Online Learning Control with Echo State Networks of an Oil Production Platform ☆

Jean P. Jordanou[a], Eric Aislan Antonelo[b], Eduardo Camponogara[a]

[a] *Department of Automation and Systems Engineering, Federal University of Santa Catarina, 88040-900 Florianópolis, Brazil, e-mail: eduardo.camponogara@ufsc.br*

[b] *Interdisciplinary Centre for Security, Reliability and Trust, Université du Luxembourg, 29 avenue JF Kennedy, L-1855 Luxembourg, e-mail: ericaislan.antonelo@uni.lu*

Abstract

The design of a control algorithm is difficult when models are unavailable, the physics are varying in time, or structural uncertainties are involved. One such case is an oil production platform in which reservoir conditions and the composition of the multiphase flow are not precisely known. Today, with streams of data generated from sensors, black-box adaptive control emerged as an alternative to control such systems. In this work, we employed an online adaptive controller based on Echo State Networks (ESNs) in diverse scenarios of controlling an oil production platform. The ESN learns an inverse model of the plant from which a control law is derived to attain set-point tracking of a simulated model. The analysis considers high steady-state gains, potentially unstable conditions, and a multi-variate control structure. All in all, this work contributes to the literature by demonstrating that online-learning control can be effective in highly complex dynamic systems (oil production platforms) devoid of suitable models, and with multiple inputs and outputs.

Keywords:
Echo State Networks, Online Learning, Oil Production Wells, Control of Unknown Systems, Inverse Model Learning, Recurrent Neural Networks

1. Introduction

In spite of the recent advances in renewable energies, oil and gas remain the most important energy sources in the world. As oil reservoirs are drained with time, more sophisticated technologies are required for extraction, to a great extent due to the loss of reservoir pressure and the accumulation of impurities, such as wax, gas hydrates, and asphaltene to name a few. These hurdles led to the concept of "flow assurance," initially coined by Petrobras in the 90's, which is concerned with the development and applications of technologies that ensure a stable and economically desirable production, under a wide range of operating conditions (Jahanshahi, 2013).

One of the main flow assurance problems is the slugging flow in wells and risers, a phenomenon associated with the high oscillations that arise from the speed difference between the oil and gas flows. Most solutions found in the literature rely on models to design feedback control strategies, such as Jahanshahi (2013), de Oliveira et al. (2015), Campos et al. (2015), and Stasiak et al. (2012). Usually, the feedback control solutions to anti-slugging problems are either model-based or tuned based on a simplified model, which tends to lead to robustness issues due to either parametric or structural model mismatch (such as assuming the incorrect flow composition) (Jahanshahi, 2013). Rather than model-based approaches, this work turns to black-box data-driven approaches that continuously train online, using Recurrent Neural Networks (RNNs) to control a nonlinear plant. We can view the RNN as embedded in the controller, simultaneously learning the inverse model and controlling the plant. Throughout this paper, we will interchangeably use the terms controller and (recurrent neural) networks with the same meaning: an RNN-based controller.

As the online learning RNN derives the control law, the controller can learn and adapt its parameters directly from streaming data in an online way, which renders it suitable to deal with unmodeled phenomena, for

☆This is a draft version of the publication to appear in Engineering Applications of Artificial Intelligence journal.

which limited knowledge of the structure and parameters is available (Nelles, 2001). However, a black-box approach can only generalize the phenomenology and physics in the region within which data were provided to fit the models (i.e., they are poor in extrapolating their predictions to regions outside the training input space), being a drawback when compared to phenomenological approaches. This is an issue specially in offline learning, where periodic retraining of the model would be necessary for learning new operating points and dynamical behaviors. In this case, a phenomenological model (if available) should be used specially if computational power is abundant and time constraints are not tight. On the other hand, online learning allows us to continuously learn from data, updating the model in real time to the most recent samples. The approach taken in this work follows that line of thought, whereby the controller endlessly adapts over time, making it capable of learning and controlling the system at any operating region, with no prior knowledge of the model.

There are several examples of neural adaptive controllers in the literature. For instance, Lungu and Lungu (2018) adds a feed-forward neural network (NN) to a model-reference adaptive framework for airplane landing control. The Model-reference controller essentially forces the plant to behave as dictated by a reference model. This structure is linear by nature, so Lungu and Lungu (2018) proposed adding a feedback linearization based approach, by canceling out the non-linearities with their inverse model identified by the NN. This methodology is gray-box, because it still uses information from the linearized system to design the controller. Besides, since their work uses a feed-forward static neural network, the dynamics must be realized at the input layer through the use of lagged inputs, increasing input dimensionality by many folds. Also, as training uses backpropagation, problems such as local optima, numerical instabilities, and slow convergence can arise.

Although recent advances in deep learning have made possible the effective training of special recurrent networks such as the LSTMs (Long Short Term Memory) for large training sets (Graves et al., 2013), in general RNNs are still hard to train, since Backpropagation through time (Werbos, 1990) is susceptible to local minima, slow convergence, bifurcation, and other instability related problems. One example of alternative ways to overcome these learning problems in nonlinear dynamic system identification is by Münker and Nelles (2018), who propose a network of local (linear) Finite Impulse Response (FIR) models where each local model is trained separately with the help of a special regularization matrix. The idea is that the network connection between multiple linear dynamic systems leads to a model capable of capturing nonlinear dynamics.

In our work, to circumvent the aforementioned learning problems, the RNN used in our controller is trained according to the Reservoir Computing (RC) paradigm (Verstraeten et al., 2007). In particular, we employ Echo State Networks (ESN) (Jaeger and Haas, 2004; Jaeger, 2001), one of the flavors of RC, as a substrate for RNN-based controller learning. Liquid State Machines (LSM), originally from the computational neuroscience field, are another flavor of RC which use spiking neural networks as reservoirs instead of analog reservoirs as in ESNs (Maass et al., 2002) (the latter being easier to implement). RC (or ESN) networks are composed of two main parts: a dynamical non-linear state equation system which represents the recurrent part called *reservoir*, whose weights are randomly fixed and not trained, thus avoiding non-linear terms and dependence of time in the loss function; and a linear readout output layer, the sole part to be trained. Usually, the readout training in the RC approach is done through linear Least Squares or Ridge Regression, which has global convergence properties in contrast to LSTMs which use gradient-based learning on a nonlinear cost function, requiring orders of magnitude of more time to train than RC networks, for instance. ESNs have been successfully applied to a wide range of problems in the literature, such as learning complex goal-directed robot behaviors (Antonelo and Schrauwen, 2015), grammatical structure processing (Hinaut and Dominey, 2012), short-term stock prediction (technical analysis) (Lin et al., 2009), predictive control (Pan and Wang, 2012; Xiang et al., 2016), and noninvasive fetal detection (Lukoševičius and Marozas, 2014). Moreover, ESNs have shown promising results in identifying the complex dynamics involving a slugging flow riser (Antonelo et al., 2017), which is considered a difficult task in system identification. Chen et al. (2017a) present a tutorial on how to apply several types of neural networks, including Echo State Networks, into applications related to wireless communication, such as: using Unmanned Aerial Vehicles (UAV) as service providers; Virtual Reality with wireless networks; Content request prediction for mobile edge catching and computing; and even Internet of Things applications. Chen et al. (2017b) apply an enhanced version of Echo State Network into the context of cloud radio access networks, as a user behavior predictor. Chen et al. (2018) use ESNs in the context of virtual reality, using wireless networks to predict the quality of service of small base stations (information receivers and transmitters in the virtual reality framework) for the purpose of solving a resource allocation problem. Another successful use for

Echo State Networks is the non-linear predictive control of a gas-lifted oil well with no prior model information (Jordanou et al., 2018).

The ESN-based controller used in this work was first proposed by Waegeman et al. (2012), being referred herein also as the inverse model on-line learning controller. An Echo State Network implements the inverse model of the plant to be controlled, which is trained by the Recursive Least Squares (RLS) algorithm. This controller has been applied in problems such as: the heat tank temperature control assuming variable delay, airplane pitch angle control for steady cruise and balancing of an inverted pendulum system by Waegeman et al. (2012); position control of a real-world hydraulic excavator by Park et al. (2014); and robotic arm control by Waegeman et al. (2013). Bo and Zhang (2018) proposed another on-line learning control strategy that uses Echo State Networks. Based on an actor-critic framework, they derive an optimal control policy from the Bellman optimality equation to regulate the dissolved oxygen concentration in a wastewater treatment. Their controller is composed of four ESNs: two for modeling critics (as an approximation of the cost function to be minimized); one for modeling the plant (the forward model); and one for the actor, obtained analytically by deriving the steepest descent from the gradient of the model and critic networks. The ESNs are trained by a method based on the Recursive Least Squares. On the other hand, the controller employed in our work consists of only two ESNs, one that learns from past observed behavior in an online fashion (with RLS), and another that actually controls the plant. Knowledge is transferred in real time at every instant from the learning ESN to the control ESN. Compared to Bo and Zhang (2018), our framework is simpler and less computationally expensive, which can be readily applied to MIMO systems as shown in the experiments hereafter. The online learning control strategy used in this work is purely regulatory, while Bo and Zhang (2018) minimize an expected cost function to induce an optimal control policy in a reinforcement learning way.

In oil and gas, our previous work (Jordanou et al., 2017) employed the online learning controller in order to control a gas-lifted oil well model, successfully realizing set-point tracking and disturbance rejection tasks. The current work can be considered as a comprehensive extension of Jordanou et al. (2017), where now:

- the plant consists of two oil wells and a riser connected through a manifold (instead of a single oil well), providing a richer set of dynamical behaviors, coupling, and higher complexity when compared to Jordanou et al. (2017);
- we use riser and oil well models that were validated using the well-known OLGA simulator, the standard choice in the oil and gas industry (Jahanshahi et al., 2012; Jahanshahi and Skogestad, 2011).
- our work tackles the multivariate aspect of control in oil and gas platforms using a single ESN architecture, which was not present in Jordanou et al. (2017). In particular, we perform experiments on SISO (single-input single-output) control, where we control the riser inlet pressure by manipulating the opening of the production choke while ignoring the gas-lift valves of both oil wells, which are fully closed. Next, as the gas-lift valves are important in governing the system dynamics, MISO (multiple-input single-output) control is proposed where the same riser inlet pressure is controlled, but now manipulating other two variables, i.e., the opening of the gas-lift valves for both oil wells whereas the production choke is set maximally open, guaranteeing the most oil production in this setting. The controller learns to satisfactorily control the plant albeit the presence of unstable configurations of the plant inputs (manipulated gas-lift valves), a situation not present in (Jordanou et al., 2017). Finally, MIMO control is presented where now the controlled variables are the bottom-hole pressure of both oil wells and the manipulated variables are the production choke of each well. Note that, in this case, a single ESN receives as input the readings of the bottom-hole pressure of both wells as well as the reference values for them (totaling 4 network inputs) and produces two outputs corresponding to the openings of the production chokes. We show that this single network can perform MIMO control on this complex coupled system on the fly, i.e., online and without a priori knowledge. Related work on MIMO control can be found in (Waegeman et al., 2013), where a modular architecture called MACOP adaptively combines a number of ESN-based controllers (where each one learns a control primitive) in a multi-variable robotic arm control task.

Finally, this work demonstrates that black-box strategies can be effective to the control of processes in oil production platforms, which is relevant for flow assurance as discussed above.

This paper is organized as follows. Section 2 presents the Echo State Network in more detail. Section 3 describes the online learning control strategy and the Re-

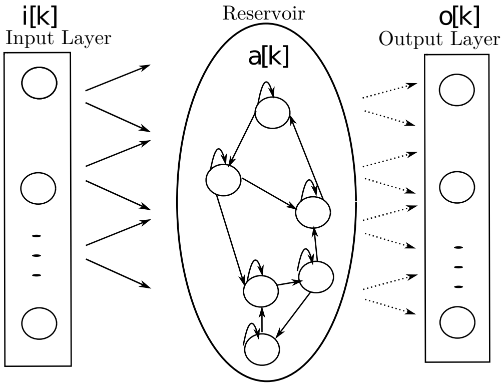


Figure 1: Representation of an Echo State Network. Dashed connections (from Reservoir to Output Layer) are trainable, while solid connections are fixed and randomly initialized. Figure extracted from Jordanou et al. (2017).

cursive Least Squares algorithm. Section 4 presents the well and riser models utilized, and further states the control challenges involved. Section 5 reports and discusses the simulation results. Section 6 concludes the work.

2. Echo State Networks

An ESN is a type of recurrent neural network with useful characteristics for system identification (Jaeger et al., 2007), as it represents nonlinear dynamics well and the training consists in solving a linear least squares problem of relatively low computational cost. Proposed by Jaeger and Haas (2004); Jaeger (2001), the ESN is governed by the following discrete-time dynamic equations:

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

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

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 $\mathbf{W_{from}^{to}}$, with "**o**" meaning the output neurons, "**r**" meaning the reservoir, and "**i**" meaning the input neurons. “**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 network inputs. As long as we use regularization, N can be as big as needed, but at the expense of increased computation time when generating the reservoir states according to Equation (1). According to Jaeger (2002), the ESN has a memory capacity bounded by the number of neurons in the reservoir (MC ≤ N), assuming that linear output units are used.

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 $\mathbf{W_r^r} < 1$. However, this restriction limits the richness of the reservoir dynamical qualities, 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. $\mathbf{W_r^r}$ is scaled so that its spectral radius ρ (Eigenvalue with the 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). However, reservoirs with $\rho > 1$ can still have the ESP since the effective spectral radius may still be lower than 1 (Ozturk et al., 2007; Verstraeten and Schrauwen, 2009).
3. $\mathbf{W_i^r}$ and $\mathbf{W_b^r}$ 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 the nonlinear representation and memory capacity of the reservoir (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 $\mathbf{W_r^o}$. In this work, reservoirs have no feedback from the output, i.e., $\mathbf{W_o^r} = 0$. Note that output feedback $\mathbf{W_o^r}\mathbf{o}[k]$ yields reservoirs without the ESP. Also, the inputs do not interfere directly in the output, as systems with di-

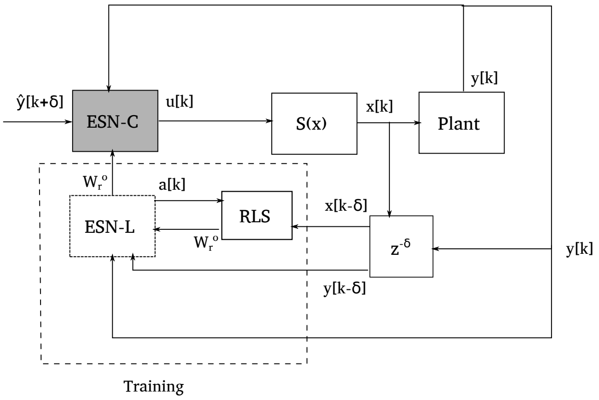


Figure 2: Block diagram of the ESN-based control framework. Details about its structure and functioning are given in the text.

rect transmission are less smooth and more sensitive to noise.

3. Online-Learning Control

Originally proposed by Waegeman et al. (2012), the ESN-based controller is an adaptive control framework composed of two recurrent neural networks: ESN-L, the "Learning Network", and ESN-C, the "Control Network". While ESN-L adapts its parameters in an online way, and is analogue to the "adaptive law" concept in an adaptive control framework (Astrom and Wittenmark, 1994), ESN-C computes the control action, and is analogue to the concept of the standard controller, whose parameters are defined by the adaptive law (Astrom and Wittenmark, 1994).

Figure 2 shows a block diagram representation of the control loop, which is composed of the two aforementioned Echo State Network blocks. The "ESN-L" 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 given at time $k-\delta$ (denoted as $\mathbf{x}[k-\delta]$ in Figure 2), corresponds to the desired output of a training sample. Aiming at finding the inverse model, the Learning Network is trained at each timestep with such a data sample (depicted by the dashed block in Figure 2) using the Recursive Least Squares (RLS) online learning algorithm which adapts the output weights $\mathbf{W_r^o}$ (Waegeman et al., 2012). Note that the latter is shared between ESN-L and ESN-C, but in practice, just after ESN-L is trained, its adapted weights $\mathbf{W_r^o}$ are copied to the output layer of ESN-C in the same timestep.

ESN-C outputs the control action $\mathbf{u}[k]$ according to the inverse model identified by "ESN-L" (defined by $\mathbf{W_r^o}$), where it is estimated that, if the plant's current output is $\mathbf{y}[k]$, the future output will be $\mathbf{y}[k+\delta] = \hat{\mathbf{y}}[k+\delta]$. The present plant output $\mathbf{y}[k]$ and the desired plant output $\hat{\mathbf{y}}[k+\delta]$ are the inputs to ESN-C. We assume that the model does not vary in the time interval δ, so this can be seen as the input to ESN-L, but displaced δ timesteps into the future. The control action $\mathbf{u}[k]$ is then processed by the $S(x)$ block before being input as $\mathbf{x}[k]$ to the plant and the Recursive Least Squares algorithm as illustrated in Figure 2. This block represents 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. Waegeman et al. (2012) present a proof of convergence for this type of control loop.

3.1. Recursive Least Squares

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):

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

$$\mathbf{e}[k] = \mathbf{W_r^o}[k-1]\mathbf{a}[k] - \mathbf{d}[k] \tag{4}$$

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

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

where k is the current timestep and $\mathbf{P}[k]$ is seen as an estimate of the covariance matrix at each time step k (Nelles, 2001); $\mathbf{e}[k]$ is the error between the desired ESN-L output $\mathbf{d}[k]$ (related to the control action input to the plant) and the actual ESN-L output ($\mathbf{W_r^o}[k-1]\mathbf{a}[k]$); I is the identity matrix. In the control framework, $\mathbf{d}[k]$ is built by using the bounded output of the ESN-C network $\mathbf{x}[k] = S(\mathbf{u}[k])$, i.e., according to previous section, $\mathbf{d}[k] = \mathbf{x}[k-\delta]$. $\mathbf{u}[k]$, in turn, is generated by exciting ESN-C with a given reference signal $\hat{\mathbf{y}}[k+\delta]$, usually generated randomly or by using staircase signals, as the experiments in this paper will show; α is a parameter that represents how much is known a priori 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 diagonal of $\mathbf{P}[0]$ becomes smaller).

The forgetting factor λ models how much weight the most recent samples will have in relation to the previous

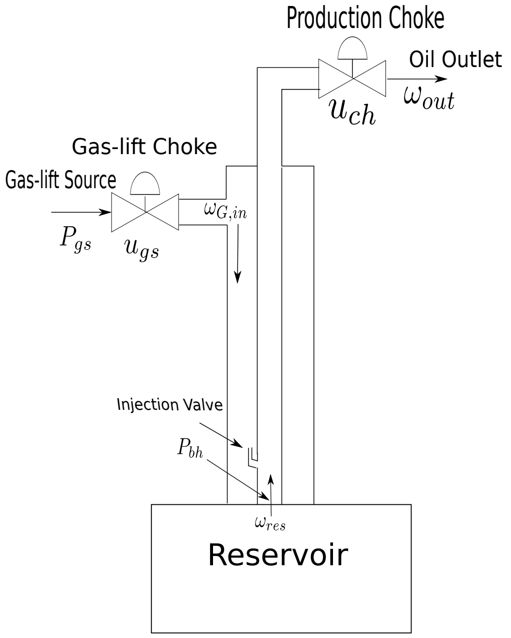


Figure 3: Schematic representation of the well model.

ones. The actual cost function for a forgetting factor included in the RLS algorithm is:

$$J = \sum_{k=0}^{N} \lambda^{N-k}(\mathbf{d}[k] - \mathbf{W}_{\mathbf{r}}^{\mathbf{o}}\mathbf{a}[k])^2 \tag{7}$$

For $\lambda < 1$, the most recent samples are penalized more strongly. Smaller values of λ tend to make the algorithm adapt better and quicker to real-time changes in the model while sacrificing steady-state performance. Heuristically, the 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.

4. Case Study

For the experiments, we used a composite model of a production platform consisting of two gas-lifted oil wells and a riser. The well model used to represent the two wells was developed in (Jahanshahi et al., 2012). We also use the same parameter values. The riser model was developed in (Jahanshahi and Skogestad, 2011). The models are connected by a manifold, which contains no load loss due to friction. We describe each model as follows.

4.1. Gas-Lifted Well

The well model consists of a gas-lift injection choke valve, an annulus, a tubing, and a production choke valve at the end of the oil outlet. Figure 3 illustrates the oil well model utilized, and also the physical location and meaning of each variable. The figure center depicts the "Tubing", where oil is produced. The borders represent the "Annulus", where gas is injected for gas lift. The gas-lifted oil well dynamics are described by the following state equations:

$$\dot{m}_{G,a} = \omega_{G,in} - \omega_{G,inj} \tag{8}$$

$$\dot{m}_{G,tb} = \omega_{G,inj} + \omega_{G,res} - \omega_{G,out} \tag{9}$$

$$\dot{m}_{L,tb} = \omega_{L,res} - \omega_{L,out} \tag{10}$$

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 is modeled. 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 subscript "res" refers to variables related to the oil and gas reservoir connected to the wells, "inj" refers to the injected flow from the annulus into the tubing, and "out" refers to the outlet variables. The parameter values are borrowed from well n^o 1 of Aguiar et al. (2015).

The model assumes only the presence of a gaseous and a liquid phase flowing through the well. The state equations derive from liquid and gas mass balance relations in each well, where one state represents the total gas mass present in the well annulus, and the other two states represent the gas and liquid total mass in the tubing. The inflow of gas in the annulus, $\omega_{G,in}$, depends on the opening of the gas-lift choke valve u_{gs}, and the gas-lift inlet pressure P_{gs}. The gas enters the tubing with mass flow $\omega_{G,inj}$ after exiting the annulus. The tubing also receives gas and liquid from the reservoir with given mass flows $\omega_{G,res}$ and $\omega_{L,res}$, respectively. These mass flows depend on the reservoir pressure, P_{res}. The liquid and gas exiting through the tubing, with mass flows $\omega_{L,out}$ and $\omega_{G,out}$, respectively, depend on the pressure at the outlet, which in this work is a manifold connected to another well and a riser, and the production choke valve opening u_{ch}.

All the flows are defined by the Bernoulli orifice equations, which depend on the pressure difference in each flow region. For more information on the formulation of the model, refer to (Jahanshahi et al., 2012) and Appendix A, where more detailed information is available. In (Jahanshahi et al., 2012), the well model was validated by comparison with the OLGA simulator, by capturing similar stability behavior, and thus serving the purpose of control benchmarking for oil and gas pro-

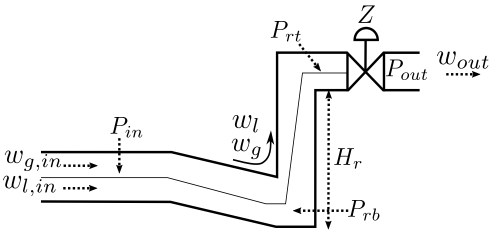


Figure 4: Representation of a pipeline-riser system. P_{out} represents the pressure in the separator. Obtained from Jahanshahi et al. (2013).

duction. The well model has 42 algebraic variables, 3 state variables, 2 input variables, and 3 boundary conditions, where two of them (reservoir pressure and gas-lift pressure) are fixed. The outlet pressure, $P_{w,out}$ is coupled via the manifold model to the rest of the system. In this work, two wells are considered, whose variables are defined by $x_{y,z,i}$, following the previously defined nomenclature, where i refers to the index of the well $i \in \mathcal{W} = \{1, 2\}$.

4.2. Pipeline-Riser

Jahanshahi and Skogestad (2011) idealized the pipeline-riser model utilized in this work. The same notation as the well model is used for the pipeline-riser model, considering only a biphase flow consisting of a liquid (oil and water) and a gaseous phase. Since slugging is related to the velocity difference between the gas and liquid phase, for anti-slug control applications, this model is reasonable. The liquid phase is also assumed to be incompressible. Jahanshahi and Skogestad (2011) demonstrate that this model provides a close approximation to an equivalent riser represented in the OLGA simulator. This model was made for benchmarking and designing anti-slug controllers, given that it represents well the dynamics related to the riser. Figure 4 is a description of the model.

The following state equations are considered:

$$\dot{m}_{G,p} = \omega_{G,in} - \omega_{G,lp} \tag{11}$$

$$\dot{m}_{L,p} = \omega_{L,in} + \omega_{L,lp} \tag{12}$$

$$\dot{m}_{G,r} = \omega_{G,lp} - \omega_{G,out} \tag{13}$$

$$\dot{m}_{L,r} = \omega_{L,lp} + \omega_{L,out} \tag{14}$$

These state equations are derived from mass balance, considering that the flow consists of only a gaseous and a liquid phase. The mass balance is evaluated in two connected regions: a horizontal pipeline, and a vertical tubing connected to it that elevates the production fluid. The mass flow rates $\omega_{G,in}$ and $\omega_{L,in}$, which refer to the fluid entering the pipeline, are either assumed constant or connected to an outside source, such as a manifold. The fluid then exits the pipeline and enters the riser with flow rates $\omega_{L,lp}$ and $\omega_{G,lp}$, and exit the system with flow rates $\omega_{G,out}$ and $\omega_{L,out}$, which depend on the separator pressure P_s and the riser choke valve opening z. As in the case of the well model, the flows are calculated through the Bernoulli orifice equation, and pressure loss due to friction is inherent in the model. For more information on the riser model, refer to Jahanshahi and Skogestad (2011) and Appendix B, where more detail of the model is available.

The pipeline-riser model possesses 4 state variables, 36 algebraic variables, and three boundary conditions, in which one (the outlet/separator pressure) is considered constant.

4.3. Manifold Connection

The entity that connects the two wells and the riser is the Manifold. The manifold considered in this work has no load loss due to friction, so the outlet pressure is equal to the two inlet pressures. This essentially means that the pressure at the outlet of both wells is equal to the riser inlet pressure P_{in}, as defined in Eqn. (15):

$$P_{in} = P_{w,out,1} = P_{w,out,2} \tag{15}$$

Given the conservation of mass, the sum of the outlet mass flow from the wells is equal to the inlet flow of the riser.

$$\omega_{G,in,r} = \omega_{G,out,w1} + \omega_{G,out,w2} \tag{16}$$

$$\omega_{L,in,r} = \omega_{L,out,w1} + \omega_{L,out,w2} \tag{17}$$

where $\omega_{x,y,r}$ is a mass flow in the riser and $\omega_{x,y,wi}$ is a mass flow in well i.

The presence of the manifold then brings the complete system to possess: 10 state variables, 110 algebraic variables, and 5 input variables. Due to the junction of the two wells and the riser, the riser inlet flow and both well outlet pressures can no longer be considered boundary conditions. In the end, the boundary conditions left for the complete system are: both well gas-lift pressures $P_{gs,1} = 200$ bar, and $P_{gs,2} = 200$ bar, both well reservoir pressures $P_{r,1}$, and $P_{r,2}$, and the riser outlet pressure, which is connected to the surface of a platform. As in (Jahanshahi and Skogestad, 2011), the outlet pressure is assumed to be 50.1 bar.

Figure 5 depicts a schematic representation of the complete plant, where the manifold connects the two wells and the riser.

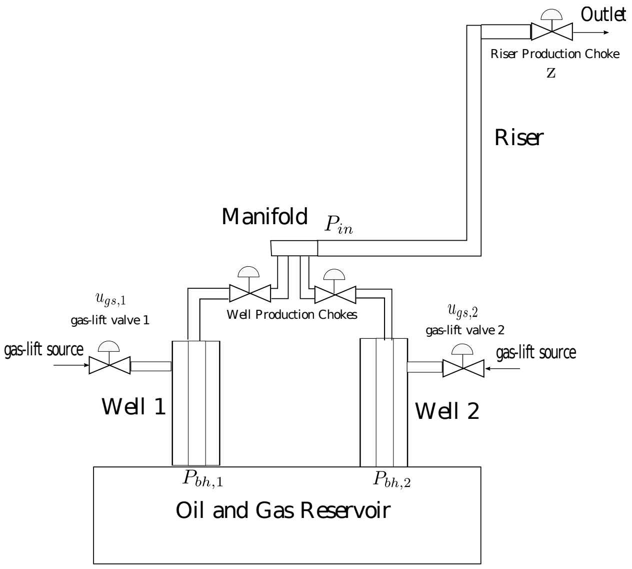


Figure 5: Schematic of the complete oil and gas production system.

4.4. Control Challenges

The oil production system depicted in Figure 5 is sufficiently complex to bring about several control challenges, which are suitable for testing different control problems such as:

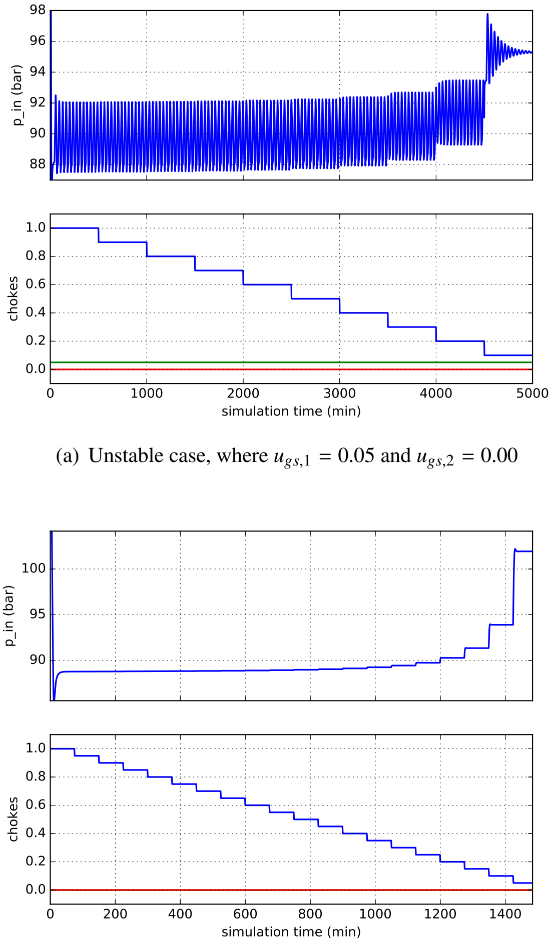
- **Riser Inlet Pressure Tracking (SISO)**: Pressure tracking is essential for production optimization in oil platforms that rely on pressure measurements (Jahn et al., 2008). In this case, the controller regulates the riser inlet pressure P_{in} by manipulating the riser choke valve z, but without lift-gas injection ($u_{gs,1} = u_{gs,2} = 0$). This characterizes a complex SISO (Single-Input, Single-Output) type of control due to the non-linearity associated with pressure tracking, mainly the gain variation. Further, the gas-lift choke valve configuration can compromise dynamical stability in the system, since they affect the gas-oil mass ratio of the flow and potentially lead to slugging flow regimes. To illustrate these phenomena, open-loop simulations are depicted in Figure 6 for different set points, namely for the cases $(u_{gs,1}, u_{gs,2}) = (0.0, 0.0)$ and $(u_{gs,1}, u_{gs,2}) = (0.05, 0.0)$. From Figure 6(b), it is evident that the same variation at each operating point implies a different gain variation, which tends to increase exponentially as the riser inlet pressure increases, characterizing a non-linear gain behavior. Notice also that a small opening variation of one gas-lift valve leads to an oscillatory regime in 6(a), where the system only manages to reach stability with a small riser choke opening $z \leq 0.1$. In this challenge, the general objective is to test the capacity of the online learning controller to track set points in oil and gas applications, using only one manipulated variable and without any prior information.

- **Super-Actuated Pressure Control (MISO)**: The gas-lift choke opening configuration is influential on whether the plant will undergo pressure oscillations or not. In the previous control problem, the plant was forced into a highly oscillatory regime to be suppressed by an anti-slug controller, which manipulates only the riser choke opening. Instead in this task, the anti-slug controller is allowed to manipulate the gas-lift choke openings ($u_{gs,1}$ and $u_{gs,2}$) but not the production choke (z), which stays fully open. This characterizes a super-actuated control for having more manipulated variables than control variables. Besides being easier to attain stability in the available setpoint range, due to the degrees of freedom, the design of a black-box controller is challenging for having to avoid unstable operating points, which might pose an issue since the model is previously unknown. In a sense, this task serves as a test whether or not the controller can avoid unstable settings and reach equilibrium for a wide variety of setpoints.

- **Coupled Well Control (MIMO)**: the objective is to control the two wells instead of the riser. Both wells are coupled by the presence of the manifold, which equates both well outlet pressures. The well production choke valves $u_{ch,i}$ are manipulated to control the bottom-hole pressure of each well, but the gas-lift choke openings are held fixed at $u_{gs,i} = 0.4,\ \forall i$. In real life, the measurement of the bottom-hole pressure is another problem by itself (Jahanshahi et al., 2012) due to the low reliability of the involved sensors. However, this can be mitigated by estimating the bottom-hole pressure (Jahanshahi et al., 2012). The challenge in this task emerges from the coupled multivariate aspect of the problem, which the previous tasks did not explore.

5. Experiments and Results

This section presents the experiments using the online learning controller for controlling the oil production plant with two wells and one riser, aiming at analyzing the controller capacity in following set-points and



(a) Unstable case, where $u_{gs,1} = 0.05$ and $u_{gs,2} = 0.00$

(b) Stable case, where $u_{gs,1} = 0.00$ and $u_{gs,2} = 0.00$

Figure 6: Open-loop variation of the riser choke opening z in the two wells, one riser system. Both well chokes are fully open.We induce a stable and an unstable case by fixing two distinct value sets for both gas-lift choke openings. The topmost subplot corresponds to the riser pressure, whereas the bottommost subplot corresponds to the choke openings for figures (a) and (b). The blue line is the riser production choke opening. The red and green lines are the gas-lift valve openings.

rejecting disturbances. We now showcase each of the control problems proposed in Section 4.4 individually.

5.1. Implementation

The plant model was implemented using *Jmodelica*, an open source dynamic system modeling language which has an interface with *Python 2.7*. We used the Python libraries *numpy* and *scipy* to implement both the Echo State Networks and the online learning controller. The results were displayed using the *matplotlib* library. For all three cases, we utilized a sampling time of 60 seconds, a common setting in oil and gas production plants.

5.2. Metrics and Experimental Setup

Hafner and Riedmiller (2011) define the metrics utilized in this work. They argue that a viable control metric is the average trajectory error e_T of the controller:

$$e_T = \frac{1}{N}\sum_{k=0}^{N} \|\widehat{\mathbf{y}}[k] - \mathbf{y}[k]\|_1 \tag{18}$$

where N is the total runtime in time steps, $\mathbf{y}[k]$ is the actual output and $\widehat{\mathbf{y}}$ is the predicted output. Actually, this error metric is related to the cost function in the predictive control strategies (Camacho and Bordons, 1999), which uses the 2-norm instead of the 1-norm. Notice that the average error e_T increases the longer the system takes to stabilize at a set-point, being an indirect evaluation of convergence time. Because this metric fails to capture the transient behavior in the controller, such as oscillations, we propose the mean variation of the control action as an alternative metric:

$$\Delta U = \sum_{k=0}^{N} \|\mathbf{u}[k] - \mathbf{u}[k-1]\|_1 \tag{19}$$

$$\mathbf{u}[-1] = \mathbf{u_0} \tag{20}$$

The variation ΔU increases if the control action differs between consecutive time-steps, for instance when the system undergoes oscillations or during transients. In process control applications, we want the behavior of the system to be as conservative as possible while respecting process time constraints. So, the total control action variation should be as low as possible.

In all experiments, the reference signal $\widehat{\mathbf{y}}$ is composed of three parts. Each part has a duration of 2000 time steps, roughly 0.6 hours, according to the following rules:

- **training:** we designed the first part so that the controller learns the plant dynamics in different, gradually varying, set points covering the whole output space.
- **validation:** the same set-point signal from the training phase is used to verify if learning converged, i.e., a controller with low trajectory error e_T and low mean variation of the control action ΔU.
- **generalization:** the third part uses a random stair signal as set-point to test the performance of the

controller over unseen plant output spaces. Both the amplitude and the frequency of the signal randomly vary. In the literature, this kind of signal is known as APRBS (Amplitude Modulated Pseudo-Random Binary Signal) (Nelles, 2001), where a PRBS is modulated over a random amplitude, achieving, in the end, a random stair signal composed of steps with random duration.

In control, better results are expected if the deviation from one set-point to the next is small, as advocated by the linear systems approximation theory (Chen, 1998). So we should expect poor control performance when the distance between one set-point and the next is high.

The mean trajectory error e_T and the total control variance ΔU are computed, for each part of the simulation, which provides insights into how the controller has progressed. At each time k, we also plot the metrics e_T and ΔU over the 100 previous time steps, which shows how the controller continuously progresses over time.

5.3. On Parameter Selection

In total, we tuned seven controller parameters according to either a grid search procedure or empirically analyzing the behavior of the controller and its parameters: for the ESN (reservoir), the leak rate γ, the spectral radius ρ, the input scaling f_i^r and bias scaling f_b^r; for the RLS training of the output layer, the forgetting factor λ and α; and for the controller (externally to the network), the prediction time step δ.

The forgetting factor λ should be small enough so that the RLS is sensitive to changes in the model, but large enough so that the covariance matrix $\mathbf{P}$ does not degenerate. The value $\lambda = (1 - 10^{-6})$ was chosen by trial and error. We have analyzed the behavior of the main diagonal of the covariance matrix over time, and chosen a value for λ sufficiently high to guarantee that $\mathbf{P}$ does not diverge, which otherwise causes numerical instability, while maintaining the "forgetfulness" of past data by the RLS.

When prior knowledge on the weights and covariance matrix is unavailable, we should select a small value for α to induce high variance in the diagonal of $\mathbf{P}$, which in turn expresses a high degree of uncertainty. Notice also that a small value for α leads to a faster convergence rate of the RLS algorithm (Nelles, 2001). For the simulations herein α was set to 10^{-3}.

To select the number of neurons for the experiments in this work, we tested the controller using different values for the number of neurons, and obtained successful results with values as low as 50. However, the performance for randomly generated reservoirs of size 50 have a high variance when compared to reservoirs with more neurons. In all cases, 300 neurons proved to be enough to perform all tasks, considering learning and control of the system. More neurons require more computational power to calculate the control action, so it is desirable to keep the number of neurons sufficiently low for the controller to learn the system.

For each experiment, Table 1 displays the rest of the parameters: the SISO case corresponds to the experiments where the controlled variable is the riser inlet pressure P_{in}, and the manipulated variable is riser choke opening z; the MISO case refers to the P_{in} control using both gas-lift valves $u_{gs,1}$ and $u_{gs,2}$; and the MIMO case refers to the control of the downhole pressures $P_{bh,i}$ of both wells ($i = 1, 2$) by simultaneously manipulating the production choke valves $u_{ch,i}$ of each well. Before doing a grid search, the reservoir's weights are randomly initialized from the normal distribution and left fixed for the different sweeps of other parameters. Borrowing from results shown in Waegeman et al. (2012), we choose to keep the internal weight matrix fully dense.

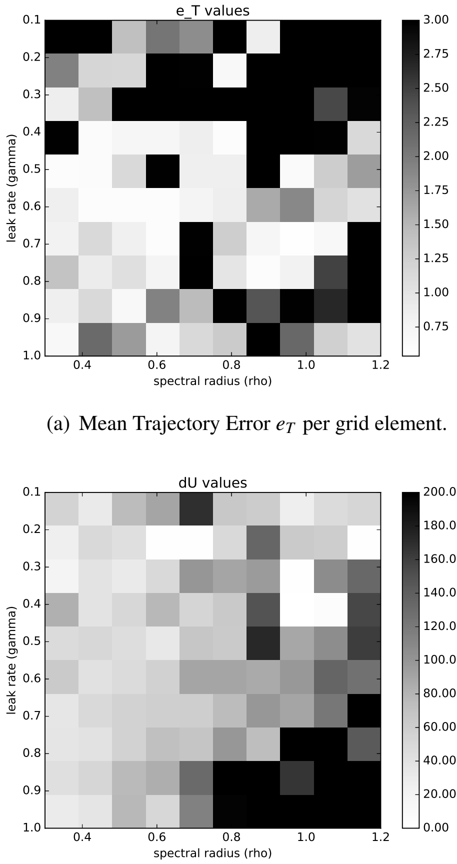
Table 1: Parameters values selected for the experiments.

Parameter	SISO	MISO	MIMO
γ: Leak Rate	0.2	0.5	0.2
ρ: Spectral Radius of $\mathbf{W_r^r}$	1.0	1.0	1.0
δ: Prediction Time steps	2	10	5
f_i^r: Scaling Factor of $\mathbf{W_i^r}$	0.3	0.3	0.3
f_b^r: Scaling Factor of $\mathbf{W_b^r}$	0.2	0.1	0.1

Lower values for f_i^r are desirable since the controller behaves less aggressively. Also, the system behaves more like a linear system when the value of f_i^r is small, since the input has small effect on the network.

The controller was not able to track set points without the presence of bias ($f_b^r = 0$) for the ESN, which shows that the task at hand is significantly complex and nonlinear. An empirical analysis of bias values revealed that $f_b^r = 0.2$ produces better results for the SISO case, whereas a bias value $f_b^r = 0.1$ was best for the super-actuated and coupled well cases.

The SISO and MIMO applications were given a leak rate of 0.2, for having fast dynamics, so there is no retardation of the controller dynamics related to the plant, and a spectral radius of 1.0, which ensure a long-lasting memory for the reservoir (slow dynamics are present). The only parameters left to decide are ρ and γ for the MISO case, and δ for all three cases.



(a) Mean Trajectory Error e_T per grid element.

(b) Total Control Variation ΔU per grid element.

Figure 7: Color Grid for e_T (a) and ΔU (b) for different values of γ (leak rate, y axis) and ρ (spectral radius, x axis). Experiment done for the MISO case with the metrics evaluated during validation and generalization stages (last 4,000 time steps).

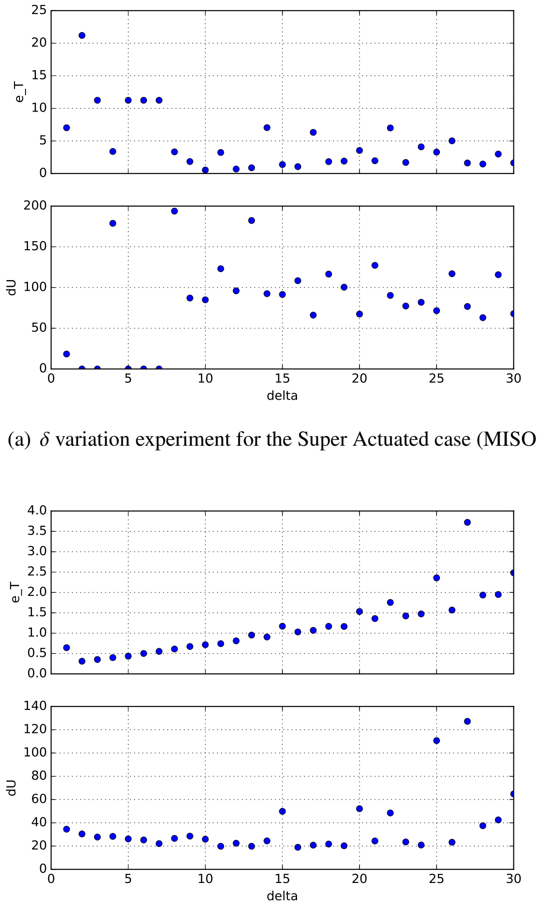
For the MISO case, the leak rate γ and spectral radius ρ were chosen by performing a grid search for $0.3 \leq \rho \leq 1.2$ and $0.1 \leq \gamma \leq 1$ on discrete intervals of 0.1, considering both ΔU and e_T evaluated during the last 4,000 time steps (validation and generalization phases) after the initial training phase of 2,000 time steps. Other parameters were fixed as shown in the Table 1. The results of this batch of experiments are presented in Figure 7, where black corresponds to high errors and white to low errors. For visualization purposes, we cut off points where $e_T > 3$ and $\Delta U > 200$ from the plot, where the performance was subpar, to better visualize the points that performed good in terms of the two metrics. From the plot, we can notice that the optimal combination of leak rate and spectral radius for the metric e_T (where ΔU can be up to 100) is located at $\gamma = 0.5$ and $\rho = 1.0$ (where $e_T < 0.6$), while other parameter combinations have led to more oscillatory behaviors in our analysis – represented by grey and black areas in the total control variation plot in Figure 7(b). Another view on the plot of ΔU is that, in general, as ρ gets higher, the dynamics of the ESN is enriched with more nonlinearity and short-term memory, causing the controller to behave more aggressively as well (darker areas of the plot). However, the interplay between the spectral radius and the leak rate in this application is not so evident, as the best blend of parameters takes place at $\gamma = 0.5$ and $\rho = 1.0$, suggesting a nonlinear relationship between parameters for the best learning and control performance.

In addition, for $\gamma < 0.5$, the ESN state dynamics slows down too much for the controller to effectively control the system – reflected by the top darker area with high error in Figure 7(a). We have also noticed that when $\gamma > 0.7$, the controller demonstrates behavior too aggressive to properly stabilize the plant – verified by the darker area with high ΔU in Figure 7(b). Although the bottom left sections of the plots in Figure 7 (where $0.7 \leq \gamma \leq 0.6$) have a greyish aspect suggesting reasonable control performance, we have noticed that for these cases the covariance matrix $\mathbf{P}$ has not decreased over time, i.e., the output weights learned by the RLS method have high associated uncertainty (given by $\mathbf{P}$) in the premature convergence of the RLS training. As a result, the norm for $\mathbf{W_r^o}$ goes up over time, bringing about numerical instability for long simulations.

In order to choose the most suitable value for the prediction time step δ, as seen in Table 1, we evaluated e_T and ΔU once again for the validation and generalization stages of the simulation during 4,000 time steps, for values of δ varying from 1 to 30, while keeping fixed the ESN weights and setting other parameters as defined in Table 1. Figure 8 depicts the results of these experiments for the MISO and MIMO cases. Note that the time step delay δ is the sole degree of freedom of the online learning control existing outside the ESN structure and adaptive RLS training blocks (Figure 2). Some remarks on the experiments follow for the following cases:

- SISO Case: the experiment was not very informative and is not shown here. The value with minimum e_T was $\delta = 2$.

- MISO Case: as shown in Figure 8(a), a subpar performance for the Super-Actuated control is obtained when $\delta < 9$, either with the system failing to learn a model at all (cases where $\Delta U = 0$



(a) δ variation experiment for the Super Actuated case (MISO).

(b) δ variation experiment for the Coupled Well Control case (MIMO).

Figure 8: Plot for e_T (y axis of topmost plot) and ΔU (y axis of bottom plot) evaluated during the last 4,000 time steps of the simulation for different values of δ (x axis).

imply a constant control action, where no learning or control is taking place), or when the control network outputs a nonconservative control action ($\Delta U > 150$) while still having undesirable performance ($e_T > 3.0$). For $\delta \geq 9$, the values for e_T and ΔU are more randomized and harder to predict. The best performance happens when $\delta = 10$, in terms of e_T, hence its choice in Table 1.

- MIMO Case: Figure 8(b), however, shows that δ has a more predictable effect on the Coupled Well Control Problem: starting from $\delta = 2$, the mean trajectory error e_T gets higher with δ. For $0 < \delta < 10$, the total control variation ΔU remains within an acceptable range of $[20, 40]$. For the controllers where $e_T \leq 0.5$, $\delta = 5$ induced the smallest ΔU.

5.4. *SISO Riser Inlet Pressure Tracking*

The experiments herein concern the first problem described in Section 4.4, the tracking of the riser inlet pressure P_{in} by manipulating only the riser choke valve. The design of a linear controller for this problem is particularly hard, since the static gain varies significantly per operating point (see Figure 6(b)). Put another way, a linear controller designed for an operating point can become unsuitable for another. No gas-lift is injected into the wells $((u_{gs,1}, u_{gs,2}) = (0, 0))$.

Before computing the control action, we scale the feasible range of the inlet pressure P_{in} from $[88.8, 95.0]$ bar to $[-1, 1]$, for better numerical conditioning, whereby 88.8 bar is the riser inlet pressure for a fully open choke, and 95.0 bar is the pressure when the choke is about 10% open. The control action is defined within the set $[0.1, 1.0]$ to prevent the well from being shut in. We then scaled the control action to $[-1, 1]$, to be in the same range as the controlled variable.

Figure 9 shows the results of the stable SISO P_{in} control experiment. In the first subplot, the red dashed line is the reference signal P_{in} and the blue solid line corresponds to the plant output signal. After approximately 1000 time steps, the Echo State Network converged[1] and was able to thoroughly learn the system operating points. The ESN learning led the controller to track the setpoints afterward without the presence of oscillations. We see this behavior in the first plot of Figure 9: the training and validation part of the simulation consist of the same tracking signal, but the controller succeeds in the validation part due to information obtained during the first 2000 time steps. The third part of the simulation (from 4000 to 6000 times steps) was designed to have low pressure set-points, whereas high pressures were set for the first two parts of the simulation (from 0 to 4000 time steps), as shown on the top plot of Figure 9. Within the third part of the simulation, large changes in the riser choke opening imply small changes in the riser inlet pressure. The top two plots of Figure 9 show this behavior. Notice that the controller can track the pressure profile within the third part, even though tracking was not consistently kept at low pressures in the first two parts of the simulation.

The third and fourth plots of Figure 9 present respectively the mean trajectory error e_T and the total control variation ΔU of the 100 previous time steps, at each time step[2]. Error peaks correspond to system saturation and oscillations (e.g., at time step 1000, e_T has a

[1] The Echo State Network converges when the weights of the output layer stop changing drastically.

[2] For time steps $k < 100$, the missing samples are assumed zero.

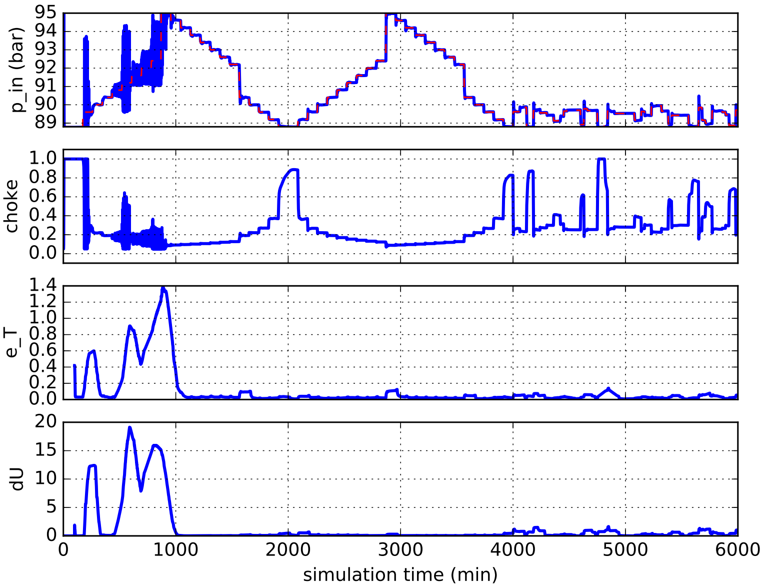


Figure 9: Results for the stable SISO case with z (riser choke opening) as plant input and P_{in} (riser inlet pressure) as plant output, where $u_{gs,1} = u_{gs,2} = 0$. The top most plot consists of the controlled variable P_{in} (full line), and the desired P_{in} value (dashed line). The second plot is the manipulated variable. The third plot gives the mean trajectory error e_T over the 100 previous time steps at each time step. The fourth plot is the total control variation ΔU over the 100 previous time steps, at each time step.

peak when P_{in} reaches its maximum value). The error in the first simulation part is higher than in the second, having a high magnitude before and a low magnitude after the ESN converged. Since the control action oscillates heavily, one expects that ΔU had higher values in the first 2000 time steps. The third part demonstrates the ESN learning ability: despite a random tracking trajectory leading to unexplored operation points, the error induced by the controller was still small. The magnitude of ΔU at the third part is also higher than the second, but this behavior is attributed to the distance between each control action to follow the desired set points at steady-state.

Table 2: Results for the mean trajectory error and maximum control variation of the stable SISO P_{in} control manipulating the production choke z.

	time steps		
	0-2000	2000-4000	4000-6000
e_T	0.28	0.028	0.037
ΔU	74.99	2.76	9.92

Table 2 gives the mean trajectory error e_T for each simulation part, as defined in Eq. (18), and likewise the control variation ΔU as defined in Eq. (19). These metrics corroborate the results shown in Figure 9.

We also have measured the time it took to compute the control action at each iteration. The average computation time was of 0.0042 seconds, the standard deviation was of 0.0032, and the worst case time was of 0.036 seconds.

The same control strategy was applied to the case presented in Figure 6(a), where the only difference from the system in this application is that $(u_{gs,1}, u_{gs,2}) = (0.05, 0)$, with one gas-lift valve fully closed and the other opened only at 5%. The controller has failed to stabilize the plant in that instance, but this might be due to the impossibility of reaching the proposed pressure set-points in a stable way using only the production choke as the manipulated variable. The difference in results shows that the gas-lift valves are essential in defining the system dynamics, making them important for pressure tracking. We tackle this approach in the next section.

5.5. Super-Actuated Control

This section considers a controller with two manipulated variables, namely the gas-lift chokes $(u_{gs,1}, u_{gs,2})$ while holding the riser choke fully open ($z = 1$) to sustain full oil production. The goal is to track a reference signal for the riser inlet pressure P_{in}, a control problem in the MISO class.

Multiple solutions for reaching a given set-point emerge from the additional degrees of freedom, posing a challenge to learn the ESN-based inverse model. For example, a pressure of 110 bar can be achieved with multiple combinations of $(u_{gs,1}, u_{gs,2})$. As the input-output relation is not bijective, the inverse model is not invertible and the controller must find an approximation that works. Another challenge to control design stems from the slow system dynamics, since a change on the gas-lift choke opening can take considerable time to influence the riser inlet pressure.

We scaled the range for the riser pipeline pressure $[100, 120]$ to $[-1, 1]$, in which 100 bar is the lowest allowed pressure and 120 bar is the upper limit. The gas-lift choke opening $[0, 1]$ was likewise scaled to $[-1, 1]$ for both valves.

The third column of Table 1 presents the parameters for this experiment.

Figure 10 depicts the results for the Super-Actuated control. Notice that the controller asymptotically stabilizes the plant for all set-points once the RLS algorithm converges, approximately after 400 time steps, even considering that, of all applications, the Super-Actuated case has the most complex dynamics, since the gas-lift valve is in the well annulus, and P_{in} is measured at the riser inlet (see Figure 5). These results show that the controller can effectively learn the inverse plant model and thus control the plant, despite the additional degrees of freedom, and avoid potential oscillatory regimes. The controller also efficiently tracked the riser inlet pressure in the third part (generalization stage) of the simulation, where the set-points are set at random, even though abrupt set-point changes make it difficult to generate smooth dynamical responses. In the second subplot, the blue line represents $u_{gs,1}$ and the green line $u_{gs,2}$.

The third and fourth plots of Figure 10 are consistent with the previous experiments as measured by e_T and ΔU. The performance in the second and third simulation parts, namely validation and generalization, respectively, are better than in the first learning part, which is devoid of prior knowledge of the inverse model. The larger set-point changes are the cause of the higher error in the third part.

Because there is no feedback from the outputs to the state of the ESN ($\mathbf{W_o^r} = 0$), the control signals $u_{gs,1}$ and $u_{s,2}$ are fully independent and computed by separate weight combinations of network states. Since both control signals are affecting the riser inlet pressure P_{in}, we can see each control action as a disturbance to one another. We can see this control structure as two separate controllers adjusting the same controlled variable by using a different manipulated variable each. The controllers will interfere with each other, rejecting disturbances in the form of multivariate coupling. Table 3 presents the metrics e_T and ΔU for each of the three parts of the simulation. These metrics endorse the conclusions drawn above. For instance, the control variation is higher in the third simulation part, which is a result of the control action required to track a randomly changing set point.

Table 3: Results for the mean trajectory error and maximum control variation of the P_{in} control, manipulating the well gas-lift valves $u_{gs,1}$ and $u_{gs,2}$.

	time steps		
	0-2000	2000-4000	4000-6000
e_T	1.57	0.33	1.06
ΔU	101.2	34.89	119.99

Measuring the control action computation time at each time step of the simulation, we have obtained an average of 0.0058 seconds, a standard deviation of 0.0041 and a worst-case computation time of 0.046 seconds.

We applied the Least Mean Squares (LMS) algorithm (Nelles, 2001) instead of RLS for riser control, however the resulting controller did not manage to attain a satisfactory performance. Alternatively, an Extreme Learning Machine (ELM) (Huang et al., 2006) and a Radial Basis Function (RBF) network (Nelles, 2001) were implemented for the plant inverse model as a means to compare with the ESN-based controller. As initially proposed by Waegeman et al. (2012), the online learning control framework can use machine learning models other than reservoirs.

The ELM operates according with the following equation:

$$\mathbf{h}[k] = \tanh(\mathbf{W_{in}}\mathbf{u}[k] + \mathbf{b}) \tag{21}$$

$$\mathbf{y}[k] = \mathbf{W_o}\mathbf{h}[k] \tag{22}$$

where $\mathbf{u}[k]$ and $\mathbf{y}[k]$ correspond to the network input (plant output) and network output (plant input) respectively. The vector $\mathbf{h}[k]$ has the values of the hidden layer units. Unlike the ESN which has an internal dynamics given by the network state, the ELM is static and therefore yields the output depending only on the current input. Akin to an ESN, only the output weights $\mathbf{W_o}$ of the ELM are trained using RLS, while the weights $\mathbf{W_{in}}$ and $\mathbf{b}$ are randomly initialized as suggested in the literature

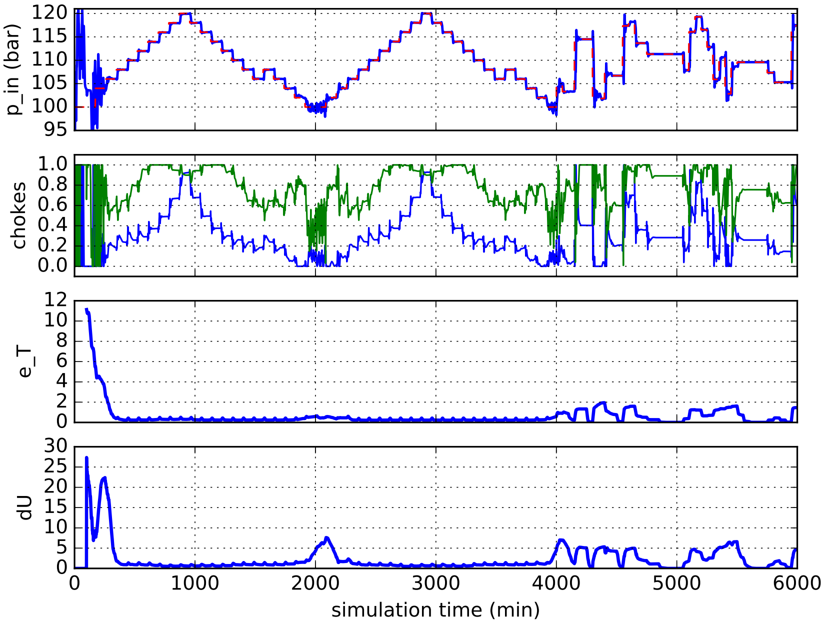


Figure 10: Results for MISO control with $u_{gs,1}$ and $u_{gs,2}$ (gas-lift choke openings) as the plant input and P_{in} as the plant output, with the riser choke fully open at $z = 1$. The first plot consists on the controlled variable P_{in} (full line), and the desired P_{in} value (dashed line). The second plot gives the manipulated variables, i.e. both gas-lift choke opening. The third plot is the mean trajectory error e_T over the 100 previous time steps at each time step. The fourth plot is the total control variation ΔU over the 100 previous time steps, at each time step.

(Huang et al., 2006), in this case, using a standard normal distribution $\mathcal{N}(0, 1)$. For comparison purposes, the vector $\mathbf{h}[k]$ has the same dimension as the reservoir in the ESN, i.e., $N = 300$ units.

A Radial Basis Function is roughly defined as a function $f(\mathbf{x}) = g(||\mathbf{x} - \mathbf{c}_i||)$ that depends on the Euclidean norm of the difference between an input $\mathbf{x}$ and a center $\mathbf{c}_i$. Each radial basis function possesses a center $\mathbf{c}_i$ and radius r_i. In this work, we adopt a Gaussian-like function as RBF, defined as:

$$f(\mathbf{x}) = e^{-\left(\frac{1}{r_i}||\mathbf{x}-\mathbf{c}_i||\right)^2} \tag{23}$$

The RBF network combines multiple RBFs for the purpose of system identification. As each RBF influences locally in the identification, the centers must be well distributed. Since every variable is normalized before being input to the inverse model, the center $\mathbf{c}_i$ of each RBF is drawn from the uniform distribution $\sqcap(0, 1)$. For the sake of simplicity, instead of using the radius r_i, we implement the RBFs in terms of the parameter $w_i = \frac{1}{r_i}$, drawn randomly from $\sqcap(0.01, 1)$. The resulting RBF network operates with the following equations:

$$h_i = e^{-(w_i||\mathbf{x}-\mathbf{c}_i||)^2} \tag{24}$$

$$\mathbf{y} = \mathbf{W_o h} \tag{25}$$

where h_i is the i^{th} element from $\mathbf{h}$, which has size $N = 300$ in this experiment. As with the ESN and ELM, we train the output weights using RLS, leaving the remaining parameters static. The output weights $\mathbf{W_o}$ dictate the influence of each RBF in the resulting inverse model.

To compare the performance of the controllers defined by the ESN, the RBF network and ELM, thirty simulations were carried out for each inverse model, with varying internal weights for the respective network. Besides the previous experiments with a time-step delay $\delta = 10$, experiments were also performed for $\delta = 30$ whereby 10 simulations were run for each inverse model. The results are reported in Table 4 in terms of the mean and standard deviation (std) of the tracking error e_T.

From the results in Table 4, we can infer that the Echo State Network outperforms the ELM as well as the RBF network on average, with smaller mean for the tracking

Table 4: Comparison of ESN with the ELM and RBF inverse models, applied to the online-learning control of the riser inlet pressure (P_{in}) by manipulating the well gas-lift valves ($u_{gs,1}$ and $u_{gs,2}$). (a) Case for default settings, with time-step delay $\delta = 10$; (b) Case for $\delta = 30$, approximately simulating a lower sampling rate of the plant.

(a) $\delta = 10$

	time steps		
Mean	0-2000	2000-4000	4000-6000
$e_T(ELM)$	9.16	8.33	8.54
$e_T(ESN)$	**5.68**	**4.73**	**5.09**
$e_T(RBF)$	6.22	4.89	5.63

	time steps		
St. Dev.	0-2000	2000-4000	4000-6000
$e_T(ELM)$	6.33	6.76	6.91
$e_T(ESN)$	4.96	4.34	**4.19**
$e_T(RBF)$	**4.21**	**3.94**	4.54

(b) $\delta = 30$

	time steps		
Mean	0-2000	2000-4000	4000-6000
$e_T(ELM)$	11.78	10.72	10.22
$e_T(ESN)$	**7.75**	**5.33**	**6.06**
$e_T(RBF)$	9.95	8.96	9.19

	time steps		
St. Dev.	0-2000	2000-4000	4000-6000
$e_T(ELM)$	**4.65**	**5.29**	6.05
$e_T(ESN)$	6.48	6.18	**6.02**
$e_T(RBF)$	6.00	6.55	6.52

error e_T. In the case of $\delta = 10$ shown in Table 4(a), the mean error values produced by the RBF network are closer to the ESN than ELM; ESN yielded a lower variance on e_T than ELM but slightly higher than the RBF network in the first two parts of the simulation. As the time-step delay δ becomes greater, that is, for tasks requiring even more memory[3] (Waegeman et al., 2012), we expect that the ESN performance surpasses the other methods more strongly as the fading memory effect will be essential for the online learning control. This tendency is observed in Table 4(b), where $\delta = 30$, which shows that the deterioration in performance of the static models (ELM and RBF model) is greater than that of the ESN when dynamics becomes more important. Considering the last column (time steps 4000-6000), while in (a) the RBF model was about 8% worse than the ESN, in (b), the RBF model is about 51% worse on average than the ESN. Considering that weights are randomly initialized, the results indicate that a good ESN controller is more easily obtained for the slow dynamics setup than a controller based on ELM and RBF networks.

5.6. Coupled Well Control

This experiment concerns the manipulation of the production choke $u_{ch,i}$ of both wells to control their bottom-hole pressure $P_{bh,i}$. For being a MIMO (Multiple Input, Multiple Output) plant, this experiment is the most complex in terms of learning. For instance, some desired set-point combination might not be feasible due to the plant physics, which imposes operational constraints. Such constraints make it harder for the controller to calculate choke openings that track the desired bottom hole pressures.

We scale the well bottom-hole pressure from $[170, 220]$, which are the minimum and maximum bottom-hole pressure that the wells can reach, to $[-1, 1]$. Likewise, the production choke opening is scaled from $[0.1, 1]$ to $[-1, 1]$ for both wells, with 0.1 being the lowest allowable choke opening.

The fourth column of Table 1 shows the parameters utilized in this experiment. Figure 11 depicts the set-point tracking experiment. It took 500 time steps for the RLS to converge and the inverse model to be learned. Once the learning has converged, the controller tracked all the proposed set-points without oscillations, which is remarkable given the lack of prior knowledge and considering variable coupling. Table 5 shows the total control variation and mean trajectory error over the three phases of the simulation. These metrics corroborate the behavior shown in Figure 11. Notice that the mean trajectory error e_T is high during the learning phase, very low for the well behaved trajectory tracking part during validation, and slightly higher than the previous period when the set points are defined randomly in the generalization phase.

Table 5: Results for the mean trajectory error and maximum control variation of the MIMO coupled wells control, without disturbances.

	time steps		
	0-2000	2000-4000	4000-6000
e_T	2.05	0.33	0.69
ΔU	27.3	3.77	25.72

With the aim of assessing the robustness of the online learning controller, we applied disturbances to the gas-lift source pressure of well 2 ($P_{gs,2}$). Figure 12 shows the results of this experiment. The bottom-most plot of the figure shows the gas-lift disturbance, which consists

[3] For example, if the sampling rate of the plant would be lower than 1 measurement per minute.

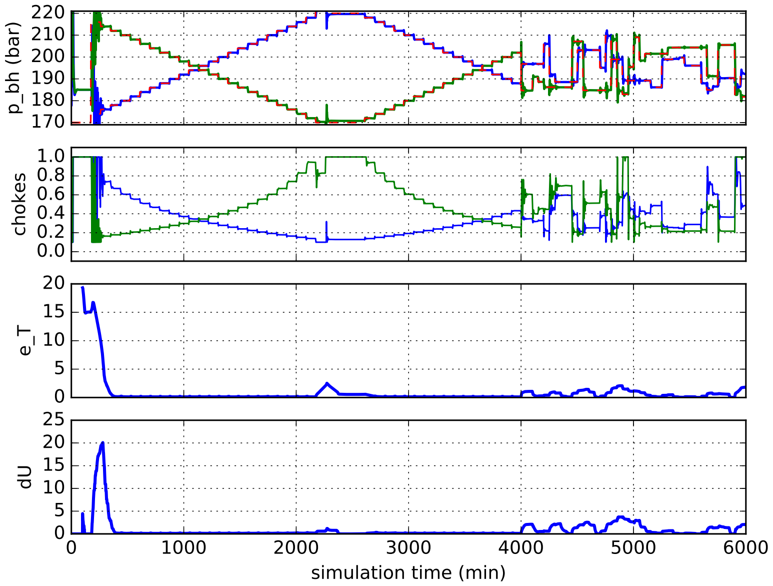


Figure 11: Results for MIMO control with $u_{ch,1}$ and $u_{ch,2}$ (production choke openings) as input and $P_{bh,1}$ and $P_{bh,2}$ (bottom hole pressures) as outputs. The first plot consists on the controlled variable P_{bh} (full line), and the desired P_{bh} value for both wells (dashed line). The second plot gives the manipulated variables, both gas-lift choke openings. The third plot shows the mean trajectory error e_T over the 100 previous time steps, at each time step. The fourth plot is the total control variation over the 100 previous time steps, at each time step.

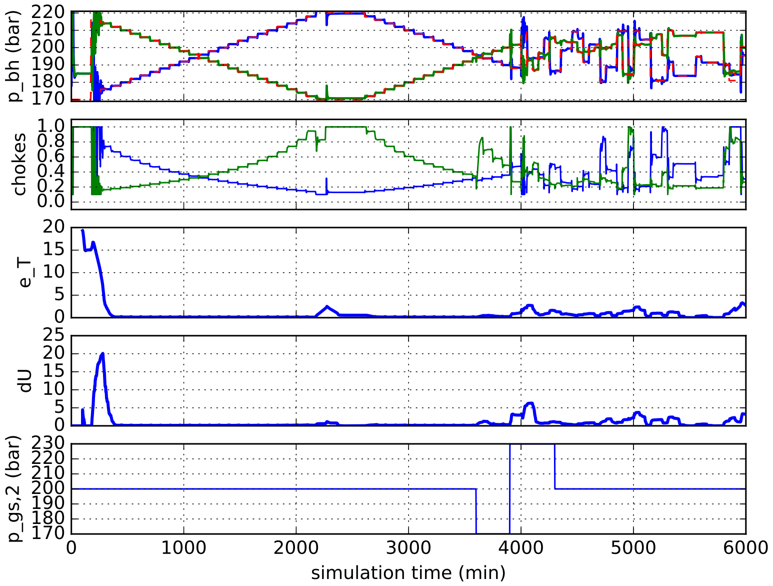


Figure 12: Results for MIMO control when a disturbance in $P_{gs,2}$ is applied. The production choke openings $u_{ch,1}$ and $u_{ch,2}$ are the plant inputs, whereas $P_{bh,1}$ and $P_{bh,2}$ are the plant outputs, and $z = 1$ and $u_{gs,1} = u_{gs,2} = 0.4$. The first plot consists of the controlled variables $P_{bh,1}$ and $P_{bh,2}$ (full line), and the desired P_{bh} value for both wells (dashed line). The second plot gives the manipulated variables, both gas-lift choke openings. The third plot shows the mean trajectory error e_T over the 100 previous time steps, at each time step. Likewise, the fourth plot gives the total control variation over the 100 previous time steps, at each time step. The fifth plot depicts the step disturbance in the gas-lift source pressure $P_{gs,2}$.

of an abrupt parametric change in the model occurring at a certain point in time (i.e., step disturbance). The step disturbances for the gas-lift source pressure $P_{gs,2}$ happen at times t = 3600 min, t = 3900 min and t = 4300 min (see Figure 12).

A 30 bar disturbance is usually considered large in the context of the application. Even though the RLS algorithm converged, which leads the learning algorithm to be less sensitive to changes in the plant, the controller still managed to compensate for the parametric change in the model. This means that the online learning controller adapts to model change with little performance degradation. Table 6 shows the result of this experiment with disturbance applied. We apply the disturbance at the end of the second phase and the beginning of the third phase. Notice that the results did not change by a large margin in comparison to the case without disturbance (see Table 5).

Table 6: Results for the mean trajectory error and maximum control variation of the MIMO coupled wells control, with disturbances.

	time steps		
	0-2000	2000-4000	4000-6000
e_T	2.05	0.413	1.042
ΔU	27.32	8.19	28.36

We measured, for the undisturbed case, the control action computation time at each time step. The average was 0.0058 seconds, the standard deviation was 0.0041 seconds, and the worst case time was of 0.052 seconds.

6. Conclusion

This work has shown that online learning control is able to control complex simulated dynamical systems in diverse scenarios without prior information. The ESN-based controller was used to control an oil production platform consisting of one riser and two wells connected by a manifold, where complex dynamics were present such as steady-state gain variation, multiple degrees of freedom, and coupled multi-variable control. Three control problems (SISO, MISO, and MIMO) involving different plant variables were tackled, extending substantially previous work. The ESN-based controller learned each different input-output dynamical behavior and performed set-point tracking and disturbance rejection, either explicitly or implicitly in a multivariate coupling scenario. Despite the remarkable challenge of controlling a plant without a model, the online learning framework has succeeded in all three proposed cases. Furthermore, the application of the learning controller using a single ESN architecture has proved effective also for multivariate problems as shown here, even when coupled variables were involved. As the worst case computation time for the control action obtained in all three simulation runs indicates, we can reliably use the online learning controller in applications where the sampling time is larger than 0.1 seconds given the utilized hardware conditions. For the oil and gas plant considered in this work, computation time is a non-issue, since the sampling time is of 1 minute, though it may pose a problem for faster applications.

As future work, we plan to compare the current framework with other control approaches in the literature as well as to devise methods for significantly reducing the initial transients when controlling a plant, an actual hindrance for real-world applications. For instance, we can train a second ESN model beforehand as a proxy forward model (Antonelo et al., 2017, 2007) of the plant to be controlled, initializing the output layer of the ESN-based controller by running the initial steps of the control loop using the previously trained proxy model as a simulated plant instead of the real plant.

Acknowledgments

This research was funded in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) from grant number 88881.119526/2016-01.

References

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

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

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

Antonelo, E.A., Schrauwen, B., Campenhout, J.V., 2007. Generative modeling of autonomous robots and their environments using reservoir computing. Neural Processing Letters 26, 233–249. doi:http://dx.doi.org/10.1007/s11063-007-9054-9.

Astrom, K.J., Wittenmark, B., 1994. Adaptive Control. 2 ed., Dover Publications.

Bo, Y.C., Zhang, X., 2018. Online adaptive dynamic programming based on echo state networks for dissolved oxygen control. Applied Soft Computing 62, 830 – 839.

Camacho, E., Bordons, C., 1999. Model Predictive Control. Springer.

Campos, M., Takahashi, T., Ashikawa, F., Simoes, S., Stender, A., Meien, O., 2015. Advanced anti-slug control for offshore production plants. IFAC-PapersOnLine 48, 83 – 88.

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

Chen, M., Challita, U., Saad, W., Yin, C., Debbah, M., 2017a. Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. CoRR abs/1710.02913.

Chen, M., Mozaffari, M., Saad, W., Yin, C., Debbah, M., Hong, C.S., 2017b. Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. IEEE Journal on Selected Areas in Communications 35, 1046–1061.

Chen, M., Saad, W., Yin, C., 2018. Virtual reality over wireless networks: Quality-of-service model and learning-based resource management. IEEE Transactions on Communications 66, 5621–5635.

Graves, A., Jaitly, N., Mohamed, A.R., 2013. Hybrid speech recognition with deep bidirectional LSTM, in: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 273–278.

Hafner, R., Riedmiller, M., 2011. Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. Machine Learning 24, 137–169.

Hinaut, X., Dominey, P.F., 2012. On-line processing of grammatical structure using reservoir computing, in: Int. Conf. on Artificial Neural Networks, pp. 596–603.

Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. Neurocomputing 70, 489 – 501.

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., 2002. Short term memory in echo state networks. Technical Report GMD Report 152. German National Research Center for Information Technology. URL: https://www.researchgate.net/publication/247514367_Short_Term_Memory_in_Echo_State_Networks.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. Science 304, 78–80.

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

Jahanshahi, E., 2013. Control Solutions for Multiphase Flow, Linear and Non-linear Approaches to Anti-slug Control. Ph.D. thesis. Norwegian University of Science and Technology. Trondheim, Norway.

Jahanshahi, E., Skogestad, S., 2011. Simplified dynamical models for control of severe slugging in multiphase risers. IFAC Proceedings Volumes 44, 1634–1639.

Jahanshahi, E., Skogestad, S., Grøtli, E.I., 2013. Anti-slug control experiments using nonlinear observers, in: American Control Conference, pp. 1056–1062.

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

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

Jordanou, J.P., Antonelo, E.A., Camponogara, E., de Aguiar, M.A., 2017. Recurrent neural network based control of an oil well, in: Brazilian Symposium on Intelligent Automation.

Jordanou, J.P., Camponogara, E., Antonelo, E.A., Aguiar, M.A.S., 2018. Nonlinear model predictive control of an oil well with echo state networks. IFAC-PapersOnLine 51, 13–18.

Lin, X., Yang, Z., Song, Y., 2009. Short-term stock price prediction based on echo state networks. Expert Systems with Applications 36, 7313–7317.

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

Lungu, M., Lungu, R., 2018. Neural network based adaptive control of airplane's lateral-directional motion during final approach phase of landing. Engineering Applications of Artificial Intelligence 74, 322–335.

Maass, W., Natschläger, T., Markram, H., 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14, 2531–2560.

Mozer, M.C., 1995. A focused backpropagation algorithm for temporal pattern recognition, in: Chauvin, Y., Rumelhart, D.E. (Eds.), Backpropagation: Theory, Architectures, and Applications. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp. 137–169.

Münker, T., Nelles, O., 2018. Nonlinear system identification with regularized local fir model networks. Engineering Applications of Artificial Intelligence 67, 345–354.

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

de Oliveira, V., Jäschke, J., Skogestad, S., 2015. An autonomous approach for driving systems towards their limit: an intelligent adaptive anti-slug control system for production maximization. IFAC-PapersOnLine 48, 104 – 111.

Ozturk, M.C., Xu, D., Príncipe, J.C., 2007. Analysis and design of echo state networks. Neural Computation 19, 111–138.

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

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

Stasiak, M., Pagano, D., Plucenio, A., 2012. A new discrete slug-flow controller for production pipeline risers. IFAC Proceedings Volumes 45, 122–127. 1st IFAC Workshop on Automatic Control in Offshore Oil and Gas Production.

Verstraeten, D., Dambre, J., Dutoit, X., Schrauwen, B., 2010. Memory versus non-linearity in reservoirs, in: Int. Joint Conference on Neural Networks, pp. 18–23.

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

Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D., 2007. An experimental unification of reservoir computing methods. Neural Networks 20, 391–403.

Waegeman, T., Hermans, M., Schrauwen, B., 2013. MACOP modular architecture with control primitives. Frontiers in Computational Neuroscience 7.

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

Werbos, P., 1990. Backpropagation through time: what it does and how to do it. IEEE Proceedings 78, 1550–1560.

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

AppendixA. Well Model Details

All the algebraic equations that depict the Well model's dynamics are displayed in this appendix. The flow equations are as follows:

$$\omega_{G_in} = K_{gs}u_2\sqrt{\rho_{G,in}\max(P_{gs} - P_{at})} \tag{A.1}$$

$$\omega_{G_inj} = K_{inj}\sqrt{\rho_{G,ab}\max(P_{ab} - P_{tb})} \tag{A.2}$$

$$\omega_{out} = K_{pr}u_1\sqrt{\rho_{mix,t}\max(P_{tt} - P_0)} \tag{A.3}$$

$$\omega_{res} = PI\max(P_{res} - P_{bh}) \tag{A.4}$$

$$\omega_{L,res} = (1 - \alpha^m_{G,b})\omega_{res} \tag{A.5}$$

$$\omega_{G,res} = \alpha^m_{G,b}\omega_{res} \tag{A.6}$$

$$\omega_{L,out} = (1 - \alpha^m_{G,t})\omega_{out} \tag{A.7}$$

$$\omega_{G,out} = \alpha^m_{G,t}\omega_{out} \tag{A.8}$$

where K_x are tuning parameters to adequate the model to a real world application. The pressures P_{gs}, P_{at}, P_{ab}, P_{tb} and P_{tt} are the gas-lift source pressure, the pressure at the top of the annulus, the pressure at the annulus bottom, the pressure at the tubing bottom, and the pressure at the top of the tubing, respectively. The pressure P_0 is the outlet pressure, normally of the manifold, P_{res} is the reservoir pressure and P_{bh} is the bottom-hole pressure. The variable u_1 is the wellhead choke valve opening and u_2 is the gas-lift choke valve opening. The variable $\alpha^m_{G,b}$ 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:

$$\alpha^m_{G,t} = \frac{(1 - \alpha_{L,t})\rho_{G,t}}{\alpha_{L,t}\rho_L + (1 - \alpha_{L,t}\rho_{G,t})} \tag{A.9}$$

$$\alpha_{L,t} = 2\overline{\alpha}_L - \alpha_{L,b} \tag{A.10}$$

$$\alpha_{L,b} = \frac{\omega_{L,res}\rho_{G,tb}}{\omega_{L,res}\rho_{G,tb} + (\omega_{G,inj} + \omega G, res)\rho_L} \tag{A.11}$$

$$\overline{\alpha}_L = \frac{m_{L,tb} - \rho_L V_{bh}}{V_t\rho_L} \tag{A.12}$$

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

$$\rho_{G,ab} = \frac{P_{ab}M_G}{RT_a} \tag{A.13}$$

$$\rho_{G,in} = \frac{P_{gs}M_G}{RT_a} \tag{A.14}$$

$$\rho_{G,t} = \frac{m_{G,tb}}{V_t + V_{bh} - m_{L,bh}/\rho_L} \tag{A.15}$$

$$\overline{\rho}_{mix} = \frac{m_{G,tb} + m_{L,tb} - \rho_L V_{bh}}{V_t} \tag{A.16}$$

$$\rho_{G,tb} = \frac{P_{tb}M_G}{RT_t} \tag{A.17}$$

$$\rho_{mix,t} = \alpha_{L,t}\rho_L + (1 - \alpha_{L,t})\rho_{G,t} \tag{A.18}$$

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:

$$P_{at} = \frac{RT_a m_{G,a}}{M_G V_a} \tag{A.19}$$

$$P_{ab} = P_{at} + \frac{m_{g,a}gL_a}{V_a} \tag{A.20}$$

$$P_{tt} = \frac{\rho_{G,t}RT_t}{M_G} \tag{A.21}$$

$$P_{tb} = P_{tt} + \overline{\rho}_{mix}gL_t + F_t \tag{A.22}$$

$$P_{bh} = P_{tb} + F_b + \rho_L g L_{bh} \tag{A.23}$$

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:

$$F_t = \frac{\lambda_b\rho_L\overline{U}^2_{l,b}L_{bh}}{2D_t} \tag{A.24}$$

$$\frac{1}{\sqrt{\lambda_t}} = -1.8\log_{10}\left(\left(\frac{\epsilon}{3.7D_t}\right)^{1.11} + \frac{6.9}{Re_t}\right) \tag{A.25}$$

$$Re_t = \frac{\overline{\rho}_{mix}\overline{U}_{m,t}D_t}{\mu} \tag{A.26}$$

$$\overline{U}_{m,t} = \overline{U}_{sl,t} + \overline{U}_{sg,t} \tag{A.27}$$

$$\overline{U}_{sg,t} = \frac{4(\omega_{G,in} + \alpha^m_{G,b}\overline{\omega}_{res})}{\rho_{G,t}\pi D_t^2} \tag{A.28}$$

The above equations are derived from Haaland's solution to the Colebrook-White equation (1983) for calculating 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.

AppendixB. Pipeline-Riser Model Details

This appendix contains all the equations utilized in the pipeline-riser model present in this work.

Calculation of the riser outlet flow:

$$
\begin{aligned}
\omega_{out} &= K_{pc} z \sqrt{\rho_t \max(P_r - P_0)} \\
\omega_{L,out} &= \alpha^m_{L,t} \omega_{out} \\
\omega_{G,out} &= (1 - \alpha^m_{L,t}) \omega_{out}
\end{aligned}
$$

where z is the production choke opening. ω_{out} is the total mass outlet flow. P_r is the pressure at the riser. $\alpha^m_{L,t}$ is the mass fraction at the top of the riser. ρ_t is the density of the fluid at the top of the riser. P_0 is the outlet pressure. K_{pc} is a tuning parameter, experimentally obtained to fit a real-life riser. Equations for $\alpha^m_{L,t}$, ρ_t and P_r, calculation:

$$
\begin{aligned}
\alpha^m_{L,t} &= \frac{\alpha_{L,t}}{\rho_t} \\
\rho_t &= \alpha_{L,t} \rho_L + (1 - \alpha_{L,t}) \rho_{G,r} \\
P_r &= \frac{\rho_{G,r} R T_r}{M_G}
\end{aligned}
$$

where R is the universal gas constant, T_r is the assumed constant temperature in the riser. M_G is the gas molar weight. $\alpha_{L,t}$ is the liquid volume fraction at the riser top. $\rho_{G,r}$ is the gas density at the riser. ρ_L is the liquid density. Equations for $\alpha_{L,t}$ and $\rho_{G,r}$ calculation:

$$
\begin{aligned}
\alpha_{L,t} &= \frac{2 m_{L,r}}{V_r \rho_L} - \frac{A_L}{\pi r_p^2} \\
\rho_{G,r} &= \frac{m_{G,r}}{V_r - m_{L,r}/\rho_L} \\
V_r &= \pi r_r^2 (L_r + L_t)
\end{aligned}
$$

V_r is the total volume of the riser. L_r is the length of the riser. L_t is the length between the top of the riser and the choke valve. A_L is the area of liquid that is "blocking" gas passage at the low-point. r_p is the radius of the pipeline. r_r is the radius of the riser. A_L is modeled as:

$$
\begin{aligned}
A_L &= \pi r_p^2 - A_G \\
A_G &= \begin{cases} \pi r_p^2 \left(\frac{h_p - h_c}{h_c}\right)^2 & h_p < h_c \\ 0 & h_p \geq h_c \end{cases}
\end{aligned}
$$

A_g represents the area free for gas passage in the low-point. h_p represents the height of liquid in the pipeline. h_c represents the critical height in which gas cannot pass from the pipeline into the riser.

Equation for h_p calculation:

$$
\begin{aligned}
h_p &= K_h h_c \overline{\alpha}_{L,p} + \left(\frac{m_{l,p} - \rho_L V_p \overline{\alpha}_{L,p}}{\pi r_p^2 (1 - \overline{\alpha}_{L,p}) \rho_L} \right) sin(\theta) \\
\overline{\alpha}_{L,p} &= \frac{\overline{\rho}_{G,p} \omega_{L,in}}{\overline{\rho}_{G,p} \omega_{L,in} + \rho_L \omega_{G,in}} \\
\overline{\rho}_{G,p} &= \frac{P_{p,nom} M_G}{R T_p} \\
V_p &= \pi r_p^2 L_p
\end{aligned}
$$

$\overline{\alpha}_{L,p}$ is the average liquid volumetric fraction in the pipeline. V_p is the volume of the pipeline. θ is the inclination angle of the pipeline relative to the low-point. $\overline{\rho}_{G,p}$ is the average gas density in the pipeline, assumed to be constant and calculated using M_g, which is the gas molecular weight, R, the universal gas constant, T_p, the assumed constant temperature of the pipeline, and $P_{p,nom}$, the assumed nominal pressure of the pipeline, obtained through steady-state experiments.

For calculation of the flows at the low-point, we use the following equations:

$$
\begin{aligned}
\omega_{L,lp} &= K_L A_L \sqrt{\rho_L \Delta P_L} \\
\omega_{G,lp} &= K_G A_G \sqrt{\rho_{G,p} \Delta P_G} \\
\rho_{G,p} &= \frac{m_{g,p}}{V_p - m_{L,p}/\rho_L}
\end{aligned}
$$

K_G and K_L are tunable gains to tune the model to behave as an experimental application. ρ_G, p is the real gas density at the pipeline. ΔP_L and ΔP_G are the liquid and gas pressure difference at the low point, respectively, and are calculated as follows:

$$
\begin{aligned}
\Delta P_L &= P_p - \Delta P_{fp} + \rho_L g h_p - P_r - \overline{\rho_m} g L_r - \Delta P_{fr} \\
\Delta P_G &= P_p - \Delta P_{fp} - P_r - \overline{\rho}_m g L_r - \Delta P_{fr} \\
P_p &= \frac{\rho_{G,p} R T_p}{M_G} \\
\overline{\rho}_m &= \frac{m_{L,r} + m_{G,r}}{V_r}
\end{aligned}
$$

in which P_p is the pressure at the pipeline. $\overline{\rho}_m$ is the average mixture density at the riser. g is the gravity acceleration. ΔP_{fr} and ΔP_{fp} are the pressure loss due to friction for the riser and pipeline respectively, and are calculated as:

$$\begin{aligned}
\Delta P_{fr} &= \frac{\overline{\alpha}_{L,r}\lambda_r\overline{\rho}_m\overline{U}_m^2(L_r + L_t)}{4r_r} \\
\Delta P_{fp} &= \frac{\overline{\alpha}_{L,p}\lambda_p\overline{\rho}_L\overline{U}_{sl,in}^2 L_p}{4r_p} \\
\overline{\alpha}_{L,r} &= \frac{m_{L,r}}{V_r\rho_L} \\
\overline{U}_{sl,in} &= \frac{\omega_{L,in}}{\pi r_p^2\rho_L} \\
\overline{U}_m &= \frac{\omega_{L,in}}{\pi r_r^2\rho_L} + \frac{\omega_{G,in}}{\pi r_r^2\rho_{G,r}}
\end{aligned}$$

$\overline{\alpha}_{L,r}$ is the average liquid fraction at the riser. $\overline{U}_{sl,in}$ is the average superficial velocity of the inlet liquid. $\overline{U}_m$ is the average superficial velocity of the mixture in the riser. λ_p and λ_r are the friction factor of the pipeline and riser, respectively. They are calculated by the following equation:

$$\lambda_x = 0.0056 + 0.5Re_x^{-0.32} \tag{B.1}$$

in which x is either p, which stands for pipeline, or r, which stands for riser. Notably, the calculation of the friction factor needs the Reynolds Number Re of the mixture in both the pipeline or the riser. Re for each part of the system is calculated as follows:

$$\begin{aligned}
Re_p &= \frac{2\rho_L\overline{U}_{sl,in}r_p}{\mu} \\
Re_r &= \frac{2\overline{\rho}_m\overline{U}_m r_r}{\mu}
\end{aligned}$$

μ is the viscosity of the fluid, which is assumed to be constant.