Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Engineering Cybernetics

Morten Fredriksen

Developing a hybrid, gray-box model of a production wellbore

A case study using historical production data

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Imsland
Co-supervisor: Mathilde Hotvedt

June 2021

NTNU
Norwegian University of
Science and Technology

Solution
Seeker

Morten Fredriksen

Developing a hybrid, gray-box model of a production wellbore

A case study using historical production data

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Imsland
Co-supervisor: Mathilde Hotvedt
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

Accurately predicting the multiphase mass flow from petroleum production wellbores is of great interest in the production of hydrocarbons. Traditional approaches to this problem include well testing, and in later times, mulitphase flow meters. In this thesis, a virtual flow meter for a production wellbore using a hybrid modeling technique was developed. This hybrid model leverages the advantages of both first-principle modeling and data-driven modeling techniques. The model was trained using a dataset sourced from a multiphase flow meter. The resulting predictive performance of the model is promising, especially considering the light-weight implementation and retention of the interpretability of the model. This work has shown the viability of using the hybrid modeling technique for mulitphase mass flow, and the results my be extended to hybrid model implementations of other systems.

Sammendrag

I produksjonssystemer for hydrokarboner er det en betydelig fordel å kunne måle multifaseflyten ut av brønnbanen. Tradisjonelt sett har man brukt brønntesting for å finne denne masseflyten, og i senere tid også fastmonterte multifasemetere. I denne oppgaven har et virtuelt multifasemeter blitt utviklet ved hjelp av hybrid modelleringsteknikk. Denne teknikken utnytter fordelene med både mekanistisk- og datadreven modellering. Hybridmodellen har blitt trent på et datasett hentet fra et multifasemeter. Med denne modellen har vi oppnådd lovende resultat for massflytprediksjonen, spesielt i lys av enkelheten til hybridmodellen. Videre beholder også hybridmodellen tolkbarheten assosiert med mekanistiske systemmodeller. Dette arbeidet har vist at man kan bruke hybrid modelleringsteknikk til a predikere multifaseflyt, og etablert grunnarbeid for å kunne utvide denne typen modellering til andre systemer.

Acknowledgements

I would like to thank my supervisor Professor Lars Imsland at NTNU for suggesting this thesis, and for continued guidance and insight during its development. Furthermore, I would like to express my gratitude towards Solution Seeker for providing me with the dataset making this case study possible. Being able to work with real-life production data has been motivating, and helped me contextualize the work I have done. Finally, I would like to extend my deepest gratitude to my co-supervisor Mathilde Hotvedt, who throughout this project has provided me with insight and advice, both of which has been instrumental in the completion of this thesis.

Contents

Nomenclature

Abbreviations

GOSP	Gas oil separation plant
MAE	Mean absolute error
MAP	Maximum a posteriori
MAPE	Mean absolute percentage error
MPFM	Multi-Phase Flow Meter
ReLU	Rectified Linear Units
VFM	Virtual Flow Meter

Subscripts

G	Gas
W	Water
O	Oil
BH	Bottomhole
WH	Well head
s	Scaled variable
DD	Data-driven
FP	First-principles
NN	Neural network
LP	Learnable parameters

WP	Well parameters

Symbols: Well Parameters

α	Expected MAPE of measurement instrument	
μ	Fluid viscosity	$N * s/m^2$
ϕ	Model parameters	
ρ	Fluid density	kg/m^3
σ	Standard deviation	
σ_e	Standard deviation of noise	
b	Neural network bias	
D	Wellbore diameter	m
f	Friction factor	
g	Gravitational constant	m/s^2
g_{DD}	Data-driven mass flow rate model	
g_{FP}	First-principles mass flow rate model	
g_{FP}	Model output	
g_H	Hybrid mass flow rate model	
h_f	Head loss due to pipe friction	m
L	Total length of wellbore	m
M_g	Molar mass of gas	$g/kmol$
p	Pressure	Pa
R	Universal gas constant	$J/kmol$
Re	Reynolds number	
T	Temperature	K
v	Fluid velocity	m/s
w	Neural network weight	
x	Model inputs	
y	Model output	
Z	Compressibility factor	
z	Wellbore height	m

List of Figures

Chapter 1

Introduction

Never before has there been produced as much hydrocarbons as today. Although public and political perceptions are leaning towards greener energy solutions, many industry experts forecast a continued reliance on oil and gas alongside emerging renewables in order to power the global electrification (Equinor, 2021). As such, the extraction and processing of hydrocarbons will most likely remain an important industry for decades to come. However, facing ever stricter emission requirements and fewer field opportunities, optimizing existing infrastructure and mature fields has become crucial to sustaining economic viability. As fields mature, the inclusion of more wells, water injection, and other revitalization schemes increase the operational complexity, leading to higher reliance on accurate production data. Also fueled by the industry-wide adoption of digital analysis tools, data is increasingly becoming an integral part of many production systems. As such, acquiring accurate production data is of vital strategic- and operational importance (Mohammadpoor and Torabi, 2020).

Although there are many production parameters to consider, one of the most useful characteristics is the multiphase mass flow rate from producing wells. As the goal of any production plant is to maximize profits, the production output is often only restricted by the capacity of the on-site processing equipment. For instance, being able to reduce the output of gas-producing wells when the gas compressor capacity is maxed out is essential for keeping the field operating at peak profitability. A thorough understanding of the multiphase mass flow rate can be used to gauge the current reservoir conditions and production rate, and can help keep the plant operating at maximum efficiency. The production rate over time is also instrumental in production decline analysis (Guo et al., 2007).

There exists several methods to measure the multiphase flow rate. Traditionally, the multiphase mass flow is measured using what is called a well testing method. Although oil well testing is a varied engineering field, the use of a test separator is a common and proven strategy (Ross, 2010). This method redirects parts of the production stream to a separate testing assembly, where desired well characteristics are measured. The prevalent technology utilized in the North Sea is the three-phase test separator, which enables the production engineers to monitor the single-phase flows for oil, gas, and water by separating the three phases. This method of measuring the multiphase flow is popular due to the high degree of certainty associated with the obtained measurements. Still, there are a number of drawbacks to the use of test separators (Ross, 2010; Falcone et al., 2013). First and foremost, test separators represents a high capital- and operational expenditure, making them economically infeasible for fields with tight profit margins. Secondly, the results from a well test is only available after the phases have stabilized in the separator, delaying test results. Finally, well tests often interrupts production, and are thus usually performed at most once per month (Monteiro et al., 2020). As such, the test fluid represents a relatively small sample size of the actual production flow in that time frame.

The drawbacks mentioned above have motivated the development of multiphase flow meters (MPFMs). Often found in the wellhead, these devices aim to continuously predict the multiphase flow rates without the need to physically separate the different phases (Bikmukhametov and Jäschke, 2020b). Undoubtedly, this technology is of great interest to the oil and gas industry, as real-time prediction enables production engineers to manage and monitor the field to much higher efficiency. Another advantage of MPFMs are their much reduced footprint and upfront cost, compared to well testing methods (Falcone et al., 2013). A range of commercial multiphase flow metering systems have already been developed, and the field is still actively being researched. However, as with well testing there are several downsides to MPFMs, the main ones being low accuracy and sensor drift. Retuning the flow meter requires costly well intervention, which has led some industry experts to question the economic viability of the current MPFM technology (Ross, 2010). Regardless of the downsides, MPFMs are still an integral part of modern production systems due to the advantage of continuous monitoring. Commonly, MPFM are used in conjunction with test separators, where the results from the well tests are used to calibrate the flow meters (Corneliussen et al., 2005a). Although this method might seem sufficient, the multiphase flow meter accuracy can degrade considerably in between calibrations (Falcone et al., 2013). Consequently, is still room for improvement when using MPFMs.

One emerging solution to the multiphase flow metering problem is the virtual flow meter (VFM). The objective of virtual flow meters is to be able to predict the multiphase flow rate using mathematical models as apposed to physical metering. According to Bikmukhametov and Jäschke (2020b) there are several advantages to using VFMs. First, virtual flow meters usually do not require additional installation of hardware on site, thus reducing costs and operational complexity. Furthermore, the underlying mathematical models usually require relatively easily obtainable measurements such as well temperatures and pressures, as opposed to phase velocities and phase cross-sectional fractions often used in multiphase flow meters (Falcone et al., 2013). Virtual flow meters may also retain the desirable real-time capabilities of multiphase flow meters in some implementations.

Virtual flow meters are commonly split in two categories: first-principle models and data-driven models. Models derived from first principles use the known mathematical description of the underlying thermodynamics and fluid dynamics to model the expected mass flow rate through the system. The complexity of the mechanistic first-principle models can range from rather simple, such as the model implemented later in Section 3.1, to quite complex models such as the ones used in industry. The industry standard VFMs such as OLGA and FlowManager utilize different subsystem models along the production string such as reservoir inflow models and choke models, and combine them to a complete production system model (Bikmukhametov and Jäschke, 2020b). Using this complete system model one can achieve high flowrate accuracy of up to 95% (Amin, 2015). The main drawbacks to this kind of virtual flow metering approach are the thorough system knowledge needed to accurately develop the subsystem models, as well as the computational complexity of the final system model. In these kinds of models, accuracy often come at the expensive of model complexity. Furthermore, models based on first-principles have a tendency to be sensitive to factors such as the gas-to-oil ratio and have to be re-tuned for various well conditions. As such, Ross (2010) argues that the current state-of-the-art first-principle virtual flow meters may be more suited as supplementary tools to physical flow meters as apposed to fully independent metering solutions.

A different virtual flow metering approach is to use data-driven models. These kind of models utilize machine learning techniques to analyze wellbore data, and thereafter develops an input-output relation that matches the well behaviour. Examples of this technique for virtual flow metering can be found in AL-Qutami et al. (2018) and Andrianov (2018), where the latter also emphasized the predictive ability of data-driven models. That is, given a suitably fitted model, future datapoints can be forecast for over a significant time horizon. Another significant advantage of data-driven models is the non-reliance on detailed system specifications; only a production dataset is required. This reduces implementation costs compared to other VFM methods. Data-driven models may also be suited for capturing unknown phenomena impacting production output which may not be easily modeled by first-principles. Despite these useful properties, data-driven models are only as good as the data used to develop them, which poses a particular challenge in the oil and gas industry. Assuming sizeable sets of field production data are even available, they are often comprised of datapoints gathered from multiphase flow meters, which as mentioned previously can be error-prone (Ganat, 2016). Consequently, its is difficult to achieve any improved accuracy over the already installed MPFMs. Ideally, one would train on the more accurate data originating from well tests, however most fields does not have a sufficient amount of such datapoints for training purposes. Another significant challenge with data-driven virtual flow meters are their "black-box" nature. That is, a known alteration of input may not result in the expected change of output, as with mechanistic models. As such, the model is detached from any physical meaning and is therefore not easily interpretable. This interpretability is key to making data-driven VFMs attractive to operators, and research on this is ongoing (Pitarch et al., 2019; Bikmukhametov and Jäschke, 2020a; Sandnes et al., 2021).

Considering the advantages and drawbacks of these two modeling approaches, a third category of modeling has been proposed. This modeling technique aims to combine the interpretability and prior system knowledge of first-principle models with the flexibility of data-driven models. This kind of model is often referred to as a *hybrid model.* In Hotvedt et al. (2020a) a hybrid model is defined as:

Definition 1 (Hybrid Model)**.** A hybrid model combines equations from first-principles with generic mathematical structures, both in model development and application.

There is a distinction between development and application because it is not uncommon to do parameter estimation of first-principle models with production data and machine learning techniques. In fact, several examples exist in industry (Oliver et al., 2011). However, this approach does not lend itself to adaptive performance after said parameters have been estimated, that is, after development. Hybrid models add to this by also including data-driven black-box components to the system model as a whole, making it hybrid also in application. Such a model is also referred to as a "grey-box" model. One of the first instances of this kind of hybrid modeling can be found in Psichogios and Ungar (1992), where a combination of a first-principle model and a neural network was used to create a system model for a bioreactor. The research showed promise in that the number of training samples needed for the hybrid model were fewer than that of purely data-driven models, and with improved extrapolation capabilities. In more recent times, several studies have enquired about the feasibility of hybrid models in real-world applications. Solle et al. (2017) found promising results in using hybrid models as substitutes for more expensive first-principle models in plant control applications. There exists several examples of data-driven techniques being utilized in the oil and gas industry as well, but they are usually reserved for various parameter estimation schemes. For instance, Kanin et al. (2019) uses three surrogate machine learning models trained on laboratory data to calculate well pressure gradients, and shows the potential for wider applicability of this methodology as opposed to purely mechanistic models. However, hybrid models in the sense of the previously mentioned definition is harder to come by. Hotvedt et al. (2020a, 2021) used a hybrid modeling technique to develop a production choke model, demonstrating an increased model predictive performance. Bikmukhametov and Jäschke (2020a) showed that inclusion of first principles improved the predictive performance of machine learning algorithms by restricting the solution space to a more well defined region. They also found that hybrid modeling enhanced the estimation of well parameters as well as improving interpretability. Hotvedt et al. (2020b) found that inclusion of data-driven modeling techniques in a mechanistic model increased the predictive performance of the model whilst retaining physical interpretability of the model. Leveraging these advantages in development of a multiphase virtual flow meter will be the objective of this thesis.

1.1 Scope of Thesis

This project thesis will implement a hybrid model for the multiphase mass flow rate through a wellbore. A proof-of-concept for this model was developed in the precursor thesis Fredriksen (2021). In this preliminary study, a simplified first-principle model was combined with a neural network to produce a virtual flow meter. Furthermore, the model accuracy and viability was tested using data from an artificially produced dataset. The preliminary study also showed promising results when introduced to noisy data, suggesting a robust system model. Using the preliminary results as a baseline, this project will aim to expand the implementation to a real-world dataset from a producing oil well. This will be done by training a similar hybrid model as used in the precursor thesis on the real-world dataset. Prior to this, the dataset will be processed and cleaned such that it is suitable for steady-state modelling. The prediction output of the trained wellbore model will be evaluated against measurements of the mass flow rate available in the dataset. Different configurations of the hybrid model will be implemented and analyzed. Furthermore, a fully data-driven model will be trained for performance comparison. The analysis will aim to provide insight into what impact the model configuration has on the predictive performance of the hybrid model. The aforementioned analysis will also be extended to several wells, where the objective is to find common trends between all the models. Finally, an analysis of the interpretability of the hybrid model will be conducted and compared to the interpretability of the fully data-driven model.

1.2 Thesis Structure

The first three chapters of this thesis will lay the groundwork by introducing the theoretical aspects needed for implementing the hybrid model. Chapter 2 will present various concepts within petroleum production systems, mathematical modeling, machine learning, and hybrid modeling. This introduction will provide the context needed for later implementation of the wellbore hybrid model. Chapter 3 will derive the mathematical models for three wellbore models: a fully mechanistic, first-principles model, a fully data-driven model, and a hybrid model. Once the models have been constructed, the following chapters will aim to implement the models and present and analyze the results. Chapter 4 will provide an overview of the dataset used to train and test the models. The implementation of the model and experiment descriptions will be provided in Chapter 5, while the results will be presented in Chapter 6. Finally, the results will discussed in Chapter 7 before the findings are summarized in Chapter 8.

Chapter 2

Theory

This chapter will outline the theoretical background necessary to implement the hybrid wellbore model. Section 2.1 will start by giving a brief introduction to petroleum production systems. Following in Section 2.2, the mathematical prerequisites for modeling the wellbore will be introduced. Section 2.3 will present the concept of machine learning, and discuss implementation details surrounding neural networks. Finally, Section 2.4 will discuss hybrid system models.

2.1 Petroleum Production Systems

Hydrocarbons appear naturally across the globe in a variety of states, from shallow on-land reservoirs to deep-water underground reservoirs. The chemical composition of the hydrocarbons in different reservoirs also vary, such as heavy oil, gas compensate, and tar sands. Every oil well is a unique challenge, but despite the differences in reservoir composition, the overall structural design for the petroleum production systems used to extract the hydrocarbons tend to be similar. Figure 2.1 illustrates a simplified petroleum production system. They hydrocarbons flow from the reservoir through the wellbore. The opening from the wellbore pipe to the reservoir is referred to as the bottomhole, while the top of the well is called a wellhead. Typically the wellhead is situated on the surface, but may also be situated on offshore platforms or on subsurface production assemblies. From the individual wellheads the reservoir fluid flows into a production manifold, forming a stream of production fluid, as depicted in Figure 2.1. The flow rate from the individual well into the manifold is controlled by the production choke, located at each wellhead. The fluid is then passed on to the gas oil separation plant (GOSP) where the fluid is cleaned and sorted into marketable oil and gas. The GOSP usually consists of production separators where oil, gas, and water is separated, followed by storage-, treatment- and export facilities. Some fields also include test separators. In this case, a separate test manifold feeds into a test separator where the amount of oil, gas, and water can be accurately measured (Devold, 2007).

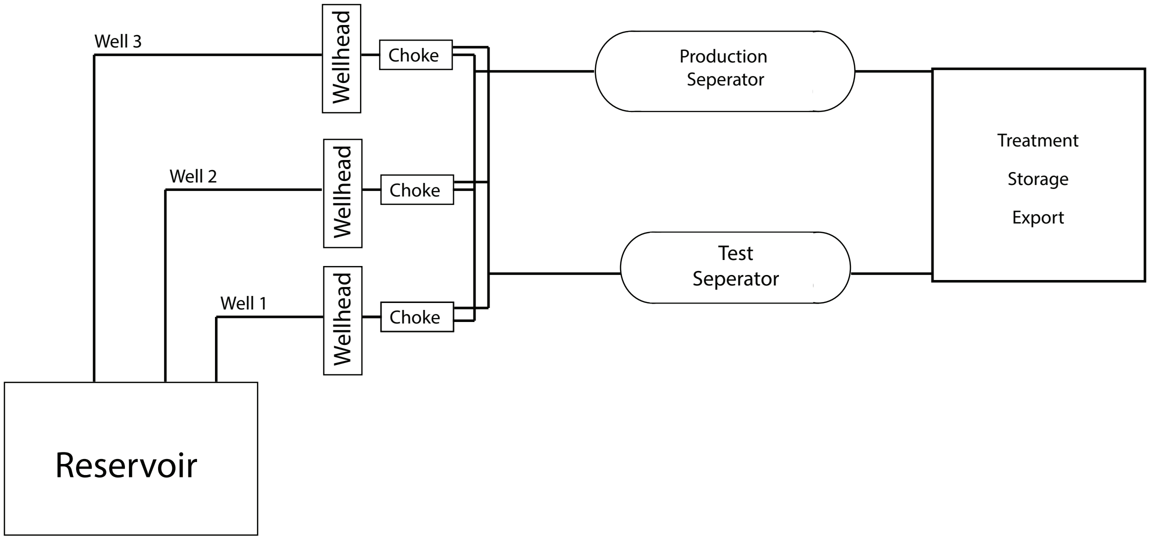


Figure 2.1: Illustration depicting a simplified petroleum production plant. Hydrocarbons flow from the reservoir through the wellbore to the wellhead. From there, the fluid flows through the choke into a production- or test separator. Finally, the separated fluids are sent to the treatment and storage facilities before the product is exported.

Optimizing the performance of the petroleum production system as a whole is critical for maintaining profitability. However, the production systems are often complex, especially with the inclusion of additional production assemblies, such as injection wells and gas treatment plants. One way of dealing with this complexity is to break the production system down into nodes, and use the sequential structure of the system to predict the fluid flow for the various sub-assemblies (Jansen, 2015). This project will focus on the first step of the production process: the flow of the reservoir fluid through the wellbore pipe. A model of the mass flow rate will be constructed, which aims to predict the outgoing mass flow rate out of a production well. The first step in that construction process is to introduce the governing fluid dynamics of flow through a pipe.

2.2 Fluid Dynamics

In this section, some of the governing equations for fluid flow through a pipe will be presented. The Bernoulli equation for flow along a streamline will form the basis of the model, and the restrictions associated with the use of the Bernoulli equation will be discussed in Section 2.2.1. The majority of this material is sourced from Cengel and Cimbala (2014). The following Section 2.2.2 will discuss some considerations to make when adapting the Bernoulli equation to a wellbore system.

2.2.1 The Bernoulli Equation

One way of describing flow through a vertical pipe is with the use of the Bernoulli equation. The Bernoulli equation provides an approximate way of describing the relationship between pressure p, velocity v, and elevation z of a fluid along a streamline. The relationship is derived from the principle of conservation of linear momentum, and assuming steady flow, is given by:

$$\int \frac{dp}{\rho} + \frac{v^2}{2} + gz = C, \tag{2.1}$$

where ρ is the fluid density, g is the gravitational acceleration, and C is a constant. Steady flow in this case means that every point along the streamline has the same amount of fluid flow over time. The Bernoulli equation expands on Equation (2.1) by also assuming incompressible flow, that is, a constant fluid density ρ along the streamline. Applying this to Equation (2.1) we obtain the Bernoulli equation:

$$\frac{p}{\rho} + \frac{v^2}{2} + gz = C. \tag{2.2}$$

Since the value of the constant C is equal over the streamline, any two points along the streamline can also be evaluated using the Bernoulli equation:

$$\frac{p_1}{\rho_1} + \frac{v_1^2}{2} + gz_1 = \frac{p_2}{\rho_2} + \frac{v_2^2}{2} + gz_2. \tag{2.3}$$

An illustration of the process described by the Bernoulli equation (2.3) is shown in Figure 2.2.

Although the Bernoulli equation is a useful tool for assessing fluid flow in a pipe, it comes with a few restrictions. In addition to the already mentioned steady- and incompressible flow limitations, the follow restrictions also apply:

- The fluid should flow along a streamline. For the case of fluid flow through a pipe, the flow should be irrotational for this to hold true.
- There should be negligible heat transfer to or from the fluid. The density of gas is proportional to temperature, so for regions where the temperature changes significantly the incompressibility of the fluid is no longer maintained.
- The system should not feature any artificial work, such as the inclusion of pumps or fans.
- There should be negligible frictional forces on the flow.

2.2.2 Fluid Dynamics in a Wellbore

Having presented the Bernoulli equation in Section 2.2.1, we are now ready to discuss its limitations in the context of a petroleum production well. For instance, an important distinction to make regarding the flow in a production wellbore is that it is rarely single-phase flow, but rather multiphase flow. The produced fluid typically consists of oil, water, and gas. As a result, the fluid density

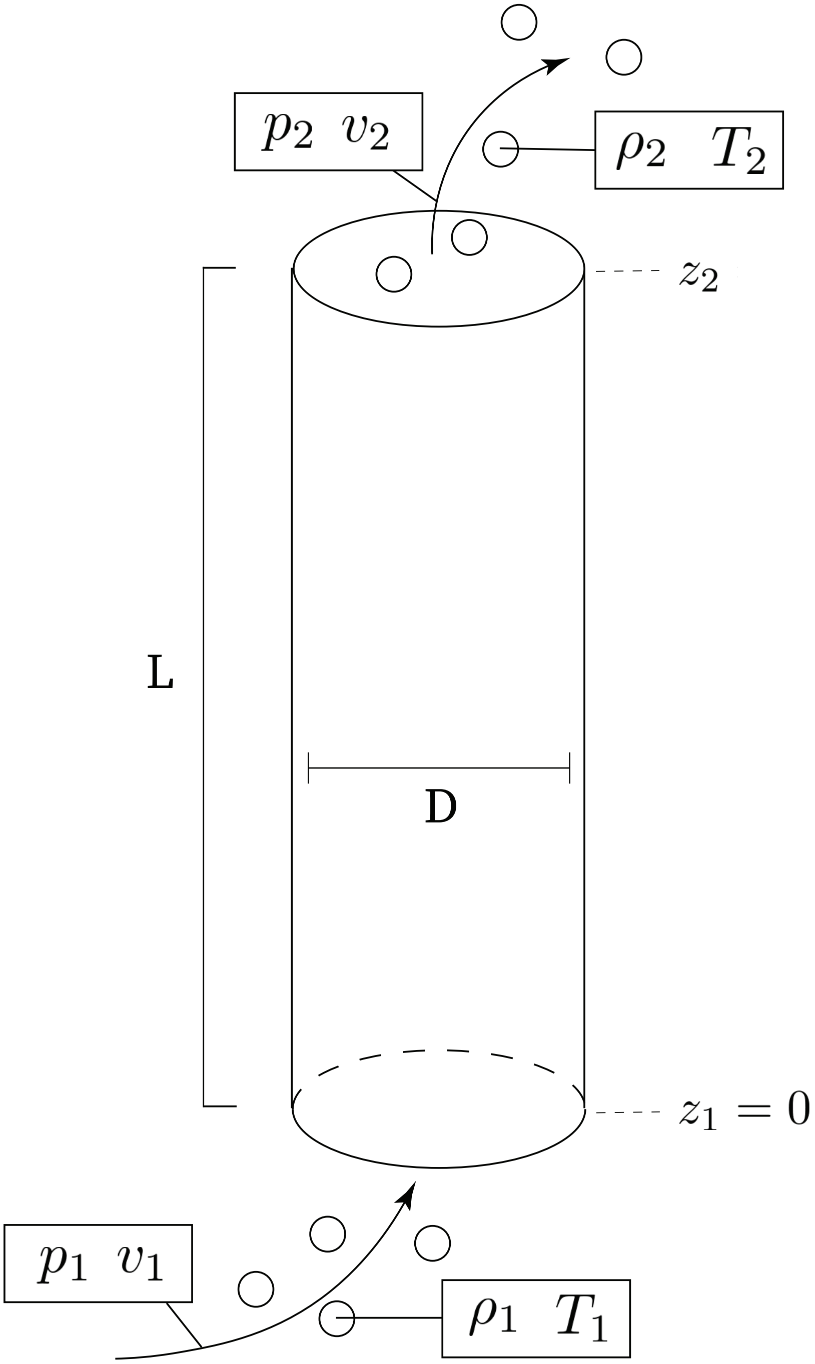


Figure 2.2: Fluid particles with temperature T_1 and density ρ_1 entering a pipe of length L and diameter D at height z_1 with velocity v_1 and pressure p_1. The fluid particles then leave the pipe at height z_2 with temperature T_2, density ρ_2, velocity v_2, and pressure p_2.

ρ becomes a product of the densities of the different phases in the fluid, and their corresponding mass fractions. One way of handling this complexity is to assume frozen flow. That is, the mass fractions of the different phases in the fluid remains the same from bottomhole to wellhead. By assuming a thoroughly mixed frozen flow, the homogeneous mixture density is expressed as (Awad and Muzychka, 2008):

$$\frac{1}{\rho} = \frac{\eta_G}{\rho_G} + \frac{\eta_L}{\rho_L}, \tag{2.4}$$

where η_G is the mass fraction of the fluid in gaseous form and η_L is the mass fraction of the fluid in liquid form, consisting of oil and water. We assume no other fluids present in the flow such that:

$$\eta_L + \eta_G = \eta_O + \eta_W + \eta_G = 1. \tag{2.5}$$

From Equation (2.4) it also follows that:

$$\eta_W = 1 - \eta_O - \eta_G. \tag{2.6}$$

Therefore, the water mass fraction η_W is no longer needed as a system input, seeing as it can be calculated from η_O and η_G.

Equation (2.4) separates the density of the fluid in liquid and gaseous forms. The liquid density ρ_L can be found using the water cut as follows:

$$wc = \frac{\eta_W}{\eta_W + \eta_O},$$

$$\rho_L = wc \cdot \rho_W + (1 - wc)\rho_O. \tag{2.7}$$

The gaseous fluid density ρ_G can be expressed through the real gas law (Baker et al., 2015):

$$\rho_G = \frac{pM_G}{ZRT}, \tag{2.8}$$

where M_G is the molar mass of the gas, R is the universal gas constant, and Z is the gas compressibility factor.

The Bernoulli equation is applicable to any system and height z, given that the requirements outlined in Section 2.2.1 hold true. However, for a production wellbore, the fluid flows through a pipe as apposed to freely along a streamline. As such, it is improbable that the pipe walls will not exert frictional forces on the fluid over the course of thousands of meters of pipe. As such, one might append a head loss term h_f to the Bernoulli equation (2.3) in order to compensate for the friction in the pipe (Dixon and Hall, 2010):

$$h_f = f\frac{Lv_2^2}{2Dg}. \tag{2.9}$$

Here L is the length of the pipe, D is the wellbore diameter, and g is the gravitational constant. The term f represents the wellbore friction factor. This factor is a measure of the amount of frictional force the pipe exerts on the fluid, and is given in large by the pipe wall roughness and current flow regime (Takacs, 2015). The flow regime is usually inferred from the Reynolds number:

$$Re = \frac{\dot{m}D}{A\mu}, \tag{2.10}$$

where μ is the fluid mixture viscosity. Assuming homogeneous flow, one way of estimating the friction factor f for laminar flow conditions is using the Hagen-Poiseuille equation (Awad and Muzychka, 2008):

$$f = \begin{cases} \frac{16}{Re} & Re < 2300 \\ \frac{0.079}{Re^{0.25}} & Re \geq 4000. \end{cases} \tag{2.11}$$

Homogeneous flow signify that all the phases in the multi-phase fluid flows together through the pipe, as opposed to being a relative motion between the phases. It is clear from this discussion that the inclusion of head loss h_f in the pipe and multiphase flow introduces more accurate modeling at the expense of computational complexity.

2.3 Machine Learning and Neural Networks

In this project, a large amount of well sensor data is available to be used in the construction of the wellbore model. In order to leverage this dataset to construct a better model, machine learning techniques will be utilized. Machine learning refers to the process of using available data to "train" an algorithm in such a way that the algorithm is able to perform a task at a certain performance level. The key concept of a machine learning is that the algorithm should be able to perform at a higher performance level given more data to learn from. This definition is not very specific, but points to one of the many benefits of machine learning: it is very versatile. However, in order to keep within the scope of this thesis, the following discussion on machine learning concepts will be confined to the specific use case of modeling the wellbore system. The material covered in this section is mostly sourced from (Goodfellow et al., 2016).

The four main components of a machine learning algorithm are: a model, a dataset, a cost function, and an optimization algorithm. The model refers to the learning task the machine learning algorithm will aim to optimize. In this project, the task will be to find the output of a function given a set of inputs, also known as regression. In order to keep training time to a minimum, the choice of model will be linear regression:

$$\hat{y} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b}, \tag{2.12}$$

where $\hat{y}$ is the estimated function output, $\boldsymbol{x}$ are the system inputs, $\boldsymbol{W}$ is a set of weights and $\boldsymbol{b}$ is a set of biases. Using linear regression allows for easier training of the model because the estimation algorithm can optimize on linear functions (Molnar, 2021). The next component is the dataset which contains the datapoints available for training and verifying the machine learning algorithm. The specific dataset used in the construction of the hybrid wellbore model will be discussed in Chapter 4. The two remaining machine learning components, cost function and optimization algorithm, will receive a more rigid introduction in Section 2.3.1 and Section 2.3.2, respectively.

2.3.1 Cost Function

In a machine learning algorithm, the cost function is a metric used to measure the performance of the estimation. This performance metric is used in training the network. Since the objective of the machine learning algorithm in this project is to perform regression, a suitable cost function is found by implementing maximum a posteriori (MAP) estimation.

Denoting the machine learning model as a function g with inputs $\boldsymbol{x}$ and parameters $\boldsymbol{\phi}$, we can express the model as:

$$y = g(\boldsymbol{x}, \boldsymbol{\phi}) + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma_e^2), \tag{2.13}$$

where y is the estimated output and ϵ is normally distributed noise. Furthermore, given a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with n measurements, where $\boldsymbol{x}_i$ is the set of inputs with a corresponding output y_i, we can perform maximum a posteriori estimation by utilizing Bayes' rule:

$$p(\phi|\boldsymbol{x}, y) = \frac{p(y|\boldsymbol{x}, \phi)p(\phi)}{p(y|\boldsymbol{x})}. \tag{2.14}$$

The MAP estimate is then given as:

$$\hat{\boldsymbol{\phi}}_{MAP} = \arg\min_{\boldsymbol{\phi}} \log p(y|\boldsymbol{x}, \phi) + \log p(\phi). \tag{2.15}$$

By extracting the log-likelihood and assuming independent Gaussian priors such that $\phi = \mathcal{N}(\bar{\phi}_i, \sigma_i)$, the MAP estimation for all parameters $\boldsymbol{\phi}$ can be expressed as:

$$\hat{\boldsymbol{\phi}}_{MAP} = \arg\min_{\boldsymbol{\phi}} \sum_{i=1}^{n} \big(y_i - g(\boldsymbol{x}_i, \boldsymbol{\phi})\big)^2 + \sum_{j=1}^{m} \frac{\sigma_\epsilon^2}{\sigma_i^2} \big(\boldsymbol{\phi}_j - \bar{\boldsymbol{\phi}}_j\big)^2. \tag{2.16}$$

The cost function in expression (2.16) can be interpreted as the mean squared error between the dataset output y and the estimated value produced by the network g, coupled with a regularization term for the parameters. This second regularization terms allows us to leverage a known prior distribution of the system parameters with mean $\bar{\phi}$ and standard deviation σ in the cost function. When used in the context of neural networks, this is often referred to as L2-regularization, where the fraction $\frac{\sigma_\epsilon^2}{\sigma_i^2}$ is replaced by a hyperparameter called the L2-regularization coefficient. In the case of parameter estimation of a sensor output, one may estimate the standard deviation of the noise using (Grimstad et al., 2021):

$$\sigma_e^2 = \Big(\sqrt{\frac{\pi}{2}\alpha|y_{ref}|}\Big)^2. \tag{2.17}$$

Here, y_{ref} is the reference reading used to calibrate the sensor and α is the expected mean percentage error of the instrument.

2.3.2 Optimization Algorithm

Equation (2.16) defines an optimization problem, where the objective is to find the set of parameters $\hat{\boldsymbol{\phi}}_{MAP}$ that minimizes the cost function. In practice, this is solved using an optimization algorithm. Most optimization algorithms used in machine learning are gradient based, where for every iteration of the algorithm, the gradient of the cost function is calculated. When a suitable gradient is found, a step proportional to the learning rate in the gradient direction is taken, and the loop reiterates until a local minima is found. Figure 2.3 illustrates this concept for a one-dimensional cost function. In physical terms, one can think of a ball rolling down a hill until it reaches the bottom of a valley. Note that stochastic gradient descent does not guarantee reaching the global minimum, only a local one. When implementing the optimization algorithm for a neural network, the cost function is not one-dimensional, but rather n-dimensional. However, gradient descent works conceptually in the same way even for a n-dimensional cost function, but at a higher computational price. As such, common practice in the field of machine learning is to use a form of *stochastic* gradient descent, where randomly selected subsets of the training data, called mini-batches, is used to calculate the gradient at any given iteration. The actual calculation of the gradients is done using back propagation, which is already implemented in most machine learning frameworks, and as such will not be explained in detail here. More information can be found in Goodfellow et al. (2016) and El-Amir and Hamdy (2020). For this project, we will use a variant of stochastic gradient descent called Adam (Kingma and Ba, 2015). This optimization algorithm incorporates adaptive moments in the training loop, allowing for faster convergence for convex functions. In physical terms, it can be seen as adding momentum to the ball rolling down the hill.

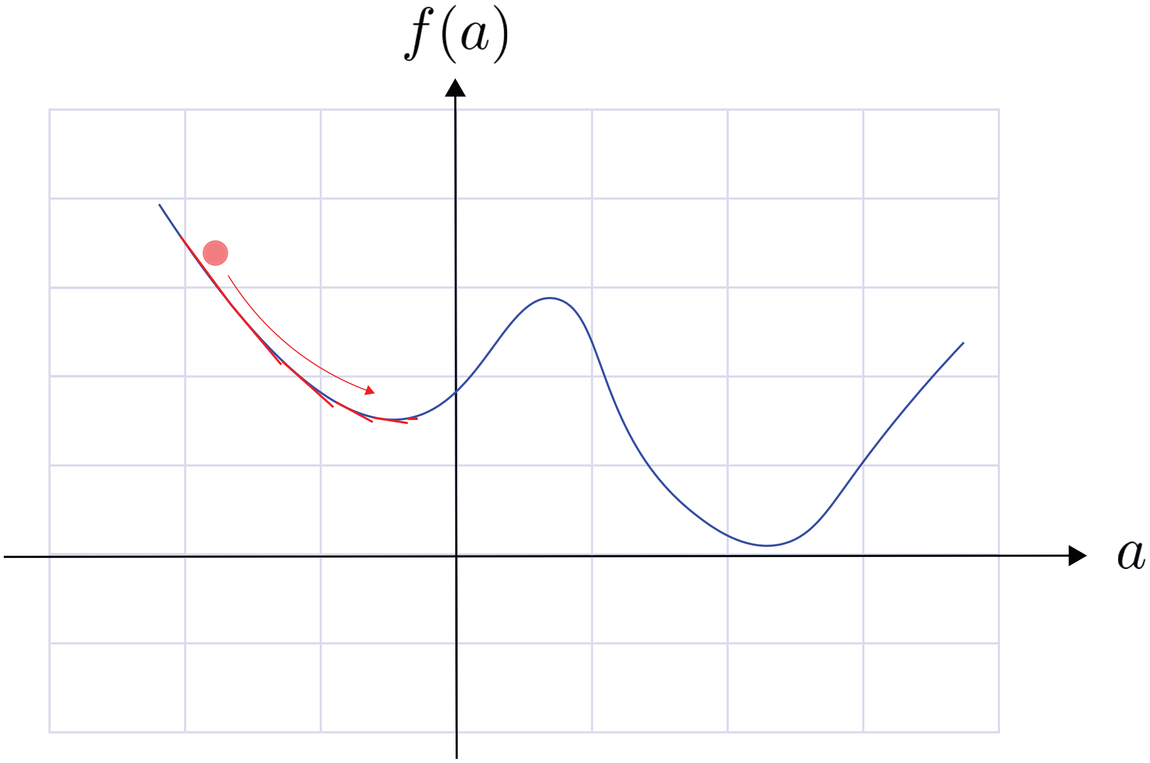


Figure 2.3: Gradient descent can be seen as a ball rolling down into a valley. The valley represents a local minimum.

2.3.3 Neural Networks

Deep learning is a class within machine learning algorithms where a common approach is neural networks. A neural network consists of several nodes sorted into layers. Every node in the network can be seen as a neuron, where the individual value associated with each node denotes the degree to which the neuron is activated. The key concepts of neural networks is its layered structure, where the activation of a node in one layer propagates to the next layer through a set of weights and biases. Figure 2.4 illustrates the connection from a layer m to a single node p in the next layer $m+1$. The value, or activation of an individual node p can be expressed as a function of the activations of the previous layer m as follows:

$$a_p^{(m+1)} = \big(w_{p,0}a_0 + w_{p,1}a_1 + \cdots + w_{p,n}a_n - b_p\big)^{(m)} = \big(\boldsymbol{w}_p\boldsymbol{a} - b_p\big)^{(m)}. \tag{2.18}$$

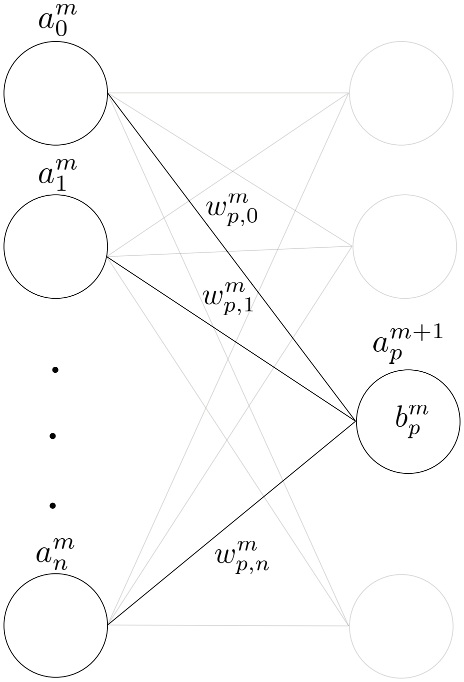


Figure 2.4: Neurons from layer m connecting to a single neuron p in layer $m+1$. The weights $w_{p,n}$ and bias b_p is also shown.

This equation is often referred to as the activation function, where $\boldsymbol{w}_p^{(m)}$ is the set of weights connecting node p in layer $m+1$ to each node in the layer m, and $b_p^{(m)}$ is the node bias for the same connection. The role of the weights is to establish the relationship between a node in one layer, and a node in the next layer. The bias is used to constrain the activation of a node to first meet the target b before activating. When training the network, all the weights and biases are considered model parameters, and are optimized using the optimization algorithm described in Section 2.3.2 on the optimization problem (2.16). As such, the set of weights and biases in the network can be expressed as:

$$\boldsymbol{\phi} = [\boldsymbol{W}, \boldsymbol{b}]. \tag{2.19}$$

In practice, the activation function (2.18) is often augmented in order to inhabit more favourable characteristics for training. One such augmentation is the use of rectified linear units (ReLU), which has the activation function

$$c(a) = max(0, a). \tag{2.20}$$

The output of such an activation function can be seen in Figure 2.5. By letting any negative activation value go to zero instead, the neurons can be "deactivated" by negative values, more closely resembling the behaviour of actual brain neurons.

.

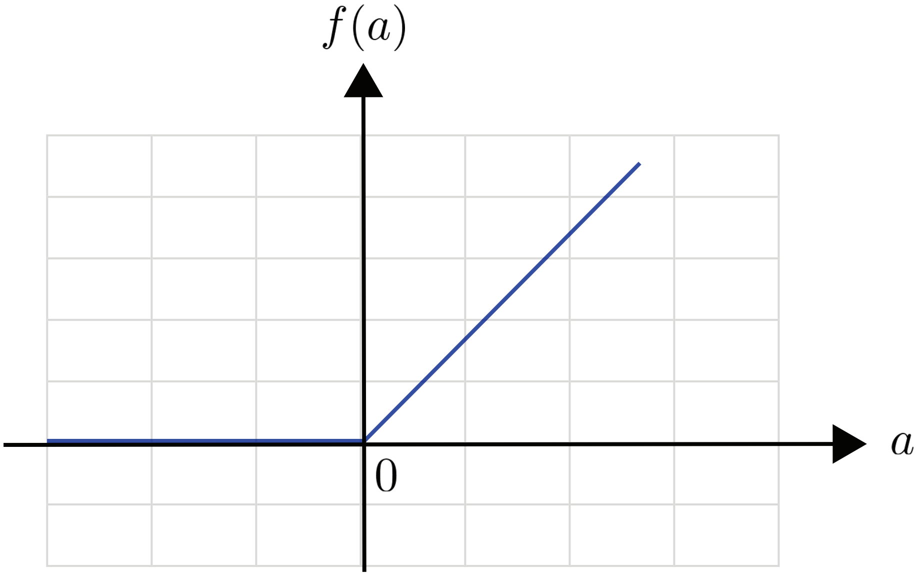


Figure 2.5: Plot showing the values of the ReLU activation function. Negative values are set to zero, while positive values are set to a.

2.3.4 Implementing a Neural Network

One of the strengths of neural networks is their versatility. One can utilize a neural network to solve many different kinds of tasks such as speech recognition, classification, or as in the case of this project, regression. Finding a network configuration appropriate to the task is essential for achieving good performance. Following will be some considerations to make when searching for an optimal network configuration.

Splitting up the dataset: It is common to split the available datapoints into three sets: the training set, the validation set, and the test set. The main bulk of the data should go into the training set, which is the collection of datapoints made available for use in the optimization problem. The training set should contain enough datapoints such that the model is trained on scenarios the system is likely to be exposed to (van Smeden et al., 2019). However, the data in the dataset must also be accurate in regard to the real system (Sessions and Valtorta, 2006). Quality and quantity are both important in this aspect. The validation set is a batch of datapoints used for evaluating the model prediction performance during training. It is common to use the result of the model on the validation set to tune hyperparameters such as regularization factors. When the model is trained and the hyperparameters set, a measure of the true model predictive performance can be gauged by using the test set. In this regard, the test set emulates how the model will generalize to new, unseen data.

Initializing the parameters: As mentioned in Section 2.3.3, every node in the network has a weight and a bias. Summarized over all the nodes in the network we get the set of weights and biases, also called the neural network parameters, as presented in Equation (2.19). The final values of these parameters are found during training, but their initial value is a design consideration. If the bias is set too high, or the weight is too large, the network might not be able to converge properly during training. As such, one should consider the system output and likely final values of the neural network parameters when initializing them. A popular way of doing this for networks using ReLU activation functions, is Kaiming initialization (He et al., 2015). The biases are then initialized at zero, while the weights are initialized to a value along a zero-centered normal distribution.

Scaling the inputs: Sometimes the input to the neural network is not binary, but rather a numeric value representing a physical property. This is normally the case for regression tasks. As mentioned above, when using certain initialization schemes such as Kaiming, the weights and biases are initialized around zero. If we feed input with large numeric values into the network, this might necessitate large modifications of the network parameters away from their initial value. Therefore, by scaling the input around the same values as the weights and biases, one can minimize large parameter changes. One way of doing this is using a min-max scaler, which scales the input between zero and one based on the minimum and maximum value present in the dataset. Most machine learning frameworks offer such functionality. The output of the machine learning model can then be multiplied by the inverse scaler to acquire the true regression output.

Regularization: The goal of a machine learning algorithm is to generalize. That is, after being trained on a specific set of data in the training set, the algorithm is able to also predict the output of datapoints it has not yet been exposed to. In order to have the model generalize well, one might impede the optimization of the model on the training data, as to avoid overfitting. Overfitting occurs when a neural network model is optimized on the training data in such a way that input not present in the training data result in poor prediction. We impede the training of the model through regularization. Goodfellow et al. (2016) defines regularization as: "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error". An example of this is the L2-regularization coefficient discussed in Section 2.3.1. The idea is that for overfitted models, the weights and biases of the model may have drifted far from their mean value. By putting a cost on the deviation to the mean value, we can prevent the parameters from exploding. The L2-regularization coefficient is a hyperparameter that has to be set by the developer implementing the model.

Early Stopping: Another way of preventing overfitting is through the use of early stopping. Early stopping is a regularization technique in which the model is evaluated on the validation set during training. When the prediction performance on the validation set stagnates, the training loop is cut short. The main benefit of using this regularization technique is that the validation set tend to plateau faster than the training set, and thus stops the training loop at the moment the model stops improving on data not present in the training set. This can be seen from Figure 2.6, where the mean squared error (MSE) loss for both the training- and validation set is plotted against number of training epochs. This model has been cut short by early stopping, as one can observe that the MSE for the training set is still declining. The metric for how long to wait after the validation performance stagnates to cut the training loop is often referred to as patience, and is a hyperparameter that has to be set during implementation of the model.

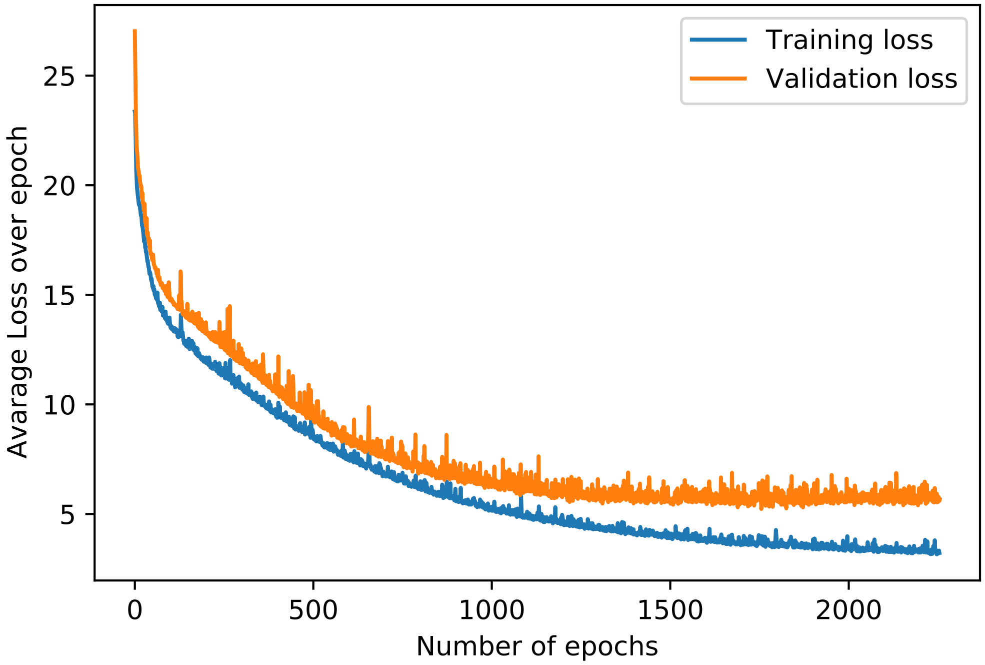


Figure 2.6: Mean squared error (MSE) loss for the prediction using the validation set and the training set during a training loop.

2.4 Hybrid System Models

Hybrid modeling is a modeling technique which combines classical first-principle mathematical modeling with data-driven modeling techniques. Hotvedt et al. (2021) presents hybrid models on a grey scale from black to white, where black models represent data-driven "black box" models, while white models are transparent, first-principles models. Figure 2.7 illustrates this. A key advantage of using hybrid models, sometimes referred to as "gray-box" models, is that they can keep some of the transparency while still exploiting the flexibility of data-driven components. In other words, the hybrid model is to a degree *interpretable* (Pitarch et al., 2019). By this we mean that the internals of the model, for instance relationships between variables, are known prior to model training.

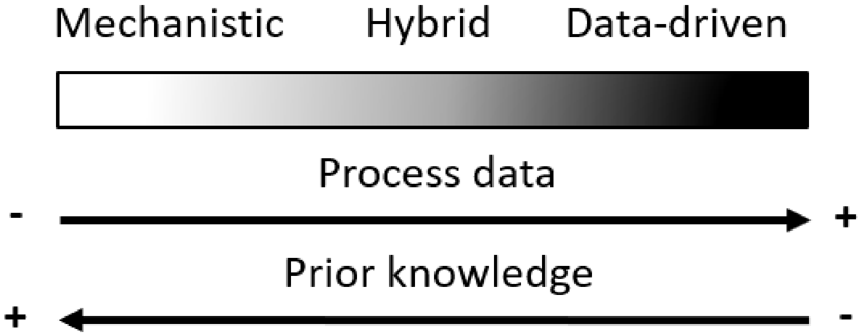


Figure 2.7: Greyscale of different hybrid model configurations. Illustration taken from Hotvedt et al. (2020a) with permission.

The architecture of the hybrid model plays a large role in where the model fits on the greyscale. There are two main architectural types of hybrid models: serial and parallel (Hotvedt et al., 2020a). In serial hybrid model structures, the two model components are dependent on each other. For instance, a data-driven model may be used to estimate a parameter to be used in the mechanistic model. Alternatively, the inputs to a data-driven model may be manipulated using first-principle equations. An example of a parallel hybrid model would be to combine the outputs of both a data-driven model and a first-principles model in order to create an ensemble output. The two model components are in this case not directly dependent of each other, as in the case of serial model configurations. Following will be some examples of the aforementioned configurations in the context of a wellbore.

One way of implementing a serial hybrid model would be to calculate an input to a neural network using first-principles. For instance, by using the real gas law (2.8) one could find the gaseous density ρ_G to supplement other measured inputs. Figure 2.8a illustrates such a configuration. Hotvedt et al. (2021) defines this type of serial model as a "black-to-gray" hybrid model. Calculating the input to a neural network like this is also often referred to as feature engineering, and is an ongoing field of research (Bikmukhametov and Jäschke, 2020a). A "white-to-gray" serial hybrid model would work in the opposite way, where the output of a neural network could estimate a parameter used in the first-principles model. A suitable candidate could be the friction factor f. This hybrid model configuration is depicted in Figure 2.8b. Finally, a parallel model could be achieved by weighting the predicted mass flow output of a neural network and a first-principles model, as shown in Figure 2.8c.

In this project, a white-to-gray serial model will be used to predict the mass flow rate of the wellbore. This configuration is chosen due to it being more towards the white part of the spectrum of hybrid models. By feeding the output of the neural network to a first-principles model, we can still see all the values and relations used to calculate the output, making the model interpretable. The only part concealed by the black box component is the calculation of the chosen data-driven parameter.

2.4.1 Learnable Parameters

A key component of the hybrid model are the learnable parameters. In this project, we define a learnable parameter as:

Definition 2 (Learnable parameter)**.** A learnable parameter is a system parameter that is changeable during training of the system model.

In the case of a wellbore, the system parameters are the well parameters that define the individual wellbore. Examples of this are the well height z, pipe diameter D, and oil density ρ_O. A learnable parameter can be any one of these. As such, the set of learnable parameters is a subset of the set of available well parameters:

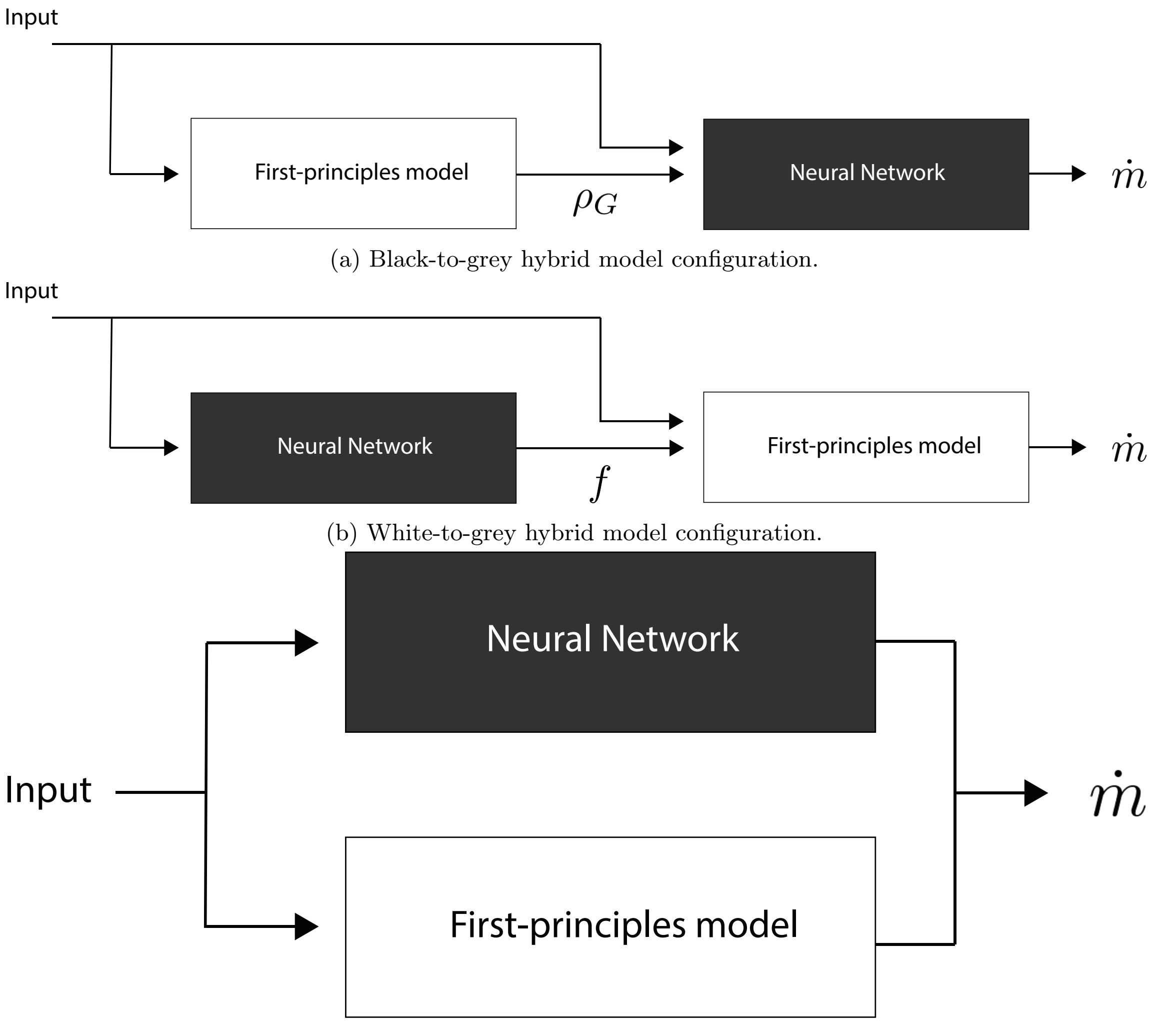
$$\boldsymbol{\phi}_{LP} \subseteq \boldsymbol{\phi}_{WP}.$$

The main advantage of using learnable parameters is that they can contribute to handling uncertainty in the system parameters. For instance, one might not know the exact value of the oil density ρ_O for a specific field, but can be reasonably certain that it falls within 800 to 1000 kg$/m^2$ (Speight, 1999). By setting the oil density as a learnable parameter, the numerical value is allowed to fluctuate during training of the system model, such that hopefully a representative value is found. This parameter estimation is fully transparent, allowing the engineer to assess the quality of the estimation by making sure the parameter does not converge outside of physical boundaries.

As one might have surmised, the estimation of the learnable parameters is done in the same fashion as with the neural network parameters discussed in Section 2.3.3. As such, it is important to distinguish between the two definitions; although network parameters are "learnable" in the sense that they are changed during training of the network, they do not carry any physical meaning and are not related to the system parameters. By using MAP estimation as defined in Equation (2.16), the set of learnable parameters can be appended to the set of estimated parameters such that:

$$\boldsymbol{\phi} = [\boldsymbol{\phi}_{NN}, \boldsymbol{\phi}_{LP}], \tag{2.21}$$

where $\boldsymbol{\phi}_{LP}$ is the set of learnable parameters and $\boldsymbol{\phi}_{NN}$ is the set of neural network parameters defined in Equation (2.19).



(a) Black-to-grey hybrid model configuration.

(b) White-to-grey hybrid model configuration.

(c) parallel hybrid model configuration.

Figure 2.8: Different hybrid model configurations.

Chapter 3

Models

This chapter will concern itself with the construction of the mass flow rate models. Section 3.1 will outline a first-principle mechanistic mass flow rate model. Next, a fully data-driven mass flow rate model will be developed in Section 3.2. Finally, the results from these two sections will be used to develop a hybrid model for the wellbore mass flow rate in Section 3.3.

3.1 Constructing a First-Principle Mass Flow Model

In this section, the first-principles necessary to construct a mechanistic mass flow rate model for the wellbore will be presented. It should be noted that a multitude of different physical phenomena influence the thermodynamic- and fluid dynamic properties of the mass flow, and modeling all these factors is not the objective of this thesis. Instead, a simplified mechanistic model will be presented. The goal of this model is not necessarily to achieve the highest prediction accuracy possible, but rather to be easy to implement and compute. This first-principles model will also serve as a baseline for the hybrid model. Readers should note that the following mechanistic model is the same as the one implemented in the precursor thesis Fredriksen (2021), allowing for experience and results to carry over to this project.

The model for the mass flow rate will be developed from the steady-state Bernoulli energy equation along a streamline presented in Section 2.2.1 (Cengel and Cimbala, 2014). Using the subscripts $i = \{1, 2\}$ to denote bottomhole and wellhead, respectively, with pressures p_i, densities ρ_i, fluid velocities v_i, and heights z_i the Bernoulli equation states:

$$\frac{p_1}{g} + \frac{v_1^2 \rho_1}{2g} + z_1 \rho_1 = \frac{p_2}{g} + \frac{v_2^2 \rho_2}{2g} + z_2 \rho_2 + \rho_2 h_f. \tag{3.1}$$

The head loss h_f is added to the Bernoulli equation as discussed in Section 2.2.2.

Substituting the head loss expression (2.9) into the Bernoulli equation (3.1) will, after some rearrangement, yield:

$$p_1 + \frac{v_1^2 \rho_1}{2} + z_1 \rho_1 g = p_2 + \frac{v_2^2 \rho_2}{2} + z_2 \rho_2 g + f \frac{L \rho_2 v^2}{2D}. \tag{3.2}$$

In order to write Equation (3.2) as a function of the desired mass flow rate $\dot{m}$, the conservation of mass from the bottomhole to the wellhead is introduced (Dixon and Hall, 2010):

$$\dot{m} = \rho_1 v_1 A = \rho_2 v_2 A. \tag{3.3}$$

Here, the cross-sectional area A is assumed constant throughout the pipe. By substituting Equation (3.3) into Equation (3.2) and rearranging, an equation for the mass flow rate through the wellbore is obtained:

$$p_1 + \frac{\dot{m}^2}{2\rho_1 A^2} + z_1 \rho_1 g = p_2 + \frac{\dot{m}^2}{2\rho_2 A^2} + z_2 \rho_2 g + f \frac{L}{D} \frac{\dot{m}^2}{2\rho_2 A^2}, \tag{3.4}$$

$$\dot{m} = \sqrt{\frac{2\rho_1 \rho_2 A^2 (p_1 - p_2 + z_1 \rho_1 g - z_2 \rho_2 g)}{D\rho_1 + fL\rho_1 - D\rho_2}}. \tag{3.5}$$

Equation (3.5) is the completed first-principle mass flow rate model.

Typically, for a well-instrumented production well the available measurements are bottomhole- and wellhead pressures and temperatures. In the wellbore system we will also assume the fluid mass fractions $\boldsymbol{\eta} = \{\eta_G, \eta_O\}$ are known, and they will be calculated using measurements from a multiphase flow meter. Furthermore, the well heights z_i and the pipe diameter D are also assumed known. We will consider a completely vertical well, such that:

$$L = z_2 - z_1, \tag{3.6}$$

as depicted in Figure 2.2. Setting the bottomhole height to zero, we can simplify the model as follows:

$$L = z_2 = h. \tag{3.7}$$

The remaining friction factor f and the fluid densities ρ_i can be calculated using the methods presented in Section 2.2.2. The inclusion of the friction factor f in the mass flow model (3.5) also necessitates the calculation of the Reynolds number. The Reynolds number can be calculated using Equation (2.10), where the mass flow rate $\dot{m}$ is also included. Consequently, the wellbore mass flow rate model (3.5) is an implicit mathematical model when f is calculated in this way. Solving implicit mechanistic models like this is computationally heavy, so for more light-weight implementations of this model one might set the friction factor f to an appropriate value instead, at the cost of model accuracy.

Having defined all the components of the first-principles (FP) mass flow rate model in Equation (3.5), the model can be restated as follows:

$$\dot{m} = g_{FP}(\mathbf{x}; \boldsymbol{\phi}_{WP}), \tag{3.8}$$

where,

$$\mathbf{x} = [p_{BH}, p_{WH}, T_{BH}, T_{WH}, Z_{BH}, Z_{WH}, \eta_O, \eta_G, \dot{m}], \tag{3.9}$$

is the set of system inputs, and

$$\boldsymbol{\phi}_{WP} = [D, L, \rho_O, \rho_W, \eta_O, \eta_G, M_g, \mu], \tag{3.10}$$

is the set of chosen well parameters. The well parameters are unique to each well, and there is often a significant amount of uncertainty associated with their numerical value. One method of approaching this uncertainty is by using data-driven modeling techniques, as was discussed in Section 2.4.1.

3.2 Constructing a Data-Driven Model

This section will outline the creation of a fully data-driven model for the multiphase mass flow rate through the wellbore. This model will serve as a comparative benchmark for the hybrid model. The central component of the model will be a neural network trained on the dataset containing real-world datapoints from a producing oil field. Using this data, the data-driven model will aim to capture the relationship between temperatures and pressures, and the outgoing mass flow.

One of the defining characteristics of this type of model is its lack of well parameters, such as well length and cross-sectional area. The model only takes the inputs given, and produces an output. Thus, the data-driven model g_{DD} can be considered a "black-box" model. There are several ways to construct the function g_{DD}, and this is an ongoing topic of research (AL-Qutami et al., 2018; Andrianov, 2018; Kanin et al., 2019; Grimstad et al., 2021). In this project, g_{DD} will be comprised of a feed-forward neural network. This network was developed following the procedures outlined in Section 2.3.3. An overview of the network architecture is presented in Figure 3.1. The neural network is fully connected using ReLU activation functions with 6 inputs, 1 hidden layer with 100 nodes, and a single output layer. As discussed in Section 2.3.3, the network contains both weights $\mathbf{W}$ and biases $\mathbf{b}$, which are the network parameters. Summarizing the discussion, the data-driven model may be expressed as follows:

$$\dot{m} = g_{DD}(\mathbf{x}_s; \boldsymbol{\phi}_{DD}) \tag{3.11}$$

where g_{DD} is the neural network,

$$\mathbf{x}_s = [T_{BH,s}, T_{WH,s}, p_{BH,s}, p_{WH,s}, Z_{BH}, Z_{WH}] \tag{3.12}$$

is the scaled set of system inputs, and

$$\boldsymbol{\phi}_{DD} = [\mathbf{W}_{DD}, \mathbf{b}_{DD}] \tag{3.13}$$

are the neural network parameters. The concept of scaling the inputs of the neural network was discussed in Section 2.3.4.

With the data-driven model defined in Equation (3.11), we can implement the cost function in accordance with the process outlined in Section 2.3.1. This will constitute the optimization problem to be solved during training of this model. Using L2-regularization, we can replace the factor $\frac{\sigma_\epsilon^2}{\sigma_i^2}$ from Equation (2.16) with the L2-regularization factor λ_{DD} in order obtain the following optimization problem (Goodfellow et al., 2016):

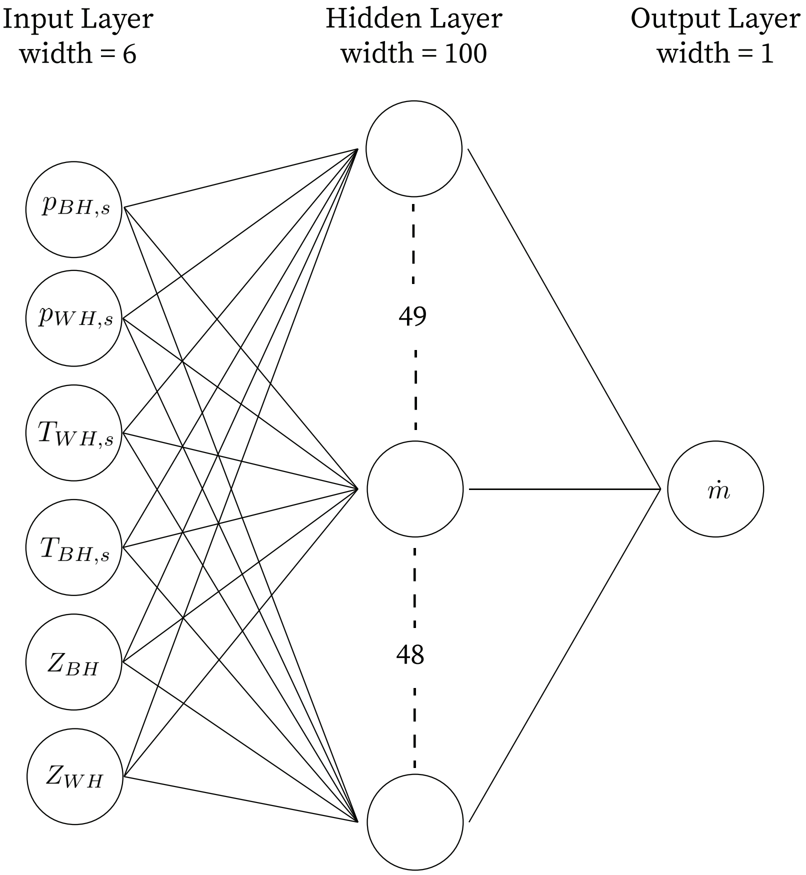


Figure 3.1: Neural network architecture for the data-driven mass flow rate model

$$
\begin{aligned}
\hat{\boldsymbol{\phi}}_{DD} &= \arg\min_{\boldsymbol{\phi}_{DD}} J\left(\boldsymbol{\phi}_{DD}, \lambda\right) \\
&= \arg\min_{\boldsymbol{\phi}_{DD}} \left(\frac{1}{n} \sum_{i=1}^{n} \left(\dot{m}_i - g_{DD}(\mathbf{x}_{s,i}, \boldsymbol{\phi}_{DD}) \right)^2 + \frac{1}{n} \sum_{j=1}^{m_{DD}} \lambda_{DD} \left(\phi_{DD,j} - \bar{\phi}_{DD,j} \right)^2 \right)
\end{aligned} \tag{3.14}
$$

In this optimization problem formulation n is the number of total data points, m_{DD} is the total number of neural network parameters, $\bar{\phi}_{DD,j}$ is the mean value of the neural network parameter $\phi_{DD,j}$, and $\hat{\boldsymbol{\phi}}_{DD}$ is the estimated set of neural network parameters $\boldsymbol{\phi}_{DD}$ that minimizes the objective function. The mean value of the neural network parameters $\bar{\boldsymbol{\phi}}_{DD}$ are set to zero in order to avoid large values, as discussed in Section 2.3.4.

3.3 Constructing the Hybrid Model

Having developed the mechanistic model in Section 3.1 and the data-driven model in Section 3.2, the groundwork has been laid for the construction of a hybrid wellbore mass flow rate model. This model will aim to combine the two previous approaches in order to leverage their respective advantages. A hybrid model can be constructed in a multitude of configurations (Bikmukhametov and Jäschke, 2020a). For this project application a serial hybrid model as depicted in Figure 2.8b will be developed. Using a similar neural network as in Section 3.2, the friction factor f from Equation (2.11) will be estimated. Although there are many viable candidates for the choice of estimated variable, the friction factor f is chosen due to its complex calculation using numerical methods. This model configuration essentially converts the friction factor to an input to the mechanistic model, thus creating a serial hybrid model. This can be seen from the model architecture depicted in Figure 3.2. An important distinction to make is that the value of f in this model configuration is updated in real-time when new well data becomes available, and is not kept constant. Another advantage with estimating the friction factor f with a neural network is that there is no longer any need for calculating the Reynolds number in Equation (2.10). As such, the well parameter μ from Equation (3.10) is no longer necessary. Furthermore, the mass flow rate $\dot{m}$ will not be required as input to the model, see Equation (3.9). In turn, this makes the mechanistic model in Equation (3.5) explicit, and computational complexity when solving the model numerically has been greatly reduced.

The neural network used to estimate the friction factor f may be expressed mathematically as follows:

$$
f = g_{NN}(\mathbf{x}_s; \boldsymbol{\phi}_{NN}), \tag{3.15}
$$

where $\mathbf{x}_s$ is the scaled set of system inputs defined in Equation (3.12), and:

$$
\boldsymbol{\phi}_{NN} = [\mathbf{W}_{NN}, \mathbf{b}_{NN}], \tag{3.16}
$$

are the weights and biases for the neural network used to estimate f. The neural network structure is the same as for the data-driven model defined in Section 3.2. The key difference is that the data-driven model in Section 3.2 is directly trained on the output $\dot{m}$, while the friction factor model in Equation (3.15) is embedded in the hybrid model. Consequently, the friction factor is indirectly

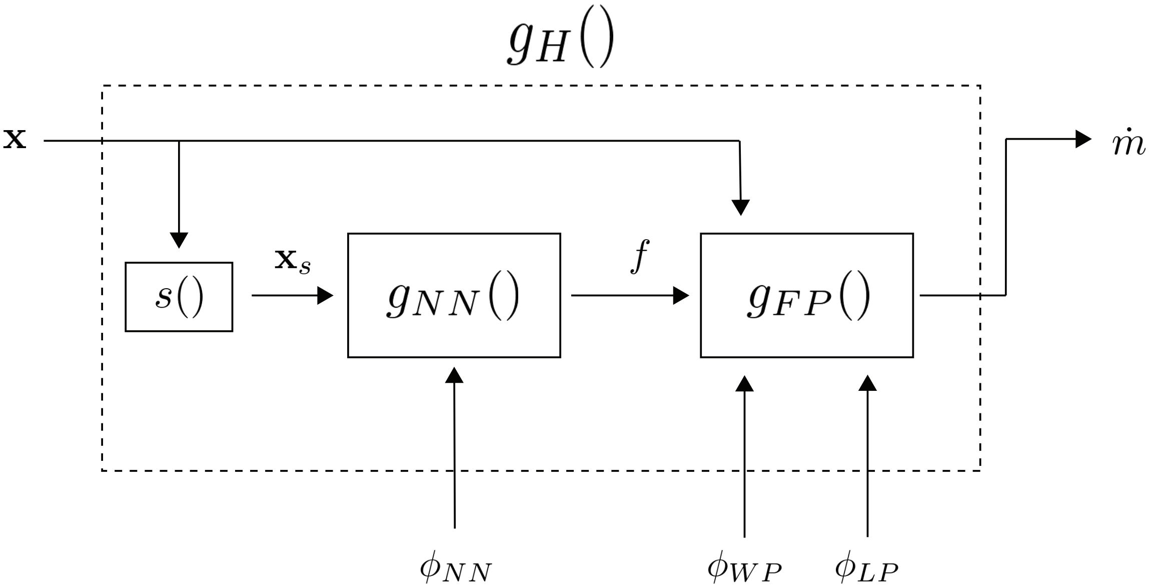


Figure 3.2: Model architecture of the hybrid mass flow rate model. The system input $\boldsymbol{x}$ goes to both a neural network component $g_{NN}()$ and a first-principle model $g_{FP}()$. Before entering the neural network, the input is scaled using a min-max scaler $s()$. The input to the neural network is then the scaled set of inputs $\boldsymbol{x}_s$, while its output is the estimated friction factor f. The set of learnable parameters $\boldsymbol{\phi}_{LP}$, the set of well parameters $\boldsymbol{\phi}_{WP}$, and the set of neural network parameters $\boldsymbol{\phi}_{NN}$ are also depicted with arrows to indicate which model component they belong to. The output of the hybrid model g_H is the mass flow rate $\dot{m}$.

trained on the system output $\dot{m}$. Using this model configuration avoids the need for friction factor measurements to train the model. This is advantageous seeing as such measurements are typically not readily available.

Combining the neural-network implementation of the friction factor f in Equation (3.15) with the first-principles mass flow rate model in Equation (3.8), we can define the following hybrid mass flow rate model:

$$\dot{m} = g_H(\mathbf{x}; \boldsymbol{\phi}_{WP}, \boldsymbol{\phi}_{LP}, \boldsymbol{\phi}_{NN}). \tag{3.17}$$

Here,

$$\mathbf{x} = [p_{BH}, p_{WH}, T_{BH}, T_{WH}, Z_{BH}, Z_{WH}, \eta_O, \eta_G] \tag{3.18}$$

is the set of model inputs, $\boldsymbol{\phi}_{NN}$ is the set of neural network parameters as defined in Equation (3.16), $\boldsymbol{\phi}_{WP}$ is the set of well parameters, and $\boldsymbol{\phi}_{LP}$ is the set of learnable parameters. The relation between learnable- and well parameters was covered in Section 2.4.1. What parameters to include in the set of well parameters, and which to include as learnable parameters is an implementation choice to be made by the developer. Two important aspects regarding the system inputs $\mathbf{x}$ should be noted. First, in order for the system inputs in Equation (3.18) to fit the neural network inputs in Equation (3.15), they have to be scaled using a scaling function function:

$$\mathbf{x}_s = s(\mathbf{x}). \tag{3.19}$$

This can also be observed in the model architecture overview in Figure 3.2. Secondly, comparing the system inputs for the hybrid model in Equation (3.18) with the system inputs of the first principles model in Equation (3.9), the mass flow rate $\dot{m}$ has been removed, as was discussed earlier.

Having defined the hybrid model and its inputs, the estimation problem can be constructed using MAP estimation as outlined in Section 2.3.1. Seeing as the values of the learnable parameters are estimated using the training data as discussed in Section 2.4.1, the set $\boldsymbol{\phi}_{LP}$ should also be included in the estimation problem definition. As such, we can define the set of parameters as:

$$\begin{aligned}\boldsymbol{\phi} &= [\boldsymbol{\phi}_{WP}, \boldsymbol{\phi}^*] \\ \boldsymbol{\phi}^* &= [\boldsymbol{\phi}_{LP}, \boldsymbol{\phi}_{NN}],\end{aligned}$$

where $\boldsymbol{\phi}^*$ is the set of parameters to be estimated. The optimization problem can now be expressed as:

$$\begin{aligned}\hat{\boldsymbol{\phi}}^* &= \arg\min_{\boldsymbol{\phi}^*} J\left(\boldsymbol{\phi}^*, \boldsymbol{\lambda}^*\right) \\ &= \arg\min_{\boldsymbol{\phi}^*} \left(\frac{1}{n} \sum_{i=1}^{n} \left(\dot{m}_i - g_H(\mathbf{X}_i; \boldsymbol{\phi})\right)^2 + \frac{1}{n} \sum_{j=1}^{m_{NN}} \lambda_{NN} \left(\phi_{NN,j} - \bar{\phi}_{NN,j}\right)^2 \right. \\ &\qquad \left. + \frac{1}{n} \sum_{k=1}^{m_{LP}} \frac{\sigma_\epsilon^2}{\sigma_{i,k}^2} \left(\phi_{LP,k} - \bar{\phi}_{LP,k}\right)^2 \right).\end{aligned} \tag{3.20}$$

Here, m_{NN} is the number of neural network parameters, m_{LP} is the number of learnable parameters, and λ_{NN} is the L2-regularization parameter for the neural network parameters. Notice also that the we only optimize on the data-driven parameters $\boldsymbol{\phi}^*$, seeing as the well parameters $\boldsymbol{\phi}_{WP}$ are assumed constant for each individual well. The variables λ_{NN}, m_{NN}, and m_{LP} are all hyperparameters that must be decided upon during implementation of the hybrid model.

Chapter 4

Dataset

Regardless of system model architecture, the accuracy of the measurements govern the accuracy of the resulting estimation output. This is especially true for system models which utilize data-driven modeling techniques, such as hybrid models. It is therefore important to examine the available measurements. In the case study of this project, a dataset consisting of historical production data will be used to train the hybrid model. In Section 4.1 the structure of this dataset will be examined, and its procurement discussed. Section 4.2 will discuss necessary augmentations applied to the base dataset, and the justifications behind their inclusion. The chapter will conclude with a discussion of correlations between data in the dataset in Section 4.3.

4.1 Dataset Procurement and Structure

In contrast with the precursor thesis Fredriksen (2021), the dataset in this project implementation will consist of real production data. The datapoints are procured from an anonymous field, and provided by Solution Seeker (Solution Seeker, 2021). Before the datapoints are added to the set, a suite of preprocessing steps are executed to synchronize, clean, and compress the data into a steady-state point structure suitable for steady-state modeling. This avoids the need of measurements of the friction factor to train the model (Grimstad et al., 2016). As such, the dataset will be suitable for the steady-state first-principles model outlined in Section 3.1. The data consists of the following variables:

- Bottomhole- and wellhead pressures p_{BH} and p_{WH}.
- Bottomhole- and wellhead temperatures T_{BH} and T_{WH}.
- Bottomhole- and wellhead compressibility factors Z_{BH} and Z_{WH}.
- Oil- and gas mass fractions η_O and η_G.

- Mass flow rate $\dot{m}$.

It should be noted that the compressibility factors Z_i are calculated from the compressed data and then added to the dataset. The available measurements are the inputs required in the first-principle, data-driven, and hybrid models defined in Section 3.1, Section 3.2, and Section 3.3, respectively. The model output will be $\dot{m}$. We organize the dataset as follows:

$$\begin{aligned} &\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}, \\ &\mathbf{x}_i = [p_{BH,i}, p_{WH,i}, T_{BH,i}, T_{WH,i}, Z_{BH,i}, Z_{WH,i}, \eta_{O,i}, \eta_{G,i}], \\ &y_i = [\dot{m}_i], \end{aligned} \tag{4.1}$$

where i is a steady-state time instance and n is the number of datapoints in the dataset.

4.1.1 Compressibility Factor

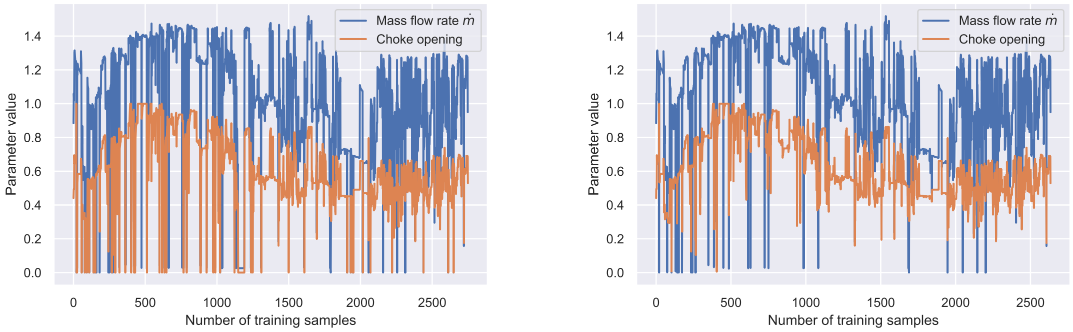
An addition made to the original dataset is the inclusion of the gas compressibility factor Z. The gas compressibilty factor is used to compensate for the deviation of a naturally occurring gas from the real gas law (LaNasa and Upp, 2014), and is used in the calculation of the gaseous fluid density in Equation (2.8). The value of the gas compressibility factor for a given pressure p and temperature T is calculated using the Hall & Yarborough equation fitted to the Standing-Katz chart as outlined in Sutton (1985). This value has then been added for both bottomhole- and wellhead conditions at every data entry, effectively making the compressibility factor Z a system input to the model. Another approach would be to calculate the Z-factor at every iteration, but having the value included in the dataset instead enables the use of this variable in training the neural networks in both the fully data-driven model in Section 3.2 and the hybrid model in Section 3.3.

4.2 Dataset Preprocessing

Before applying the dataset in training of the models, it has to be preprocessed in order to exclude erroneous measurements. Yet, during preprocessing, it is important to exclude only the erroneous measurements and not measurements capturing unexpected or interesting physical phenomena. As such, the dataset has been altered with a set of filters applied to all existing datapoints. Below follows a description of the filters.

4.2.1 Validation Tests

Sensor data from multiphase flow meters are not always sufficiently accurate, and a plethora of factors may lead to corrupted data samples. As such, the first preprocessing step applied to the dataset is that of various validation test to confirm data validity. Primarily, this involves removing entries with missing values from the dataset, as well as setting negative mass fractions to zero $\eta_i < 0 \rightarrow \eta_i = 0$. Another validity test that is implemented involves checking the pressure values at the bottomhole and wellhead, p_{BH} and p_{WH}, respectively. Entries where $p_{BH} < p_{WH}$ would



(a) Dataset before zero choke opening test.

(b) Dataset after zero choke opening test.

Figure 4.1: Overview of mass flow rate at different time instances in the dataset, before and after removing instances with zero choke value. The mass flow rate has been scaled, and does not represent actual mass flow rate. The choke opening is represented on a value from zero to one, where one represents full opening.

indicate negative direction fluid flow, which can safely be assumed to not represent the actual well conditions. As such, these entries are also removed.

4.2.2 Closed Choke Test

Looking at Figure 4.1a it is clear that the well is not always in operation at the time of data sampling, meaning zero mass flow rate $\dot{m}$. There may be several reasons for a well not operating, such as well maintenance. However, from the perspective of predicting the mass flow rate through the wellbore, data points in which the well is not in operation are anomalous and should therefore not be considered in the training of the model. That is, when the production choke is closed, there is no need to estimate the mass flow rate through it. Measurements of the choke opening was given as an additional variable in the original dataset for preprocessing purposes only. By removing datapoints with zero choke opening, the dataset will contain a higher percentage of datapoints in normal flow conditions. An updated dataset with the zero choke values removed can be seen in Figure 4.1b. As expected, the number of datapoints available for training has decreased. As discussed in Section 2.3.4, it is not necessarily advantageous to lose datapoints when training machine learning models. However, the increased quality of the dataset hopefully offsets this disadvantage.

4.3 Data Correlations

As the models developed in this case study are trained using the available data, it is of interest to analyze whether the data coincides with the expected physical behavior of the well. Therefore, this section will provide a short discussion on the correlation between the available measurements. In this analysis, the compressibility factor Z is not included, as it is the product the data already included in the dataset. It is important to emphasise that the following discussion will limit itself to the correlation between any two variables at a time, disregarding multi-variable correlations in the dataset.

A heatmap depicting the correlation between the variables in the dataset is provided in Figure 4.2. Key observations from this figure is summarized below.

1. First, notice the correlation between the bottomhole pressure P_{BH} and wellhead pressure P_{WH}. With a correlation of 0.97, it shows that any pressure change manifesting itself at the bottomhole is also carried to the wellhead in equal amounts of change, and vice-versa. It is believed that the head loss due to friction during the operational conditions the dataset was procured, was approximately constant. This further implies that the friction factor f does not exhibit significant nonlinear behaviour under said operational conditions either, making it easier to estimate.

2. The second interesting result can be observed in the complete negative correlation between the mass fractions of oil and gas, η_O and η_G, respectively. In practice, this means that there exists an equilibrium of hydrocarbons in the form of oil and gas, where any phase change from one component is fully reflected in the other. Furthermore, this also means it is safe to assume that there is little to no water in this stage of production.

3. The two first results were both expected results considering the physics of the wellbore system. However, some of the correlations present in the dataset are harder to explain in the same sense. For instance, looking at the correlation between bottomhole pressure p_{BH} and mass flow rate $\dot{m}$, one would assume that increased pressure would lead to a higher mass flow rate. Instead, we see an almost complete *negative* correlation between the two variables. One suggestion as to why this happens might be that during the course of sampling this well, the reservoir pressure decreases. This happens naturally as fluid from the reservoir is tapped through the wellbore. However, using the production choke, the produced fluid mass flow rate can be held constant. This phenomena would manifest itself in the data by a decrease in bottomhole pressure p_{BH} equaling a continued outgoing mass flow rate $\dot{m}$. In this case, an external factor beyond the wellbore system influence the data used to train the wellbore model; the production choke. This is an important consideration to make when evaluating the wellbore model performance.

Summarizing the results stated above, we see that although most correlations are representative of the physical wellbore system, there are also some that deviate from the expected physical behavior. If trained on this dataset alone, a data-driven model such as the one from Section 3.2 will attempt to mimic these partly erroneous correlations, possibly adapting to a nonphysical wellbore behavior. The hybrid model from Section 3.3 will also be trained on the dataset, yet leveraging the first-principles from the mechanistic model in Section 3.1 to hopefully counteract some of the nonphysical artifacts from the dataset.

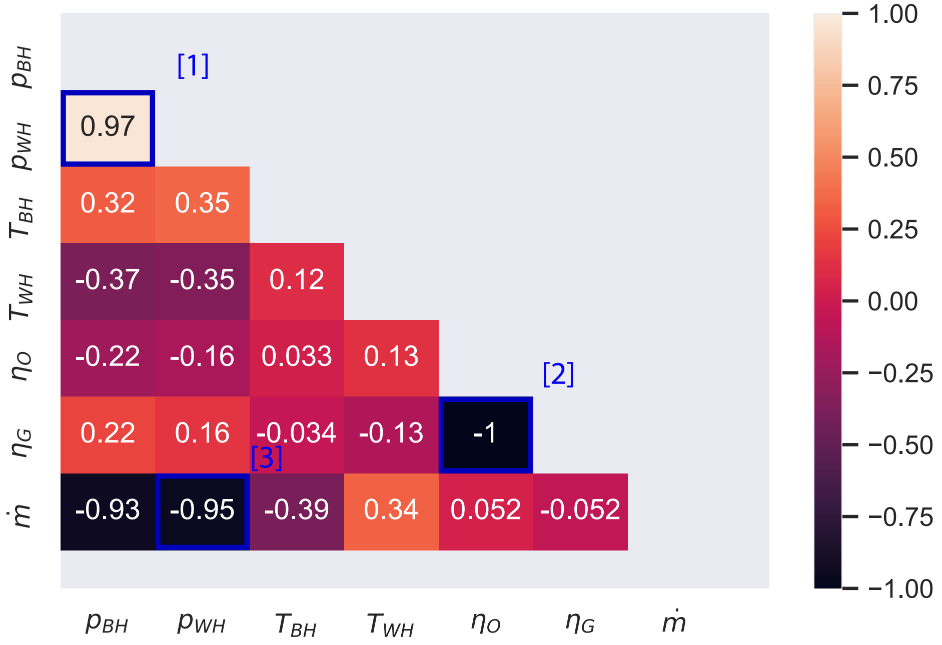


Figure 4.2: Heatmap depicting the correlations between the variables in the dataset. Notable results of interest is highlighted and enumerated in accordance with the discussion in Section 4.3

.

Chapter 5

Method

The previous chapters have laid the groundwork by developing the hybrid- and data-driven models. In Section 5.1, the experiments to be conducted to test the predictive performance of these models will be presented. Section 5.2 will discuss the remaining implementation details for the machine learning algorithms.

5.1 Experimental Setup

The experiment will consists of three parts. First, we will try to optimize the prediction performance of the hybrid model for a single well. Next, we will investigate the differences between the hybrid model and a purely data-driven model. Finally, we will apply the results of the two previous experiments to new wells.

5.1.1 Experiment: Optimizing the hybrid model for a single well

The first experiment will utilize all the tools outlined so far in this project to predict the mass flow from a single wellbore. This will be done using a hybrid model as presented in Section 3.3 with the dataset outlined in Equation (4.1). Using this setup, we will investigate the following:

- How does the choice of learnable parameters influence mass flow rate prediction accuracy from the model?
- How does parameter regularization of the learnable parameters impact the model?
- Is the achieved prediction accuracy satisfactory?

For this experiment, the potential learnable parameters to be considered are oil density ρ_O, water density ρ_W, well height z, and gaseous molar mass M_g. These parameters were chosen as there are uncertainties related to their prior values.

5.1.2 Experiment: Investigate the differences between the hybrid model and a data-driven model

The second experiment will investigate whether or not the use of a hybrid model, as opposed to a fully data-driven model, actually has the expected advantages of interpretability and mechanistic behaviour. To do this, we will use the data-driven model presented in Section 3.2 and the hybrid model in Section 3.3. We will then:

- Compare the prediction performance of the hybrid- and data-driven model.
- Perform a sensitivity analysis for both models.

The sensitivity analysis will be conducted by extracting a sample from the dataset, and then perturb the input variables around the datapoint. The corresponding output will then be analyzed accordingly. For instance, increasing the wellhead pressure p_{BH}, we would assume an increase in the mass flow rate $\dot{m}$ through the pipe.

5.1.3 Experiment: Checking results for new wells

The third and final experiment will aim to check the results for the two previous experiments with other wells in the same field. The dataset presented in Section 4 has 6 additional wells with the same structure to compare with. The same preprocessing steps used on the first well will also be applied to the rest of the wells. These steps were outlined in Section 4.2. By comparing with other wells, we hope to gain insight into general characteristics of the hybrid wellbore model, as opposed to results only reflected in a single well. Also, by looking at the performance metrics of several wells, we can investigate whether of not there is a large spread in performance from well to well.

5.2 Initializing the Machine Learning Algorithms

The final step of the preparation is to initialize the machine learning algorithms. This entails initializing the weights and biases of the neural networks, as well as deciding on the hyperparameters as discussed in Section 2.3.4. The following parameters are set in this implementation:

- The dataset for each well is split into training-, validation- and test sets with a ratio of 0.62-0.16-0.2, respectively.
- The neural network weights are initialized using Kaiming initialization (He et al., 2015).

- The learning rate is set to $l_{rate} = 0.0001$.
- The L2-regularization factors are set to $\lambda_{NN} = \lambda_{DD} = 0.0001$.
- The expected MAPE of the measuring instruments, as included in Equation (2.17), is set to $\alpha = 0.1$.
- The reference reading y_{ref} in Equation (2.17) is estimated using the mean value of y in the dataset for each well.
- Early stopping is implemented with a patience of 500 epochs.

Chapter 6

Results

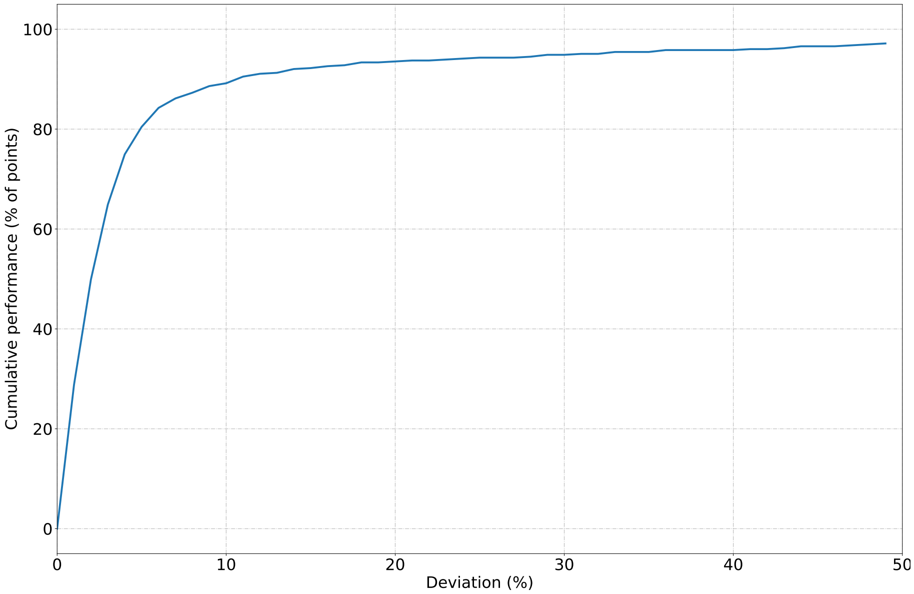
6.1 Result: Optimizing the hybrid model for a single well

In order to investigate the influence choice of learnable parameters has on the prediction accuracy, a cumulative performance plot for a hybrid model with, and without learnable parameters are presented in Figure 6.1. A cumulative performance plot illustrates the percentage of predictions that falls within a given deviation from the true value (Corneliussen et al., 2005b). The configuration with learnable parameters has all four activated. That is, oil density ρ_O, water density ρ_W, well height z, and gaseous molar mass M_g are all considered learnable parameters. For the configuration without learnable parameters, these variables are set to their prior mean value, and are thus constant. The model prediction accuracy is given by the performance metrics mean absolute error (MAE) and mean absolute percentage error (MAPE), seen in Table 6.1.

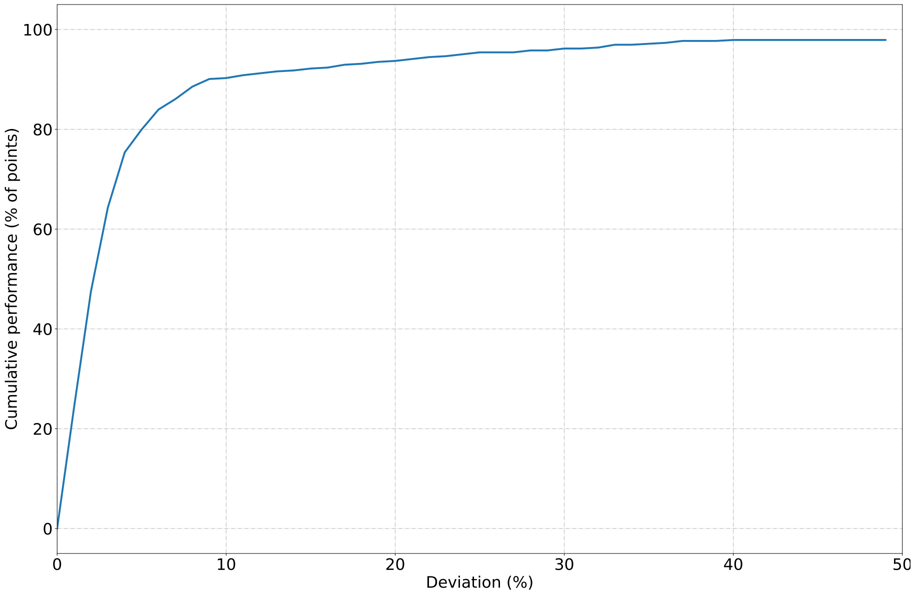
We can also use the cumulative performance plot to investigate the effect of parameter regularization. Figure 6.2 illustrates the cumulative performance plots for a hybrid model with- and without parameter regularization on the learnable parameters. Furthermore, Figure 6.3 shows the values of the learnable parameters during training for both these model configurations. The left side of the figure shows the parameters for the non-regularized model, while the right side shows the corresponding values for the regularized model configuration. The prediction accuracy of the two models are summarized in Table 6.1. The table also includes the final vaule of the the learnable parameters at the end of the training cycle.

Table 6.1: Table showing the estimation performance of various model configurations. The performance metrics are: mean absolute error (MAE) of the mass flow rate $\dot{m}$ and mean absolute percentage error (MAPE) of the mass flow rate $\dot{m}$. Also included is the final value of the learnable parameters oil density ρ_O, water density ρ_W, gaseous molar mass M_g, and well height h.
The model configurations are: hybrid model with all learnable parameters, hybrid model with all learnable parameters without parameter regularization, hybrid model with no learnable parameters, and the fully data-driven model.

Model configuration	MAE $\dot{m}$	MAPE $\dot{m}$	ρ_O	ρ_W	M_g	h
No learn. par.	1.400	4.182%	-	-	-	-
All learn. par.	1.268	4.921%	901.5	1029	3.550e-2	2323
All learn. par. no reg.	1.177	5.174%	872.4	-947.9	4.954e-2	2036
data-driven	1.095	3.701%	-	-	-	-

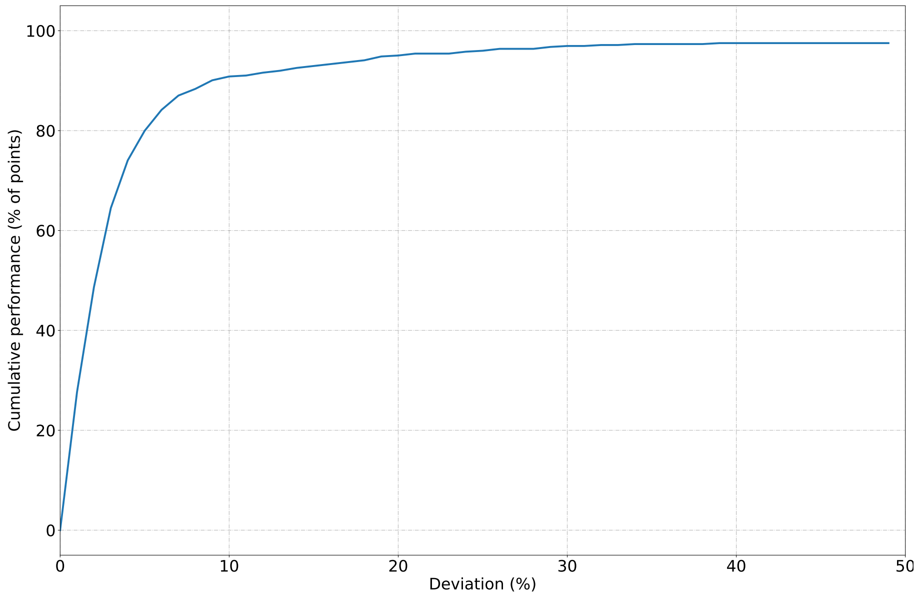


(a) CPP for well 1 with no learnable parameters.

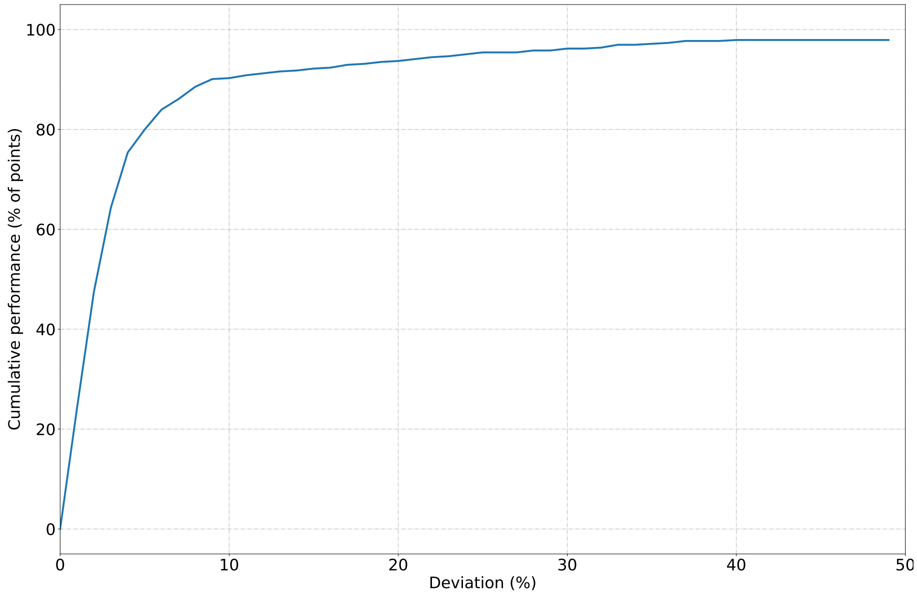


(b) CPP for well 1 with 4 learnable parameters.

Figure 6.1: Comparison of the cumulative performance plots for well 1 for model configurations with- and without learnable parameters.



(a) CPP for well 1 with no parameter regularization on the learnable parameters.



(b) CPP for well 1 with parameter regularization on the learnable parameters.

Figure 6.2: Comparison of the cumulative performance plots for well 1 for regularized- and non-regularized model configurations.

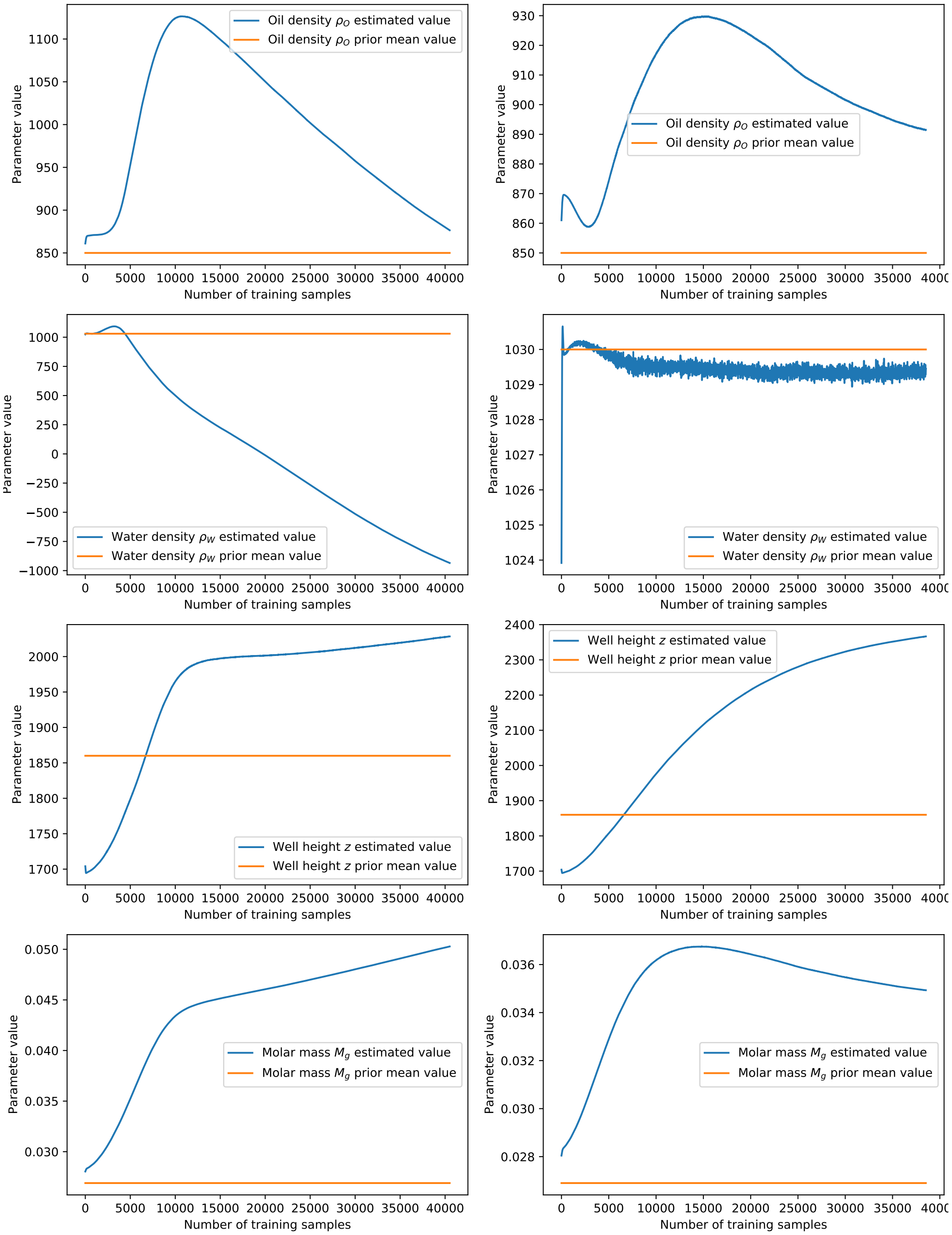
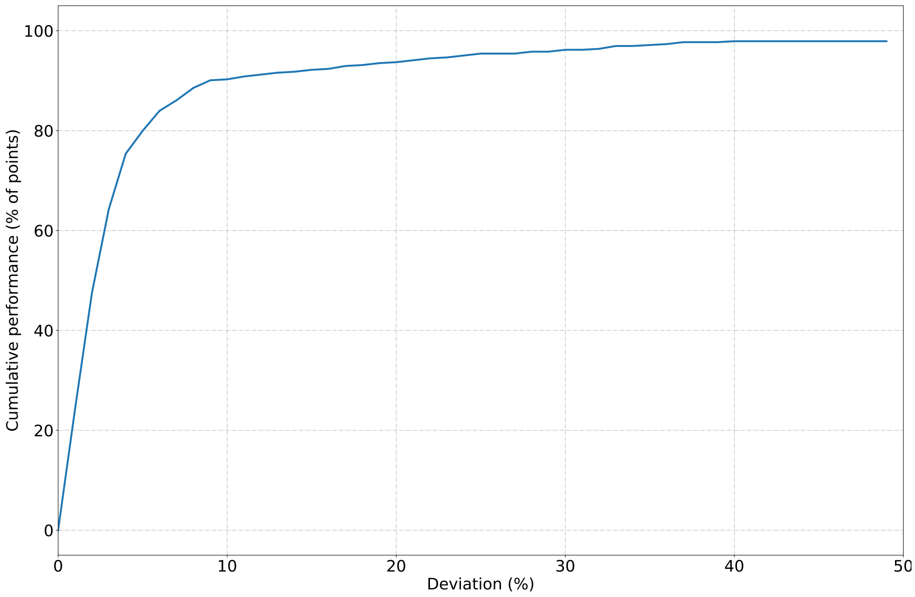


Figure 6.3: Learnable parameter values during training. *Left*: without parameter regularization. *Right*: with parameter regularization.

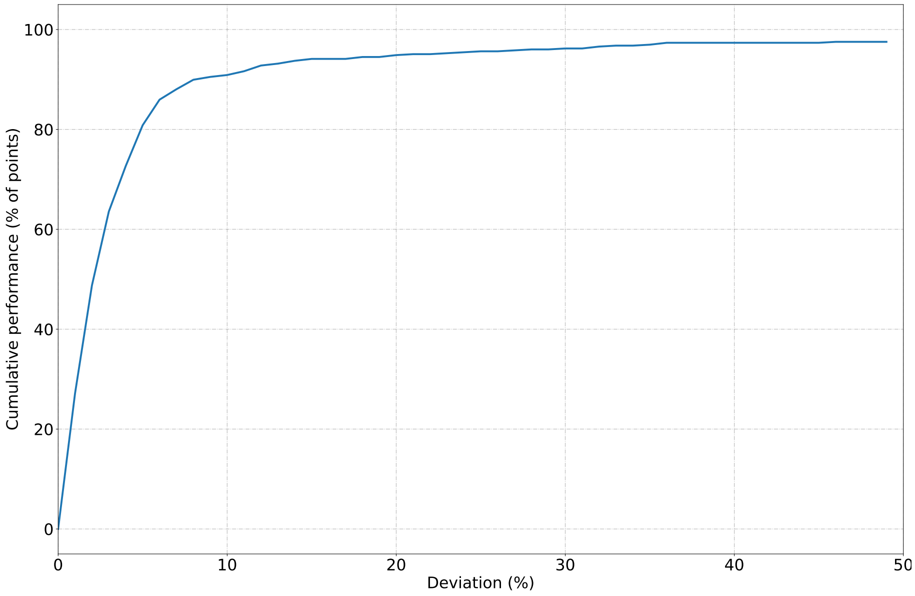
6.2 Result: Investigate the differences between the hybrid model and a data-driven model

For the second experiment, results comparing the performance of the hybrid model with the data-driven model will be presented. First, the cumulative performance plots for the two models can be seen in figure Figure 6.4. The hybrid model is configured with all 4 learnable parameters, using parameter regularization. Furthermore, a scatter plot of the model output $\dot{m}$ for both the hybrid model and the data-driven model is provided in the in Figure 6.5. Here, the original true value of $\dot{m}$ from the dataset is included for comparison. Note that the mass flow rate has been scaled, and does not reflect actual flow rates in the wellbore.

The second task outlined in the experiment description in Section 5.1.2 was to perform a sensitivity analysis on the two models. The result can be seen in Figure 6.6. This plot illustrates the change in model output for both the hybrid and data-driven model as bottomhole pressure p_{BH} is increased. Also included is the value of the estimated friction coefficient f from the hybrid model over the same pressure range.



(a) CPP for well 1 using the hybrid model.



(b) CPP for well 1 using the fully data-driven model.

Figure 6.4: Comparison of the cumulative performance plots for well 1 when using the hybrid model and the fully data-driven model.

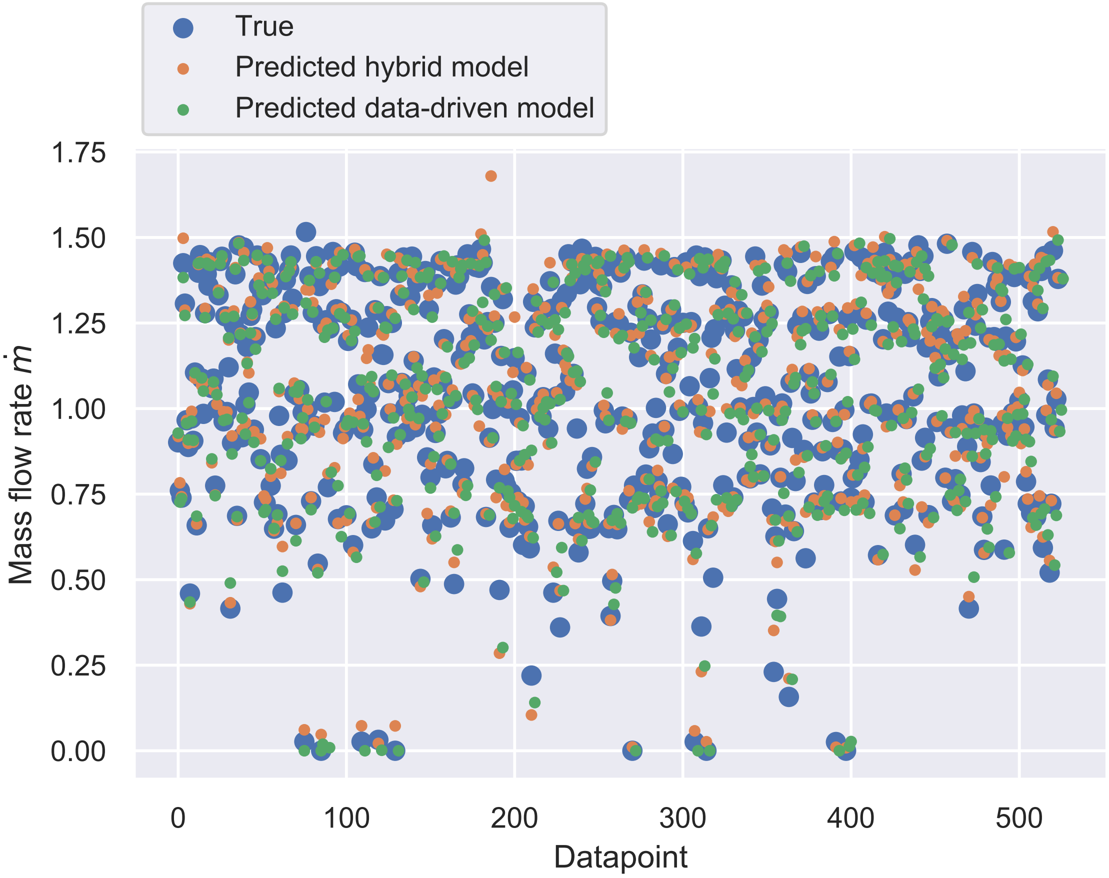


Figure 6.5: Scatter plot depicting the output of the hybrid model and data-driven model. The scatter plot shows every predicted- and measured mass flow rate pair in the test set for both models.

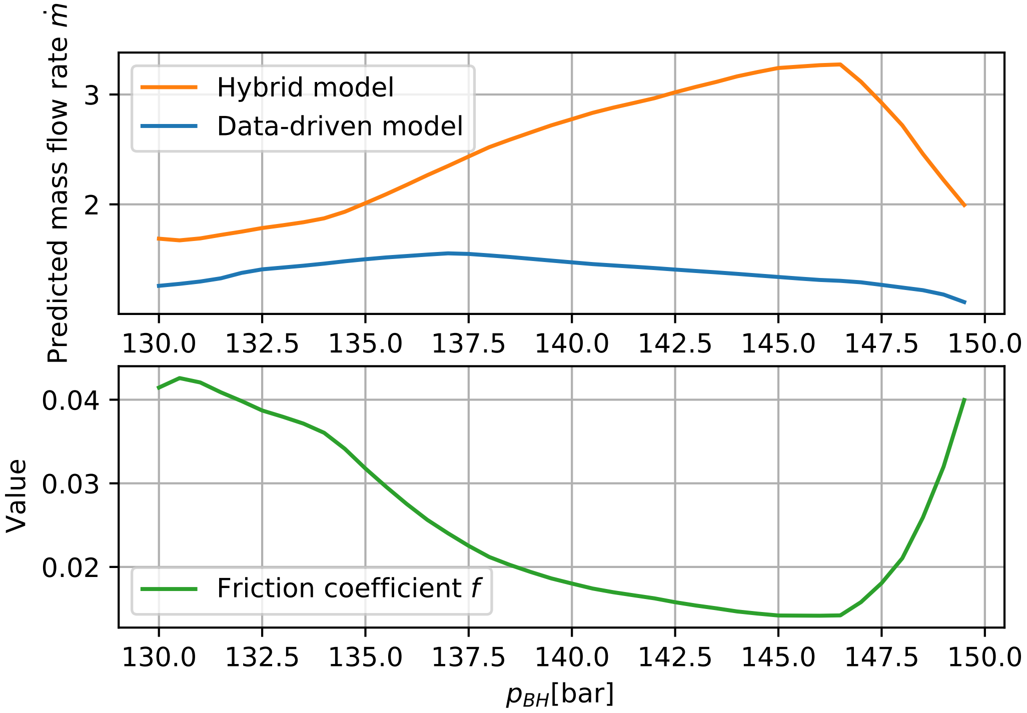


Figure 6.6: The upper subplot illustrates the mass flow rate output of the hybrid model and data-driven model as the bottomhole pressure p_{BH} increases. The rest of the inputs are kept constant. The mass flow rate has been scaled. The lower subplot shows the estimated friction factor from the neural network in the hybrid model over the same pressure increase.

6.3 Result: Checking results for new wells

In order to cross-reference the results from the previous two experiments with other wells, the models have been trained on six more wells with measurements available in the dataset. The models used for comparison are: the hybrid model with learnable parameters, the hybrid model without learnable parameters, and the data-driven model. Table 6.2 summarizes the performance of these models on all 7 wells, with the MAPE of the mass flow rate as the performance criteria. The wells in the table have been ordered in such a fashion that well number one has the least amount of mass flow rate, and well number seven has the most.

Table 6.2: Table summarizing the MAPE of the mass flow rate $\dot{m}$ for 7 wells using a hybrid model with learnable parameters, a hybrid model with no learnable parameters, and a fully data driven model. The well number is ordered from the well with least mass flow rate to the well with most.

Well	MAPE learnable	MAPE no learnable	MAPE data-driven
1	4.666%	4.970%	3.916%
2	3.091%	3.401%	3.273%
3	4.622%	4.345%	6.122%
4	5.855%	5.188%	5.460%
5	3.181%	**2.533**%	2.641%
6	4.921%	4.182%	3.701%
7	5.393%	4.379%	**6.250**%
Average	4.533%	4.143%	4.480%

Chapter 7

Discussion

7.1 Discussion: Optimizing the hybrid model for a single well

This section will discuss the results of the experiments aimed at optimizing the performance of a single well. This will be done by analysing the results from Section 6.1 in accordance with the experiment description in Section 5.1.1.

The first issue to be investigated is the effect of learnable parameters on the estimation accuracy. This was done by training the hybrid model outlined in Section 3.3 with-, and without learnable parameters. The corresponding model estimation performance is presented in Figure 6.1. Observing both graphs, we see that both model configurations achieve a 90% cumulative performance at 10% deviation. That is, 90% of the datapoints in the test set result in a deviation between the predicted model output and the true output value of less than 10%. That being said, the model with no learnable parameters actually achieves a slightly better performance than the model with learnable parameters. This can be seen when comparing the MAPE of the mass flow rate between the two model configurations in Table 6.1. The table shows that the model with learnable parameters has a MAPE of 4.921%, while the model without has 4.182%. One explanation for this might be that the estimation task for the model with learnable parameters has a larger function domain, increasing the number of local minima, and making the optimization of the model harder. Another explanation might be that the friction factor f has a larger influence on the model performance than the learnable parameters. As such, tuning only this parameter might be more efficient than also optimizing the learnable parameters. In this case, the learnable parameters must already be initialized to a reasonably good value, which might be the case in this configuration of the hybrid model. Considering the result as a whole, the performance difference in this case is too slight to be able to draw any strong conclusion from this one well. Section 6.3 will investigate whether or not this result also holds true for the other wells.

The second issue was whether or not regularizing the learnable parameters had any impact on the prediction performance. Once again, the performance can be gauged using cumulative performance plots, as seen in Figure 6.2. These results are almost indistinguishable, but one might give a small edge to the non-regularized configuration for this well. Regardless, there seem to be little impact on the performance of the model as a consequence of parameter regularization. However, by observing the values of the learnable parameters in Figure 6.3, it is clear that regularization plays a key role in the interpretability and physicality of the model. If regularization is left out, the learnable parameters may take any value that helps minimize the cost function. These findings are also echoed in Table 6.1, where the differences in model prediction accuracy are small, but with large spreads in the absolute errors of the learnable parameters, especially for water density ρ_W. Unlike neural network parameters, learnable parameters carry a physical meaning. Without regularization, the learnable parameters takes on values that are infeasible, and thus the model behaves more like a conventional black-box model than a model based on system dynamics. This phenomenon is clearly demonstrated by the water densities in Figure 6.3. Water density can not be a negative value and still carry a physical meaning, and as such this variable can no longer provide a production engineer with useful information.

The third question to be answered is whether or not these results are of acceptable accuracy. To do this we can start by comparing the results to other hybrid model applications. For instance, the hybrid model performance in this project is comparable to some of the hybrid model types outlined in Hotvedt et al. (2021). Another way of looking at the results so far is to consider them in relation to the resources required to produce them. The hybrid model used in this project has a mechanistic component based on the Bernoulli equation, which can be considered a basic fluid model when compared to industry leading models such as FlowManager (Amin, 2015). It also has multiple restrictions, such as steady-state homogeneous flow, which rarely holds true for physical systems. As such, one might expect this model to perform poorly on real well data. Similarly, the neural network used for the data-driven model component also has a shallow network architecture, perfectly trainable on a standard desktop computer. Overall, the hybrid model used in this project is a relatively light-weight implementation of the wellbore model. Despite this, the model still achieves accuracy close to results reported in literature (Bikmukhametov and Jäschke, 2020a; Amin, 2015; Hotvedt et al., 2020a). Nevertheless, the models are trained on measurements from multiphase flow meters, which have are known to drift in time (Falcone et al., 2013), yielding inaccurate measurements. Therefore, this hybrid model will never exceed the accuracy of the multiphase flow meters in which it is being trained on. However, multiphase flow meters themselves often suffer from accuracy discrepancies, especially over time. The hybrid model thus inherits this discrepancy when trained on data sourced from multi-phase flow meters. As such, we expect the accuracy of the hybrid model to be lower when compared to data gathered during the more accurate well tests. Future work should look into training and validating the hybrid model on well tests.

7.2 Discussion: Investigate the differences between the hybrid model and a data-driven model

The second experiment outlined in Section 5.1.2 is concerned with finding differences in the performance of the hybrid model and the data-driven model. The following discussion on this will be based on the results presented in Section 6.2.

We start by looking at the prediction performance of the two models. The cumulative performance of both the hybrid- and data-driven model is presented in Figure 6.4. Both these cumulative performance plots are comparable to those discussed earlier in Section 7.1. As such, the same conclusion regarding the accuracy of the models carry over to the data-driven model as well. From these graphs, it is difficult to perceive any noticeable performance difference. Instead, we consult Table 6.1. The mean absolute error (MAE) of the hybrid model is 1.268, while the data-driven model is 1.095. Furthermore, the mean absolute percentage error (MAPE) of the hybrid model is 4.921% and the data-driven model is 3.701%. From this it seems that the data-driven model inches out the hybrid model on performance by about 200 grams per second in absolute mass flow rate and 1 percentage point in percentage error. Looking at the scatter plot in Figure 6.5, it is once again not easy to make any certain claims about which model performs better. However, the scatter plot shows that both model predictions are usually close to each other. This means that datapoints difficult to predict for the data-driven model are also difficult for the hybrid model, and vice-versa. One might deduce from this result that the neural network components of the hybrid model has a large influence on the model prediction as a whole.

The second point outlined in the experiment description in Section 5.1.3 was a sensitivity analysis of the hybrid- and data-driven model. The result of this analysis can be seen in Figure 6.6. There are several interesting results from this experiment worth pointing out. The first is the abrupt change in friction coefficient value at $p_{BH} = 146$. This change is also reflected in the sharp decline of predicted mass flow rate from the hybrid model at the same pressure value. One explanation for this rapid decline in pressure might be that the bottomhole pressure starts approaching the reservoir pressure. In such a case, the pressure difference between the bottomhole and the reservoir would start to become too small to allow for fluid flow into the wellbore. However, the mechanistic model outlined in Section 3.1 does not include reservoir pressure as a parameter. As such, the dynamic between reservoir- and bottomhole pressure falls outside the realm of the first-principles model. Still, this dynamic is captured by the data, and as such is also reflected in the data-driven component of the hybrid model, in this case f. Observing the behavior of the data-driven model prediction, it seems to correlate to this discussion seeing as the predicted mass flow rate starts declining already at $p_{BH} = 137.5$. The reason why the hybrid model is increasing in the interval $p_{BH} = 137.5 \rightarrow 146$ where the data-driven model is decreasing, is because of the mechanistic model component of the hybrid model. The mechanistic component based on Equation (3.5) predicts an increased mass flow rate with increasing bottomhole pressure. The key point is that this first-principles model only models the wellbore, whereas the dataset reflects a complete system from reservoir through production choke. This is a drawback of this hybrid model. Ideally, one would have both the mechanistic model and the data covering the same system space. This difference in system space is reflected in the delta between the two model predictions in Figure 6.6.

Another interesting result from the sensitivity analysis is the relation between estimated friction coefficient f and the predicted mass flow rate $\dot{m}$ from the hybrid model. We compare the shape of the two plots in Figure 6.6 with the mechanistic expression for f in Equation (2.10) and Equation (2.11). According to the relations presented in Section 2.2.2, an increase in $\dot{m}$ would result in a decrease f. This is the same relation observed in the plots. This is also clear from the first-principles mass flow rate model in Equation 3.5. However, the expressions (2.10) and (2.11) can also be used to calculate the expected value of f given the the predicted output $\dot{m}$ of the hybrid model. The viscosity μ can be found through correlations as with f (Carr et al., 1954), but will in this illustrative example be set to a representative value $\mu = 0.00246[Pas]$. Using the same mass flow rate as predicted by the hybrid model at $p_{BH} = 130[Pa]$, the resulting friction coefficient is then $f_m = 0.0039$. This is about one order of magnitude smaller than the estimated f from the neural network component of the hybrid model. Seeing as the estimated friction coefficient is so large, the neural network component of the hybrid model might have a large influence on the predicted model output. One reason as to why the estimated f is so large, is that all the differences between the mechanistic model output and the data in the dataset needs to be represented through the friction factor, seeing as this is the output of the data-driven part of the hybrid model. Furthermore, this output is not regularized. As such, a production engineer may not gain any insight to the actual value of the friction factor by looking at the output of the neural network. This is also a drawback of this hybrid model configuration, and might be something worth investigating in future work.

7.3 Discussion: Checking results for new wells

The final experiment outlined in Chapter 5 was the cross-referencing of the results from the previous experiments with all the wells in the dataset. The most natural way of doing this is to look at the performance of the different wells, summarized in Table 6.2. First we will look at the overall performance of all the wells and see if there are any outliers. After that we can compare the results with the findings from Section 7.1 and 7.2 .

We start by getting an overview of the general performance of all the models. From Table 6.2, the best performing well configuration is the hybrid model with no learnable parameters on well 5 with a MAPE of 2.533%. The worst performing well configuration is the data-driven model on well 7 with a MAPE of 6.250%. The spread in accuracy suggests that the model prediction performance is influenced by the differences between the wells. For instance, such differences may be the number and accuracy of the datapoints in the corresponding datasets for each well, or the accuracy of the given well parameters. Furthermore, each well operates at different flow rates and temperatures, which may impact the prediction accuracy. Unforeseen phenomena not captured by the first-principles may also occur at different rates on different wells. Consequently, it is not easy to pin-point the exact reason for the accuracy spread. However, the overall model prediction accuracy is satisfactory compared to results reported in literature.

One of the findings from the first experiment discussed in Section 7.1 was that the hybrid model with no learnable parameters performed better than the configuration with learnable parameters. This result is also shown in Table 6.2 by looking at well 6, which is the well that has been used so far in the analysis. Comparing the two columns of hybrid model configurations in Table 6.2 reveals that there are only two wells in which the result does not hold true: well one and two. Furthermore, by looking at the average MAPE over all the wells, one can observe that the hybrid model without learnable parameters performs the best of the three models by a slight margin. The hybrid model without learnable parameters has an average MAPE of 4.143%, against the data-driven models of 4.480%, and the hybrid models with learnable parameters of 4.533%. This result suggests two things. First, comparing the hybrid model without learnable parameters and the data-driven model, the results suggests that introducing first-principles to the model has increased the model predictive performance. At first glance, this might be refuted by the fact that with learnable parameters, the hybrid model performs worse than the purely data-driven model. However, as discussed earlier, it may be that the inclusion of the learnable parameters increase the complexity of the optimization problem. This may again result in non-optimal convergence of the system parameters. In this case, one might investigate other estimation problem formulations to handle this increased complexity. Secondly, it seems that the inclusion of learnable parameters may pull the predicted mass flow rate away from the value given in the dataset. Once again, it is important to stress that the mass flow rates in the dataset do not represent the true mass flow rate in the system, only the best prediction available to us. Consequently, predicting a different mass flow rate than the one included in the dataset is not necessarily a disadvantage of the model. Further work should investigate whether or not the hybrid model with learnable parameters perform better than the configuration without when compared to data gathered from well tests.

Chapter 8

Conclusion

The results from Chapter 6 show that the hybrid wellbore model can be implemented for a real production system, with accuracy comparable to existing literature. The average MAPE for the hybrid model with learnable parameters was less than 5%. This result is promising, seeing as the hybrid model implemented here is light-weight and cheap to implement.

The characteristics of different hybrid models have been tested and analysed. The work done in this thesis suggests that the serial hybrid model used is largely influenced by the neural network component. Still, the sensitivity analysis shows that some degree of physical behaviour remains from the first-principles model. The inclusion of learnable parameters was also shown to push the model away from the measured mass flow rate gathered from the multiphase flow meters. Overall however, all the model configurations were able to predict a mass flow rate close to the measured rate from the dataset.

Some limitations of the hybrid wellbore model was also discovered. The most significant limitation is the lack of mulitphase flow data directly sourced from the wellbore. By using the data from the MPFMs, the behaviour of both the reservoir and the production choke is also included. Consequently, the data-driven model component is trained on data representing a bigger system than just the wellbore itself. Another limitation discovered was the non-interpretability of the friction factor f. By letting the neural network estimate f freely, all the non-modeled phenomena present in the dataset has to be represented through the value of f. As such, the estimated friction coefficient no longer becomes representative of a real friction coefficient of the wellbore.

In summary, this thesis has shown the viability of hybrid modeling in the context of virtual flow metering in petroleum production wellbores. Additionally, it was shown that hybrid models are able to achieve accuracy comparable to that of fully data-driven models, while retaining interpretability. Different configurations of the hybrid model was also analysed, laying groundwork for other developers to adapt hybrid modeling techniques for different systems.

8.1 Further Work

Following will be suggestions for further work on this topic.

- The biggest limitation to the work done in this thesis, is applying the hybrid model to only the wellbore when the data available also includes the reservoir and production choke. Work could continue to extend the wellbore model to also include these two aspects into a more complete mass flow model. The data from the multiphase flow meters and the hybrid model would then describe the same system.
- Considering the inaccuracies of many multiphase flow meters, the results in this thesis could be evaluated against more accurate measurements from well test. Ideally, the model would also be trained on this kind of data. Future work could include training hybrid models on well test data when these measurements become numerous enough.
- In order to preserve the interpretability of the friction factor in the hybrid model, future work could implement parameter regularization on this neural network output.

Bibliography

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

Amin, A. (2015). Evaluation of Commercially Available Virtual Flow Meters (VFMs). In *OTC Offshore Technology Conference*. OTC-25764-MS.

Andrianov, N. (2018). A machine learning approach for virtual flow metering and forecasting. *IFAC-PapersOnLine*, 51(8):191–196. 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production OOGP 2018.

Awad, M. and Muzychka, Y. (2008). Effective property models for homogeneous two-phase flows. *Experimental Thermal and Fluid Science*, 33(1):106 – 113.

Baker, R. O., Yarranton, H. W., and Jensen, J. L. (2015). 2 - rock and fluid properties. In Baker, R. O., Yarranton, H. W., and Jensen, J. L., editors, *Practical Reservoir Engineering and Characterization*, pages 35–66. Gulf Professional Publishing, Boston.

Bikmukhametov, T. and Jäschke, J. (2020a). Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers & Chemical Engineering*, 138:106834.

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

Carr, N. L., Kobayashi, R., and Burrows., D. B. (1954). Viscosity of hydrocarbon gases under pressure. *J Pet Tecnol*, 6:47–55.

Cengel, Y. A. and Cimbala, J. M. (2014). *Fluid Mechanics: Fundamentals and Applications, 3rd ed.* McGraw Hill.

Corneliussen, S., Couput, J.-P., Dahl, E., Dykesteen, E., Frøysa, K.-E., Malde, E., Moestue, H., Moksnes, P. O., Scheers, L., and Tunheim, H. (2005a). Handbook of multiphase flow metering.

Corneliussen, S., Couput, J.-P., Dahl, E., Dykesteen, E., Frøysa, K.-E., Malde, E., Moestue, H., Moksnes, P. O., Scheers, L., and Tunheim, H. (2005b). Handbook of multiphase flow metering.

Devold, H. (2007). Oil and gas production handbook an introduction to oil and gas production.

Dixon, S. and Hall, C. (2010). *Fluid Mechanics and Thermodynamics of Tur- bomachinery, 6th ed.* Elsevier.

El-Amir, H. and Hamdy, M. (2020). *Deep Learning Pipeline.* Apress.

Equinor (2021). Energy perspectives 2021. https://www.equinor.com/en/sustainability/energy-perspectives.html.

Falcone, G., Hewitt, G., Alimonti, C., and Harrison, B. (2013). Multiphase flow metering: Current trends and future developments. *Journal of Petroleum Technology.*

Fredriksen, M. (2021). Hybrid modeling of a production wellbore. Master's thesis, Norwegian University of Science and Technology NTNU, The address of the publisher. An optional note.

Ganat, T. (2016). Reliability analysis of multiphase flow measurements in different oil fields. *OIL GAS-EUROPEAN MAGAZINE.*

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* The MIT Press.

Grimstad, B., Gunnerud, V., Sandnes, A., Shamlou, S., Skrondal, I. S., Uglane, V., Ursin-Holm, S., and Foss, B. (2016). A Simple Data-Driven Approach to Production Estimation and Optimization. volume All Days of *SPE Intelligent Energy International Conference and Exhibition.* SPE-181104-MS.

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

Guo, B., Lyons, W., and Ghalambor, A. (2007). *Petroleum Production Engineering, A Computer-Assisted Approach.* Elsevier.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.

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

Hotvedt, M., Grimstad, B., and Imsland, L. (2020b). Identifiability and interpretability of hybrid, gray-box models.

Hotvedt, M., Grimstad, B., Ljungquist, D., and Imsland, L. (2021). On gray-box modeling for virtual flow metering. *CoRR*, abs/2103.12513.

Jansen, J.-D. (2015). *Nodal Analysis of Oil and Gas Wells – Theory and Numerical Implementation.* Addison-Wesley, Delft University of Technology (TU Delft) Department of Geoscience and Engineering, The Netherlands.

Kanin, E., Osiptsov, A., Vainshtein, A., and Burnaev, E. (2019). A predictive model for steady-state multiphase pipe flow: Machine learning on lab data. *Journal of Petroleum Science and Engineering*, 180:727 – 746.

Kingma, D. and Ba, J. L. (2015). Adam: a method for stochastic optimization. *International conference on learning representations.*

LaNasa, P. J. and Upp, E. L. (2014). 5 - from theory to practice. In LaNasa, P. J. and Upp, E. L., editors, *Fluid Flow Measurement (Third Edition)*, pages 69–87. Butterworth-Heinemann, Oxford, third edition edition.

Mohammadpoor, M. and Torabi, F. (2020). Big data analytics in oil and gas industry: An emerging trend. *Petroleum*, 6(4):321–328. SI: Artificial Intelligence (AI), Knowledge-based Systems (KBS), and Machine Learning (ML).

Molnar, C. (2021). A guide for making black box models explainable. https://christophm.github.io/interpretable-ml-book/limo.html. Acccessed: 13.06.2021.

Monteiro, D. D., Duque, M. M., Chaves, G. S., Filho, V. M. F., and Baioco, J. S. (2020). Using data analytics to quantify the impact of production test uncertainty on oil flow rate forecast. *IFP Energies Nouvelles*, 75:1–15.

Oliver, D. S., Zhang, Y., Phale, H. A., and Chen, Y. (2011). Distributed parameter and state estimation in petroleum reservoirs. *Computers & Fluids*, 46(1):70–77. 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).

Pitarch, J. L., Sala, A., and de Prada, C. (2019). A systematic grey-box modeling methodology via data reconciliation and sos constrained regression. *Processes*, 7(3).

Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511.

Ross, A. (2010). Well testing - an evaluation of test separators and multiphase flow meters. In *28th International North Sea Flow Measurement Workshop*. TUV NEL.

Sandnes, A. T., Grimstad, B., and Kolbjørnsen, O. (2021). Multi-task learning for virtual flow metering.

Sessions, V. and Valtorta, M. (2006). The effects of data quality on machine learning algorithms. pages 485–498.

Solle, D., Hitzmann, B., Herwig, C., Pereira Remelhe, M., Ulonska, S., Wuerth, L., Prata, A., and Steckenreiter, T. (2017). Between the poles of data-driven and mechanistic modeling for process operation. *Chemie Ingenieur Technik*, 89.

Solution Seeker (2021). Technology - data mining and squashing. `https://www.solutionseeker.no/#technology-big-data`. Acccessed: 18.01.2021.

Speight, J. G. (1999). 2.2 density and specific gravity. *The Chemistry and Technology of Petroleum.*

Sutton, R. P. (1985). Compressibility factors for high-molecular-weight reservoir gases. In *SPE Annual Technical Conference and Exhibition.*

Takacs, G. (2015). Chapter 2 - a review of production engineering fundamentals. In Takacs, G., editor, *Sucker-Rod Pumping Handbook*, pages 13–56. Gulf Professional Publishing, Boston.

van Smeden, M., Moons, K. G., de Groot, J. A., Collins, G. S., Altman, D. G., Eijkemans, M. J., and Reitsma, J. B. (2019). Sample size for binary logistic prediction models: Beyond events per variable criteria. *Statistical Methods in Medical Research*, 28(8):2455–2474. PMID: 29966490.

NTNU
Norwegian University of
Science and Technology

Solution
Seeker