DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Recurrent Neural Network Based Control for Risers and Oil Wells

Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS 5511: Projeto de Final de Curso

Jean Panaioti Jordanou

Florianópolis, Agosto de 2017

Recurrent Neural Network Based Control for Risers and Oil Wells

Jean Panaioti Jordanou

Esta monografia foi julgada no contexto da disciplina

DAS 5511: Projeto de Final de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

Prof. Eduardo Camponogara

Eric Antonelo,Ph.D

Marco Aurélio Aguiar,Ph.D Candidate

Banca Examinadora:

Marco Aurélio Aguiar
Orientador na Empresa

Prof. Eduardo Camponogara
Orientador no Curso

Prof. Hector Bessa Siveira
Responsável pela disciplina

Eng. Luis Kin Miyatake, Avaliador

Manoel Alvares Guidi, Debatedor

Daniel Manzoni Seerig, Debatedor

Acknowledgements

First, in regards to advising, I would like to thank prof. Eduardo Camponogara for all the advising and support, enabling the development of this work, and also for all the hints in the writing, for the disposition and patience, and for believing in me. I would also like to thank Eric Antonelo, for all the advising, support, and enabling me to gather most of the knowledge involved in this work, and also wisdom on basics such as how to write an academic paper. Also, Eric was the main instigator for me to begin this work. Further, I would also like to thank Marco Aurélio Aguiar, for the advising and support, and for helping me with Modelica and other oil and gas modeling issues. Also, for helping me with the basics of oil and gas industry.

I would like to thank all the people at my research group, for all the support, fun times, and help regarding the development of my work. Their company was valuable and they gave me the inspiration and insight necessary for the development of this work.

I would like to thank Ingrid Eidt, for being there and giving me the emotional support I needed, also for having patience with me and providing me unforgettable moments; my friends for all the fun moments we had; and my family, due to their support. Without it, I would not even be where I am now.

"The only true wisdom is in knowing you know nothing."
(Socrates)

Abstract

Recurrent Neural Networks (RNN) tend to be costly to optimize, though they posess desirable properties for dynamic system identification and serve as an universal approximator for these systems. To diminish this cost which can make RNNs impracticable, Echo State Networks were proposed in literature. Echo State Networks (ESN) are Recurrent Neural Networks and are divided in two parts: a recurrent netwok, named reservoir, in which the weights involved are fixed and randomly initialized; and a readout layer, composed of static neurons, where the output of an Echo State Network is computed. Only the weights from the readout layer are trained. In this training, relatively low cost algorithms such as the least squares can be used. Due to these properties, ESN can approximate complex dynamical systems with relatively low computational effort and global minima guarantee, and has obtained promising results in system identification and closed loop control of dynamic systems. There are successful demonstrations of ESN application in oil and gas plants. At the same time, in oil industry, several approaches are developed to solve the slugging flow problem utilizing feedback control. slugging flow problems are pertinent in oil platforms due to being capable of hindering significantly oil production, implying severe financial loss. With this application in mind, this work uses an adaptive control utilizing ESN to approximate the controlled system's inverse model to calculate the control action. This approach was applied to control the bottomhole pressure of an oil well and to apply anti-slug control of a pipeline-riser system which was subject to severe slugging flow regime. For the experiments, computer simulations were made utilizing models already stablished in literature. The closed-loop control of the oil well was subject to setpoint tracking and disturbance rejection tests. For the riser, it was tested which is the largest choke opening in which the riser maintains pressure stability, which corresponds to the maximum production without slugging flow. Based on the obtained results, this work demonstrated te applicability of ESN in oil production plants control and stabilization of severe slugging.

Keywords: Adaptive Control, Feedback Control, Neural Network (NN), Reservoir Computing (RC). Oil and Gas.

Resumo

Redes Neurais Recorrentes tendem a ser custosas de se otimizar, porém possuem propriedades desejáveis para identificação de sistemas dinâmicos e servem como aproximadores universais dos mesmos. Para diminuir este custo considerado impraticável, surgiu na literatura as Redes de Estado de Echo (Echo State Networks). Echo State Networks são Redes Neurais Recorrentes divididas em duas partes: uma rede de neurônios reccorentes, chamada de reservatório, onde os pesos são fixos e inicializados aleatóriamente e uma camada composta de neurônios estáticos, utilizados para computar a saída do modelo de aprendizagem dinâmica. Somente os pesos de saída desta rede são treinados, podendo ser utilizados algoritmos do tipo mínimos quadrados. Devido a estas propriedades, tais redes podem aproximar sistemas dinâmicos complexos custando baixo esforço computational, tendo obtido resultados promissores em aplicações de identificação e controle em malha fechada de sistemas dinâmicos. Há demonstrações promissoras do uso desse tipo de modelo em problemas envolvendo a indústria de petróleo e gás. Ao mesmo tempo, na industria de petróleo, várias abordagens são desenvolvidas para resolver o problem de golfadas utilizando controle em malha fechada. O problema de golfadas é pertinente numa plataforma de produção por ser capaz de causar grandes prejuizos na produção de petróleo, acarretando em perdas financeiras severas. Pensando nesta aplicação, este trabalho emprega uma estratégia de controle adaptativo utilizando Redes de Estado de Eco para se aproximar o modelo inverso do sistema controlado para o cálculo da ação de controle. Esta abordagem foi aplicada no controle da pressão de fundo de um poço de petróleo, juntamente com o controle anti-golfadas de um "riser", cujo modelo estava submetido à um severo regime de golfadas. Para os experimentos, foram utilizados modelos já presentes em literatura para simulações. Testes de rejeição de perturbação e seguimento de referência foram aplicados no poço de petróleo. Para o riser, foi testado qual o ponto de equilíbrio estável com maior abertura do choke de produção que o riser consegue manter. Com base nos resultados obtidos, o presente trabalho demonstrou a aplicabilidade das Redes de Estado de Eco ao controle de plantas de produção da indústria de petróleo e gás e também demonstrou sua capacidade em efetuar a estabilização de regimes severos de golfadas.

Keywords:Controle Adaptativo, Controle em Malha Fechada, Redes Neurais, Computação por Reservatório, Petróleo e Gás.

List of Figures

Contents

1 Introduction

1.1 Motivation

The Echo State Network (ESN) is a recurrent neural network (RNN) with a hidden layer whose weights are fixed and randomly assigned. The training takes place only at the output layer, usually by linear regression methods [8], yielding an efficient learning process with global convergence properties and relatively low computational cost. Echo State Networks are largely utilized for applications such as time series prediction and system identification, arguably so because of their ability to model and reproduce spatio-temporal patterns. Examples of works using this technique are: stock price prediction [9], steady-state detection during industrial compressor performance tests [10], learning of robot navigation behaviors [11], noninvasive fetal QRS detection [12], and even language modeling and processing [13].

In the literature, there are also applications of the use of Echo State Networks in control systems. [14], for example, utilizes an adaptive control system involving echo state networks to stabilize three different types of systems: a variable transport delay heating tank, a steady cruize airplane, and an inverted pendulum. This adaptive control system involves utilizing Echo State Networks to learn the inverse model of the plant in online mode. The model is identified while simultaneously learning the control action. Together with a sliding mode strategy, [15] used the same control structure as [14] to control a hydraulic excavator, a system with heavy nonlinearities. [16] presents another type of control structure, which uses least squares to train both the output and the input weights of only one ESN. [17] presents a strategy combining Echo State Networks and Model Predictive Control. The model is identified using an Echo State Network, which is used as the predictor for the algorithm. A technique was presented in the article to approximate the predictive control problem of an echo state network into a convex problem.

The oil and gas industry influences economics in a worldwide scale. Oil production systems are essential for the harnessing of this resource, and a large amount of money is lost each time a facility is not run as smoothly as expected [3]. One of the problems is called slug-flow. Slug flow consists in the constant accumulation and expulsion of gas due to the difference in gas and liquid velocity, compromising production and damaging equipment. [7] analyzes thoroughly the use of feedback control to attenuate or eliminate slug flow, developing simplified models for the control of risers and gas-lifted oil wells [5, 18] and robust linear control strategies. [19], [20], and [21] also use advanced control strategies to attenuate and/or supress slug flow. Solving slug flow problems is specially difficult, since the avaiable solutions in the industry are either too expensive or impractical [7],

which sparks the research on solving anti-slug through feedback control.

[22] demonstrates that an Echo State Network is able to identify and approximate not only a slugging flow riser model, but operational data of a slug flow happening in one of Petrobras's production platforms, showcasing the power of these identification models.

All these considerations brings us to one question: "could we use echo state networks for anti-slug control in oil wells and risers?". The technique presented has the advantage of not depending on a priori information of the model to tune its parameters, unlike classic linear or nonlinear control strategies.

1.2 Objectives

The objective of this work is to utilize an RNN control structure based on Echo State Networks [14] in the context of online identification and control of oil wells and risers. A simulated environment is set up so that the ESN controller is applied into the gas-lifted oil well model presented in [5] and the pipeline-riser model presented in [18]. In this environment, we utilize the controller to track setpoints and reject disturbances for the gas-lifted well model. Also, the riser is modeled with exactly the same parameters as [18], which manages to approximate a severe slugging regime according to a more exact reference model. The slug flow presented in the model featured in [18] is severe due to instabilizing with the production choke valve being almost closed. This presents a challange to the ESN controller due to the unstable dynamics. For the riser, the objective is to bring the controlled system to its limit. What is the stable operation point with the highest choke opening? A fully open choke induces no pressure drop and the lowest bottom-hole pressure, thereby inducing the highest production of fluids. This will evaluate how successful was the controller in negating slug flow.

1.3 Document Structure

The document is divided into five chapters.

Chapter 2 reviews basic concepts such as control theory, system identification, and adaptive control.

Chapter 3 introduces Echo State Networks, the inverse-model online control featured in [14] and the algorithm utilized to learn the model online.

Chapter 4 will introduce oil and gas, oil and gas production systems, the models which are utilized for this work's simulations, and a brief review of the literature on control strategies involved in oil production systems. Also, slug flow will be described, along with the methods on anti-slug utilizing feedback control.

Chapter 5 contains the experiments, implementation details, and simulation results

involving this work.

Chapter 6, then, concludes this document.

2 Control Theory Review

This chapter introduces the fundamental theory for the understanding of this work, featuring a brief review of Dynamical Systems, Control Problems, System Identification and Adaptive Control theory.

2.1 Dynamical Systems

A dynamical system is characterized by having memory. Memory essentially means dependance not only on current input, but also on past input [23].

Generally, a dynamical system response at current time t depends on all inputs applied from time $-\infty$ to t. To avoid computing that, which is impractical if not impossible, a concept called state is used.

A state is, as defined by [23], a variable that, when paired up with the input, defines uniquely the value of the output. A state, due to being dependent on the current input and recursively dependent on itself, is a representation of all the inputs applied to the system from time $-\infty$ to t.

A dynamical system also has outputs, which are functions of the states of the system and, unlike the states, can be measured in the real world. In practice, inputs are variables which we can manipulate directly, such as the opening of a tank valve or the steering wheel of a car. Outputs are the data that can be gathered by human beings or instruments, such as the speed-reading pointer at a car or a temperature measurement from a thermometer.

Memory is what defines a dynamical system, but a system can have other classifications regarding a few properties:

- A system is either causal or non-causal. A causal system does not depend on future inputs to compute the output. All physical dynamical systems are causal [23]. It is not possible for non-causal systems to exist.

- A system is either continuous-time or discrete-time. A continuous-time (discrete-time) system has its input, state and output signals in continuous-time (discrete-time). A discrete-time signal is a signal that, when computed from $-\infty$ to ∞, has an infinite but countable number of values. When a continous-time signal is computed on a small interval $[t, t+\delta]$, δ being a small number, it has infinitely uncountable points.

- A system is either linear or non-linear. A function f is linear, if and only if $\alpha f(x) + \beta f(y) = f(\alpha x + \beta y)$. This is analogue to systems. Several mathematical tools are

avaiable in literature, such as in [23], to deal with linear systems. The systems used for this work, along with almost all physical systems in the real world, are always non-linear. When the non-linearity is weak, a non-linear system can be approximated locally by a linear system. These approximations are used to define stability properties of a nonlinear system, as seen below.

- Systems are either time-variant or time-invariant. If a system is time-invariant, its behavior will never change over time. As with linearity, this assumption can also facilitate calculations, but systems tend to have time-variance, which is ignorable or not, depending on how slow the effect is. A slow time variant parameter such as the pressure in a oil and gas reservoir can be considered to be time-invariant for control purposes.

It is difficult to pinpoint exactly how a dynamical system of the real world behaves mathematically, so we tend to use models. A model is essentially a less complex, easier to understand approximation of the real world system. The model's complexity is directly related to its precision, though, the more complex a model is, the harder its computing becomes, so it is ideal to specify the model as simple as an application needs it to be.

There are several ways to represent a dynamic system by models. If the model is continuous and non-linear, it can be represented as an O.D.E (Ordinary Differential Equation), as follows:

$$f(\mathbf{y}(t), \dot{\mathbf{y}}(t)..., \mathbf{y}^{(n)}(t)) = g(\mathbf{u}(t), \dot{\mathbf{u}}(t), ...\mathbf{u}^{(n)}(t)) \tag{2.1}$$

where $\mathbf{y}(t)$ is the output vector and $\mathbf{u}(t)$ is the input vector. For a generic function $x(t)$, $\dot{x}(t)$ is defined as the 1st derivative of x in time. $x^{(n)}(t)$ is defined as the n-th derivative of x in time. Or it can be represented as a system of first order ODEs, called state equations:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.2}$$

$$\mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.3}$$

with $\mathbf{x}(t)$ being the state vector at time t.

In case the model is in discrete time, the model can be represented by n-th order difference equations:

$$f(\mathbf{y}[k], \mathbf{y}[k-1], ..., \mathbf{y}[k-n]) = g(\mathbf{u}[k], \mathbf{u}[k-1], ..., \mathbf{u}[k-n]) \tag{2.4}$$

or by a system of n first order difference equations:

$$\mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k]) \tag{2.5}$$

$$\mathbf{y}[k] = g(\mathbf{x}[k], \mathbf{u}[k]) \tag{2.6}$$

If a system is linear, there are other forms that can be used to analyze the system

and the exact function that describes the system response can be analitically known. Such methods are not dealt with in this work. For more information, refer to [23].

An important concept related to non-linear systems is the equilibrium point, which is also referred to as "operation point". For continuous systenms, an operation point is a state vector $\overline{\mathbf{x}}$ that satisfies the following condition:

$$\mathbf{0} = f(\overline{\mathbf{x}}, \mathbf{u}) \tag{2.7}$$

whereby $\dot{\mathbf{x}} = 0$ means that the state is constant over time. Equilibrium points can be stable or unstable. This is easily analyzed if the eigenvalues of the Jacobian of $f()$ are all nonzero and finite. The jacobian is utilized to linearize the system in the neighborhood of the operating points. If one of the Jacobian's eigenvalue is either zero or infinite, the linearized system's behavior does not represent the nonlinear system.

An equilibrium point being stable means that, for any state at an instant t, $x(t) = \overline{x} + \delta$, with δ being a vector of sufficiently small numbers, the dynamic system state will converge to $\overline{x}$ at $t \to \infty$. An unstable equilibrium point has the opposite behavior. Any $\mathbf{x}(t) = \overline{\mathbf{x}} + \delta$ for small δ will diverge from $\overline{\mathbf{x}}$.

2.2 Control Problem

A control problem is: given a certain dynamical system with input $\mathbf{u}(t)$, state $\mathbf{x}(t)$ and output $\mathbf{y}(t)$, and a desired output trajectory $\widehat{\mathbf{y}}(t)$, what input trajectory should be applied so that $\mathbf{y}(t) = \widehat{\mathbf{y}}(t)$? The set of rules defining $\mathbf{u}(t)$ is also referred to as "control law", and a "control strategy" is a structure which the control law is derived from. Inputs are also reffered to as "manipulated variables (MV)" and outputs as "controlled variables (CV)".

There are two main types of control strategies:

- **Feedforward control**: No information from the output $\mathbf{y}(t)$ is used to compute the control action. Also called "open-loop" control.
- **Feedback control**: Information from the output is used to compute the control action. Also called "closed-loop" control.

Generally in industry, $\widehat{\mathbf{y}}(t)$ is a constant signal, also called the setpoint. In this case, the control problem is about maintaining a dynamical system (also called a plant) in a certain operation point.

"Open-loop" control could be applied, but if the dynamical system suffers some slight parametric change, it would deviate further from the setpoint. Even with this change, called "disturbance", the setpoint could be determined if the closed-loop control strategy has certain properties. For more information, refer to [23]. There are two main subtypes of control problem:

- **Setpoint Tracking**: Assuming that the system's output is not $\widehat{\mathbf{y}}(t)$, could the system be brought into the operating point so that $\mathbf{y}(t) = \widehat{\mathbf{y}}(t)$, how fast?
- **Disturbance Rejection**: Assuming that the the system output $\mathbf{y}(t)$ is $\mathbf{y}(t) = \widehat{\mathbf{y}}(t)$, if a parametric change in the model occurs, could $\mathbf{y}(t)$ still equal $\widehat{\mathbf{y}}(t)$ at steady state? How quick would the system come back to its operation point?

This work's objective is to deal with control problems found in oil and gas production platforms, with both tracking and disturbance rejection in piecewise constant setpoints.

2.3 System Identification

As explained in Section 2.1, there is a difference between a mathematical model and a real-life system. A model is almost always a simpler approximation of a system which is present in real life.

When reffering to a real-life application, three levels of prior knowledge of a certain physical system are considered in the literature [24]:

- **White-Box Model**: Sufficient information of the physical phenomena involved in the system modeling is known. Requires identification of very few, if any, parameters.
- **Grey-Box Model**: A certain amount of information about the system is known. Some dynamics are unknown or too hard to model, needing identifcation.
- **Black-Box Model**: No prior knowledge of system is avaiable or it is too hard to model. The identification problem extends for all dynamics of interest in the desired application.

System identification consists in using data driven information so that it finds a model that behaves the closest possible to the real-life system in a certain operating region. Ideally, it would be desirable that the model behaves just like the system in all possible regions of operation but, if that task is not impossible, its difficulty is impracticably high and the model would have to be too complex to be tractable. A simple model would be easier to control and/or optimze.

There are two ways to gather data for model identification: online or offline. A model is identified offline when all the data is given at once, from a separate runtime of the process. The model is identified online when data is given to the model and the model is validated all at the same time the data is output by the plant.

According to [24], a system identification problem consists in these eight steps:

1. **Choice of Model Inputs**

2. **Choice of Excitation Signal**
3. **Choice of Model Architecture**
4. **Choice of Dynamics Representation**
5. **Choice of Model Order**
6. **Choice of a Model Structure**
7. **Choice of Model Parameters**
8. **Model Validation**

For Step 1, in a control context, usually the inputs of an identified model are all the manipulated variables available.

In step 2, the excitation signal in a linear system is easily defined by a Pseudo Random Binary Signal (PRBS), due to this signal class having a well defined frequency spectra, even though being pseudo-random in the time domain. For a non-linear system, this choice is nontrivial, due to the fact that a PRBS takes advantage of the constant input-output gain of a linear system, which does not happen on a non-linear system. [24] gives an introdution to the vast theory that is nonlinear system excitation for identification. This work deals with both a nonlinear system and a non-linear identification model, but since the identification model is in closed loop with the plant, this is a non-issue due to the loop being not free for system excitation. Sometimes a noisy reference signal $\widehat{\mathbf{y}}(t)$ is used.

For step 3, the model architecture depends on the intended use, the problem type, the problem dimensionality, the avaiable amount of data, time constraint, memory restrictions, or if the model is learned online and offline.

Step 4 is problem-dependent since it consists of which variables will be used to represent the dynamics of the system.

Step 5 and 6 are selected by trial and error. The higher a model order and the more complex a structure, the higher the capacity of fitting the data. High complexity models can lead to overfitting problems, which can be avoided through grid search and cross-validation strategies for parameter decision.

Step 7 is, as described by [24], the easiest to automate. Generally the parameter choice can be reduced to optimization problems. If the parameters are linear, then the optimization problem that is solved is a linear least squares problem. The fitting of the model is also called "training", and the data used for fitting is called "training data".

Step 8 is the model validation, which evaluates how well the model fits the dynamic system identified. The criteria to evaluate a certain model's performance is application-dependent. The easiest metric to evaluate performance is the training error of the model, which is the error between the system and the model, evaluated at the training points. [24].

In Machine Learning literature, a model not having low enough training error is called "underfitting", which means that the model is not complex enough to fit the data and a more intricate model should be chosen. The opposite of underfitting would be "overfitting", when the model is too complex and can fit almost perfectly the training data, but it is not capable of generalization. In other words, it has high "test error", error of approximating acquired data which are not used for fitting the model. Both problems are structural by nature, but could be solved by regularization [25].

In this work we use system identification to an adaptive control application, which recquires the parameters to be identified online. The model which we are trying to identify is a dynamic that approximates the "inverse model" of the system.

An inverse model, is a system g that, given a system represented by f that has an input vector $\mathbf{u}(t)$ and an output $\mathbf{y}(t)$, $\mathbf{y}(t) = f(\mathbf{u}(t))$, we desire to find g, so that $\mathbf{u}(t) \approx g(\mathbf{y}(t))$. Then we can say that g is an inverse model of f.

2.4 Adaptive Control

In section 2.2 we defined what a control problem is. An adaptive control problem is essentially a control problem for dynamic systems that varies its parameters over time [1]. Adaptive Control is generally applied when the varying dynamics of the system are too critical so that a linear controller does not work. This is due to the fact that they tend to be complicated to tune [1]. Though, systems with heavy nonlinearities can be seen as time variant systems in terms of black box identification models.

Adaptive control strategies typically have two different control loops: the standard closed-loop control which computes the control action, and a secondary loop which computes the control parameters. Figure 1 contains a diagram representing this loop.

This is a list of examples of common Adaptive Control Strategies, for computing the control parameters, from [1]:

- **Gain Scheduling**: Used when measurable variables correlate well with dynamics change. Basically, the gain of the controller is a function of the operating point the system finds itself in. Gain Scheduling was developed for flight control systems in airplanes. Could be implemented as a function or look-up table.
- **Model-Reference Adaptive Systems**: They are designed in terms of a "reference model". This reference model is a representation of the desired performance of the system. The adaptive law tries to bring the dynamics as close as possible to this reference model. For tuning, generally either the gradient method is used or a stability criteria is used, such as Lyapunov Stability [1].
- **Self-Tuning Regulators**: A self-tuning regulator is a process that automates the procedure of tuning an offline controller, including modeling, control law design,

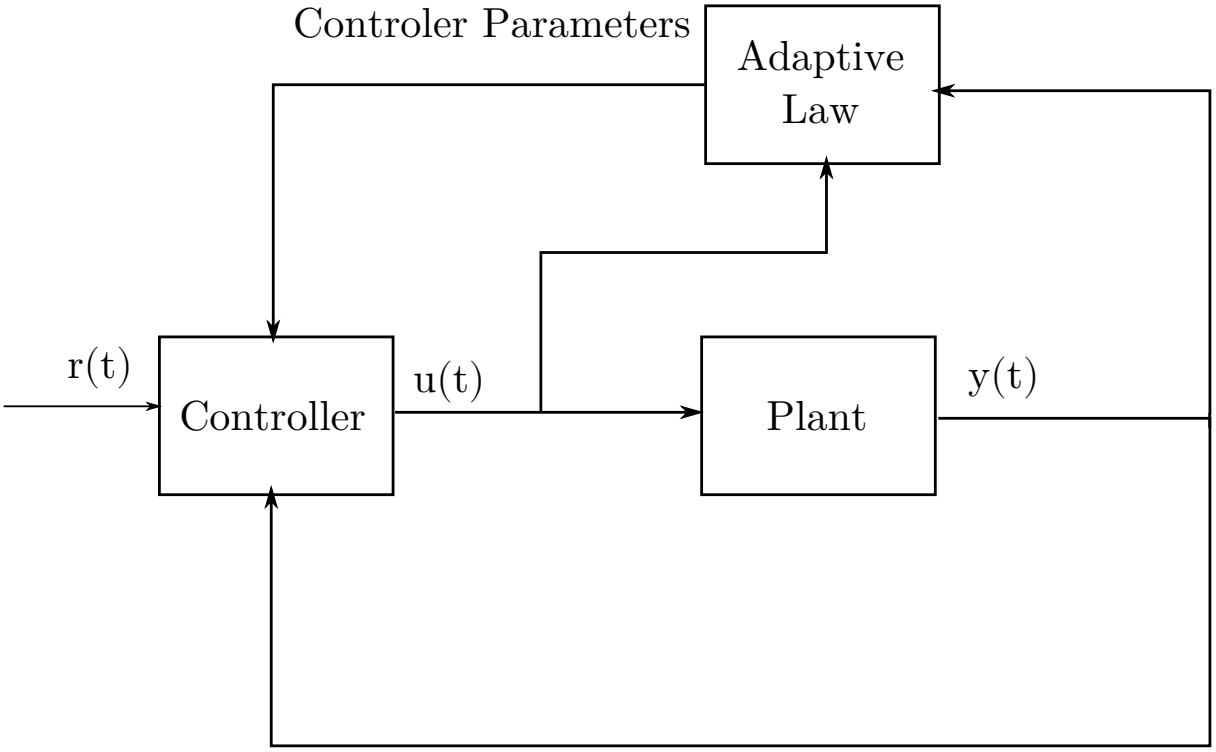

Figure 1 – This is an example of a generic adaptive control loop [1]. A controller receives as input the reference signal $r(t)$, the plant output $y(t)$ and the controller parameters. For computation of the controller parameters, the adaptive law uses both the control action $u(t)$ and the manipulated variable $y(t)$.

implementation and validation. Those processess, normally done offline, are done in the loop. There is a system identification model, which is trained online. The parameters of the control are selected as a function of the identification model's parameters, just like in an offline procedure, but updated constantly due to the online identification of the model.

- **Dual Control**: Dual Control, according to [26], is defined as optimal control for a system which has "dual effect" and the choice of the decision variables depends on the uncertainty. It is by far the most formally stated adaptive control problem, but the hardest to compute the control action of the frameworks presented here [1]. A system having "dual effect" means essentially that the control action can affect the shape of a probability distribution in a stochastic system. This approach is more theoretical than the others, and depends on assuming a probability distribution and an optimization problem formulation to compute the control action. The adaptive aspect from this type of framework is that it is a Model Predictive Control (MPC) in which the model has adaptive parameters $\mathbf{\Theta}$. This parameter adaptation consists in an identification problem by itself. In [26] it is presented as an expected value minimization of $y^T Q y + u^T R u$ given output observation $\hat{y}(t)$ to $\hat{y}(t+N-1)$, N being the control horizon. The restrictions associated both with the state x, output y

and adapted parameter Θ all contain noise which is assumed to be uncorrelated. To efficiently solve this non-relaxed (referred to "ideal dual control" in [26]) optimization problem is still an open problem as of the writing of [26].

All these techniques have in commom that they are harder to design than classic, offline control strategies. Nonetheless, they are used in industry for applications recquiring tracking and quick operation point change, such as the airplane industry. Some industrial controllers contain auto-tuning functions [24]. For oil and gas, adaptive control would be convenient because of the high uncertainity involved in the production platform models. The version of risers and oil wells presented in this work are simplification models, and it is important for a controller to quickly adapt to changes in condition (e.g., the temperature for all the model is considered to be constant, but there is high variability in temperature underground).

2.5 Summary

In this chapter, an introduction of control theory and system identification concepts essential for the understanding of this work is made. We briefly reviewed the theory related to Dynamic Systems, the basic definition of control problems, System Identification, and Adaptive Control. This entails all the theory that is used for the understanding of the Recurrent Neural Network based controller.

3 Recurrent Neural Network Based Control

In this chapter, we will introduce the tools used in this work. We introduce the concept of Echo State Networks, which is the identification model type utilized, and how a model is trained online. Then, we introduce the control loop featured in this work: an adaptive control loop which uses the echo state network to compute the inverse model of the plant, using the obtained model to calculate the control action.

3.1 Echo State Networks

An Echo State network is a type of identification model with very convenient properties for the use in the online-learning control [27]:

- A nonlinear high-dimensional model, capable of approximating intricate nonlinear dynamics.
- Linear training, since it can be trained using relatively quick algorithms such as Least Squares and Recursive Least Squares.

.

This Recurrent Neural Network model was proposed in [8] and obeys the following

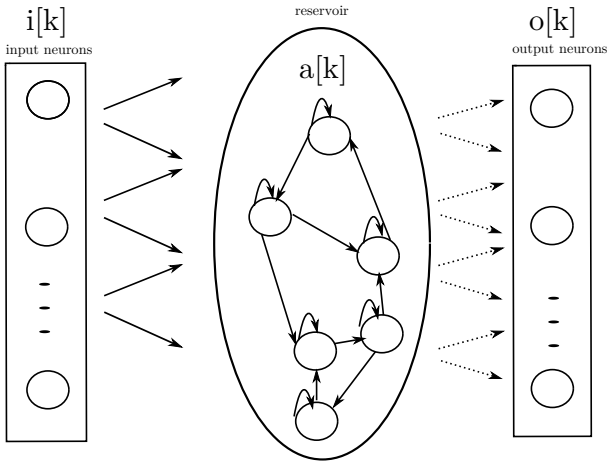

Figure 2 – Representation of an Echo State Network. $\mathbf{i}[k]$ is the input to the network. $\mathbf{a}[k]$ represents the neuron's activations in the reservoir (hidden layer). $\mathbf{o}[k]$ is the output layer of the network. Dashed connections are trainable while solid connections are fixed and randomly initialized. Figure inspired in [2]

discrete-time dynamic equations:

$$\begin{aligned}\mathbf{a}[k+1] &= (1-\gamma)\mathbf{a}[k] \\ &\quad + \gamma f(\mathbf{W}^{\mathbf{r}}_{\mathbf{r}}\mathbf{a}[k] + \mathbf{W}^{\mathbf{r}}_{\mathbf{i}}\mathbf{i}[k] + \mathbf{W}^{\mathbf{r}}_{\mathbf{b}}) \end{aligned} \tag{3.1}$$

$$\mathbf{o}[k+1] = \mathbf{W}^{\mathbf{o}}_{\mathbf{r}}\mathbf{a}[k+1] \tag{3.2}$$

A graphical description of these equations appears in Figure 2. 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 a parameter called leak rate [27], which governs the percentage of the current $\mathbf{a}[k]$ into $\mathbf{a}[k+1]$. The weights are represented in the notation $\mathbf{W}^{\mathbf{to}}_{\mathbf{from}}$, with $\mathbf{o}$ meaning the output neurons, $\mathbf{r}$ meaning the reservoir, and $\mathbf{i}$ meaning the input neurons. "$\mathbf{b}$" represents the bias.

The network has a number N of neurons, which is the dimension of $\mathbf{a}[k]$ and must be several orders higher than the number of inputs or outputs. The higher N is, the more complex the model for having more states, so it is more capable of dynamics approximation. As described in section 2.3, models that are more complex are also more propense to overfitting. Furthermore, the computation of the output becomes more expensive and can become excessive. So, it is best for N to be just as large as to not underfit the model, unless a regularization method is used (then, N can be as large as needed). f is the activation function which, in this work and in [8, 14, 27], is the hyperbolic tangent $f = \tanh(\cdot)$. $f = \tanh(\cdot)$ is what is called a base function in system identification theory [24] being widely used in the literature.

In a standard Recurrent Neural Network (RNN), all the weights are trained using Backpropagation Through Time [28]. Such a training method of RNNs is computationally expensive, which leads into the advantage of the Echo State theory described above.

The Echo State Property [8, 27] is achieved when, for any input sequence, the state sequence is unique at steady state, which is referred to as "having fading memory". When a recurrent neural network has Echo State, it is capable to learn by training only the output weights, which reduces the problem to a Least Squares Problem, given that the objective function is the squared error.

In this work, the Echo State Network is initialized as proposed by [14]. This initialization follows the following steps:

1. Every weight of the network is initialized from a normal distribution $\mathcal{N}(0, 1)$.

2. $\mathbf{W}^{\mathbf{r}}_{\mathbf{r}}$ is scaled so that its spectral radius (eigenvalue with largest module) ρ is at a certain value able to create reservoirs with rich dynamical capabilities. [14], [8] and [27] argue that setting $\rho < 1$ in practice generates reservoirs with the echo state property (and in many cases maximizes the performance of the network).

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 parameters are crucial in the learning performance of the network [22].

These procedures do not necessarily guarantee Echo State, but are very effective in practice [22]. In order to guarantee the ESP, one must set the singular values of $W_r^r < 1$, although this restriction limits the richness of dynamical qualities of the reservoir. Because of this, the spectral radius has been used more often as a scaling reference.

3.2 Online Optimization

This section serves to briefly describe the online optimization method used in this work. First, we introduce the Least Squares Problem. Then, we will show how to solve it analytically. The Recursive Least Squares (RLS) algorithm is derived from the analytical solution.

3.2.1 Least Squares Problem

The least squares problem is an unconstrained optimization problem which minimizes the following cost function:

$$J = \sum_{k=0}^{N} \|\hat{y}[k] - y[k]\|_2^2 \tag{3.3}$$

This cost function represents the quadratic error between the identification model's estimated output $\hat{y}[k]$ and the real output $y[k]$ at time k, for the N gathered data points. This cost function assumes that both $y[k]$ and $\hat{y}[k]$ are scalars but, in case of a multi-output identification problem, this problem is solved for each output. The model is described as $\hat{y}[k] = \boldsymbol{\theta}^T \mathbf{x}[k]$, the linear parameter vector $\boldsymbol{\theta}$, which is the decision variable in this problem, multiplied by the input vector $\mathbf{x}[k]$, which contains all features that are used to describe a model (e.g., $\mathbf{x}[k] = (u[k], y[k], y[k-1])^T$, with $u[k]$ being an input to the system and $y[k]$,$y[k-1]$ being two values of the output at different instants of time). In the case of an Echo State Network, $\boldsymbol{\theta} = \mathbf{W_r^{o}}^T$, and $\mathbf{x}[k] = \mathbf{a}[k]$.

Since an unconstrained quadratic error formulation is used, all variables can assume any real value and the optimization problem is convex, so it has one global minimum. This implies that there is a value for $\boldsymbol{\theta}$ that brings $\hat{y}[k]$ as close as possible to $y[k]$.

In matrix form, J is represented as:

$$J = (\mathbf{X}\boldsymbol{\theta} - \mathbf{Y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}) \tag{3.4}$$

with $\mathbf{X}$ being evaluated as $\mathbf{x}^T[k]$ in line k and $\mathbf{Y}$ is defined as the vector that the element at line k equals to $y[k]$. If a symetric, positive definite weight matrix $\mathbf{Q}$ is added in a way that J becomes:

$$J = (\mathbf{X}\boldsymbol{\theta} - \mathbf{Y})^T\mathbf{Q}(\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}) \tag{3.5}$$

this version of the problem is known as "Weighted Least Squares". Since a row belonging to matrices $\mathbf{X}$ and $\mathbf{Y}$ represents the point collected at time k in a system identification training process, the importance of each observation at time k can be ponderated. This is actually used in the Recursive Least Squares algorithm presented in this work.

The Least Squares Problem applies not only to purely linear functions, but to functions which are linear in the parameters, such as an Echo State Network or any n-degree polynomial. All it needs to be done is treating the non-linear terms multiplying the linear parameters as new, separate inputs [24]. (e.g., to fit a model $\theta_1 u^2 + \theta_2 u + \theta_3$, all it needs to be done is letting $x[k]$ be $(u[k], u^2[k], 1)^T$ in the point of view of the cost function).

As mentioned in section 2.3, a model which has a large number of parameters can perform better at the minimization of the "training error", which is the error that this problem is trying to minimize, but will not be able to fit points outside the training region. One way to work around this problem is with regularization [22], [24], which consists in penalizing the magnitude of the decision variables. This boosts the capacity of a model to generalize and is easier than optimizing the model structurally [24]. Adding regularization, the cost function would be:

$$J = (\mathbf{X}\boldsymbol{\theta} - \mathbf{Y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}) + \beta\boldsymbol{\theta}^T\boldsymbol{\theta} \tag{3.6}$$

with β being a scalar whose purpose is to penalize the $\mathcal{L}_2$-norm of $\boldsymbol{\theta}$.

3.2.2 Analytic Solution

[24] shows how to solve all the problems presented in Section 3.2.1. To analytically solve a convex problem, we must find $\boldsymbol{\theta}$ so that $\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$. Since J is quadratic, this implies solving a linear system. Below is the solution to the three versions of the least squares problem presented:

Least Squares:

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{3.7}$$

Weighted Least Squares:

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}\mathbf{Y} \tag{3.8}$$

Regularized Least Squares (Also known as Ridge Regression):

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X} + \beta\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \tag{3.9}$$

The inverse shown in the solution of all the Least Squares version is not actually computed in a numerical least squares solver. A cheaper way to compute the solution is to find $\boldsymbol{\theta}$ as a solution to the following linear system:

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^T\mathbf{Y} \tag{3.10}$$

This way, the computation of the inverse matrix of $\mathbf{X}^T\mathbf{X}$ is avoided.

3.2.3 Recursive Least Squares

[24] and [29] show how to derive the Recursive Least Squares from the analytic solutions presented in this work. This version of the Recursive Least Squares is derived from a Weighted Least Squares Problem where:

$$\mathbf{Q} = \begin{pmatrix} \lambda^n & 0 & \cdots & 0 \\ 0 & \lambda^{n-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda^0 \end{pmatrix} \tag{3.11}$$

and λ is called the forgetting factor which is generally $0.9 \leq \lambda \leq 1$. If $\lambda = 1$, then $\mathbf{Q}$ would be the identity matrix, so it would be equivalent to a normal Least Squares problem. λ is called the forgetting factor due to the fact that, if $\lambda \leq 1$, and assuming each training example corresponds to a time step in a simulation or the sample time of a generic data acquisition system, then the quadratic error of recent samples receives more consideration during optimization. The result would then perform better for these selected points. In a sense, the optimizer is "forgetting" older samples.

Another way to represent the Weighted Least Squares cost function is:

$$J = \sum_{k=0}^{N} \lambda^{N-k} \|\hat{y}[k] - y[k]\|_2^2 \tag{3.12}$$

with N being the number of time steps which were sampled.

To derive the least squares problem for the Recursive Least Squares algorithm,

some auxiliary definitions are recursively made:

$$\mathbf{X}[k+1] = \begin{pmatrix} \mathbf{X}[k] \\ \mathbf{x}^T[k+1] \end{pmatrix} \tag{3.13}$$

$$\mathbf{Y}[k+1] = \begin{pmatrix} \mathbf{Y}[k] \\ y[k+1] \end{pmatrix} \tag{3.14}$$

$$\mathbf{Y}[0] = y[0] \tag{3.15}$$

$$\mathbf{X}[0] = \mathbf{x}^T[0] \tag{3.16}$$

Naturally, we would have to define the value of $\boldsymbol{\theta}$ at time step k, which would be:

$$\boldsymbol{\theta}[k] = (\mathbf{X}^T[k]\mathbf{Q}\mathbf{X}[k])^{-1}\mathbf{X}^T[k]\mathbf{Q}\mathbf{Y}[k] \tag{3.17}$$

$$\boldsymbol{\theta}[k+1] = (\mathbf{X}^T[k+1]\mathbf{Q}\mathbf{X}[k+1])^{-1}\mathbf{X}^T[k+1]\mathbf{Q}\mathbf{Y}[k+1] \tag{3.18}$$

Then, we define:

$$\mathbf{P}[k] = (\mathbf{X}^T[k]\mathbf{X}[k])^{-1} \tag{3.19}$$

$$\tag{3.20}$$

which is known as the correlation matrix and carries all the runtime information that the system has learned. Due to it being defined as the inverse of $\mathbf{X}^T\mathbf{X}$, the computation of an inverse matrix is avoided.

Using these definitions, a way to compute $\boldsymbol{\theta}[k+1]$ from $\boldsymbol{\theta}[k]$ is derived from [24], as follows:

$$\mathbf{P}[0] = \frac{1}{\alpha} I \tag{3.21}$$

$$e[k] = \boldsymbol{\theta}^T[k-1]\mathbf{x}[k] - y[k] \tag{3.22}$$

$$\mathbf{P}[k] = \frac{\mathbf{P}[k-1]}{\lambda} - \frac{\mathbf{P}[k-1]\mathbf{x}[k]\mathbf{x}^T[k]\mathbf{P}[k-1]}{\lambda(\lambda + \mathbf{x}^T[k]\mathbf{P}[k-1]\mathbf{x}[k])} \tag{3.23}$$

$$\boldsymbol{\theta}[k] = \boldsymbol{\theta}[k-1] - e[k]\mathbf{P}[k]\mathbf{x}[k] \tag{3.24}$$

It is important to reitarate that, in this work, $\boldsymbol{\theta} = \mathbf{W}_\mathbf{r}^{\mathbf{o}T}$ and $\mathbf{x}[k] = \mathbf{a}[k]$. $\boldsymbol{\theta}$ is assumed to be a vector for this algorithm, but in case $\boldsymbol{\theta}$ is a matrix and, in consequence, $\mathbf{W}_\mathbf{r}^\mathbf{o}$ is also a matrix, then these equations are computed for each collumn of $\boldsymbol{\theta}$ separately.

$e[k]$ is called the "a priori error", since it computes the error using $\boldsymbol{\theta}[k-1]$ (the parameters of the previous iteration's model).

α is the "learning rate". If a priori data for the system is already provided, then $\mathbf{P}[0]$ could be $(\mathbf{X}^T\mathbf{X})^{-1}$ with $\mathbf{X}$ being the data already gathered about the system. If not, $\mathbf{P}[0]$

is, for simplicity purposes the identity matrix multiplied by $\frac{1}{\alpha}$. The smaller the parameter α is, the more it is assumed that we do not know about the system. Heuristically, it would be ideal that $\alpha = 0.01$ ou $\alpha = 0.001$ [24], for leading to high corrections on the error.

3.2.4 Adaptive Forgetting Factor

In Section 3.2.3, RLS is defined for a fixed forgetting factor λ, but there is the option of varying it. The justification for that, as well as the development that leads to the algorithm used in this work, is from [30].

When deciding on a fixed λ, a compromise must be made. Lower values of λ can boost the performance of the algorithm for changes within the system after the parameters are in steady state. This is due to the "forgetting" effect a lower value of λ provides. The downside of this is loss of performance when the system to be identified stays too long in the same operating point, which could even lead to instability. Also, having values of λ lower than 1 compromises noise rejection in a system identification application. So it is ideal that when setting a forgetting factor, it is not far from 1. the literature tends to mention the lower bound of $\lambda = 0.9$ [24].

Given this compromise, a new method is proposed so λ is modified to meet good performance in both the transient and the steady state, [30]:

$$\lambda[k] = \min\left(\frac{\sigma_q[k]\sigma_v[k]}{\epsilon + |\sigma_e[k] - \sigma_v[k]|}, \lambda_{\max}\right) \tag{3.25}$$

$$\sigma_e^2[k+1] = \alpha\sigma_e^2[k+1] + (1-\alpha)e^2[k] \tag{3.26}$$
$$\sigma_q^2[k+1] = \alpha\sigma_e^2[k+1] + (1-\alpha)q^2[k] \tag{3.27}$$
$$\sigma_v^2[k+1] = \beta\sigma_v^2[k+1] + (1-\beta)e^2[k] \tag{3.28}$$

$$\alpha = 1 - \frac{1}{K_\alpha N} \tag{3.29}$$
$$\beta = 1 - \frac{1}{K_\beta N} \tag{3.30}$$

in which σ_u is an estimation of the expected value $E(u)$, u being an unknown signal. v is the noise associated with the error, and $q[k] = \mathbf{x}^T[k]\mathbf{P}[k]\mathbf{x}[k]$. The σ_x are updated by moving averages using time constants α and β, with $K_\alpha \geq 2$ and $K_\beta \geq K_\alpha$. This is to ensure that the equation defining σ_v^2 has a longer exponential window than σ_e^2. $\lambda_{\max}$ is the maximum value of the forgetting factor. An upper bound is set so that λ cannot grow indefinitely. In this work, the $\lambda_{\max}$ used is 0.9999. The algorithm lowers λ while the system is undergoing change (where the error rises) and raises λ in steady state (where the error is near 0). ϵ is a small number to avoid division by zero in the equation deciding λ, since at steady state σ_e is close to σ_v theoretically [30].

The general idea of this algorithm is to estimate the expected value of the noise, input and error, and decide λ according to their magnitude. The equation for defining λ at time k is obtained in [30] by solving the value of λ when the expected value of the error is equal to the expected value of the noise. Then, the moving averages are used to practically estimate the noise expected value. [30] argued that the Variable Forgetting Factor Recursive Least Squares would be more robust than the regular RLS.

3.3 Online-learning Inverse Model for Feedback Control

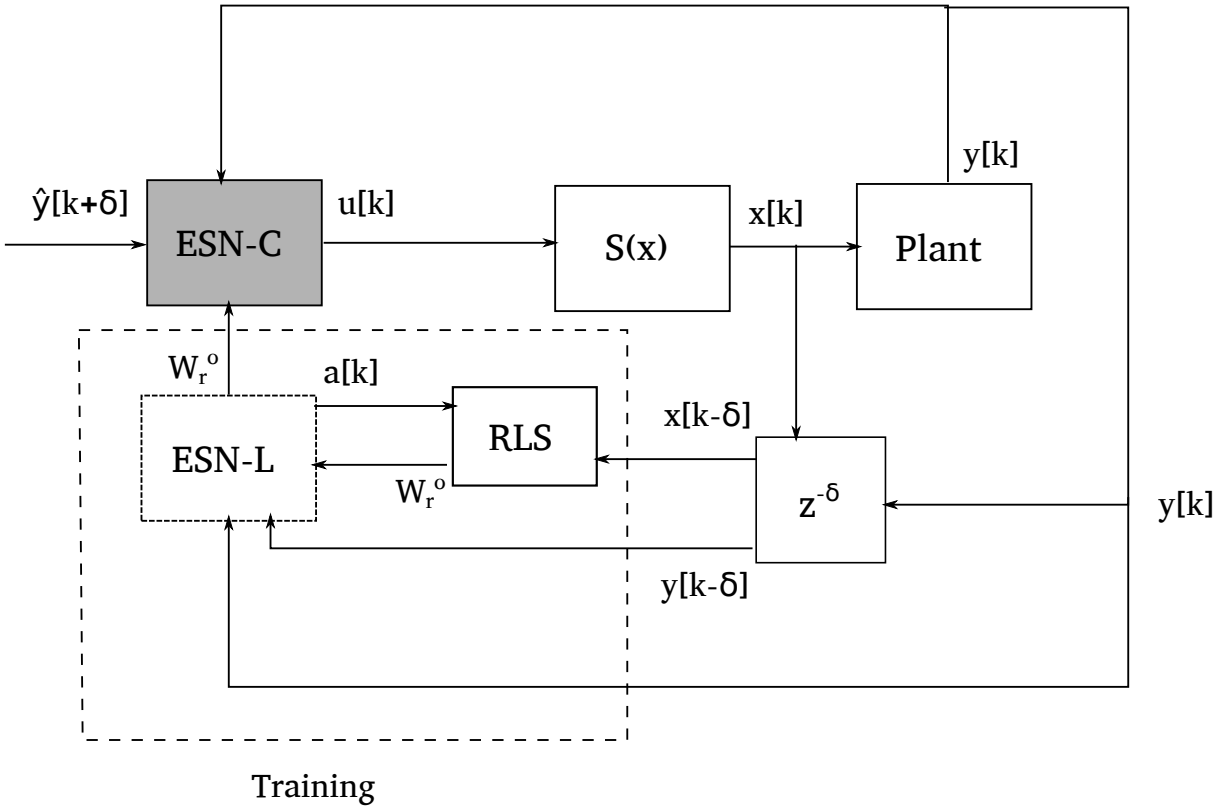

Figure 3 – Schematic of the ESN-based control framework. L stands for "Learning Network," and C stands for "Control Network." $S(x)$ represents the physical limitations imposed onto the manipulated variable, such as saturation and rate limiting. This figure was extracted from [2]

This control strategy was originally proposed by [14]. The Inverse Model-based controller utilizes two copies of the same identification model. This work uses Echo State Networks, but other identification models can be used as well [14]. One, denoted "Learning Network" (represented by ESN-L in Figure 3 due to the use of Echo State Networks), has the job akin of a parameter estimator in an adaptive control framework. It serves to approximate the inverse model of the controlled plant. The parameters are trained online using Recursive Least Squares (RLS) with forgetting factor. The following information is used to train the learning network:

- $x[k-\delta]$: The input to the plant at δ time steps before the present time k.
- $y[k-\delta]$: The output of the plant at δ time steps before the present time k.
- $y[k]$: The current output of the plant.

in which δ is an user-chosen parameter.

The other network, referred to as "Control Network" (represented by ESN-C in Figure 3 due to the use of Echo State Networks), acts as the "Controller" in an adaptive control strategy as depicted in Figure 1. It receives the parameters (weights) of the parameter estimator (the Learning Network, i.e., ESN-L). Thus, while ESN-L learns the weights or parameters in an online fashion, ESN-C uses these learned weights to control the plant, what makes these adaptive weights a shared parameter vector between both networks. The controller uses the parameters given by the "Learning Network" to compute the control action $u[k]$, based on these information:

- $y[k]$: The current output.
- $\hat{y}[k+\delta]$: The desired output at time step $k+\delta$.

It can be easily noticed that the inputs and outputs of the "Control Network" are the ones in the "Learning Network", displaced δ timesteps in the future. In other words, the Control Network uses information learned by the Learning Network about the inverse model to compute the control action $u[k]$ that leads $y[k]$ to $\hat{y}[k+\delta]$ in δ timesteps.

A detailed proof of convergence of this method can be found at [14], as well as its use in three interesting applications: A variable delay model of a heating tank, which uses the flow as the sole manipulated variable, a linear model of a plane in steady cruise using the attack angle and an inverted pendulum.

3.4 Summary

In this chapter, we introduced the concepts of Echo State Networks, Online-Learning Feedback control, and Variable Forgetting Factor Recursive Least Squares, which will be the tools used for the control of the Riser and the Oil Well. This entails all the theory that is used for the Recurrent Neural Network based controller.

4 Control of Risers and Oil Wells

4.1 Introduction to Oil and Gas

This section will introduce basic concepts associated with oil and gas, for contextualization purposes.

4.1.1 Motivation

Oil and Gas is one of the most important energy sources in the world. It all began in 1859, with the first successful oil well drilling by "Colonel" Edwin Drake, [31], which began a whole international search for industrial use of petroleum. By the second industrial revolution, in the 19th century, oil has replaced most other fuels for motorized transport [31], which is still true nowadays. Besides being used as energy fuel for mobile applications, it is also heavily used by the petrochemical industry in daily objects, such as [31]:

- Synthetic Rubber,
- Plastic,
- Clothing Material (Polyester),
- PET Bottles,
- PVC (Polyvinyl Chloride) for tubing,
- And many others.

Due to being very present in society, it is important to study oil and gas processes. A significant amount of money is involved in all phases of an oil field's life cycle. Any problem associated with exploration, production, refining and transportation can entail a huge amount of money loss, which is why there is a large investment in techniques to work around these problems.

Another problematic issue involved in the Oil and Gas industry is that the process of exploring, producing and refining are very taxing to the environment, culminating in greenhouse effect intensification and global warming. So environmentally conscious government agencies are imposing rigorous regulations so that all the petroleum related processes cannot damage the environment. How can we maximize production while minimizing environmental harm? This is also a problem to be studied in oil and gas

industry. Works such as [4] follow that line of work, trying to maximize production while avoiding a process called "Flare", which is basically burning excess gas into the atmosphere.

4.1.2 Petroleum Overview

Petroleum, also referred to as Oil and Gas, are hydrocarbons acumulated over million years several meters underground, by the decomposition of long deceased bodies into organic matter. Due to this reason, it is also called "Fossil Fuel". Hydrocarbons are, as implied by the name, organic compunds of Hydrogen and Carbon [32]. The higher the number of carbons in a hydrocarbon, the higher the density of the substance and higher is its boiling point. Adopting the notation used in [3], C_x is read as "a hydrocarbon with a number x of carbons". "Oil" refers to the liquid phase of the mixture and "Gas" refers to the gaseous phase. Various, different type of hydrocarbons, along with water, sand, sulfur and other impurities compose "Crude Oil" which is how Petroleum is found at its initial location, the reservoir. This sparks the need for separation and refining, to transform this mix of hydrocarbons, impurities and water into more refinable, marketable products.

A reservoir consists in underground rocks that are able to retain oil and gas in a certain fixed location. Those rocks are called "reservoir rocks" and tend to have high porosity to be able to "hold" the hydrocarbons. Reservoir rocks tend to be of "clastic" or "carbonite" composition [3]. Since water, which is also present, is denser than hydrocarbons, there must be some structure that stops oil and gas from going up and being replaced by water. Due to this, an underground reservoir consists in "reservoir rocks" and "traps", which are almost impermeable gelogical formation forcing the petroleum's stay at the reservoir rocks [3]. Also important to define a reservoir, is the presence of "source rocks", which are sedimentary rock where the hydrocarbons come from in the first place.

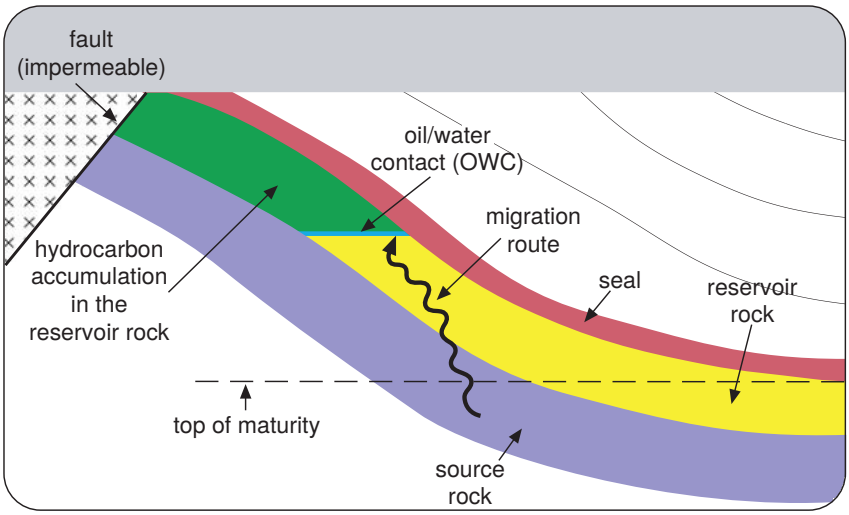

Figure 4 – Schematic representation of an oil underground reservoir. Figure is from [3].

To find these structures to locate possible reservoirs underground is the job of a

well trained team of geologists. A significant number of techniques is avaiable to detect the adequate geological structures that can be labeled as source rocks, reservoir rocks and geological traps. An example is gravimetry, which uses sensor to make an educated guess about the geological environment based on a measure for g, the gravity acceleration in a given region. Another example is Seismography, which consists in using sound to map the underground according to how the sound wave is reflected back at the sonar. More information about these methods, as well as other methods' description, are found in [3].

There are certain properties that describe the petroleum found at a reservoir (also known as reservoir fluid). One of these measurements is the "degree API"; API standing for American Petroleum Institute, the institution which defined this unit.

API is a form of measuring and classifying oil according to its density, and is defined as:

$$API = \frac{141.15}{\gamma_0} - 131.5 \tag{4.1}$$

γ_0 is the specific gravity of the oil found at the reservoir. The Specific Gravity is proportional to the oil density, which makes the API degree measure inversely proportional to it. The larger the API of the oil, the more valuable it is to the market [31]. Heavier hydrocarbon (hydrocarbon with more carbon in each molecule) present in an oil mixture is a decreasing factor for API. API is measured in the field using a calibrated hydrometer [3].

Another parameter that classifies a reservoir is the Gas Oil Ratio (GOR), which is simply the ratio of gas dissolved in the oil. The GOR is important to be analyzed due to changing how a whole field is projected around the reservoir [3]. Gas exploration and oil exploration have different specifications, such as transportation logistics. The five main types of petroleum, according to [3], are:

- **Dry Gas**: Pure gas, composed mainly of 96,3% methane and 3% ethane. Has the lowest specific gravity due to this. Due to being past the critical point in a phase diagram and having no possible liquid phase, it is not possible to extract oil from it.
- **Wet Gas**: Has a small amount of liquid phase, hence the name. Tends to have oil with an API of 60-70. It still has a significant amount of methane (88.7%), but C_2-C_6 hydrocarbons are more abundant than in the dry gas case. The heavier components tend to sell well as oil, due to the high API associated with them.
- **Gas Condensate**: At its initial conditions, Gas Condensate is in gas phase due to pressure and temperature conditions in the reservoir. It has hydrocarbon components which as oil would be profitable, due to a possible API of $50 - 70$. The problem is that normally these liquids are not able to be moved from the reservoir, unless the pressure is maintained above the dew point.

- **Volatile Oil**: This fluid's main phase is liquid in the reservoir. Contains a relatively large fraction of volatile components. The API of the oil tends to be $40 - 50$.
- **Black Oil**: The heaviest main variety of reservoir fluid. It contains a large amount of heavier hydrocarbons C_{7+}. Due to this, it tends to have lower API than the other types (< 40).

There are no definitions for cathegorizing reservoir fluids though, but these are the ones most commonly shown. A more detailed table about these reservoir fluid types are shown in [3].

This work deals with offshore oil production. This means that the adequate apparel was already set up to "produce" the oil, which means bringing the reservoir fluid into the surface and storing it for transportation. The fluid must be separated into oil, water and gas at the end of the operation [3].

4.1.3 Oil Field Life Cycle

When a reservoir is discovered, an "oil field" is stablished. The oil field is owned by a company and is the exploration and production site for the reservoir. The oil field life cycle consists in these phases:

1. Access Gaining
2. Exploration
3. Appraisal
4. Development
5. Production
6. Decomissioning

Access Gaining consists in negotiating with the government to gain rights to explore and produce in the land that contains the reservoir. Before negotiating with the government, an oil company must take into consideration certain technical, political, social, economical aspects of the region where the reservoir is located. Do we have the technical resources to explore the reservoir? How much oil and gas is there? Is the workforce around the region good? Is there any political tension? Is the country where it is located diplomatically aligned with our government? These are some questions which must be taken into consideration before the start of negotiation. A detailed explanation of all the procedures that must be taken is found at [3].

Exploration consists in answering the question: "Is there oil there? How much?"

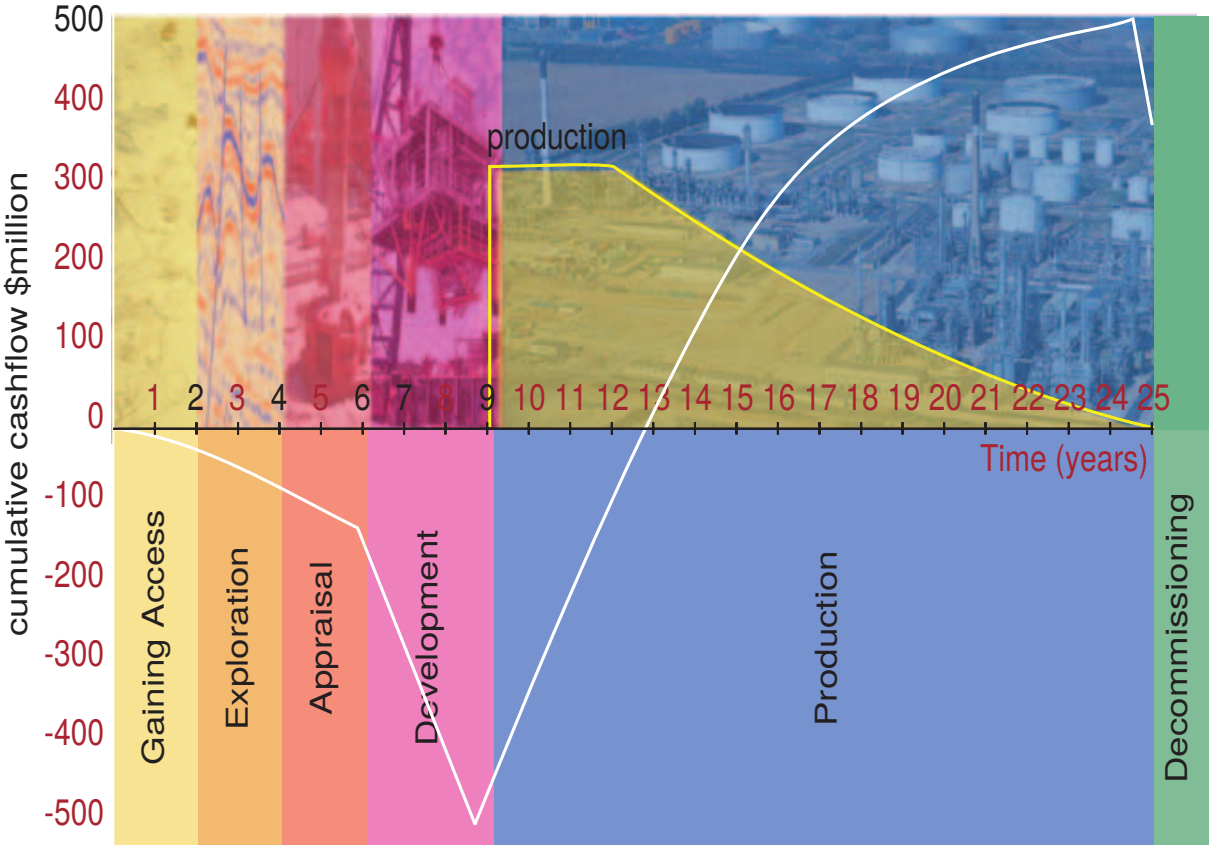

Figure 5 – Timeline of a field life cycle. y axis is the accumulated cash flow. Figure from [3].

Exploration is a high risk activity due to the large amount of money that is spent in it ($100.000.000,00 when drilling is executed, according to [3]). That money would be an empty spending if the reservoir is found to not be profitable. Due to this, a lot of research must be done, some of which are mentioned in Section 4.1.2, before drilling is started. These processes take several years.

Appraisal consists in analyzing the reservoir to see if it is worth exploring, once a profitable amount of hydrocarbon is found by the drilling in the exploration phase. The process to find accurate data about the reservoir size and feasibility, risks involved, possible uncertainty, a cost-benefit analysis, among other things. An oil company has the option to skip that step. Though having quicker imediate gain and lower pre-production cost, it may compromise the profits later. Also, there are companies specialized in exploration and appraisal. Their modus operandi is to sell the discoveries about a certain reservoir to companies who are willing to explore it.

The next step is field development planning (FDP). The FDP consists in stablishing the structures which will be used to produce oil, as well as stablishing the objectives of development, operating and maintenance principles, description of engineering facilities, cost and manpower estimates, project planning and summary of economics and a budget proposal. Once the FDP is approved, all the facilities are designed, fabricated and installed and all equipment is comissioned according to what was planned in the FDP phase.

Production is the only phase where all that is spent is compensated, as shown in Figure 5. All the equipment needed for the oil production is mounted and means to bring the oil to the surface are provided. There are three main sub-phases of production. Build-up period is a period when production increases, along with the cash-flow. The Plateau period is when maximum production is reached. Production facilities are running at full capacity and a constant production rate is mantained. This constant production stays 2-5 years for an oil field [3]. The decline period is the final and usually the longest period, where all wells start declining in production, due to the depletion of the hydrocarbons underground, which are a finite resource.

Decomissioning happens when it is not profitable anymore to produce in an oil field. In other words, when the cash flow associated with that field becomes permanently negative. It includes the abandonment of the field, or tax reduction if that is possible.

Since this work focus on production offshore platforms, those will be detailed in the sections below.

4.1.4 Oil Facilities

There are many phases of oil and gas processing, each of them has different facilities to execute the phase's designed operation. Different companies can take responsibility for each of these processes. In the literature, each of these phases is described as:

- **Exploration**: As described in Section 4.1.3, exploration occurs at the beggining of an oil field life cycle. Exploration is about gathering information about the reservoir whose oil and gas shall be extracted, then drilling the reservoir before the field development is decided.
- **Upstream**: Consists in production and transportation of the oil from the production platform to the refineries.
- **Midstream**: Consists in transforming the crude oil obtained from Upstream into products useful for daily affairs. This process normally involves oil fractioning and distillation. Also incluses the petrochemical phase, where oil will be transformed in different products through chemical reaction.
- **Downstream**: Using the products obtained from Midstream, downstream consists in all processes related to the distribution and selling of the products. (e.g. Gasoline being transported to a gasoline station and gasoline being sold is a downstream process.)

As emphasized in Section 4.1.3, this work focuses on production, which involves the extraction of the gas up until its storage in the oil field.

4.2 Offshore Oil Production Systems

This section describres general components present in oil production facilities. Figure 6 shows an example of a complete platform, which was considered in the work

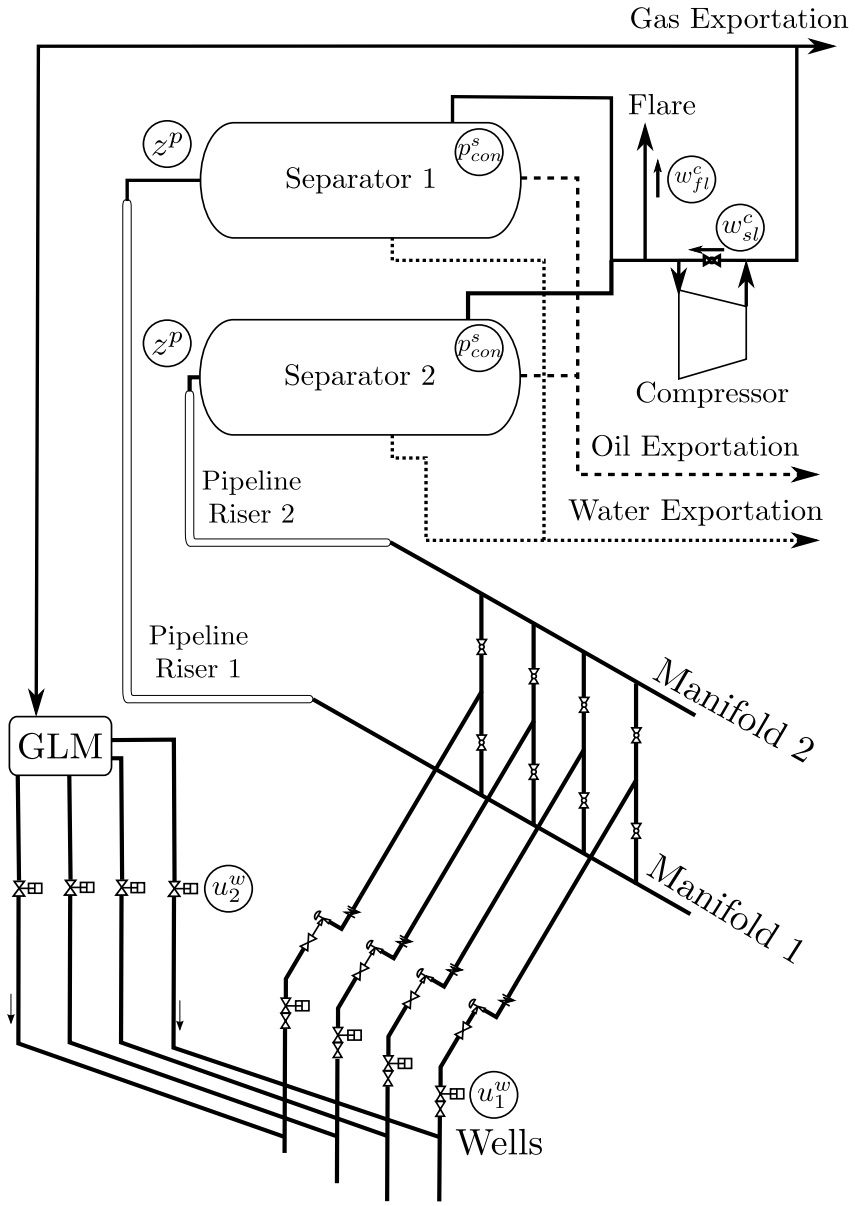

Figure 6 – Example of a diagram of an offshore production facility. GLM stands for Gas Lift Manifold. There are two risers and separators due to the fact that one of the separators is used for testing purposes [3]. Figure from [4].

slu

of [4].

4.2.1 The Wells

A well is an apparatus responsible for conducting oil and gas from the reservoir to the surface. 7 shows the example of an offshore oil well. The tubulation in the well needs

to have the flow capacity for production (or injection) and be robust facing problems such as sand production, corrosion, high pressures or temperatures, mechanical failure and production chemistry issues such as waxes, scales and hydrates [3]. A production platform can have one or more wells. This number is obtained during tests on the appraisal phase of the oil field. A well which is able to produce oil at a commercial rate without help from a lifting system is called a natural flowing well [7].

This work considers a vertical, offshore well, operated using gas lift. Gas lift consists in reinjecting produced gas into the well, thus lowering the density of the produced fluid and easening its ascent, increasing the production flow [3, 4]. A gas-lifted well can be split in two different parts: the annulus, which is the medium where the gas for gas lift is injected, and the tubing, where flows the produced fluid. The gas used for gas lift comes from a valve which controls the amount of gas injected, passes through the annulus and goes to the tubing through an injection valve.

In each well, production tests are performed at least once per month by diverting the production for certain measurements in the test separator. Measures include the tubing head pressure, the flow, the velocity distribution, how it agrees with the simulation model, and others.

4.2.2 Subsea Processing

Some wells are included with subsea processing. Wells tend to also produce water, which is undesirable for commercial applications, so a separator is used to inject water back into the ocean [7]. Sand and other undesirable substances are also handled by the subsea separator. This has shown improvements both in the production and in the top separation efficiency [7].

4.2.3 Manifolds

If more than one well is used, it is expensive to directly send their production to the surface, so they are connected directly to each other using a manifold. A manifold is the component that connects the different wells associated with an oilfield, merging their flows into one. This tubulation gathers all fluids which are being produced by the wells and direct the mixture into a pipeline-riser system. This equipment must be properly designed to resist the intensity of the flow coming from the most productive well.

4.2.4 Pipeline-Riser

This is the structure responsible for bringing the produced reservoir fluid into the surface. This structure can be considered as the blood vessel of an offshore oil platform [7].

In this work, it consists of a horizontal pipeline that receives fluid from the manifold as inflow, and the outlet flow goes to what is called the Riser. The Riser is a vertical

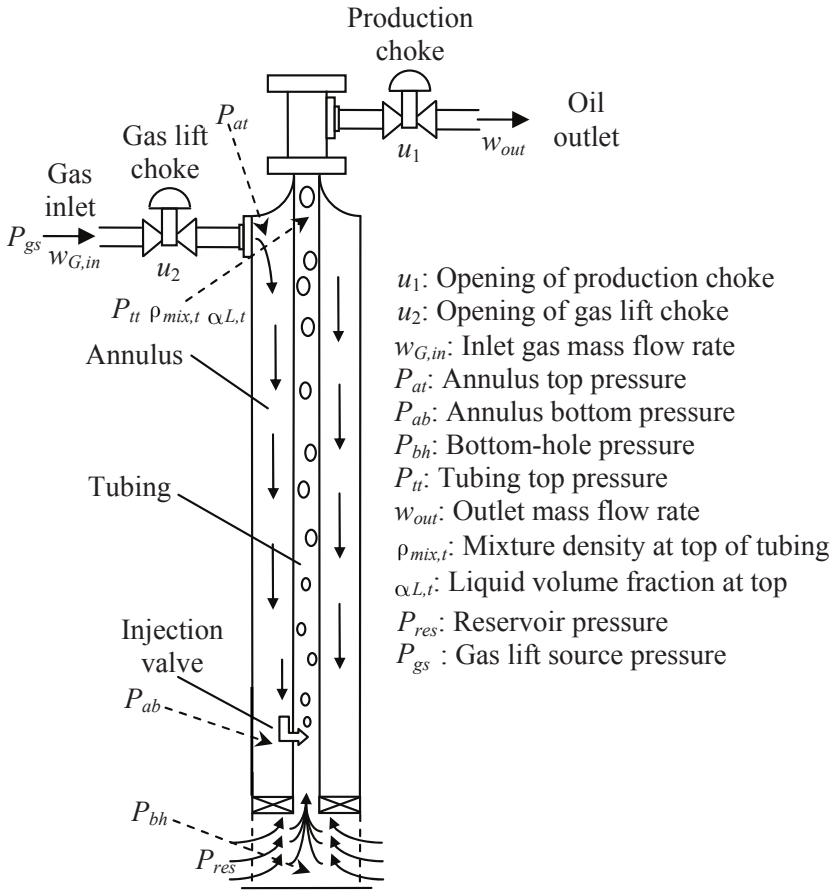

Figure 7 – Schematic representation of the well considered in this work. Adapted from [5].

tubulation used to transport fluid from the pipeline into the surface. These components incur a major cost in the implementation of an oil field, due to the need to be specifically designed to certain temperature and depth configuration [7]. Figure 8 shows an example of pipeline-riser system.

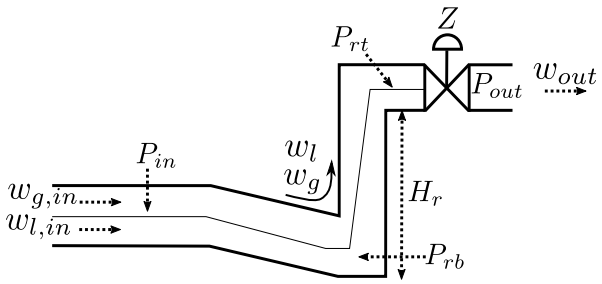

Figure 8 – Representation of a pipeline-riser system. P_{out} represents the pressure in the separator. Obtained from [6].

4.2.5 Separator and Top-side Processing

The flow of a riser can come in three (or more) different phases: oil, gas or water. The separator receives the flow as inlet and has the purpose of separating the flow into oil, gas or water. The outflow will then be sent to the other processes in the top-side processing, and then transported for selling and/or refining, or being discarded in an (ideally) environmental friendly way.

Separators are classified into two-phased, if they separate the flow into gas and liquid, or three-phased, if they separate crude oil into gas, water and oil.

The other units in the top-side processing are defined by a process engineer, which must find the minimum necessary steps to turn crude oil into refinable products. Other processes can include, degassing, dehydration (for oil), dew point conditioning, contaminant removal, compression (for gas), and de-oiling (for water to be disposed).

4.2.6 Flow Assurance Issues

Assuming a complete production platform, there are many issues which can affect production, costing millions of dollars to the oil company. This subsection is based on the information from [7]. Below is a list of effects that can affect the quality of the fow in production.

- **Hydrates**: Crystaline materials where water molecules are mixed with certain gases or gas mixtures, forming at low temperature and elevated pressure. This substance can block gas flowlines. What is geneally used is applying MEG (Mono-ethylene Glycol) at the blocked area.

- **Wax**: Wax is a natural constituent in any crude oil and most gas condensates. They increase the oil viscosity, increase wall roughness, lessen flow and compromise storage. The most usual strategy for Wax removal is pigging [7]. Pigs is an entity used to clean an oil platform's pipes, being described at [3].

- **Asphaltenes**: The heaviest fractions of crude oil. They can precipitate during production due to changes of pressure, temperature and fluid composition. The precipitated particles are then deposited in the pipeline, causing production rate decline and other operational problems. The industrial solution to this problem is avoiding the operation point at which asphaltene is precipitated.

- **Scales**: Deposits of inorganic salts, reducing capacity of the flowline. To deal with scales, chemicals named "scale inhibitors" and "scale dissolvers" are applied.

- **Corrosion**: In fields which produce large quantities of water, pipe corrosion can be a possible issue. Carbon steel is considered an economic solution for this problem.

- **Emulsions**: Emulsions can compromise separation efficiency and the processing facilities, which can cause loss in production. The solution to this is using de-emulsifiers.
- **Slugging Flow**: Happens due to gas and liquid being transported at the same time, more is explained below.

4.2.7 Slugging Flow

Gas and liquid phases normally do not travel at same velocity in the pileline. This is due to differences in density and viscosity. For an upward flow, such as in a riser, the gas phase flows at a higher velocity than the liquid phase. Also, predicting how will a multiphase composition will flow is a complex task, even in a simple pipeline geometry.

Slug flow can be caused by either of these factors [7]:

- Hydrodynamics,
- The upward flow within a riser,
- Irregular surface of seabeds,
- Induction by pigging,
- Gas compression in the annulus of a gas lifted well,
- Accumulation of gas at the bottom of a long well.

The slugging induced by the presence of a riser is one of the most important flow assurance challanges. In slugging, liquid accumulates in the entrance to the riser, blocking gas entrance, leading to the compression of the gas in the pipeline. Each instance of this accumulation is called a slug. This happens if gas and liquid velocities are sufficiently low. The slug continues to grow as the hydrostatic head of the liquid in the riser is higher than the pressure drop over the riser [7]. When the pressure drop over the riser exceeds the hydrostatic head due to slug accumulation, the liquid is pushed out of the riser and, when all liquid has left the riser, liquid falls back into the bottom due to low velocity and starts to accumulate again.

The presence of slugging depends on inflow conditions, the topside chocke valve, geometry and dimensions of the riser as well as the separator pressure. If a riser is designed to avoid slugging, only problems related to the separator pressure, topside choke valve and inflow conditions remain. A slug can be detected as a pressure oscillation in a sensor.

The conventional anti-slug solutions available have either operational problems or no economic viability, so studies have arisen for feedback control solutions. In this case, the controlled variable would be the pressure in the pipeline, though existing anti-slug control

systems are not operating in practice due to robustness problems because of changes in the model and disturbances. This justifies the use of the adaptive control strategy described in the previous chapter, for not assuming full knowledge of the process model. The dynamics involved in the slug are represented in Figure 9

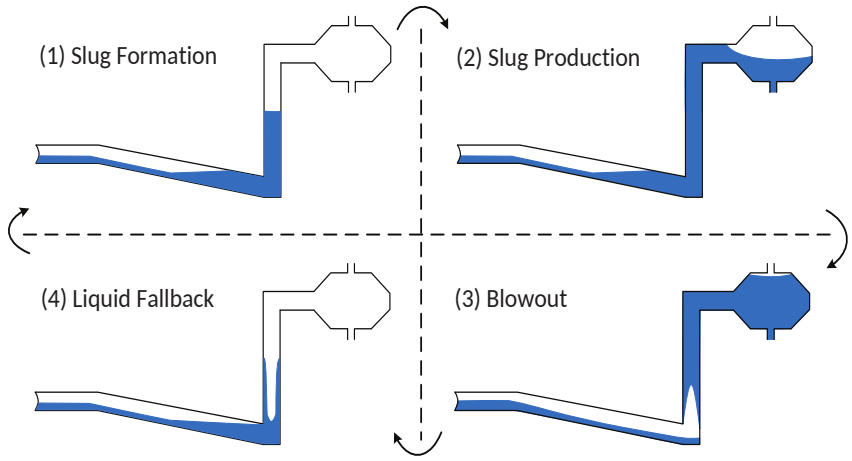

Figure 9 – Representation of a slug flow. Obtained from [7].

4.3 Well Model

This section describes the well model which is considered for the experiments in this work. This model originates from [5]. The model considers only the liquid and gas phases, treats oil and water as the same phase, and is a system of state equations consisting in:

$$\dot{m}_{G,a} = \omega_{G,in} - \omega_{G,inj} \tag{4.2}$$

$$\dot{m}_{G,tb} = \omega_{G,inj} + \omega_{G,res} - \omega_{G,out} \tag{4.3}$$

$$\dot{m}_{L,tb} = \omega_{L,res} - \omega_{L,out} \tag{4.4}$$

- x is the nature of the variable, with m being the mass and ω the mass flow.
- y is phase represented by the variable, with G being the gas and L the liquid/oil phase, since the model assumes no water phase.
- z is the location of the variable 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. $m_{G,a}$, $m_{G,tb}$ and $m_{L,tb}$ are the state variables considered in this model (in kg). This model is represented by Figure 7. $m_{G,a}$ is the total mass of gas that is currently in the annulus of the well. This is the gas that comes from the gas lift source, as represented by its state equation: $\omega_{G,in}$ is the mass flow (kg/s) of the gas coming from the source

into the annulus. $\omega_{G,inj}$ is the mass flow (kg/s) of the annulus gas coming into the tubing. $m_{G,tb}$ is the total mass of gas that is currently in the tubing of the well. It has two sources, $\omega_{G,inj}$ and $\omega_{G,res}$, which is the gas mass flow that comes from the reservoir. $\omega_{G,out}$ is the outlet gas mass flow, which leaves the well tubing into the platform or a manifold. $m_{L,tb}$ is the mass of liquid in the well tubing. The liquid comes from the reservoir with mass inlet flow (kg/s) $\omega_{L,res}$ and leaves with outlet mass flow $\omega_{L,out}$.

These mass flows are computed using Bernouilli's orifice equation:

$$\begin{aligned}
\omega_{G_{in}} &= K_{gs} u_2 \sqrt{\rho_{G,in} \max(P_{gs} - P_{at})} \\
\omega_{G_{inj}} &= K_{inj} \sqrt{\rho_{G,ab} \max(P_{ab} - P_{tb})} \\
\omega_{out} &= K_{pr} u_1 \sqrt{\rho_{mix,t} \max(P_{tt} - P_0)} \\
\omega_{res} &= PI \max(P_{res} - P_{bh}) \\
\omega_{L,res} &= (1 - \alpha^m_{G,b}) \omega_{res} \\
\omega_{G,res} &= \alpha^m_{G,b} \omega_{res} \\
\omega_{L,out} &= (1 - \alpha^m_{G,t}) \omega_{out} \\
\omega_{G,out} &= \alpha^m_{G,t} \omega_{out}
\end{aligned}$$

These variables follow the notation according to Figure 7. K_{gs},K_{inj}, and K_{pr} are experimental variable parameters which depend on the practical application. P_0 is the outlet pressure. $\alpha^m_{G,b}$ is the mass fraction of the bottom flow, and $\alpha^m_{G,t}$ is the mass fraction of the outlet flow. $\alpha^m_{G,b}$ is assumed to be constant and u_1 and u_2, the choke valve opening and gas lift valve opening, respectively, are the model inputs and manipulated variable candidates. $\alpha^m_{G,t}$ is calculated as:

$$\begin{aligned}
\alpha^m_{G,t} &= \frac{(1 - \alpha_{L,t}) \rho_{G,t}}{\alpha_{L,t} \rho_L + (1 - \alpha_{L,t} \rho_{G,t})} \\
\alpha_{L,t} &= 2\overline{\alpha}_L - \alpha_{L,b} \\
\alpha_{L,b} &= \frac{\omega_{L,res} \rho_{G,tb}}{\omega_{L,res} \rho_{G,tb} + (\omega_{G,inj} + \omega G, res) \rho_L} \\
\overline{\alpha}_L &= \frac{m_{L,tb} - \rho_L V_{bh}}{V_t \rho_L}
\end{aligned}$$

where $\overline{\alpha}_L$ is the average liquid fraction inside the tubing, V_{bh} is the assumed volume at the bottomhole, V_t is the volume in the tubing, and ρ_L, the liquid density, is assumed to be constant. The other densities present in the previous equations are variable and calculated as follows, derived from either the ideal gas law or the definition of density:

$$
\begin{aligned}
\rho_{G,ab} &= \frac{P_{ab} M_G}{R T_a} \\
\rho_{G,in} &= \frac{P_{gs} M_G}{R T_a} \\
\rho_{G,t} &= \frac{m_{G,tb}}{V_t + V_{bh} - m_{L,bh}/\rho_L} \\
\overline{\rho}_{mix} &= \frac{m_{G,tb} + m_{L,tb} - \rho_L V_{bh}}{V_t} \\
\rho_{G,tb} &= \frac{P_{tb} M_G}{R T_t} \\
\rho_{mix,t} &= \alpha_{L,t} \rho_L + (1 - \alpha_{L,t}) \rho_{G,t}
\end{aligned}
$$

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 valve equations depend on the pressures, which are calculated as follows:

$$
\begin{aligned}
P_{at} &= \frac{R T_a m_{G,a}}{M_G V_a} \\
P_{ab} &= P_{at} + \frac{m_{g,a} g L_a}{V_a} \\
P_{tt} &= \frac{\rho_{G,t} R T_t}{M_G} \\
P_{tb} &= P_{tt} + \overline{\rho}_{mix} g L_t + F_t \\
P_{bh} &= P_{tb} + F_b + \rho_L g L_{bh}
\end{aligned}
$$

P_{at} ans P_{tt} are derived from the ideal gas law. P_{ab} is P_{at} plus the gas's hydrostatic pressure. P_{bh} and P_{tb} contain not only the hydrostatic pressure imposed by the liquid, but F_b and F_t, which are the pressure loss due to friction in the bottom-hole and the tubing. L_a is the length of the annulus, L_t is the length of the tubing, and L_{bh} is the assumed length of the bottom hole. P_{res}, P_{gs} and P_0 are considered to be disturbances in this work. These pressures depend on exogenous factors such as the reservoir, the gas lift source, and the manifold, respectively, and can be potentially variable, but are considered to be initially constant for modeling purposes. The pressure loss in the bottom-hole F_b is assumed to be constant, but the pressure loss due to friction in the tubing F_t is calculated as:

$$
\begin{aligned}
F_t &= \frac{\lambda_b \rho_L \overline{U}_{l,b}^2 L_{bh}}{2D_t} \\
\frac{1}{\sqrt{\lambda_t}} &= -1.8 \log_{10}\left(\left(\frac{\epsilon}{3.7D_t}\right)^{1.11} + \frac{6.9}{Re_t}\right) \\
Re_t &= \frac{\overline{\rho}_{mix}\overline{U}_{m,t}D_t}{\mu} \\
\overline{U}_{m,t} &= \overline{U}_{sl,t} + \overline{U}_{sg,t} \\
\overline{U}_{sg,t} &= \frac{4(\omega_{G,in} + \alpha_{G,b}^m \overline{\omega}_{res})}{\rho_{G,t}\pi D_t^2}
\end{aligned}
$$

The above equations are derived from Haaland's solution to the Colebrook-White equation (1983) for the calculation of the friction factor of the tubing λ_t. 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. $\overline{U}_{sl,t}$ is the average superficial velocity of the liquid phase, assumed to be constant. D_t is the tube's diameter, μ is the viscosity of the fluid. $\overline{\omega}_{res}$ is the assumed average inlet flowrate, which is obtained experimentally and is constant for dynamics simplification purposes.

This model is stable and is chosen as a case study due to initial tests on the controllability of plants of oil and gas nature with the controller proposed in Chapter 2. Is the echo state network capable of approximating complex non-linear dynamic phenomena such as friction loss? This application is also found in [2].

The parameter values are set as follows for the simulation:

Parameter	Value
M_g: Molecular Gas Weight	0.0195 kg/mol
μ: Viscosity	3.64×10^{-3} $Pa.s$
ρ_L: Liquid Density	970 kg/m^3
ϵ: Piping Superficial Roughness	2.8×10^{-5} m
T_a: Annulus Temperature	350 K
V_a: Annulus Volume	30.16 m^3
L_a: Annulus Lenght	1500.0 m
D_a: Annulus Diameter	0.16 m^2
T_t: Tubing Temperature	350 K
V_t: Tubing Volume	18.11 m^3
L_t: Tubing Lenght	1500.0 m
D_t: Tubing Diameter	0.124 m^2
$\overline{U}_{sl,t}$: Tubing Avarage Liquid Phase Velocity	0.163 m/s
F_b: Friction Loss in Bottom Hole	313 Pa
L_{bh}: Length below Injection Point	75 m
$\alpha^m_{G,b}$: Gas Mass Fraction at Bottomhole	4.58×10^{-2}
$\overline{\omega}_{res}$: Average Production Mass Flow	2.0 kg/s
P_r: Reservoir Pressure	250×10^5 Pa
PI: Reservoir Production Index	2.47×10^{-6} $kg/(s.Pa)$
K_{gs}: Gas-Lift Choke Constant	1.6×10^{-4} $kg/(s.Pa)$
K_{inj}: Injection Valve Constant	1.6×10^{-4} $kg/(s.Pa)$
K_{pr}: Production Choke Constant	1.4×10^{-3} $kg/(s.Pa)$

4.4 Pipeline-Riser Model

The pipeline-riser model utilized in this work was idealized in [18]. The same notation as the well model is used for the pipeline-riser model. This model uses the same consideration as the well model: the model considers only a biphasic flow consisting of liquid (oil and water) and gas phases. 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. [18] demonstrates that this model approximates well to an equivalent riser modeled in the OLGA commercial simulator. The

following state equations are considered:

$$\dot{m}_{G,p} = \omega_{G,in} - \omega_{G,lp} \tag{4.5}$$

$$\dot{m}_{L,P} = \omega_{L,in} + \omega_{L,lp} \tag{4.6}$$

$$\dot{m}_{G,r} = \omega_{G,lp} - \omega_{G,out} \tag{4.7}$$

$$\dot{m}_{L,r} = \omega_{L,lp} + \omega_{L,out} \tag{4.8}$$

$$\tag{4.9}$$

The states represent:

- The total gas mass in the horizontal piping, $m_{G,p}$.
- The total liquid mass in the horizontal piping, $m_{L,p}$.
- The total gas mass in the riser, $m_{G,r}$.
- The total liquid mass in the riser, $m_{G,r}$.

$\omega_{G,in}$ is the inlet gas mass flow rate. $\omega_{G,lp}$ is the gas mass flow rate of the fluid leaving the pipeline and entering the riser. $\omega_{L,in}$ is the inlet liquid mass flow rate. $\omega_{L,lp}$ is the liquid mass flow rate of the fluid leaving the pipeline and entering the riser. The acronym *lp* stands for "low point". $\omega_{G,out}$ and $\omega_{L,out}$ are the gas and liquid outlet mass flowrate, respectively, which then goes to a top-side separator.

In the simulations featured in this work, $\omega_{G,in}$ and $\omega_{L,in}$ are boundary conditions assumed to be constant, though these variables could be the outflow of a manifold connected to multiple wells.

To calculate the outlet flow of the riser, the following equations are used:

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

z is the production choke opening, which is the only possible manipulated variable in this model. 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, regulated to approximate the dynamic of the model to a real life riser. The top of the riser could be connected to a separator. To calculate $\alpha^m_{L,t}$, ρ_t and P_r, these equations are used:

$$\alpha_{L,t}^{m} = \frac{\alpha_{L,t}}{\rho_t}$$
$$\rho_t = \alpha_{L,t}\rho_L + (1 - \alpha_{L,t})\rho_{G,r}$$
$$P_r = \frac{\rho_{G,r} R T_r}{M_G}$$

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 volume fraction of liquid at que top of the riser. $\rho_{G,r}$ is the gas density at the riser. ρ_L is the liquid density. To calculate $\alpha_{L,t}$ and $\rho_{G,r}$ these equations are used:

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

V_r is the total volume of the riser. L_r represents 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:

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

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. As demonstrated by this equation, gas passage is not allowed in the model when the height in the pipeline become higher than the critical height. h_p is calculated as follows:

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

$\overline{\alpha}_{L,p}$ 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 in relation 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:

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

$\overline{\alpha}_{L,r}$ 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:

$$\lambda_x = 0.0056 + 0.5 Re_x^{-0.32} \tag{4.10}$$

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:

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

μ is the viscosity of the fluid, which is assumed to be constant. The model's parameters used are exactly the same as [7] and [18], due to it ensuring that the riser will endure severe slugging in these conditions. The inlet mass flow is assumed to be 9 kg/s, in which 8.36 kg/s are from the liquid phase and 0.64 kg/s comes from the gaseous phase. This implies gas accumulation in the low point due to the low velocity of the gas phase. In these conditions, it is shown in [18] that the riser is only capable of attaining open loop production stability (detected by a constant pressure, with $z \leq 0.05$). This low value for production is not adequate. According to the biffurcation diagrams in [18], this model's limit cycle has larger amplitude than the reference model from the commercial software OLGA used.

4.5 Control Applications in Oil Wells and Risers

In the literature, a lot of applications of control strategies involving production platform plants can be found.

An example of a systemwide control, involving all the variables of the whole platform in a simplified model, is [4]. [4] sucessfuly uses optimal control theory applied in a whole oil production platform to maximize production and reduce flare. The production maximization (using an economical objective function as the performance function of the controller) and the smart tracking (a tracking cost function penalizing the flare) manage to maintain optimal production with no need for flaring the gas. The model of the platform used also assumes the scheduled maintenance of the compressor.

[33] exposes examples of systems from the oil industry that have variable delays and proposes an initial method on how to solve them. As demonstrated by [14], the controller proposed in Chapter 3 of this work is also able to thoroughly deal with plants with variable transport delay.

[20] provides a real application of anti-slug control in both the Campos and Santos basin. The controller proposed in this paper is divided into three components:

- Diagnostic Module - For detection of severe slugs. Detected by oscillating pressures.

- Protection Module - For minimizing the damage done by slugs into external equipment. A data-driven emergency choke-closing strategy.
- Anti-Slug Control - Module Responsible for minimizing or eliminating the slugs.

The work also proposes the use of some advanced control strategies such as the ONFC (Online Neuro-Fuzzy COntroller), the gamma algorithm, and the commutational use of three PIDs.

[19] utilizes an adaptive + supervisory control strategy. The supervisory control consists in basically an oscillation detector which increases the bottomhole pressure in presence of oscillations, and decreases the setpoint in the abscence of oscillation, therefore driving the oil platform close to its limit. For the oscillation detection, the algorithm checks if the frequency components of a fast fourier transform (FFT) have more energy at the neighborhood of the slug frequency. The adaptive control strategy utilized is derived from the Model-Reference adaptive control, which is called Robust Adaptive Control [19]. The controller assumes a state-space linear model and uses a closed loop LQR controlled system as the reference model.

[21] utilizes a control strategy which uses the derivative of the pipeline pressure to supress the slug. The resulting strategy is a simple, linear controller whose purpose is to bring the derivative of the bottom pressure to zero. [21] also shows a mathematical proof on why a stable bottom hole pressure is equivalent to no slug flow. [21] also estimates the variation in the bottom-hole pressure based on a linear filter of the choke pressure, thus eliminating the need to measure the pressure at the low-point.

Besides introducing the well model used in this work, [5] presents a robust control design for unstable regions in the well. The idea for this controller is to find the linear parameters by solving an H_∞ problem. Also, a controllability analysis is made, which finds that the best performance of disturbance rejection in a SISO (Single Input, Single Output) controller is when the bottomhole pressure is used as the controlled variable and the well choke opening is used as the manipulated variable.

[6] utilizes the same riser model as [18] and this work's simulation model, but the parameters are changed so the limit cycle happens when $z = 0.15$, in which z is the production choke opening at the top of the riser. [6] focuses on the use of nonlinear observers for the task of estimating the states of the system based on the riser top pressure, so that control could be applied directly to the system states. It tests techniques such as the Unscented Kalman Filter(UKF), which proved to be not robust enough for the application. [6] also tests the use of a high-gain Luenberger observer, and an alteration of UKF which is called Fast Unscented Kalman Filter. The fast UFK is simply the use of a coordinate transformation which P_r, the measured variable, is used as a state instead of m_{gr}. This is shown to increase robustness of the UKF. The control of the riser utilized in this work managed to stabilize the pressure at an operation point at which $z = 0.2$. This

was the maximum production that could be achieved according to the paper.

[34] utilizes the same model as [18] with the same biffurcation diagram as [6] (open-loop stability at $z = 0.15$). This paper tries using output feedback linearization to stabilize the slugging flow in the riser, managing to successfully stabilize the riser at $z = 0.6$, which is a good result.

For the control strategy of the well, this work uses the results from the controllability analysis featured in [5], which argues that the best controllability happens when u_1, the well outlet choke opening is used as the manipulated variable, and p_{bh}, the bottom-hole pressure, is used as the controlled variable. u_2, the gas-lift valve in which gas in injected into the annulus, is fixed at $u_2 = 0.4$, as per [5]. This also serves as a rough comparison to the performance shown in [5], though the parameters used for this work are slightly different. The capacity of reference tracking and disturbance rejection in this model using the control strategy depicted in Chapter 3 will be tested.

For the control strategy involving the pipeline-riser system, the controlled variable used is the p_p, which is the pressure at the pipeline, and the manipulated variable is z, the production choke valve. The slug flow in the configuration of the model in [18] is more severe than in [6] and [34], with maximum open-loop stability at $z = 0.05$. The controller will test the lowest pressure in which the riser can be stabilized with, in turn, testing the maximum choke opening z in which a maximum operating point can be maintained.

4.6 Summary

In this chapter, oil and gas basic concepts were briefly introduced to the reader. We have demonstrated the whole oil field lifecycle and the facilities involved with oil production and trasnportation to refinery facilities (upstream), the fine processing in the refineries and petrochemical facilities (midstream) and the distribution and selling of the end-user products (downstream). We detailed the functioning of offshore production system's components, such as wells, separators, risers, and the topside processing unit. We showed the well and pipeline-riser models which will be considered for the simulation experiments featured in Chapter 5, and the control problems involved with each plant. At the end, a brief review on production platform control methods was made, which influenced the choice of manipulated variables and controlled variables which will be utilized in Chapter 5.

5 Experiments and Results

In this chapter, the experimentation related to the Oil Well and Pipeline-Riser presented in Chapter 4 is described. Here it is shown how the simulation was implemented, the metrics used, the experimental process performed, and the obtained results. All the experiments related to the well model are obtained from [2].

5.1 Implementation

The whole experiment was implemented using the following tools:

- NumPy
- CUDAMat
- JModelica

NumPy is the numerical computation library generally used in Python. It was used for the integration of the control loop in general.

CUDAMat is a GPU calculation library for Python, utilizing GPUs for parallel calculations. It is used for all the computation involving the Echo State Networks, since matrices with high dimensions are involved.

JModelica is a type of Modelica framework. It is implemented using python, C and java, though its API is mainly implemented in Python. In other words, a Python code is used for compiling a code written in JModelica, that makes the communication with the other components in this work easier. Modelica is a language used for model representation. The model is described by the language and then, while in simulation, it is solved by a Sundials package solver, namely CVode. All the plants used in this work are represented using JModelica.

For the communication, the controller, implemented using the Python libraries, only receives feedback from the system and can influence it for each sample period T_s. To emulate that, for each control action $u[k]$ at timestep k, a Jmodelica simulation is run for a duration of T_s. The input is held constant for each simulation. A save/load logic was also implemented, in which the weights of a neural network is saved in a .mat file to be used in later experiments.

5.2 Metrics

The performance of the controller in the well is evaluated qualitatively in terms of:

- The steady state error: $e_{track} = y[k] - \hat{y}[k]$, $\forall k$ where $y[k-1] \approx y[k]$.
- The overshoot: The difference between the maximum pressure and the steady state pressure.
- The settling time: How long does the closed loop system take to reach setpoint.
- The damping: How oscillatory the response is.

The reservoir is generated at random so, for making the selection easier and to avoid inspection of each simulation plot, they are selected in terms of the quadratic sum of the tracking error of all simulation time N.

$$E = \sum_{k=0}^{N} e_{track}^2[k] \tag{5.1}$$

For evaluating the training algorithm, the following metric is used:

$$e_{mean}[k] = \frac{k}{k+1} e_{mean}[k-1] + \frac{e_{train}[k]}{k+1} \tag{5.2}$$

Here,e_{mean} is a moving average of the training errors of all the simulation, and e_{train} is the quadratic training error at time step k. The resultant curve is called the "learning curve" in this work.

The evaluation of the performance in the riser consists in a stability test. What is the highest production (operation point with the largest value of z) that can be stable in the closed loop system?

5.3 Experiment Overview

In the following, the procedures for model initialization and finding model hyper-parameters (spectral radius, input scaling, etc.) as well as for performance evaluation are presented.

5.3.1 Reservoir Selection

As an Echo State Network's weights are initialized randomly, the performance of the control system may vary. A method of selecting a good reservoir for experimentation needs to be applied. A necessary condition for a good weight configuration is that, in a constant setpoint signal, the closed loop system can be stabilized. The selection will folow these steps:

1. The initial values for the parameters γ(the leak rate) ,α (the initial condition for the covariance matrix P of RLS), δ (the timestep delay), f_i^r (the input scaling),f_b^r (the bias scaling), ρ (the spectral redius of the reservoir's weights matrix), ψ (the connectivity of the reservoir) and T_s (the controller sampling time) are arbitrarily selected for this experiment, The number of neurons N is selected according to how well the network can train the inverse model, so that underfitting is avoided.

2. The echo state networks which managed to stabilize the system will be evaluated according to Equation 5.1 described in the previous section. The reservoir which has the highest value for the total tracking error (Equation 5.1) is selected for the next experiment.

5.3.2 Optimizing Parameter Selection

After the reservoir configuration is selected, we need to answer the question: Is it possible to boost the performance of this system? The initial parameters (γ,α, δ, f_i^r,f_b^r, ρ, ψ, T_s) were selected arbitrarily, so the parameters are not necessarily optimal. As the reservoir is randomly generated, the number of neurons and its spectral radius should not change after Procedure in Section 5.3.1, and thus, are left fixed here. The next parameters to tune are the timestep delay δ and the leak rate γ using a grid search process. A grid search through a set of values would be tested. The configuration with the best result in terms of the total sum of the error is chosen. After that, the next pair of parameters experimented would be the input scaling f_i^r and the bias scaling f_b^r, with exactly the same procedure. The resulting parameter configuration would be used for the next experiment.

5.3.3 Final Performance Tests

This step of the experiment is merely a performance showcase for the resultant weights and tuning parameters. It is important to show how the closed-loop system behaves in terms of disturbance rejection and setpoint tracking, so the following tests are made in the well:

1. *Setpoint Tracking, no disturbance*: An arbitrarily chosen signal is used for tests, which covers the possible output space.

2. *Small Disturbance Rejection*: The operation point is fixed, but disturbances in the model are applied. The disturbances are small in magnitude.

3. *Large Disturbance Rejection*: The operation point is also fixed, but larger disturbances is applied, this is to test if the control loop is able to converge even though the model has drastically changed.

4. *Setpoint Tracking + Small disturbance rejection*: The mix of step 1 and 2. This is to test if the system is able to complete both tasks at the same time.

5. *Setpoint Tracking + Large disturbance*: same as above, but since a larger disturbance is involved, some setpoints may not be able to be reached due to violating system constraints. This situation tests how it reacts to windup, which is error accumulation due to the inability of reaching a certain setpoint. In windup, this error would greatly influence the behavior after setpoint change. Theoretically, this controller should not be affected by windup, since no error integration occurs in the online-learning control strategy.

For the riser, we will gradually decrease the setpoint p_{in}, which is the pressure at the pipeline, so that we find a p_{in} in which the system becomes oscillatory. Then, we will try to set p_{in} into a higher value to see if the slugging flow is able to be stabilized by this setpoint change. In a way, this test is similar to the supervisory control featured in [19].

5.4 Results and Discussion

In this section, the results of the experiments will be presented.

5.4.1 The Well

All the experiments in this section were also used in [2].

The parameters used in this simulation, obtained according to procedures described in the previous section, were:

Parameter	Value
γ: Leak Rate	0.3
ρ: Spectral Radius of $\mathbf{W_r^r}$	0.999
δ: Prediction Timesteps	3
ψ: Connectivity of $\mathbf{W_r^r}$	1
N:Number of Neurons	1000
f_i^r: Scaling Factor of $\mathbf{W_i^r}$	0.5
f_b^r: Scaling Factor of $\mathbf{W_b^r}$	0.1
T_s: Sampling Time	10 s

The number of neurons $N = 1000$ was the number of neurons in which the model would not underfit. The sampling time T_s was chosen arbitrarily and shown to have better performance in simulations. ρ was set to 0.999 due to a empirical condition in which the model can show echo state. [27]. It was shown from previous works [14] that the

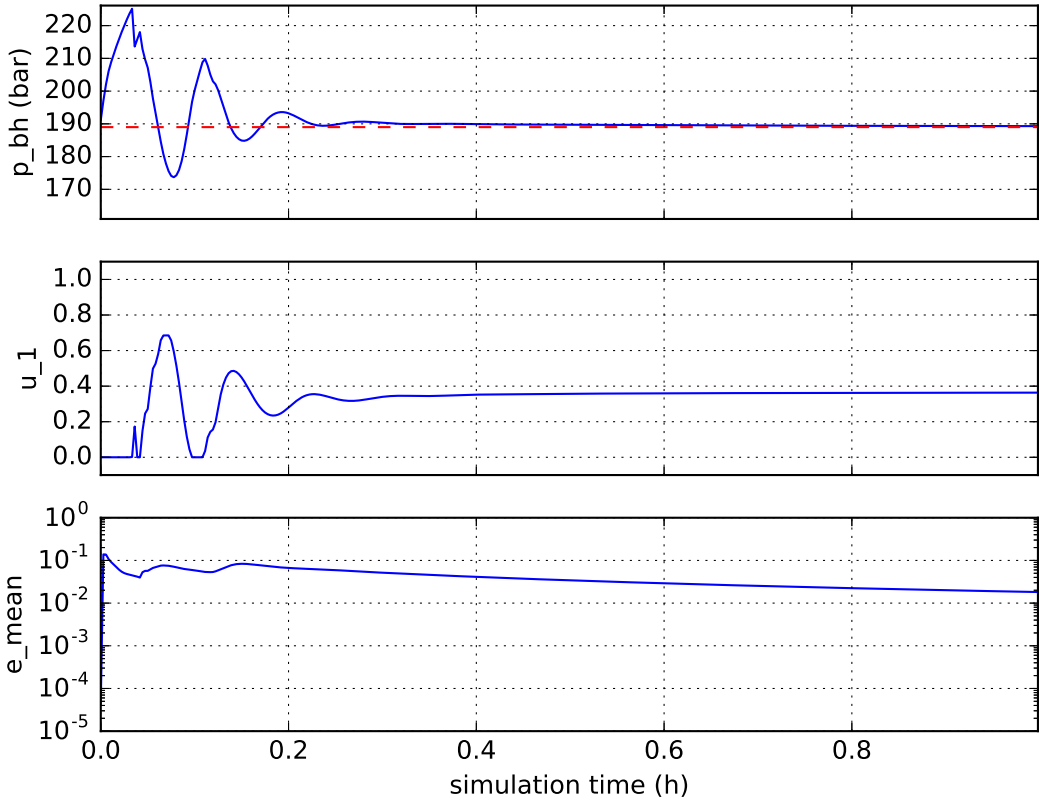

Figure 10 – First hour of the tracking experiment simulation. For the first subplot, the dashed line is the desired bottom-hole pressure and the solid line is the actual p_{bh}. u_1 is the choke opening and $e_{mean}[k]$ is the moving average of the training error at time [k], defined by equation 5.2. Figure from [2].

connectivity of a neural network contributes little to the performance of the controller, so $\psi = 1$ is chosen.

Random reservoirs were tested according to the criteria shown in the previous section. The reservoirs were tested over 3000 timestep simulations. After obtaining a sufficiently "good" reservoir, it was saved and used in the rest of the experiments.

At the parameter tuning phase, it was shown that, between $1 \leq \delta \leq 50$, the system performs better for timestep delay $\delta = 3$. For the leak rate $0 < \gamma \leq 1$, the optimal value was $\gamma = 0.3$. This value allows the state of the Echo State Network to have enough memory of the inputs of the system. f_i^r and f_b^r were chosen arbitratiely. α, the initial condition for the values at the main diagonal at the covariance matrix P of the RLS, was set to 1. In [14], $\alpha = 10$ was utilized for their applications, but this same value did not work for this application probably due to the nonlinearities of the well model.

The plant output $y[k]$ is scaled from [165, 193] bar to [0,1] before feeding it as input to the ESN.

Figures 10 and 11 represent the result of the first experiment: Tracking with no

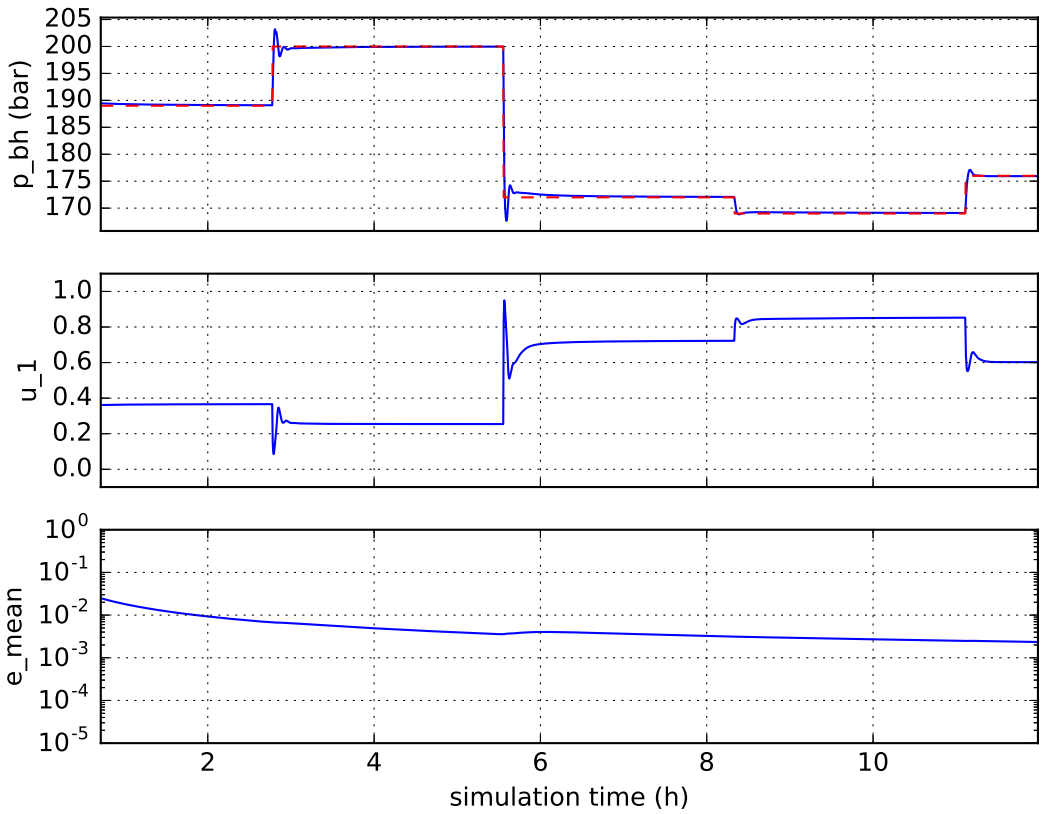

Figure 11 – Tracking experiment simulation. This plot is a direct continuation of the plot in Figure 10. Figure from [2].

disturbance. Figure 11 shows that, when the learning parameters converge (shown by the e_{mean} curve converging), tracking can be achieved with little overshoot. That shows that the echo state network is capable of learning the inverse model given the plant output space given in this figure. Figure 10 shows a strong defect in the strategy described in this work: while the inverse model is not yet learned (shown by the oscillatory behaviour at the beggining of the learninhg curve), the system can be led to hazardous states. This result is actually present (but omitted, for the sake of visualization) in all subsequent results. A solution to this problem could be using supervisory control: If the system reaches a certain undesired value, a "watchdog" logic could shut down the control network and change the control strategy to a simple PI or an open-loop failsafe operation mode. Data would be given to the learning network and training would go on as normally. The echo state controller could be back into operation after the system is back to a non-hazardous point. This has yet to be tested and could be used in future works.

Figures 12 and 13 represent the disturbance rejection only experiments. A disturbance was applied in the gas-lift inlet pressure (p_{gs}). A large one, for Figure 13, and a small one, for Figure 12. It is shown, by Figure 13, that the system does not deviate much from

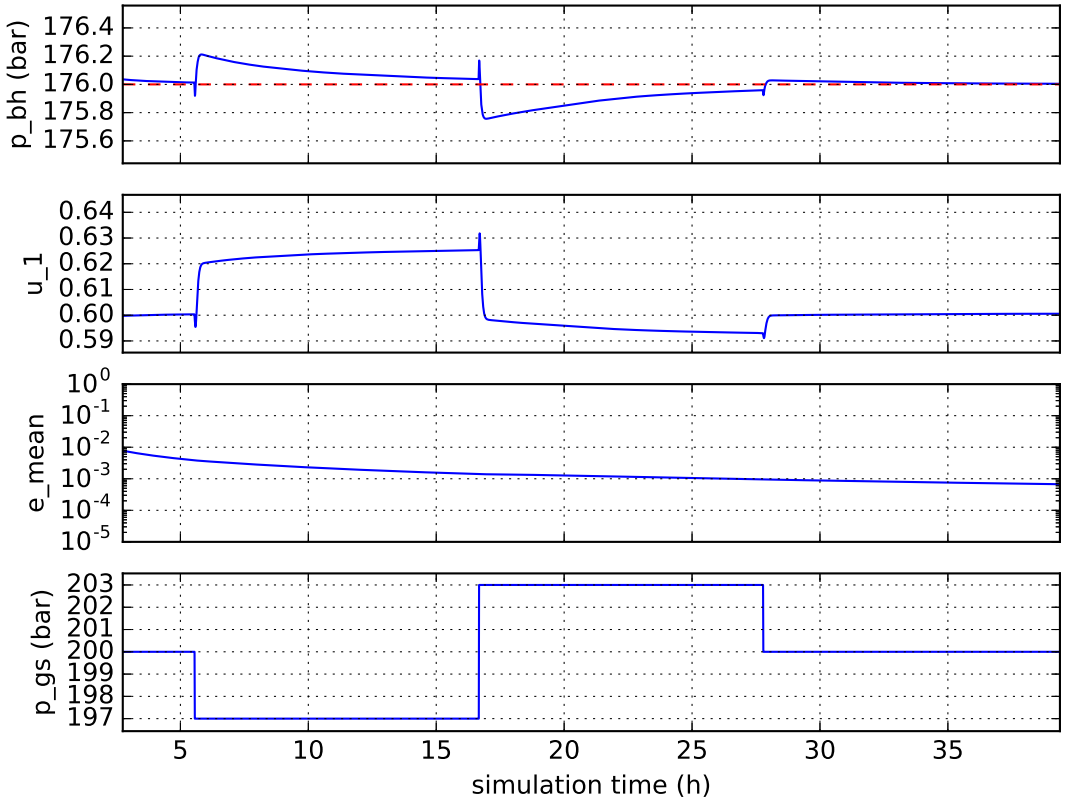

Figure 12 – Simulation of disturbance in the gas lift inlet pressure p_{gs} of -3 and of $+3$ bar at times: $5,5$h and $16,6$. The disturbance ceases at $27,7$h. The dashed line is the desired value while the solid line is the output value.Figure from [2].

the setpoint, even though the disturbance is as large as 30 bar. When the large disturbance is applied, the resultant deviation from the setpoint is rather small, of only 2 bar. This is due to the controller reacting quickly to when the disturbance is applied, almost as it tries to "guess" which value would take it back to the setpoint. As a counterpoint, both figures 12 and 13 show that it is very probable that the "guess" of the controller is wrong, which is why there is deviation in the first place. The plant takes long to converge to the initial operation point in both cases. This is due to, as disturbances can be seen as changes in the model, the network having difficulty to "forget" the previous model and adapt to the new one. This robustness depends on the value of the maximum forgetting factor described in Chapter 3. Though, sufficiently low values of the forgetting factor λ_{max} can hinder the closed loop steady state stability. The value used in this work was 0.9999, which was the lowest value for $\lambda_{\max}$ that did not instabilize the system.

Figures 14 and 15 show the experiment made for solving the tracking + disturbance problem. It is shown that having disturbances of small or large magnitudes does not affect the tracking performance. Figure 15 also shows an instant where the wellhead choke u_1 is

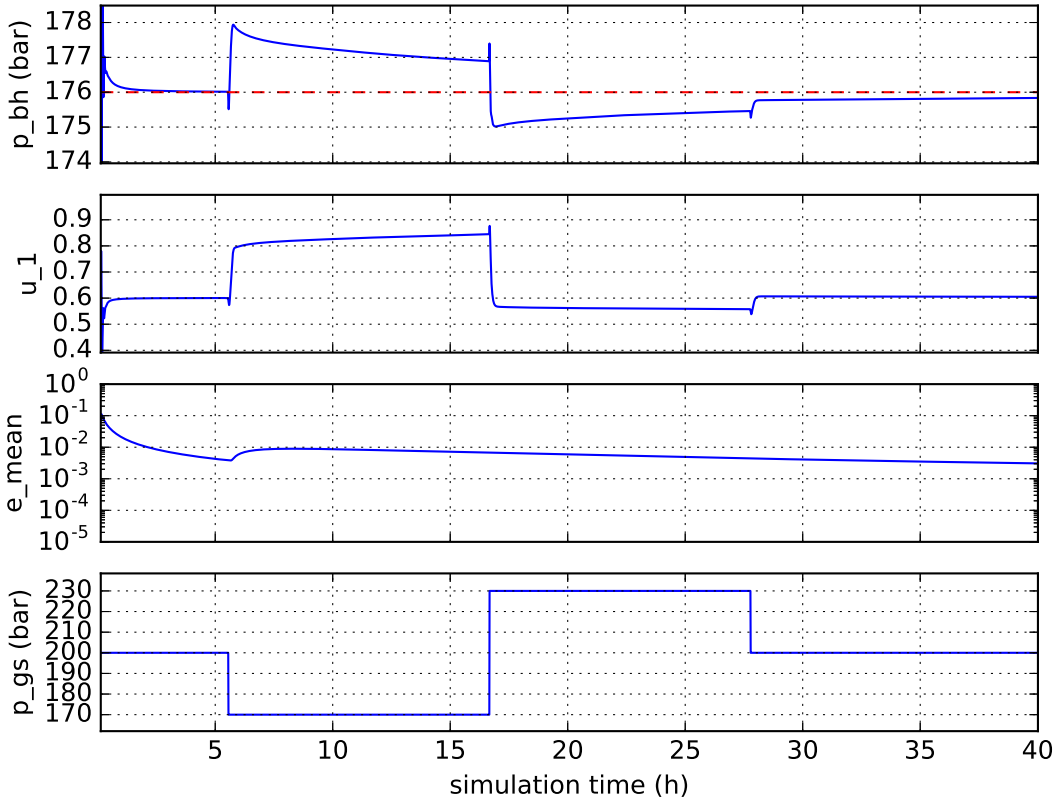

Figure 13 – Same experiment as in Figure 12, except with a larger load disturbance of -30 and of $+30$ bar in p_{gs} at times: $5,5$h and $16,6$. The disturbance ceases at $27,7$h.Figure from [2].

fully open, so the setpoint pressure which was asked could not be reached. This is shown roughly at instant $t = 11h$. After a disturbance is applied, the system is able to go back smoothly to the setpoint, showing no windup-like effects.

5.4.2 The Riser

This subsection depicts the simulation of the control in the riser. In this experiment, the riser is assumed to have a constant inflow w_{out} of 9 kg/s, which 8.64 kg/s is liquid mass flow and the rest is gas flow. This, as seen in Chapter 4, characterizes a regime prone to slugging flow. The manipulated variable was the production choke at the top of the riser. The controlled variable was the pressure at pipeline p_{in}, which is equivalent to p_p, depicted in the riser model description at Chapter 4.

The following parameters were used:

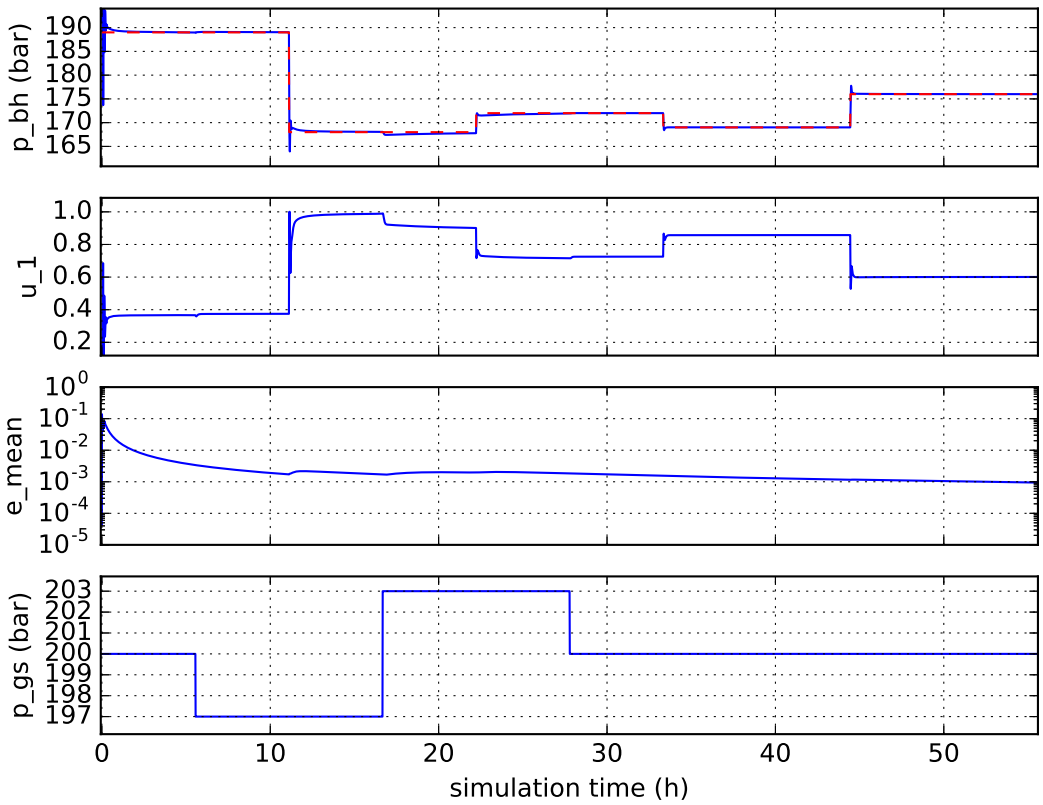

Figure 14 – Experiment showcasing both the tracking aspect and the low magnitude of the effect of the disturbance. The transient was omitted to better showcase the tracking effect.Figure from [2].

Parameter	Value
γ: Leak Rate	1.0
ρ: Spectral Radius of $\mathbf{W_r^r}$	0.999
δ: Prediction Timesteps	2
ψ: Connectivity of $\mathbf{W_r^r}$	1
N:Number of Neurons	1000
f_i^r: Scaling Factor of $\mathbf{W_i^r}$	3.0
f_b^r: Scaling Factor of $\mathbf{W_b^r}$	0
T_s: Sampling Time	2 s

All the parameters were obtained through grid searching. Since the dynamics involved in this plant are unstable, more agressive parameters, such as the high value for f_{ir} or $\gamma = 1$, showed significantly better performance than more conservative ones. Since no parameter configuration that could fully stabilize the plant was obtained, only the tracking experiment was made. The purpose of this experiment differs from the tracking experiment in the previous section. This experiment has the objective to show the limits of

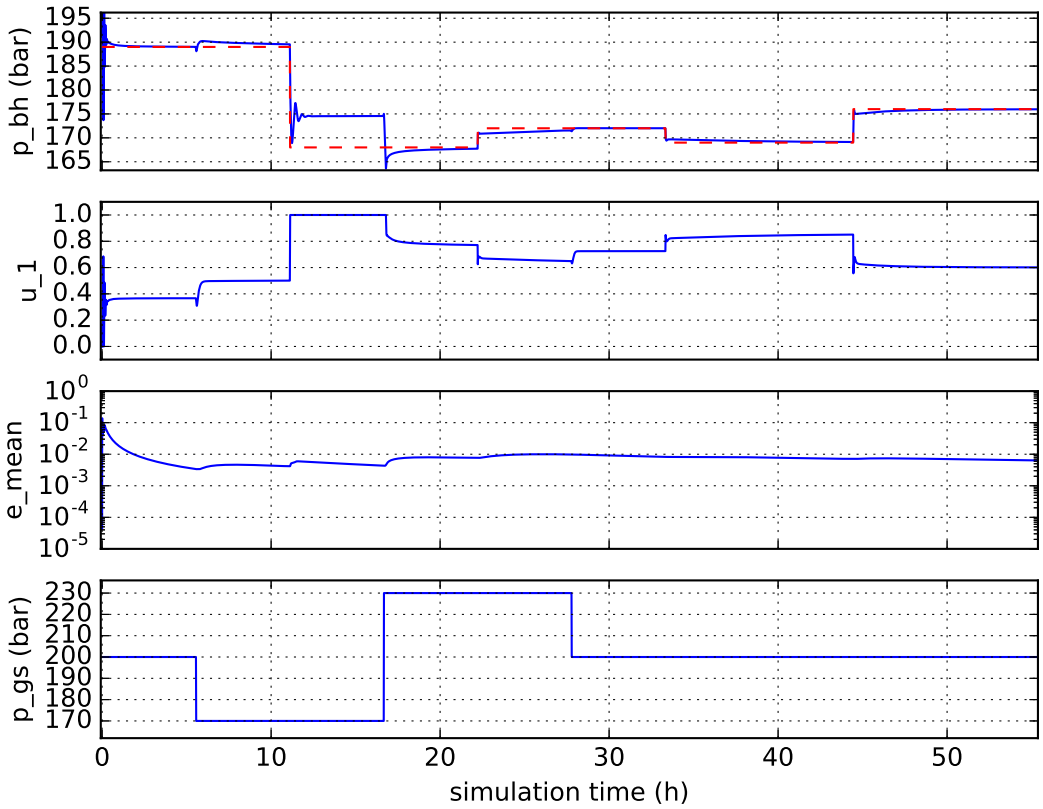

Figure 15 – Same experiment as in Figure 14, except with a larger load disturbance.Figure from [2].

this method or, in other words, test the lowest p_{in} value in which the closed loop maintains stability.

Figure 16 shows that the inverse-model online learning control can manage to stabilize the plant at a maximum of choke opening $z = 0.3$. For larger choke opening, or higher production rates, the system cannot manage to maintain stability, due to the incapability of learning the new model (shown by the increase of e_{mean} at $t = 130$ min). The open-loop theoretical maximum stability was at $z = 0.05$, which means we could open the valve 6 times more without slugging flow using this controller.

It is also shown, by the learning curve, that each change in setpoint implies in a huge change in model, until the Recursive Least Squares is not capable to converge. The system is also able to go back to stability if a higher pressure setpoint is used later.

Figure 17 and 18 show the behavior of other variables during the simulation depicted in Figure 16. It shows what is actually happening in the system, which cannot be seen by the controller.

As shown between instants $t = 100$ min and $t = 130$ min, which is the lowest stable

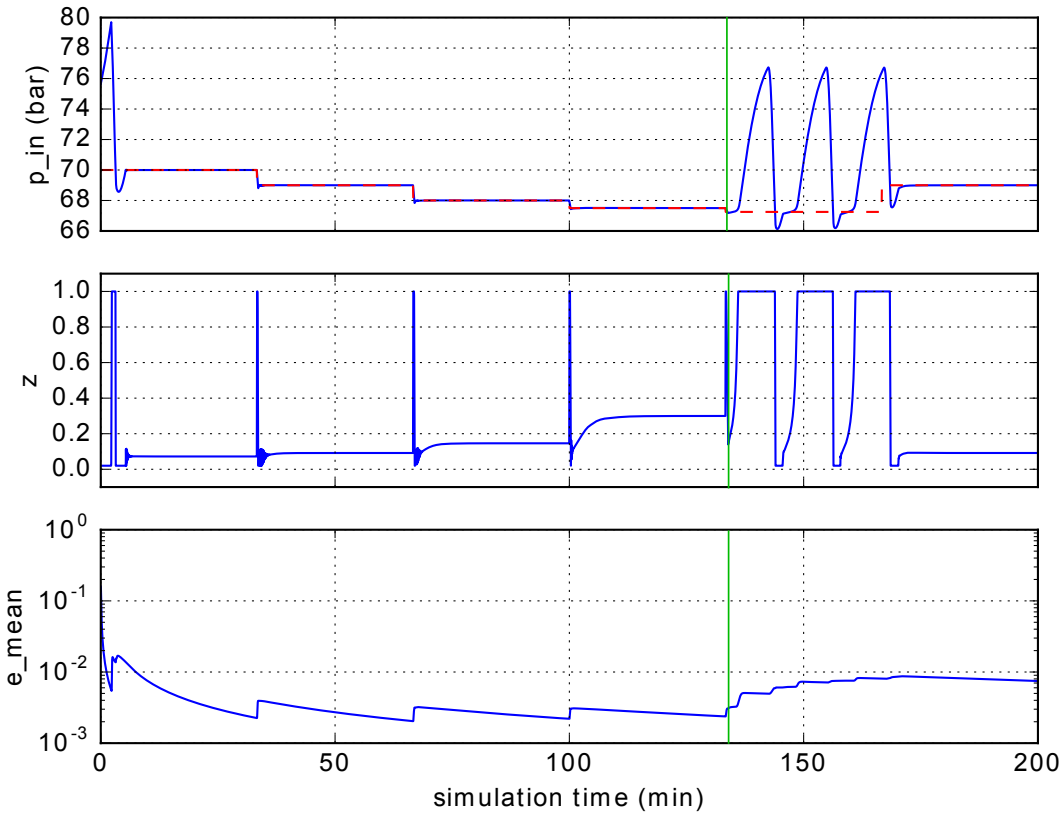

Figure 16 – Tracking experiment for the riser. This experiment was to test the limits of the online-control stabilization capability. It was shown to not be successful at certain points. The solid vertical line roughly indicates where the slugging flow happens due to the system reaching its limits.

p_{in} point, the riser top pressure p_{rt} is almost near the value of the separator pressure p_s. This is a point very close to the physical limit of stability in the riser, so lower pressure setpoints cannot to be achieved.

All of this is actually due to the fact that the inflow of the riser is constant, which is the assumption of this simulation. Generally, this is not the case. The bottom of a riser is always connected to a manifold which, in turn, is connected to a number of wells, which some may have gas injection systems, which could add gas mass flow to the riser and bring about less severe slugging flows.

5.5 Summary

In this section, we showed experimental results regarding tracking and disturbance rejection of the well, and tracking of the riser. It was shown that the inverse model online-learning controller could succesfully learn the dynamical system and effectively control the system simultaneously. The only problem was at a certain operation point in

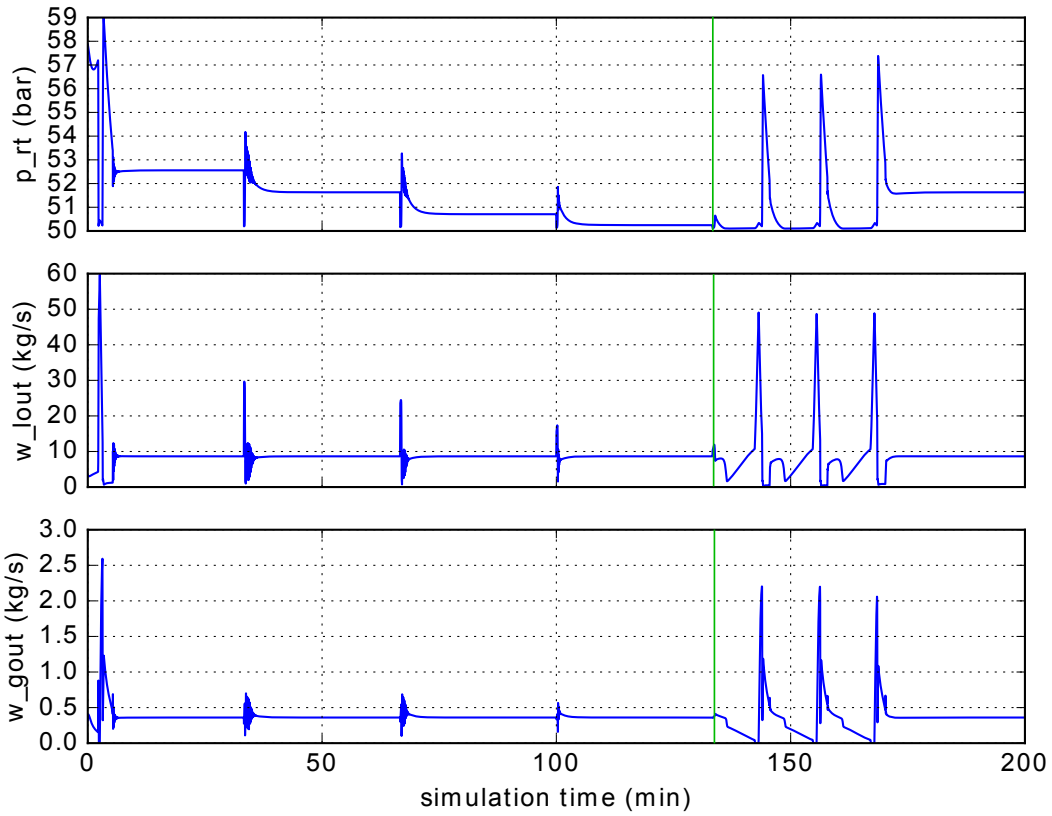

Figure 17 – Measurements of p_{rt}, $w_{l,out}$ and $w_{g,out}$ at the same simulation as Figure 16. The solid vertical line roughly indicates where the slugging flow happens due to the system reaching its limits.

the riser, in which the controller didn't manage to stabilize the pressure, though this was due to the system being driven to its physical limit, as shown by the value of p_{rt} being near the separator pressure p_s, whose value is 50.1 bar.

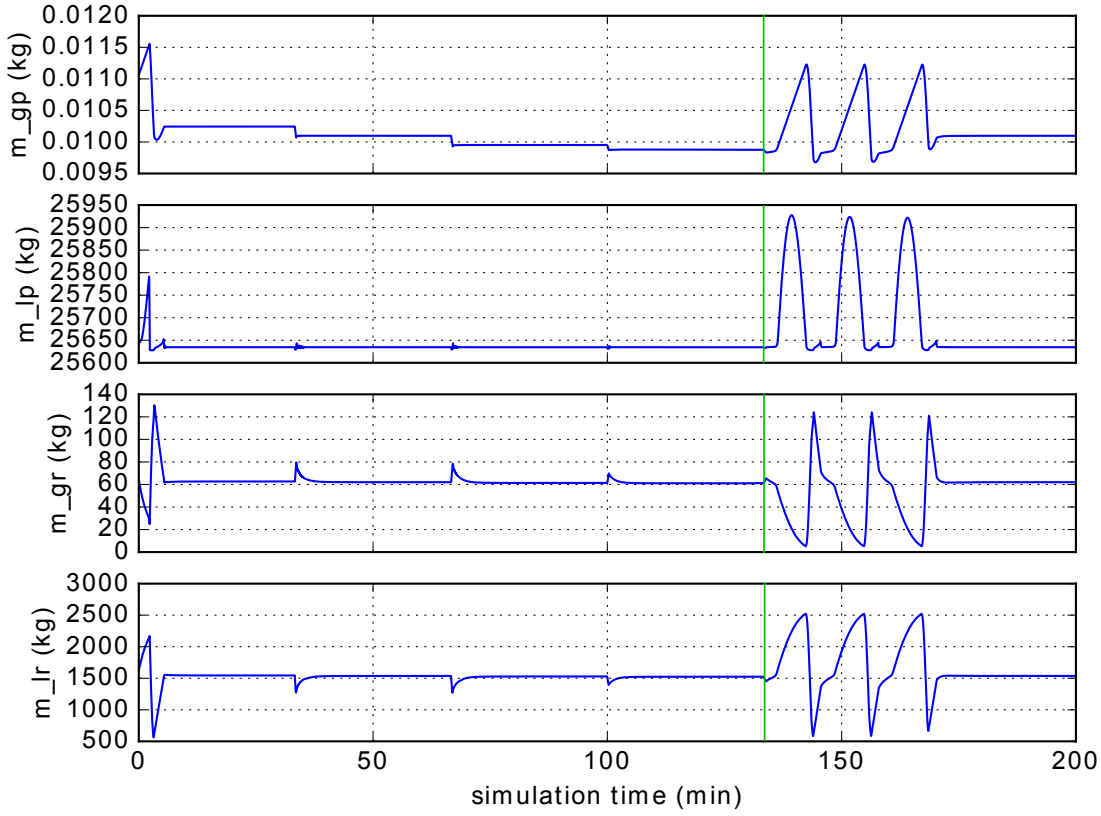

Figure 18 – Measurements of m_{gp}, m_{lp}, $m_{g,r}$ and m_{lr} at the same simulation as Figure 16. The solid vertical line roughly indicates where the slugging flow happens due to the system reaching its limits.

6 Conclusion

In this work, Echo State Networks (ESNs) were used as controllers for plants found in the oil industry. The control approach is based on [14], where the black-box ESN learns the inverse model online while it controls the plant. The main goal was to test the effectiveness of the method to control plants of that nature, testing reference tracking and disturbance rejection for oil wells and anti-slug stablization for risers.

For an oil well model [5], the plant was successfully able to perform reference tracking and disturbance rejection in all the tested operating points, which actually shows the effective potential uses of the ESN-based controller in the online learning of inverse models feedback control strategy in the context of oil and gas production systems. Usually, the control of a system whose model is unknown, is an ill-posed problem, but this control strategy was able to do setpoint tracking and disturbance rejection without prior knowledge of the well. The ability to track setpoints also means that the Echo State Network was able to capture the relatively complicated inverse model dynamics of the oil well. In terms of disturbance rejection, the assymptotic convergence was slower than in a classical control strategy, but the controller was able to have a low overshoot compared to classical control strategies such as [5].

For the riser, the system was able to execute slugging flow rejection up to a choke opening of 30%, which is a significant increase from its open-loop counterpart, with maximum choke opening of 5%. Some of the literature presented in Section 4.5 refers to a version of the pipeline-riser model featured in this work in which the maximum possible stable open-loop choke open is 15% [6, 19, 34]. A fair comparison is not possible to be drawn, though 30% of the production choke valve opening is a significant increase in performance to the initial open loop 5%.

Although the systems showed satisfactory performance over tracking and disturbance rejection, this control strategy is still not applicable in practice during the phase in which the inverse model is being learned. Also, this strategy lacks the computational simplicity that is, for example, a working PI controller. Further, the control law can be complex for a low-memory, small processing power, cheap microcontroller to compute, specially for echo state reservoirs with a large number of neurons. This is not an impediment if the system has slow dynamics which implies higher sampling periods. The weights of the echo state network are randomly chosen, so there is also the problem with respect to the weight configuration varying performance.

To validate and enhance this technique for real life uses the following could be considered in future works:

- The addition of measured noise was not considered in this implementation. Theo-

retically, the recursive least squares algorithm could ignore white noise due to the properties involving the solution to the Least Squares Problem [24], though the effect of noise in this controller has yet to be seen.

- The controlled variables,bottom-hole pressure p_{bh} and pressure at the pipeline p_{in}, are intrinsically difficult to measure, due to being at the bottom of the sea. [22] proposes a data-driven estimator for measured variables of this nature. Will this controller work with the adding of an estimator? [6] and [34] hint at a control performance decrease when the pressure at the top of the riser p_{rt} is used as a controlled variable.

- The pipeline-riser system has reached its physical limit at $z = 0.3$. This is due to assuming that the inflow of the pipeline is constant, which is not true. The inflow comes from a manifold, whose inflow, in turn, come from multiple wells with different oil and gas mass flow rates. So a future work proposal is to apply this controller in a riser coupled with multiple wells by a manifold, using the models proposed in this work.

- In [19], a supervisory control logic was incorporated in the adaptive control utilized, whose purpose was to detect the limit of the system. A supervisory control, combined with this echo state could be useful, would be able to correct faults such as the bad performance before convergence by disabling the control network, or could implement a Real Time Optimization system in combination with the controller.

Bibliography

1 ASTROM, K. J.; WITTENMARK, B. *Adaptive control*. 2. ed. [S.l.]: Dover Publications, 1994. Citado 3 vezes nas páginas 7, 19, and 20.

2 JORDANOU, J. P. et al. Recurrent neural network based control of an oil well. *Accepted for SBAI proceedings*, 2017. Citado 13 vezes nas páginas 7, 8, 22, 29, 45, 53, 56, 57, 58, 59, 60, 61, and 62.

3 JAHN, M. C. F.; GRAHAM, M. *Hydrocarbon Exploration and Production*. 2nd ed. ed. [S.l.]: Elsevier, 2008. (Developments in Petroleum Science 55). Citado 10 vezes nas páginas 7, 11, 32, 33, 34, 35, 36, 37, 38, and 40.

4 AGUIAR, M. A.; CODAS, A.; CAMPONOGARA, E. Systemwide optimal control of offshore oil production networks with time dependent constraints. *IFAC-PapersOnLine*, v. 48, n. 6, p. 200 – 207, 2015. Citado 5 vezes nas páginas 7, 32, 37, 38, and 50.

5 JAHANSHAHI, E.; SKOGESTAD, S.; HANSEN, H. Control structure design for stabilizing unstable gas-lift oil wells. *IFAC Proceedings Volumes*, Elsevier, v. 45, n. 15, p. 93–100, 2012. Citado 8 vezes nas páginas 7, 11, 12, 39, 42, 51, 52, and 66.

6 JAHANSHAHI, E.; SKOGESTAD, S.; GRØTLI, E. I. Anti-slug control experiments using nonlinear observers. In: *2013 American Control Conference*. [S.l.: s.n.], 2013. p. 1056–1062. Citado 6 vezes nas páginas 7, 39, 51, 52, 66, and 67.

7 JAHANSHAHI, E. *Control Solutions for Multiphase Flow, Linear and Non-linear Approaches to Anti-slug Control*. Tese (Doutorado), 2013. Citado 8 vezes nas páginas 7, 11, 38, 39, 40, 41, 42, and 50.

8 JAEGER, H. The ‘‘echo state’’ approach to analysing and training recurrent neural networks - with an erratum note. 2001. Citado 3 vezes nas páginas 11, 22, and 23.

9 LIN, X.; YANG, Z.; SONG, Y. Short-term stock price prediction based on echo state networks. *Expert Systems with Applications*, Pergamon Press, Inc., v. 36, n. 3, p. 7313–7317, 2009. Citado na página 11.

10 ANTONELO, E. A.; FLESCH, C. Reservoir computing for detection of steady state in performance tests of compressors. 2016. Citado na página 11.

11 ANTONELO, E. A.; SCHRAUWEN, B. On learning navigation behaviors for small mobile robots with reservoir computing architectures. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 4, p. 763–780, 2015. Citado na página 11.

12 LUKOŠEVIČIUS, M.; MAROZAS, V. Noninvasive fetal QRS detection using an echo state network and dynamic programming. *Physiological Measurement*, v. 35, n. 8, p. 1685–1697, 07 2014. Citado na página 11.

13 HINAUT, X.; DOMINEY, P. F. On-line processing of grammatical structure using reservoir computing. In: VILLA, A. E. P. et al. (Ed.). *ICANN: International Conference on Artificial Neural Networks*. [S.l.: s.n.], 2012. p. 596–603. Citado na página 11.

14 WAEGEMAN, T.; WYFFELS, F.; SCHRAUWEN, B. Feedback control by online learning an inverse model. *IEEE Transactions on Neural Networks and Learning Systems*, v. 23, n. 10, p. 1637 – 1648, 10 2012. Citado 9 vezes nas páginas 11, 12, 23, 29, 30, 50, 56, 57, and 66.

15 PARK, J. et al. Utilizing online learning based on echo-state networks for the control of a hydraulic excavator. *Mechatronics*, v. 24, p. 986–1000, 2014. Citado na página 11.

16 GALTIER; MATHIEU. Ideomotor feedback control in a recurrent neural network. *Biological Cybernetics*, v. 109, n. 3, p. 363–375, 2015. Citado na página 11.

17 PAN, Y.; WANG, J. Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics*, v. 59, n. 8, p. 3089–3101, 2012. Citado na página 11.

18 JAHANSHAHI, E.; SKOGESTAD, S. Simplified dynamical models for control of severe slugging in multiphase risers. *IFAC Proceedings Volumes*, Elsevier, v. 44, n. 1, p. 1634–1639, 2011. Citado 6 vezes nas páginas 11, 12, 46, 50, 51, and 52.

19 OLIVEIRA, V. de; JäSCHKE, J.; SKOGESTAD, S. An autonomous approach for driving systems towards their limit: an intelligent adaptive anti-slug control system for production maximization. *IFAC-PapersOnLine*, v. 48, n. 6, p. 104 – 111, 2015. Citado 5 vezes nas páginas 11, 51, 56, 66, and 67.

20 CAMPOS, M. et al. Advanced anti-slug control for offshore production plants. *IFAC-PapersOnLine*, v. 48, n. 6, p. 83 – 88, 2015. Citado 2 vezes nas páginas 11 and 50.

21 PLUCENIO, A.; CAMPOS, M. M.; TEIXEIRA, A. F. New developments in the control of fluid dynamics of wells and risers in oil production systems. *IFAC-PapersOnLine*, v. 48, n. 6, p. 97 – 103, 2015. Citado 2 vezes nas páginas 11 and 51.

22 ANTONELO, E. A.; CAMPONOGARA, E.; FOSS, B. Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells. *Neural Networks*, v. 85, p. 106–117, 2017. Citado 4 vezes nas páginas 12, 24, 25, and 67.

23 CHEN, C.-T. *Linear System Theory and Design*. 3rd. ed. New York, NY, USA: Oxford University Press, Inc., 1998. ISBN 0195117778. Citado 3 vezes nas páginas 14, 15, and 16.

24 NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. 1. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2001. Citado 9 vezes nas páginas 17, 18, 21, 23, 25, 26, 27, 28, and 67.

25 BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. [S.l.]: Springer-Verlag New York, Inc., 2006. Citado na página 19.

26 HEIRUNG, T. A. N. *Dual Control: Optimal, Adaptive Decision Making Under Uncertainity*. Tese (Doutorado), 2016. Citado 2 vezes nas páginas 20 and 21.

27 JAEGER, H. et al. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, v. 20, n. 3, p. 335–352, 2007. Citado 3 vezes nas páginas 22, 23, and 56.

28 MOZER, M. C. Backpropagation. In: CHAUVIN, Y.; RUMELHART, D. E. (Ed.). Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1995. cap. A Focused Backpropagation Algorithm for Temporal Pattern Recognition, p. 137–169. Citado na página 23.

29 COELHO, A. A. R.; COELHO, L. d. S. *Sistemas Dinâmicos Lineares*. [S.l.]: editora UFSC, 2004. Citado na página 26.

30 PALEOLOGU, C.; BENESTY, J.; CIOCHINã, S. A robust variable forgetting factor recursive least-squares algorithm for system identification. *IEEE Signal Processing Letters*, v. 15, n. 10, p. 597 – 600, 2008. Citado 2 vezes nas páginas 28 and 29.

31 DEVOLD, H. *Oil and Gas Production Handbook: An Introduction to Oil and Gas*. [S.l.]: SRH MEdia, 2013. Citado 2 vezes nas páginas 31 and 33.

32 WARREN, J. C. N. G. S. G. *Organic chemistry*. 2ed. ed. [S.l.]: Oxford University Press, 2012. Citado na página 32.

33 PETIT, N. Systems with uncertain and variable delays in the oil industry: some examples and first solutions. *IFAC-PapersOnLine*, v. 48, n. 6, p. 68 – 76, 2015. Citado na página 50.

34 JAHANSHAHI, E.; SKOGESTAD, S.; GRØTLI, E. I. Nonlinear model-based control of two-phase flow in risers by feedback linearization. *IFAC Proceedings Volumes*, Elsevier, v. 46, n. 23, p. 301–306, 2013. Citado 3 vezes nas páginas 52, 66, and 67.