Nonlinear Model Predictive Control of an Oil Well with Echo State Networks

Jean P. Jordanou * Eduardo Camponogara *
Eric Aislan Antonelo ** Marco Aurélio Schmitz Aguiar ***

* *Federal University of Santa Catarina, Florianópolis, SC 88040, Brazil (e-mail: jeanpanaioti@gmail.com).*
** *Interdisciplinary Centre for Security, Reliability and Trust University of Luxembourg (e-mail: ericaislan.antonelo@uni.lu).*
*** *Norwegian University of Science and Technology, Trondheim, NO-7491, Brazil (e-mail: marco.aurelio.aguiar@ntnu.no).*

Abstract: In oil production platforms, processes are nonlinear and prone to modeling errors, as the flow regime and components are not entirely known and can bring about structural uncertainties, making the design of predictive control algorithms a challenge. In this work, an efficient data-driven framework for Model Predictive Control (MPC) using Echo State Networks (ESN) as the prediction model is proposed. Unlike previous works, the ESN model for MPC is only linearized partially: while the free response of the system is kept fully nonlinear, only the forced response is linearized. This MPC framework is known in the literature as the Practical Nonlinear Model Predictive Controller (PNMPC). In this work, by using the analytically computed gradient from the ESN model, a finite difference method is not needed to compute derivatives as in PNMPC. The proposed method, called PNMPC-ESN, is applied to control a simplified model of a gas-lifted oil well, managing to successfully control the plant, obeying the established constraints while maintaining setpoint tracking.

Keywords: Model Predictive Control, Echo State Networks, Oil and Gas, System Identification

1. INTRODUCTION

The Echo State Network (ESN) is a type of Recurrent Neural Network (RNN) that provides good precision in modeling nonlinear dynamic systems, while enjoying easy training which makes it ideal for system identification and control applications. The basic assumption is that training is based only on a linear readout output layer (with linear regression methods) while keeping the weights of the hidden recurrent layer of neurons fixed (Jaeger et al., 2007). Among several successful uses of Echo State Networks in the literature, we can cite: slugging flow vertical riser system identification (Antonelo et al., 2017), grammatical structure processing (Hinaut and Dominey, 2012), and noninvasive fetal detection (Lukoševičius and Marozas, 2014).

Antonelo et al. (2017) was the first work to employ ESNs for applications in the oil and gas industry, performing a difficult task of system identification of slugging flow in vertical risers. They obtained ESN-based surrogate models that can be used as soft sensors and in control applications. In Jordanou et al. (2017), an on-line learning control framework based on ESNs was implemented for the control of an oil well. The controller, based on Waegeman et al. (2012), is adapted on-line by the Recursive Least Squares algorithm to model the inverse plant dynamics, being able to successfully perform control tasks such as reference tracking and disturbance rejection. A key disadvantage of the inverse model controller is the lack of a clear relation between the controller parameters and its effects on the system, such as in PID controllers. Thus, before they are deployed for industrial applications, further research is needed to provide safe guards and stability guarantees for inverse model controllers.

Model Predictive Control (MPC) is a class of control strategies that basically consists of two main steps: the use of a model to predict the future outputs given a certain control action; and the solution of an optimization problem according to the predictions made at each time step (Camacho and Bordons, 1999). Assuming that the model is sufficiently precise and the optimization problem is accurately solved, the control actions are taken within a safe operating region defined by the system constraints, and a good solution is found according to some criteria defined by the objective function.

Previous works using ESNs in MPC are discussed in the following. Pan and Wang (2012) obtained a linearized version of the Echo State Network at each time step, and used an unsupervised learning strategy to compensate the error generated by the Taylor series truncation, which also serves to reject disturbances. Not unlike the previous work, Xiang et al. (2016) linearized an identified ESN about an operating point and applied regularization to compensate for the truncation error.

This work proposes to introduce ESNs into the Practical Nonlinear Model Predictive Controller (PNMPC) frame-

⋆ This work was funded in part by NTNU, Petrobras, and CNPq.

work (Plucênio et al., 2007). PNMPC uses a fully nonlinear model to obtain the free response of the system, combined with a first-order Taylor expansion as the forced response relating the inputs to the outputs. This way, besides retaining the model precision for the free response, the calculation of the control action becomes a quadratic programming problem, akin to linear MPC strategies such as Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC) (Camacho and Bordons, 1999).

PNMPC has been shown to achieve good performance in several applications, one being the control of oil and gas processes (Plucênio, 2013), but with one drawback: since the prediction model is not specified, a finite difference method is needed to compute the derivative terms involved. This incurs a high computational cost if the number of process inputs and outputs is large, due to the combinatorial nature of the finite differences calculation. Since the derivative of an ESN can be analytically computed, this difficulty can be mitigated by using a correctly trained ESN as the prediction model.

The proposed data-driven framework, called PNMPC-ESN, is evaluated on controlling a simulated model of a gas-lifted oil well (Jahanshahi et al., 2012), which is a necessary condition to validate its usability in real offshore production plants. This framework is beneficial in three ways: only process data are required to run the plant control; the free response of the prediction model is kept fully nonlinear; and the cost function can be altered to fit any economic criteria, as done in (Plucênio, 2013) for a gas-lifted oil well application. The data-driven approach is particularly useful because of the structural uncertainties associated to the multiphase flow in oil production (Jahn et al., 2008).

Section 2 describes the echo state network. Section 3 presents PNMPC and the adaptations made to fit the echo state network. Section 4 gives a brief description of the well model used for the formulation of the optimization problem in the experiments. Section 5 consists of the results and Section 6 concludes the work.

2. ECHO STATE NETWORKS

An ESN is a type of recurrent neural network with convenient properties for system identification (Jaeger et al., 2007), such as being capable of nonlinear dynamics representation and Least Squares training. Proposed by Jaeger (2001), ESNs are described by the following discrete-time dynamic equations:

$$\begin{aligned}\mathbf{a}[k+1] &= (1-\gamma)\mathbf{a}[k] \\ &\quad + \gamma f(\mathbf{W}_\mathbf{r}^\mathbf{r}\mathbf{a}[k] + \mathbf{W}_\mathbf{i}^\mathbf{r}\mathbf{i}[k] + \mathbf{W}_\mathbf{b}^\mathbf{r} + \mathbf{W}_\mathbf{o}^\mathbf{r}\mathbf{o}[k])\end{aligned} \tag{1}$$

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

where: the state of the reservoir neurons at time k is given by $\mathbf{a}[k]$; the current values of the input and output neurons are represented by $\mathbf{i}[k]$ and $\mathbf{o}[k]$, respectively; γ is called leak rate (Jaeger et al., 2007), which governs the percentage of the current state $\mathbf{a}[k]$ that is transferred into the next state $\mathbf{a}[k+1]$. The weights are represented in the notation $\mathbf{W}_\mathbf{from}^\mathbf{to}$, with $\mathbf{o}$ meaning the output neurons, $\mathbf{r}$ meaning the reservoir, and $\mathbf{i}$ meaning the input neurons. "$\mathbf{b}$" represents the bias; and $f = \tanh(\cdot)$ is the activation function also called a base function in system identification theory (Nelles, 2001) being widely used in the literature. Figure 1 depicts the schematic of an echo state network.

The network has N neurons, which is the dimension of $\mathbf{a}[k]$ that must be several orders higher than the number of inputs. As long as training is regularized, N can be as big as needed, but at the expense of increased computation time when generating the reservoir states with (1).

The recurrent reservoir should have the so called Echo State Property (ESP) (Jaeger, 2001), i.e., a fading memory of its previous inputs, meaning that influences from past inputs on the reservoir states vanish with time. The ESP is guaranteed for reservoirs with $\tanh(\cdot)$ as the activation function when the singular values of $\mathbf{W}_\mathbf{r}^\mathbf{r} < 1$. However, this restriction limits the richness of dynamical qualities of the reservoir, and is not used in practice. Note that all connections going to the reservoir are randomly initialized, usually according to the following steps:

(1) Every weight of the network is initialized from a normal distribution $\mathcal{N}(0,1)$.
(2) $\mathbf{W}_\mathbf{r}^\mathbf{r}$ is scaled so that its spectral radius ρ (Eigenvalue with largest module) is at a certain value which is able to create reservoirs with rich dynamical capabilities. It has been often observed that setting $\rho < 1$ in practice generates reservoirs with the ESP (Jaeger et al., 2007).
(3) $\mathbf{W}_\mathbf{i}^\mathbf{r}$ and $\mathbf{W}_\mathbf{b}^\mathbf{r}$ are multiplied by scaling factors f_i^r and f_b^r, respectively, to determine how the input will influence the network.

These scaling parameters, ρ and f_i^r, f_b^r are crucial in the learning performance of the network, having an impact on how nonlinear and how much memory the reservoir has (Verstraeten et al., 2010). Also, low leak rates allow for higher memory capacity in reservoirs, while high leak rates should be used for quickly varying inputs and/or outputs. The settings of these parameters should be such that the generalization performance of the network (loss on a validation set) is enhanced.

While in standard RNNs all weights are trained iteratively using backpropagation through time (Mozer, 1995), ESNs restrict the training to the output layer $\mathbf{W}_\mathbf{r}^\mathbf{o}$, usually done in one-shot learning via ridge regression as follows (Bishop, 2006):

$$\mathbf{W}_\mathbf{r}^\mathbf{o} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \tag{3}$$

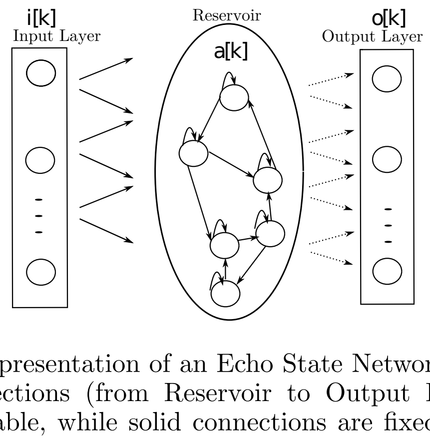

Fig. 1. Representation of an Echo State Network. Dashed connections (from Reservoir to Output Layer) are trainable, while solid connections are fixed and randomly initialized.

where $\mathbf{X}$, called the design matrix, is built by simulating the reservoir with (1) for $k = 1, \ldots, N_f$ (with initial state $\mathbf{a}[0] = 0$) and concatenating the corresponding reservoir states $\mathbf{a}[k]$ for $N_{wd} \leq k \leq N_f$; N_{wd} is known as the warm up drop time (Antonelo et al., 2017), which serves to cut off the examples during an initial undesired transient, and N_f is the final simulation step; $\mathbf{Y}$ consists of the target output of the training examples and λ is known as the regularization weight, which serves to avoid overfitting (Bishop, 2006). Output feedback $\mathbf{W_o^r o}[k]$ is not used in this work because it yields reservoirs without the ESP, thus $\mathbf{W_o^r} = 0$.

3. MODEL PREDICTIVE CONTROL STRATEGY

Model Predictive Control (MPC) emerged from three principles (Camacho and Bordons, 1999): the use of a prediction model to anticipate future outputs in a certain time period (also known as horizon); the calculation of a control sequence solving an optimization problem; and the application of, in most cases, the control action calculated for the first prediction. MPC algorithms differ mostly on the type of the model utilized. In this work, we use the closed-loop approach whereby, at the beginning of every sample time, a correction to the model prediction is made using the measured output. A predictive model is typically separated into a *free response*, which is the state the system would assume if no control action was applied, and a *forced response*, which is modeled as the application of a given control action. When the model is linear, this separation is rather trivial to be made, though the same cannot be said if the system is nonlinear (Camacho and Bordons, 1999).

Developed by Plucênio et al. (2007), the Practical Nonlinear Model Predictive Controller (PNMPC) provides a way of separating a nonlinear generic model into a free response and a forced response, utilizing a first order Taylor expansion. Solving a Nonlinear Programming Problem per time step might be computationally expensive, so the PNMPC has a computational advantage in which only a Quadratic Program (QP) is solved, which is significantly faster. Assuming a dynamic system in the form

$$\mathbf{x}[k+i] = \mathbf{f}(\mathbf{x}[k+i-1], \mathbf{u}[k+i-1]) \tag{4}$$
$$\mathbf{y}[k+i] = \mathbf{g}(\mathbf{x}[k+i]) \tag{5}$$

$$\mathbf{u}[k+i-1]) = \mathbf{u}[k-1] + \sum_{j=0}^{i-1} \Delta\mathbf{u}[k+j] \tag{6}$$

the prediction vector in PNMPC is calculated as follows:

$$\widehat{\mathbf{Y}} = \mathbf{G} \cdot \mathbf{\Delta U} + \mathbf{F}$$

$$\mathbf{\Delta U} = \begin{pmatrix} \mathbf{\Delta u}[k] \\ \mathbf{\Delta u}[k+1] \\ \vdots \\ \mathbf{\Delta u}[k+N_u-1] \end{pmatrix}$$

$$\mathbf{F} = \begin{pmatrix} \mathbf{g}(\mathbf{f}(\mathbf{x}[k], \mathbf{u}[k-1])) \\ \mathbf{g}(\mathbf{f}(\mathbf{x}[k+1], \mathbf{u}[k-1])) \\ \vdots \\ \mathbf{g}(\mathbf{f}(\mathbf{x}[k+N_y-1], \mathbf{u}[k-1])) \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} \frac{\partial \mathbf{y}[k+1]}{\partial \mathbf{u}[k]} & 0 & \cdots & 0 \\ \frac{\partial \mathbf{y}[k+2]}{\partial \mathbf{u}[k]} & \frac{\partial \mathbf{y}[k+2]}{\partial \mathbf{u}[k+1]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{y}[k+N_y]}{\partial \mathbf{u}[k]} & \frac{\partial \mathbf{y}[k+N_y]}{\partial \mathbf{u}[k+1]} & \cdots & \frac{\partial \mathbf{y}[k+N_y]}{\partial \mathbf{u}[k+N_u-1]} \end{pmatrix}$$

where N_y is the prediction horizon and N_u is the control horizon. The derivatives inside $\mathbf{G}$ are taken with respect to $\Delta\mathbf{u}[k+i] = 0, \forall i$, and $\mathbf{u}$ represents the manipulated variable vector. The vector $\mathbf{\Delta U}$ consists of the control increment values concatenated along N_u.

The equations above derive from the first-order Taylor series expansion in relation to the manipulated variables, whereby the free-response retains the nonlinearity, but the forced-response is linearized so that the control increment is calculated through a quadratic program. Since Plucênio et al. (2007) assume a generic nonlinear system, a finite-difference method is used to approximate derivatives, but this is computationally expensive when multiple variables are involved. A black-box model whose derivatives are calculated analytically drastically reduces the computation time, hence making the solution of the QP the only computationally expensive aspect of the proposed algorithm.

By using the chain rule, the derivatives are calculated:

$$\partial_{\mathbf{\Delta u}[k+j]}\mathbf{y}[k+i] = \partial_{\mathbf{x}[k+i]}\mathbf{g}\partial_{\mathbf{\Delta u}[k+j]}\mathbf{x}[k+i] \tag{7}$$
$$\partial_{\mathbf{\Delta u}[k+j]}\mathbf{x}[k+i] = \partial_{\mathbf{\Delta u}[k+j]}\mathbf{f} + \partial_{\mathbf{x}[k+i-1]}\mathbf{f}\partial_{\mathbf{\Delta u}[k+j]}\mathbf{x}[k+i-1] \tag{8}$$

Equation (8) enables us to recursively obtain $\mathbf{G}$. Considering that the dynamic matrix is evaluated at $\mathbf{\Delta U} = 0$, all the derivatives are evaluated at some $\mathbf{x}[k+i]$ and $\mathbf{u}[k-1]$, and $\partial_{\Delta\mathbf{u}}\mathbf{f}(\mathbf{x}[k+i]) = \partial_{\mathbf{u}}\mathbf{f}(\mathbf{x}[k+i])$. If $i > j$, for any two scalars j_1 and j_2, $\partial_{\mathbf{\Delta u}[k+j_1]}\mathbf{f}(\mathbf{x}[k+i]) = \partial_{\mathbf{\Delta u}[k+j_2]}\mathbf{f}(\mathbf{x}[k+i])$. So, for convenience, $\partial_{\Delta\mathbf{u}}\mathbf{f}(\mathbf{x}[k+i])$ is referred to as $\mathbf{J}(i)$. $\mathbf{S}(i)$ means $\partial_{\mathbf{x}}\mathbf{f}(\mathbf{x}[k+i])$.

By adapting Eqs. (7)-(8) to the above definitions, results:

$$\mathbf{G}_{ij} = \partial_{\mathbf{x}}\mathbf{g}\partial_{\mathbf{\Delta u_j}}\mathbf{x}[k+i] \tag{9}$$

$$\partial_{\mathbf{\Delta u_j}}\mathbf{x}[k+i] = \begin{cases} \mathbf{J}(i-1) + \mathbf{S}(i-1)\partial_{\mathbf{\Delta u}_j}\mathbf{x}[k+i-1] & i > j \\ \mathbf{J}(i-1) & i = j \\ 0 & i < j \end{cases} \tag{10}$$

where $p = N_y, m = N_u$, and $\mathbf{G}_{ij}$ represents the block element of $\mathbf{G}$ at row i and column j.

If an off-line trained ESN is used as the prediction model for the PNMPC, the derivatives are well defined (Pan and Wang, 2012; Xiang et al., 2016), being given as follows:

$$\partial_{\mathbf{x}}\mathbf{g} = \mathbf{W_r^o}$$
$$\mathbf{J}(j) = \partial_{\mathbf{z}_j}(\mathbf{f})\mathbf{W_i^r}$$
$$\mathbf{S}(j) = (1-\gamma)\mathbf{I} + \gamma\partial_{\mathbf{z}_j}(\mathbf{f})(\mathbf{W_r^r} + \mathbf{W_o^r}\mathbf{W_r^o})$$
$$\mathbf{z}_j = \mathbf{W_r^r}\mathbf{a}[k+j] + \mathbf{W_i^r}\mathbf{u}[k-1] + \mathbf{W_o^r}\mathbf{W_r^o}\mathbf{a}[k+j] + \mathbf{W_b^r}$$

Since, in this work, $\mathbf{f} = \tanh(\cdot)$, $\partial_{\mathbf{z}_j}\mathbf{f}$ is a diagonal matrix with all nonzero elements being $[1 - \tanh^2(\mathbf{z}_j)]$.

Summarizing, the trained ESN is used to calculate the free-response predictions and the Taylor approximation is calculated on-line to formulate the QP, which is solved at the current iteration. Since a Taylor expansion is used, an associated error is present in the predictive model. Also, errors inherent to disturbances and modeling are involved. Pan and Wang (2012) rely on a supervised learning strategy to estimate the Taylor expansion error, using the actual and predicted outputs as information. On the PNMPC, we consider the Taylor expansion error as part of the disturbance model. To treat disturbances and modeling errors, Plucênio et al. (2007) advocate the use of a low pass discrete filter on the error between the current measured output and the current prediction, which is computed as part of the free response. If the model were exactly equal to the plant and no disturbances were applied, the presence of the filter and the proposed closed-loop framework would be not different than an open-loop implementation. A slower filter could slow down the disturbance response, though it also increases the robustness of the controller. In practice, this is merely a different perspective to the problem, since the approach taken by Pan and Wang (2012) is equivalent to utilizing a variable static gain as a filter.

Then the free- and forced-response are obtained as follows:

$$\mathbf{F} = \begin{pmatrix} \mathbf{g}(\mathbf{f}(\mathbf{x}[k], \mathbf{u}[k-1])) \\ \mathbf{g}(\mathbf{f}(\mathbf{x}[k+1], \mathbf{u}[k-1])) \\ \vdots \\ \mathbf{g}(\mathbf{f}(\mathbf{x}[k+N_y-1], \mathbf{u}[k-1])) \end{pmatrix} + \mathbf{1}\boldsymbol{\eta}$$

$$\Delta\boldsymbol{\eta}[k] = K(1-\omega)(\widehat{\mathbf{y}}[k|k-1] - \mathbf{y}_m[k]) + \omega\Delta\boldsymbol{\eta}[k-1]$$

$$\widehat{\mathbf{y}}[k|k-1] = \mathbf{g}(\mathbf{f}(\mathbf{x}[k-1], \mathbf{u}[k-1])) + \boldsymbol{\eta}[k-1]$$

with $\mathbf{y}_m[k]$ being the measured variable and (K, ω) being the gain and leak rate of the filter, respectively, used to enhance the robustness capability of the controller. The cost function associated with a generic reference tracking problem, in matrix form, is as follows:

$$J = (\mathbf{Y}_{ref} - \widehat{\mathbf{Y}})^T \mathbf{Q} (\mathbf{Y}_{ref} - \widehat{\mathbf{Y}}) + \Delta\mathbf{U}^T \mathbf{R} \Delta\mathbf{U}$$

The diagonal matrices $\mathbf{Q}$ and $\mathbf{R}$ are the output and control weighting, whose utility is to express a variable's importance in the cost function.

Since the predicted output is stated in a form akin to the GPC and DMC strategies for MPC (Camacho and Bordons, 1999), the cost function is formulated as follows:

$$J = \Delta\mathbf{U}^T \mathbf{H} \Delta\mathbf{U} + \mathbf{c}^T \Delta\mathbf{U}$$
$$\mathbf{H} = \mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}$$
$$\mathbf{c} = \mathbf{G}^T \mathbf{Q}^T (\mathbf{Y}_{ref} - \mathbf{F})$$

The saturation constraints are formulated as follows:

$$\mathbf{1}\mathbf{u}_{\min} - \mathbf{1}\mathbf{u}[k-1] \leq \mathbf{T}\Delta\mathbf{U} \leq \mathbf{1}\mathbf{u}_{\max} - \mathbf{1}\mathbf{u}[k-1]$$

where $\mathbf{1}$ is a vector composed only of ones which matches the dimension and form of $\Delta\mathbf{U}$. If the problem was structured as a SISO (Single-Input Single-Output), $\mathbf{T}$ would be a lower triangle matrix. In this case, since a MIMO (Multi-Input Multi-Output) formulation is used where each variable is directly concatenated, $\mathbf{T}$ is postulated as follows:

$$\mathbf{T} = \begin{pmatrix} \mathbf{I}_{n_{in}} & \mathbf{0}_{n_{in}} & \mathbf{0}_{n_{in}} \\ \mathbf{I}_{n_{in}} & \ddots & \mathbf{0}_{n_{in}} \\ \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} \end{pmatrix}$$

where $\mathbf{I}_{n_{in}}$ is a n_{in} sized identity matrix and $\mathbf{0}_{n_{in}}$ is a n_{in} sized square matrix of zeros, n_{in} being the number of inputs to the system. Summarizing, $\mathbf{T}$ is a block triangular matrix of n_{in}-dimensional square matrices, where each column of the block matrix represents an instant in the prediction horizon.

The rate limiting constraints are stated as follows:

$$\Delta\mathbf{U}_{\min} \leq \mathbf{I}\Delta\mathbf{U} \leq \Delta\mathbf{U}_{\max}$$

where $\mathbf{I}$ is the identity matrix, with dimension $n_{in}N_u$.

As long as $\mathbf{Q}$ and $\mathbf{R}$ contain only positive values, $\mathbf{H}$ is positive definite, due to being composed by a lower triangular matrix and its transpose. This guarantees that the constraints and objective function, along with any other linear constraints, compose a convex quadratic programming problem.

4. CASE STUDY: OIL WELL

The oil well model from Jahanshahi et al. (2012) was selected as case study for the predictive controller testing. Figure 2 illustrates the oil well, and also the physical location and meaning of each variable. The figure center depicts the "Tubing", where oil is produced. The borders represents the "Annulus", where gas is injected for gas lift. The well model consists of a gas-lift injection choke valve,

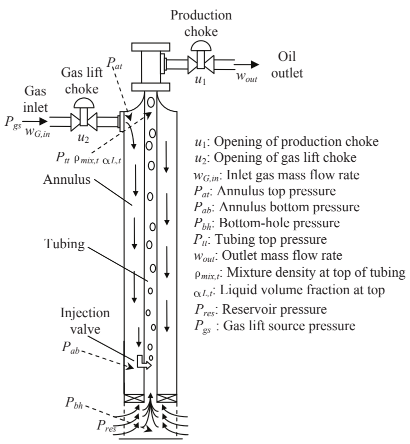

Fig. 2. Schematic representation of the well model. Adapted from (Jahanshahi et al.,2012).

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

$$\dot{m}_{G,a} = \omega_{G,in} - \omega_{G,inj} \tag{11}$$
$$\dot{m}_{G,tb} = \omega_{G,inj} + \omega_{G,res} - \omega_{G,out} \tag{12}$$
$$\dot{m}_{L,tb} = \omega_{L,res} - \omega_{L,out} \tag{13}$$

in which 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, and no water phase in the model. 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 parameter values are borrowed from well nº 1 of Aguiar et al. (2015), and for boundary conditions, the outlet pressure is 90 bar, the gas lift pressure is 200 bar, and the reservoir pressure is 250 bar. For more information on the formulation of the model, refer to Jahanshahi et al. (2012), who give more detailed information of the model.

Since a PNMPC framework is utilized, we can state the control objectives in the form of a quadratic optimization problem which calculates the control increment. The objective of the control problem is to track a reference signal for the well tubing bottom hole pressure P_{bh}, using both the gas-lift choke u_2 and the production choke u_1.

Due to the nature of the choke valves, $\mathbf{U}_{\max} = \mathbf{1}$ and $\mathbf{U}_{\min} = \mathbf{0}$. The choke valves have limited capacity and the larger the control increment is, the larger the magnitude of the error, so an increment limit is also applied. To limit the increment of the control action, the values of $\mathbf{\Delta} U_{\max} = 0.2$ and $\mathbf{\Delta U}_{\min} = -0.2$ are used.

The pressure at the top of the well tubing P_{tt} has an upper bound safe value $P_{\max}$ which must not be exceeded.

This leads to dividing the problem in two parts: first, an echo state network where $\mathbf{u}_1$ and $\mathbf{u}_2$ serve as input, and the pressures P_{bh} and P_{tt} serve as output is trained using data from (Jahanshahi et al., 2012). An ESN with two outputs can be viewed as two separate ESNs, one with P_{bh} as the output, and the other with P_{tt} as the output, due to the decoupling of the outputs in the network formulation. Since no output feedback is used in this work, at the second part, the following predictive control problem is solved per iteration using the identified ESN:

$$\min_{\mathbf{\Delta U}} J(\mathbf{\Delta U}) = \mathbf{\Delta U}^T \mathbf{H} \mathbf{\Delta U} + \mathbf{c}^T \mathbf{\Delta U}$$

$$\begin{aligned}
\text{s.t. } \mathbf{I\Delta U} &\leq \mathbf{\Delta U}_{\max} \\
-\mathbf{I\Delta U} &\leq -\mathbf{\Delta U}_{\min} \\
\mathbf{T\Delta U} &\leq \mathbf{1u}_{\max} - \mathbf{1u}[k-1] \\
-\mathbf{T\Delta U} &\leq -\mathbf{1u}_{\min} + \mathbf{1u}[k-1] \\
\mathbf{G_2\Delta U} &\leq \mathbf{1P}_{\max} - \mathbf{F_2}[k-1]
\end{aligned}$$

$$\mathbf{H} = \mathbf{G_1}^T \mathbf{Q} \mathbf{G_1} + \mathbf{R}$$
$$\mathbf{c} = \mathbf{G_1}^T \mathbf{Q}^T (\mathbf{Y}_{ref} - \mathbf{F_1})$$

where $\mathbf{G_1}$ ($\mathbf{G_2}$) and $\mathbf{F_1}$ ($\mathbf{F_2}$) correspond to the dynamic matrix and free response of the P_{bh} (P_{tt}) echo state network respectively, according to the PNMPC framework. $P_{\max}$ is set to 110 bar, and N_u, N_y, $\mathbf{Q}$ and $\mathbf{R}$ are left as tuning parameters. The cost function in this work does not consider economic aspects of the problem, though an economic cost function is intended in future works.

5. RESULTS

This section presents the results of the experiments. The algorithms were implemented in Python. The well model was implemented using Jmodelica, and the solver used for quadratic programming was Cvxopt.

5.1 Identification

For the identification of the ESN prediction model, 40,000 simulation samples are used for offline training and 10,000 for validation, which in practice can be obtained if the data log of the sensors and actuators in the plant are avaiable. By setting $\gamma = 0.8, \psi = 0.05, f_i^r = 0.2, f_b^r = 0, \rho = 0.999, N = 300, \lambda = 0.1$ (refer to Jaeger et al. (2007) for more information on tuning), not using output feedback and normalizing the training data, we obtained mean squared errors by the order of $2 \cdot 10^{-3}$ for both P_{tt} and P_{bh}, in both training and validation phases. The excitation signal was a stair sequence, random in both amplitude and signal frequency. The scalings used for normalization of u_1, u_2, P_{bh} and P_{tt} are respectively: $[0.02, 1]$, $[0, 1]$, $[150, 250]$, $[90, 150]$. A sampling time T_s of 10 s was used both during the system identification task and afterwards in the predictive control experiment. The ESN was used without output feedback to test the method's robustness to larger modeling errors, since the model is then less efficient in capturing oscillatory dynamics (Jaeger, 2001; Antonelo et al., 2017). The presence of output feedback would enable the echo state network to perform better.

5.2 Tracking Experiment

To test the application of the proposed predictive controller, we applied in the plant by setting $N_u = 6$, $N_y = 40$, control weights $\mathbf{R}$ are twice as large as the prediction weights $\mathbf{Q}$, and, for each measured parameter, $K = 0.001$ and $\omega = 0.3$ for both measured variables P_{bh} and P_{tt}.

The goal of the experiment is to assess if the controller can solve the proposed problem. To this end, a stair reference signal is applied at different points of operation.

Figure 3 depicts the result obtained during 1500 time steps of the experiment. The top plot consists of the control signal, representing the production choke valve u_1 and the green line represents the the gas-lift valve u_2. The middle plot gives the pressures P_{bh} (solid and thick blue line), P_{tt} (solid green line), the desired value for P_{bh} (red dashed line), and the hard constraint on P_{tt} (light blue dashed and thick line). The bottom plot represents the percentage relative estimation error of the ESN for P_{bh} and P_{tt}, which depicts the model accuracy:

$$e = 100 \cdot \frac{|estimated - measured|}{|measured|} \tag{14}$$

In the problem formulation, the bound is modeled as a constraint and the reference error as a penalization factor at the cost function. So naturally the priority for the optimization solver is to maintain the constraint, and the cost function penalization comes in second, though the controller still manages to obtain a good tracking response. All the setpoint change responses where super-critically damped (no presence of overshoot), except for the third, which might be related to the operating point. Only on the first setpoint was the bottom-hole pressure constraint

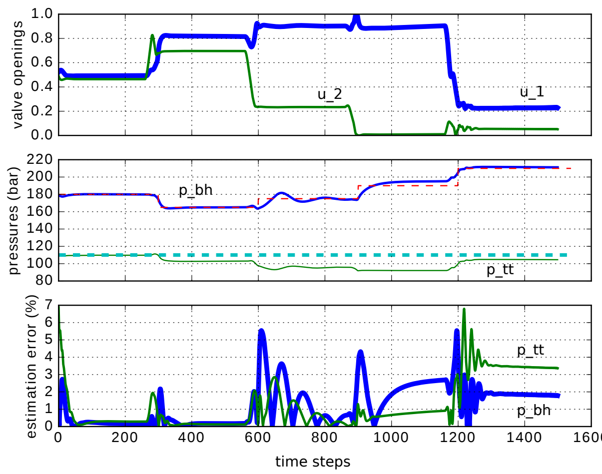

Fig. 3. Bottom-hole pressure (P_{bh}) tracking experiment.

active. There was setpoint error in the fourth setpoint due to the saturation constraint being reached, as the forced response prediction comes from a linear approximation model, which is not capable of finding a better solution.

Even though a large estimation error is present when the system is not at steady state, the controller succeeds in controlling the well with satisfactory transient behavior, even in the last setpoint, where larger relative modeling error is involved. This was due to the presence of the filter, which has the role of increasing controller robustness (Plucênio et al., 2007) and estimating a correction factor for modeling error. Though, in this case, larger values for K would lead to infeasibility of the optimization solver. Small K values lead to more robust filter designs (Plucênio, 2013), which is why a smaller gain, such as the one used, seemed to work. The filter tuning managed to successfully correct estimation errors between the echo state network and the plant.

6. CONCLUSION

This work proposed the PNMPC-ESN framework for MPC whose advantages are two-fold: from the ESN side, it obtains its data-driven capability for efficient system identification without a priori knowledge (Antonelo et al., 2017); from the PNMPC side, it formulates the problem such that only the forced response of the ESN is linearized for MPC, keeping the free response of the ESN model fully nonlinear (Plucenio et al., 2015), and thus more precise. Further, PNMPC-ESN eliminates the need of finite difference algorithms, enhancing computational efficiency for multivariable control problems. ESNs were shown to be suitable approximators for oil and gas production systems, to a great extent because offshore production platforms are subject to structural model uncertainties.

The results have shown that given a predictive control optimization problem formulation, the proposed PNMPC-ESN controller is able to respond satisfactorily to the given objectives and constraints. A more detailed parameter search and study of the ESN for system identification, as well as a more thorough study on filtering and controller tuning are open possibilities for future studies. Also, more realistic real-time optimization and predictive control problems can be proposed. Integrating this controller with online system identification is also a valid idea, since it would eliminate the need to have prior data on the model for the controller.

ACKNOWLEDGEMENTS

We thank NTNU, Petrobras and CNPq for the financial support, and also Julio Normey Rico and Rodolfo Flesch for insights into predictive control theory.

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.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Camacho, E. and Bordons, C. (1999). *Model Predictive Control*. Springer.

Hinaut, X. and Dominey, P.F. (2012). On-line processing of grammatical structure using reservoir computing. In *Int. Conf. on Artificial Neural Networks*, 596–603.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks - with an erratum note. Technical Report GMD 148, German National Research Center for Information Technology.

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.

Jahn, F., Cook, M., and Graham, M. (2008). *Hydrocarbon Exploration and Production*. Developments in Petroleum Science 55. Elsevier, 2nd ed edition.

Jordanou, J.P., Antonelo, E.A., Camponogara, E., and de Aguiar, M.A. (2017). Recurrent neural network based control of an oil well. In *Brazilian Symposium on Intelligent Automation*.

Lukoševičius, M. and Marozas, V. (2014). Noninvasive fetal QRS detection using an echo state network and dynamic programming. *Physiological Measurement*, 35(8), 1685–1697.

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

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verlag Berlin Heidelberg, 1 edition.

Pan, Y. and Wang, J. (2012). Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics*, 59(8), 3089–3101.

Plucênio, A., Pagano, D., Bruciapaglia, A., and Normey-Rico, J. (2007). A practical approach to predictive control for nonlinear processes. *IFAC Proceedings Volumes*, 40(12), 210–215.

Plucênio, A. (2013). *Development of Non Linear Control Techniques for the Lifting of Multiphase Fluids*. Ph.D. thesis, Federal University of Santa Catarina, Brazil.

Plucenio, A., Campos, M.M., and Teixeira, A.F. (2015). New developments in the control of fluid dynamics of wells and risers in oil production systems. *IFAC-PapersOnLine*, 48(6), 97–103.

Verstraeten, D., Dambre, J., Dutoit, X., and Schrauwen, B. (2010). Memory versus non-linearity in reservoirs. In *Int. Joint Conference on Neural Networks*, 18–23.

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.

Xiang, K., Li, B.N., Zhang, L., Pang, M., Wang, M., and Li, X. (2016). Regularized Taylor echo state networks for predictive control of partially observed systems. *IEEE Access*, 4, 3300–3309.