RECURRENT NEURAL NETWORK BASED CONTROL OF AN OIL WELL

Jean Panaioti Jordanou* , Eric Aislan Antonelo† , Eduardo Camponogara* , Marco Aurélio S. de Aguiar*

*Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina

†Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg

Email: jeanpanaioti@gmail.com, ericaislan.antonelo@uni.lu, eduardo.camponogara@ufsc.br, marcoaaguiar@gmail.com

Abstract— Echo State Networks (ESN) are dynamical learning models composed of two parts: a recurrent network (reservoir) with fixed weights and a linear adaptive readout output layer. The output layer's weights are learned for the ESN to reproduce temporal patterns usually by solving a least-squares problem. Such recurrent networks have shown promising results in previous applications to dynamic system identification and closed-loop control. This work applies an echo state network to control the bottom hole pressure of an oil well, whereby the opening of the production choke is manipulated. The controller utilizes a network to learn the plant inverse model, whose model input is the plant output and the vice-versa, and another network to compute the control action that induces a desired plant behavior. Despite the nonlinearities of the well model, the ESN effectively learned the inverse model and achieved near global setpoint tracking and disturbance rejection, with little setpoint deviation in the latter case. These results show that echo state networks are a viable tool for the control of complex dynamic systems by means of online inverse-model learning.

Keywords— Online System Identification, Reservoir Computing, Echo State Networks, Intelligent Control

Resumo— Redes de Estado de Eco (*Echo State Network*, ESN) são modelos de aprendizagem dinâmicos compostos por duas partes: uma rede neural recorrente com ponderação fixa e uma camada de saída de pesos adaptativos. Com a resolução de um problema de mínimos quadrados, os pesos da camada de saída são treinados para a rede reproduzir padrões temporais. Tais redes obtiverem resultados promissores na identificação e controle em malha-fechada de sistemas dinâmicos. Este trabalho aplica um controle adaptativo usando redes de eco no controle da pressão de fundo de um poço de petróleo, atuando sobre a abertura da válvula de produção. O controlador utiliza as redes de eco para identificar o modelo inverso da planta, que calcula a ação de controle para a planta seguir um comportamento desejado, utilizando a saída da planta como entrada da rede e vice-versa. Apesar da não-linearidade do modelo do poço, a rede de eco foi eficaz em aprender um modelo inverso, demontrando excelente desempenho em seguimento de trajetória e rejeição de perturbação, com baixo desvio do ponto de operação no último caso. Tais resultados demonstram a viabilidade das redes de eco para controle de sistema dinâmicos complexos por identificação de modelo inverso.

Palavras-chave— Identificação de Sistemas Online, Computação por Reservatório, Redes de Estado de Eco, Controle Inteligente

1 Introduction

An Echo State Network (ESN) is a recurrent neural network (RNN) with a (usually sparsely connected) hidden layer whose weights are fixed and randomly assigned. The training takes place only at the output layer, usually by linear regression methods (Jaeger, 2001), yielding an efficient learning process with global convergence properties. Echo State Networks are widely used nowadays for applications involving time series prediction and system identification, arguably so because of their ability to model and reproduce spatio-temporal patterns. Examples of works using this technique are: stock price prediction (Lin et al., 2009), learning of robot navigation behaviors (Antonelo and Schrauwen, 2015), noninvasive fetal QRS detection (Lukoševičius and Marozas, 2014) and even language modeling and processing (Hinaut and Dominey, 2012).

Applications to dynamic system control that use the structure presented in this work are also found in the literature. Proposed initially by Waegeman et al. (2012), the ESN based control structure considered in this work was applied to the control of a variable delay heating tank, a steady cruise airplane pitch control, and to balance control of an inverted double pendulum. Together with a sliding mode strategy, Park et al. (2014) used this structure to control a hydraulic excavator, a system with heavy nonlinearities. Galtier and Mathieu (2015) present another type of control structure, which uses least squares to train both the output and the input weights of only one ESN.

This work is inspired by (Waegeman et al., 2012) and other ESN-based applications in the oil industry: Antonelo et al. (2017) showcases not only the model identification of an offshore riser to serve as an observer, but also the design of soft sensors for the bottom-hole pressure since the real physical sensors operate in hazardous deep wells and, thus, are prone to failures. That work showed that an ESN can successfully model the nonlin-

ear behavior of a riser. However, the riser model did not account for pressure drop due to friction along the tubing. Motivated by the promising performance of the online learning control structure (Waegeman et al., 2012) and the universal dynamic system approximation of ESNs (Antonelo et al., 2017), the current work proposes their application to the control of an oil well.

In this context, this work faces two challenges. First, the ESN must learn an inverse model of the plant, meaning that if the plant reacts with an output y to an input u, $y = f(u)$, then the network must output u to an input y, $u = f^{-1}(y)$. Second, the control structure must compute a control action that will bring the plant's controlled variable to a certain desired value at a future time.

The riser model considered herein presents heavy nonlinearities, such as the pressure loss due to friction dynamics. Different from the classical approach, such as the PID or the $\mathcal{H}_\infty$ control strategy shown in (Jahanshahi et al., 2012), from which the riser model used in this work was originated, the ESN based controller is non-linear and, as such, can possibly achieve good performance without assigning an operating point. Also, for being a technique derived from machine learning, almost no prior knowledge of the model is needed to successfully tune the ESN based controller.

In section 2, we define the Echo State Network, the learning algorithm used, and describe the control loop used. In section 3, the oil well model is defined. In section 4, the results of the experiments made in this work are shown. Section 5 is the conclusion of this work.

2 Echo State Network Based Controller

Originally proposed by Waegeman et al. (2012), the ESN-based controller is composed of two RNNs, one denoted "Learning Network" (ESN-L) and the other referred to as "Control Network" (ESN-C). The Echo State Network is represented in Figure 1 and is a Recurrent Neural Network proposed by Jaeger (2001) which is governed by the following dynamic discrete time equations:

$$\begin{aligned}\mathbf{a}[k+1] &= (1-\gamma)\mathbf{a}[k] \\ &\quad + \gamma f(\mathbf{W_r^r}\mathbf{a}[k] + \mathbf{W_i^r}\mathbf{i}[k] + \mathbf{W_b^r}) \end{aligned} \tag{1}$$

$$\mathbf{o}[k+1] = \mathbf{W_r^o}\mathbf{a}[k+1] \tag{2}$$

where: the current state of the reservoir neurons, represented by the nodes in the center of Figure 1, is given by $\mathbf{a}[k]$, a vector usually with a dimension magnitudes higher than the input or the output; the current values of the input and output neurons, represented by the nodes in the left and right rectangles in Figure 1 respectively, are $\mathbf{i}[k]$ and $\mathbf{o}[k]$, respectively; γ is a tunable parameter called leak rate, which governs how much of

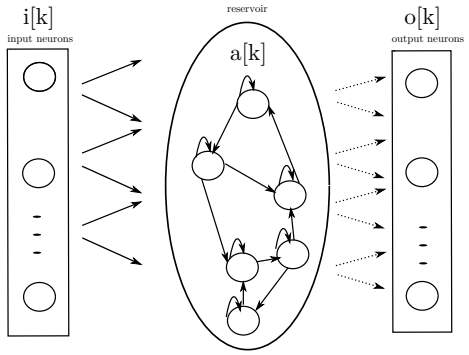

Figure 1: Representation of an Echo State Network.

the previous state is kept into the current reservoir's state. The weights, represented by the arrows in Figure 1, follow the notation W^{to}_{from}, with "o" meaning the output neurons, "r" meaning the reservoir, and "i" meaning the input neurons. The bias is denoted by "b." The dashed arrows in Figure 1 are the trainable ones and the solid arrows are fixed weights. The network has a number N of neurons, which is the dimension of $\mathbf{a}[k]$. The function f is the hyperbolic tangent $f = \tanh(\cdot)$ and is called an activation function in neural network literature.

In a standard Recurrent Neural Network (RNN), all the weights are trained by Backpropagation Through Time (Mozer, 1995). Such a training method of RNNs incurs a heavy computational cost, so ways of lowering this cost must be devised for RNNs to become practical.

The Echo State Property (Jaeger et al., 2007; Jaeger, 2001) is achieved when both the internal stability and steady state do not depend on the initial conditions (i.e., the reservoir has a fading memory). It has been shown that, when a recurrent neural network has this property, the network can learn by training only the output weights. Consequently, this reduces the problem of RNN training to a least squares problem (when trained to minimize the mean squared error). The number N of neurons is proportional to the learning capacity of the Echo State Network and the computational cost thereof. So the number N should be set only large enough for the network to not underfit.

Herein, the initialization method proposed by Waegeman et al. (2012) is followed, whereby every weight of the network is initialized from a normal distribution $\mathcal{N}(0,1)$. Then, the reservoir weight matrix $\mathbf{W_r^r}$ is scaled so that its spectral radius (eigenvalue with largest module) ρ is at a certain value able to create reservoirs with rich dynamical capabilities. Waegeman et al. (2012), Jaeger (2001) and Jaeger et al. (2007) argue that setting $\rho < 1$ in practice generates reservoirs with the echo state property (and in many cases maximize the performance of the network). Further, $\mathbf{W_i^r}$ and $\mathbf{W_b^r}$ are then scaled with the scaling fac-

tors f_i^r and f_b^r, respectively, to determine how the input will influence the network.

Figure 2 shows a block diagram representation of the control loop, which is composed of two Echo State Network blocks. One of the blocks depicted, "ESN-L", also referred to as "Learning Network," takes both the present plant output (denoted as $\mathbf{y}[k]$ in Figure 2) and a past plant output shifted δ timesteps to the past (denoted as $\mathbf{y}[k-\delta]$) as the network input. The constrained control action at time $k-\delta$ (denoted as $\mathbf{x}[k-\delta]$ in Figure 2), becomes the desired output of the corresponding training example. The Learning Network then tries to find the inverse model by inserting these data into a learning algorithm which updates the learning parameters (depicted by the dashed block in Figure 2). The algorithm used in this work is the Recursive Least Squares (RLS).

The Recursive Least Squares is an adaptive filter that solves a Weighted Linear Least Squares Problem using a recursive update formula for its parameters. The RLS used in this work is derived from the analytic solution of the Weighted Linear Least Squares Problem and obeys the following equations (Waegeman et al., 2012):

$$\mathbf{P}[0] = \frac{1}{\alpha} I \tag{3}$$

$$\mathbf{e}[k] = \mathbf{W_o^r}[k-1]\mathbf{a}[k] - \mathbf{x}[k] \tag{4}$$

$$\mathbf{P}[k] = \frac{\mathbf{P}[k-1]}{\lambda} - \frac{\mathbf{P}[k-1]\mathbf{a}[k]\mathbf{a}^T[k]\mathbf{P}[k-1]}{\lambda(\lambda + \mathbf{a}^T[k]\mathbf{P}[k-1]\mathbf{a}[k])} \tag{5}$$

$$\mathbf{W}_r^o[k] = \mathbf{W}_r^o[k-1] - \mathbf{e}[k]\mathbf{P}[k]\mathbf{a}[k] \tag{6}$$

where k is the current timestep and $\mathbf{P}[k]$ is called the covariance matrix, which can be considered as $\mathbf{P}[k] = (\mathbf{a}[k]\mathbf{a}^T[k])^{-1}$, as defined by Waegeman et al. (2012); $\mathbf{e}[k]$ is the error between the desided output and the actual output; I is the identity matrix. Also, α represents how much is known about the system, for it serves to evaluate $\mathbf{P}[0]$. The larger the value of α, the more one is admitting to not know the nature of the system.

The forgetting factor λ models how much weight the most recent samples will have in relation to the previous ones. The actual cost function for a forgetting factor included in the RLS algorithm is:

$$J = \sum_{k=0}^{N} \lambda^{-k}(\mathbf{y}[k] - \mathbf{W}_r^o\mathbf{a}[k]) \tag{7}$$

For $\lambda < 1$, the most recent samples are penalized more strongly. Smaller values of λ tend to make the algorithm adapt better to changes in the model, sacrificing steady state performance. So, heuristically literature recommends that $\lambda > 0.9$. The parameter λ should be close to 1 when the dynamic system is in steady state, however it is desirable to have λ close to 0.9 when the system is undergoing transients. This work implements the method for an adaptive forgetting factor proposed by Paleologu et al. (2008). The resulting equations that govern the dynamics of the forgetting factor are as follows:

$$\lambda[k] = \min\left(\frac{\sigma_q[k]\sigma_v[k]}{\epsilon + |\sigma_e[k] - \sigma_v[k]|}, \lambda_{\max}\right) \tag{8}$$

$$\sigma_e^2[k+1] = \alpha\sigma_e^2[k+1] + (1-\alpha)e^2[k] \tag{9}$$

$$\sigma_q^2[k+1] = \alpha\sigma_e^2[k+1] + (1-\alpha)q^2[k] \tag{10}$$

$$\sigma_v^2[k+1] = \beta\sigma_v^2[k+1] + (1-\beta)e^2[k] \tag{11}$$

$$\alpha = 1 - \frac{1}{K_\alpha N} \tag{12}$$

$$\beta = 1 - \frac{1}{K_\beta N} \tag{13}$$

$$q[k] = \mathbf{a}^T[k]\mathbf{P}[k]\mathbf{a}[k] \tag{14}$$

in which σ_x is an estimation of the expected value $E(x)$ and v is the noise associated with the error. The σ_x are updated by moving averages using time constants α and β, with $K_\alpha = 6$ and $K_\beta = 18$. The maximum value of the forgetting factor is denoted by $\lambda_{\max}$, whose value is $\lambda_{\max} = 0.9999$ so as not to compromise setpoint tracking. The algorithm lowers λ while the system is undergoing a transient (where the error rises) and raises λ in steady state (where the error is near 0). The threshold $\lambda_{\max}$ is imposed in order to ensure $\lambda \leq 1$. ϵ is a small number to avoid division by zero.

At the same time-step that ESN-L is trained, the ESN-C block receives a copy of $\mathbf{W_r^o}$ and computes the control action $\mathbf{u}[k]$ that, according to the inverse model identified by "ESN-L", guarantees that, given that the plant's current output is $\mathbf{y}[k]$, the future output will be $\mathbf{y}[k+\delta] = \hat{\mathbf{y}}[k+\delta]$. Since information learned from ESN-L is copied, ESN-C is essentially an approximation to the inverse model which is used to compute $\mathbf{x}[k]$. As input, ESN-C receives the present plant output $\mathbf{y}[k]$ and the desired plant output at δ time-steps into the future, which is referred to as $\hat{\mathbf{y}}[k+\delta]$. This is essentially the input to ESN-L, but displaced δ time-steps into the future. The control

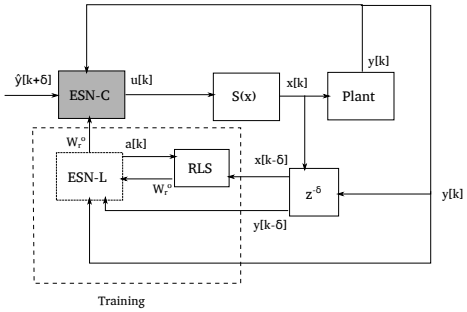

Figure 2: Block diagram of the ESN-based control framework.

action $\mathbf{u}[k]$ is then processed by the $S(x)$ block before being input to the plant and the Recursive Least Squares algorithm as $\mathbf{x}[k]$, as demonstrated in Figure 2. This block represents the system constraints, such as saturation and rate limiting. The timestep delay represented by δ is a tunable parameter of the framework, which is proportional to the time constants of the system. A proof of convergence of this type of control loop is found in (Waegeman et al., 2012).

3 The Well Model

The oil well model from (Jahanshahi et al., 2012) was selected for the case study to test the control methodology based on echo state networks. Figure 3 is a diagram representing the well structure, and also the physical location and meaning of each variable involved. The center of the figure depicts the tubulation where the oil is produced, called "Tubing". The borders represents where gas is injected for gas-lift, denoted as "Annulus".

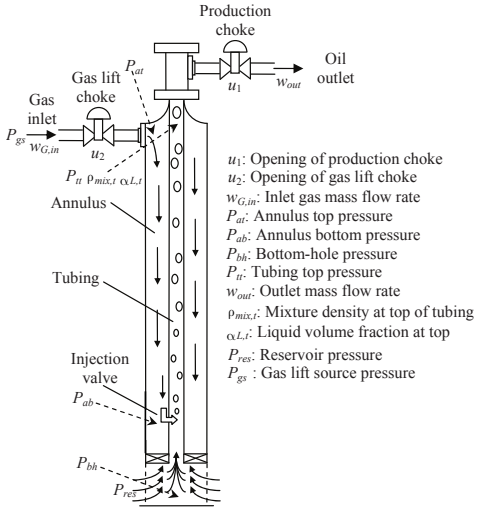

Figure 3: Schematic representation of the well model. Adapted from (Jahanshahi et al.,2012)

The well model consists of a choke valve for gas-lift injection, an annulus, a tubing, and a production choke valve at the end of the oil outlet. This is a typical configuration of a subsea satellite oil well, whose dynamics are described by the following state equations:

$$\dot{m}_{G,a} = \omega_{G,in} - \omega_{G,inj} \tag{15}$$
$$\dot{m}_{G,tb} = \omega_{G,inj} + \omega_{G,res} - \omega_{G,out} \tag{16}$$
$$\dot{m}_{L,tb} = \omega_{L,res} - \omega_{L,out} \tag{17}$$

where the name convention for variables is $x_{y,z}$:

- The x represents the variable's nature, with m being the mass and ω the mass flow.
- The y represents the variable's phase, with G being the gas and L the liquid/oil phase. The model assumes no water phase.
- The z represents the variable's location in the well, where tb is the tubing and a is the annulus.

If y is absent and the variable is in the form x_z, then the variable does not describe a specific phase. The flow equations are as follows:

$$\omega_{G,in} = K_{gs} u_2 \sqrt{\rho_{G,in} \max(P_{gs} - P_{at})} \tag{18}$$
$$\omega_{G,inj} = K_{inj} \sqrt{\rho_{G,ab} \max(P_{ab} - P_{tb})} \tag{19}$$
$$\omega_{out} = K_{pr} u_1 \sqrt{\rho_{mix,t} \max(P_{tt} - P_0)} \tag{20}$$
$$\omega_{res} = PI \max(P_{res} - P_{bh}) \tag{21}$$
$$\omega_{L,res} = (1 - \alpha^m_{G,b})\omega_{res} \tag{22}$$
$$\omega_{G,res} = \alpha^m_{G,b}\omega_{res} \tag{23}$$
$$\omega_{L,out} = (1 - \alpha^m_{G,t})\omega_{out} \tag{24}$$
$$\omega_{G,out} = \alpha^m_{G,t}\omega_{out} \tag{25}$$

in which K_x are tuning parameters to bring the dynamics close to a certain application. These variables are described in Figure 3, except P_0 which is the outlet pressure, normally of the manifold, $\alpha^m_{G,b}$ is the mass fraction of the bottom flow, and $\alpha^m_{G,t}$ is the mass fraction of the outlet flow. Further, $\alpha^m_{G,b}$ is assumed to be constant and $\alpha^m_{G,t}$ is calculated as:

$$\alpha^m_{G,t} = \frac{(1 - \alpha_{L,t})\rho_{G,t}}{\alpha_{L,t}\rho_L + (1 - \alpha_{L,t}\rho_{G,t})} \tag{26}$$
$$\alpha_{L,t} = 2\overline{\alpha}_L - \alpha_{L,b} \tag{27}$$
$$\alpha_{L,b} = \frac{\omega_{L,res}\rho_{G,tb}}{\omega_{L,res}\rho_{G,tb} + (\omega_{G,inj} + \omega G, res)\rho_L} \tag{28}$$
$$\overline{\alpha}_L = \frac{m_{L,tb} - \rho_L V_{bh}}{V_t \rho_L} \tag{29}$$

where $\overline{\alpha}_L$ is the average liquid fraction inside the tubing, V_{bh} is the assumed volume at the bottomhole, V_t is the volume in the tubing, and ρ_L is assumed to be constant. The rest of the densities are evaluated as:

$$\rho_{G,ab} = \frac{P_{ab} M_G}{R T_a} \tag{30}$$
$$\rho_{G,in} = \frac{P_{gs} M_G}{R T_a} \tag{31}$$
$$\rho_{G,t} = \frac{m_{G,tb}}{V_t + V_{bh} - m_{L,bh}/\rho_L} \tag{32}$$
$$\overline{\rho}_{mix} = \frac{m_{G,tb} + m_{L,tb} - \rho_L V_{bh}}{V_t} \tag{33}$$
$$\rho_{G,tb} = \frac{P_{tb} M_G}{R T_t} \tag{34}$$
$$\rho_{mix,t} = \alpha_{L,t}\rho_L + (1 - \alpha_{L,t})\rho_{G,t} \tag{35}$$

with $\overline{\rho}_{mix}$ being the average mixture density inside the tubing, T_a and T_t the temperatures in the annulus and tubing, assumed to be constant, R the

universal gas constant, and M_g the gas molecular weight.

The pressures are calculated as follows:

$$P_{at} = \frac{RT_a m_{G,a}}{M_G V_a} \tag{36}$$

$$P_{ab} = P_{at} + \frac{m_{g,a} g L_a}{V_a} \tag{37}$$

$$P_{tt} = \frac{\rho_{G,t} R T_t}{M_G} \tag{38}$$

$$P_{tb} = P_{tt} + \overline{\rho}_{mix} g L_t + F_t \tag{39}$$

$$P_{bh} = P_{tb} + F_b + \rho_L g L_{bh} \tag{40}$$

in which L_a is the length of the annulus, L_t is the length of the tubing, L_{bh} is the assumed length of the bottom hole. For being modeled as disturbances, P_{res}, P_{gs} and P_0 are considered to be constant. The F_b is the loss due to friction from the bottom hole to the injection point, also assumed to be constant, and F_t is the loss due to friction along the tubing, calculated as:

$$F_t = \frac{\lambda_b \rho_L \overline{U}_{l,b}^2 L_{bh}}{2D_t} \tag{41}$$

$$\frac{1}{\sqrt{\lambda_t}} = -1.8 \log_{10} \left(\left(\frac{\epsilon}{3.7 D_t} \right)^{1.11} + \frac{6.9}{Re_t} \right) \tag{42}$$

$$Re_t = \frac{\overline{\rho}_{mix} \overline{U}_{m,t} D_t}{\mu} \tag{43}$$

$$\overline{U}_{m,t} = \overline{U}_{sl,t} + \overline{U}_{sg,t} \tag{44}$$

$$\overline{U}_{sg,t} = \frac{4(\omega_{G,in} + \alpha_{G,b}^m \overline{\omega}_{res})}{\rho_{G,t} \pi D_t^2} \tag{45}$$

The above equations are derived from Haaland's solution to the Colebrook-White equation (1983) for the calculation of the friction factor of the tubing λ_t. The Re_t is the Reynolds number of the flow at the tubing, $\overline{U}_{m,t}$ is the average velocity in the tubing, $\overline{U}_{sg,t}$ is the average superficial velocity of the gas phase and $\overline{U}_{sl,t}$ is the average superficial velocity of the liquid phase, assumed to be constant. The D_t is the tube diameter, μ is the viscosity of the fluid. $\overline{\omega}_{res}$ is the average inlet flow rate, which is assumed to be constant to simplify the dynamics of the system. The parameters' values are borrowed from well n^o 1 of (Aguiar et al., 2015).

As it can be seen from all the equations that describe the oil well, the model is interesting for the control methodology presented in this work for its high nonlinearities. For the closed loop system to converge to the setpoint, the inverse model must be correctly learned by ESN-L, not only minimizing training error, but also making predictions in the form of the control action output by ESN-C. For the control strategy, the results on mixed sensitivity from (Jahanshahi et al., 2012) were considered for the purpose of analysis, whereby the bottom-hole pressure p_{bh} is controlled by adjusting the choke opening u_1. Also as in (Jahanshahi et al., 2012), the value of the gas-lift choke opening u_2 is fixed at 0.4. In this work, the control objectives are pressure setpoint tracking and small and large disturbance rejection. This allows us to test the Echo State Network's capacity to globally control the well model, although no formal guarantee is presented.

4 Results and Discussion

4.1 Setup

The results of the ESN-based control of the wellhead model are given in this section by simulation analyses carried out using the values for the controller's parameters in Table 1, which lists the value for each parameter of the controller described in Section 2.

Parameter	Value
γ: Leak Rate	0.3
ρ: Spectral Radius of $\mathbf{W_r^r}$	0.999
δ: Prediction Timesteps	3
ψ: Sparseness of $\mathbf{W_r^r}$	0
N: Number of Neurons	1000
f_i^r: Scaling Factor of $\mathbf{W_i^r}$	0.5
f_b^r: Scaling Factor of $\mathbf{W_b^r}$	0.1
T_s: Sampling Time	10 s

Table 1: Parameter value list for the Recurrent Neural Network controller.

First, a random reservoir was selected with $\rho = 0.999$ (shown empirically to produce rich reservoirs with the Echo State Property). Since performance depends on the randomly initialized weights $\mathbf{W_r^r}$, $\mathbf{W_i^r}$ and $\mathbf{W_b^r}$, a configuration which can guarantee setpoint tracking must be found. To find "working" weights, 3000 timesteps simulations were run with the same setpoint signal. If the reservoir manages to track the setpoint, the weights are saved and used in the next experiments. Around 5 trials were needed to find a "working" reservoir.

Then, the following procedures were executed for finding the values in Table 1. The neural network was implemented using *CUDAmat*, a *GPU* library for *Python*, due to high dimensional matrix multiplication. Due to a previous work (Waegeman et al., 2012) stating that sparseness has little effect on tracking, the network is fully dense, so $\psi = 0$. The scaling factors f_i^r and f_b^r were chosen arbitrarily. The neurons' number $N = 1000$ was chosen due to the large training errors at lower values for N. The sampling time $T_s = 10s$ was set in order to avoid oversampling of the model. The parameters γ and δ were found by grid search, where the goal was to minimize the sum of the quadratic error between $\hat{y}[k+\delta]$ and $y[k+\delta]$, $\forall k$ when testing the reservoir in tracking the same setpoint for 3000 timesteps (as in

the search procedure for the "working" weights described previously). The optimal δ between 1 and 50 was 3. The values $1 \leq \delta \leq 10$ for the timestep delay performed better in terms of settling time. The leak rate γ was tested from 0.1 to 1. The best result was obtained with $\gamma = 0.3$, probably because it allows the reservoir's state to carry enough information about the plant's past outputs for the control to work (Jaeger, 2001),(Jaeger et al., 2007). α was set to 1 (other values showed poor performance or affect numerical stability). Although Waegeman et al. (2012) utilized $\alpha = 10$ for their applications, but this value did not work for us probably due to the nonlinearities of the well model. For simulation of the oil well, the platform *Jmodelica* was used because of its interface with *Python 2.7*, where the control loop was implemented. The plant output $y[k]$ is scaled from [165, 193] bar to [0,1] before feeding it as input to the ESN.

4.2 Experiments

To show the performance of the control loop described in this work, five types of experiments were carried out. They were inspired by Jahanshahi et al. (2012), which showed an $\mathcal{H}_\infty$ approach to solve this problem. The ESN control loop is tested in order to verify whether it can globally track reference signals and reject disturbances.

Figure 4 and Figure 5 showcase the first experiment of tracking, where we test the capability of the ESN controller to take the bottom-hole pressure to different operating points in its domain over time. For all the plots featured in this work, the first subplot's dashed line is the desired bottom-hole pressure and the solid line is the actual p_{bh}. The u_1 described in the second plot is the choke opening and the e_{mean} in the third plot is the mean error. For the plots which are not Figure 4 and Figure 5, the transient due to intense learning at the beginning of the experiment can not be clearly inspected due to scaling reasons, though the effect is the same for all the experimetns.

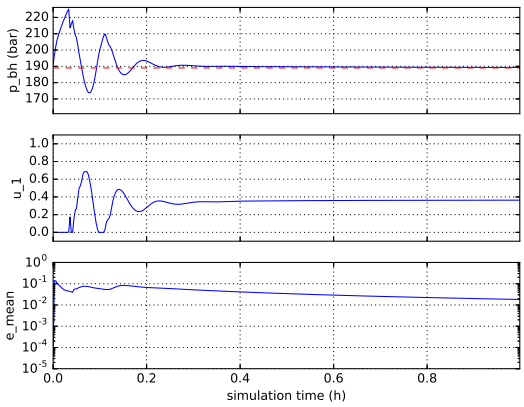

Figure 4: First hour of the tracking experiment simulation.

The first plot shows the manipulated variable being controlled, where the solid and dashed lines represent the actual output and the desired output over time. The second plot shows the control signal over time, while the third plot displays the "learning curve", which is $e_{mean}[k] = \frac{k}{k+1} e_{mean}[k-1] + \frac{e[k]}{k+1}$, the time evolution of the mean quadratic training error over all previous time steps. Although tracking is achieved, poor behavior with damping takes place in the first few minutes since learning is still insufficient, as shown in Figure 4. The system is better damped after the convergence of the RLS training, i.e., when the mean training error is in the order of 10^{-3}.

Figure 6 shows the result after applying small disturbances in the gas-lift inlet pressure p_{gs} of -3 and of $+3$ bar, respectively, while keeping fixed the bottom-hole pressure setpoint. The disturbance ceases at 27, 7h. The return to the setpoint is slower than the mixed sensitivity $\mathcal{H}_\infty$ approach presented in Jahanshahi et al. (2012). The maximum deviation over the reference signal was lower than 0.2 bar, which is quite low. Disturbances can be thought of changes in the nature of the model during the control task. Here, we note that the

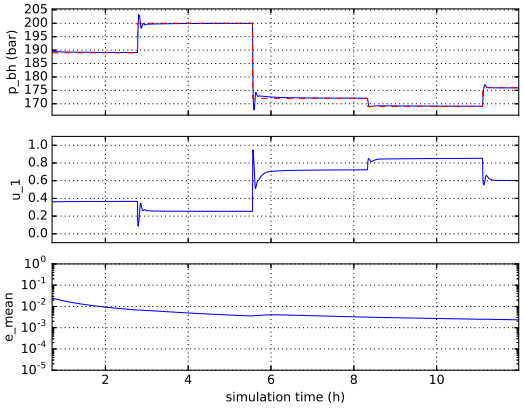

Figure 5: Tracking experiment simulation. This plot is a direct continuation of the simulation in Figure 4.

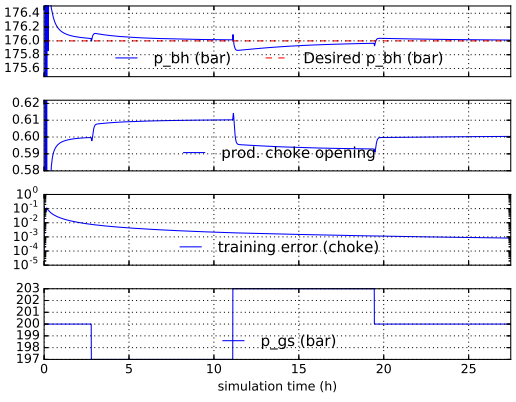

Figure 6: Simulation of disturbance in the gas-lift inlet pressure p_{gs} of -3 and of $+3$ bar at times: 5.5h and 16.6.

RLS training of the ESN can cope with disturbances quite well due to the presence of the forgetting factor λ, which boosts RLS's reaction to changes in the model.

Figure 7 shows that the control loop can cope with larger changes in the model, i.e., disturbances of ± 30 bar in the gas-lift inlet pressure. The disturbance ceases at 20h Although the slow convergence compromises effective tracking, the values deviate no more than 2 bar. This shows that large disturbance rejection is possible using Echo State Networks for control.

In order to further test the capabilities of the proposed ESN controller, we carried out experiments with the goal of not only tracking but also of disturbance rejection during the same simulation. Figures 8 and 9 show the effect of small disturbances and of large disturbances, respectively, and with starting times, duration and magnitude equivalent to the previous experiment on disturbance rejection.

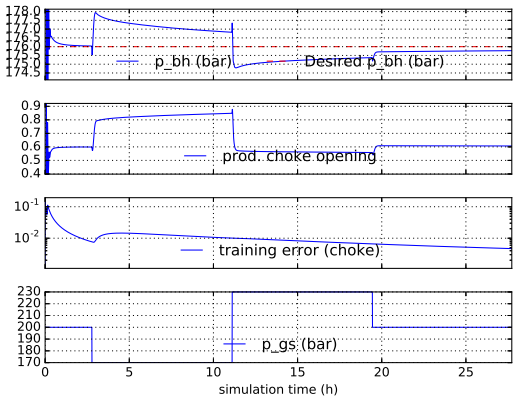

Figure 7: Same experiment as in Figure 6, except with a larger load disturbance of -30 and of $+30$ bar in p_{gs} at times: 5.5h and 16.6.

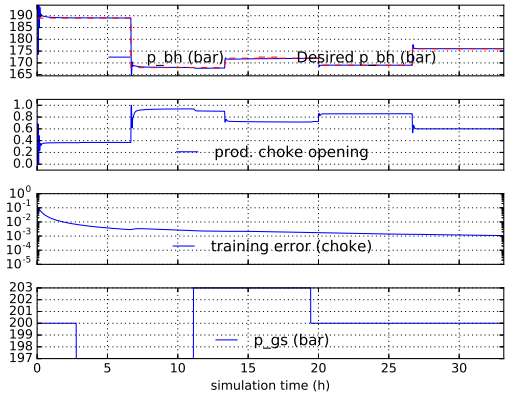

Figure 8: Experiment showcasing both the tracking aspect and the low magnitude of the effect of the disturbance.

We can note that tracking is still possible even with the model changing dynamically due to the disturbances applied. Besides, in Figure 9,when the setpoint $p_{bh} = 167$ bar with $p_{gs} = 170$ bar, the production choke opening is at its maximum. However, when p_{gs} changes from 170 bar to 230 bar, the closed loop system is able to track the setpoint with no oscillations or large overshoot, showing the absence of windup-like effects on the control loop.

5 Conclusions

This work has shown that an online learning control framework based on Echo State Networks and Recursive Least Squares is able to learn the inverse model of a system with strong nonlinearities and simultaneously control the same plant. Usually, the control of such a system, whose model is unknown, form an ill-posed problem. Considering this, it is relevant to highlight that this framework is able to learn an unknown model of the system on the fly: it requires no a priori knowledge of the system to be controlled. Furthermore, it effectively performs disturbance rejection with smaller overshoots than the ones obtained with linear approaches based on $\mathcal{H}_\infty$ (Jahanshahi et al., 2012). On the other hand, the disturbance rejection takes longer time to converge when compared to (Jahanshahi et al., 2012) (probably due to the RLS method used for the weight's update).

Although the computational cost of the control framework for one iteration is high (specially for large reservoirs), this does not represent an impediment in controlling such systems exhibiting slow dynamics, like the well model considered here. The second, most critical problem stems from the erratic transient behavior that can happen in the first iterations due to the intense period of inverse model learning taking place during this start-up period. Future work would aim

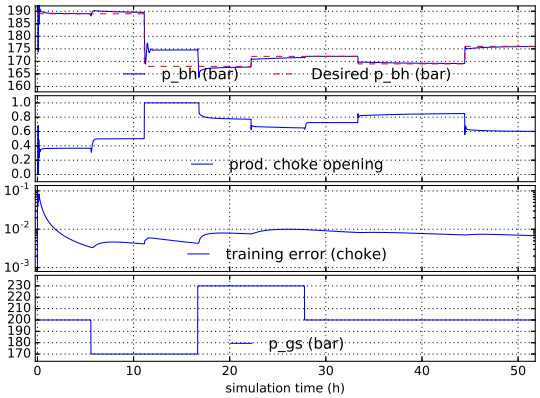

Figure 9: Same experiment as in Figure 8, except with a larger load disturbance.

at devising methods that guarantee the protection of the plant in view of this initial response of the closed loop, e.g., by observing another type of controller for the first iterations. Another issue is related to the procedure described here to initialize (randomly) the reservoir weights in order to find a "working" reservoir when considering a real plant. In this case, this procedure should be carried out using a phenomenological simulation of the plant in the control loop. When a good reservoir is found, it is transfered to the control of the real physical plant.

For future works in general, other loops in this plant could be tested. For example, using u_2 instead of u_1 to control p_{bh}, or using both p_{bh} and another variable to control the system. Since p_{bh} is difficult to measure, another controlled variable or an observer could be used. Also, it could be tested if convergence to a setpoint could be obtained if the plant is a whole oil and gas extraction platform, or a coupling of two wells and a riser. Considering that maximization of production is the goal in real oil plants, and that the control strategy presented in this work is of regulatory nature, an algorithm could be developed so that the control loop also maximizes production by integrating into it a cost function that penalizes pressure. Future work will also analyze the addition of measurement noise.

Acknowledgements

This research was supported partly by Petrobras. We would like to thank our colleagues at UFSC for providing insight and help in this research. We would like to thank Thiago Lima for assistance and insights related to the structure and development of this paper. We also would like to thank Agustinho Plucênio for control theory and oil platform modeling insights, and Tim Waegeman for the Matlab code provided for reference.

References

Aguiar, M. A., Codas, A. and Camponogara, E. (2015). Systemwide optimal control of offshore oil production networks with time dependent constraints, *IFAC-PapersOnLine* **48**(6): 200 – 207.

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

Antonelo, E. A. and Schrauwen, B. (2015). On learning navigation behaviors for small mobile robots with reservoir computing architectures, *IEEE Transactions on Neural Networks and Learning Systems* **26**(4): 763–780.

Galtier and Mathieu (2015). Ideomotor feedback control in a recurrent neural network, *Biological Cybernetics* **109**(3): 363–375.

Hinaut, X. and Dominey, P. F. (2012). Online processing of grammatical structure using reservoir computing, *in* A. E. P. Villa, W. Duch, P. Érdi, F. Masulli and G. Palm (eds), *ICANN: International Conference on Artificial Neural Networks*, pp. 596–603.

Jaeger, H. (2001). The ''echo state'' approach to analysing and training recurrent neural networks - with an erratum note.

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

Jahanshahi, E., Skogestad, S. and Hansen, H. (2012). Control structure design for stabilizing unstable gas-lift oil wells, *IFAC Proceedings Volumes* **45**(15): 93 – 100.

Lin, X., Yang, Z. and Song, Y. (2009). Short-term stock price prediction based on echo state networks, *Expert Syst. Appl.* **36**(3): 7313–7317.

Lukoševičius, M. and Marozas, V. (2014). Noninvasive fetal qrs detection using an echo state network and dynamic programming, *Physiological Measurement* **35**(8): 1685.

Mozer, M. C. (1995). Backpropagation, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, chapter A Focused Backpropagation Algorithm for Temporal Pattern Recognition, pp. 137–169.

Paleologu, C., Benesty, J. and Ciochinã, S. (2008). A robust variable forgetting factor recursive least-squares algorithm for system identification, *IEEE signal processing letters* **15**(10): 597 – 600.

Park, J., Cho, D., Kim, S., Kim, Y. B., Kim, P. Y. and Kim, H. J. (2014). Utilizing online learning based on echo-state networks for the control of a hydraulic excavator, *Mechatronics* **24**: 986–1000.

Waegeman, T., Wyffels, F. and Schrauwen, B. (2012). Feedback control by online learning an inverse model, *IEEE Transactions on Neural Networks and Learning Systems* **23**(10): 1637 – 1648.