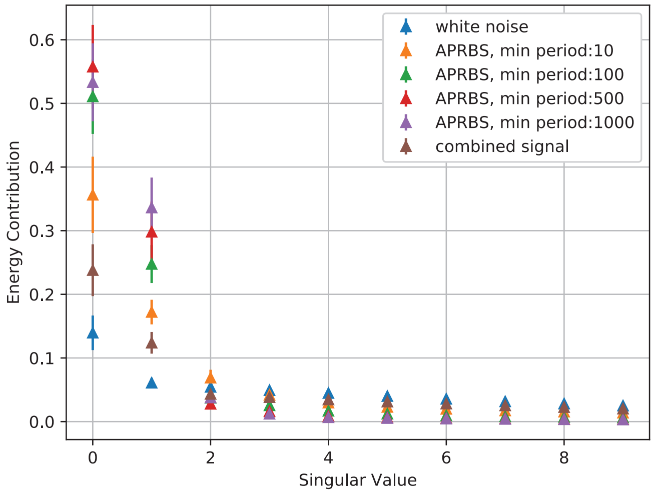


Fig. 3. Mean and Standard deviation of the first ordered 10 singular values (with 0 corresponding to the highest and 9 to the lowest) obtained from the snapshots of 20 different ESN reservoirs. Each color corresponds to a different input signal fed to the ESN reservoir, shown in Fig. 2.

value. In fact, real-life dynamic systems work as low pass filters [33] and, therefore, they are expected to have lower frequency information. The slower the system dynamics are, the larger the minimum period of an APRBS signal needs to be, which directly affects the singular value profile of the model order reduction.

This experiment implies that, since the distribution of the energy contribution depends entirely on the input signal frequency, the number of states pruned by MOR is higher for cases with low-frequency dynamics. After all, since the energy contribution is more concentrated on the first singular values, the number of columns pruned is higher than when the singular values are more evenly distributed (as in the case of high-frequency signals like white noise). As an easy example, the highest energy contribution singular value for the APRBS signal with a minimum period of $1,000$ timesteps contributes more to the total energy of the snapshots than the sum of the 10 highest singular values for the white noise shown in the plot.

5.2. Memory Capacity Evaluation

Short-term Memory Capacity (MC) is a well-known metric for Echo State Networks [18] that measures how well an ESN can remember past inputs and general dynamic storage capacity. MC serves as a performance measurement for ESN reservoirs which is obtained from the following procedure:

- For an arbitrary n, train a single-input, single-output Echo State Network so that the input is a given white noise $\eta[k]$, and the output is the same white noise delayed n timesteps $\eta[k-n]$. In layman's terms, the ESN is supposed to "memorize" the input from n timesteps ago.
- Obtain the correlation coefficient R_n for the training with an arbitrary n,

$$R_n = \frac{\mathtt{cov}(y_{esn}, \eta[k-n])}{\mathtt{var}(\mathbf{y_{esn}})\mathtt{var}(\eta[k-n])} \tag{24}$$

where $\mathtt{cov}(\cdot)$ is the covariance operator, y_{esn} is the single ESN output, $\mathtt{var}(\cdot)$ is the variance operator, and, therefore, R_n is merely the determination coefficient for a given delay n.
- The memory capacity is calculated, in theory, as:

$$MC = \sum_{n=1}^{\infty} R_n \tag{25}$$

The MC of an ESN was mathematically proven to have an upper bound in its number of neurons N [18], which means that it is directly related to the number of network neurons.

For this work, we propose an experiment to compare the memory capacity of the reduced-order model of the ESN, and the original ESN, since the number of neurons is the upper bound for MC. Because it is impossible to run infinite training experiments, we define the memory capacity for this experiment as follows:

$$MC = \sum_{n=1}^{N_{MC}} R_n \tag{26}$$

where $N_{MC} = 100$ is a sufficiently large number to measure the memory capacity of the network. As preliminary tests show, after a given n, the determination coefficient converges to a low value. Therefore, the information regarding memory capacity is more concentrated in the lower n spectrum, endorsing the limited number of experiments ($N_{MC} = 100$) for comparison purposes.

5.2.1. POD Reduction

We ran the memory capacity experiment for different numbers of neurons ($N = \{400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200\}$) with an Energy Cutoff (EC) of $1\%, 5\%$, and 10%. After initializing the ESN reservoir at random, we perform model order reduction for 12 different reservoirs in each configuration. We then measure the mean and standard deviation for the memory capacity of these twelve runs while also obtaining the range of the reduced dimension for a given energy cutoff. This analysis allows us to measure the memory capacity drop for the model order reduction and assess how reservoir-dependent the order-reduction procedure is.

All reservoirs analyzed are fully leaked ($\gamma = 1.0$) and have input and bias scaling at 0.1. Also, the reservoir spectral radius is $\rho = 0.99$.

Fig. 4 showcases the results of the Memory Capacity experiments when performing MOR at the tested ESNs given different energy cutoffs, depicting both mean and standard deviation of the 12 runs.

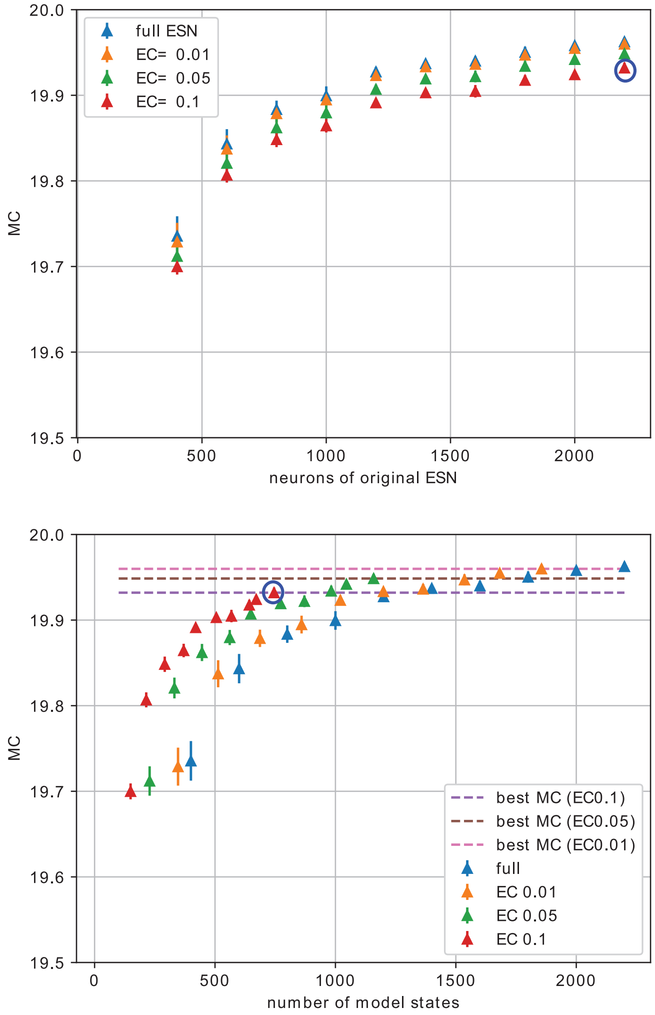


Fig. 4. Plot of the memory capacity as a function of the number of neurons of the original network (upper plot), and as a function of the number of states (lower plot). Each point is colored according to the energy cutoff of the POD-ESN that obtained the MC shown (points in blue are the MC obtained from full ESNs). EC means the energy cutoff of the applied POD.

The first plot depicts the number of ESN neurons before applying POD to a given network. It shows the expected drop in MC resulting from applying MOR with more energy cutoff.

Meanwhile, the second plot portrays the MC as a function of a given network's exact number of states after performing MOR through POD. As MC progresses monotonically, given the number of states, either in an ESN or in a given MOR of that ESN, it becomes easy to map a point of the second plot into the first one: for example, the last red point (from left to right) of both plots (marked within a blue circle) have the same memory capacity since they correspond to the same network/EC configuration. Thus, the MOR of an ESN with 2200 neurons (first plot) has roughly 750 states (second plot) at 1% energy cutoff.

As per the previous section, since this experiment traditionally employs a white noise signal, the drop in the number of reduced states is not very significant; however, the drop in MC is still small, given that a large number of states were still cut off (even in the case of 10% energy cutoff for the 2200 neuron network, the number of states was reduced to almost a third). In fact, the second plot shows that a POD-reduced network ends up being more powerful in terms of MC than a full (non-reduced) ESN with the same number of states: when we compare an ESN with a given reservoir size to a POD-reduced network from a larger ESN with the same number of states as that ESN reservoir size, the POD-reduced ESN consistently achieves a higher MC. Of course, the better performance is justifiable because a POD-reduced ESN is still more structurally complex (originated from a larger ESN) than an ESN (randomly generated) with the same number of neurons as the reduced network.

5.2.2. DEIM Reduction

We also performed DEIM for each POD-reduced ESN to further reduce the number of tanh in the computations and evaluate the drop in MC compared to the POD-reduced ESN. We tested four different energy cutoff configurations for the DEIM: $\{1\%, 5\%, 10\%, 20\%\}$. This choice of four values is justified because they represent distinct magnitudes of energy cutoff, testing how the DEIM behaves on four different approximation precision requirements.

Table 1 shows the results of applying these DEIM configurations into each POD for three original reservoir sizes $N = \{800, 1400, 2000\}$ (from the topmost table to the bottommost one, respectively). It presents the results for the DEIM reduction, where the memory capacity is evaluated for each configuration in energy cutoff for both POD and DEIM. The number in parenthesis is the actual dimension resulting from the reduction. Each column corresponds to a different energy cutoff configuration for DEIM, evaluated in the first row. In contrast, each row represents a different energy cutoff configuration for POD, evaluated in the first column. For instance, the MC of an ESN with a 1% energy cutoff POD (yielding 1,119 states when $N = 1,400$) and a 5% energy cutoff DEIM (yielding 748 tanh function evaluations when $N = 1,400$) is 0.099, 0.03 and 0.02 for $N = 800, 1400, 2000$ respectively. Notice that there was no POD reduction for the first row of each table and no DEIM reduction for the first column of each table. The empty cells indicate that DEIM can not be employed without first applying the POD reduction.

The only time DEIM achieved an MC close to the MOR was when there was a 1% energy cutoff for DEIM considering 10% energy cutoff for POD. That is, DEIM is performed for smaller reduced-order models. Regarding the experiments, performance is generally mildly better whenever DEIM has a higher number of states ratio than the POD states. For this experiment, DEIM did not perform well as expected since the white noise signal does not allow for a significant reduction of states, as it is a highly heavy information signal.

5.3. NARMA System

As an initial case study for the POD reduction of the ESN, we try to identify the behavior of a so-called NARMA (Nonlinear Autoregressive Moving Average) difference equation system [19], equated as follows:

$$y[k] = 0.3y[k-1] + 0.05y[k-1]\sum_{i=1}^{m} y[k-i] + 1.5u[k-m+1]u[k] + 0.1 \quad (27)$$

where $m = 10$ is the order of the system.

As in [19], the excitation signal applied in (27) is drawn from the random uniform distribution with a value range of $0 \leqslant u[k] \leqslant 0.05$. A simulation performs 5,000 time steps where the first 2,000 samples are labeled as training data and the rest is labeled as test data. This work employs the R^2 metric to measure network performance.

With the dataset mentioned above, we train an ESN with the following configuration: 1,400 neurons in the reservoir layer, high enough so that we show the MOR potential at work; a leak rate of $\gamma = 0.7$; scaling of 0.1 for both bias and input connections; and

Table 1
Memory capacity evaluated for different energy cutoffs used in POD and DEIM. Each table considers an original ESN with a different size N, to be reduced.

$N = 800$	Energy Cutoff (EC) for DEIM				
EC (POD)	0%	1%(678)	5%(430)	10%(279)	20%(128)
0%(800)	19.88 ± 0.01	–	–	–	–
1%(686)	19.87 ± 0.01	0.44 ± 0.20	0.099 ± 0.01	0.08 ± 0.04	0.54 ± 0.18
5%(445)	19.86 ± 0.01	16.48 ± 2.21	0.059 ± 0.026	0.096 ± 0.02	0.55 ± 0.17
10%(291)	19.84 ± 0.008	19.68 ± 0.03	0.99 ± 0.25	0.097 ± 0.02	0.54 ± 0.18
$N = 1,400$	Energy Cutoff (EC) for DEIM				
EC (POD)	0%	1%(1,186)	5%(748)	10%(484)	20%(226)
0%(1,400)	19.93 ± 0.003	–	–	–	–
1%(1,119)	19.93 ± 0.003	0.11 ± 0.03	0.03 ± 0.03	0.04 ± 0.02	0.17 ± 0.05
5%(772)	19.91 ± 0.003	3.189 ± 1.09	0.03 ± 0.02	0.04 ± 0.02	0.17 ± 0.04
10%(505)	19.90 ± 0.003	19.18 ± 0.50	0.19 ± 0.02	0.025 ± 0.02	0.17 ± 0.05
$N = 2,000$	Energy Cutoff (EC) for DEIM				
EC (POD)	0%	1%(1,835)	5%(1,122)	10%(713)	20%(333)
0%(2,000)	19.96 ± 0.002	–	–	–	–
1%(1,682)	19.95 ± 0.002	0.06 ± 0.01	0.02 ± 0.02	0.03 ± 0.01	0.03 ± 0.03
5%(1,045)	19.94 ± 0.002	1.2 ± 0.4	0.01 ± 0.02	0.02 ± 0.02	0.08 ± 0.03
10%(671)	19.92 ± 0.002	18.37 ± 0.90	0.09 ± 0.03	0.04 ± 0.01	0.07 ± 0.03

spectral radius of $\rho = 0.99$. In terms of R^2, the network had a performance of 0.95949337 for the NARMA model output. We will now carry out experiments of POD for this network to evaluate how the MOR performs in terms of R^2 concerning the original 1,400 units network.

Fig. 5 showcases the experiment regarding applying POD reduction so that the number of states of the POD-reduced ESN appears in the x axis (blue dots). For comparison, we also plotted the R^2 for the same NARMA experiment with 10 runs of full (non-reduced) ESNs with the same reservoir size as the networks that underwent POD reduction (orange triangles). The POD-ESN reduction generally achieved superior performance over the full ESN at the same reservoir size, which is understandable, as the POD-reduced ESN is not only supposed to be an emulation of a larger ESN behavior but also more complex in structure. The NARMA experiment also shows that the R^2 metric for ESNs reduced to at least 50 states is very similar to the metric achieved by the original 1,400 units ESN, i.e., the blue dots are very close to the horizontal blue line in the plot of Fig. 5 when the number of states is higher than 50.

5.4. Two Wells and One Riser Platform

We now test the MOR over the ESN for a physical problem: an oil production platform consisting of two gas-lifted oil wells and one riser, as illustrated in Fig. 6. To gather data, we utilize a composite model consisting of two well models, a riser model, and a manifold that connects the three units.

All models assume a 2-phase fluid containing gas and liquid. The well model assumes two control volumes in the gas injection annulus and the production tubing, with boundary conditions for gas-lift, reservoir, and outlet pressure. The riser model considers a horizontal pipeline and the vertical portion of the riser as two separate control volumes while assuming the inlet flow and outlet pressure as boundary conditions. The manifold assumes no load loss due to friction; therefore, it equates the sum of the output flow

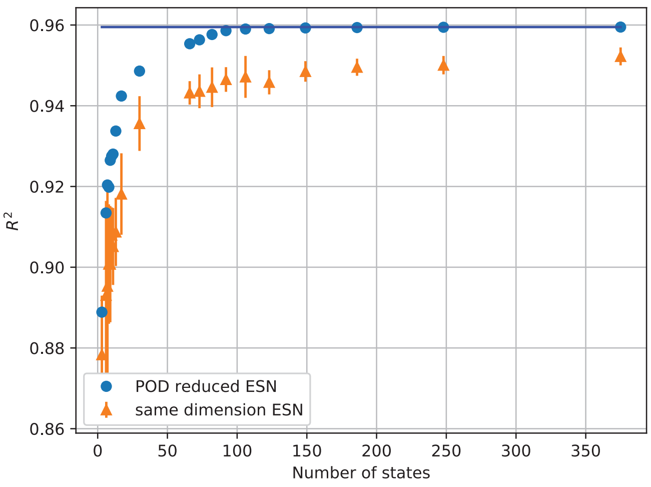


Fig. 5. Experiment comparing a POD-reduced ESN (blue dots) with an ESN of equivalent size (to the reduced ESN) (orange triangles) for the 10th-order NARMA task. The POD reduction is applied on an ESN with 1,400 units in the reservoir. The horizontal axis is the number of states (units) of the reduced (full) network, while the vertical axis is the R^2 metric on the test set. The plot's blue horizontal line corresponds to the R^2 of the 1,400 units ESN.

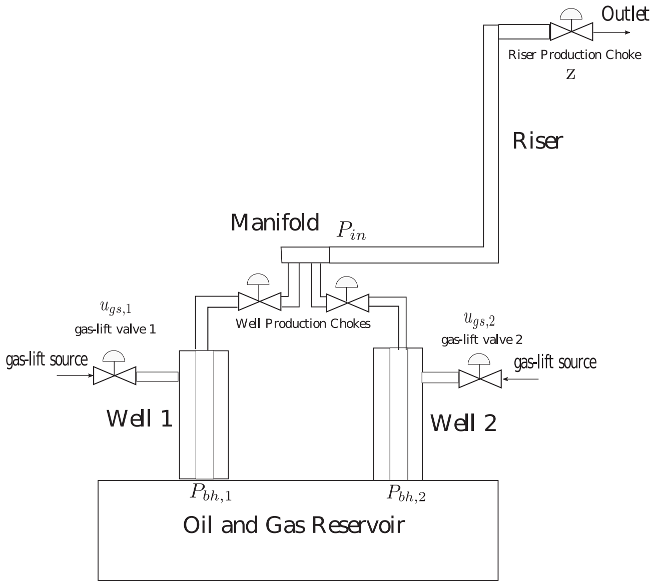


Fig. 6. Representation of an oil platform containing two wells and one riser. From [20].

from the wells to the riser input flow and the output pressure of each well to the riser inlet pressure.

Overall, the system has 120 algebraic variables, 10 state variables, 5 input variables, and precisely 5 boundary conditions. [36] presents the model in more detail, while [37] describes the riser model. The model configuration is the same as the one described in [38]. The reader can refer to these works for more details on the mathematical modeling of the platform.

The experiment with the two-well production platform depicts how to achieve MOR with ESNs from a system identification standpoint. First, we must train an ESN model for the two-well one-riser platform. We generate 50,000 timesteps of data from numerical simulation of the platform model, yielding a dataset where the 2-dimensional input to the ESN is composed of both well-production chokes $u_{ch,1}$ and $u_{ch,2}$. Further, the desired 2-dimensional output of the network corresponds to each well bottom-hole pressure: $P_{bh,1}, P_{bh,2}$. The training dataset consists of the first 10,000 timesteps, while the segment from $k = 20,000$ to $k = 30,000$ serves as a validation set, and the rest ($k > 30,000$) as a test set. With the described dataset, we train an ESN with 1,400 reservoir units (chosen this high for the sake of demonstrating the MOR potential at work), a leak rate of $\gamma = 0.7$, scalings for both bias and input equal to 0.1, and spectral radius $\rho = 0.99$. In terms of R^2 metric, the network had a test performance of $(0.99881673, 0.99900379)$ for each individual well bottom-hole pressure.

Now, we run POD experiments with the previously trained network to assess how MOR performs in terms of R^2 compared to the original 1,400 units network. Fig. 7 depicts an experiment where MOR of different state sizes was tested in terms of R^2 over the test data. One can infer that, after a given number of states (150), the performance remains consistently close to the original network in terms of R^2, despite having only 10% of the original number of states.

POD reduction that resulted in 92 states also showcased good performance compared to the original network of 1,400 neurons. However, with only POD, the computational problem of computing $\mathbf{T}^T\mathbf{f}$ remains. We select the case where the reduced network has 92 states (representing an energy cutoff of 1%) and try performing DEIM on it. Fig. 8 depicts a simulation for the ESN, POD-ESN, and POD-DEIM ESN for the test data of the two-wells and one riser platform. Even though there was a reduction from 1,400 to only 92 states, the behavior of the ESN and the POD-ESN managed to be close in terms of dynamics. The application of DEIM reduced the computation nodes from 1,400 to 1,073; however, some overshooting emerged, which was not present in the ESN and POD-ESN. Concerning the simulation run in Fig. 8, the R^2 for the normalized bottom-hole pressure of each well was: $(0.9988, 0.9990)$ for the ESN, $(0.9979, 0.9981)$ for the POD-ESN, and $(0.9873, 0.9671)$ for the DEIM-POD-ESN. There is little drop in response quality from reducing the number of states from 1,400 to 92 through POD, but performing interpolation from a standard POD to a POD-DEIM framework seems to affect the response more significantly. The small drop in response quality concerning the POD-ESN is expected, as the POD was performed requesting a 1% energy cutoff. In other words, the reduced-order model is 99% close to the original ESN regarding dynamic information.

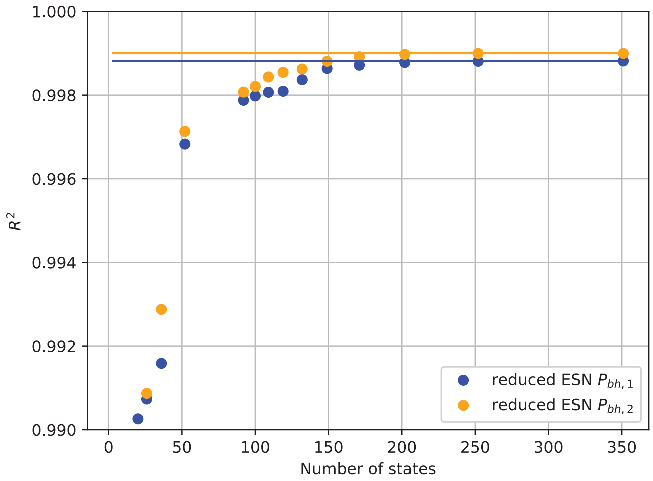


Fig. 7. POD-ESN for a system identification task. The full ESN network has 1,400 neurons and was trained to model the platform with two wells and one riser. The x axis is the number of states of the reduced network, whereas the y axis is the R^2metric on the test set for each output variable (bottom-hole pressures). The bottom-hole pressure of the first well is represented in blue, while the orange color denotes the bottom-hole pressure of the second well. The R^2 of the original network corresponds to the horizontal lines of the respective colors for comparison.

6. Discussion

POD-reduced ESN achieved a response close to the original ESN for the NARMA and the two-well one-riser case study, while it incurred a minor performance loss in the MC experiments. However, DEIM did not reach the same performance as POD in those experiments. These findings indicate that DEIM incurs more dynamic-information loss than POD, as the latter retains the number of activation functions in the reduced model even though it reduces the number of states. Thus, we conjecture that the capacity of a reservoir to represent a nonlinear system accurately is more influenced by the combination of the nonlinear functions in a high-dimensional space than by maintaining a high-dimensionality of the reservoir states themselves. In the context of MOR, this function combination is given by lifting the reduced states back to the original space just before applying the tanh nonlinearity.

The application of POD leads to some reduction in the memory required for storing and using the POD-reduced ESN. First, the state-to-output linear combination matrix $\mathbf{W_r^oT}$ maps the reduced space directly to the output, invariably reducing its size. Also, the computation of the activation functions becomes slightly less expensive memory-wise because the resulting matrix $\mathbf{W_r^rT}$, which is a product computed offline, has fewer elements. Of course, the re-

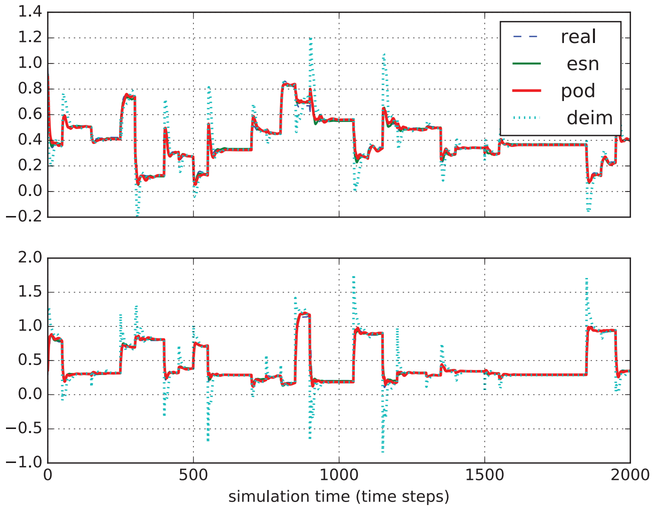


Fig. 8. Single simulation run involving a POD with 92 states (0.01 energy cutoff) and a DEIM interpolation with $m = 1,073$, put side by side with the original data for the bottom hole pressure p_{bh} of both wells (normalized), and the original ESN.

Table 2
Mean execution time for the NARMA experiment composed of 5,000 time steps.

	Mean Execution Time (ms)	St. Dev. (ms)
ESN (size = 1400)	0.767	0.537
POD (size = 3)	0.072	0.0498
POD (size = 6)	0.078	0.0251
POD (size = 7)	0.141	0.391
POD (size = 8)	0.131	0.340
POD (size = 9)	0.160	0.543
POD (size = 10)	0.140	0.467
POD (size = 11)	0.233	0.955
POD (size = 13)	0.105	0.122
POD (size = 17)	0.106	0.0738
POD (size = 30)	0.135	0.0856
POD (size = 66)	0.147	0.254
POD (size = 73)	0.144	0.138
POD (size = 82)	0.141	0.140
POD (size = 92)	0.155	0.109
POD (size = 106)	0.151	0.0744
POD (size = 123)	0.149	0.0907
POD (size = 149)	0.183	0.407
POD (size = 186)	0.201	0.145
POD (size = 248)	0.230	0.134
POD (size = 375)	0.408	0.199

sulting matrix is still large compared to an ESN with the same size as the reduction, rendering the same-size ESN less complex than the POD-reduced one.

Even though POD computes the same number of activation functions as the original ESN, the computation time is significantly reduced, as shown in Table 2. This table shows the mean time it took to execute a step in the full ESN against the time it took to perform a POD-ESN computation step for the NARMA experiment. For instance, when applying POD-ESN to reduce from 1400 states to 66 states, we get an 80% decrease in mean execution time (from 0.767 ms to 0.147 ms) while still maintaining excellent performance, as this setup is near the horizontal line in Fig. 5. All experiments were performed under similar conditions and with the same computer.

As shown in Table 2, even though there is no computation reduction in the nonlinear nodes, the computational time for a POD-ESN to compute a time step is reduced, even if by a small margin. This computation-speed gain happens precisely because the reduced-order ESN has fewer states, despite the nonlinear node computation remaining unchanged.

As previously discussed, the poor performance of DEIM in the memory capacity experiments corroborates the loss of stability incurred in the DEIM-reduced ESNs. Besides, even when the DEIM-reduced ESN dynamic system remained stable, as in the two-well experiment illustrated in Fig. 8, the system experienced high overshoots translating into modeling errors. The independent work [39] that also implements POD/DEIM on ESN, which appeared in the literature during the writing of this research, proposes a method to deal with the stability issue. However, the method is restricted to the particular class of ESNs with dynamic equations without the bias term. That method relies on expanding the nonlinear dynamics reduced by the DEIM so that the Jacobian contribution of the terms affected by $\left(\mathbf{P}^T\mathbf{U}\right)$ becomes null concerning $\mathbf{u} = \mathbf{0}$. In this context, generalized methods (which account for the bias term as well) to guarantee stability retention of an ESN interpolated by DEIM are an interesting topic for future works.

7. Conclusion

In this investigation, the POD achieved exceptional results in reducing the number of states of an ESN and maintaining performance. The reduced ESN performed nearly as well as the original ESN, despite the drastic reduction of states in a typical system identification task. This work also showcased how the nature of the excitation signal changes the singular value profile of the SVD, concluding that lower-frequency input signals can result in more efficient reductions. Ideally, the excitation signal should be as slow as necessary to identify a system.

However, despite performing MC tests considering signals that carry information from all frequencies, the POD-reduced network performed better than an ESN of the same size trained on the data. Arguably, the superior performance of the POD-reduced ESN may be attributed to its ability to emulate the behavior of the larger original ESN. Additionally, the increased complexity of the reduced network, compared to an ESN of the same size, could contribute to its enhanced performance.

These findings imply that applying POD to reduce the number of states (reservoir size) of an ESN is an excellent strategy to obtain a smaller model that behaves almost equivalently to the original one. However, some adaptation to the DEIM method may be necessary before it can be applied to increase model efficiency further. Also, reducing the reservoir size using POD has the advantage of interpretability since the states are sorted and pruned according to the energy contribution metric. Finally, applying POD to an ESN can show which linear combination of states contributes more significantly to the ESN dynamic behavior.

For possible future work, we will test the developed POD-ESN model in predictive control applications, comparing the performance of the reduced-order model to its full-order counterpart. Further, there are applications in reservoir computing, such as time series prediction problems, which could benefit from a reservoir reduction using the POD-ESN. Another direction for future research is the study of ways to adapt DEIM to perform model reductions more consistently.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded in part by FAPESC (grant 2021TR2265), CAPES (grant 88882.182533/2011-01), and CNPq (grant 308624/2021-1).

References

[1] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models, 1 ed., Springer, Berlin, 2001.
[2] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag Inc., New York, 2006.
[3] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
[4] M.C. Mozer, A focused backpropagation algorithm for temporal pattern recognition, Complex Systems 3 (1989) 349–381.
[5] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[6] H. Jaeger, M. Lukosevicius, D. Popovici, U. Siewert, Optimization and applications of echo state networks with leaky-integrator neurons, Neural Networks 20 (2007) 335–352, https://doi.org/10.1016/j.neunet.2007.04.016.
[7] W. Maass, Liquid state machines: Motivation, theory, and applications, in: Computability in Context, Imperial College Presss, 2011, pp. 275–296, https://doi.org/10.1142/9781848162778_0008.
[8] E.A. Antonelo, B. Schrauwen, On learning navigation behaviors for small mobile robots with reservoir computing architectures, IEEE Transactions on Neural

Networks and Learning Systems 26 (2015) 763–780, https://doi.org/10.1109/TNNLS.2014.2323247.

[9] R. Mezzi, N. Yousfi-Steiner, M.C. Péra, D. Hissel, L. Larger, An echo state network for fuel cell lifetime prediction under a dynamic micro-cogeneration load profile, Applied Energy 283 (2021), https://doi.org/10.1016/j.apenergy.2020.116297.

[10] Y. Bai, M.-D. Liu, L. Ding, Y.-J. Ma, Double-layer staged training echo-state networks for wind speed prediction using variational mode decomposition, Applied Energy 301 (2021), https://doi.org/10.1016/j.apenergy.2021.117461.

[11] S.F. Stefenon, L.O. Seman, N.F. Sopelsa Neto, L.H. Meyer, A. Nied, K.-C. Yow, Echo state network applied for classification of medium voltage insulators, International Journal of Electrical Power & Energy Systems 134 (2022), https://doi.org/10.1016/j.ijepes.2021.107336.

[12] R. Gao, P. Suganthan, Q. Zhou, K. Fai Yuen, M. Tanveer, Echo state neural network based ensemble deep learning for short-term load forecasting, in: IEEE Symposium Series on Computational Intelligence (SSCI), 2022, pp. 277–284. DOI: 10.1109/SSCI51031.2022.10022067.

[13] H. Wang, Y. Liu, P. Lu, Y. Luo, D. Wang, X. Xu, Echo state network with logistic mapping and bias dropout for time series prediction, Neurocomputing 489 (2022) 196–210, https://doi.org/10.1016/j.neucom.2022.03.018.

[14] Z. Li, G. Tanaka, Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction, Neurocomputing 467 (2022) 115–129, https://doi.org/10.1016/j.neucom.2021.08.122.

[15] E. Camacho, C. Bordons, Model Predictive Control, Springer, 1999.

[16] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (2010) 2737–2764, https://doi.org/10.1137/090766498.

[17] Y. Wang, B. Yu, Y. Wang, Acceleration of gas reservoir simulation using proper orthogonal decomposition, Geofluids 2018 (2018) 1–15, https://doi.org/10.1155/2018/8482352.

[18] H. Jaeger, Short term memory in echo state networks, Technical Report GMD Report 152, German National Research Center for Information Technology, 2002.

[19] Y. Sakemi, K. Morino, T. Leleu, K. Aihara, Model-size reduction for reservoir computing by concatenating internal states through time, Scientific Reports 10 (2020), https://doi.org/10.1038/s41598-020-78725-0.

[20] J.P. Jordanou, E.A. Antonelo, E. Camponogara, Online learning control with echo state networks of an oil production platform, Eng. Appl. Artif. Intell. 85 (2019) 214–228, https://doi.org/10.1016/j.engappai.2019.06.011.

[21] B. Whiteaker, P. Gerstoft, Reducing echo state network size with controllability matrices, Chaos: An Interdisciplinary, Journal of Nonlinear Science 32 (2022), https://doi.org/10.1063/5.0071926.

[22] W. Liu, Y. Bai, X. Jin, X. Wang, T. Su, J. Kong, Broad echo state network with reservoir pruning for nonstationary time series prediction, Computational Intelligence and Neuroscience 2022 (2022) 1–15, https://doi.org/10.1155/2022/3672905.

[23] C. Yang, Z. Wu, Multi-objective sparse echo state network, Neural Computing and Applications (2022), https://doi.org/10.1007/s00521-022-07711-6.

[24] A. Rodan, P. Tino, Minimum complexity echo state network, IEEE Transactions on Neural Networks 22 (2011) 131–144, https://doi.org/10.1109/TNN.2010.2089641.

[25] S. Løkse, F.M. Bianchi, R. Jenssen, Training echo state networks with regularization through dimensionality reduction, Cognitive Computation 9 (2017) 364–378, https://doi.org/10.1007/s12559-017-9450-z.

[26] A. Haluszczynski, J. Aumeier, J. Herteux, C. Räth, Reducing network size and improving prediction stability of reservoir computing, Chaos: An Interdisciplinary, Journal of Nonlinear Science 30 (2020), https://doi.org/10.1063/5.0006869.

[27] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication, Science 304 (2004) 78–80, https://doi.org/10.1126/science.1091277.

[28] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks – with an Erratum note, Fraunhofer Institute for Autonomous Intelligent Systems, 2001, Technical Report GMD 148,.

[29] E.A. Antonelo, E. Camponogara, B. Foss, Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells, Neural Networks 85 (2017) 106–117.

[30] M.C. Ozturk, D. Xu, J.C. Príncipe, Analysis and design of echo state networks, Neural Computation 19 (2007) 111–138, https://doi.org/10.1162/neco.2007.19.1.111.

[31] D. Verstraeten, B. Schrauwen, On the quantification of dynamics in reservoir computing, in: C. Alippi, M. Polycarpou, C. Panayiotou, G. Ellinas (Eds.), Artificial Neural Networks, 2009, pp. 985–994.

[32] D. Verstraeten, J. Dambre, X. Dutoit, B. Schrauwen, Memory versus non-linearity in reservoirs, in: Int. Joint Conference on Neural Networks, IEEE, Barcelona, Spain, 2010, pp. 18–23, https://doi.org/10.1109/IJCNN.2010.5596492.

[33] C.-T. Chen, Linear System Theory and Design, 3rd ed., Oxford University Press Inc, New York, NY, USA, 1998.

[34] X. Sun, M. Xu, Optimal control of water flooding reservoir using proper orthogonal decomposition, Journal of Computational and Applied Mathematics 320 (2017) 120–137, https://doi.org/10.1016/j.cam.2017.01.020.

[35] R.C. Selga, B. Lohmann, R. Eid, Stability preservation in projection-based model order reduction of large scale systems, European Journal of Control 18 (2012) 122–132, https://doi.org/10.3166/ejc.18.122-132.

[36] E. Jahanshahi, S. Skogestad, H. Hansen, Control structure design for stabilizing unstable gas-lift oil wells, IFAC Proceedings Volumes 45 (2012) 93–100, https://doi.org/10.3182/20120710-4-SG-2026.00110.

[37] E. Jahanshahi, S. Skogestad, Simplified dynamical models for control of severe slugging in multiphase risers, IFAC Proceedings Volumes 44 (2011) 1634–1639, https://doi.org/10.3182/20110828-6-IT-1002.00981.

[38] J.P. Jordanou, E.A. Antonelo, E. Camponogara, Echo state networks for practical nonlinear model predictive control of unknown dynamic systems, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 2615–2629, https://doi.org/10.1109/TNNLS.2021.3136357.

[39] H. Wang, X. Long, X.-X. Liu, fastESN: Fast echo state network, IEEE Transactions on Neural Networks and Learning Systems (2022), https://doi.org/10.1109/TNNLS.2022.3167466.

Jean P. Jordanou received the M.Sc. degree in automation and systems engineering from the Federal University of Santa Catarina, Brazil, in 2019. He joined as a Ph.D. student at the same institution rightly afterwards. His research interests include data-driven algorithms for optimization and control, reservoir computing, model order reduction and model predictive control.

Eric Aislan Antonelo received the Ph.D. and M.Sc. degrees in Computer Engineering respectively from Ghent University, Belgium, in 2011 and Halmstad University, Sweden, in 2006. He is currently a faculty member of the Department of Automation and Systems Engineering at the Federal University of Santa Catarina, Brazil. His research is mainly focused on reservoir computing and machine learning for industrial applications (modeling, detection, and control tasks), as well as imitation learning approaches for autonomous vehicles.

Eduardo Camponogara received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, USA, in 2000. He is a professor in the Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florian?polis, Brazil, since 2002. His research interests include systems optimization, distributed optimization algorithms, and data-driven algorithms for optimization and control.

Dr. Eduardo Gildin is a Professor of Petroleum Engineering at Texas A&M University and is the holder of the L.F. Peterson '36 Professorship in Petroleum Engineering. Dr. Gildin received his PhD from The University of Texas at Austin in Aerospace Engineering and has held post-doctoral positions with Rice University and UT Austin. His research has been supported by grants from NSF, DOE, DOD, NASA and Industry, with main topics in (1) physics-based and data-driven reduced-order modeling for reservoir simulation and optimization; and (2) drilling modeling, control and automation. Dr. Gildin was inducted into the SPE Distinguish Membership in 2021.