Digital Chemical Engineering 5 (2022) 100056

Contents lists available at ScienceDirect

Digital Chemical Engineering

journal homepage: www.elsevier.com/locate/dche

Best practices and methods

A Physics-Informed Neural Networks (PINN) oriented approach to flow metering in oil wells: an ESP lifted oil well system as a case study

Taniel S. Franklin[a,*], Leonardo S. Souza[a], Raony M. Fontes[b,a], Márcio A.F. Martins[b,a]

[a] Mechatronics Program, Polytechnic School, Federal University of Bahia, Salvador-BA, Brazil
[b] Department of Chemical Engineering Polytechnic School, Federal University of Bahia, Salvador-BA, Brazil

ARTICLE INFO

Keywords:
soft sensor
Physics-Informed Neural Networks
electrical submersible pump
virtual flow meter
recurrent neural network

ABSTRACT

A virtual flow meter (VFM) is an interesting and growing method to replace or support expensive and high-maintenance dependent physical flow rate sensors in oil and gas production systems. This paper presents a new use of Physics-Informed Neural Networks (PINN) as a hybrid virtual sensor to be used in an oil well system. This strategy combines prior knowledge about the system dynamics in the form of a phenomenological model in order to drive a long short-term memory (LSTM)-type recurrent neural network (RNN) training. The resulting model is able to predict the average flow rate in the production column some time steps ahead using measurements of states and exogenous inputs available in an oil well information system. The proposed scheme is also flexible as it can estimate fluid properties via their model parameters as an additional feature. The results obtained in this study through simulation used a benchmark electrical submersible pump (ESP) model to demonstrate the proposed PINN-based VFM potential as an accurate alternative to infer flow rate in oil wells.

1. Introduction

A petroleum production system optimizer takes concurrent and multi-objective decisions to make the field profitable and successful. It has several decision variables that help to control and optimize production. Figure 1 depicts an offshore exploration system with multiple wells connected by a production pipeline transporting the fluid to the platform. Note that each choke valve controls the individual average production rate of each well, contributing to the overall production of the field.

The control system requires accurate measurements to guarantee production performance as efficiently as possible. The well flow measurement is an economic variable that affects the production flow measurement of the field, which means it deserves special attention since fluid properties (high viscosity, multi-phase) impose limitations to getting accurate measurements through physical devices. In offshore oil production systems, the best measurement scenario requires the installation of accurate physical flow meters into each well that have costly technology and are prone to failure, erosion, and blockage (Grimstad et al., 2021; Toskey, 2012). These flow measurements are usually obtained through periodic well-tests (e.g., weekly or monthly), assuming they are constants between individual well-tests intervals. In addition, the well production test requires a logistics and management specialist (Bikmukhametov and Jschke, 2020; Chaudhry, 2004; Monteiro et al., 2020).

Due to economic and logistical constraints, alternative strategies have been explored to support or replace physical flow meters. Alternatively, the well flow rates can be measured indirectly by virtual flow meters (VFM). A VFM is a soft-sensing technology that estimates flow rates using mathematical methods and ancillary measurements (Grimstad et al., 2021; Toskey, 2012). VFMs utilize models based on system phenomenology (in accordance with fundamental theory), data-driven, or a combination of both (Hotvedt et al., 2020). Usually, these flow rate inferences are easily supported by pressure and temperature sensors that do not suffer from the same limitations as physical flow meters.

Several works discussing VFM in oil wells are available in the literature. AL-Qutami et al. (2018) propose a VFM system based on ensemble learning methods with several neural networks (NN) learners. An adaptive simulated annealing optimization is used to select the best subset of learners and the optimal strategy to combine them. The method described in Góes et al. (2021) predicts liquid and gas flow rates of each oil well as a function of real data of the offshore platform using virtual meters based on phenomenological models. This method is simulated in a computational environment called EMSO® and validated by measurements obtained from fiscal well-test reports. Andrade et al. (2022) de-

[*] Corresponding author.
E-mail addresses: tanielfranklin@gmail.com (T.S. Franklin), leosilvasouza2001@gmail.com (L.S. Souza), raony@ufba.br (R.M. Fontes), marciomartins@ufba.br (M.A.F. Martins).

https://doi.org/10.1016/j.dche.2022.100056
Received 8 June 2022; Received in revised form 27 August 2022; Accepted 5 September 2022

Table 1
Comparative analysis of VFM methods.

	Virtual Flow Metering	
	pros	*cons*
General	Real-time monitoring Low-cost implementation and maintenance Physical intervention is not required Easy integration into the optimization system and the supervisory layer	Depends on the sensor accuracy Requires periodical tuning and an expert team Can fail if most of the sensors fail
Phenomenological Model	Can be used to infer unmeasured variables Long operational experience Several competitors Can be used for parameter estimation	Needs deep knowledge about phenomena A specific and well-defined operational range A complex parameter tuning process High computational cost compared to data-driven VFM
Data-driven model	Dismisses deep knowledge about phenomena Operation with low computational cost Learns model parameters easily	It is not recommended for short-term historical data High computational cost for training Weak learning about system dynamics Most of the methods operate at a steady-state regime

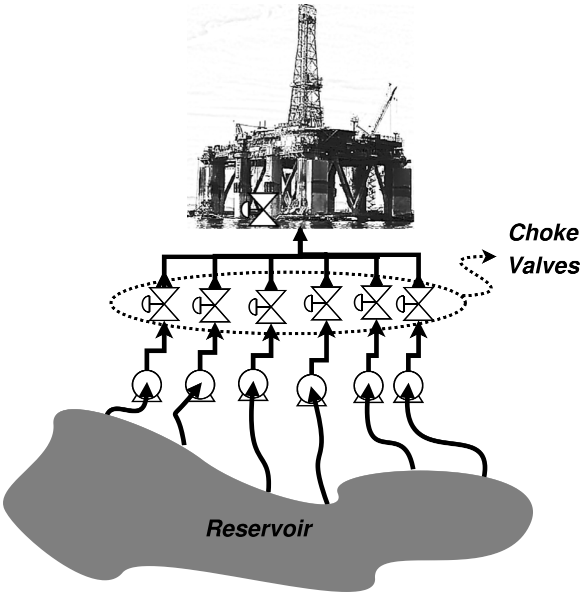


Fig. 1. Simplified representation of an offshore oil production system.

veloped a tool (called data reconciliation-based VFM) to infer the flow rate data of individual wells operated simultaneously in an oil and gas offshore exploration system. This proposal uses data reconciliation to calibrate phenomenological models using oil wells data like pressure and temperature well profiles and the overall gas, oil, and water flow rates. Despite the many advantages in favor of VFMs, many data-driven machine learning (ML) techniques fail to guarantee convergence and robustness. Beyond that, many problems face prohibitive measurement challenges caused by high costs, technology, and precision limitations associated with physical sensors (Bikmukhametov and Jschke, 2020).

Bikmukhametov and Jschke (2020) prepared a literature overview concerning VFMs and Table 1 presents some pros and cons involving the two approaches. Scenarios with limited operational experience demand phenomenological approaches. On the other hand, data-driven methods do not require a deep knowledge of physical phenomena, which does not guarantee good performance for all operating conditions because of their limitation to generalize to scenarios not encountered in training data. Hybrid modeling (or gray-box modeling) is considered a kind of science-based knowledge integration into machine learning, as discussed in Willard et al. (2020), who named it physics-based ML. This integration usually aims to obtain a surrogate model that reproduces the behavior of system dynamics with a low computational cost. Physics-based ML can be categorized by (Willard et al., 2020):

Physics-guided loss function aims to incorporate physical knowledge into the loss function that drives the ML training to capture model dynamics following physical laws. A typical strategy is to include constraints based on physical laws in the loss function. Raissi et al. (2017a,b) show how to find approximators (physics-informed neural networks - PINN) able to solve and discover partial differential equations (PDE) and ordinary differential equations (ODE).

Physics-guided initialization is based on exploring different ways to start a model state (weights and bias) with the help of physical knowledge to avoid the training loss being fixed at a local minimum (random initialization) and accelerate convergence;

Physics-guided design of architecture aims to build a neural network architecture that preserves characteristics of the problem that is being solved, like a node connection used to capture dependencies among variables, for example. Or designate physical meaning to neurons of the NN.

The residual modeling approach makes the ML model learns to predict the errors (or residuals) made by physics-based models. They need to operate simultaneously with the same input signals.

Hybrid physics-ML models can be considered a generalization of residual modeling. One possibility to develop a hybrid physics-ML model is to consider the output of the physics-based model an additional input of the physics-ML model. Others strategies replace one or more components of a physics-based model that were not well modeled using physics by ML models.

The data-driven approach finds the solution using mathematical learning techniques that use datasets that characterize the system dynamics (Blechschmidt and Ernst, 2021). However, it is not easy to incorporate information from system dynamics into learning-based methods. The performance of most data-driven approaches is acceptable when modeling steady-state systems but can fail to model transient fluid flow accurately (Bikmukhametov and Jschke, 2020). Data-driven methods typically discard the knowledge about phenomenology modeling, whereas phenomenological VFM utilizes this prior knowledge through differential equations representing the system dynamics. VFMs based on state estimation, such as extended Kalman filter (EKF) and moving horizon estimator (MHE), are viable solutions, but with insufficient industrial applications due to their complex approaches (Bikmukhametov and Jschke, 2020).

In this scenario, the oil and gas industry requires a VFM that uses a smaller dataset than a purely data-driven VFM approach while retaining physical information about the system without numerical and convergence issues. A VFM integrated into each well information system can provide individual flow estimates used to compute the global field production, allowing faster problem identification before the next well test.

The PINN proposed in Raissi et al. (2017a,b) is powerful to solve PDEs using phenomenological models combined with a data-oriented

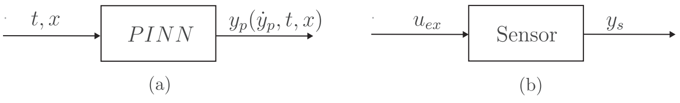


Fig. 2. Conventional PINN use (a) with one state y_p dependent from x(space) and t(time) and, VFM-like PINN use (b) that senses exogenous inputs samples u_{ex} and provides the measured value y_s.

approach. It focuses on autonomous systems with a fixed time domain and only utilizes independent variables as input. PINN is indeed used to find solutions and unknown parameters of ordinary differential equations (ODEs) and partial differential equations (PDEs). Cheng and Zhang (2021) propose a PINN architecture that improves PINN performance based on Residual Neural Networks (Resnet) in the context of fluid flow problems.

Despite the wide success of PINN in various fields, the conventional architecture cannot be used as a VFM because VFM-type soft-sensor estimates state variables along the time continuously using arbitrary exogenous inputs. Arnold and King (2021) propose a state space model based on the PINN approach that enables control synthesis and state estimation for dynamic systems. Figure 2 illustrates the main differences between the use of PINN as a virtual sensor (proposed technique) and the conventional PINN after training. A conventional PINN seeks to solve ODE or PDE for different values of the independent variables (time and space) but it does not allow the inclusion of arbitrary exogenous inputs and continuous prediction of states. Before training, PINNs require a fixed time-space domain and specific initial (and/or boundary) conditions. It is possible to include an exogenous input if it is defined in the fixed space- time domain before training as well. However, it must be re-trained to make a prediction when new exogenous and initial (and/or boundary) conditions are considered.

Hotvedt et al. (2022) compare different hybrid VFMs similar to the physics-ML model presented by Willard et al. (2020). They estimate multi-phase flow supported by data-driven approaches to replace non-trustful simplifications adopted in the mechanistic model development. Hybrid VFMs discussed in Hotvedt et al. (2022) need to solve mathematical models when operating, becoming computationally expensive due to the use of multivariate nonlinear solvers to find unique solutions.

In this context, it is helpful to present a VFM that takes advantage of the interrelated benefits of the two approaches described in Table 1. This paper proposes an extended PINN as an alternative method to develop a data-driven VFM that integrates phenomenological information from an artificial lift method system, namely PINN-oriented VFM. It is computationally feasible for use in real-time applications and has scientifically consistent behavior. Its potential is demonstrated using a traditional ESP-lifted oil well model, predicting responses to arbitrary exogenous inputs, operating at a steady-state and transient regime, inferring the average flow rate in the production column, and estimating unknown model parameters.

This paper is structured as follows. Section 2 introduces PINN fundamentals briefly with the PINN-oriented VFM proposal. Section 3 presents the PINN-oriented VFM proposal applied to the ESP-lifted oil well model. Section 4 demonstrates and discusses the PINN-oriented VFM potential through illustrative case study simulation. The conclusion is presented in Section 5.

2. A proposed PINN-oriented VFM

Deep NNs approximate complex mathematical functions through a large and representative dataset (Blechschmidt and Ernst, 2021). Problems with a smaller dataset can utilize deep NN combined with physical knowledge to complement the lack of data. The PINN is a scientific machine learning technique aimed to solve supervised learning tasks that do not violate nonlinear partial differential equations related to physical laws. (Raissi et al., 2019).

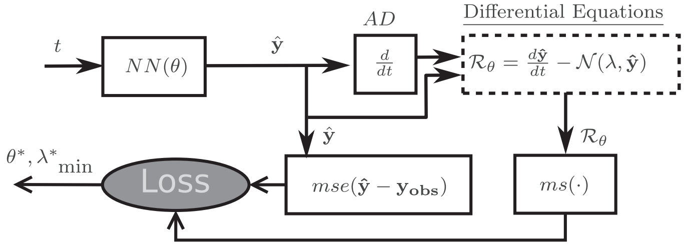

t
$NN(\theta)$
$\hat{\mathbf{y}}$
AD
$\frac{d}{dt}$
Differential Equations
$\mathcal{R}_\theta = \frac{d\hat{\mathbf{y}}}{dt} - \mathcal{N}(\lambda, \hat{\mathbf{y}})$
$\hat{\mathbf{y}}$
$\mathcal{R}_\theta$
$\theta^*, \lambda^*{}_{\min}$
Loss
$mse(\hat{\mathbf{y}} - \mathbf{y_{obs}})$
$ms(\cdot)$

Fig. 3. Schematic of a conventional PINN to solve ODE.

The general form of a typical data-driven problem to solve ordinary differential equations is represented by :

$$\frac{\partial y}{\partial t} + \mathcal{N}[y] = 0, \quad t \in [0, T], \tag{1}$$

where $y(t)$ represents the exact solution of Eq. (1) in the time domain and $\mathcal{N}[\cdot]$ is a nonlinear differential operator. Machine learning methods usually find a purely data-driven solution, in which the latter is the approximated solution to the true underlying problem, minimizing an objective function defined by the following equation:

$$\mathcal{L}(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} [\hat{y}^i_\theta - y^i_{obs}]^2, \tag{2}$$

where $\hat{y}_\theta$ is the predicted surrogate solution provided by NN (i.e. $\hat{y}_\theta \approx y$), y_{obs} is the observed data with N_s samples, and θ represents NN parameters. The PINN concept is based on the insertion of additional terms in Eq. (2) that contemplate the physical phenomena through the underlying ODE model, which can be in continuous or discrete form. This additional term in Eq. (2) is obtained by approximating y by a deep NN $\hat{y}_\theta$ (Raissi et al., 2019). The NN output is used to compute the ODE residual defined by:

$$\mathcal{R}_\theta := \frac{d\hat{y}_\theta}{dt} + \mathcal{N}[\hat{y}_\theta], \tag{3}$$

where $\mathcal{R}_\theta$ denotes the residues provided by deep neural network approximation y_θ. So the right-hand side of Eq. (3) is inserted in Eq. (2) to constrain the training by physical phenomena following the PINN concept previously described. Therefore, a PINN-based loss equation is written as follows:

$$\mathcal{L}(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} [\hat{y}^i_\theta - y^i_{obs}]^2 + \frac{1}{N_\mathcal{R}} \sum_{i=1}^{N_\mathcal{R}} (\mathcal{R}^i_\theta)^2, \tag{4}$$

where $N_\mathcal{R}$ defines the number of collocation points inside the time domain. Automatic differentiation (AD) or a numerical integration technique is used to compute $\mathcal{R}^i_\theta$, computing derivatives of the neural network with respect to their inputs coordinates and model parameters (Raissi et al., 2019). The samples $\{t^i, y_{obs}(t^i)\}_{i=1}^{N_s}$ represent the initial conditions, and observed values, while $\{t^i_\mathcal{R}, y(t^i_\mathcal{R})\}_{i=1}^{N_\mathcal{R}}$ are the collocation points related to $\mathcal{R}^i_\theta$. The solution needs to respect all initial conditions and observed values simultaneously, so they are considered particular types of boundary conditions. In this case, initial conditions are sufficient to minimize Eq. (4), enabling PINN to predict states without observed values. Figure 3 shows a conventional PINN schematic considering a model with multiple states ($\mathbf{y}$). Note that $mse(\cdot)$ represents the mean squared error and $ms(\cdot)$ the mean squared value.

As Raissi et al. (2019), Blechschmidt and Ernst (2021) use Runge-Kutta time-stepping semi-discretization to obtain a discrete form of Eq. (3). Since the initial solution at time $t^n \in [0, T)$, the solution at time t^{n+1} is computed by:

$$\begin{aligned} y^{n+c_i} &= \quad y^n - \Delta t \textstyle\sum_{j-1}^{q} a_{ij} \mathcal{N}[y^{n+c_j}] \quad i = 1, ..., q \\ y^{n+1} &= \quad y^n - \Delta t \textstyle\sum_{j-1}^{q} b_j \mathcal{N}[y^{n+c_j}] \end{aligned} \tag{5}$$

where $y^{n+c_j} \approx y(t^n + c_j \Delta t)$ and $a_{ij},\ b_j,\ c_j$ are coefficients that depend on the discretization methods and q is the number of stages of the Runge-

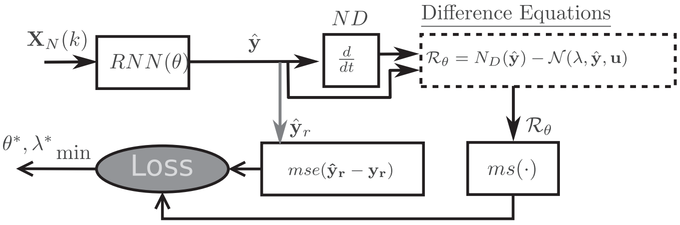


Fig. 4. Schematic of the PINN extension. Note that all predicted states are presented in $\hat{\mathbf{Y}}$, its reduced form $\hat{\mathbf{Y}}_\mathbf{r}$ is used with measured observed states $\mathbf{Y}_\mathbf{r}$ to compute the prediction error loss.

Kutta method. In this case, the $\mathcal{R}_\theta$ term is:

$$y^n - \Delta t \sum_{j-1}^{q} b_j \mathcal{N}[y^{n+c_j}] - y^{n+1} := \mathcal{R}_\theta. \tag{6}$$

The schematic proposed in Figure 4 extends PINN to be used as a VFM. It explores the concepts of the PINN presented before taking advantage of a phenomenological model to drive a NN training and extend it to accept exogenous inputs instead of independent variables. Note that $\hat{y}$ in Figure 3 allows calculating residues using AD because it is time-dependent, but it is not possible with the $\hat{y}$ in Figure 4, which does not depend on t directly. Because of that, PINN-based VFM adopts numerical differentiation to compute the residue terms. Moreover, conventional PINN calculates ODE/PDE solutions in a bounded time domain defined before the training stage i.e. the time variable is considered as exogenous input in the conventional PINN use.

Many PINN applications adopt feedforward NN with success, but the PINN-based VFM uses an LSTM NN (special RNN) that has long-short-term memory (essential for dynamic systems modeling) and resilience against the gradient vanishing problem. Different RNNs can also be used, as the recent GRU (Gate Recurrent Unit) network derived from the LSTM structure. Further investigations can be addressed to different NN topologies. LSTM can predict future states from historical sequences. The network inputs are a sequence of past measured states and past exogenous variables. This model can predict nonmeasured states using only information from other past measurements. Since uncertainties are present in real systems the training can also be able to track unknown model parameters, which means solving the inverse PINN problem.

The training is based on observed data, and each residue is computed using a sequence whose length must be sufficient to compute the numerical differentiation. Each output is a consequence of historical inputs with N_h time-steps. This RNN architecture is many-to-one (also called sequence-to-one). So, each network input/output training sample is defined by $\mathbf{X}_N(k)$ and $\hat{\mathbf{y}}(k)$ so that:

$$\hat{\mathbf{y}}(k) = \mathcal{N}\left(\mathbf{X}_{N_h}(k)\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{Y}_{rN_h} \\ \mathbf{U}_{N_h}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y_r}(k-1) \\ \mathbf{y_r}(k-2) \\ \vdots \\ \mathbf{y_r}(k-N_h) \\ \mathbf{u}(k-1) \\ \mathbf{u}(k-2) \\ \vdots \\ \mathbf{u}(k-N_h)\end{bmatrix}\right), \tag{7}$$

where $\mathbf{u}$ and $\mathbf{y}$ denote vectors of exogenous measured inputs and states, respectively. The uppercase notation in $\mathbf{U}$ and $\mathbf{Y}$ represents the time sequence of $\mathbf{u}$ and $\mathbf{y}$. $\mathbf{Y_r}$ and $\mathbf{y_r}$ represent the measured states vectors.

Since time is not a network input, the differential term of Eq. (3) is obtained by numerical differentiation (N_D) of the network sequences output using progressive with three points, central with two points, and regressive with three points first order approximation. Therefore, the residues are obtained as follows:

$$\mathcal{R}_\theta = N_D(\hat{\mathbf{y}}) - \mathcal{N}(\lambda, \hat{\mathbf{y}}, \mathbf{u}) = [r_1, r_2, \cdots, r_j]^T, \quad \text{with} \quad j = 1, 2, \cdots, N_R, \tag{8}$$

where N_R represents the number of differential equations, λ denotes a generic model parameter. The residues defined by r_j fulfill the role of prior knowledge required to constrain a typical NN loss function given by:

$$\mathcal{L}(\theta, \lambda) = \mathcal{L}_{bc} + \mathcal{L}_{y_1} + \mathcal{L}_{y_2} + \cdots + \mathcal{L}_{y_j}, \quad \text{with} \quad j = 1, 2, \cdots, N_R, \tag{9}$$

where residues and boundary conditions terms are defined by:

$$\mathcal{L}_{y_j} = \frac{1}{N_s}\sum_{i=0}^{N_s}[r_j(k+i)]^2, \quad \text{with} \quad j = 1, 2, \cdots, N_R \quad \text{and}$$
$$\mathcal{L}_{bc} = \frac{1}{N_s}\sum_{i=0}^{N_s}[\hat{\mathbf{y}}_\mathbf{r} - \mathbf{y}_\mathbf{r}]^2, \tag{10}$$

completing the terms of Eq. (9). Notice that Eqs. (8) and (9) are similar to Eqs. (3) and (4). Each unknown model parameter to be identified should be included in Eq. (8) as a trainable variable.

The ODE should be scaled to avoid numerical problems during the training because the complex nonlinear equations that compose the model can result in unbalanced residues. Furthermore, the PINN training is a multi-objective optimization problem that requires a balanced form of all loss function terms. A complete procedure to obtain a dimensionless system with scaling techniques applied to differential equations is available in Langtangen and Pedersen (2016).

Remark 1. The main differences between this approach and conventional PINN are: i) the network is fed by exogenous inputs and states instead of t or x (i.e. time and space), ii) since AD cannot compute the exact derivatives related to inputs, the residues computing uses numerical differentiation applied in a network output sequence, iii) after model training, it can predict the output one time-step ahead using the history of exogenous and states as inputs as typical VFM, and iv) a nonmeasured state can also be estimated.

Remark 2. It is noteworthy that the PINN-based VFM approach encapsulates some limitations, like any machine learning scheme, namely: careful tuning of the weights on the loss function and phenomenological knowledge integration.

3. Application: ESP-lifted oil well systems

The PINN-oriented VFM applied to ESP-lifted oil well systems depends on its dynamic model as well as data system information composed of measured variables and known parameters combined with empirical test data. The method presented in this paper can also be extended to different artificial lift methods. Since the method aims to calculate the residues of an ESP-lifted oil well model using experimental data, its helpful to briefly cover the mathematical model fundamentals.

3.1. ESP-lifted oil well system model

A typical ESP-lifted oil well system is represented in Figure 5. The main model variables are bottomhole pressure and wellhead pressure denoted as P_{bh} and P_{wh}, respectively. The average flow rate in the well is denoted by q, the flow rate across the choke valve is q_{ch} and q_r is the inflow from the reservoir. P_m and P_r are manifold and reservoir pressures. The following assumptions are considered:

- there are pressure sensors installed at the production manifold so that measurements of P_{wh} and P_m are available;
- the ESP is equipped with a pressure sensor providing measure of differential pressure across it;
- the choke opening Z_c is measured;
- manifold and reservoir pressures are considered measured disturbances since they affect the state/controlled variables, and a control system of each well, in practice, is independent;
- the fluid has a low gas-oil ratio and its properties do not change significantly along the well.

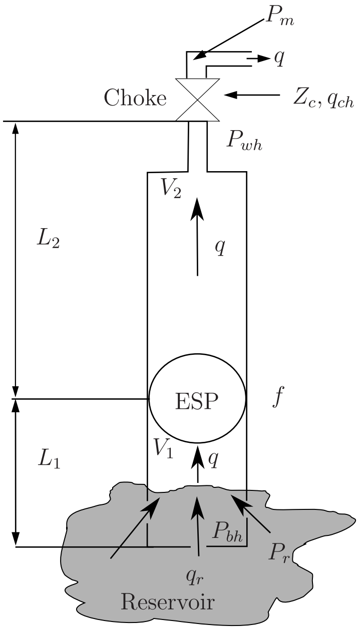


Fig. 5. Scheme of ESP-lifted oil well system model adapted from Binder et al. (2015).

Reservoir pressure is difficult to be measured, but it is assumed to be known from dedicated well tests. If P_r is unknown, its value can be estimated in the same way that PI and ρ when training the model.

The control system must keep controlled variables inside the operational envelope-like area bounded by pump downthrust and upthrust limits that guarantee safe and stable operation. At the same time, the controller has to prioritize maximum flow rate production, economic profit, or low energy consumption. The flow rate is controlled by the frequency of the pump and the choke valve opening.

The well system can be modeled by a set of nonlinear ODEs. A two-volume well model presented in Rnning (2011) and Binder et al. (2015) that governs the ESP-lifted oil well system is represented by a third-order nonlinear differential equation described as follows:

$$\frac{dy}{dt} = \begin{bmatrix} \dot{P}_{bh} \\ \dot{P}_{wh} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{b_1}{V_1}(q_r - q) \\ \frac{b_2}{V_2}(q - q_{ch}) \\ \frac{1}{M}\left(P_{bh} - P_{wh} - \rho g h_w - F_1(q) - F_2(q) + \rho g H(f, q)\right) \end{bmatrix}, \tag{11}$$

where

$$q_r = PI(P_r - P_{bh}) \quad \text{and} \quad q_{ch} = Cc(Z_c/100)\sqrt{|P_{wh} - P_m|},$$

represent additional algebraic equations related to flow, whereas

$$F_1 = 0.158\left(\frac{\rho L_1 q^2}{D_1 A_1^2} \cdot \left(\frac{\mu}{\rho D_1 q}\right)^{1/4}\right) \quad \text{and}$$

$$F_2 = 0.158\left(\frac{\rho L_2 q^2}{D_2 A_2^2} \cdot \left(\frac{\mu}{\rho D_2 q}\right)^{1/4}\right),$$

are related to friction and

$$H = C_H \cdot H_0\left(\frac{f_q}{f_0}\right)^2, \quad H_0 = -1.2454 \cdot 10^6 \cdot q_0^2$$
$$+7.4959 \cdot 10^3 \cdot q_0 + 9.5970 \cdot 10^2,$$

$$C_H = -0.03\mu + 1, \quad q_0 = \frac{q}{C_q} \cdot \left(\frac{f_0}{f_q}\right),$$

$$Cq = 2.7944\mu^4 - 6.8104\mu^3 + 6.0032\mu^2 - 2.6266\mu + 1,$$

$$Cp = -4.4376\mu^4 + 11.091\mu^3 - 9.9306\mu^2 + 3.9042\mu + 1,$$

are related to ESP characteristic curves and viscosity correction factors. A description of model parameters and constants used are summarized in Table 2. Some of them are fixed (like pipe diameter and height related to well structure) and others are obtained during well-tests (ρ, PI, μ, b_1, b_2). The uncertainties associated with model parameters establish the inverse PINN problem, which focuses on tracking relevant parameters of well and fluid properties as PI and ρ using the model and measurements available. These parameters are time-varying and the first one can be used to estimate well production lifetime.

Table 2
Model parameters and constants (Binder et al., 2015).

Parameter	Value	Description
g	9.81	Gravitational acceleration constant $[m/s^2]$
Cc	2×10^{-5}	Choke valve constant
A1	0.008107	The cross-section area of pipe below ESP $[m^2]$
A2	0.008107	The cross-section area of pipe above ESP $[m^2]$
D1	0.1016	Pipe diameter below ESP [m]
D2	0.1016	Pipe diameter above ESP [m]
hw	1000	Total vertical distance in well [m]
L1	500	Length from the reservoir to ESP [m]
L2	1200	Length from the ESP to choke [m]
V_1	4.054	Pipe volume below ESP $[m^3]$
V_2	9.729	Pipe volume above ESP $[m^3]$
f_0	60	ESP characteristics reference freq [Hz]
b_1	1.5×10^9	Bulk modulus below ESP [Pa]
b_2	1.5×10^9	Bulk modulus above ESP [Pa]
M	1.992×10^8	Fluid inertia parameters [kg/m4]
ρ	950	The density of the produced fluid [kg/m3]
P_r	1.26×10^7	Reservoir pressure[Pa]
PI	2.32×10^{-9}	Well productivity index [m3/s/Pa]
μ	0.025	Viscosity $[Pa \cdot s]$

The sequences of data available in the oil lift information system fit RNNs requirements perfectly because they are naturally suited to processing time-series data.

Let $\hat{\mathbf{y}} = [\hat{P}_{bh}(k), \hat{P}_{wh}(k), \hat{q}(k)]^T$, $\mathbf{u}(k) = [f(k), Z_c(k), P_m(k), P_r(k)]^T$, and $\hat{\mathbf{y}}_{\mathbf{r}} = [\hat{P}_{bh}(k), \hat{P}_{wh}(k)]^T$. The proposed VFM states output $\hat{\mathbf{y}}$ is obtained from historical sequences formed by $\mathbf{y_r}$ and $\mathbf{u}$.

This model is a natural multi-scale system if the scaling procedure does not put all variables in a balanced form, so during network training, some residue can be more relevant than others, therefore, affecting convergence. The loss function (9) must have additional weights set firstly to achieve suitable learning curves, then:

$$\mathcal{L}(\mathcal{b}, \mathcal{f}) = w_{bc}\mathcal{L}_{bc} + w_1\mathcal{L}_{y_1} + w_2\mathcal{L}_{y_2} + w_3\mathcal{L}_{y_3}, \tag{12}$$

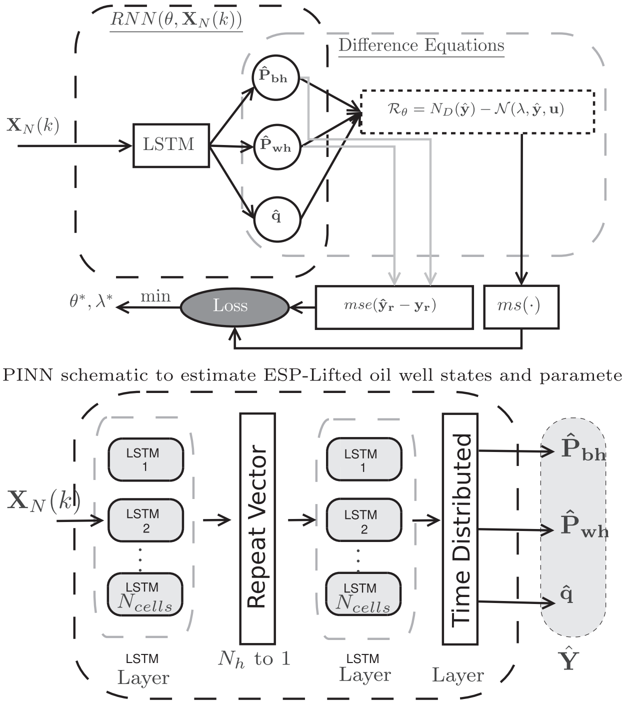
where $w_{bc}, w_1, w_2, and w_3$ are weights of the parts of the loss function (12) corresponding to boundary conditions (initial conditions and observed values) and residues of ODE. If these weights are carefully chosen, some training problems can be avoided, as pointed out by van der Meer et al. (2021); Wang et al. (2020).

Remark 3. Notice that, if the optimizer minimizes the balanced form of Eq. (12) such that $\hat{P}_{bh}(k)$ and $\hat{P}_{wh}(k)$ fit with their observed values, $\hat{q}(k)$ can also converge to its expected value.

Figure 6 a illustrates the proposed architecture while the box (Figure 6b) is composed of two layers with N_{cells} LSTM cells each. The repeat vector layer converts the number (N_h) of input time steps to one, and the time distributed layer arranges the output sequence for each state with low effort.

The preprocessing stage and data normalization are standard practices necessary to facilitate network training (Hagan et al., 2014). After scaling, the training dataset can be built based on Eq. (7) and following NN operation mode requirements.

The training is performed using Adam, the popularized first-order optimizer, followed by the prominent L-BFGS to achieve better accuracy


$RNN(\theta, \mathbf{X}_N(k))$
Difference Equations
$\mathbf{X}_N(k)$
LSTM
$\hat{\mathbf{P}}_{\mathbf{bh}}$
$\hat{\mathbf{P}}_{\mathbf{wh}}$
$\hat{\mathbf{q}}$
$\mathcal{R}_\theta = N_D(\hat{\mathbf{y}}) - \mathcal{N}(\lambda, \hat{\mathbf{y}}, \mathbf{u})$
θ^*, λ^*
min
Loss
$mse(\hat{\mathbf{y}}_\mathbf{r} - \mathbf{y}_\mathbf{r})$
$ms(\cdot)$
LSTM 1
LSTM 2
LSTM N_{cells}
Repeat Vector
Time Distributed
$\hat{\mathbf{Y}}$
LSTM Layer
N_h to 1
Layer

(a) PINN schematic to estimate ESP-Lifted oil well states and parameters.

(b) LSTM structure.

Fig. 6. PINN-oriented VFM for ESP-lifted oil well systems architecture.

Table 3
The standard deviation of noise measurements.

Signal	P_{bh}	P_{wh}	q	P_{man}	z	f
Standard deviation	0.01	0.01	0.01	0.01	0.005	0.001

with this quasi-Newton method. Some problems can consider L-BFGS prohibitive because of its full batch training requirement (van der Meer et al., 2021). The validation loss computing utilizes mean square error as a metric.

4. Results and Discussion

A phenomenological model simulation using amplitude-modulated pseudorandom signals (APRBS) as exogenous inputs and the parameters listed in Table 2 produced a dataset with 6000 samples presented in Figure 7. The sampling time considered was one second. Part of this dataset (70%) is dedicated to training and the remaining to validation. In addition, a noisy measurements scenario similar to Binder et al. (2015) is defined by the following equation:

$$s_n = (1 + n) \cdot s_r, \tag{13}$$

where s_r, s_n, and n are the signal, noisy signal, and noise, respectively. These noises have a normal distribution and standard deviation described in Table 3. The inverse PINN problem considers a mismatch of 10% of PI and ρ values.

The set of weights from Eq. (12) is defined by $S = [w_{bc}, w_1, w_2, w_3]$. The set $S_0 = [1, 1, 1, 1]$ is used to start the training using the Adam optimizer until 300 epochs, and the L-BFGS assumes the remaining num-

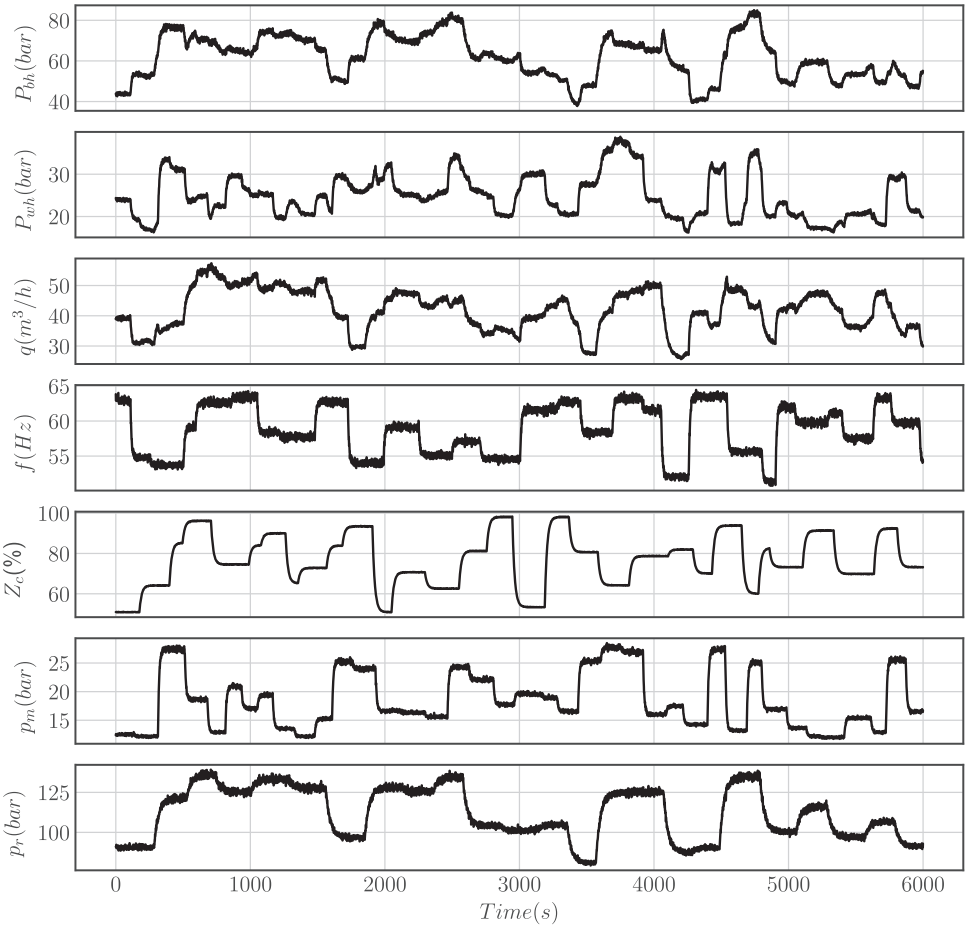


Fig. 7. Dataset used for model training and validation.

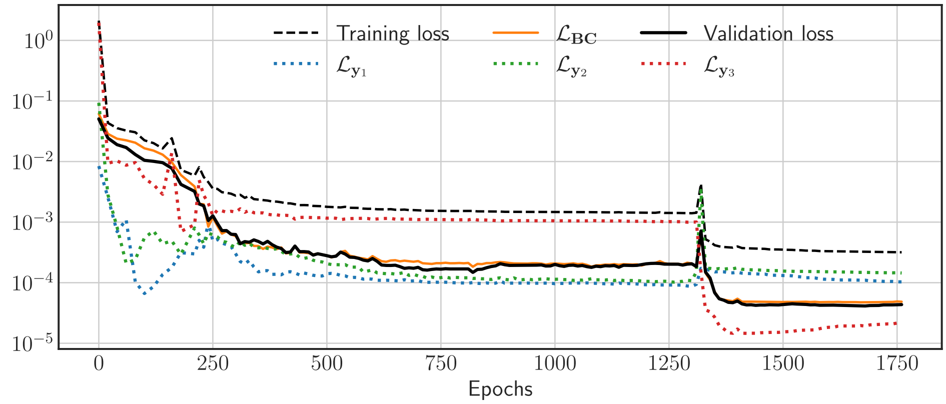


Fig. 8. The training losses affected by the change of the set of weights.

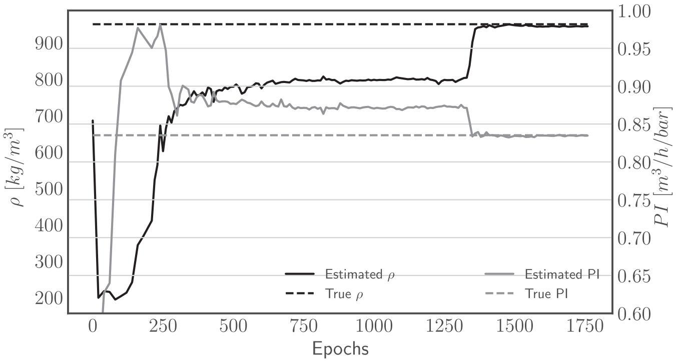


Fig. 9. Weights influence on parameters tracking. The parameter estimation achieved $PI = 0.8342 m^3/h/bar$ (0.12% of mismatch) and $\rho = 946.5 kg/m^3$ (0.37% of mismatch).

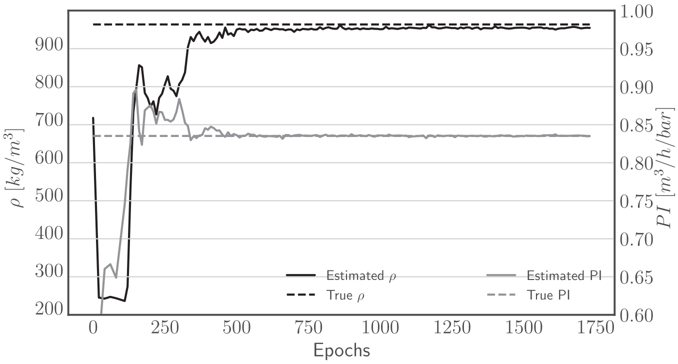


Fig. 11. Model parameters ρ and PI converging to their true values during the training process. $PI = 0.836\ m^3/h/bar$ (0.095% of mismatch) and $\rho = 940.5\ kg/m^3$ (1% of mismatch).

ber of epochs. Lu et al. (2021) solve many PINN problems using Adam followed by L-BFGS, and Markidis (2021) comments that L-BFGS can converge fast to local minima if used directly. For this reason, it is recommendable to start the training with Adam to avoid local minima and use L-BFGS to get a refined solution faster. Additional tuning trials find a new set of weights $S_1 = [1, 2, 1.9, 0.01]$. The weights change from S_0 to S_1 at epoch 1300, improving convergence significantly, as noted in Figure 8. Observe that training loss is bigger than validation loss because the first is affected by the ODE scaling and nonzero residues given by Eq. 8 computed using numerical differentiation.

The evolution of Eq. (12) terms along the training depicted in Figure 9 allows for identifying the imbalance sources. Firstly $\mathcal{L}_{y_3}$ must be penalized because a visual inspection between epochs 750 and 1250 could infer that each term should decrease closer to the other ones. Before the weights changed, $\mathcal{L}_{y_3}$ dominated the training.

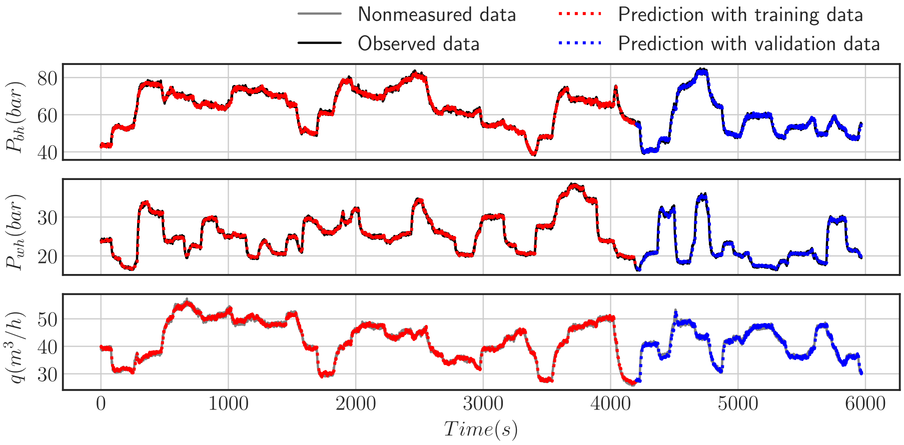


Fig. 10. Predictions using training and validation data. Note that both nonmeasured data (q in gray) and observed data (P_{bh} and P_{wh} in black) are difficult to see because of the good fit.

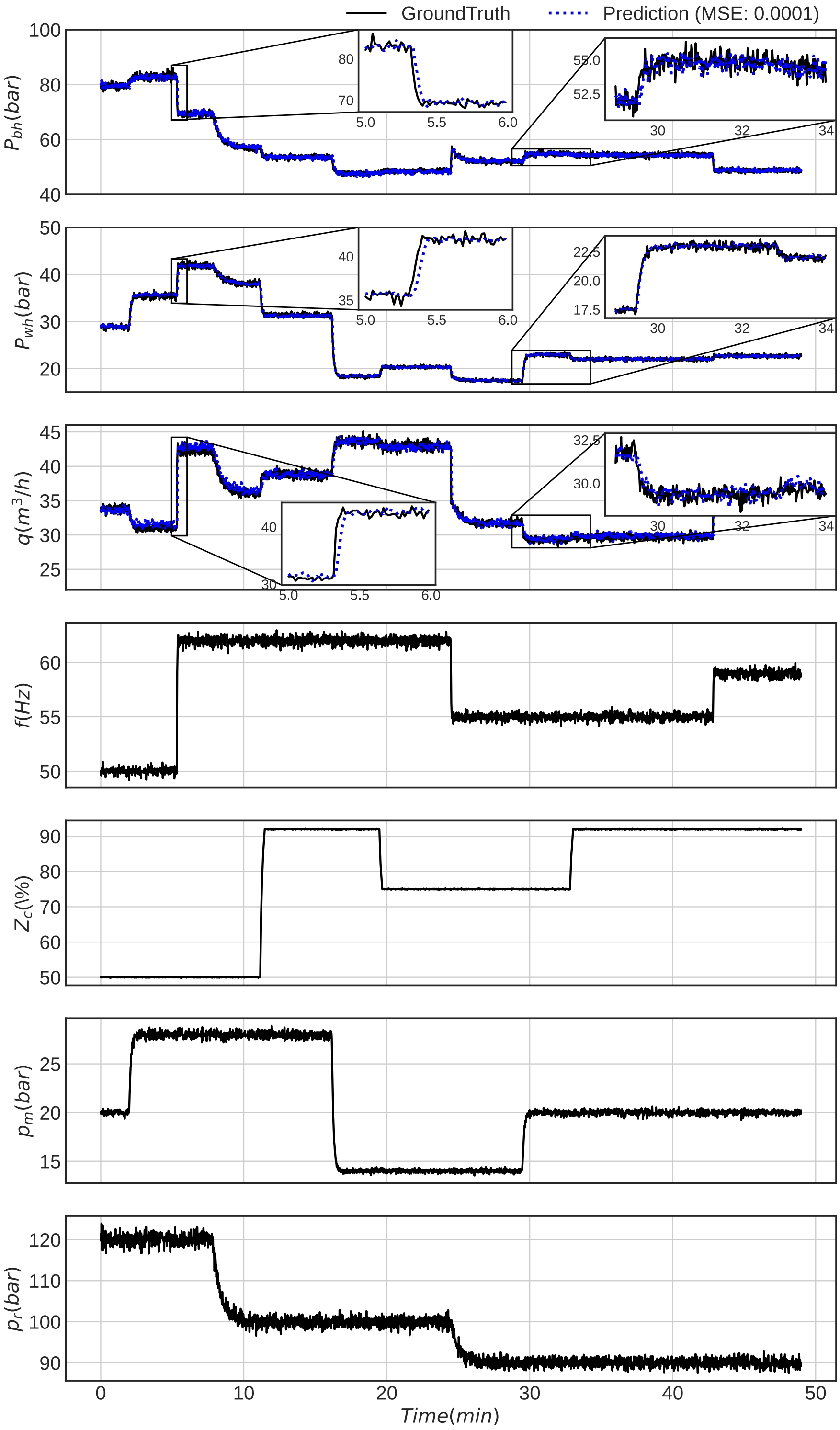


Fig. 12. States predictions obtained under typical operating conditions defined by a new test dataset.

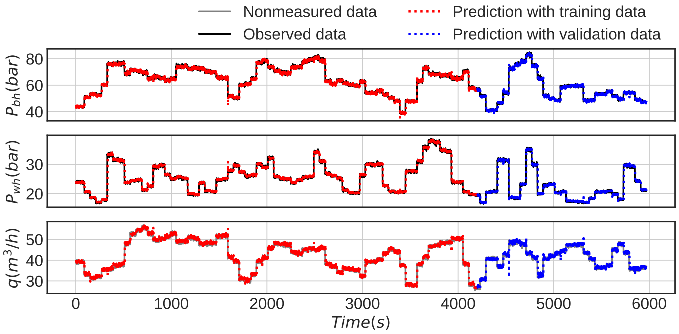


Fig. 13. Predictions using training and validation (dataset with one-minute samples). Note that nonmeasured data (q in gray) and observed data (P_{bh} and P_{wh} in black) are difficult to see because of the good fit.

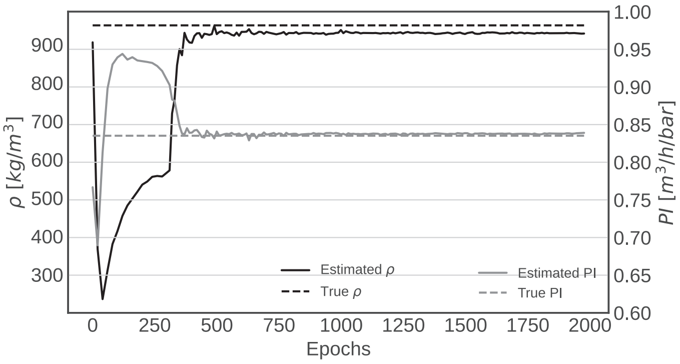


Fig. 14. Model parameters ρ and PI converging to their true values during the training process (dataset with one-minute samples). $PI = 0.839\ m^3/h/bar$ (0.42% of mismatch) and $\rho = 930.4\ kg/m^3$ (2% of mismatch).

Figure 9 shows that the parameter tracking process also demonstrates the weights changing effect. It is noted that the parameters get closer to their true values after the weights change.

To find a reasonable value for N_h, independent runs indicated that $N_h = 30$ satisfies with low mse. The prediction of the first output timestep does not require a long history as input. There is an $N_h =$ value beyond that no improvements in mse values can be noted, which means that a history of 30 samples is enough to predict the first future sample.

The simpler model which explains the data was obtained by the growing approach which defines the number of LSTM cells (N_{cells}) (Hagan et al., 2014). It considers the law of parsimony to get the simplest structure that achieves acceptable performance. In this case, a lower validation loss is obtained using $N_{cells} = 15$ with sufficient accuracy after five runs using N_{cells}= 2, 4, 6, 8, 10, 15.

It is relevant to mention that starting the training with S_1 caused convergence problems either with Adam or L-BFGS. From now, the training starts with Adam optimizer using S_0 until 300 epochs and changes to L-BFGS with S_1. Figure 10 shows predictions of each state output where red and blue lines correspond to training and validation data, respectively. This amount of data achieves enough performance (low validation loss 4.74×10^{-5}) after about 1639 epochs.

Figure 14 depicts the evolution of the parameters obtained with S_1. Notice that inferred parameters are in good agreement with their target values.

The dataset depicted in Figure 7, used for training, has too much dynamic information about the process, therefore distant from operational reality. To represent the well dynamics closer to operational reality, a new dataset was collected by simulation and applied to the model trained before. The results illustrated in Figure 12 demonstrate that the model makes accurate predictions using the new dataset, confirmed by low test error ($mse = 10^{-4}$). A sampling time of one minute is considered to introduce additional uncertainty. The same dataset depicted in Figure 7 was downsampled to 60 seconds and used for training. In comparison to the model obtained with data sampled with one second, this model, as expected, presents a worse validation loss ($mse = 0.7 \times 10^{-3}$), but estimates the parameters accurately and retains the ability to predict, see Figures 11 and 15. The prediction quality is maintained even during transients with negligible steady-state offset. Note that as reservoir pressure decrease all states (P_{bh}, P_{wh}, and q) also decrease. An increase in manifold pressure causes an increase in P_{bh} and P_{wh} but reduces the flow rates. As expected, the pump pressure gain influences the flow rate directly as the choke valve opening affects the P_{wh}.

All these model characteristics validate the potential of PINN-oriented VFM as a representation of system data. A nonmeasured state (q) is estimated accurately thanks to the mathematical model supported by other measured states demonstrating this approach's potential as VFM. This approach cannot insert an arbitrary number $N_{\mathcal{R}}$ of collocation points in the training dataset as conventional PINN can; however, it achieves acceptable results using a small dataset combined with a mathematical model. Training time was not a problem for this model, because it is trained in less than twenty minutes in a google colaboratory free account.

5. Conclusions

This paper presents a PINN-oriented VFM method using RNN to model a dynamic system representing an ESP-lifted oil well system. The results demonstrate its potential application as a hybrid soft sensor based on data and prior mathematical knowledge of phenomena that governs the process. It is important to note this approach does not require measurements of all states. In addition, the model uncertainties can be minimized through parameter tracking along the training process supported by available measured data. This aspect is relevant in oil field production systems since flow measurement remains a challenge. The proposed PINN-oriented VFM is an alternative to full data-driven VFM that estimates variables and model parameters using small datasets. Unlike the majority of data-driven VFMs, this approach allows average flow rate prediction even during the transient regime. The ESP-lifted oil well systems have a very known model whose most of the parameters are available in well test reports. Operational data (exogenous inputs and states) are present on a supervisory system ready to use that makes its implementation easier and with low intervention. Since this architecture uses LSTM, it has the potential to be used in, for instance, model predictive control systems, since it can compute predictions faster than non-linear solvers. It allows exploring distributed parameters models and multi-phase flow measuring. Although the case study is based on a single-phase scenario, this method is flexible and adaptable to accept multi-phase models with more tuning efforts. In future investigations, a more realistic scenario and extensions to different lift methods such as gas-lift or PCP (progressive cavity pump) are required to show the applicability of this approach.

Declaration of Competing Interest

The authors declare they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed by the Brazilian Agency of Petroleum and Biofuels (ANP) through the Human Resources Program (PRH 35.1) and in part by the Coordenao de Aperfeioamento de Pessoal de Nvel Superior Brasil (CAPES) Finance Code 001.

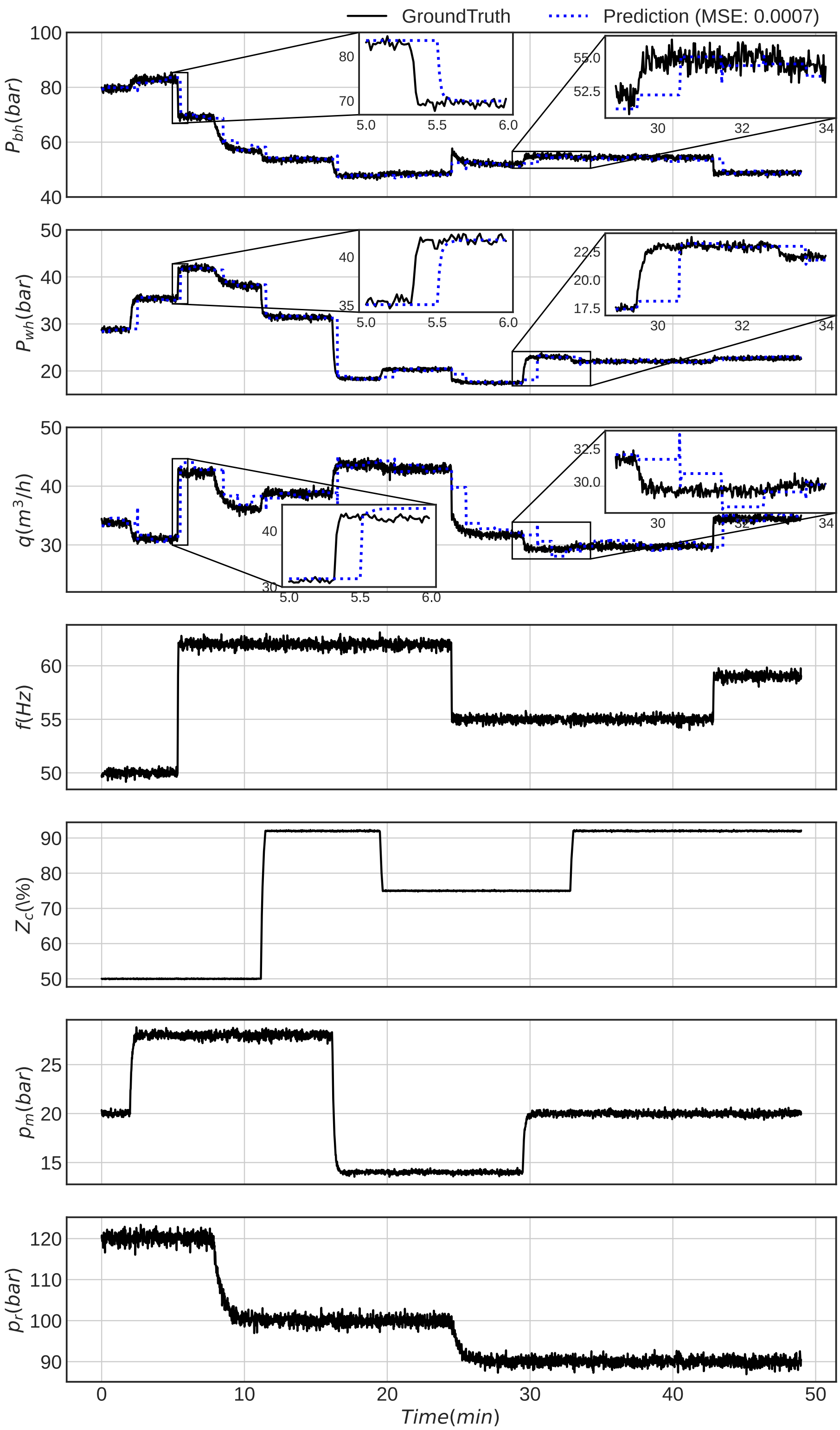


Fig. 15. States predictions obtained under typical operating conditions defined by a new test dataset with one-minute samples.

References

AL-Qutami, T.A., Ibrahim, R., Ismail, I., Ishak, M.A., 2018. Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing. Expert Systems with Applications 93, 72–85.

Andrade, G.M., de Menezes, D.Q., Soares, R.M., Lemos, T.S., Teixeira, A.F., Ribeiro, L.D., Vieira, B.F., Pinto, J.C., 2022. Virtual flow metering of production flow rates of individual wells in oil and gas platforms through data reconciliation. Journal of Petroleum Science and Engineering 208, 109772.

Arnold, F., King, R., 2021. State space modeling for control based on physics-informed neural networks. Engineering Applications of Artificial Intelligence 101, 104195.

Bikmukhametov, T., Jschke, J., 2020. First principles and machine learning virtual flow metering: A literature review. Journal of Petroleum Science and Engineering 184, 106487.

Binder, B.T., Pavlov, A., Johansen, T., 2015. Estimation of flow rate and viscosity in a well with an electric submersible pump using moving horizon estimation. IFAC-PapersOnLine 48, 140–146.

Blechschmidt, J., Ernst, O. G., 2021. Three ways to solve partial differential equations with neural networks – a review. 2102.11802.

Chaudhry, A., 2004. Oil well testing handbook. Elsevier.

Cheng, C., Zhang, G.-T., 2021. Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. Water 13. doi:10.3390/w13040423.

Góes, M.R.R., Guedes, T.A., deAvila, T.C., Vieira, B.F., Ribeiro, L.D., de Campos, M.C., Secchi, A.R., 2021. Virtual flow metering of oil wells for a pre-salt field. Journal of Petroleum Science and Engineering 203, 108586.

Grimstad, B., Hotvedt, M., Sandnes, A.T., Kolbjørnsen, O., Imsland, L.S., 2021. Bayesian neural networks for virtual flow metering: An empirical study. arXiv preprint arXiv:2102.01391.

Hagan, M., Demuth, H., Beale, M., De Jesús, O., 2014. Neural Network Design. Martin Hagan.

Hotvedt, M., Grimstad, B., Imsland, L., 2020. Developing a hybrid data-driven, mechanistic virtual flow meter-a case study. IFAC-PapersOnLine 53 (2), 11692–11697.

Hotvedt, M., Grimstad, B., Ljungquist, D., Imsland, L., 2022. On gray-box modeling for virtual flow metering. Control Engineering Practice 118, 104974. doi:10.1016/j.conengprac.2021.104974.

Langtangen, H.P., Pedersen, G.K., 2016. Scaling of Differential Equations. Springer, University of Oslo.

Lu, L., Meng, X., Mao, Z., Karniadakis, G.E., 2021. DeepXDE: A deep learning library for solving differential equations. SIAM Review 63 (1), 208–228. doi:10.1137/19M1274067.

Markidis, S., 2021. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? Frontiers in Big Data 4. doi:10.3389/fdata.2021.669097.

van der Meer, R., Oosterlee, C., Borovykh, A., 2021. Optimally weighted loss functions for solving pdes with neural networks. 2002.06269.

Monteiro, D.D., Duque, M.M., Chaves, G.S., Ferreira Filho, V.M., Baioco, J.S., 2020. Using data analytics to quantify the impact of production test uncertainty on oil flow rate forecast. Oil & Gas Science and Technology–Revue dIFP Energies nouvelles 75, 7.

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686–707.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566.

Rnning, R., 2011. Automatic Start-up and Control of Artificially Lifted Wells. Department of Engineering Cybernetics, Norwegian University of Science and Technology.

Toskey, E.D., 2012. Improvements to deepwater subsea measurements rpsea program: evaluation of flow modelling. In: Offshore Technology Conference. OnePetro.

Wang, S., Teng, Y., Perdikaris, P., 2020. Understanding and mitigating gradient pathologies in physics-informed neural networks. 2001.04536.

Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V., 2020. Integrating scientific knowledge with machine learning for engineering and environmental systems. https://arxiv.org/abs/2003.04919. 10.48550/ARXIV.2003.04919.