An Echo State Network-based Soft Sensor of Downhole Pressure for a Gas-lift Oil Well

Eric Aislan Antonelo and Eduardo Camponogara

Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, Brazil,
erantone@elis.ugent.be

Abstract. [1] Soft sensor technology has been increasingly used in industry. Its importance is magnified when the process variable to be estimated is key to control and monitoring processes and the respective sensor either has a high probability of failure or is unreliable due to harsh environment conditions. This is the case for permanent downhole gauge (PDG) sensors in the oil and gas industry, which measure pressure and temperature in deepwater oil wells. In this paper, historical data obtained from an actual offshore oil well is used to build a black box model that estimates the PDG downhole pressure from platform variables, using Echo State Networks (ESNs), which are a class of recurrent networks with powerful modeling capabilities. These networks, differently from other neural networks models used by most soft sensors in literature, can model the nonlinear dynamical properties present in the noisy real-world data by using a two-layer structure with efficient training: a recurrent nonlinear layer with fixed randomly generated weights and a linear adaptive readout output layer. Experimental results show that ESNs are a promising technique to model soft sensors in an industrial setting.

Keywords: echo state network, soft sensor, gas-lift oil well, reservoir computing

1 Introduction

With the advancement of powerful machine learning methods in the last decades, soft sensor technology has been made possible for a broad range of industries. Soft sensors aim to provide an additional source of information for process variables which are not reliable enough or whose expensive sensors can fail permanently, for instance, in harzadous environments. These soft sensors are predictive models built with methods which can infer an output $y(t)$ based on a number of input measurements $\mathbf{u}(t)$ [1]. The most common way of building soft sensors is through system identification using historical time series which show the (likely nonlinear) relationship between process variables. The resulting soft sensors are called data-driven since they are empirically obtained. Grey-box models

[1] This paper is a draft version of the publication presented at the 16th International Conference on Engineering Applications of Neural Networks.

and black-box models can be used to fit the empirical data. However, black-box models are more often used for soft sensor technology than grey-box models [2], which have in artificial neural networks (ANNs) their most important and frequently used method [3]. This is because ANNs can efficiently model nonlinear relationships in process variables, which is usually the case for real-world processes.

Deepwater or low pressure oil wells usually require gas-lift technology in order to extract the oil from the well. The artificially injected gas diminishes the density of the well fluid, which, in turn, makes possible its extraction with the created difference in pressure. The downhole pressure is essential in assessing the dynamics of the oil well and must be monitored for controlling the productivity and stability of the well through the gas-lift flow rate as well as the production choke. The problem comes from the fact that permanent downhole gauge (PDG) sensors have a prohibitive cost for maintainance or replacement [4], and also that their premature failure is not uncommmon. Additionally, pertubations and noise can affect the PDG sensor measurements, making it not always a reliable information source.

Therefore, ANN-based soft sensors represent an important and alternative way of monitoring these variables. As the objective is to model an unknown dinamical system from real-world data, it is necessary that the used model maintains an internal state as a dynamical system does. This can be directly achieved by using a recurrent neural network (RNN), considered an universal approximation method for dynamical systems. An alternative is to use a tapped delay line with feedforward networks, which provides a finite window of past inputs, but provides no internal state for the network as the RNN does. Most models in literature use this last approach [5] or alternatively NARMAX models [6], since training an RNN with backpropagation-through-time is not trivial due to slow convergence properties and existence of bifurcations during training.

This paper aims to build a soft sensor of downhole pressure for gas-lift oil wells using a particular model of RNNs which exhibit fast training without local optima. Analog recurrent networks of these type are called Echo State Networks (ESNs) [7]. Their distinct feature is to separate the network in two main layers: one randomly generated pool of recurrent nonlinear neurons (called the reservoir [2]), and a linear adaptative readout layer (see Fig. 1). As the recurrent reservoir has its weights randomly generated, only the readout output layer needs to be trained, usually using linear regression methods such as the Least Squares method which has global convergence properties. Thus, the complexity of training recurrent weights is nonexistent. It is also possible to see the reservoir as a nonlinear dynamical kernel which projects the input into a high-dimensional nonlinear space, facilitating learning complex models. Methods which share this type of feature are called in literature Reservoir Computing (RC) methods [8].

In [9], a simulated vertical riser model was identified using ESNs as a black-box model. The ESN, by using feedback from the output to the reservoir layer, was able to sustain oscillations as well as steady states with only a single ESN.

[2] The term reservoir is not related to reservoirs in oil and gas industry.

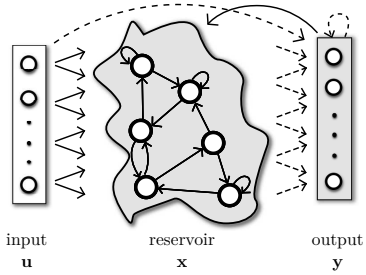


Fig. 1. Reservoir Computing (RC) network. The reservoir is a non-linear dynamical system usually composed of recurrent sigmoid units. Solid lines represent fixed, randomly generated connections, while dashed lines represent trainable or adaptive weights.

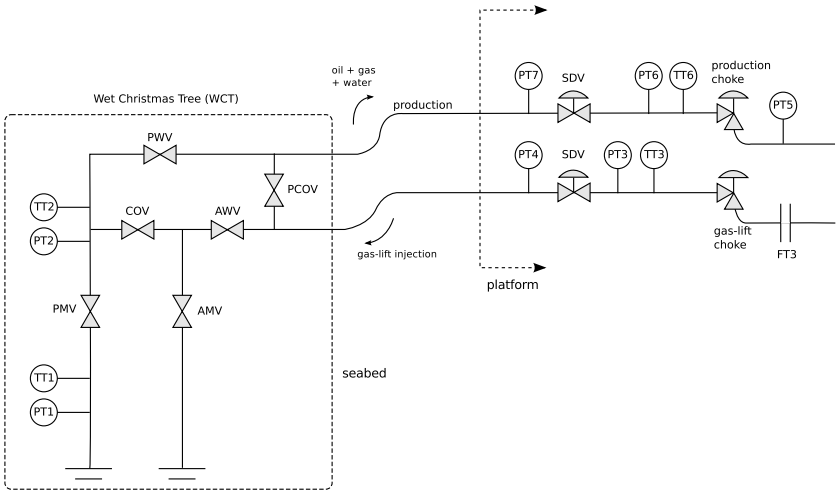


Fig. 2. Oil well scheme showing the location of sensors and chokes. FT3 is a flow rate sensor; PT# and TT# are pressure and temperature sensors. SDV stands for ShutDown Valve.

The input consisted of a single variable, the opening of the production choke, and the output to be estimated was the downhole pressure. In the current paper, instead of using simulation data, the ESN-based model is built from real-world data obtained from a particular deepwater oil well. The PDG downhole pressure is estimated based on input from 8 sensor measurements plus 2 choke openings (gas-lift and production chokes). All these input measurements come from the platform variables since seabed variables are not always available in offshore oil wells (see Fig. 2 for a scheme with position of sensors and chokes). The main contribution of this paper lies in the analysis of the powerful RC approach in modeling real-world noisy data from the gas and oil industry.

In the following section, the ESN model and training is presented. Section 3 presents the procedure to build the ESN-based soft sensor. Next, experimental results are shown in Section 4 and the conclusion drawn in Section 5.

2 Reservoir Computing

2.1 ESN model

An ESN is composed of a discrete hyperbolic-tangent RNN, the reservoir, and of a linear readout output layer which maps the reservoir states to the actual output. Let n_i, n_r and n_o represent the number of input, reservoir and output units, respectively, $\mathbf{u}[n]$ the n_i-dimensional external input, $\mathbf{x}[n]$ the n_r-dimensional reservoir activation state, $\mathbf{y}[n]$ the n_o-dimensional output vector, at discrete time n. Then the discrete time dynamics of the ESN is given by the state update equation

$$\begin{aligned}\mathbf{x}[n+1] =&(1-\alpha)\mathbf{x}[n] + \alpha f(\mathbf{W}^{\mathrm{r}}_{\mathrm{r}}\mathbf{x}[n] + \mathbf{W}^{\mathrm{r}}_{\mathrm{i}}\mathbf{u}[n]+ \\ &\mathbf{W}^{\mathrm{r}}_{\mathrm{o}}\mathbf{y}[n] + \mathbf{W}^{\mathrm{r}}_{\mathrm{b}}),\end{aligned} \tag{1}$$

and by the output computed as:

$$\mathbf{y}[n+1] = g\left(\mathbf{W}^{\mathrm{o}}_{\mathrm{r}}\mathbf{x}[n+1] + \mathbf{W}^{\mathrm{o}}_{\mathrm{i}}\mathbf{u}[n] + \mathbf{W}^{\mathrm{o}}_{\mathrm{o}}\mathbf{y}[n] + \mathbf{W}^{\mathrm{o}}_{\mathrm{b}}\right) \tag{2}$$

$$= g\left(\mathbf{W}^{\mathrm{out}}\left(\mathbf{x}[n+1], \mathbf{u}[n], \mathbf{y}[n], 1\right)\right) \tag{3}$$

$$= g\left(\mathbf{W}^{\mathrm{out}}\mathbf{z}[n+1]\right), \tag{4}$$

where: α is the leak rate $[10, 11]$; $f(\cdot) = \tanh(\cdot)$ is the hyperbolic tangent activation function, commonly used for ESNs; g is a post-processing activation function (in this paper, g is the identity function); $\mathbf{W}^{\mathrm{out}}$ is the column-wise concatenation of $\mathbf{W}^{\mathrm{o}}_{\mathrm{r}}$, $\mathbf{W}^{\mathrm{o}}_{\mathrm{i}}$, $\mathbf{W}^{\mathrm{o}}_{\mathrm{o}}$ and $\mathbf{W}^{\mathrm{o}}_{\mathrm{b}}$; and $\mathbf{z}[n+1] = (\mathbf{x}[n+1], \mathbf{u}[n], \mathbf{y}[n], 1)$ is the extended reservoir state, i.e., the concatenation of the state, the previous input and output vectors and a bias term, respectively.

The matrices $\mathbf{W}^{\mathrm{to}}_{\mathrm{from}}$ represent the connection weights between the nodes of the complete network, where r, i, o, b denotes *reservoir*, *input*, *output*, and *bias*, respectively. All weight matrices representing the connections to the reservoir, denoted as $\mathbf{W}^{\mathrm{r}}_{\cdot}$, are initialized randomly (represented by solid arrows in Figure 1), whereas all connections to the output layer, denoted as $\mathbf{W}^{\mathrm{o}}_{\cdot}$, are trained (represented by dashed arrows in Figure 1). For the experiments in this paper, output feedback and bias to reservoir are not used ($\mathbf{W}^{\mathrm{r}}_{\mathrm{o}}$ and $\mathbf{W}^{\mathrm{r}}_{\mathrm{b}}$ are not present). Additionaly, $\mathbf{W}^{\mathrm{o}}_{\mathrm{o}}$, $\mathbf{W}^{\mathrm{o}}_{\mathrm{b}}$ and $\mathbf{W}^{\mathrm{o}}_{\mathrm{i}}$ are also absent. Thus, equations (1) and (2) become:

$$\mathbf{x}[n+1] = (1-\alpha)\mathbf{x}[n] + \alpha f\left(\mathbf{W}^{\mathrm{r}}_{\mathrm{r}}\mathbf{x}[n] + \mathbf{W}^{\mathrm{r}}_{\mathrm{i}}\mathbf{u}[n]\right) \tag{5}$$

$$\mathbf{y}[n+1] = g\left(\mathbf{W}^{\mathrm{out}}\mathbf{x}[n+1]\right). \tag{6}$$

The non-trainable connection matrices $\mathbf{W}^{\mathrm{r}}_{\mathrm{r}}, \mathbf{W}^{\mathrm{r}}_{\mathrm{i}}$ are usually generated from a Gaussian distribution $N(0, 1)$ or a uniform discrete set $\{-1, 0, 1\}$. During this

random initialization, sparsity can be obtained by using a parameter called connection fraction $c^{\text{to}}_{\text{from}}$ which determines the percentage of nonzero weights in the respective connection matrix $\mathbf{W}^{\text{to}}_{\text{from}}$. Another parameter in this procedure is the input scaling $\upsilon^{\text{to}}_{\text{from}}$ which is a scalar multiplied by the respective matrix $\mathbf{W}^{\text{to}}_{\text{from}}$. This *scaling* of matrices is important because it influences greatly the reservoir dynamics [8] and, in this way, must be tuned for optimal performance in most cases.

The weights from the reservoir connection matrix $\mathbf{W}^{\text{r}}_{\text{r}}$ are obtained randomly through a Normal distribution ($N(0,1)$) and then rescaled such that the resulting system is stable but still exhibits rich dynamics. A general rule to create good reservoirs is to set the reservoir weights such that the reservoir has the *Echo State Property* (ESP) [12], i.e., a reservoir with fading memory. A common method used in literature is to rescale $\mathbf{W}^{\text{r}}_{\text{r}}$ such that its spectral radius $\rho(\mathbf{W}^{\text{r}}_{\text{r}}) < 1$ [12]. Although it does not guarantee the ESP, in practice it has been empirically observed that this criterium works well and often produces analog sigmoid ESNs with ESP for any input.

It is important to note that spectral radius closer to unity as well as larger input scaling makes the reservoir more non-linear, which has a deterioration impact on the memory capacity as side-effect [13]. The scaling of these non-trainable weights is a parameter which should be chosen according to the task at hand empirically, analyzing the behavior of the reservoir state over time, or by grid searching over parameter ranges.

Most temporal learning tasks require that the timescale present in the reservoir match the timescales present in the input signal as well as in the task space. This matching can be done by the use of a leak rate ($\alpha \in (0,1]$) and/or by resampling the input signal. For instance, low leak rates yield reservoirs with more memory which can *hold* the previous stimuli for longer time spans.

2.2 Training

Training the RC network means finding $\mathbf{W}^{\text{out}}$ in (2) or (6), that is, the weights for readout output layer from Fig. 1. For that, the reservoir is driven by an input sequence $\mathbf{u}(1), \ldots, \mathbf{u}(n_s)$ which yields a sequence of extended reservoir states $\mathbf{z}(1), \ldots, \mathbf{z}(n_s)$ using (1) (the initial state is $\mathbf{x}(0) = \mathbf{0}$). The desired target outputs $\hat{\mathbf{y}}[n]$ are collected row-wise into a matrix $\hat{\mathbf{Y}}$. The generated extended states are collected row-wise into a matrix $\mathbf{X}$ of size $n_s \times (n_r + n_i + n_o + 1)$ if using (1) or $n_s \times n_r$ if using (5).

Then, the training of the output layer is done by using the **Ridge Regression** method [14], also called *Regularized Linear Least Squares* or *Tikhonov regularization* [15]:

$$\tilde{\mathbf{W}}^{\text{out}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \hat{\mathbf{Y}} \tag{7}$$

where $\tilde{\mathbf{W}}^{\text{out}}$ is the column-wise concatenation of $\mathbf{W}^{\text{o}}_{\text{r}}$, and the optional matrices $\mathbf{W}^{\text{o}}_{\text{i}}$, $\mathbf{W}^{\text{o}}_{\text{o}}$ and n_s denotes the total number of training samples.

In the generation of $\mathbf{X}$, a process called **warm-up drop** is used to disregard a possible undesired initial transient in the reservoir starting at $\mathbf{x}(0) = \mathbf{0}$. This is achieved by dropping the first n_{wd} samples so that only the samples $\mathbf{z}[n], n = n_{wd}, n_{wd}+1, ..., n_s$ are collected into the matrix $\mathbf{X}$.

The learning of the RC network is a fast process without local minima. Once trained, the resulting RC-based system can be used for real-time operation on moderate hardware since the computations are very fast (only matrix multiplications of small matrices).

Error measure For regression tasks, the Normalized Root Mean Square Error (NRMSE) is used as a performance measure and is defined as:

$$\text{NRMSE} = \sqrt{\frac{\langle(\hat{y}[n] - y[n])^2\rangle}{\sigma^2_{\hat{y}[n]}}}, \tag{8}$$

where the numerator is the mean squared error of the output $y[n]$ and the denominator is the variance of desired output $\hat{y}[n]$.

3 Soft Sensor through ESNs

The task is to infer the downhole pressure at the PDG sensor, located in the seabed, from sensor measurements obtained at the more easily accessible platform location (see Fig. 2). The sets of input and output variables are given in Table 1. In this work, the ESN or RC network is used to learn a dynamical mapping between the input variables $\mathbf{u}(t)$ (which, in principle, consists of 10 inputs normalized to the interval $[0, 1]$, corresponding to the 8 platform variables from Table 1 plus the openings of the gas-lift choke and production choke) and the output variable $y(t)$ (PDG pressure sensor). The available dataset contains 5 months of measurement data: 08/2010, 01/2011, 07/2011, 11/2011 and 12/2011. The measured downhole pressure for the complete period can seen in Fig. 3.

Table 1. Process variables

Tag	Process variable	Location	Variables Set
PT1	Downhole pressure	Seabed	Output
TT1	Downhole temperature	Seabed	—
PT2	WCT pressure	Seabed	—
TT2	WCT temperature	Seabed	—
PT3	Pressure before SDV	Platform	Input
TT3	Temperature before SDV	Platform	Input
FT3	Instantaneous gas-lift flow rate	Platform	Input
PT4	Pressure after SDV	Platform	Input
PT5	Pressure after production choke	Platform	Input
PT6	Pressure before production choke	Platform	Input
TT6	Temperature before production choke	Platform	Input
PT7	Pressure before SDV	Platform	Input

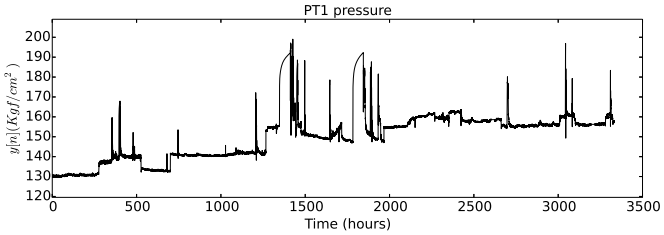


Fig. 3. Downhole Pressure (PT1) measured over the complete period of 5 months.

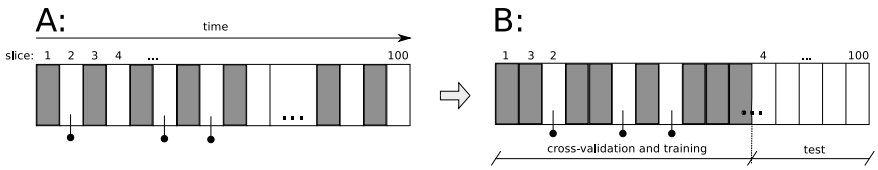


Fig. 4. Scheme showing how the dataset is divided for training and test. Grey-shaded rectangles plus randomly chosen white rectangles (indicated by sticks) are selected for training, while white ones are used for test. See text for more information.

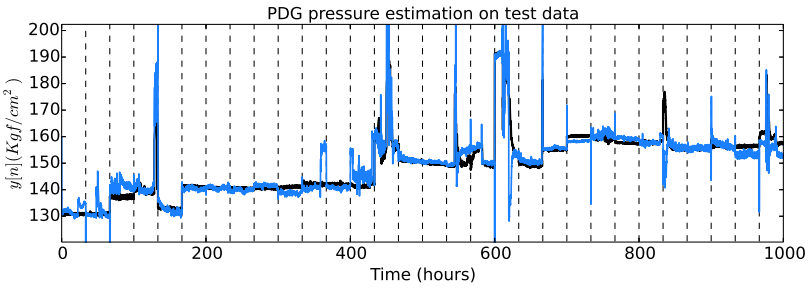
The soft sensor is built according to one of the following approaches: using all the available sampled data (5 months); and using only a limited number of samples such as from only one month. Considering the former case, i.e., when the whole dataset is used, the exhibited dynamical nonlinear behavior range is very wide. This is because the measurements can also include cases when the well is closed, causing abnormal behavior, and also that a prolonged period will include different or evolving relationships between variables for samples very distant from each other in time. Taking this into account, we propose a method to select a representative subset of the dataset for training and the rest for test. The method is graphically represented in Fig. 4 and described in the following. We decide that 70% of the dataset is selected for training, while the rest 30% is used for test. For that, first, the dataset is divided into two groups, gray and white rectangles, which are alternatingly chosen in the time axis, as *individual sample slices* each containing 2,000 measurements (the sampling rate is one measurement per minute). Now, we have two sets, each having 50% of the dataset (situation A in the figure). To form the training dataset, we add more 20% of randomly chosen white rectangles to the the training dataset, indicated in Fig. 4 by little sticks. The resulting operation leads to the re-arranged sample slices used for training and test (Situation B in the plot), where now the temporal order is only valid between the samples contained in the slices.

4 Experimental Results

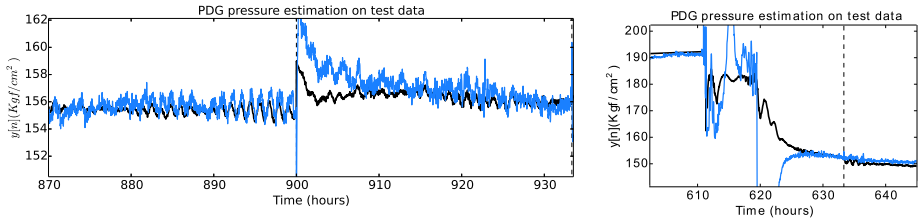
4.1 Whole dataset

Using the method described in the previous section and shown in Fig. 4, 70% of the dataset was selected for training and validation whereas the rest 30% of the data was reserved for testing. With the selected 70%, a grid-search experiment was accomplished to find the best performing parameters: input scaling $\upsilon_{\mathrm{i}}^{\mathrm{r}}$, leak rate α, and spectral radius $\rho(\mathbf{W}_{\mathrm{r}}^{\mathrm{r}})$. Fixed parameters were as follows: size of reservoir $n_r = 200$ and all connection fractions were $c_{\cdot} = 1$. Each run of the grid search was made through a 5-fold random cross-validation procedure, considering 3 randomly generated reservoirs, where the ridge regression parameter λ was optimized for each generated reservoir. The values found were as follows: $\upsilon_{\mathrm{i}}^{\mathrm{r}} = 0.2$, $\alpha = 0.5$, and $\rho(\mathbf{W}_{\mathrm{r}}^{\mathrm{r}}) = 0.5$.

Next, the RC network was trained using above parameters on the whole training dataset, and the generalization results were obtained evaluating the trained network on the test dataset (30% of the samples). Fig. 5 shows these results comparing the predicted network output (given by a light blue line) to the target output (or the measured downhole pressure PT1, given by a black line). Each vertical dashed line marks the frontier between one sample slice and the next one (see Section 3), each one containing 2,000 consecutive samples (measured each minute). Fig. 5 (a) shows the prediction over the complete test set, revealing



(a) Complete test dataset



(b) Zoomed periods with particular dynamical behaviors

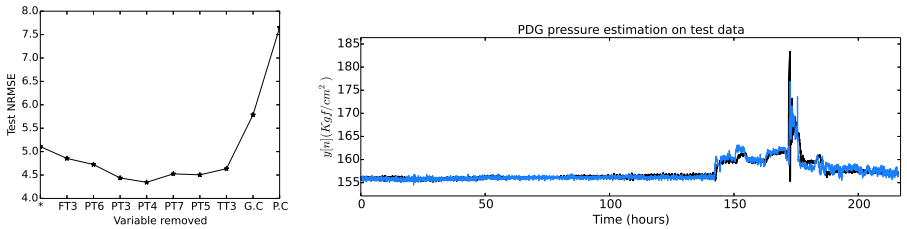
Fig. 5. Results considering 5 months of well sample data. The estimated and real PDG pressures are given by blue and black solid lines, respectively. Each vertical dashed line delimits slices containing 2,000 minutes of samples each.

that most of the operating points of the downhole pressure were correctly estimated. The dynamical properties of this signal varied considerably taking into account all the 5 months. Some nonlinear dynamical behaviors present in the dataset were feasible to be learned while other periods showed inconsistencies (Fig. 5 (b)), probably due to abnormal and less frequent behaviors, which may not be inferred from the input variables. The NRMSE on test data was 5.03 for a reservoir of 200 units and 4.81 for a reservoir of 400 units.

Furthermore, to verify which variables were most important for the estimation task, a procedure called backward variable removal was executed. It consists of starting with all 10 variables as inputs, and removing the one that results in the least error at that iteration. At the next iteration, the same logic is executed. Fig. 6 (a) shows the results of this procedure. The minimal error is reached when variables FT3, PT6, PT3, and PT4 are removed. The most important variables are TT6 (which was not removed during the complete process), P.C (production choke), G.C (gas-lift choke).

4.2 One month dataset

Considering a smaller section of the dataset, corresponding to the samples obtained on December 2011, an RC network of 200 reservoir units was trained with the same parameters from the previous section: $v_{\mathrm{i}}^{\mathrm{r}} = 0.2$, $\alpha = 0.5$, and $\rho(\mathbf{W}_{\mathrm{r}}^{\mathrm{r}}) = 0.5$. The regularization parameter λ was chosen using a randomly selected validation set. In Fig. 6 (b), the predicted downhole pressure for the last 30% samples of the month can be seen using an RC network trained on the first 70% samples. The blue line shows that the estimation is considerably good when compared to the target PT1 downhole pressure. The test NRMSE is 5.84 (note that the NRMSE has a denominator equal to the variance of the target signal, which means that a smaller variance will increase the NRMSE, which is the case here compared to previous section's results).



(a) (b)

Fig. 6. (a) Backward variable removal results showing the test NRMSE as variables are gradually removed. (b) Results considering the last month (December 2011) for both training and test. The lines represent the same concepts as in Fig. 5.

5 Conclusion

In this work, ESNs (RC networks) have been used to construct a black-box model of a soft sensor in deepwater oil wells. The task was to infer the PDG downhole pressure in an actual oil well based on readings from the platform sensor variables. A single ESN was used for such task, considering a dataset of either 5 months or 1 month of samples. The task reached reasonable results for the first experiment, although using 5 months was more difficult since there were undetected outliers and inconsistencies which were not removed in the current work. The second experiment, modeling the sensor with data from only a month, showed very good results using the same parameters (input scaling, spectral radius, leak rate and reservoir size) for the ESN as in the first experiment. This is because the total data used to train the RC model is very close in the temporal dimension to the test data. This indicates that, if data is always available, the RC model can benefit from a self-update mechanism (e.g. online learning) for achieving superior performance.

Future work will tackle several points: removal of outliers and inconsistent sampling periods for training dataset selection; architectural network changes to accommodate different timescales present in the process variables (i.e., for improving modeling small oscillations in apparently steady states conditions); and building of an ESN-based observation model which can be used to correct the process model output.

References

1. Fortuna, L., Graziani, S., Rizzo, A., Xibilia, M.G.: Soft sensors for monitoring and control of industrial processes. Springer Science & Business Media (2007)
2. Sbarbaro, D., Ascencio, P., Espinoza, P., Mujica, F., Cortes, G.: Adaptive soft-sensors for on-line particle size estimation in wet grinding circuits. Control Engineering Practice **16**(2) (2008) 171–178
3. Fujiwara, K., Kano, M., Hasebe, S.: Development of correlation-based pattern recognition algorithm and adaptive soft-sensor design. Control Engineering Practice **20**(4) (2012) 371–378
4. Eck, J., et al.: Downhole monitoring: the story so far. Oilfield Review **11**(3) (1999) 18–29
5. Teixeira, B.O., Castro, W.S., Teixeira, A.F., Aguirre, L.A.: Data-driven soft sensor of downhole pressure for a gas-lift oil well. Control Eng. Practice **22** (2014) 34–43
6. Billings, S.A.: Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains. Wiley (2013)
7. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. Science **304**(5667) (Apr 2004) 78–80
8. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Networks **20**(3) (2007) 391–403
9. Antonelo, E.A., Camponogara, E., Plucenio, A.: System identification of a vertical riser model with echo state networks. In: 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production. (2015)
10. Jaeger, H., Lukosevicius, M., Popovici, D.: Optimization and applications of echo state networks with leaky integrator neurons. Neur. Netw. **20**(3) (2007) 335–352

11. Schrauwen, B., Defour, J., Verstraeten, D., Van Campenhout, J.: The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In: Proc. of the 17th ICANN. Volume 4668 of LNCS. Springer (2007) 471–479
12. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology (2001)
13. Verstraeten, D., Dambre, J., Dutoit, X., Schrauwen, B.: Memory versus non-linearity in reservoirs. In: Proc. of the IEEE IJCNN. (Jul. 2010) 1–8
14. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (August 2006)
15. Tychonoff, A., Arsenin, V.Y.: Solutions of Ill-Posed Problems. Washington: Winston & Sons (1977)