



Nonlinear MIMO System Identification with Echo-State Networks

Bernardo Barancelli Schwedersky¹  · Rodolfo César Costa Flesch²  · Hiago Antonio Sirino Danguì¹

Received: 2 April 2021 / Revised: 9 October 2021 / Accepted: 19 November 2021 / Published online: 3 January 2022
© Brazilian Society for Automatics–SBA 2021

Abstract

This paper presents a formalization of steps for identification of echo-state network (ESN)-based models for multiple-input and multiple-output (MIMO) nonlinear dynamic systems models, employing the best practices found in the literature. The ESN is a recurrent neural network capable to model nonlinear dynamics and it can be trained using computationally efficient algorithms. A simplified ESN architecture from the literature for process identification tasks is used and a procedure to create the model and tune the main hyperparameters is proposed. The proposed method is evaluated using both a simulated pH neutralization process and a real refrigeration compressor test rig. For each case study, the procedure to obtain the data series, the model creation, and the influence of each hyperparameter in the model performance are discussed. In both cases, the proposed model architecture presented better results than the linear model and the two nonlinear models used as baseline, namely an extreme learning machine-based Hammerstein model and a long short-term memory network model. The results show that the proposed ESN-based system identification method can be used to obtain MIMO models for nonlinear processes using a computationally efficient training algorithm. For the ESN architecture considered in the case studies, the training time is about 80 s, while a prediction can be obtained in about 15 ms.

Keywords System identification · Echo-state networks · Nonlinear systems · Recurrent neural networks · Ridge regression

1 Introduction

The modeling and identification of dynamic systems are of fundamental importance in engineering. Tasks, such as the analysis of existing processes, the design of new processes and controllers, optimization, simulation, fault detection, among others, are based on process models and directly impacted by the quality of the obtained models. In the cases where obtaining a phenomenological model is complex, system identification techniques are typically employed (Schoukens and Ljung 2019; Nelles 2001).

Many processes exhibit static and dynamic nonlinear behaviors, predominantly when large changes in operating conditions are considered. Recurrent artificial neural networks (RNN) are an interesting option for modeling systems

based on data, since they allow the identification of nonlinear processes in which there is little or no knowledge of process governing physics (Ljung et al. 2020; Yu 2004; Isermann and Münchhof 2011).

The application of RNNs in the identification of dynamic systems is a vast topic and has been widely investigated in the literature. Shortly after the proposition of the multilayer perceptron (MLP) network, the first studies using RNN for system identification emerged and they considered recurrent architectures derived from the classic MLP (Fernandez et al. 1990; Ayoubi 1994). Although recurrent MLP networks are still used in system identification problems, such as in Bousaada et al. (2018), which use them to identify the daily solar radiation dynamics, new RNN architectures have been proposed and enable to obtain more easily identified models with good accuracy. One of such architectures which stands out due to its computationally efficient training procedure is the echo-state network (ESN), proposed in Jaeger (2001).

Some approaches for identifying nonlinear systems using ESN have been presented in the literature. In Jaeger (2003), the use of the ESN network in an online identification approach was first proposed. In Rodan and Tino (2010), some simplified ESN architectures were used for system

✉ Rodolfo César Costa Flesch
rodolfo.flesch@ufsc.br

¹ Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil

² Department of Automation and Systems Engineering, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil

identification. Recently, the study presented in Yang et al. (2019) proposed methods for online identification using ESN models with sparse recursive algorithms. Some applications which make use of this modeling approach to identify different types of dynamic systems were also reported recently, such as wind generation systems (Chen et al. 2019), oil wells (Antonelo et al. 2017), and cooling systems (Schwedersky et al. 2018).

Even though there are some examples of ESN application for system identification in the literature, this type of application is still incipient and the works that use it present specific architectures and model tuning methods for each application. As an approach to unify system identification tasks based on ESNs, in this paper we present a general architecture for identifying multiple-input and multiple-output (MIMO) nonlinear dynamical systems using ESNs. We provide a formalization of the best practices in the literature to create the ESN model, to guarantee that its reservoir is stable and has rich dynamics, and propose a tuning method to define the model main hyperparameters. In Sect. 2, a general simplified architecture is presented and includes the procedure to build ESN models and a method for tuning the main hyperparameters. The application of the proposed method is demonstrated through two case studies detailed in Sect. 3: the identification of a pH neutralization process, and the identification of a real industrial process, which is a test rig for hermetic refrigeration compressors. The results of the proposed method are compared with those presented by traditional system identification techniques, such as linear and nonlinear models, in Sect. 4. The conclusions are presented in Sect. 5.

2 System Identification with Echo-State Networks

The echo-state network (ESN) consists of a recurrent neural network architecture which is based on the reservoir computation (Lukoševičius and Jaeger 2009), an alternative learning paradigm to the traditional RNN paradigm. It is based on the use of a nonlinear dynamic system with randomly generated fixed weights to map the inputs into a high-dimensional space, in which the classification or regression task is easier to perform (Lukoševičius 2012).

The states of this dynamic system make up a structure generally denominated as the reservoir. Such structure can be understood as a temporal kernel, which projects the input into a dynamic nonlinear space. During the model operation, the reservoir states evolve in a trajectory dependent on external stimuli and also on the memory of past stimuli. The network output is obtained using an output layer, which processes the instantaneous states of the reservoir. A general representation of the ESN network is presented in Fig. 1.

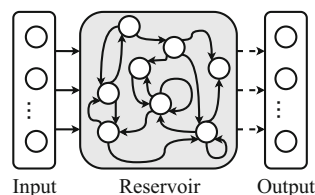


Fig. 1 ESN model architecture. Solid lines represent fixed connections, while dashed lines represent trainable weights

2.1 Model Structure

The ESN models have, generally, two main structures, the reservoir and its readout mechanism. The reservoir consists of a neural network with recurrent connections, which is responsible for its dynamic behavior. The connections between the reservoir neurons are fixed and randomly generated prior to the training phase, so the only trainable elements of the network are the weights and biases that make up the reservoir readout mechanism.

The ESN network can be described by a pair of equations, defining the state update and the model output. The state update equation

$$\mathbf{x}_e(k) = (1 - \alpha)\mathbf{x}_e(k - 1) + \alpha\mathbf{f}(\mathbf{W}_{\text{in}}^{\text{res}}\mathbf{u}(k) + \mathbf{W}_{\text{res}}^{\text{res}}\mathbf{x}_e(k - 1) + \mathbf{W}_{\text{bias}}^{\text{res}}), \quad (1)$$

describes the dynamic behavior of the reservoir states, where: $\mathbf{u}(k) \in \mathbb{R}^{N_u}$ is the input vector at time step k ; $\mathbf{x}_e(k) \in \mathbb{R}^{N_x}$ is the reservoir state vector; α is the reservoir leak rate; $\mathbf{f}(\cdot)$ is the neuron activation function; $\mathbf{W}_{\text{in}}^{\text{res}} \in \mathbb{R}^{N_x \times N_u}$ and $\mathbf{W}_{\text{bias}}^{\text{res}} \in \mathbb{R}^{N_x \times 1}$ are matrices that connect the input and bias with the reservoir, respectively; and $\mathbf{W}_{\text{res}}^{\text{res}} \in \mathbb{R}^{N_x \times N_x}$ is the matrix that represents the reservoir recurrent connections.

In this formulation, the output is considered as a linear combination of the states, which have a nonlinear update mechanism, and the corresponding biases as

$$\mathbf{y}_e(k) = \mathbf{W}_{\text{res}}^{\text{out}}\mathbf{x}_e(k) + \mathbf{W}_{\text{bias}}^{\text{out}}, \quad (2)$$

where $\mathbf{y}_e(k) \in \mathbb{R}^{N_y}$ is the network output vector; matrix $\mathbf{W}_{\text{res}}^{\text{out}} \in \mathbb{R}^{N_y \times N_x}$ represents the connections between reservoir and output; and $\mathbf{W}_{\text{bias}}^{\text{out}} \in \mathbb{R}^{N_y \times 1}$ is the matrix with output bias connections. However, it is also possible to consider a nonlinear mapping from the states to the outputs, as detailed in Lukoševičius (2012).

The general choice for the activation function, $\mathbf{f}(\cdot)$, is a sigmoid function, and the hyperbolic tangent function is the most common option (Jaeger et al. 2007). Other functions can be used to activate the reservoir states, such as the linear function (Inubushi and Yoshimura 2017; Ganguli et al. 2008) and a combination of sigmoid and linear functions (Lun et al. 2015). Self-normalizing activation functions can

also be used, following the architecture proposed in Verzelli et al. (2019).

The reservoir creation is fundamental to the model success, as the reservoir connection weights are fixed. The non-trainable connections, \mathbf{W}_{res}^{res} , \mathbf{W}_{in}^{res} , \mathbf{W}_{out}^{res} , and \mathbf{W}_{bias}^{res} , are generated following a specific distribution, generally uniform. The main design parameters are the connection rate (c_{from}^{to}) and the scaling (v_{from}^{to}), which define the matrix sparsity and the connection strength, respectively (Lukoševičius 2012).

The recurrent weight matrix, \mathbf{W}_{res}^{res} , which defines the reservoir neuron connections, deserves special attention. Its main design parameter is the spectral radius, $\rho(\mathbf{W}_{res}^{res})$, which corresponds to the largest absolute value of the eigenvalues of \mathbf{W}_{res}^{res} , thus being directly associated with the stability of the reservoir matrix. Generally, matrix \mathbf{W}_{res}^{res} is created following a predetermined sparsity, being further re-scaled to guarantee that its largest eigenvalue is equal to the desired spectral radius.

An important condition for the successful creation of an ESN is the design of a reservoir that presents the echo-state property (ESP). This property implies that the reservoir states converge asymptotically, under a driving input. For ESNs that do not use leaky neurons in the reservoir, i.e., $\alpha = 1$, a sufficient condition for the echo-state property is $\rho(|\mathbf{W}_{res}^{res}|) < 1$, with $|\mathbf{W}_{res}^{res}|$ representing the matrix formed by the absolute values of each original matrix element. For the case which considers leaky units, a sufficient condition to obtain the echo-state property is $\rho(\mathbf{M}) < 1$, with $\mathbf{M} = |\mathbf{W}_{res}^{res}| + (1-\alpha)\mathbf{I}$, where \mathbf{I} represents the identity matrix with the same dimension of \mathbf{W}_{res}^{res} (Yildiz et al. 2012). More details about the echo-state property and the ESN model initialization are found in Yildiz et al. (2012), Wainrib and Galtier (2016), and Lukoševičius (2012).

The reservoir time scale can be adjusted using the parameter α , the leak rate. This parameter defines the amount of the last reservoir state that will be preserved, as these neurons are leaky integrator units. By selecting $\alpha \in (0, 1]$, the reservoir memory is changed, with lower values resulting in a reservoir with more memory, while values closer to 1 result in less memory.

2.2 Model Training

The ESN output is formed by a linear combination of the reservoir internal states plus a bias. The reservoir readout mechanism, \mathbf{W}_{res}^{out} , and the bias matrix associated with the output, \mathbf{W}_{bias}^{out} , are the only trainable matrices in the ESN, and the ridge regression is the most commonly used alternative to obtain such matrices (Lukoševičius 2012). As an alternative for the ESN training, the least absolute shrinkage and selection operator (LASSO) regression can also be applied (Qiao et al. 2018).

As the reservoir readout weights are the only trainable parameters in the ESN model, the ESN can be trained by solving a regular least squares problem, which is computationally efficient in comparison with the training algorithms used for classical RNNs. The recursive estimation of the reservoir readout weights using the recursive least squares method is also possible, as described in (Jaeger 2003). Besides being computationally efficiency, the ESN training mechanism avoids some drawbacks verified in RNN training algorithms based on the backpropagation algorithm, such as the vanishing and exploding gradient problem (Hochreiter 1998; Lukoševičius and Jaeger 2009).

Considering $\mathbf{W}^{out} = [\mathbf{W}_{bias}^{out} \ \mathbf{W}_{res}^{out}]$, the ESN output can be written as

$$\mathbf{Y}_e = \mathbf{W}^{out} \mathbf{X}_e, \tag{3}$$

with the matrix $\mathbf{Y}_e \in \mathbb{R}^{N_y \times T}$ being defined as

$$\mathbf{Y}_e = [\mathbf{y}_e(k) \ \mathbf{y}_e(k+1) \ \dots \ \mathbf{y}_e(k+T-1)], \tag{4}$$

where $\mathbf{X}_e \in \mathbb{R}^{(1+N_x) \times T}$ is a matrix formed by the reservoir states x_e and an always active input, written as

$$\mathbf{X}_e = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \mathbf{x}_e(k) & \mathbf{x}_e(k+1) & \dots & \mathbf{x}_e(k+T-1) \end{bmatrix}, \tag{5}$$

both generated from a reservoir excited by $\mathbf{u}_e(k)$ during a training period with T samples.

The learning procedure consists of finding the optimal value of $\mathbf{W}^{out} \in \mathbb{R}^{N_y \times (1+N_x)}$ which minimizes the root mean-square error between $\mathbf{y}_e(k)$ and $\mathbf{y}(k)$, for all instants $k, k+1, \dots, k+T-1$.

If the ridge regression method is used, matrix \mathbf{W}^{out} is defined as (Lukoševičius 2012)

$$\begin{aligned} \mathbf{W}^{out} = \underset{\mathbf{W}^{out}}{\operatorname{argmin}} \quad & \frac{1}{N_y} \sum_{i=1}^{N_y} \left(\sum_{n=1}^T (y_{e_i}(n) - y_i(n))^2 \right. \\ & \left. + \beta \|\mathbf{w}_i^{out}\|^2 \right) \\ \text{s.t.} \quad & y_{e_i} = \mathbf{w}_i^{out} \begin{bmatrix} 1 \\ \mathbf{x}_e \end{bmatrix} \quad \forall i \in 1 \dots N_y \end{aligned} \tag{6}$$

with $y_{e_i}(n)$ and $y_i(n)$ representing the i -th network and measured outputs, respectively; \mathbf{w}_i^{out} being the i -th row of \mathbf{W}^{out} , representing the reservoir readout weights and bias associated with the i -th output; and the regularization term, β , being used to penalize weights with large absolute values, which contributes to avoiding undesired overfitting. By solving this optimization problem, through the objective function minimization, the reservoir readout weight matrix \mathbf{W}^{out} is obtained. A solution for this optimization problem can be described as

$$\mathbf{W}^{\text{out}} = (\mathbf{X}_e^T \mathbf{X}_e + \beta \mathbf{I})^{-1} \mathbf{X}_e^T \mathbf{Y}, \quad (7)$$

where matrix \mathbf{Y} represents the target values for the model outputs and has the same structure of \mathbf{Y}_e .

This regularization approach is advised when there is risk of overfitting or feedback instability. Extremely large $\mathbf{w}_i^{\text{out}}$ values are an indicator of a very sensitive solution, which can easily become unstable. This is usually a result of overfitting to process noise and in some cases can be avoided with a proper regularization method (Lukoševičius 2012). Other literature approaches can also be used to improve the ESN robustness, such as using special loss functions (Li et al. 2012; Guo et al. 2017; Han and Xu 2018), using nonlinear reservoir readouts as a replacement for the conventional linear output layer, like the formulations based on the support vector machine (Shi and Han 2007) and kernel adaptive filtering (Zhou et al. 2018). There are also robust ESN alternatives based on the recursive estimation of the output layer weights, using the recursive Least M-estimate algorithm (Bessa and Barreto 2019).

2.3 System Identification Procedure

For the identification of nonlinear dynamic systems, the ESN can be applied in a standard system identification scheme, as shown in Fig. 2. The training of the ESN model uses examples of the process dynamics, organized as a data series of process inputs, $\mathbf{u}(k) \in \mathbb{R}^{N_u}$, and outputs, $\mathbf{y}(k) \in \mathbb{R}^{N_y}$, at time step k obtained through an identification test procedure. A perturbation signal, $\mathbf{d}(k)$, may also affect the system output, but it cannot be measured; otherwise, it can be considered as part of $\mathbf{u}(k)$ for system identification purposes. The ESN input layer can be used to scale the inputs, so that the reservoir is excited using values with proper amplitudes, with its output exciting the reservoir. The reservoir states and the recorded process outputs are used to obtain the reservoir readout weights, as presented in Sect. 2.2.

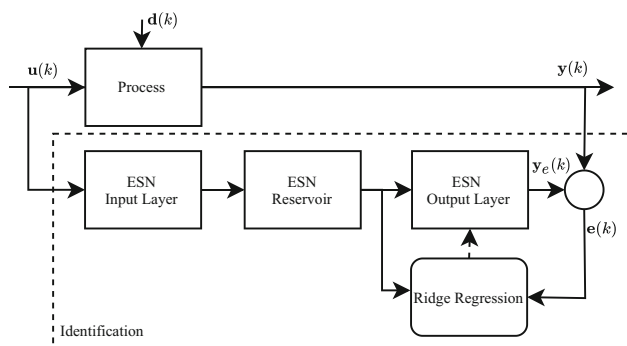


Fig. 2 Block diagram for the nonlinear system identification with ESN. The ESN model is trained using data from an identification test

The procedure to obtain an identified system, based on the ESN model is summarized as:

1. *Data series acquisition* the input and output data series are obtained by applying an excitation signal on the process. For linear systems, a pseudo-random binary signal (PRBS) is a usual choice and its frequency is chosen based on the process dynamics. For nonlinear systems, in addition to the frequency, the signal amplitude must be carefully chosen to reach all desired operating conditions. To accomplish this, an alternative is the use of an amplitude-modulated PRBS signal (APRBS), where each step has a different amplitude (Nelles 2001).
2. *Data series division* the data series acquired during the identification test should be divided into three portions, used to train, develop, and test the model. The first portion is used to train the ESN models. The development set is used to verify the performance of the trained model, select model architectures, and tune hyperparameters during the model selection phase. The test set is used to evaluate the final model performance, in order to compare it with other models. Each set is formed by a contiguous series of the identification test data.
3. *Model creation* the procedure to create the ESN model is described in detail in Sect. 2.1. Matrices $\mathbf{W}_{\text{res}}^{\text{res}}$, $\mathbf{W}_{\text{in}}^{\text{res}}$, and $\mathbf{W}_{\text{bias}}^{\text{res}}$ are generated using a uniform distribution, with the matrix $\mathbf{W}_{\text{in}}^{\text{res}}$ being scaled using information about the input range, so that the expected reservoir inputs are roughly mapped into a $[-1, +1]$ range. Matrix $\mathbf{W}_{\text{res}}^{\text{res}}$ is created considering all of its elements with positive values, is scaled considering the desired spectral radius, and then some connections are transformed into negative values, to achieve a stable reservoir with diverse dynamics. This procedure is sufficient to achieve the ESP when the selected spectral radius is lower than 1, as discussed in Sect. 2.1.
4. *Model training* the ESN training procedure is performed using the ridge regression method, described in Sect. 2.2. Matrix \mathbf{W}^{out} is obtained by solving (7).
5. *Hyperparameters tuning* the main ESN hyperparameters (reservoir size, connection rate, leak rate, and reservoir spectral radius) are tuned using a grid search procedure. The hyperparameters are initialized with feasible values, and through successive model creation, training, and evaluation using the development set, the influence of each of these parameters is assessed. Details about this step are presented in the case studies, in Sect. 3.
6. *Model evaluation* the final model performance is evaluated using the test data series, by feeding the model with the test data series inputs and comparing the model output with the true output of the system in the test data series.

3 Case Studies: Model Development and Tuning

In this section, the proposed nonlinear system identification procedure based on ESN models, presented in Section 2, is demonstrated. Two nonlinear system identification problems are presented, for which the data series acquisition, model creation, training, hyperparameter tuning, and performance evaluations are detailed. The first case study, which is presented in Sect. 3.1, details the identification of a simulated pH neutralization process. The second one, presented in Sect. 3.2, consists of the identification of a real MIMO test rig used to test refrigeration compressors. The first three steps of the model development procedure, presented in Sect. 2.3, are detailed in the individual sections of each case study, while the last step, the model evaluation, is presented in Sect. 4.

3.1 pH Neutralization Process

The pH neutralization is a process used in many literature sources to benchmark nonlinear system identification and nonlinear control algorithms. This process is widely used in the literature because of its relevance in the chemical industry and its challenging nonlinear dynamics. This case study considers the formulation presented in Gomez et al. (2004), which is a simplification of the one detailed in Henson and Seborg (1994).

The neutralization reactor process consists of the mixing, in a tank with constant volume V , of a NaOH base stream q_1 , a NaHCO₃ buffer stream q_2 , and an HNO₃ acid stream q_3 , as shown in the process diagram presented in Fig. 3. The main challenge in this process consists of the identification of the process operating with only strong acids and strong bases, case in which the process operates in a highly nonlinear operating region, near the pH neutral zone.

The process output, y , is the pH of the effluent solution q_4 , being manipulated by the base flow rate q_1 and the buffer flow rate q_2 , which is considered as an unmeasured disturbance. In this formulation, q_3 stream is assumed to be constant. All the flow variables are expressed in milliliter per second.

The first principles modeling of the pH neutralization process is presented in detail in Henson and Seborg (1994). The

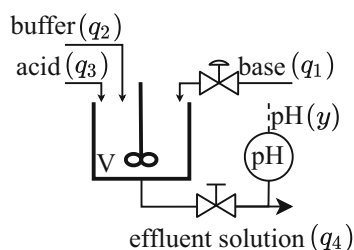


Fig. 3 P&ID diagram of the pH neutralization process

dynamic model is obtained using conservation equations and equilibrium relations, considering perfect mixing, constant density, and complete solubility of the ions. The chemical reaction is defined as



with the equilibrium constants corresponding to

$$K_{a1} = \frac{[\text{HCO}_3^-][\text{H}^+]}{[\text{H}_2\text{CO}_3]}, \tag{11}$$

$$K_{a2} = \frac{[\text{CO}_3^{2-}][\text{H}^+]}{[\text{HCO}_3^-]}, \tag{12}$$

$$K_w = [\text{H}^+][\text{OH}^-]. \tag{13}$$

By defining two reaction invariants, W_a , which is a charge related quantity, and W_b , that represents the CO₃²⁻ ion concentration, the chemical equilibria for each stream $i \in [1, 4]$ are

$$W_{a_i} = [\text{H}^+]_i - [\text{OH}^-]_i - [\text{HCO}_3^-]_i - 2[\text{CO}_3^{2-}]_i, \tag{14}$$

$$W_{b_i} = [\text{H}_2\text{CO}_3]_i + [\text{HCO}_3^-]_i + [\text{CO}_3^{2-}]_i. \tag{15}$$

From the quantities W_a and W_b , the pH can be obtained as

$$W_b \frac{\frac{K_{a1}}{[\text{H}^+]} + \frac{2K_{a1}K_{a2}}{[\text{H}^+]^2}}{1 + \frac{K_{a1}}{[\text{H}^+]} + \frac{K_{a1}K_{a2}}{[\text{H}^+]^2}} + W_a + \frac{K_w}{[\text{H}^+]} - [\text{H}^+] = 0, \tag{16}$$

$$\text{pH} = -\log([\text{H}^+]). \tag{17}$$

Since the tank volume is constant, the mass balance results in

$$q_1 + q_2 + q_3 - q_4 = 0, \tag{18}$$

which can be combined with the mass balance for each ionic species to obtain the differential equations for the reaction invariants W_{a4} and W_{b4} , given by

$$\frac{dW_{a4}(t)}{dt} = \frac{q_1(t)(W_{a1} - W_{a4}(t))}{V} + \frac{q_2(t)(W_{a2} - W_{a4}(t))}{V} + \frac{q_3(t)(W_{a3} - W_{a4}(t))}{V} \tag{19}$$

$$\frac{dW_{b4}(t)}{dt} = \frac{q_1(t)(W_{b1} - W_{b4}(t))}{V} + \frac{q_2(t)(W_{b2} - W_{b4}(t))}{V} + \frac{q_3(t)(W_{b3} - W_{b4}(t))}{V}. \tag{20}$$

Table 1 Nominal operating conditions of the neutralization process phenomenological model

$q_1 = 15.55 \text{ mL s}^{-1}$	$W_a = -4.32 \times 10^{-4} \text{ mol L}^{-1}$
$q_2 = 0.55 \text{ mL s}^{-1}$	$W_b = 5.28 \times 10^{-4} \text{ mol L}^{-1}$
$q_3 = 16.60 \text{ mL s}^{-1}$	$W_{a_1} = -3.05 \times 10^{-3} \text{ mol L}^{-1}$
$K_1 = 6.35$	$W_{a_3} = 3 \times 10^{-3} \text{ mol L}^{-1}$
$K_2 = 10.25$	$W_{b_1} = 5 \times 10^{-5} \text{ mol L}^{-1}$
$V = 2900 \text{ mL}$	$W_{b_2} = 3 \times 10^{-2} \text{ mol L}^{-1}$
	$W_{b_3} = 0$

The process dynamics, in a state space formulation, is defined as

$$\dot{\mathbf{x}} = \mathbf{r}(\mathbf{x}) + \mathbf{g}(\mathbf{x})q_1 + \mathbf{p}(\mathbf{x})q_2, \quad (21)$$

$$h(\mathbf{x}, y) = 0, \quad (22)$$

where the process states are

$$\mathbf{x} = [x_1 \ x_2]^T = [W_{a_4} \ W_{b_4}]^T, \quad (23)$$

and

$$\mathbf{r}(\mathbf{x}) = \left[\frac{q_3(t)(W_{a_3} - x_1)}{V} \quad \frac{q_3(t)(W_{b_3} - x_2)}{V} \right]^T, \quad (24)$$

$$\mathbf{g}(\mathbf{x}) = \left[\frac{(W_{a_1} - x_1)}{V} \quad \frac{(W_{b_1} - x_2)}{V} \right]^T,$$

$$\mathbf{p}(\mathbf{x}) = \left[\frac{(W_{a_2} - x_1)}{V} \quad \frac{(W_{b_2} - x_2)}{V} \right]^T, \quad (25)$$

$$h(\mathbf{x}, y) = x_1 + 10^{y(t)-14} - 10^{-y(t)} + x_2 \frac{1 + 2 \times 10^{y(t)-K_2}}{1 + 10^{K_1-y(t)} + 10^{y(t)-K_2}}, \quad (26)$$

with K_1 and K_2 representing the first and second disassociation constants of the weak acid H_2CO_3 , respectively.

The nominal operating values for this process are presented in Table 1.

3.1.1 Data Series Acquisition

The phenomenological model of the pH neutralization process was used as the real process. The numerical integration Runge–Kutta 45 was used to solve the ordinary differential equations, presented in (20). To represent measurement noise, white noise with variance 0.05 was considered in the pH value, which is obtained by solving the process output Eq. (26) in the simulation.

An open-loop simulation was performed to obtain the data series for the system identification procedure. The sampling time selected for the model is 10 s. For the process excitation, an APRBS was designed to take the system to operate near

the main desired operating conditions, represented by pH values between 5 and 10.

3.1.2 Model Creation and Hyperparameter Tuning

The selection of the best hyperparameters was performed through a grid search, where a new model was instanced using the selected hyperparameter set, was trained by solving Eq. (7), and had its performance accessed using the test set. This search considered as performance metric the mean absolute percentage error (MAPE) and evaluated the four main hyperparameters within the applicable range. The hyperparameters selected for the grid search were the number of reservoir units, the spectral radius, the leak rate, and the regularization parameter. The number of reservoir units is usually in the order of hundreds or thousands, and the maximum value tested in this work was 5000 units. To obtain reservoirs with the ESP, the spectral radius should be selected as discussed in Sect. 2.1. Since the leak rate is proposed to be tested in the range, we propose that the spectral radius is kept at values less than 1, which results in reservoirs with the ESP regardless of the leak rate value, when the reservoir matrix is created using the procedure proposed in Yildiz et al. (2012). The leak rate should be tested within the (0, 1] range. For the regularization parameter choice, values in the $[10^{-4}, 10^1]$ range were considered. The reservoir connection rate was selected as $c_{\text{res}}^{\text{res}} = 0.001$, to create a sparse reservoir. The choice for the reservoir neurons activation function, $f(\cdot)$, was the hyperbolic tangent function, given by

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (27)$$

The grid search considered 5 models being trained for each configuration. The tuning that presented the lowest MAPE in the grid search has a 4000 unit reservoir, a leak rate of 0.3, and a spectral radius $\rho(\mathbf{W}_{\text{res}}^{\text{res}}) = 0.99$. The best training regularization parameter was $\beta = 10^{-4}$. A graphical representation of the grid search experiments is presented in Fig. 4, which presents the mean values and the upper and lower limits defined by two standard deviations around the mean of the MAPE value for the validation set. Each representation considers the impact of a specific hyperparameter, while the others are kept at their best tuning values. The four main hyperparameters are considered in this analysis. The selected values for the final model are presented as an asterisk in Fig. 4.

The hyperparameters whose tuning presented higher impact in the model performance are the spectral radius and the leak rate, both associated with the reservoir creation. Similar behavior is reported in Lukoševičius (2012). For these parameters, a fine-tuning can result in model performance improvements. The ridge regression regularization

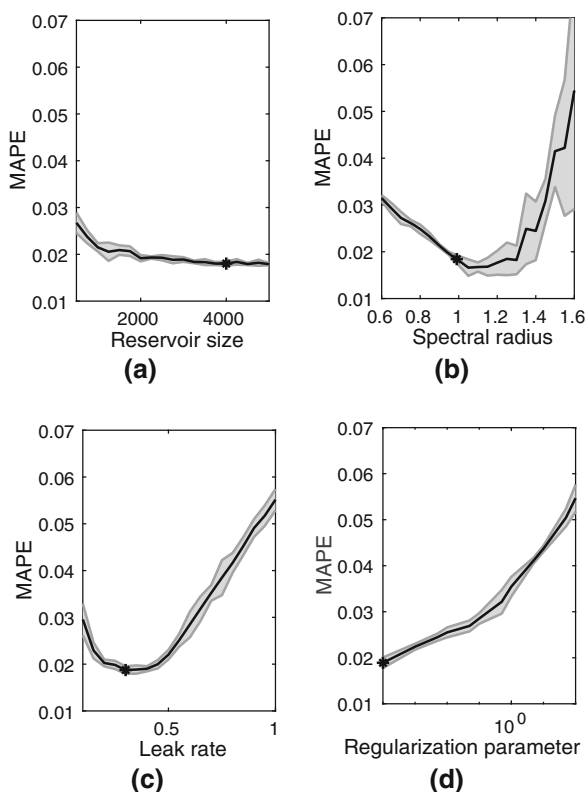


Fig. 4 Hyperparameter impact on the ESN model performance for the pH neutralization process. Evaluated hyperparameters were: **a** reservoir size, **b** spectral radius, **c** leak rate, and **d** regularization. The black lines represent the mean MAPE value, the gray regions represent two standard deviations, and the asterisks represent the final selected values

parameter, which tunes the model bias/variance trade-off by penalizing high readout weights, and the reservoir size also impact the model performance, but not as much as the other two hyperparameters. In this specific case study, tuning the ridge regression parameter with low values presented the best results. In addition, larger reservoir sizes result in better prediction performance, but the performance gain is small for reservoirs larger than 3000.

This model creation considered a conservative tuning, with $\rho(\mathbf{W}_{res}^{res})$ being tuned to guarantee the echo-state property, because $\rho(\mathbf{W}_{res}^{res}) < 1$ is a necessary condition for the ESP. If a less conservative tuning is considered, with values larger than one, the model performance may improve; however, it is possible that this tuning does not have the ESP, which can pose a problem, as the model stability cannot be guaranteed.

3.2 Refrigeration Compressor Test Rig

The refrigeration compressor test rig considered in this section is used in industry to perform tests in refrigeration compressor samples, by subjecting the sample to different operating conditions, thus emulating its use in several types

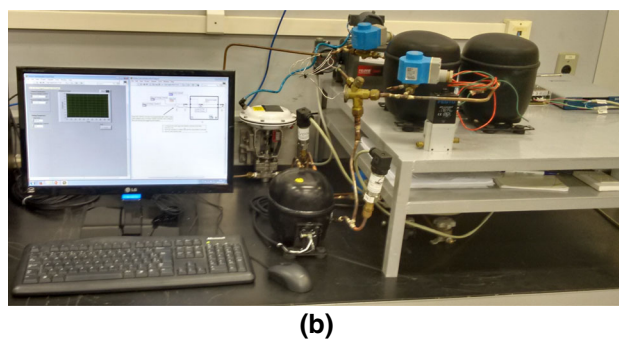
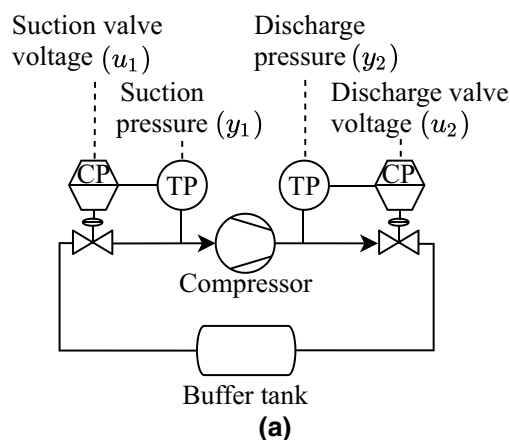


Fig. 5 Refrigeration compressor test rig: **a** simplified piping and instrumentation diagram and **b** experimental process picture

of refrigeration systems. The test rig has valves connected with the compressor suction and discharge lines, in order to manipulate the pressures associated with these lines. These valves are connected to a buffer tank, which decouples partially the suction and discharge pressures. A simplified piping and instrumentation diagram of the process and a picture of the experimental test rig are presented in Fig. 5.

The valve associated with the suction line is normally closed, which makes the suction pressure directly proportional to the voltage applied. The valve connected with the discharge line is normally open, making the discharge pressure also directly proportional to the voltage applied. Due to the process nature, there is a coupling between the suction and discharge pressures, so changes in the suction valve position affect not only the suction pressure, but also the discharge pressure. However, the coupling between the discharge valve and the suction pressure is negligible, being almost completely compensated by the buffer tank.

3.2.1 Data Series Acquisition

For this case study, the data series was obtained from a real test rig. In this rig, the valves are manipulated in order to control the pressures, which are acquired using a sampling

time of 0.1 s. Based on previous knowledge of the process typical operating conditions and the time duration of the process dynamics, three APRBS signals were generated to reach all the desired process operating conditions. The valves were manipulated once at a time, so at each change in a valve the other one was kept at a fixed position (with different but constant values at each change). These identification tests were divided into three datasets, to train, develop, and test the models.

3.2.2 Model Creation and Hyperparameter Tuning

Similarly to the procedure presented for the pH neutralization process, to create the ESN model for the refrigeration compressor test rig, an initial model structure was selected as the starting point for fine-tuning the hyperparameters, which was performed in a grid search. The same activation function used in the pH neutralization process and presented in (27) was used in this case study. The ESN model was built with a MIMO structure, due to the nature of the process, considering the voltages of the two valves as inputs and returning the pressures in the suction and discharge of the compressor as the model outputs. Since this case study considers a MIMO process which presents output variables with different magnitude levels, it is important to consider an appropriate performance index which makes it possible to evaluate the model outputs with equal importance to perform the hyperparameter tuning. For this purpose, the MAPE for each model output was used.

The ESN model tuning was performed through a grid search of the reservoir size, spectral radius, leak rate, and regularization parameter, considering the same range presented for the pH neutralization process. The tuning considered a performance metrics in which the average MAPE of the two variables of interest is minimal. The optimal model, obtained through the grid search, had a reservoir with 2000 units and a leak rate of 0.7. The spectral radius optimal value was $\rho(W_{\text{res}}^{\text{res}}) = 0.9$ and a value $\beta = 0.01$ was used as the regularization parameter. In this tuning, matrix $W_{\text{res}}^{\text{res}}$ was initialized considering a connection rate $c_{\text{res}}^{\text{res}} = 0.01$. The impact of the hyperparameter tuning in the model performance is summarized in Fig. 6. In this case study, for the sake of brevity, only the two most relevant hyperparameters are presented: spectral radius and leak rate.

The model presented the best results for spectral radius values in the [0.8, 1.0] range, with both outputs presenting the best performance for values closer to 0.9. For the leak rate there is a different error behavior for each model output: y_2 shows smaller errors for $\alpha = 0.9$, while y_1 has a lower MAPE for $\alpha = 0.7$. When considering both outputs, selecting $\alpha = 0.7$ resulted in the best performance compromise.

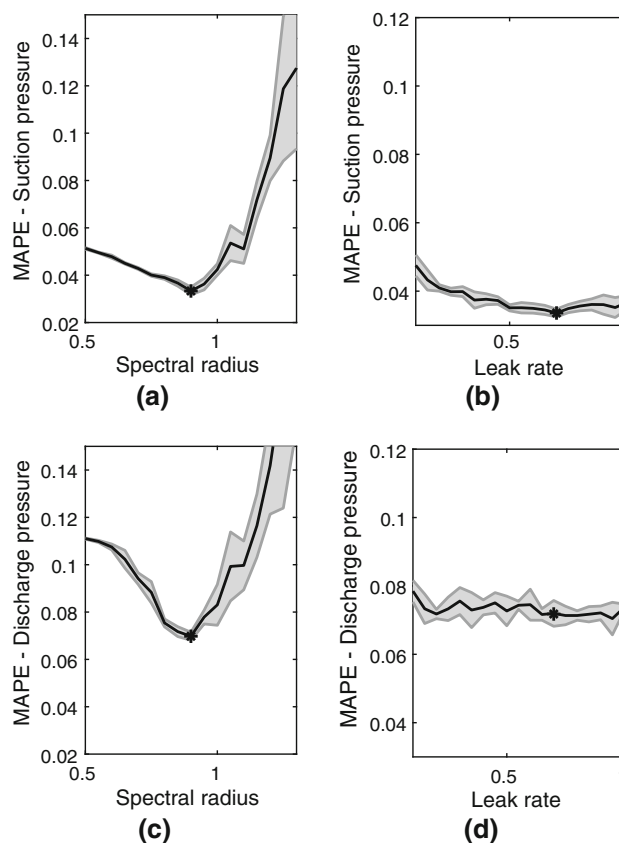


Fig. 6 Hyperparameter impact on the ESN model performance for the refrigeration compressor test rig. Evaluated hyperparameters were: **a** spectral radius and **b** leak rate for the suction pressure, and **c** spectral radius and **d** leak rate for the discharge pressure. The black lines represent the mean MAPE value, the gray regions represent two standard deviations, and the asterisks represent the final selected values

4 Case Studies: Results and Discussion

The final selected models were evaluated using the test portions of the data series. This portion is contiguous and represents a complete sequence of steps in the input variables. A graphical representation of the results is presented in Fig. 7, with the results for the pH neutralization process presented in Fig. 7a, and the results for the refrigeration compressor test rig shown in Fig. 7b. For both cases, the ESN model outputs are compared with the process outputs.

The performance indexes of the test results are summarized in a table for each case study. For this analysis, the mean squared error (MSE), the R^2 correlation criterion, and the mean absolute percentage error (MAPE) are used as performance indexes. In Table 2, the results for the pH neutralization process are presented, while in Table 3, the results for both outputs of the refrigeration compressor case study are detailed. The results of the proposed ESN approach are compared with linear and nonlinear baseline models. As linear models, first- and second-order structures

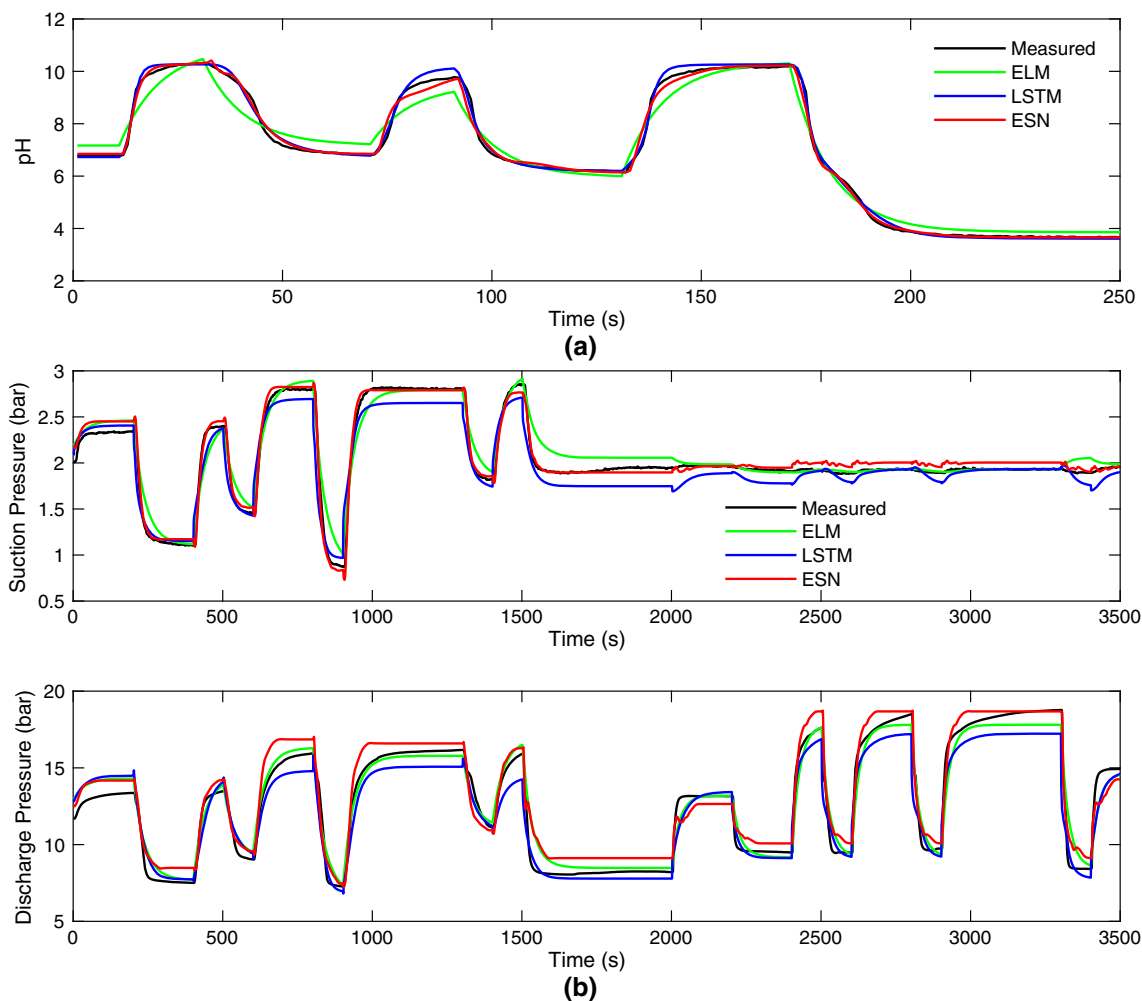


Fig. 7 Comparison between the process output (black line) and the ESN model output (red line) for: **a** pH neutralization process and **b** refrigeration compressor test rig (Color figure online)

were considered. As nonlinear baseline model, a structure based on an extreme learning machine and a Hammerstein nonlinear model (ELM–Hammerstein) was used (Tang et al. 2014). This model consists of a Hammerstein model with an ELM neural network as its nonlinear static part. This baseline model was selected because the ELM neural network presents a learning paradigm similar to the one of ESN, where a neural network is generated using a random distribution and only a portion of the full model is trained, which results in a less computationally expensive model training procedure. The ELM model was built with the same number of neurons used for the final ESN model considered in each case study. The linear portion of the ELM–Hammerstein structure was implemented as a first-order model in both case studies. The last baseline is a nonlinear model which considers a recurrent neural network based on the long short-term memory (LSTM), as detailed in Schwedersky et al. (2019). This model was built using a single fully connected hidden layer, with 10 LSTM units.

Table 2 Comparison between the model performance metrics for the pH neutralization process

Model	MSE	R^2	MAPE
Linear (first order)	0.37	0.93	0.066
Linear (second order)	0.33	0.94	0.060
ELM–Hammerstein	0.22	0.96	0.059
LSTM network	0.04	0.99	0.018
ESN	0.04	0.99	0.018

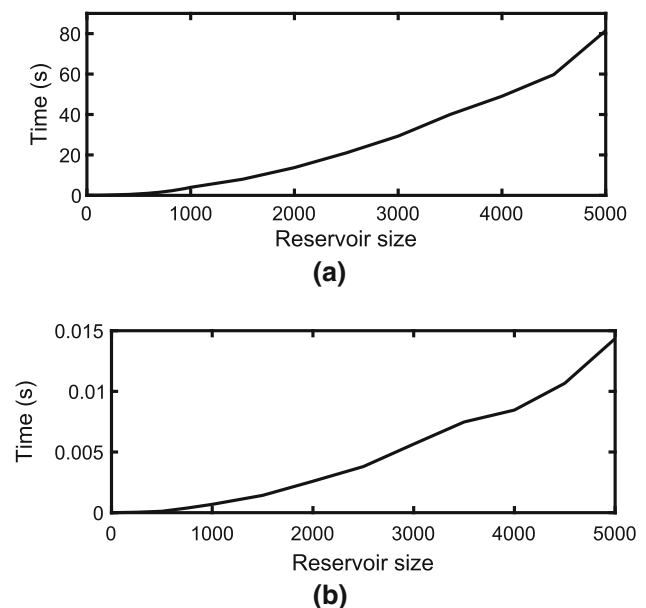
The model obtained with the proposed ESN approach presented good results for both case studies. For the pH neutralization process, the ESN model presented a high fidelity, which can be verified through inspection of 7a. This fact is also supported by the performance indexes presented in 2, which shows that for this case study both the LSTM and the proposed model are the choices. Similarly, the ESN model presented the best results for suction pressure in the refrigerer-

Table 3 Comparison between the model performance metrics for the refrigeration compressor test rig

Model	Suction pressure			Discharge pressure		
	MSE (bar ²)	R ²	MAPE	MSE (bar ²)	R ²	MAPE
Linear (first order)	0.040	0.79	0.094	6.50	0.47	0.213
Linear (second order)	0.041	0.78	0.097	6.31	0.49	0.210
ELM–Hammerstein	0.015	0.92	0.051	0.76	0.94	0.059
LSTM network	0.023	0.87	0.056	0.95	0.92	0.063
ESN	0.008	0.96	0.036	0.85	0.93	0.073

ation compressor test rig case study, and similar performance to the other nonlinear baseline models for discharge pressure. Even though the ELM–Hammerstein model provided the best results for discharge pressure, the differences obtained for this variable with the three nonlinear model alternatives are quite small. Even though a combination of an ESN and an ELM–Hammerstein model is the best choice for this process, it is usually not worth the implementation effort required to obtain this model. If a single model is to be chosen, the natural choice would be the ESN model, as it presented better results for suction pressure and almost the same result as the best model for discharge pressure. For this process, it is important to note that some errors related to the static characteristics of the process appear, especially in the discharge pressure. These errors are present in all models and are related to unmeasured disturbances, mainly the compressor and ambient temperatures, which can change between tests and affect the compressor operating condition. Temperature changes affect both the density of the refrigerant fluid and the solubility of the refrigerant in the oil used for lubricating the compressor moving parts (Björk and Palm 2006). As a consequence, temperature changes affect the mass flow rate and the charge of refrigerant fluid, which reflect as changes in the operating pressures. The performance indexes for this case study, summarized in Table 3, show that, even for a real process subject to unmeasured disturbances, the ESN model presented better results than the baseline methods, except for an equivalent result to the one of the ELM–Hammerstein for the discharge pressure.

Even though both case studies consider ESN models with large reservoir sizes, the computational cost required to train the proposed models is still reasonable for practical applications. The time required for training the model for the pH neutralization process considering a data series of 5000 samples, a 3.6 GHz AMD Ryzen 2400G processor, and a MATLAB implementation is presented in Fig. 8a for different reservoir sizes. The time required to train a model with reservoir sizes of 500 units was less than one second, while the time required for a reservoir with 5000 units was about 80 s. Since the training process is usually performed offline, using larger reservoir sizes is not an obstacle to the implementation of the proposed method. In addition, the computational

**Fig. 8** Comparison between the total computation time required to: **a** train the ESN model and **b** perform a one step-ahead prediction considering different reservoir sizes

cost to perform predictions using the model is also reasonable, as shown in Fig. 8b. For reservoir sizes smaller than 1000 units, the required time is in the order of a few milliseconds, while a reservoir with 5000 units requires computing times smaller than 15 ms, which is a reasonable value for a wide range of applications. The other case study presented equivalent results, but they are not presented in this paper for the sake of brevity.

5 Conclusions

In this paper, an approach to multiple-input and multiple-output nonlinear system identification based on the echo-state network was presented. This approach formalizes some of the best practices in the literature to create an ESN, and proposes a procedure to tune its main hyperparameters. The application of the proposed method was demonstrated in two case studies, a simulated pH neutralization process, as pre-

sented in Henson and Seborg (1994), and an experimental case which considers a real refrigeration compressor test rig used in industry.

Based on the results of both case studies, it was possible to show that the ESN model creation and its tuning can have a strong impact on the final model performance. Besides the use of good practices to create the ESN reservoir, the process to acquire a suitable dataset for system identification and the choice of the ESN hyperparameters deserve special attention, since input data which contain strong information about the system dynamics and careful tuning are necessary for obtaining a good model. For both case studies, the spectral radius and the leak rate were the hyperparameters that had the largest impact on the final model performance. For both case studies, the spectral radius and the leak rate were the hyperparameters whose tuning presented the greatest impact in the final model performance.

For all the case studies, the proposed models based on the ESN approach obtained the best results among all the evaluated methods (two linear models, an ELM–Hammerstein model, and an LSTM network model). These results indicate that the use of the proposed approach, based on the ESN, is an alternative to classical techniques for nonlinear system identification. It enables the identification of MIMO nonlinear systems in a data-driven approach, without much need for defining model structures and orders based on knowledge about the process dynamics and its nonlinearities, which is the case of classical approaches, such as the linear and Hammerstein models.

Acknowledgements An early version of paper was presented at XXIII Congresso Brasileiro de Automática (CBA 2020). This study was funded in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grants 140283/2018-8 and 309244/2018-8, and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)-Finance Code 001.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Antonelo, E. A., Camponogara, E., & Foss, B. (2017). Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells. *Neural Networks*, 85, 106–117.
- Ayoubi, M. (1994). Nonlinear dynamic systems identification with dynamic neural networks for fault diagnosis in technical processes. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 3, 2120–2125.
- Bessa, R., & Barreto, G. A. (2019). Robust echo state network for recursive system identification. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in computational intelligence* (pp. 247–258). Springer.

- Björk, E., & Palm, B. (2006). Performance of a domestic refrigerator under influence of varied expansion device capacity, refrigerant charge and ambient temperature. *International Journal of Refrigeration*, 29(5), 789–798. <https://doi.org/10.1016/j.ijrefrig.2005.11.008>
- Boussaada, Z., Curea, O., Remaci, A., Camblong, H., & Mrabet Bellaaj, N. (2018). A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies*, 11(3), 620.
- Chen, Y., He, Z., Shang, Z., Li, C., Li, L., & Xu, M. (2019). A novel combined model based on echo state network for multi-step ahead wind speed forecasting: A case study of NREL. *Energy conversion and management*, 179, 13–29.
- Fernandez, B., Parlos, A., & Tsai, W. (1990). Nonlinear dynamic system identification using artificial neural networks (ANNs). In 1990 *IJCNN international joint conference on neural networks*, (IEEE).
- Ganguli, S., Huh, D., & Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48), 18970–18975. <https://doi.org/10.1073/pnas.0804451105>
- Gomez, J., Jutan, A., & Baeyens, E. (2004). Wiener model identification and predictive control of a pH neutralisation process. *IEEE Proceedings-Control Theory and Applications*, 151(3), 329–338.
- Guo, Y., Wang, F., Chen, B., & Xin, J. (2017). Robust echo state networks based on correntropy induced loss function. *Neurocomputing*, 267, 295–303. <https://doi.org/10.1016/j.neucom.2017.05.087>
- Han, M., & Xu, M. (2018). Laplacian echo state network for multivariate time series prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1), 238–244. <https://doi.org/10.1109/tnnls.2016.2574963>
- Henson, M. A., & Seborg, D. E. (1994). Adaptive nonlinear control of a pH neutralization process. *IEEE Transactions on Control Systems Technology*, 2(3), 169–182.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2), 107–116.
- Inubushi, M., & Yoshimura, K. (2017). Reservoir computing beyond memory-nonlinearity trade-off. *Scientific Reports*. <https://doi.org/10.1038/s41598-017-10257-6>
- Isermann, R., & Münchhof, M. (2011). *Identification of dynamic systems*. Springer.
- Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks. Tech. Rep. 34, German National Research Center for Information Technology, Bonn, Germany.
- Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. *Advances in Neural Information Processing Systems (NIPS 2003)* (pp. 609–616).
- Jaeger, H., Lukoševičius, M., Popovici, D., & Siewert, U. (2007). Optimization and applications of echo state networks with leaky: Integrator neurons. *Neural Networks*, 20(3), 335–352. <https://doi.org/10.1016/j.neunet.2007.04.016>
- Li, D., Han, M., & Wang, J. (2012). Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5), 787–799. <https://doi.org/10.1109/tnnls.2012.2188414>
- Ljung, L., Andersson, C., Tiels, K., & Schön, T. B. (2020). Deep learning and system identification. In Proc IFAC Congress.
- Lukoševičius, M. (2012). A practical guide to applying echo state networks. In G. Montavon, G. B. Orr, & K. R. Müller (Eds.), *Neural networks tricks of the trade* (pp. 659–686). Springer.
- Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127–149.

- Lun, S. X., Yao, X. S., Qi, H. Y., & Hu, H. F. (2015). A novel model of leaky integrator echo state network for time-series prediction. *Neurocomputing*, *159*, 58–66. <https://doi.org/10.1016/j.neucom.2015.02.029>
- Nelles, O. (2001). *Nonlinear system identification*. Springer.
- Qiao, J., Wang, L., & Yang, C. (2018). Adaptive lasso echo state network based on modified Bayesian information criterion for nonlinear system modeling. *Neural Computing and Applications* (pp. 1–15).
- Rodan, A., & Tino, P. (2010). Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, *22*(1), 131–144.
- Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, *39*(6), 28–99. <https://doi.org/10.1109/MCS.2019.2938121>
- Schwedersky, B. B., Flesch, R. C. C., & Dangui, H. A. C. (2018). Practical nonlinear model predictive control using an echo state network model. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). <https://doi.org/10.1109/IJCNN.2018.8489446>
- Schwedersky, B. B., Flesch, R. C. C., & Dangui, H. A. (2019). Practical nonlinear model predictive control algorithm for long short-term memory networks. *IFAC-PapersOnLine*, *52*(1), 468–473. <https://doi.org/10.1016/j.ifacol.2019.06.106>
- Shi, Z., & Han, M. (2007). Support vector echo-state machine for chaotic time-series prediction. *IEEE Transactions on Neural Networks*, *18*(2), 359–372. <https://doi.org/10.1109/tnn.2006.885113>
- Tang, Y., Li, Z., & Guan, X. (2014). Identification of nonlinear system using extreme learning machine based Hammerstein model. *Communications in Nonlinear Science and Numerical Simulation*, *19*(9), 3171–3183.
- Verzelli, P., Alippi, C., & Livi, L. (2019). Echo state networks with self-normalizing activations on the hyper-sphere. *Scientific Reports*. <https://doi.org/10.1038/s41598-019-50158-4>
- Wainrib, G., & Galtier, M. N. (2016). A local echo state property through the largest Lyapunov exponent. *Neural Networks*, *76*, 39–45. <https://doi.org/10.1016/j.neunet.2015.12.013>
- Yang, C., Qiao, J., Ahmad, Z., Nie, K., & Wang, L. (2019). Online sequential echo state network with sparse RLS algorithm for time series prediction. *Neural Networks*, *118*, 32–42.
- Yildiz, I. B., Jaeger, H., & Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, *35*, 1–9.
- Yu, W. (2004). Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. *Information Sciences*, *158*, 131–147. <https://doi.org/10.1016/j.ins.2003.08.002>
- Zhou, H., Huang, J., Lu, F., Thiyagalingam, J., & Kirubarajan, T. (2018). Echo state kernel recursive least squares algorithm for machine condition prediction. *Mechanical Systems and Signal Processing*, *111*, 68–86. <https://doi.org/10.1016/j.ymssp.2018.03.047>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.