NTNU – Trondheim
Norwegian University of
Science and Technology

Dynamic Estimation for Controlling a Subsea Production System

Virtual Flow Metering using B-spline
Surrogate Models

Patrick Michael Robertson

Master of Science in Engineering Cybernetics (2 year))
Submission date: June 2014
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Bjarne Grimstad, ITK
Tore Tjøstheim, BP

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of Science and Technology

Faculty of Information Technology,
Mathematics and Electrical
Engineering

Department of Engineering Cybernetics

Master project

Name of candidate:	Patrick Robertson
Subject:	Engineering Cybernetics
Title:	Dynamic estimation for controlling a subsea production system
Title (in Norwegian):	Dynamisk estimering for regulering av et subsea produksjonssystem

An oil and gas production system, where wells are connected to manifolds with multiple pipelines, needs to be operated within a number of constraints. Common constraints include capacity of processing facilities, well drawdown and erosional limits in pipelines.

This project will study the use of dynamic estimation for automatic control of a production system. In particular, dynamic estimation of flows in a production network where only pressure and temperature measurements are available. The goal is to continuously push against constraints on flow rates and velocities. The system should resemble the Skarv field which is located in the Norwegian sea and operated by BP.

Task description:

1. Literature review on state estimation methods and mathematical modeling of oil production networks. This should include advantages of using optimization based methods instead of the Kalman Filter.
2. Develop/select suitable steady-state and dynamic models for one or two specific templates with wells. Fit the model to one template at the Skarv field using available data and other information.
3. Implement a detailed model of the same system for comparison purposes. OLGA or similar tool to be used.
4. Define a set of typical operating scenarios; some with predominantly stable operation and others with extensive variability.
5. Develop, implement and test methods for optimization based methods (using simulation).
6. Compare the performance of steady-state versus dynamic models for the operating scenarios defined in item 4. Use the detailed model defined in item 3, and historical data from Skarv to assess the performance of the estimators.

Starting date:	20.01.2014
End date:	20.06.2014
Co-supervisor:	Bjarne Grimstad, NTNU Tore Tjøstheim, BP

Trondheim, 20.01.2014

Bjarne Foss
Professor/supervisor

Abstract

Knowledge of the flow rates from individual wells in a subsea production system can greatly improve decision-making processes. This thesis investigates flow estimation in the subsea template Tilje in the BP-operated Skarv field, with automatic rate control in mind. Two optimization-based estimation methods, one with static models and one with dynamic models, are developed and compared against each other by means of benchmark data from an OLGA model and historical field data from Tilje. Ideas from recent developments in production optimization are applied in the derivation of the production network models. This amounts to representing the flow network as individual network components with B-spline approximated models., leading to a transparent model which allows for easier estimator tuning, easier constraint handling and faster solution times.

The results presented in this thesis show that the method with static models is able to predict flow rates with acceptable accuracy, has good robustness properties and fast solution times, indicating that the method has potential for use in an automatic rate control system. The method with dynamic models is less robust, more complex, and does not seem to improve the flow rate estimates. However, it does include estimates of mass and holdup and could potentially be useful if the method was improved. In addition to providing feedback for automatic rate control systems, the methods described in this thesis could potentially be used as cornerstones in advanced decision support tools, such as flow assurance systems and condition monitoring systems.

Sammendrag

Kjennskap til individuelle brønnrater i et subsea produksjonssystem kan bedre beslutningsprosessene relatert til drift av systemet. Denne masteroppgaven dreier seg om rateestimering i subseasystemet Tilje i det BP-opererte Skarv-feltet, med fokus på automatisk regulering av brønnrater. To optimaliseringsbaserte metoder, en med statiske modeller og en med dynamiske modeller, blir utviklet og sammenlignet ved hjelp av referansedata fra OLGA og historiske feltdata fra Tilje. Metodene benytter seg av senere tids ideer fra produksjonsoptimalisering. Produksjonsnettverket representeres ved individuelle nettverkskomponenter med B-spline-approksimerte modeller. Dette fører til en transparent modell som gjør det lettere å tune estimatorene, lettere å håndtere begrensninger i systemet, og raskere løsning av optimaliseringsproblemet.

Resultatene presentert i oppgaven viser at metoden med statiske modeller er i stand til å estimere rater med tilfredsstillende nøyaktighet, har gode robusthetsegenskaper og raske løsningstider. Dette indikerer at metoden har potensiale for å brukes i et system for automatisk regulering av brønnrater. Metoden med dynamiske modeller er mindre robust, mer kompleks og synes ikke å forbedre rateestimatene betraktelig. Likevel inkluderer den estimering av masse og holdup, og kan potensielt være av nytte hvis metoden forbedres. I tillegg til å levere estimater til et automatisk ratereguleringssystem, kan metodene beskrevet i oppgaven potensielt benyttes som hjørnesteiner i avanserte beslutningsverktøy, som systemer for strømningssikring og tilstandsmonitorering.

Preface

This thesis was written during the final semester of the two-year Master's programme at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). The thesis has been written in collaboration with the Center for Integrated Operations in the Petroleum Industry (IO Center) at NTNU, and British Petroleum (BP). I first came in contact with the IO Center and BP during the summer of 2013, where I had a summer internship with the IO Center at BP's business and technology center (ICBT) in Sunbury, UK. This work revolved around the C++ optimization framework CENSO, and continued with a B-spline specialization project last semester. This thesis can be considered an application of the knowledge and experience I have gained through the summer internship and specialization project (and, of course, the two years at NTNU).

I would like to take this opportunity to thank several people who have been a great help throughout this semester. First, I would like to thank my supervisor, professor Bjarne Foss, both for putting me in contact with BP, and for his valuable insight. Special thanks go to my co-supervisor Bjarne Grimstad for his encouragement, guidance, constructive feedback and friendship. The discussions we have had through the past year have been interesting, challenging and highly appreciated. I would also like to thank Esmaeil Jahanshahi at the IO Center for giving me a crash course in OLGA simulations.

Several people at the BP Stavanger office have been a great help; first and foremost I must thank Tore Tjøstheim and Paul Hocking for their encouragement and insight. I must also thank Fabian Bjørnseth for patiently answering my questions about well testing and modelling, and Cathy Rawsthorne for providing me with the necessary OLGA files to construct a realistic model of the Tilje template. I have been fortunate enough to receive financial support from the IO Center and BP for two trips to the BP office in Stavanger, for which I am grateful.

Finally, i must thank my classmates in the two-year Engineering Cybernetics Master's programme for making the past two years awesome (and this last semester less lonely); my parents Janicke and Mike for their continued encouragement and support, and my sweetheart Linda, for just being her.

Trondheim, June 20, 2014

Patrick Michael Robertson

Contents

List of Figures

List of Tables

List of Abbreviations

CENSO	Convex ENvelopes for Spline Optimization
EKF	Extended Kalman Filter
FAS	Flow Assurance System
FPSO	Floating Production, Storage and Offloading
GOR	Gas-Oil Ratio
IO Center	Center for Integrated Operations in the Petroleum Industry
IP	Interior Point
IPR	Inflow Performance Relationship
LQR	Linear Quadratic Regulator
MHE	Moving-Horizon Estimation
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
MPC	Model Predictive Control
MPFM	Multiphase Flow Meter
PDF	Probability Density Function
PI	Productivity Index
SPE	Society of Petroleum Engineers
SQP	Sequential Quadratic Programming
SWLS	Static Weighted Least-Squares
UKF	Unscented Kalman Filter
VFM	Virtual Flow Metering
VLP	Vertical Lift Performance
WC	Water Cut
WPC	Well Performance Curve

Chapter 1

Introduction

In the last few decades, we have seen a shift in the offshore petroleum industry, from conventional platform operations to increasingly remote subsea production systems. This has resulted in an increased demand for real-time monitoring and control (Bringedal et al., 2010). In this thesis, we will look at using available real-time data to estimate flow rates in a subsea production network, also known as *virtual flow metering* (VFM). This introductory chapter will provide some background information and motivate the need for flow estimation. Thereafter, the particular case discussed in this thesis, namely the Skarv field, will be presented. Finally, we will go through the structure of the thesis and present the general notation which will be used.

1.1 Background and motivation

This section covers some basic background information and motivates the need for flow estimation. In addition, we will take a quick look at some of the solutions in use today.

1.1.1 Operational decisions in petroleum production and the need for flow monitoring

In short, oil and gas is produced by drilling wells into a petroleum reservoir, and connecting these wells to a gathering system/production network which brings the reservoir fluids (usually a composition of oil, gas and water) from the wells to a surface processing plant. The flow is driven by the high pressure in the reservoir, sometimes aided by the use of various artificial lift methods. The main purpose of the processing plant is to separate oil, gas and water. After processing, the oil and gas is sold, and produced water is purified before being e.g. discharged to sea. Some gas or water may be re-injected to the reservoir to maintain pressure and aid production (Gunnerud, 2011).

The task of operating a petroleum asset (i.e. a reservoir with wells and a production system) is not trivial. As described in (Foss, 2012), there are several decisions to be made on different time horizons, from long-term investment strategies to short-term control decisions. These decisions can be categorized and organized in a multi-level control hierarchy, where each level provides targets/set points to and receives feedback from the level below (Fig. 1.1).

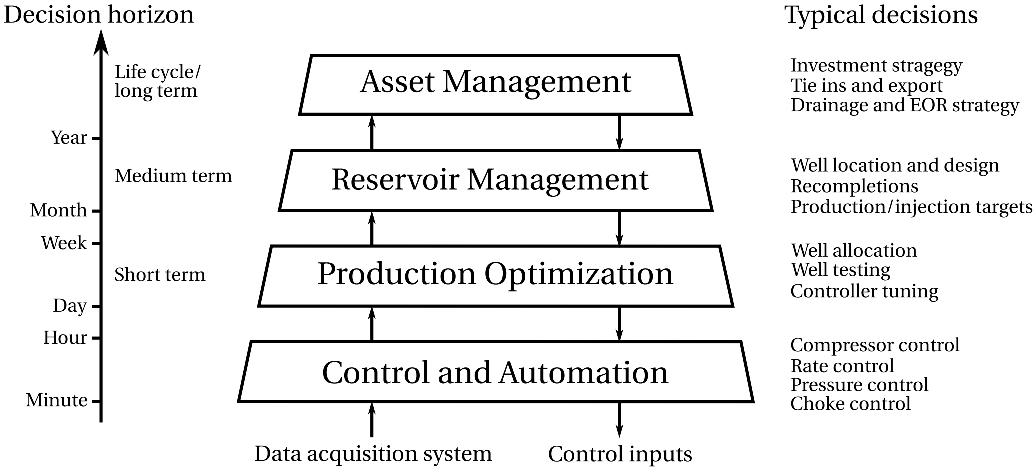


Figure 1.1: Multi-level control hierarchy (Foss, 2012).

In order to make good decisions, oil companies use an array of *decision support* tools which, based on field data and various mathematical models, provide suggestions for how to optimally operate the asset. An important part of this is the ability to determine the individual flow rates from each well. With reference to Figure 1.1, such information can support decision making on several time scales:

- *Reservoir Management*: Information about individual well rates can increase understanding about fluid flows in the reservoir and aid depletion planning (Heddle et al., 2012). In addition, accurate well rate monitoring is important with respect to e.g. production reporting and taxes/royalties, in particular for reservoirs spanning more than one license (Lerma et al., 2006).
- *Production Optimization*: Production optimization involves the determination of the optimal contribution from each well to the total production (i.e. the *well allocation*). The goal may be to maximize oil production, or to produce at a target rate. Accurate well rate monitoring can increase the confidence and acceptance of proposed suggestions from production optimization software.
- *Control and Automation*: If a well is rate controlled, some measure of the actual well rate is necessary for feedback.

In addition, we need to ensure the flow rate does not violate operational limits at any given location. Potential problems include flow-induced vibration and erosion of pipelines and other equipment. Knowledge of the flow rates in the production system is also an important input to flow assurance systems (FAS) in the sense that the flow rates can be used together with pressure and temperature measurements to predict (and thus avoid) unwanted conditions in the network, e.g. hydrate formation and wax or asphaltene deposition. Thus, providing feedback for rate control, which is the goal suggested by the title of this thesis, is just one of many potential benefits of having accurate well rate monitoring.

1.1.2 Flow monitoring in petroleum production networks

Several approaches can be found for determining well flow rates. To leave this section short, we will mention the most simple approach and go straight to the "state of the art", without all the in-betweens.

The conventional way to determine well flow rates is to measure the flow directly or indirectly over a period of time, on a weekly or monthly basis. This process is known as *single-rate well testing*, and typically results in a set of measured average flow rates (oil, gas, water), relative gas and water content (gas-oil ratio and water cut; these terms are discussed in Chapter 3), and pressure/temperature conditions. Then, the production of the well is assumed to not change significantly until the next well test. Consequently, declining production rates, change in gas/water content or other issues (e.g. well instability) may not detected until the next well test (Goh et al., 2007).

Newer subsea developments (e.g. Skarv) are often equipped with multiphase flow meters (MPFM), which have the ability to measure one or more of the individual flow rates of oil, gas and water. In this case, a well test can be performed by routing the well stream through the MPFM and recording the resulting measurements, assuming the MPFM is properly calibrated. For fields without MPFMs (or with malfunctioning/poorly calibrated MPFMs), the well must be routed to a dedicated test separator to determine the performance of the well. Well tests require planning and personnel, and typically result in some lost production (depending on the type of test). As a result, they are expensive to perform (Bieker et al., 2007). In addition, MPFMs are also very expensive, both with respect to investment costs and maintenance (Melbø et al., 2003).

Subsea production systems may have long tiebacks[1], which complicates well testing. For example, single-well testing to a topside separator may be difficult due to slugging or other issues. Although MPFMs solve this problem, they are expensive, errorprone, and some fields simply do not have them (Bringedal et al., 2010). However, most subsea production systems are well equipped with pressure and temperature measurements, which are easily available through high-capacity data acquisition and storage systems. This has enabled the industry to develop methods for using available sensor information around each well (typically pressure and temperature transmitters) to *esti-*

[1] The *tieback* is the length of a common pipeline which connects a cluster of subsea wells to the processing plant, e.g. a platform or FPSO vessel.

mate well flow rates in real time. These methods typically rely on a mathematical model of the well, which is updated/fitted to measured flow rates from available well tests. Since these methods provide real-time or near-real-time flow estimates without relying on a physical flow meter, they are aptly named *virtual flow metering* (VFM) methods. As we will see in the next section, industry experience has shown that VFM implementations can increase the intervals between expensive well tests, and provide an accurate alternative and/or backup to expensive and error-prone MPFMs.

1.1.3 Existing VFM solutions and industry experience

Although journal publications regarding VFM development are scarce (probably due to industry-driven development and intellectual property rights), some general impressions can be found in various conference papers. These papers are listed together with a few examples of commercial and in-house VFM solutions in Table 1.1 below.

Vendor/operator	**Product name**	**Reference**
FMC Technologies	FlowManager VFM	(Holmås & Løvli, 2011)
ABB	Well Monitoring System	(Melbø et al., 2003)
SPT Group	OLGA Online VFM	-
Emerson	Roxar FieldWatch VFM	-
Baker Hughes	Neuraflow VFM	-
Belsim	ValiUpstream	-
BP	ISIS Rate&Phase	(Heddle et al., 2012)
Shell	FieldWare PU	(Goh et al., 2007)

Table 1.1: Examples of commercial and in-house Virtual Flow Metering systems.

All VFM systems discussed in these papers seem to rely on optimization and reconcilation methods (however, due to the relatively superficial literature on the subject, the author cannot rule out that other approaches are in use). A modular approach is taken where a library of model components (e.g. well, choke, venturi and pipeline models) are used, together with mass, momentum and energy balance equations, to piece together a model of the production system in question. The resulting model is treated as a black-box simulator where flow rates and other parameters like e.g. reservoir pressure are inputs and resulting pressures in the production system (corresponding to available measurements) are outputs. Then, an optimization algorithm iteratively finds the flow rates/parameters which correspond to a "best fit" with the measured pressures; i.e. solves an optimization problem which will look something like

$$\underset{\mathbf{q},\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} w_i \left(p_i - \tilde{p}_i\right)^2, \tag{1.1a}$$

$$\text{subject to} \quad [p_1, \ldots, p_n]^\top = \mathbf{f}(\mathbf{q}, \boldsymbol{\theta}, \tilde{\mathbf{T}}), \tag{1.1b}$$

where $\mathbf{f}(\cdot)$ is a black-box model of the production network, $\mathbf{q}, \boldsymbol{\theta}, \tilde{\mathbf{T}}$ are flow rates, parameters and measured temperatures (respectively), n is the number of pressure measurements, $\{w_i\}_{i=1}^n$ are weights on each pressure error, $\{p_i\}_{i=1}^n$ are model-predicted pressures and $\{\tilde{p}_i\}_{i=1}^n$ are measured pressures. Most VFM systems rely on static/steady-state models, although there has been a few implementations with dynamic models, e.g. FMC's FlowManager Dynamic (Holmås & Løvli, 2011) and SPT Group's OLGA Online VFM.

For densely instrumented systems, VFM systems may be able to calculate a large number of unknowns, e.g. individual oil, gas and water rates in addition to several other parameters. However, if a limited amount of measurements are available due to sparse instrumentation or malfunctioning instruments, this will typically result in increased uncertainty or estimation of fewer unknowns. This is elaborated on in (Melbø et al., 2003), and we will also discuss it briefly at the end of Chapter 2.

Model maintenance

The model components mentioned above need to be maintained in order to continuously provide accurate estimates. This is typically achieved by fitting, or calibrating, the models to available well tests. Multi-rate tests[2] are preferred, since this will result in a model which is valid over a larger operational envelope (Heddle et al., 2012). To limit the amount of needed maintenance, some VFM solutions allow in-house (customer-maintained) models to be used for flow estimation, enabling easier integration with already existing workflows. One example of this is FMC's "common field model" approach, which is discussed in (Bakken et al., 2011). As mentioned in (Bringedal et al., 2010), a major challenge in industrial VFM solutions is the cost of maintaining several models in parallel.

Benefits

Aside from the obvious benefit of providing real-time monitoring of well rates, experience from the industry has shown that VFM methods have a number of additional benefits (Heddle et al., 2012; Lerma et al., 2006; Hauge & Horn, 2005; Bakken et al., 2011):

- A significant reduction in the effort required to compute/report daily production.
- Increased interval between well tests - VFM system notifies operator when a new well test is needed for model calibration.
- Early detection of gas coning and water breakthrough (i.e. sudden increases in gas or water content).

[2] Multi-rate well tests are (as the name entails) well tests where several rates are tested in order to span a larger operational envelope. Such tests take longer, and are thus more expensive, than single-rate tests (Bieker et al., 2007).

- Assurance that operation is within integrity limits, i.e. minimizing the risk of damage or wear due to vibration or erosion.
- Consistency checking of pressure, temperature and any physical flow rate metering systems, which enables redundancy and detection of faulty equipment.
- More accurate well rate information enables more accurate history matching of reservoir simulation models, which in turn leads to better reservoir depletion planning.
- Valuable input to flow assurance systems, and can thus be used to predict and avoid e.g. hydrate formation, wax and asphaltene deposition.

Based on the need and proved benefits of flow estimation, the motivational aspects of this thesis should now be clear. We will now move on to describe the specific field under investigation; namely, the Skarv field.

1.2 The Skarv field

This section will present the Skarv field, which is the specific case studied in this thesis.

1.2.1 General information

The Skarv field was discovered in 1998, and is located in license blocks 212, 159 and 262, on the Haltenbanken terrace in the Norwegian Sea, about 210 kilometres offshore Sandnessjøen (see Figure 1.2). It is operated by BP Norge AS and partnered by Statoil ASA, E.ON E&P Norge AS and PGNIG Norge AS. The wells are located at 350-450 m water depth and the tieback distances range from 4 to 14 km. Production started on New Year's Eve 2012, and the field is expected to produce for 25 years (Larsen & Hocking, 2012; BP, 2014).

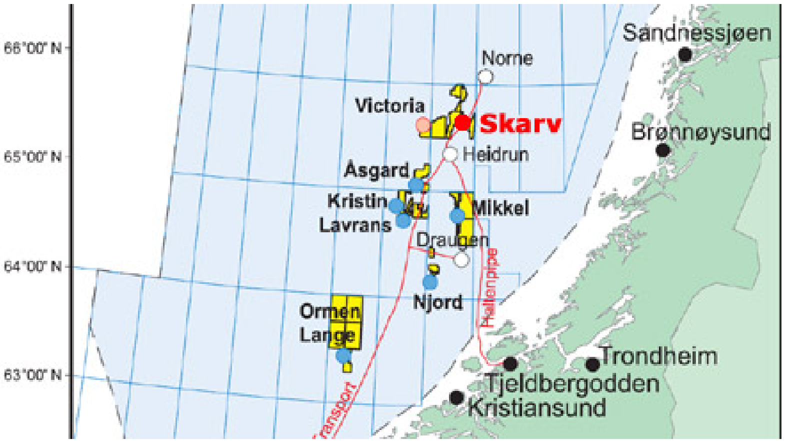


Figure 1.2: Location of the Skarv field (Ptil, 2009).

Topology of subsea production network

Currently, the Skarv field is operating twelve production wells and four injection wells. The wells are distributed over five subsea templates, and produced fluids are transported via pipelines and flexible risers to the Floating Production, Storage and Offloading (FPSO) vessel "*Skarv FPSO*". Produced oil is offloaded via shuttle tanker, while produced gas is compressed and exported via pipelines. An overview over the system topology is shown in Figure 1.3, and a well summary is given in Table 1.2.

Instrumentation

The Skarv field is a brand new development which enjoys dense instrumentation and state-of-the-art data-to-desktop solutions. The wells and pipeline systems are equipped with a large amount of pressure and temperature sensors, which transmit measurements in near real time[3] to a PI (historian)[4] server. One MPFM and an associated routing system is fitted in each template, which enables the continuous monitoring of a single well, or a group of wells (see Fig. 6.1 in Chapter 6).

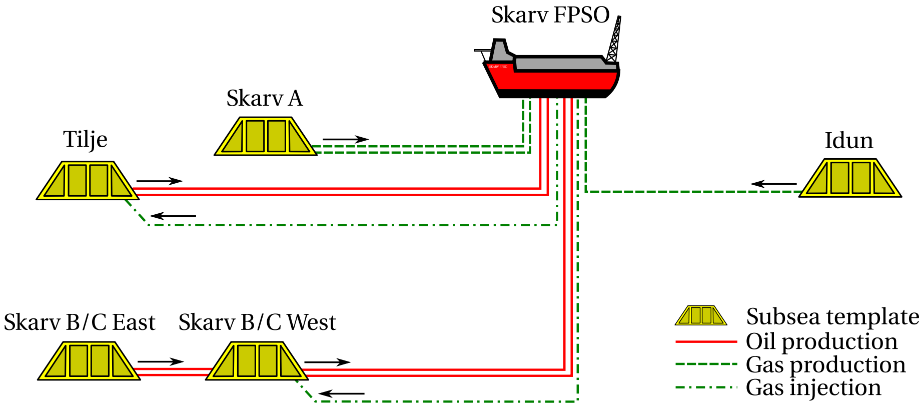


Figure 1.3: The Skarv field subsea production system.

Template	Production wells	Injection wells	Flowlines	Tieback distance
Tilje	2 (oil)	2 (gas)	Dual	5 km
Idun	2 (gas)	0	Single	14 km
Skarv A	4 (gas)	0	Dual	5 km
Skarv B/C	4 (oil)	2 (gas)	Dual	4 km

Table 1.2: Well summary, Skarv field.

[3] While measurements are sent in real time to a SCADA system, data is sent in "bursts" to the PI server every 30 seconds.

[4] PI is a much-used system for data storage and analysis, vendored by OSIsoft. For more information, see http://www.osisoft.com.

Operational constraints

Gas producing wells are rate/velocity constrained to prevent erosion of the corrosion protective layer in the pipelines. In addition, total gas production may be limited by export demand. The most common constraints in oil wells are drawdown constraints, meaning that the bottom hole pressure must be kept above a certain value. For Skarv, this is to prevent the wells from drawing sand from the reservoir. In extreme cases, a too low bottom hole pressure can cause formation damage due to high pressure gradients in the near-well region of the reservoir. In addition, the choke valves have a limited rate of change (one or two steps per half hour) - this is to prevent rapid changes in bottom hole pressure which could also result in some sand production.

Well testing and MPFM calibration

The Skarv wells are tested at regular intervals (every 1-2 weeks) by routing the well through the MPFM fitted on the subsea template. The tests are typically single-rate tests with a duration of 4-24 hours. Average flow rates, pressures and temperatures are recorded and used for updating well models. The MPFM is calibrated by isolating it from the well streams; in this case the phase detector should fill up with gas, and the MPFM should indicate 100 % gas fraction. If it does not, parameters are adjusted until 100 % is indicated. Occasionally, the MPFM should also be calibrated by routing both wells through the MPFM to a test separator, where the rates of each phase are measured and compared to the MPFM.

1.2.2 Important simplifications and assumptions

In this section, we will make a few assumptions to simplify the estimation problem and limit the scope of the literature study. The assignment text calls for closer investigation of one or two templates; we choose to take a closer look at Tilje. Although all the templates would be good candidates, we choose Tilje since (1) it produces a three-phase mixture of oil, gas and water, which calls for the use of multiphase modelling, and (2) under reasonable assumptions, the network topology is quite simple, which means we can present most of the estimation results in a reasonable amount of space. Although Tilje only has two wells, the template is equipped with two additional slots for future wells. To accommodate for the possibility of future wells being drilled, it makes sense to allow for an arbitrary number of wells in the model. This also enables us to apply our results to other templates at Skarv or other fields.

The actual production templates at Skarv are relatively complex networks. In addition to the equipment required for bringing reservoir fluids to the surface, they are equipped with a large number of pipes, valves and sensors to facilitate additional requirements, such as safety/shutdown functions and methanol injection for preventing hydrate formation. However, during normal production, this additional equipment is not in use; therefore we choose to ignore it. To prevent slugging in the flowline-riser system, the Tilje wells always produce to a single pipeline. Therefore, we further simplify the network and assume a single tieback pipeline, ending up with the simplified model shown in Figure 1.4 below. This figure also shows the (relevant) available instrumentation used for estimation.

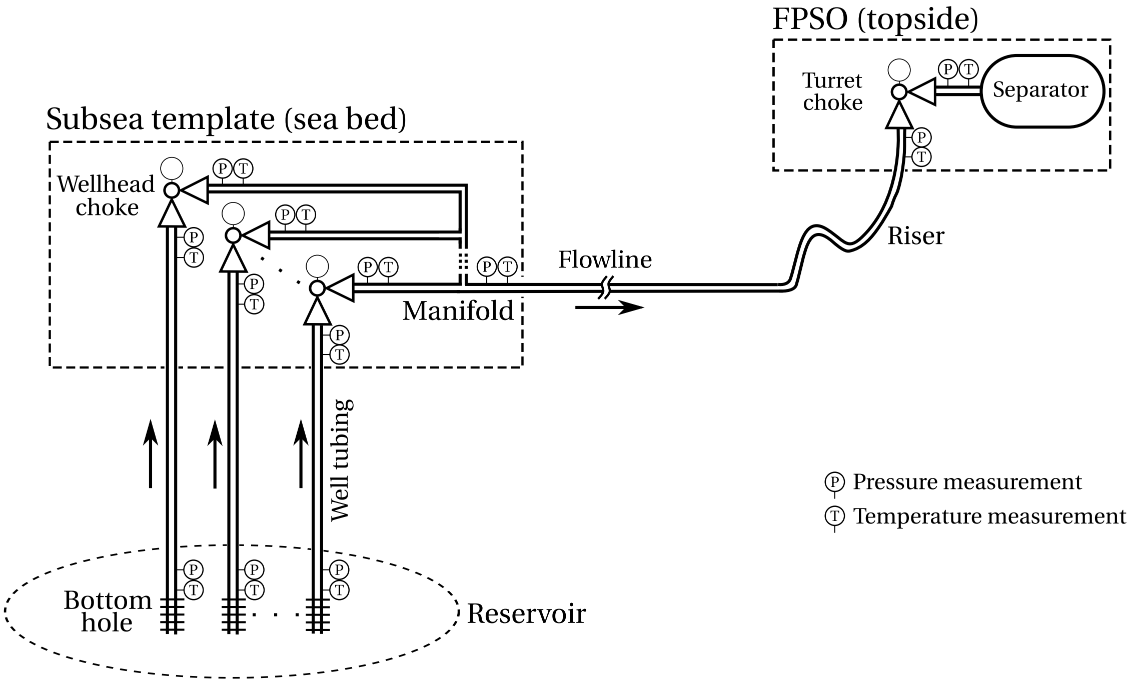


Figure 1.4: Simplified Tilje template with instrumentation.

We will also make the important assumption that reservoir pressure, GORs and water cuts are fixed between well tests. This means we can focus on formulating the model in terms of liquid rates.

Finally, we assume the wells are to be calibrated using single-rate tests. This is to reflect the current operational situation at Skarv. Nevertheless, in Chapter 6 we will discuss calibration in a more general setting, i.e. applicable to multi-rate tests.

1.3 Overall goals and strategy

Based on the assignment text, two overall goals can be identified. The first is to select/develop appropriate steady-state and dynamic models for flow estimation. The second is to implement static and dynamic estimation algorithms based on these models, and compare their performance with the aid of OLGA and some test cases. As indicated by the title of this thesis, our goal is to develop an estimator which produces good-quality flow rate estimates which could potentially be used for rate feedback control. Based on this, we can formulate a few requirements to be fulfilled by the estimation algorithm:

- Obviously, it should provide sufficiently accurate flow rate estimates. Since the flow rate controller has no choice but to treat the flow rate estimate as the truth, any offset in the flow rate estimate will result in a corresponding offset in the controlled rate.

- It should be robust, in the sense that it is not too sensitive to poor tuning choices, noise and "difficult" operational conditions.

- Estimates should be calculated in a reasonable amount of time after new measurements are available, i.e. well within the sampling time of the system. This is because the calculation time contributes directly to the time delay/lag in the feedback loop.

These are thus important criteria to be considered when comparing the two estimation algorithms. *Velocity* is also mentioned briefly in the assignment text, which is due to the fact that gas velocity and erosion is an operational concern at Skarv. Although we will focus on rate estimation throughout the thesis, some remarks on velocity are made in the discussion chapter.

As mentioned in the background section, it seems that most commercial VFM systems rely on coupling a black-box model/simulator with an optimization algorithm and NLP solver to generate estimates. This method is inherently inefficient, since (1) simulators may take some time to calculate pressures for a given set of flow rates, since the entire network model must converge for each evaluation, and (2) black-box simulators to not generally provide gradients, meaning gradients must be calculated by e.g. finite differences, or a derivative-free optimization algorithm must be used. In e.g. (Gunnerud & Foss, 2009) and (Sandnes, 2013), we see that solving the *production optimization* problem can be made more efficient by exploiting the structure of the problem and breaking down the simulator into a transparent model consisting of smaller network components, i.e. wells and pipelines. These components all have their own (smaller) model, which may be obtained by sampling the simulator for a sufficient range of the input variables and approximating the resulting data sets using some approximation scheme. For instance, (Kosmidis et al., 2005; Gunnerud & Foss, 2009) use piecewise linear approximations, while (Sandnes, 2013) uses B-spline (piecewise polynomial) approximations. The resulting approximated models are then connected with mass balance equations to form one large optimization problem.

These methods transfer nicely to the flow estimation problem, since the resulting optimization problem is very similar if we ignore the routing decisions sometimes present in production optimization. In fact, the only main differences are the objective function and the free variables; while the production optimization problem may seek to maximize flow rates by adjusting e.g. choke pressure drops or lift gas, the flow estimation problem seeks to minimize discrepancies between pressure estimates and pressure measurements by adjusting the flow rates. The basic model constraints (i.e. models and mass balance) are mostly the same.

This motivates the use of a similar approach for the flow estimation problem. Such an approach also supports the inclusion of in-house models, which eliminates the need for maintaining two separate models of the production system. Having worked with B-splines in my project assignment last semester (Robertson, 2013), this seems a natural choice for approximating BP's in-house models of the equipment on the Tilje template.

1.4 Scope and interpretation of assignment text

In this section, we will go through the assignment text item by item and give a short interpretation and a reference to where in the report the item is covered.

> 1. Literature review on state estimation methods and mathematical modelling of oil production networks. This should include advantages of using optimization based methods instead of the Kalman Filter.

This literature review assignment is covered in Chapters 2 (state estimation methods) and 3 (mathematical modelling). The state estimation part focuses on Kalman filtering and optimization-based estimation methods, and discusses some general concepts needed to formulate the optimization problems in Chapters 4 and 5. We also look at flow estimation (VFM) from a more practical viewpoint. The modelling part focuses on rate/pressure relations in wells, choke valves and pipelines, which are the essential building blocks of a VFM system.

> 2. Develop/select suitable steady-state and dynamic models for one or two specific templates with wells. Fit the model to one template at the Skarv field using available data and other information.

The Tilje template is selected for implementation of the flow estimation methods, due to its relatively simple topology and the fact that it produces a three-phase mixture of oil, gas and water. As mentioned in the background section, industry experience has shown that a VFM implementation preferably should support the inclusion of customer maintained models. Motivated by the work of (Sandnes, 2013), and the desire to introduce some novelty, the selected approach is to implement models of wells and pipelines in the Skarv field which are maintained by BP, but use B-spline approximations to form surrogate models as an alternative to the black-box approach which seems to be common in commercial VFM solutions. The static model is described in Chapter 4, and in Chapter 5 the static model is extended to include dynamics. Since the models are to be implemented as constraints in an optimization problem, these chapters are more "optimization problem formulation" chapters than modelling chapters *per se*. In Chapter 6 we fit the models to well test data using a fairly simple nonlinear least-sqares approach.

> 3. Implement a detailed model of the same system for comparison purposes. OLGA or similar tool to be used.

The detailed model is implemented in OLGA using well and pipeline geometries from Tilje, supplied by BP. A short description of the model is given in the beginning of Chapter 8.

> 4. Define a set of typical operating scenarios; some with predominantly stable operation and others with extensive variability.

Three operating scenarios are defined in Chapter 8; one well adjustment case with little to moderate dynamics, one riser slugging case with little to severe dynamics, and one case with field data, with little to moderate dynamics.

> 5. Develop, implement and test methods for optimization based estimation (using simulation).

This assignment spans both the modelling, implementation and results chapters. While the main part of this is the formulation of the optimization problems (Chapters 4 and 5), the actual implementation of the estimators is discussed briefly in Chapter 7, and the estimators are tested in Chapter 8.

> 6. Compare the performance of steady-state versus dynamic models for the operating scenarios defined in Item 4, and historical data from Skarv to assess the performance of the estimators.

A performance comparison of the steady-state and dynamic estimator is given objectively in the Chapter 8, while the discussion chapter (Chapter 9) will elaborate further on both advantages and shortcomings of each method. Here, we will emphasize on each method's ability to fulfill requirements given at the start of Section 1.3 to discuss whether the methods are feasible for use in a closed-loop rate control system.

1.5 Structure of the thesis

To summarize the section above, we will describe the contents of each chapter in this thesis. **Chapter 1** is this introduction, which covers background information about flow estimation and the Skarv field, motivation, overall goals and strategy. We then go on with some theoretical chapters; **Chapter 2** deals with state estimation, while **Chapter 3** covers the theory on modelling petroleum production networks. Thereafter, the models and optimization problems used for estimation are formulated in **Chapters 4** (static) and **5** (dynamic). Model fitting/calibration is discussed in **Chapter 6**. In **Chapter 7** we provide a short chapter on how the optimization problems are included in flow estimation algorithms, and briefly discuss the C++ implementation. In **Chapter 8**, the estimators are put to the test on two simulation cases and one case with data from the Skarv field. Finally, a discussion is given in **Chapter 9** and a conclusion in **Chapter 10**.

1.6 Notation

This section describes some common notation used throughout the thesis, summarized in Table 1.3 (the most common quantities, symbols and units) and Table 1.4 (general notation).

Quantity	**Symbol**	**Unit**
Volumetric flow rate	q	[Sm^3/h]
Pressure	p	[bara]
Differential pressure	Δp	[bar]
Temperature	T	[°C]
Density	ρ	[kg/m^3]
Density at standard conditions	ρ^{std}	[kg/m^3]
Viscosity	μ	[N/sm^2]

Table 1.3: Common quantities, symbols and units.

Type	**Description**
Scalars	For scalars and scalar-valued functions, the usual italic font is used, e.g. n, T for scalars and $f(\cdot)$ for a scalar-valued function.
Vectors	Vectors are denoted with bold face lower-case letters, e.g. $\mathbf{x}$. Vector-valued functions follow the same notation, e.g. $\mathbf{f}(\cdot)$.
Matrices	Matrices are denoted with bold face upper-case letters, e.g. $\mathbf{A}$. The transpose of $\mathbf{A}$ is denoted $\mathbf{A}^\top$, and the inverse of $\mathbf{A}$ is $\mathbf{A}^{-1}$. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. $\mathbf{0}$ and $\mathbf{1}$ are matrices filled with zeros and ones, respectively (the size of these should be clear from context).
Sets	Vector spaces are denoted with the so-called blackboard bold font, e.g. $\mathbb{X}$. $\mathbb{R}^n$ denotes the n-dimensional space of real numbers. $\mathbb{M}^{n\times m}$ denotes the space of $n \times m$ matrices. Index sets are denoted with the calligraphic font, e.g. $\mathcal{W}, \mathcal{H}$.
Norms	The weighted norm is used extensively, and for an n-vector $\mathbf{x}$ it is given as $\lVert\mathbf{x}\rVert_{\mathbf{P}}^2 = \mathbf{x}^\top \mathbf{P}\mathbf{x}$, where $\mathbf{P}$ is an $n \times n$ weighting matrix.
Sequences	$\{\mathbf{x}_k\}_{k=1}^T$ denotes the sequence $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$.
Variable bounds	Simple underline and overline notation is used to denote bounds on variables. For example, a pressure p may be bounded by a lower bound $\underline{p}$ and an upper bound $\overline{p}$.
B-spline approximations	The Greek letter Φ is used to emphasize when a B-spline approximation is used. For example, the B-spline approximation of the function $f_{\text{vlp}}(\cdot)$ is denoted $\Phi_{\text{vlp}}(\cdot)$.

Table 1.4: General notation.

Chapter 2

State estimation methods

The aim of this study is to estimate unknown flow rates when only pressure and temperature measurements are available. In this sense, we are presented with a *state estimation* problem. This chapter provides a literature review of selected state estimation methods. In the introductory section, the state estimation problem is introduced in the setting of a general discrete-time nonlinear system. In section 2.2, a description of the Kalman filter (KF) will be given, including a couple of extensions to nonlinear systems, namely the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). Sections 2.3 and 2.4 will present two optimization-based methods; static weighted least-squares (SWLS) and moving-horizon estimation (MHE), respectively. We then go on to discuss benefits of optimization-based methods in light of the task at hand, before we conclude the chapter with a more practical view on flow estimation in subsea production networks. The main sources of information have been (Brown & Hwang, 2012) for Kalman filtering, and (Rawlings & Mayne, 2013) for SWLS and MHE. Additional sources will be cited when appropriate.

2.1 Introduction

In most real-world systems, including subsea production networks, the full set of states/-variables cannot be measured directly. For instance, some variables may not be practically or economically feasible to measure due to their location in the system. Furthermore, the measurements which *are* available may be subject to unknown disturbances or noise, which means that we must expect there to be some error between the measured and true values. For example, an instrument may be calibrated incorrectly or its accuracy may deteriorate over time, or signal noise may occur due to e.g. electromagnetic interference. The *state estimation problem* amounts to (1) finding estimates for the unknown states, and (2) using information about measurement accuracy to estimate the true value of the measured variables. A *state estimator* is thus an algorithm designed to use available information, such as (potentially noisy) measurements of inputs/outputs and a mathematical model of the system, to generate estimates of the current internal state of the system. This is useful for two main reasons:

- *Supervision:* State estimators provide important information about unmeasured states in the system. This may be valuable to operators and engineers, e.g. for monitoring and/or decision support.
- *Control:* When estimates of unmeasured states are available, these estimates can be used for feedback and/or feedforward control.

This concept is illustrated in the familiar Figure 2.1, which shows the architecture of a state estimation based control system. Note that using the state estimate for control is not necessarily something we need; some applications may use the estimator for monitoring purposes only.

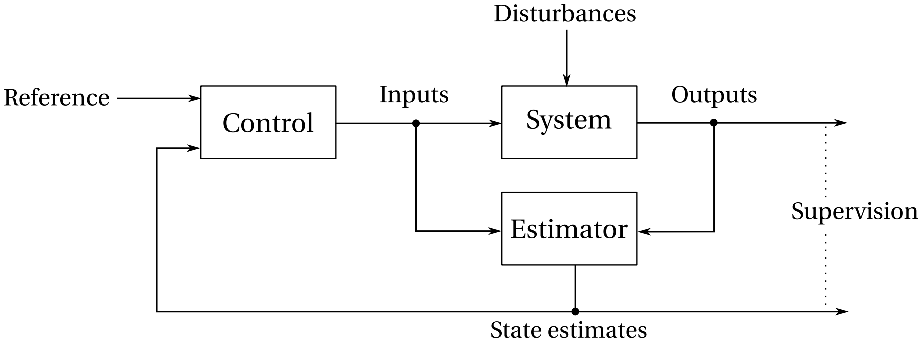


Figure 2.1: Control and supervision using state estimator.

Control engineering is all about designing the internal structures of the controller and estimator, and a vast amount of different methods have been developed over the years. As mentioned above, this chapter will focus on a few selected methods for state estimation. This chapter will be presented in a discrete-time setting, since both the Kalman filter and the moving-horizon estimator are usually treated in discrete time. General discrete-time nonlinear systems are given by the following:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{2.1a}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \tag{2.1b}$$

Assuming a constant sampling time Δt, the time index k is used to denote a discrete point in time, i.e. $\mathbf{x}_k = \mathbf{x}(t_k)$. $\mathbf{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$ is an n-vector of states, $\mathbf{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$ is an m-vector of known inputs, and $\mathbf{y}_k \in \mathbb{Y} \subseteq \mathbb{R}^q$ is a q-vector of measured outputs. $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{Y}$ are constraint sets for the state, input and output vectors, respectively, which usually contain bounds on each variable. Since our goal in this thesis is state estimation and not control, both $\mathbf{u}_k$ and $\mathbf{y}_k$ will be treated as *measurements*, which serve as inputs to the state estimation algorithm. $\mathbf{w}_k \in \mathbb{R}^n$ represents errors in the state equations and $\mathbf{v}_k \in \mathbb{R}^q$ represents errors in the output equations. These errors stem from noise, disturbances, and discrepancies between the model and the actual system. The system dynamics are described by the nonlinear function $\mathbf{f} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$. In (2.1a), $\mathbf{x}_{k+1}$ is given explicitly, however $\mathbf{f}(\cdot)$ may or may not be possible to write in closed form. If not, (2.1a) is given implicitly, and must be replaced with $\mathbf{0} = \mathbf{f}(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{u}_k) + \mathbf{w}_k$. The output vector is linked to the states by the nonlinear output function $\mathbf{h} : \mathbb{X} \to \mathbb{Y}$. Sometimes the input vector $\mathbf{u}_k$ is included as an input to $\mathbf{h}(\cdot)$, but this is omitted here due to the fact that the outputs of the system in question are simply a subset of the states.

2.1.1 Classification of state estimation methods

State estimation methods can be classified according to their properties. In the literature, a common divide is between *stochastic* and *deterministic* state estimation. Stochastic methods base their estimates on probability and statistics, and are sometimes called *maximum likelihood* estimation methods. Deterministic methods calculate estimates which somehow cause the observed measurements and deterministic model predictions to converge. For the methods discussed here, Kalman filtering is regarded a stochastic method, while SWLS and MHE are deterministic. However, it should be noted that in some special cases the two approaches are simply different ways of obtaining the same result. For instance, it is shown in (Rawlings & Mayne, 2013) that for linear systems, Kalman filtering is equivalent to solving a deterministic weighted least-squares problem, provided the weights are selected appropriately. Estimation methods may be further categorized by what assumptions are being made about the system (static/dynamic/linear/nonlinear/Gaussian noise etc.) and how much information is being considered when computing estimates (full-information/reduced information).

2.1.2 Observability

Observability is a key term when talking about state estimation. In short, an internal (unmeasured) system variable is said to be *observable* if it is mathematically possible to determine its value from observations of inputs and outputs only. In other words, if a variable is not observable, we cannot expect to be able to estimate its value. A *system* is said to be observable if all its internal states are observable. A relaxed term related to observability is *detectability*, which only requires unstable states to be observable. For simplification, model and process errors are most often not considered when checking for observability - it is an inherent property of the system equations. Determining whether a system is observable or not is simple for linear systems such as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{2.2a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \tag{2.2b}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are $n \times n$, $n \times m$ and $q \times n$ matrices, respectively. We simply check the rank of the *observability matrix* $\mathcal{O} = [\mathbf{C}, \mathbf{CA}, \mathbf{CA}^2, \ldots, \mathbf{CA}^{n-1}]^\top$; if it has full rank, the system is observable (Chen, 1995). For nonlinear systems like (2.1), observability is less trivial, but the topic has been quite thoroughly explored since the early 1970s (Hermann & Krener, 1977). Rawlings & Mayne (2013) give rigorous definitions of both observability and detectability for discrete-time nonlinear systems. Here, observability implies that the output sequence must originate from a unique initial state. In other words, if some disturbance sequence is given, and two different initial states may result in the same output sequence, the system is *not* observable (which is basically the same generic definition we gave at the start of this section). The relaxed detectability condition requires two converging disturbance and output sequences to yield converging state trajectories, and they show that this is satisfied for incrementally input/output-to-state stable (i-IOSS) systems.

In this study, the question of observability boils down to whether or not we can use the observed measurements of pressures and choke valve positions to determine the flow rates. At least two fundamental requirements are clear; For one, the flow network must have an adequate number of measurements. Furthermore, every set of measurements must yield a unique set of flow rates. If this is not the case, we cannot be certain of which of the possible solutions is the correct one.

2.2 Kalman filtering

The Kalman filter and its variations is a well established stochastic state estimation method developed in the 1960s by Rudolph Kalman (Kalman, 1960). Its applications range from navigation in the marine and aerospace industries to weather forecasting. In the oil industry, variations of the Kalman filter are used (among other things) for history matching of reservoir models (Sui et al., 2011) and for state/parameter estimation of oil and gas processing plants (Imsland et al., 2010). This section will give a short description of three variations, namely the standard linear Kalman filter, the extended Kalman filter and the unscented Kalman filter.

2.2.1 The Kalman filter (KF)

The original Kalman filter introduced in 1960 is well established to be the statistically optimal state estimator, given the assumptions that

1. $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ in (2.1) are known linear functions,

2. $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be white sequences with zero mean and known covariance structure[1], and

[1] When the noise is assumed coloured, this is accounted for by filtering the noise through a linear filter.

3. the state vector is unconstrained ($\mathbb{X} = \mathbb{R}^n$).

The system (2.1) thus becomes

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2.3a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2.3b}$$

where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{B}$ is an $n \times m$ matrix, and $\mathbf{C}$ is a $q \times n$ matrix. $\mathbf{Q} > 0$ is a positive definite matrix describing the covariance of the process noise, while $\mathbf{R} > 0$ similarly describes the covariance of the measurement noise. $\mathcal{N}(\mathbf{m}, \mathbf{P})$ denotes the normal (Gaussian) distribution with mean $\mathbf{m}$ and covariance $\mathbf{P}$. The Kalman filter models (2.3) as a *random process*, meaning that the state vector $\mathbf{x}_k$ is treated as a random variable. It is optimal in the sense that it minimizes the expected mean-square estimation error of the random variable $\mathbf{x}_k$, which for iteration k can be written as

$$e_k(\hat{\mathbf{x}}_k) = E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top (\mathbf{x}_k - \hat{\mathbf{x}}_k) | \mathbf{y}_k^*\right], \tag{2.4}$$

where $E[\cdot]$ denotes expectation, $\hat{\mathbf{x}}_k$ denotes the estimate of the true state $\mathbf{x}_k$, and $\mathbf{y}_k^* = \{\mathbf{y}_i\}_{i=0}^k$ indicates that this expectation is conditioned on the entire measurement stream. Like the classical Luenberger observer[2] from basic linear systems theory, the Kalman filter is a predictor-corrector algorithm in the sense that it first predicts a so-called *a priori* state estimate $\hat{\mathbf{x}}_k^-$ by using the state equation (2.3a), and corrects this estimate by using new measurements $\mathbf{y}_k$ (and $\mathbf{u}_k$):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-\right). \tag{2.5}$$

The updated estimate $\hat{\mathbf{x}}_k$ is called the *a posteriori* estimate. The only difference between the Luenberger observer and the Kalman filter is the choice of the weighting matrix $\mathbf{K}_k$[3]: While the Luenberger observer uses a constant matrix which places the poles of the error dynamics at desired positions, the Kalman filter updates the matrix on each iteration to keep the estimate optimal with respect to minimizing (2.4). To do this, the filter needs to keep track of the *error covariance* $\mathbf{P}_k$, that is, the (conditioned) expected value of the estimation error covariance:

$$\mathbf{P}_k = E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top | \mathbf{y}_k^*\right]. \tag{2.6}$$

In addition to the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, two parameters are required to start off the Kalman filter: We must make an initial guess on the mean and covariance of the state, which we denote $\mathbf{x}_0^-$ and $\mathbf{P}_0^-$, respectively. Figure 2.2 shows one iteration loop of the Kalman filter.

This can be accomplished by augmenting the model with additional states.

[2] see e.g. (Chen, 1995).

[3] For the Luenberger observer, the weighting matrix is usually referred to as the *gain matrix*, and denoted $\mathbf{L}$. The matrix used in the Kalman filter is usually referred to as the *Kalman gain matrix*.

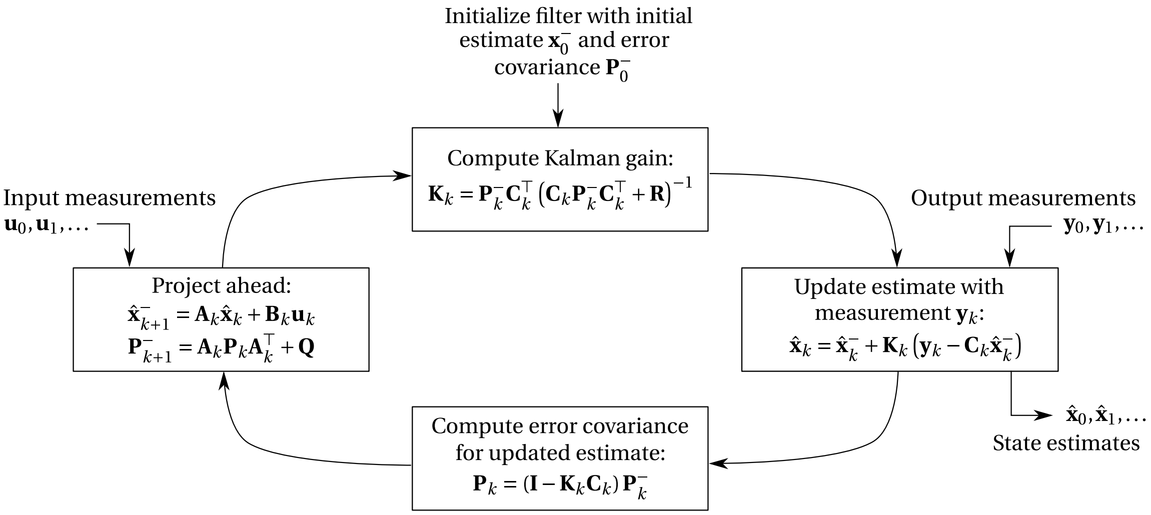

Initialize filter with initial estimate $\mathbf{x}_0^-$ and error covariance $\mathbf{P}_0^-$
Compute Kalman gain:
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^\top \left(\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^\top + \mathbf{R}\right)^{-1}$
Input measurements
$\mathbf{u}_0, \mathbf{u}_1, \ldots$
Output measurements
$\mathbf{y}_0, \mathbf{y}_1, \ldots$
Project ahead:
$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k$
$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^\top + \mathbf{Q}$
Update estimate with measurement $\mathbf{y}_k$:
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-\right)$
$\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots$
State estimates
Compute error covariance for updated estimate:
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^-$

Figure 2.2: Kalman filter loop (adopted from (Brown & Hwang, 2012)).

Here, the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are equipped with time indices, however, the simple linear Kalman filter assumes these to be constant matrices, i.e. $\mathbf{A}_k = \mathbf{A}$ for all k and likewise for $\mathbf{B}_k$ and $\mathbf{C}_k$.

Another way to look at this, which appears to be preferred in the literature, is the probabilistic view; minimizing (2.4) is equivalent to finding the state estimate which is most *likely* to be true, given the observed measurements. In other words, we want to maximize an approximated probability density function (PDF) of the state vector, which for a normal distribution occurs at the mean[4]. In this sense, we can interpret the state estimate as the solution obtained either by minimizing $e_k(\hat{\mathbf{x}}_k)$ from (2.4), or by maximizing the PDF of $\mathbf{x}_k$, which we denote $p_{\mathbf{x}_k}(\hat{\mathbf{x}}_k)$:

$$\hat{\mathbf{x}}_k = \arg\min_{\hat{\mathbf{x}}_k} e_k(\hat{\mathbf{x}}_k) = \arg\max_{\hat{\mathbf{x}}_k} p_{\mathbf{x}_k}(\hat{\mathbf{x}}_k) \tag{2.7}$$

At each iteration, the Kalman filter keeps track of an approximated PDF of the state. Since $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be normal, and all transformations are linear, the PDF of the state will also be normal, and can be completely characterized by its mean and covariance. Since we want to maximize a normal PDF, the estimate is simply taken as the mean. Hence, Kalman filtering boils down to propagating the mean and covariance of the state vector through the system equations in an optimal (minimum mean-square error) way with respect to the observed measurements (Kandepu et al., 2008). With respect to this probabilistic viewpoint, the *a posteriori* estimate $\hat{\mathbf{x}}_k$ in (2.5) is the propagated mean, while $\mathbf{P}_k$ in (2.6) is the propagated covariance.

[4] The multivariate PDF of the normally distributed random variable $\mathbf{x} \in \mathbb{R}^n$ with mean $\mathbf{m}$ and covariance $\mathbf{P}$ is given as $p_{\mathbf{x}}(\hat{\mathbf{x}}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\mathbf{P})}} \exp\left[-\frac{1}{2}(\hat{\mathbf{x}}-\mathbf{m})^\top \mathbf{P}^{-1}(\hat{\mathbf{x}}-\mathbf{m})\right]$, which is easily maximized by taking $\hat{\mathbf{x}} = \mathbf{m}$.

2.2.2 The extended Kalman filter (EKF)

Soon after the Kalman filter was introduced, extensions to handle nonlinear systems were proposed by (Kopp & Orford, 1963) and (Cox, 1964). The extended Kalman filter is a modified version of the Kalman filter which estimates the state of nonlinear systems. This is accomplished by linearizing the system at each time step and applying the standard (linear) Kalman filter equations to generate the state estimate. With reference to (2.1) and Figure 2.2, the matrices $\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{C}_k$ at each iteration are obtained by linearizing $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ around the most recent estimate of $\mathbf{x}$ and the current input:

$$\mathbf{A}_k := \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}_k,\mathbf{u}_k)}, \quad \mathbf{B}_k := \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{(\hat{\mathbf{x}}_k,\mathbf{u}_k)}, \quad \text{and} \quad \mathbf{C}_k := \left.\frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}_k^-,\mathbf{u}_k)}. \tag{2.8}$$

We could also say the system is linearized along the estimated state trajectory. Note that this requires $\mathscr{C}^1$ smoothness of $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$. The projection and update equations are replaced by updates which use the nonlinear functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ in (2.1) directly;

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k,\mathbf{u}_k), \qquad \text{(Project)} \tag{2.9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-,\mathbf{u}_k)\right). \qquad \text{(Update)} \tag{2.10}$$

The remaining equations are the same as for the standard Kalman filter, except that the system matrices are calculated using (2.8). From (2.9) and (2.10), we see that the state estimate (actually, the mean of the approximated state PDF) is propagated through the nonlinear system equations. However, the covariance is propagated through the linearized system matrices.

The EKF is established in the industry as perhaps the most widely used estimation algorithm for nonlinear systems; probably due to its relative simplicity and demonstrated efficiency in handling nonlinear systems (Haseltine & Rawlings, 2005). Despite its widespread use, the EKF has received a fair amount of criticism over the years (especially from the MHE research community), due to its failure to converge when it is not properly tuned and/or initialized, and inability to properly handle state constraints (see e.g. (Haseltine & Rawlings, 2005) and/or (Robertson et al., 1996)).

2.2.3 The unscented Kalman filter (UKF)

The unscented Kalman filter was first presented in (Julier & Uhlmann, 1997) as an improved estimator for nonlinear systems compared to the EKF. The UKF is based on the fact that a normally distributed random variable will no longer be normally distributed after being propagated through a nonlinear function. The authors argue that

> Although the EKF (in its many forms) is a widely used filtering strategy, over thirty years of experience with it has led to a general consensus within the tracking and control community that it is difficult to implement, difficult to

tune, and only reliable for systems which are almost linear on the time scale of the update intervals.

In a sense, the UKF can be considered a derivative-free alternative to the EKF. While the EKF uses the Jacobians of the system matrices (cf. Eqn. (2.8)), the UKF avoids this by sampling the approximated state PDF at several points, which are denoted the *sigma points*, and propagating each point individually through the nonlinear system equations. This way, the UKF is able to maintain a *nonlinear* approximation of the state PDF, which in most cases is closer to reality than the normal distributions assumed by the EKF (Brown & Hwang, 2012). The literature is full of EKF/UKF comparative studies, and the general consensus seems to be that the UKF and EKF perform similarly for mildly nonlinear systems, but that the UKF outperforms the EKF when the system dynamics are highly nonlinear.

2.3 Static weighted least-squares estimation

Static weighted least-squares estimation is an optimization-based deterministic approach to state estimation. Optimization-based state estimators choose estimates which minimize deviations between predicted and observed measurements, i.e. solving a model-constrained optimization problem. This is also known as *data reconciliation* (Narasimhan & Jordache, 1999). This approach to state estimation has had great success when applied to large-scale systems like chemical plants (Dempf & List, 1998) and electrical power grids (Huang et al., 2012). It has also been used for flow estimation in petroleum production networks (VFM), for instance in FMC Technologies' FlowManager™(Holmås & Løvli, 2011). Consider the static, or steady-state, version of (2.1):

$$\mathbf{0} = \mathbf{f}^{\infty}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \tag{2.11a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{2.11b}$$

The function $\mathbf{f}^{\infty} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ represents the steady-state (long-term) response of the system, i.e. as $t \to \infty$. Recall that $\mathbf{w}$ represents model errors, while $\mathbf{v}$ represents measurement errors in the sense that it contains the difference between observed ($\mathbf{y}$) and estimated ($\mathbf{h}(\mathbf{x})$) measurements. Traditionally, data reconcilation methods have assumed that the process model is exact, i.e. $\mathbf{w} = \mathbf{0}$. This may be the result of choice, or that the model is a treated as a "black box", i.e. (2.11a) is not available to the solver used in the optimization algorithm. The optimization problem is thus formulated in terms of minimizing $\mathbf{v}$ in a least-squares sense, subject to model equations and bounds. However, as Maquin et al. (2000) point out, the assumption of a perfect process model is quite naive, and it makes sense to factor in model uncertainties in the optimization problem. Therefore, we seek to find the state estimate which minimizes both the model errors and the measurement errors. This leads to the following nonlinear constrained least-squares problem:

Problem 2.1: Static Weighted Least-Squares (SWLS)

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{w},\mathbf{v}}{\text{minimize}}\ f(\mathbf{w},\mathbf{v}) \quad & = \|\mathbf{w}\|^2_{\mathbf{Q}^{-1}} + \|\mathbf{v}\|^2_{\mathbf{R}^{-1}} \\
\text{subject to} \quad & \mathbf{0} = \mathbf{f}^{\infty}(\mathbf{x},\mathbf{u}) + \mathbf{w} \\
& \mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \\
& \mathbf{x} \in \mathbb{X}
\end{aligned}
$$

$\mathbf{Q}^{-1}$ is a diagonal weighting matrix for the model errors, while $\mathbf{R}^{-1}$ is a diagonal weighting matrix for the measurement errors. The elements of $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ thus represent our relative confidence in the process model equations and the measurements, respectively. Weighting in terms of inverted matrices is common, since errors are often expressed in terms of error variance, i.e. a small error variance gives a large weight (cf. the Kalman filter). The least-squares formulation with quadratic cost is popular, since it is simple at the same time as large errors are penalized more than small ones. Since Problem 2.1 is a nonlinear program, an NLP solver is required to find a solution, e.g. a Sequential Quadratic Programming (SQP) or Interior Point (IP) method. The solver used in this thesis, IPOPT (Wächter & Biegler, 2006), is an example of the latter. Note that the nonlinear equality constraints make Problem (2.1) non-convex, which means it may have an unknown number of suboptimal local minima. Given an optimal solution $(\mathbf{x}^*,\mathbf{w}^*,\mathbf{v}^*)$, the state estimate is taken as $\mathbf{x}^*$, while the elements of $\mathbf{y}^* = \mathbf{h}(\mathbf{x}^*) + \mathbf{v}^*$ are called the *reconciled measurements.*

Some SWLS formulations may include parameter estimation. However, in this thesis it is assumed that the relevant parameters do not change significantly over the time frame between well tests, and they are therefore treated as constants. Parameters are estimated in a separate calibration problem; see Chapter 6 about model calibration.

2.4 Moving-horizon estimation

While SWLS might yield fairly good state estimates, it does not consider the dynamics of the system. Therefore, it cannot be expected to perform well during transient behaviour. One way to include dynamics is to exchange the static model (2.11) with the dynamic model (2.1), and solve an optimization problem which considers all the available measurements. This is called the full-information problem. However, as time goes by, the problem grows and quickly becomes computationally intractable. Moving-Horizon Estimation (MHE) is a scheme which solves this problem by only considering the N most recent measurements. MHE is considered "dual" to model predictive control (MPC), the same way the Kalman filter is considered dual to the linear quadratic regulator (LQR) (Rawlings & Mayne, 2013)[5]. The basic operation of MHE and its close relation to MPC is shown in Table 2.1.

Step	MHE	MPC
1	Find an optimal state trajectory by solving an optimization problem constrained by the dynamic model equations (2.1), and available *past* measurements and inputs.	Find an optimal *future* input trajectory by solving an optimization problem constrained by the dynamic model equations (2.1), and available measurements.
2	Take the *last* state vector in the optimal sequence as the state estimate.	Take the *first* input vector in the sequence as the plant input.
3	Wait for new measurements and shift the time horizon one step forward. Repeat from step 1.	Wait for new measurements and shift the time horizon one step forward. Repeat from step 1.

Table 2.1: Operation of MHE and MPC.

The optimization problem to be solved at time index T is the following:

Problem 2.2: Moving-Horizon Estimation

$$\begin{aligned} \underset{\mathbf{x}_{T-N},\mathbf{w},\mathbf{v}}{\text{minimize}}\ f(\mathbf{x}_{T-N},\mathbf{w},\mathbf{v}) \quad & = \Gamma_{T-N}(\mathbf{x}_{T-N}) + \sum_{k=T-N}^{T-1} L_{\mathbf{w}}(\mathbf{w}_k) + \sum_{k=T-N}^{T} L_{\mathbf{v}}(\mathbf{v}_k) \\ \text{subject to} \quad \mathbf{x}_{k+1} & = \mathbf{f}(\mathbf{x}_k,\mathbf{u}_k) + \mathbf{w}_k, \qquad k = T-N,\dots,T-1 \\ \mathbf{y}_k & = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \qquad k = T-N,\dots,T \\ \mathbf{x}_k & \in \mathbb{X}, \qquad k = T-N,\dots,T \end{aligned}$$

where $\mathbf{w} = [\mathbf{w}_{T-N},\dots,\mathbf{w}_{T-1}]^\top$ and $\mathbf{v} = [\mathbf{v}_{T-N},\dots,\mathbf{v}_T]^\top$. Note that we only optimize the state vector at the start of the time horizon - $\mathbf{x}_{T-N}$ - since the remaining state trajectory is given by the state equations together with model error trajectory $\mathbf{w}$ and (measured) input sequence $\mathbf{u} = [\mathbf{u}_{T-N},\dots,\mathbf{u}_{T-1}]^\top$. $\Gamma_{T-N}(\cdot)$ is called the *arrival cost*, while $L_{\mathbf{w}}(\cdot), L_{\mathbf{w}}(\cdot)$ are the *stage costs*. Like Problem 2.1, Problem 2.2 must be solved using an NLP solver.

The stage costs are typically selected as quadratic least-squares terms, i.e. $L_{\mathbf{w}}(\mathbf{w}_k) = \|\mathbf{w}_k\|^2_{\mathbf{Q}^{-1}}$ and $L_{\mathbf{v}}(\mathbf{v}_k) = \|\mathbf{v}_k\|^2_{\mathbf{R}^{-1}}$, similar to the objective function in Problem 2.1. However, Rawlings & Mayne (2013) point out that the stage costs must reflect the disturbances affecting the system - systems subjected to slowly decaying disturbances may

[5] Rawlings & Mayne (2013) show that MHE can be formulated as "optimal control of estimation error". Consequently, many of the techniques developed for MPC can be transferred to MHE.

require "stronger" than quadratic stage costs in order for the sums in the objective function to converge. In any case, the stage costs should be positive definite.

The arrival cost is included to transfer the information from past measurements before the estimation window into the objective function. This is comparable to the way the Kalman filter stores information about previous estimates in the covariance matrix, and is analogous to the *terminal cost* in MPC. Selecting an arrival cost which captures all the past information in an optimal way with respect to the system constraints is difficult for nonlinear systems, and quite a few results have been presented which describe how this term affects estimator stability and robustness. Practical MHE methods need to use some approximation of the arrival cost (Johansen, 2011). One way of doing this is coupling the MHE with some variant of the Kalman filter and approximating the arrival cost as

$$\Gamma_{T-N}(\mathbf{x}_{T-N}) = \|\mathbf{x}_{T-N} - \hat{\mathbf{x}}_{T-N}\|^2_{\mathbf{P}_{T-N}}, \tag{2.14}$$

where $\hat{\mathbf{x}}_{T-N}$ and $\mathbf{P}_{T-N}$ are the state estimate and covariance matrix maintained by e.g. an EKF or UKF (Rao et al., 2001). Another, more *ad hoc* approach is to assume that our previous estimate of $\mathbf{x}_{T-N}$ is good enough; we simply use the first state estimate from the previous time index, $\mathbf{x}_{T-N-1}$, and compute an *a priori* estimate $\mathbf{x}^-_{T-N}$ using the nonlinear system equations:

$$\mathbf{x}^-_{T-N} = \mathbf{f}(\mathbf{x}_{T-N-1}, \mathbf{u}_{T-N-1}) \tag{2.15}$$

Then, we can use this *a priori* estimate to approximate the arrival cost as

$$\Gamma_{T-N}(\mathbf{x}_{T-N}) = \mu \left\|\mathbf{x}_{T-N} - \mathbf{x}^-_{T-N}\right\|^2, \tag{2.16}$$

where μ is a scalar (Alessandri et al., 2008). We could also use a weighing matrix to give each state its own weight. Such a term (2.16) is often called a *regularization* term, and helps in ensuring estimator robustness and graceful performance degradation in the absence of an exciting system input (Johansen, 2011). Taking $\Gamma_{T-N}(\cdot) = 0$ (zero prior weighing) completely discards all previous information, and such a choice usually renders the MHE unable to estimate the state trajectory unless there is sufficient excitation in the system.

Note that for $T < N$, Problem 2.2 is not well-defined, since the number of available measurements is less than the size of estimation horizon. In this case, it is common to just truncate the horizon and use the measurements which are available (which is equivalent to solving the full-information problem). In this thesis, we will not dwell on this. In stead, we assume that $T \geq N$, i.e. that the estimation horizon is full.

2.5 Advantages of optimization-based estimators

Like all other things, optimization-based estimation has its pros and cons. In this section, we discuss a few of these in light of the task at hand. As mentioned above, the EKF is established in the industry as perhaps the most widely used estimation algorithm for nonlinear systems. This is also the case for industrial MPC applications which (as of 2004) use either a variant of the EKF or some *ad hoc* output correction term for state estimation (Qin & Badgwell, 2003). However, a shift towards MHE has been predicted (Moriari & Lee, 1999), and there exists current commercial MPC packages which employ MHE estimation techniques, e.g. Cybernetica's CENIT (Cybernetica, 2014). Some advantages of optimization-based estimators are:

Direct use of nonlinear equations: While the nonlinear variants of the Kalman filter use some approximation of the nonlinear system (2.1), optimization-based estimators use the nonlinear equations *directly* by imposing them as equality constraints. As we will see in Chapter 3, physical relationships in petroleum production networks are generally modelled as implicit nonlinear functions, which fits nicely into an optimization-based scheme with equality constraints.

Constraint handling: An optimization-based formulation enables us to impose explicit constraints on the state vector. This allows us to include information such as nonnegative pressures and flow rates and other bounds, which the Kalman filter cannot handle as elegantly[6].

Increased estimation accuracy and robustness: The direct use of the nonlinear equations and effective constraint handling means that optimization-based estimators will in most cases perform petter than the Kalman filter. In addition, comparative studies show that MHE is less fragile than the Kalman filter with respect to poor tuning and guesses of the initial state (Haseltine & Rawlings, 2005).

Increased computational abilities: While the Kalman filter is computationally efficient in the sense that it computes estimates recursively, MHE requires the solution to a difficult optimization problem. Thus, MHE has previously not been possible to implement online. However, with the advances in hardware and algorithms seen in the last few decades, MHE has become feasible as an online replacement to the Kalman filter.

[6] Constraint handling in the Kalman filter has been investigated, and solutions include projecting an unconstrained solution onto the state constraint surface (Simon & Chia, 2002) and *ad hoc* clipping strategies, i.e. using truncated PDFs (Haseltine & Rawlings, 2005).

2.6 A practical view on flow estimation

In the preceding sections, we have talked about state estimation in a rather generic setting. In this section, we will look at the flow estimation problem from a more practical viewpoint, i.e. we will discuss some basic concepts in virtual flow metering. As we will see in the next chapter, typical models of petroleum production networks include linear mass balance equations and nonlinear equations relating flow rates and differential pressures. An intuitive rule of thumb is that for each measured pressure drop with an associated pressure drop-flow rate model, we can estimate the flow rate of one phase. Thus, if we have more measurements, we can estimate more flow rates. By examining the placement and number of measurements, we can systematically determine how many flow rates we will be able to estimate (assuming that all the appropriate pressure drop/rate equations are available). This is discussed in relation to chemical processing plants in e.g. (Narasimhan & Jordache, 1999). Typically, we want to determine three flow rates (oil, gas and water rates) for each well and pipeline in the production system. Based on the number of available measurements, we may find ourselves in one of three situations;

1. We do not have enough measurements (and associated equations) to determine the desired number of flow rates. In this sense, we may say that some of the flow rates are *unobservable* (cf. Section 2.1.2). In this case, we may increase the number of equations by making some additional assumptions, such as fixed gas-oil ratios/water cuts.

2. We have exactly enough measurements to determine the desired number of flow rates. While we may be pleased with this situation, a sensor failure will immediately send us back to the situation above.

3. We have more than enough measurements to determine the flow rate. In this case, some of the measurements are *redundant*. While this would typically result in an overdetermined system without an exact solution, this is not a big concern, since we seek a best-fit solution rather than an exact one (Melbø et al., 2003).

These three situations are analogous to (respectively) underdetermined, consistent and overdetermined systems of equations. While the question of existence and uniqueness of solutions is not trivial for nonlinear systems like ours, the rule of thumb stated above is in general a good indication.

The Tilje template shown in Figure 1.4 is well equipped with measurements. Assuming they all work properly, we are clearly in Situation 3 above. For instance, we could estimate the flow rates in the following way;

- Use the bottom hole pressure (or rather, the differential pressure between the reservoir and the bottom hole) to estimate the oil rate in each well.

- Use the differential pressure between the bottom hole and the wellhead to estimate the gas rate in each well.
- Use the differential pressure over the wellhead choke to estimate the liquid rate in each well, and take the water rate as the difference between the liquid rate and the oil rate estimated above.
- Use mass balance equations to estimate the flow rates in the pipeline as sums of the flow rates from each well.

Note that we have not used all the available pressure drops here, for instance we have pressure drops in the pipeline and over the turret choke which also could be used to obtain an overdetermined system. However, their inclusion is not necessary, and in this sense they can be considered redundant. This is a fairly robust situation, for instance, let us assume that all of the sensors in one well fail. We can still estimate two of the flow rates from the well via the mass balance equations, since they can be taken as the difference between the flow rates estimated by the pipeline and turret choke models and the flow rates from the remaining wells. The last remaining rate can then be estimated by e.g. assuming a fixed gas-oil ratio or water cut for the sensorless well. If we assumed the gas-oil ratios and water cuts were fixed for *all* the wells, a minimal virtual flow metering system would require only one or two sensors per well. Although this is not an ideal situation, it is not far-fetched; subsea measurements have a tendency to fail over time, and repairing them is time-consuming and very expensive. In fact, the Skarv field is equipped with quite a few redundant sensors, i.e. two or more sensors at the same location, in case a sensor should fail at some point. The point of this discussion is merely to illustrate the flexibility and robustness of virtual flow metering systems, which further motivates their use in subsea production systems.

Although Tilje is well equipped with measurements, we will still assume fixed gas-oil ratios for the wells, since well tests have showed that these do not vary much between well tests. This will result in a robust estimator which will still work even if quite a few measurements should fail. We will not handle such a situation directly in this thesis, but rather assume that the measurements are functioning correctly.

Chapter 3

Modelling of petroleum production networks

Perhaps the most critical component of any model-based state estimation algorithm is the mathematical model of the system. Mathematical modelling of petroleum production networks requires knowledge of the physical relationships present in such networks. Thus, an important part of the literature study has been to gain some proficiency in how different elements of the network are modelled, and how to piece these elements together using mass conservation laws. This chapter will attempt to summarize a few key concepts, without diving into too much detail. A valuable reference in this chapter has been (Beggs, 2003). After a short introduction, some important aspects of multiphase flow are discussed. The following sections present methods for well modelling, choke modelling and pipeline modelling, before the chapter is concluded with a short section on mass balance equations.

3.1 Introduction

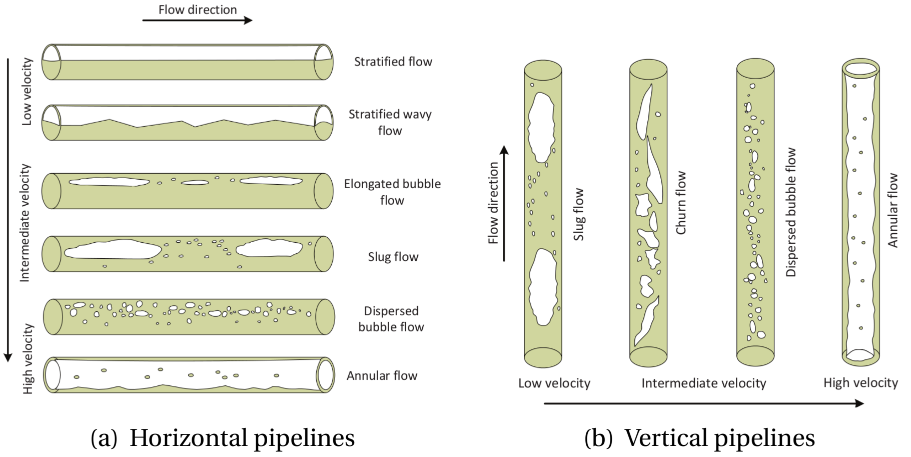
A traditional way of representing petroleum production networks is to use a graph representation with vertices and interconnecting edges. This view is also found in previous work done within the IO center[1], e.g (Sandnes, 2013), and an ongoing case study by co-supervisor Bjarne Grimstad (Grimstad et al., 2014), also in cooperation with BP. In such a representation, vertices are associated with pressures, while edges are associated with flows and pressure drops, such as the ones found in pipelines and choke valves. The source vertices represent points at which fluids flow into the network, e.g. production wells. The sink vertices represent points at which fluids leave the network, which could be a separator or export pipeline. The source and sink vertices usually provide the model with some boundary conditions, such as fixed reservoir and separator pres-

[1] Center for Integrated Operations in the Petroleum Industry

sures. Figure 4.1 shows the graph representation of the specific flow network studied in this thesis. As previously mentioned, the Skarv field is well equipped with pressure and temperature measurements. To fully utilize this available information, vertices are placed at points in which measurements are available (cf. Fig. 1.4), namely the bottom holes, wellheads, downstream the wellhead chokes, the manifold, upstream the turret choke and the separator. This choice of vertex placement means the edges will represent the well tubings, wellhead chokes, manifold jumpers, pipeline (flowline + riser) and the turret choke. For this reason, these are the components which will be focused on in this chapter. First, however, it is necessary to introduce a few terms related to multiphase flow which will be used in the remainder of the thesis.

3.2 Multiphase flow

The term *multiphase flow* refers to situations where two or more phases/components flow together in a pipeline. The flow may be (assumed) *homogenous*, which means the phases are perfectly mixed and flow with the same velocity, or *separated*, which means the phases are treated separately and may have different velocities. In our case, we have three phases, namely oil, gas and water. This is actually a simplification, since flows in petroleum production pipelines usually contain a large number of components: All the light, gaseous hydrocarbon components are treated as gas, while heavier components are treated as condensate or oil. In addition to hydrocarbons and water, the flows may contain sand, wax and other materials, but these are ignored in this thesis. Rigorous modelling of multiphase flow systems is complicated. The main reason for this is that the flow will behave quite differently for different pressure and temperature conditions. The behaviour of the flow, or *flow regime*, depends largely on the amount of gas present in a section of a pipeline compared to the amount of liquid, and the gas and liquid velocities (see Figure 3.1).



(a) Horizontal pipelines (b) Vertical pipelines

Figure 3.1: Multiphase flow patterns in pipelines (Jahanshahi, 2013).

3.2.1 Phase split and standard/in-situ conditions

In this thesis, the term *phase split* refers to the relative quantities of oil, gas and water at a given point in the flow network. The properties of hydrocarbon fluids vary with pressure and temperature. Consequently, the phase split in a pipeline also will vary with pressure and temperature. For instance, when hydrocarbons initially flow into the well tubing at high pressure and temperature, a substantial amount of gas may be dissolved in the oil. However, when the pressure and temperature decreases as the flow ascends through the network, some gas will bubble out of the oil, resulting in a larger relative gas content. This phenomenon is known as *flashing*. The properties of water are less dependent of pressure and temperature, so it is not uncommon to make the simplifying assumption that the water content does not change. This is illustrated in Figure 3.2, which shows (qualitatively) the phase splits for three points in the flow network with different pressure/temperature conditions.

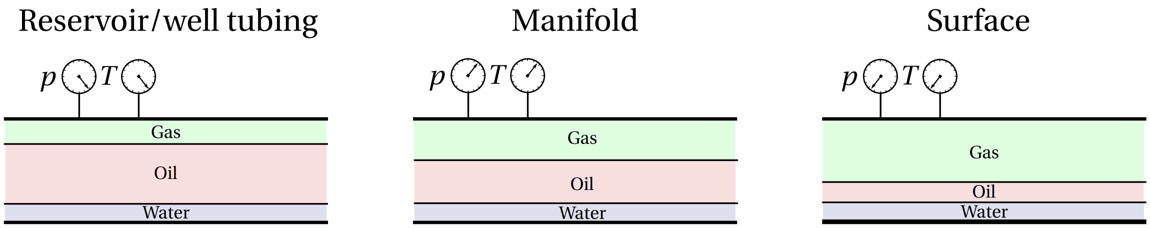


Figure 3.2: Effect of decreasing pressure and temperature on phase split in a pipeline.

As the flow nears the surface, more and more gas bubbles out of the oil, and the phase split changes. When we quantify flow rates and fluid properties at a point in the flow network, we may use *in-situ*[2] or *standard* conditions. The *in-situ* conditions describe the actual conditions at the point we are investigating, while standard conditions describe what the conditions would be if we took the same sample to the surface. In the oil industry, it is common practice to express quantities of oil and gas throughout the network using standard conditions, which is reasonable since this quantifies how much oil and gas will flow into the processing plant on the surface. However, when modelling, it may be necessary to know what the *in-situ* conditions are to calculate e.g. pressure drops. In this thesis, the standard conditions are defined as 15 °C (or 288.15 °K) and one atmosphere of pressure (1 bara or 100 kPa), which conforms to the standard issued by the Society of Petroleum Engineers (SPE) in 1982 (SPE, 1982).

[2] *In situ* is a Latin phrase which translates to *in position* or *in place*.

Gas-oil ratio and water cut

The *gas-oil ratio* (GOR) describes the volumetric ratio between gas flow and oil flow at standard conditions. In this thesis, the GOR will be denoted r_{go}, and is given by

$$r_{go} = \frac{q_g}{q_o}, \qquad [\mathrm{Sm}^3/\mathrm{Sm}^3] \tag{3.1}$$

where q_g is the gas volumetric flow and q_o is the oil volumetric flow, both at standard conditions. In this thesis, we will use the unit [Sm^3/Sm^3] for the GOR[3]. The *water cut* (WC) is defined as the ratio between the amount of produced water and the total amount of produced liquid. The WC is thus a dimensionless number between 0 (no water) and 1 (no oil)[4]. We denote the water cut as r_{wc}, and it is given by

$$r_{wc} = \frac{q_w}{q_l} = \frac{q_w}{q_w + q_o}, \qquad [\text{-}] \tag{3.2}$$

where q_w is the volumetic flow rate of water, q_l is the volumetric flow rate of liquid, and q_o is the volumetric flow rate of oil. Although the GOR and WC for a given well vary over time, they are usually assumed constant on a short-term horizon.

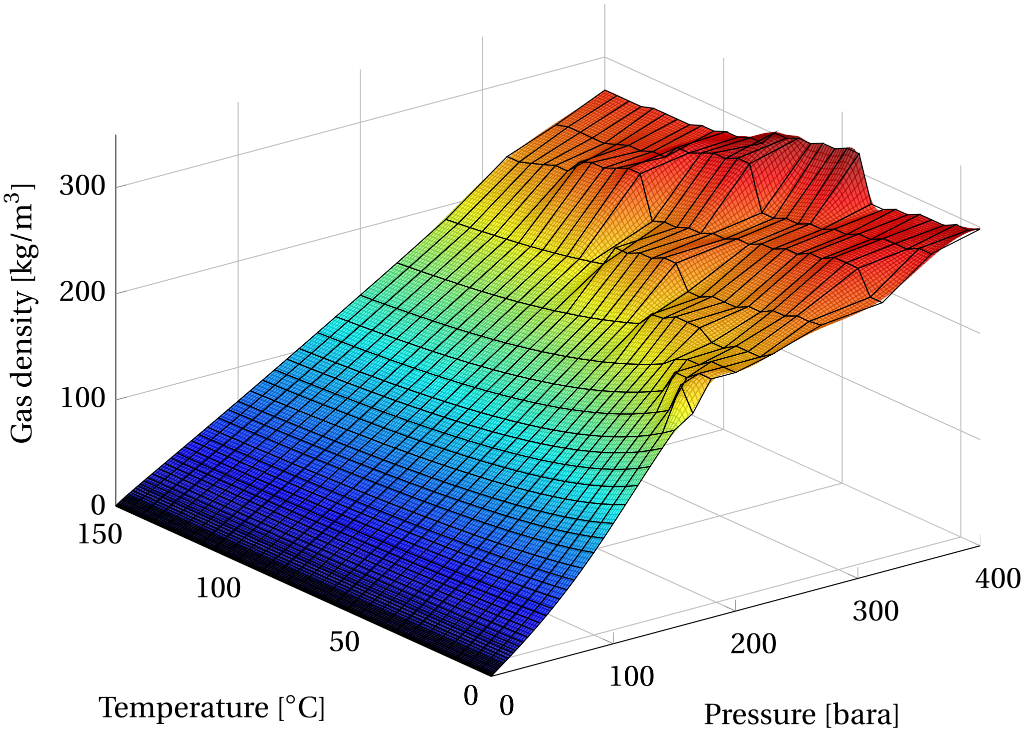
3.2.2 Fluid properties and PVT methods

For the hydrocarbon mixture present in the pipelines, fluid properties such as density, viscosity and gas mass fraction are highly nonlinear functions of pressure and temperature. Consequently, knowledge of how these properties behave is very important when modelling multiphase flows. Thermodynamic simulation software such as PVTsim from Calsep[5] are often used to generate property tables for a given reservoir fluid, and this is also the case for the Skarv field. By using B-spline approximations, we can approximate the tabular data from PVTsim as twice continuously differentiable piecewise polynomial functions, which is very convenient in an optimization framework (Sandnes, 2013; Robertson, 2013). A couple of examples are shown in Figure 3.3. The fluid properties used in this thesis are oil, gas and water densities, oil and water viscosities, and the gas mass fraction.

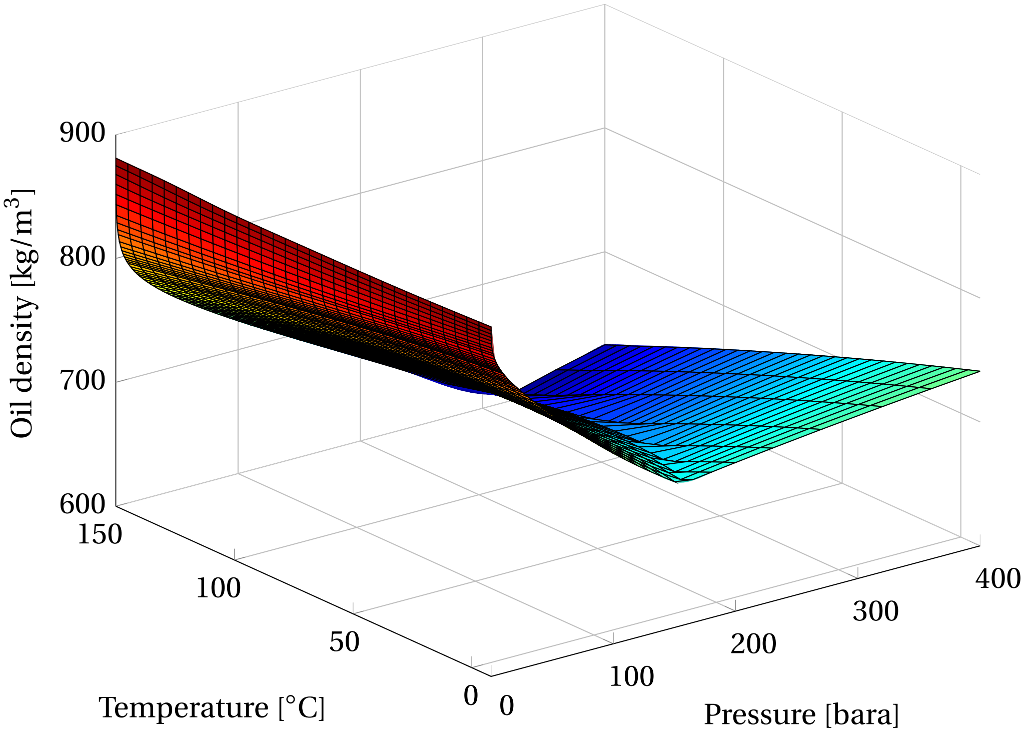
[3] Although this actually becomes a dimensionless number, this is not always the case. For instance, the GOR is expressed in standard cubic feet of gas per oil barrel [scf/sbbl] in US field units.

[4] In the oil industry, it is common to express the water cut as a percentage, i.e. $100 \cdot r_{wc}$.

[5] More information about PVTsim can be found on Calsep web pages (http://www.calsep.com/).



(a) Gas density, ρ_g



(b) Oil density, ρ_o. The reduction in density with increased pressure is due to light hydrocarbon components (gas) dissolving in the oil. The "kink" occurs when the oil is saturated with gas.

Figure 3.3: Examples of B-spline approximated fluid properties. The black grid shows the tabular data obtained from the PVT table, while the surface plot shows the spline approximation.

Gas mass fraction

The gas mass fraction is useful, since it enables us to calculate in-situ individual mass flow rates for each phase. The gas mass fraction α_g is defined as the ratio between in-situ gas mass flow and the total hydrocarbon mass flow:

$$\alpha_g = \frac{w_g^{\text{is}}}{w_g^{\text{is}} + w_o^{\text{is}}}, \qquad \text{[-]} \qquad (3.3)$$

where $w_g^{\text{is}}, w_o^{\text{is}}$ are the in situ mass flow rates of gas and oil, respectively. In this thesis, the gas mass fraction is used along with the fluid densities when calculating the pressure drop through a choke valve, as we will see in Section 3.4.

3.3 Well performance

3.3.1 Inflow from reservoir

The flow rates in the production network will ultimately depend on the amount of fluid entering the network. Hence, a critical part of the model is the *inflow performance*, which describes the flow from the reservoir into a well. This flow depends on a number of factors, such as rock properties, fluid properties, flow regime, fluid saturations in the rock, formation damage, gas and water content, et cetera (Beggs, 2003). In addition, the well completion and any inflow control devices will clearly affect the flow. The driving force for the flow is the *drawdown*, i.e. the differential pressure between the reservoir and the bottom hole. Although reservoir pressure decreases gradually as the reservoir is produced, it is reasonable to assume the pressure will remain constant on the short-term horizon (Gunnerud & Foss, 2009). Therefore, it is common to treat the reservoir pressure as a constant and express the inflow rate q as a function of the flowing bottom hole pressure p^{bhf} only, i.e.

$$q = f_{\text{ipr}}(p^{bhf}). \qquad (3.4)$$

Such a function $f_{\text{ipr}}(\cdot)$ is called an *inflow performance relationship* (IPR) . q may be an oil, gas, water or liquid rate, depending on the IPR and how it is matched to field measurements. Quite a number of different IPRs have been developed for different types of wells, however, they are in general based on Darcy's law $q = -\frac{kA}{\mu}\frac{\mathrm{d}p}{\mathrm{d}x}$. Darcy's law describes the flow of a fluid through a porous medium (such as the rock in a petroleum reservoir), and states that the volumetric flow rate q is proportional to the permeability[6] k of the medium, the cross-sectional area A, and the spatial pressure gradient $\mathrm{d}p/\mathrm{d}x$, and inversely proportional to the fluid viscosity μ.

[6] *Permeability* is a measure of how easily fluids can flow through the porous rock. Its unit of measure is the Darcy (D), which is roughly equivalent to 10^{-12} m^2.

For single-phase wells, the IPR is usually assumed linear, and can be described by a single *productivity index* (PI)[7]. In such a case, the inflow is simply assumed proportional to the drawdown, and the IPR becomes $f_{\mathrm{ipr}}(p^{bhf}) = \mathrm{PI} \cdot (p^r - p^{bhf})$, where PI is the (constant) productivity index. However, for multiphase inflow, the IPR is generally assumed nonlinear, and the productivity index is not constant. This is due to flashing of gas as the pressure decreases toward the well, i.e. the relative gas content increases as the fluids near the well. An example of an IPR curve is shown in Figure 3.4(a). A well-known IPR for multiphase wells is Vogel's IPR (Vogel, 1968), which is given by

$$\frac{q}{q_{\max}} = 1 - 0.2\frac{p^{bhf}}{p^r} - 0.8\left(\frac{p^{bhf}}{p^r}\right)^2, \tag{3.5}$$

where $q_{\max}$ is the maximum (theoretical) flow rate, i.e. the flow rate that would result from a flowing bottom hole pressure of zero. Another example of an IPR is the purely empirical Fetkovich method (sometimes referred to as the C&n method or backpressure equation[8]):

$$q = C\left[\left(p^r\right)^2 - \left(p^{bhf}\right)^2\right]^n, \tag{3.6}$$

where the constants C and n are determined from (at least two, but preferably more) flow tests. Vogel's and Fetkovich's methods are only two examples of many IPR models developed over the years. In addition, several modifications and combinations of IPRs have been derived to account for e.g. undersaturated reservoirs, skin factor[9], horizontal wells, special reservoir conditions, etc. Commercial simulation packages such as OLGA provide the user with a choice of which IPR is to be used for simulation, moreover some allow the user to create a custom (tabular) IPR. Both the simple PI, Vogel and Fetkovich methods (and the many other methods not mentioned here) must be matched to current reservoir conditions to be able to predict flow rates well.

3.3.2 Vertical Lift Performance

The *Vertical Lift Performance* (VLP) curve describes the flow through the well tubing, i.e. from the bottom hole to the wellhead on the sea bed (or topside). The VLP curve represents pressure drop due to gravity (loss of potential energy) and drag/friction for a given flow rate, and should take into account factors like well geometry, water cut, GOR and fluid properties. This gives rise to a complicated expression, especially for multiphase flow. Different flow velocities result in different flow patterns/regimes, making it difficult to predict the behaviour of the flow. Much effort has been made in developing correlations for predicting multiphase pressure gradients in producing wells and

[7] The productivity index is a term used to describe slope of the IPR.

[8] The backpressure equation has traditionally been used for gas wells, but Fetkovich showed by using well tests that the method also applies to oil wells (Beggs, 2003).

[9] The skin factor is a measure of well damage.

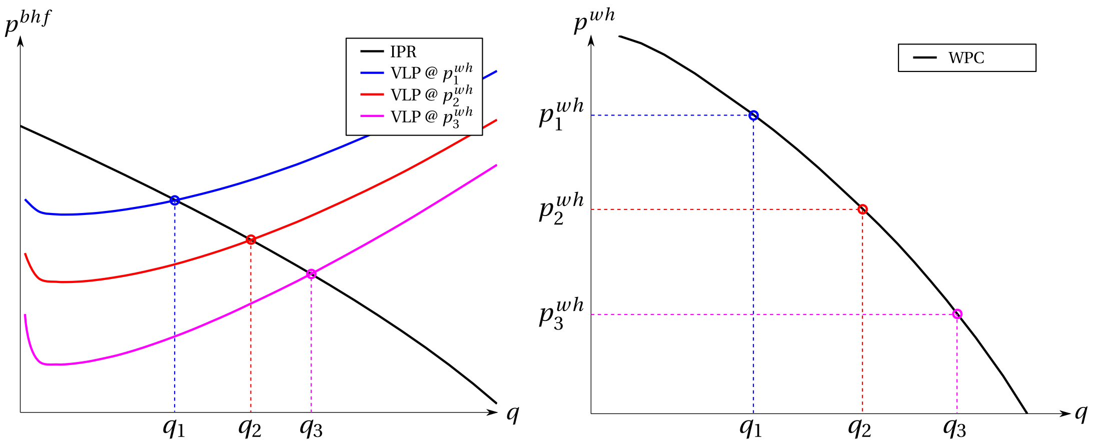
pipelines. This is discussed further in Section 3.5 about pipeline pressure drop. A VLP curve may include a number of variables, depending on its use. However, when plotting a VLP curve, the GOR, water cut and temperature are typically assumed fixed, which results in a flow rate/wellhead pressure versus bottom hole pressure relation, i.e.

$$p^{bhf} = f_{\text{vlp}}(p^{wh}, q), \tag{3.7}$$

or $p^{bhf} = f_{\text{vlp}}(q)$ if we also fix the wellhead pressure p^{wh}. Three VLP curves are shown in Figure 3.4(a). Here, the GOR, water cut, temperature and wellhead pressure are fixed, and each VLP represents a different wellhead pressure.

3.3.3 The Well Performance Curve (IPR/VLP intersection)

The *Well Performance Curve* (WPC) summarizes the information contained in the IPR and VLP curves. For a given wellhead pressure, we can plot the IPR together with the VLP curve, since they both map a flow rate to a bottom hole pressure. At some flow rate these curves will intersect, which gives a relation between the flow rate and the (given) wellhead pressure. By doing this for every wellhead pressure, we obtain the WPC. This is shown graphically in Figure 3.4, which shows the IPR/VLP intersection for three wellhead pressures (Fig. 3.4(a)) and the resulting WPC (Fig. 3.4(b)).



(a) IPR/VLP intersection

(b) WPC

Figure 3.4: The IPR/VLP intersection and Well Performance Curve.

Using the IPR and VLP curves to model the well would require the two equations (3.4) and (3.7). Note that the VLP curve is ambiguous with respect to flow rate; as seen in Figure 3.4(a), it may have two solutions for a single bottom hole pressure. This is caused by the initial drop as the flow rate starts to increase, which is due to gas bubbles lifting the fluid and decreasing the pressure drop. Eventually, friction takes hold and the pressure drop increases[10]. This illustrates the usefulness of the WPC; we can replace (3.4) and (3.7) with a single (usually well-behaved) equation

$$p^{wh} = f_{\text{wpc}}(q). \tag{3.8}$$

3.4 Choke valves

The wellhead choke valves are used to control the flow from each well, by varying the flow restriction. In addition to the choke valves on each wellhead, one choke valve is fitted at the top of each riser in the turret, upstream the separator. Many flow metering systems are based on the principle of restricted flow, i.e. determining the flow rate based on an accurate relationship between flow rate and the pressure drop over a known restriction. The wellhead and turret choke valves represent the main restriction of flow in the production system, when one makes the reasonable assumption that any on/off shut-in/emergency valves are fully open. An accurate model of the choke valves can thus be used to estimate flow rates. Multiphase choke models are usually based on either the valve equation, Bernoulli's principle or a control volume approach. In this section, we will look at an example of the former, namely the multiplier model. A good reference for this section has been (Schüller et al., 2003), which includes a brief introduction to models for two-phase pressure drop through choke valves. For the interested reader, this paper also describes the more advanced control volume-based Hydro model, which is the model used in the OLGA simulations of Chapter 8.

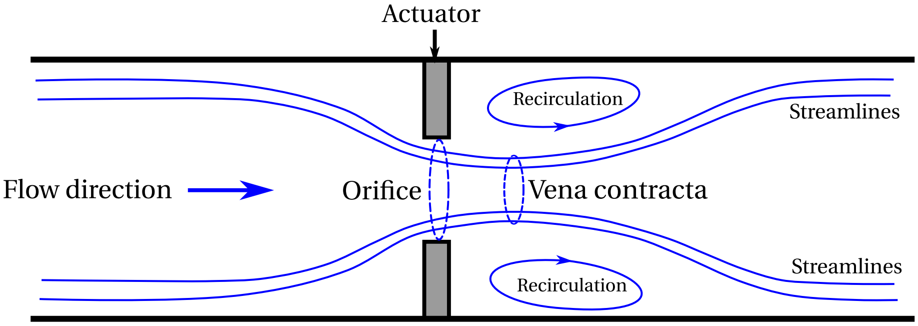


Figure 3.5: Simplified choke model.

3.4.1 Multiplier models

A traditional way to model choke valves with single-phase (turbulent) flow is to use the valve equation, i.e. let the volumetric flow q be a function of the discharge coefficient

[10] The two solutions are referred to as the *low-temperature* and *high-temperature* solutions, since the low flow rate is usually accompanied by a low temperature and vice versa.

C_d, the orifice area A, the fluid density ρ and the differential pressure over the choke Δp (Egeland & Gravdahl, 2002):

$$q = C_d A \sqrt{\frac{2}{\rho}\Delta p}. \tag{3.9}$$

The discharge coefficient is used to account for unrecoverable energy loss through the choke, which is difficult to determine exactly. In addition, the orifice area A is usually somewhat larger than the actual area of the flow (the smallest flow area is known as the *vena contracta*, see Fig. 3.5). The actual flow area depends on both flow regime and the choke position u. Some models include both choke position and unrecoverable energy loss in a single *flow coefficient* C_v, which is usually a function of the choke position, i.e.

$$q = C_v \sqrt{\frac{\Delta p}{\rho}}, \quad \text{where} \quad C_v = f_{C_v}(u), \tag{3.10}$$

assuming C_v is given in appropriate units[11]. By installing the choke valve in a test bench, the flow coefficient can be determined for a set of choke positions by varying the upstream pressure and recording a set of resulting pressure conditions and flow rates. This results in a $u \rightarrow C_v$ curve unique to each valve. Usually, this test is done with water, which means there will inevitably be some error in the model when hydrocarbons flow through the valve. The main differences between the actual flows present in the production system and the single-phase laboratory tests, are phenomena related to flashing of light hydrocarbon components (gas) and the different upstream flow patterns which occur in multiphase flow. A simple way to include the effect of multiphase flow is to introduce a *multiphase multiplier* Ψ_{lo}, and replacing the density with a *relative density* ρ_{rel} to further correct the error, i.e.

$$q = \Psi_{lo} C_v \sqrt{\frac{\Delta p}{\rho_{rel}}}, \quad \text{or} \quad \Delta p = \Psi_{lo}^2 \rho_{rel} \left(\frac{q}{C_v}\right)^2, \tag{3.11}$$

where $\rho_{rel} = \rho_m / \rho_w^{\text{std}}$; ρ_m is the density of the hydrocarbon mixture and ρ_w^{std} is the density of water at flow test (standard) conditions. The multiphase multiplier represents the ratio between the actual (multiphase) pressure drop and the single-phase pressure drop for similar mass flow rates. A fundamental assumption in multiplier models is incompressible flow, i.e. the fluid density is assumed constant throughout the choke valve. This is because the multipliers are calculated from upstream conditions only. Another important assumption is the the flow conditions are *subcritical*, which means that all velocities inside the choke are subsonic. Additional assumptions depend on the

[11] The unit for C_v is per definition [USGPM/psi], i.e. US gallons per minute for a pressure drop of 1 psi. (Schüller et al., 2003), which means appropriate unit conversion factors must be used before using the C_v curve in an equation with SI units.

choice of multiplier. For instance, the homogenous equilibrium model (HEM) assumes homogenous flow, i.e. the multiphase mixture is treated as a single fluid and all the phases are assumed to have the same velocity (Mayinger & Kiederle, 1993). The HEM multiplier is given by

$$\Psi_{lo}^2 = 1 + \alpha_g \left(\frac{\rho_l}{\rho_g} - 1 \right), \tag{3.12}$$

where α_g is the gas mass fraction, ρ_l is the liquid density and ρ_g is the gas density. The liquid density can be calculated from the oil and water densities using a flowing volume fraction, assuming there is no slippage between the oil and water phases (Beggs, 2003):

$$\rho_l = \frac{q_o^{\text{is}}}{q_o^{\text{is}} + q_w^{\text{is}}} \rho_o + \frac{q_w^{\text{is}}}{q_o^{\text{is}} + q_w^{\text{is}}} \rho_w, \tag{3.13}$$

where ρ_o is the oil density, ρ_w is the water density, and q_o^{is}, q_w^{is} are the in-situ volumetric flow rates of oil and water, respectively. More advanced multipliers may account for separated flow. In these models each phase is treated separately with only frictional interaction between the phases. One example is Morris' multiplier, which is the multiplier used in this thesis;

$$\Psi_{lo}^2 = \left[\alpha_g \frac{\rho_l}{\rho_g} + k(1 - \alpha_g) \right] \left[\alpha_g + \frac{(1-\alpha_g)}{k} + \left(1 + \frac{(k-1)^2}{\sqrt{\rho_l/\rho_g}} \right) \right], \tag{3.14}$$

where k is the Chisholm slip[12] correlation; $k = \sqrt{\alpha_g \frac{\rho_l}{\rho_g} + (1-\alpha_g)}$. The choke valves used in the Skarv production system have indeed been tested before installation, using a water test facility, meaning that $f_{C_v}(u)$ is available. In addition, fluid properties are easily available through evaluation of the spline-approximated PVT tables. This makes a multiplier model a natural choice for implementation.

3.5 Pipeline pressure drop

Like the well VLP curve, pipeline flow models become complicated when multiphase flow is to be considered. However, their overall structures are fairly simple. The pressure loss Δp in a pipeline is the sum of three terms; the hydrostatic pressure loss due to elevation change, the frictional pressure loss due to drag/friction, and the accelerational pressure loss which is related to the kinetic energy change of the fluids in the pipeline (Çengel & Cimbala, 2010):

$$\Delta p = \underbrace{\Delta p_{\text{hs}}}_{\text{Hydrostatic}} + \underbrace{\Delta p_{\text{fr}}}_{\text{Friction}} + \underbrace{\Delta p_{\text{acc}}}_{\text{Acceleration}} . \tag{3.15}$$

[12] *Slip* is a term which refers to the relative velocity between the phases.

Since fluid properties are constantly changing along the pipeline due to changing pressure and temperature, the three terms are expressed as gradients, i.e. differential pressure drops for infinitesimal pipe segments with length dL. In practice, a spatial discretization is performed by dividing the pipeline (or well) into finite segments.

Hydrostatic pressure gradient: The hydrostatic pressure gradient is given by

$$\frac{\mathrm{d}p_{\mathrm{hs}}}{\mathrm{d}L} = -\rho_m g \sin\theta, \tag{3.16}$$

where ρ_m is the multiphase fluid density, g is the acceleration of gravity, and θ is the inclination of the pipe segment. Although this equation is familiar, it is far from trivial to find a density which represents the multiphase mixture present in the pipeline. According to (Beggs, 2003), a common way to calculate this density is to use the *liquid holdup* H_l to take a weighted sum of the liquid (ρ_l) and gas (ρ_g) densities:

$$\rho_m = \rho_l H_l + \rho_g (1 - H_l). \tag{3.17}$$

The liquid holdup is defined as the volume fraction occupied by liquid, i.e. $H_l = V_l / V_p$, where V_l is the liquid volume and V_p is the total volume of the pipe or pipe segment under consideration. H_l is thus a number between zero (single-phase gas flow) and one (no gas at all). $(1 - H_l) = H_g$ is the volume fraction occupied by gas, and this is usually called the *gas holdup* or *void fraction*.

Frictional pressure gradient: The frictional pressure gradient is generally given by

$$\frac{\mathrm{d}p_{\mathrm{fr}}}{\mathrm{d}L} = \frac{\lambda_m \rho_m v_m^2}{2gD}, \tag{3.18}$$

where λ_m is some friction factor, ρ_m is some density, v_m is some velocity, g is the acceleration of gravity, and D is the segment diameter. The excessive use of the word "some" is here due to the fact that it is difficult to decide which friction factor, density and velocity to use when dealing with multiphase flow. When the flow regime is dominated by gas, it may be most appropriate to base (3.18) on gas properties, however, if the flow regime is dominated by liquid (e.g. bubble flow, cf. Fig. 3.1), (3.18) should be based on liquid properties.

Accelerational pressure gradient: The accelerational pressure gradient is given by

$$\frac{\mathrm{d}p_{\mathrm{acc}}}{\mathrm{d}L} = \frac{\rho_m v_m}{D} \frac{\mathrm{d}v_m}{\mathrm{d}L}, \tag{3.19}$$

where dv_m/dL is the velocity gradient along the pipeline. In other words, a rapid change in velocity will generate an accelerational pressure loss, which is due to the kinetic energy change of the fluids. Although small, this term will be nonzero for any pipeline with compressible flow and/or changing cross-sectional area (Demneh & Mesbah, 2008). Again, finding an appropriate velocity and density is difficult for multiphase flow. Due to its small contribution, the acceleration term is often ignored.

The main difference between different multiphase pressure drop models is how the multiphase parameters in (3.16), (3.18) and (3.19) are decided. This amounts to selecting some method to find the fluid properties (e.g. black-oil or PVT table evaluation), and somehow calculating an appropriate liquid holdup and multiphase friction factor. This is usually done by some correlation, or a (semi-)mechanistic model. A few examples of multiphase flow correlations/models are given in Table 3.1 (many more are available).

Method	**Applies to**	**Type**
Duns & Ros (1963)	Vertical wells	Empirical correlation
Beggs & Brill (1973)	Wells and pipelines	Empirical correlation
OLGAS (Bendiksen et al., 1991)	Wells and pipelines	Mechanistic model
GRE (proprietary BP model)	Wells and pipelines	Mechanistic model

Table 3.1: A few correlations/models for multiphase pipeline flow.

When fitting these models to observed data, the hydrostatic and friction terms are calibrated separately. To illustrate how one such method may work, we will take a closer look at the model used in this thesis, namely the Beggs and Brill method.

3.5.1 The Beggs and Brill method

The Beggs and Brill method (Beggs & Brill, 1973) is perhaps one of the most well-known correlations for multiphase flow. It has also been used within the IO center, in the reservoir optimization framework RESOPT[13]. The method is based on measurements taken from a test rig with air and water. To calculate the pressure drop in a pipe segment, a Newton-iterative approach is used. First, an initial guess is made for the pressure drop, which is used to calculate an average pressure in the segment. Then, fluid properties are calculated for the average pressure using e.g. PVT tables or some other compositional model. This is used to calculate velocities and a set of correlation parameters, which together with the segment inclination decide the flow regime and liquid holdup. Based on this, a multiphase density is calculated which is used together with a set of correlation functions to determine hydrostatic and frictional pressure gradients. Then, based on the segment length, the total pressure drop is calculated (including an acceleration correction term). If the calculated pressure drop is close enough to the initial estimate,

[13] See IO center web page, http://www.iocenter.no/presentation/resopt.

the method returns the calculated pressure drop. If not, the process is repeated with the calculated pressure drop as a new initial guess. Further details are not given here; refer to the cited paper or (Beggs, 2003, pp. 194-196) for a concrete example. Since the pressure drops are calculated segment-wise, the calculation must be started at one end of the pipeline. The pressure drop is accumulated along the way to obtain upstream pressures for the next segment to be calculated. Another thing to note is that the correlation is given for US field units, meaning that appropriate conversions but be used for other unit systems.

3.6 Mass balance equations

The production system is finally pieced together using mass balance equations (conservation of mass). This is perhaps the simplest part of the modelling process; in the graph representation framework we simply require the total incoming mass flow rate to a vertex to be equal to the total mass flow leaving the vertex (Fig. 3.6).

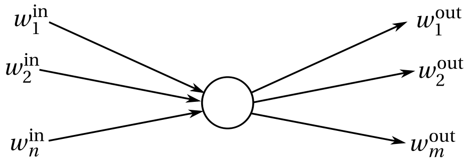


Figure 3.6: Mass balance in a graph represenation.

For a vertex with n inflows and m outflows, this can be expressed as

$$\sum_{i=1}^{n} w_i^{\text{in}} = \sum_{i=1}^{m} w_i^{\text{out}}. \tag{3.20}$$

We can also state this equation in terms of standard volumetric flow rates, since these can be converted to mass flow rates by multiplying with the density at standard conditions. (3.20) is usually applied separately to each phase.

Chapter 4

The static flow estimation problem

In this chapter, we formulate a weighted least-squares optimization problem for estimating flow rates in the template shown in Figure 1.4. A steady-state, or static, model for the network is presented and included as constraints in the optimization problem. The introductory section will give a quick overview of the model structure and indexing scheme. Then, we will present an overview over the variables involved in the optimization problem, before we go on to the modelling part. First, we explain how we extract BP's steady-state models from their modelling tool GAP and approximate them using B-splines. Then, the model itself is presented in the form of constraints in the optimization problem. Finally, we discuss the objective function, variable bounds and the selection of a good starting point for the solver, and we give a summary of the complete optimization problem.

4.1 Introduction

Following a traditional approach, the flow network shown in Figure 1.4 is represented by a graph consisting of vertices and edges, as shown in Figure 4.1.

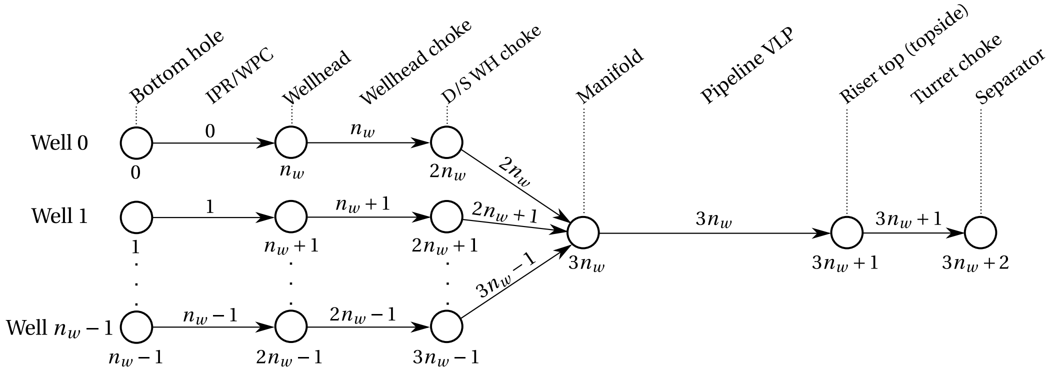


Figure 4.1: Flow network model with indexing.

The vertices represent pressure nodes in the network, and are equipped with pressure and temperature variables. The edges represent wells, choke valves and pipelines, and are thus equipped with some model which describes the pressure loss across the edge.

4.1.1 Indexing

Two sets of indices are defined; one for the vertices and one for the edges. Each vertex and each edge is identified by a unique number. The indexing starts at zero, mostly because this indexing is compatible with the C++ implementation. Defining n_w as the number of wells, the first n_w vertices represent the bottom holes of the wells. The following n_w vertices represent the wellheads, and the next n_w represent the positions downstream the wellhead chokes. The last three vertices represent the manifold, riser top/outlet and finally the separator. Each edge is indexed with the index of the vertex it is leaving. Overviews of the vertex and edge numbering for the template in Fig. 4.1 and for a two-well template (columns labelled 2W) are given in Tables 4.1 and 4.2. These tables should be used together with Figure 4.1 as a reference if the indexing becomes confusing later in the chapter. For convenience, an index set for the wells is defined as $\mathscr{W} = \{i\}_{i=0}^{n_w-1}$.

Vertex index	2W	Description
0	0	Bottom hole, well 0
1	1	Bottom hole, well 1
⋮	⋮	⋮
n_w	2	Wellhead, well 0
n_w+1	3	Wellhead, well 1
⋮	⋮	⋮
$2n_w$	4	Downstr. choke, well 0
$2n_w+1$	5	Downstr. choke, well 1
⋮	⋮	⋮
$3n_w$	6	Manifold
$3n_w+1$	7	Upstr. turret choke
$3n_w+2$	8	Separator

Table 4.1: Vertex indexing.

Edge index	2W	Description
0	0	Well tubing, well 0
1	1	Well tubing, well 1
⋮	⋮	⋮
n_w	2	Wellhead choke, well 0
n_w+1	3	Wellhead choke, well 1
⋮	⋮	⋮
$2n_w$	4	Manifold jumper, well 0
$2n_w+1$	5	Manifold jumper, well 1
⋮	⋮	⋮
3_w	6	Pipeline (flowline + riser)
$3n_w+1$	7	Turret choke

Table 4.2: Edge indexing.

4.2 Variable overview

This section provides an overview of the variable vector used in the static flow estimation problem. Each vertex in the flow network has three associated variables; the reconciled (estimated) pressure, the measured pressure and the measured temperature.

Each edge has four associated variables, namely the oil, gas, water and liquid flow rates. The choke valves each have three associated variables; the choke position, the flow coefficient and the average temperature in the choke. In addition, separate variables are defined for model errors, pressure errors and objective function value. Thus, the variable vector **x** is given as

$$\mathbf{x} = \left[\mathbf{q}^\top, \mathbf{p}^\top, r_{go}^p, r_{wc}^p, \tilde{\mathbf{p}}^\top, \tilde{\mathbf{T}}^\top, \tilde{\mathbf{u}}^\top \mathbf{w}^\top, \mathbf{v}^\top, \mathbf{C}_v^\top, \overline{\mathbf{T}}_{\text{chk}}^\top, J\right]^\top . \tag{4.1}$$

These variables are explained in Table 4.3 below, which also shows the number of variables in each subvector and the units used. As indicated by the four blocks in the table, the variable vector can be divided into four categories; estimates, measurements, error variables and auxiliary variables.

Symbol	**Definition**	**Description**	**Variables**	**Unit**
$\mathbf{q}$	$[\mathbf{q}_o^\top, \mathbf{q}_g^\top, \mathbf{q}_w^\top, \mathbf{q}_l^\top]^\top$	Flow rate estimates (see below)	-	[Sm^3/h]
$\mathbf{q}_o$	$[q_o^0, q_o^1, \dots, q_o^{3n_w+1}]^\top$	Oil rate estimates	$3n_w + 2$	[Sm^3/h]
$\mathbf{q}_g$	$[q_g^0, q_g^1, \dots, q_g^{3n_w+1}]^\top$	Gas rate estimates	$3n_w + 2$	[Sm^3/h]
$\mathbf{q}_w$	$[q_w^0, q_w^1, \dots, q_w^{3n_w+1}]^\top$	Water rate estimates	$3n_w + 2$	[Sm^3/h]
$\mathbf{q}_l$	$[q_l^0, q_l^1, \dots, q_l^{3n_w+1}]^\top$	Liquid rate estimates	$3n_w + 2$	[Sm^3/h]
$\mathbf{p}$	$[p^0, p^1, \dots, p^{3n_w+2}]^\top$	Reconciled pressures	$3n_w + 3$	[bara]
r_{go}^p	-	Pipeline GOR	1	[Sm^3/Sm^3]
r_{wc}^p	-	Pipeline water cut	1	[-]
$\tilde{\mathbf{p}}$	$[\tilde{p}^0, \tilde{p}^1, \dots, \tilde{p}^{3n_w+2}]^\top$	Pressure measurements	$3n_w + 3$	[bara]
$\tilde{\mathbf{T}}$	$[\tilde{T}^0, \tilde{T}^1, \dots, \tilde{T}^{3n_w+2}]^\top$	Temperature measurements	$3n_w + 3$	[°C]
$\tilde{\mathbf{u}}$	$[\tilde{u}^0, \dots, \tilde{u}^{n_w-1}, \tilde{u}^t]^\top$:	Choke positions	$n_w + 1$	[-]
$\mathbf{w}$	$[\mathbf{w}_{\text{ipr}}^\top, \mathbf{w}_{\text{wpc}}^\top, \mathbf{w}_{\text{chk}}^\top, w_{\text{vlp}}, w_{\text{chk}}^t]^\top$	Model errors (see below)	-	[bar]
$\mathbf{w}_{\text{ipr}}^\top$	$[w_{\text{ipr}}^0, w_{\text{ipr}}^1, \dots, w_{\text{ipr}}^{n_w-1}]^\top$	IPR model errors	n_w	[bar]
$\mathbf{w}_{\text{wpc}}^\top$	$[w_{\text{wpc}}^0, w_{\text{wpc}}^1, \dots, w_{\text{wpc}}^{n_w-1}]^\top$	WPC model errors	n_w	[bar]
$\mathbf{w}_{\text{chk}}^\top$	$[w_{\text{chk}}^0, w_{\text{chk}}^1, \dots, w_{\text{chk}}^{n_w-1}]^\top$	Wellhead choke model errors	n_w	[bar]
w_{vlp}	-	Pipeline VLP model error	1	[bar]
w_{chk}^t	-	Turret choke model error	1	[bar]
$\mathbf{v}$	$[v^0, v^1, \dots, v^{3n_w+2}]^\top$	Pressure measurement errors	$3n_w + 3$	[bar]
v^i	$p^i - \tilde{p}^i$	Press. meas. error for vertex i	-	[bar]
$\mathbf{C}_v$	$[C_v^0, \dots, C_v^{n_w-1}, C_v^t]^\top$:	Choke flow coefficients	$n_w + 1$	[USGPM/psi]
$\overline{\mathbf{T}}_{\text{chk}}$	$[\overline{T}_{\text{chk}}^0, \dots, \overline{T}_{\text{chk}}^{n_w-1}, \overline{T}_{\text{chk}}^t]^\top$	Average temperature in chokes	$n_w + 1$	[°C]
J	-	Objective function value	1	[bar^2]

Table 4.3: Variable overview, static flow estimation problem.

The general notation is as follows: Superscripts denote the location in the network, while subscripts denote phase (for flow rate variables) or model (for error variables). A few examples; p^0 is the pressure in vertex 0 (i.e. the bottom hole of well 0), q_o^0 is the flow rate of oil through edge 0 (well tubing of well 0), and w^0_{ipr} is the IPR model error for edge 0. We will use a tilde to denote a measurement, i.e. $\tilde{p}^0$ is the measured pressure in vertex 0. For the choke variables, the superscript denotes well number, i.e. C^0_v is the flow coefficient for the wellhead choke on well 0. For the turret choke, a t-superscript is used, i.e. C^t_v is the flow coefficient for the turret choke. Defining $n_{\mathbf{x}}$ as the number of variables, this gives a total of $n_{\mathbf{x}} = 30n_w + 27$ variables, or 87 variables for a two-well template. Note that the size of the variable vector scales linearly with the number of wells in the model. We also define the number of measurements $n_{\mathbf{y}}$ as the size of a vector containing all the measurements, i.e. $[\tilde{\mathbf{p}}^\top, \tilde{\mathbf{T}}^\top, \tilde{\mathbf{u}}^\top]^\top$, which gives $n_{\mathbf{y}} = 7n_w + 7$. Having defined the variables and model structure, we will now go on to the modelling part, starting with GAP models and B-spline approximations.

4.3 GAP models and B-spline approximations

As mentioned above, each edge in the flow network is equipped with some model which describes the pressure loss across the edge. BP's model of the Skarv field in GAP contains such pressure loss models for wells and pipelines, which we wish to include in our optimization problem. In this section, we will summarize the process of extracting the relevant models from GAP and representing them as B-spline functions which can be included in the optimization problem.

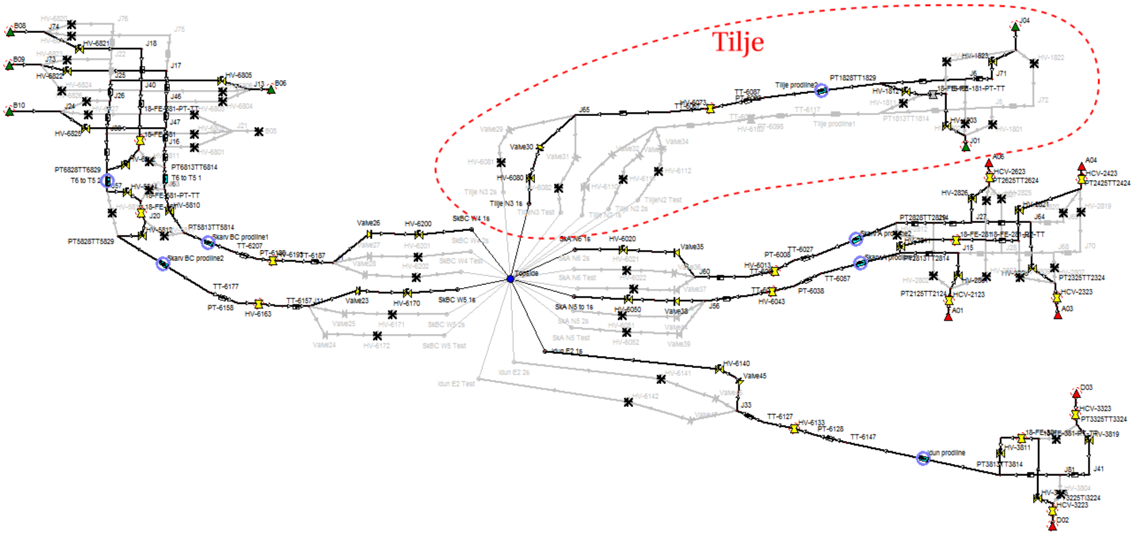


Figure 4.2: BP's GAP model of the Skarv subsea production system.

4.3.1 GAP and PROSPER

GAP stands for General Allocation Package, and is a steady-state multiphase simulation and optimization tool for petroleum production networks. GAP links to PROSPER, which is a well modelling tool (i.e. well models in GAP are actually PROSPER models). Both GAP and PROSPER are vendored by Petroleum Experts[1]. In GAP, the production network is modelled by adding and connecting elements like wells, pipelines, valves etc. in a graphical user interface, and associating each element with a model. In Figure 4.2, we show BP's model of the Skarv production network as it appears in GAP. The models used in this thesis are taken from the Tilje template, which is indicated in the figure.

4.3.2 GAP models

The Tilje models taken from GAP are the well models (IPRs and WPCs), and the pipeline VLP. Choke models are not included in BP's GAP model; the choke valves are represented by a fixed pressure drop. Therefore, we have a separate choke model (see Section 4.5.3). The IPRs in GAP are, as discussed in Chapter 3, nonlinear relations between bottom hole pressures and liquid flow rates, and represent the pressure drop from the reservoir to the bottom hole. The WPCs relate wellhead pressure to liquid flow rate, and represent the pressure drop from the reservoir to the wellhead. The pipeline VLP represents the pressure drop in the pipeline, and is represented by a black-box function which calculates the downstream pressure for a given liquid flow rate, upstream pressure, upstream temperature, GOR and water cut. In our case, this black-box function is the Beggs and Brill correlation, but GAP allows the user to select from a list of 22 different correlations[2].

These three models are summarized in Table 4.4, where we list the model representations described above in our own nomenclature from Table 4.3. Here, we treat $\left\{f^i_{\text{ipr}}(\cdot)\right\}_{i\in\mathcal{W}}$, $\left\{f^i_{\text{wpc}}(\cdot)\right\}_{i\in\mathcal{W}}$ and $f_{\text{vlp}}(\cdot)$ as black-box functions which can be evaluated, but whos internal structure is not known.

Model	Representation	Number of models
IPR	$q^i_l = f^i_{\text{ipr}}(p^i),\ \ \forall\ i \in \mathcal{W}$	n_w
WPC	$q^i_l = f^i_{\text{wpc}}(p^{n_w+i}),\ \ \forall\ i \in \mathcal{W}$	n_w
Pipeline VLP	$p^{3n_w+1} = f_{\text{vlp}}(q^{3n_w}_l, p^{3n_w}, \tilde{T}^{3n_w}, r^p_{go}, r^p_{wc})$	1

Table 4.4: GAP models.

[1] see http://www.petex.com/products for more information about GAP, PROSPER and other software from Petroleum Experts.

[2] BP do not actually use the Beggs and Brill correlation for Skarv, but in the interest of transparency, we choose to use a well-known and "open-domain" method here.

4.3.3 B-spline approximations

The question now is: How can we include the models in Table 4.4 in our optimization problem? One possible way might be to couple our solver with GAP in such a way that the solver can query GAP for function evaluations. However, this is not efficient, because (1) model evaluation would be (relatively) slow, and (2) GAP does not provide gradients, which means the solver would have to approximate gradients by e.g. finite differencing (resulting in *multiple* evaluations for each internal iteration). Promising results have been achieved with approximation schemes, i.e. sampling the models in Table 4.4 in a grid and feeding the resulting data sets to some approximation or interpolation scheme with e.g. piecewise linear functions (Kosmidis et al., 2005; Gunnerud & Foss, 2009) or B-spline (piecewise polynomial) functions (Sandnes, 2013). Here, we choose the latter approach, based on the promising results of (Sandnes, 2013) and (Grimstad & Sandnes, 2014), and previous experience with B-splines (Robertson, 2013). We will not go into the details of function approximation with B-splines here, but a short introduction is given in Appendix A.

In short, B-splines are piecewise polynomial functions which, due to their flexibility, are able to approximate any function (linear or nonlinear) quite accurately. In addition, they can be made twice continuously differentiable ($\mathscr{C}^2$), meaning we can evaluate Jacobians and Hessians directly and perform sensitivity analyses on each model. The accuracy of the approximation depends on how much detail is captured through sampling of the function. To obtain good approximations of the functions in Table 4.4, we inspect the functions in the relevant ranges of the input variables to determine how the functions should be sampled to capture a sufficient amount of detail. Then, we sample the functions and pass the resulting tables of samples to a B-spline approximation scheme which returns the B-spline representation of the function. This results in the B-spline functions shown in Table 4.5 below.

Function to be approximated	**Resulting B-spline approximation**
$p^i = {f_{\text{ipr}}^i}^{-1}(q_l^i) + \Delta p^i, \ \forall\ i \in \mathscr{W}$	$p^i = \Phi_{\text{ipr}}^i(q_l^i), \ \forall\ i \in \mathscr{W}$
$p^{n_w+i} = {f_{\text{wpc}}^i}^{-1}(q_l^i) + \Delta p^{n_w+i}, \ \forall\ i \in \mathscr{W}$	$p^{n_w+i} = \Phi_{\text{wpc}}^i(q_l^i), \ \forall\ i \in \mathscr{W}$
$p^{3n_w+1} = f_{\text{vlp}}(q_l^{3n_w}, p^{3n_w}, \tilde{T}^{3n_w}, r_{go}^p, r_{wc}^p)$	$p^{3n_w+1} = \Phi_{\text{vlp}}(q_l^{3n_w}, p^{3n_w}, \tilde{T}^{3n_w}, r_{go}^p, r_{wc}^p)$

Table 4.5: B-spline approximated GAP models.

Two things are to be noted for the IPRs and WPCs. Firstly, we have switched the input and output variables, resulting an approximation of the *inverse* IPRs and WPCs (i.e. pressure as a function of rate). This is because we prefer all edge models to be evaluated as pressures (see Section 4.5.2). Secondly, we have included the offsets Δp^i and Δp^{n_w+i} before creating the approximations. These are calibration constants which we will discuss in Chapter 6.

4.4 Mass balance constraints

We will now proceed with defining all the constraints which are to be included in our optimization problem. The mass balance constraints account for mass flows in and out of each node, and use the GORs and water cuts to determine the phase split. The mass balance constraints are so-called *hard* constraints, i.e. they have to be satisfied exactly.

4.4.1 Vertex in/outflow constraints

The vertex in/outflow constraints require the sum of inflows into a vertex to equal the sum of outflows, as explained in Chapter 3. Each phase is treated individually, i.e. there are separate mass balance constraints for oil, gas, water and liquid. To write these constraints more efficiently, we use an *incidence matrix* **M**. Such a matrix is often used to describe how the vertices in a flow network (or in general, graph) are connected (Cormen et al., 2001). The elements of **M** are defined as follows:

$$\mathbf{M} = \left\{m_{ij}\right\}_{ij}, \quad m_{ij} = \begin{cases} -1, & \text{if edge } j \text{ is an outgoing edge from vertex } i \\ 1, & \text{if edge } j \text{ is an incoming edge to vertex } i \\ 0, & \text{otherwise} \end{cases}, \tag{4.2}$$

The rows of **M** represent vertices, while the columns represent edges. For the flow network shown in Figure 4.1, **M** becomes the $3n_w + 4 \times 3n_w + 3$ matrix shown below:

$$\mathbf{M} = \left[\begin{array}{c:c:c:ccc}
 & & & 0 & 0 & 0 \\
-\mathbf{I}_{n_w} & \mathbf{0} & \mathbf{0} & & \vdots & \\
 & & & 0 & 0 & 0 \\ \hdashline
 & & & 0 & 0 & 0 \\
\mathbf{I}_{n_w} & -\mathbf{I}_{n_w} & \mathbf{0} & & \vdots & \\
 & & & 0 & 0 & 0 \\ \hdashline
 & & & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{I}_{n_w} & -\mathbf{I}_{n_w} & & \vdots & \\
 & & & 0 & 0 & 0 \\ \hdashline
0 \;\; \dots \;\; 0 & 0 \;\; \dots \;\; 0 & 1 \;\; \dots \;\; 1 & -1 & 0 & 0 \\
0 \qquad\; 0 & 0 \qquad\; 0 & 0 \qquad\; 0 & 1 & -1 & 0 \\
0 \;\; \dots \;\; 0 & 0 \;\; \dots \;\; 0 & 0 \;\; \dots \;\; 0 & 0 & 1 & -1 \\
0 \qquad\; 0 & 0 \qquad\; 0 & 0 \qquad\; 0 & 0 & 0 & 1
\end{array}\right], \tag{4.3}$$

where $\mathbf{I}_{n_w}$ is an $n_w \times n_w$ identity matrix. Further, we define an *internal incidence matrix* $\mathbf{M}_{\text{int}}$, which is simply the incidence matrix with the source vertices (wells) and sink vertex (separator) removed. This corresponds to removing n_w rows from the top of **M** and one row from the bottom:

$$\mathbf{M}_{\text{int}} = \left[\begin{array}{ccc:ccc:ccc:ccc}
 & & & & & & & & & 0 & 0 & 0\\
 & \mathbf{I}_{n_w} & & & -\mathbf{I}_{n_w} & & & \mathbf{0} & & & \vdots & \\
 & & & & & & & & & 0 & 0 & 0\\
\hdashline
 & & & & & & & & & 0 & 0 & 0\\
 & \mathbf{0} & & & \mathbf{I}_{n_w} & & & -\mathbf{I}_{n_w} & & & \vdots & \\
 & & & & & & & & & 0 & 0 & 0\\
\hdashline
0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 & -1 & 0 & 0\\
0 & & 0 & 0 & & 0 & 0 & & 0 & 1 & -1 & 0\\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 1 & -1
\end{array}\right] \tag{4.4}$$

Now we can efficiently write the mass balance constraints as

$$\mathbf{M}_{\text{int}}\mathbf{q}_o = \mathbf{0}, \tag{4.5}$$

$$\mathbf{M}_{\text{int}}\mathbf{q}_g = \mathbf{0}, \tag{4.6}$$

$$\mathbf{M}_{\text{int}}\mathbf{q}_w = \mathbf{0}, \tag{4.7}$$

$$\mathbf{M}_{\text{int}}\mathbf{q}_l = \mathbf{0}, \tag{4.8}$$

where $\mathbf{q}_o, \mathbf{q}_g, \mathbf{q}_w$ and $\mathbf{q}_l$ are vectors consisting of the oil, gas, water and liquid rates (respectively) in each edge, as described in Section 4.2.

4.4.2 GOR, water cut and liquid rate constraints

The individual GORs for each well are used to link the oil and gas rates from that well, i.e.

$$q_g^i = r_{go}^i q_o^i, \qquad \forall\ i \in \mathcal{W} \tag{4.9}$$

Similarly, the individual water cuts for each well are used to link the oil and water rates:

$$q_w^i = \frac{r_{wc}^i}{1 - r_{wc}^i} q_o^i, \qquad \forall\ i \in \mathcal{W} \tag{4.10}$$

Note that r_{go}^i and r_{wc}^i are treated as constant well parameters. The liquid rates are obtained by simply adding the oil rates and water rates:

$$q_l^i = q_o^i + q_w^i, \qquad \forall\ i \in \mathcal{W} \tag{4.11}$$

Due to the mass balance constraints (4.5) through (4.8), the correct GORs, water cuts and liquid rates will propagate though the edges in each well. However, when the well streams mix in the manifold, we must calculate a new GOR and water cut for the common pipeline. These are needed to calculate the pressure drop in the common pipeline.

Since the individual flows of each phase are calculated through the mass balance constraints, we can use these flows to calculate the pipeline GOR r_{go}^{p} through the constraint

$$q_g^{3n_w} = q_o^{3n_w} r_{go}^{p}, \tag{4.12}$$

and the pipeline water cut r_{wc}^{p} as

$$q_w^{3n_w} = q_l^{3n_w} r_{wc}^{p}. \tag{4.13}$$

Recall that the index $3n_w$ denotes the edge leaving the manifold vertex. Note that (4.12) and (4.13) are expressed as *bilinear* rather than fractional/rational constraints; if we had written $r_{go}^{p} = q_g^{3n_w} / q_o^{3n_w}$, the GOR would not be well-defined for $q_o^{3n_w} = 0$.

4.5 Momentum balance constraints

The momentum balance constraints relate the pressures throughout the flow network to each other.

4.5.1 Decreasing pressure constraints

First of all, we assume a positive flow through the network, i.e. hydrocarbons flow from the wells, into the production network and out at the top (separator). This means, with reference to Figure 4.1, that the pressures will decrease from left to right in the flow network. For example, the wellhead pressure for a well will be less than the bottom hole pressure, i.e.

$$p^{n_w+i} \leq p^i, \text{ or equivalently, } -p^i + p^{n_w+i} \leq 0, \;\; \forall \; i \in \mathscr{W}.$$

Again, we make use of the incidence matrix (4.3) to express these constraints efficiently as

$$\mathbf{M}^{\top}\mathbf{p} \leq \mathbf{0}, \tag{4.14}$$

where $\mathbf{p} = [p^0, p^1, \ldots, p^{3n_w+2}]$ is a vector consisting of all the vertex pressures.

4.5.2 Well performance

IPR constraints

The IPR curves are used to relate the bottom hole pressures to the liquid rates in each well. Recall from Section 4.3.2 that the IPR from GAP is given as $q_l^i = f_{\text{ipr}}(p^i)$. However, we want to penalize the amount of model error in the objective function, and to achieve better scaling between the terms, it is more convenient to express all the model errors as pressures. Therefore, the IPR constraints are given as

$$w^i_{\mathrm{ipr}} = p^i - \Phi^i_{\mathrm{ipr}}(q^i_l), \quad \forall\ i \in \mathscr{W}, \tag{4.15}$$

where w^i_{ipr} is the model error for IPR i, p^i is the bottom hole pressure for well i, and q^i_l is the liquid rate from well i. $\Phi^i_{\mathrm{ipr}}(\cdot)$ is the B-spline approximation of the IPR in BP's GAP model, cf. Section 4.3.

WPC constraints

The WPC constraints give a relation between the wellhead pressures and the liquid rates in each well. Like the case was for the IPR curve, we would like to express the model error as a pressure. Therefore, the WPC constraints are given as

$$w^i_{\mathrm{wpc}} = p^{n_w+i} - \Phi_{\mathrm{wpc}}(q^i_l), \quad \forall\ i \in \mathscr{W}, \tag{4.16}$$

where w^i_{wpc} is the model error for WPC i, p^{n_w+i} is the wellhead pressure for well i, and q^i_l is the liquid rate from well i. $\Phi_{\mathrm{wpc}}(\cdot)$ is a B-spline approximation of the WPC in BP's GAP model, cf. Section 4.3.

4.5.3 Choke valves

As mentioned in Chapter 3, the choke valves used in the Skarv field have been tested in a water test facility, meaning that the relation between the choke opening $\tilde{u}$ and the flow coefficient C_v is available.

$$C_v = f_{C_v}(\tilde{u}) \tag{4.17}$$

A multiplier model is chosen because of its relative simplicity, and the fact that it facilitates the use of the flow coefficient characteristic and fluid property tables. However, it does not account for compressible flow, due to the fact that fluid properties are determined from upstream conditions only. Therefore, in an attempt to consider compressible flow, we use a Newton-iterative approach much like the one used for pipelines in (Beggs & Brill, 1973) to allow the fluid properties to be evaluated at the *average* pressure and temperature in the choke. This results in a model which can be evaluated, but not written in closed form. The model calculates a downstream pressure p^{ds}, based on the liquid flow rate q_l, the upstream pressure p^{us}, an average temperature $\overline{T}$, the flow coefficient C_v, the GOR r_{go}, the water cut r_{wc}, and a calibration factor c, i.e.

$$p^{ds} = f_{\mathrm{chk}}(q_l, p^{us}, \overline{T}, C_v, r_{go}, r_{wc}, c), \tag{4.18}$$

where $\overline{T} = \frac{1}{2}T^{us} + \frac{1}{2}T^{ds}$ is the average of the upstream and downstream temperatures. The evaluation of $f_{\mathrm{chk}}(\cdot)$ involves guessing a pressure drop, evaluating the choke model

(3.11) for the resulting average pressure and taking the calculated pressure drop as a new guess. This is repeated until the guessed and calculated pressure drops coincide. This is described in detail in Algorithm C.1.

We now use the B-spline to approximate (4.18). For the wellhead chokes, the GOR, water cut and calibration factor are assumed fixed, which means we need to sample a grid of liquid rates, flow coefficients, upstream pressures and average temperatures. This results in a B-spline $\Phi_{\text{chk}}(q_l, p^{us}, \overline{T}, C_v)$. We also use the B-spline to approximate $f_{C_v}(\tilde{u})$ as $\Phi_{C_v}(\tilde{u})$. As with the well models, we formulate the model constraint in terms of model errors, so for the wellhead chokes the necessary constraints are

$$C_v^i = \Phi_{C_v}^i(\tilde{u}^i), \qquad \forall\ i \in \mathscr{W}, \tag{4.19}$$

$$\overline{T}_{\text{chk}}^i = \frac{1}{2}\left(\tilde{T}^{n_w+i} + \tilde{T}^{2n_w+i}\right) \qquad \forall\ i \in \mathscr{W}, \tag{4.20}$$

$$w_{\text{chk}}^i = p^{2n_w+i} - \Phi_{\text{chk}}^i(q_l^{n_w+i}, p^{n_w+i}, \overline{T}_{\text{chk}}^i, C_v^i), \qquad \forall\ i \in \mathscr{W}. \tag{4.21}$$

Note that index $n_w + i$ denotes the wellhead while $2n_w + i$ denotes downstream the choke. The turret choke is modelled similarly; except we cannot fix the GOR and water cut since these will vary according to the mixing of the well streams. Therefore, the choke model is sampled in two extra dimensions (GOR and water cut). Again, we use another B-spline to approximate the flow coefficient characteristic. This results in the following constraints for the turret choke:

$$C_v^t = \Phi_{C_v}^t(\tilde{u}^t), \tag{4.22}$$

$$\overline{T}_{\text{chk}}^t = \frac{1}{2}\left(\tilde{T}^{3n_w+1} + \tilde{T}^{3n_w+2}\right), \tag{4.23}$$

$$w_{\text{chk}}^t = p^{3n_w+2} - \Phi_{\text{chk}}^t(q_l^{3n_w+1}, p^{3n_w+1}, \overline{T}_{\text{chk}}^t, r_{go}^p, r_{wc}^p C_v^t). \tag{4.24}$$

Here, the index $3n_w+1$ denotes upstream the turret choke, while $3n_w+2$ denotes downstream. To give the reader an impression of how the flow coefficient and choke model may look, Figure 4.3 shows the flow coefficient characteristic for one of the Tilje wellhead chokes, and Figure 4.4 shows (for the same choke) the B-spline approximated pressure drop model for different flow coefficients, fixed at a representative temperature and upstream pressure, i.e. showing the pressure drop as a function of liquid flow rate.

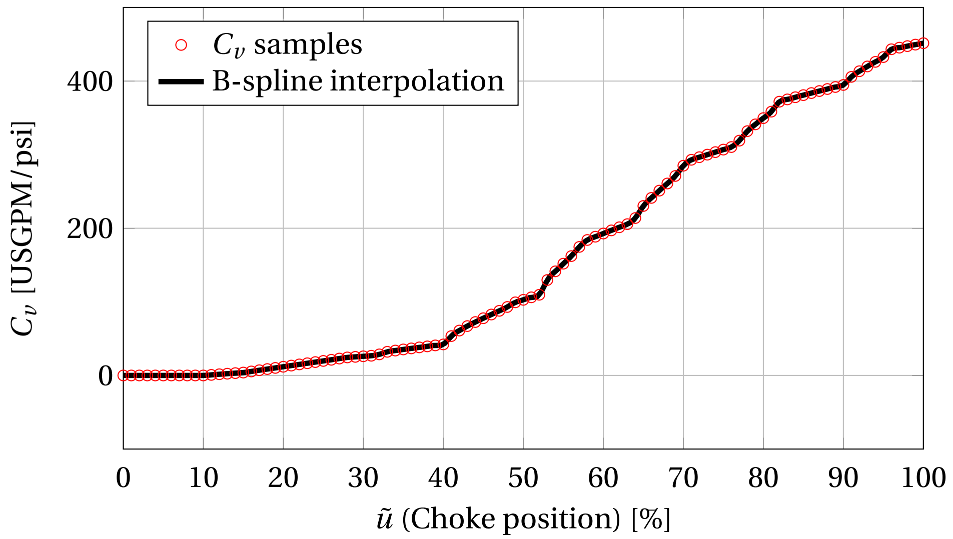


Figure 4.3: Choke flow coefficient (C_v) characteristic.

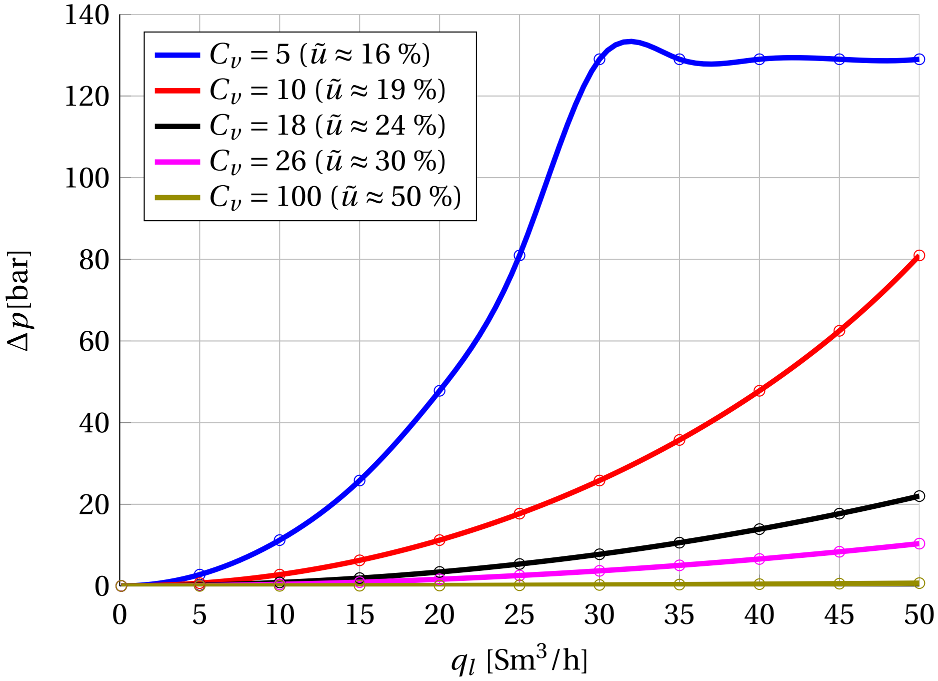


Figure 4.4: Choke model, shown as $q_l \rightarrow \Delta p$ plots for fixed C_v's (in [USGPM/psi]). Sampling points are indicated with circles. Note how increased flow rates become less significant for the pressure drop as C_v increases; for $C_v = 100$, i.e. $\tilde{u} \approx 50$ %, the pressure drop curve is almost flat.

4.5.4 Pipeline

The pipeline (flowline-riser) model is taken as a spline-approximated VLP table generated in GAP. This table is created in GAP by using the Beggs and Brill multiphase flow correlation to calculate downstream pressures for a grid of upstream pressures, upstream temperatures, GORs and water cuts. In this sense, the pipeline model may be seen as a Beggs and Brill correlation. The model is given as

$$w_{\text{vlp}} = p^{3n_w+1} - \Phi_{\text{vlp}}(q_l^{3n_w}, p^{3n_w}, \tilde{T}^{3n_w}, r_{go}^p, r_{wc}^p), \tag{4.25}$$

where w_{vlp} is the model error, p^{3n_w+1} is the pressure at the top of the riser (upstream the turret choke), $q_l^{3n_w}$ is the liquid flow through the pipeline p^{3n_w} is the manifold pressure, $\tilde{T}^{3n_w}$ is the manifold temperature r_{go}^p is the pipeline GOR and r_{wc}^p is the pipeline water cut.

4.5.5 Measurements

An additional constraint is added to include measurements of pressures, temperatures and choke positions. Letting $\mathbf{y}_T$ be the measurement vector at time index T, we add the following constraint:

$$\mathbf{0} = \mathbf{y}_T - [\tilde{\mathbf{p}}^\top, \tilde{\mathbf{T}}^\top, \tilde{\mathbf{u}}^\top]^\top. \tag{4.26}$$

Obviously, the measurements in $\mathbf{y}_T$ must be arranged accordingly.

4.6 Model sensitivity

An important part of modelling is to determine the model *sensitivity*, which boils down to measuring the magnitude of the partial derivatives with respect to each variable. This is useful for determining which variables to pay special attention when modelling, and could also be used to provide better scaling between the terms in the objective function (although we have not attempted to do so here). In our case, we are particularly interested in investigating the model sensitivity with respect to pressure, since most of the other variables are explicitly given through parameters or other constraints. For instance, GORs and water cuts are given either directly (for wells) or through mass balance constraints (pipeline and turret choke), and the choke positions are given by measurements which are assumed accurate. As mentioned in Section 4.3, the B-spline approximations of each model are $\mathscr{C}^2$, which enables sensitivity analysis. We will not discuss this at length, but in summary, the IPRs and WPCs have nice sensitivity properties, the choke valves have good sensitivity properties in the relevant ranges for the flow coefficient, and the pipeline VLP has bad sensitivity properties. We will devote a section to the latter in the results chapter (Section 8.6).

4.7 Formulation of the static flow estimation NLP

4.7.1 Objective function

The objective is the standard least-squares quadratic function:

$$f(\mathbf{q}_l, \mathbf{w}, \mathbf{v}) = \left\|\mathbf{q}_l^- - \mathbf{q}_l\right\|_{\mathbf{P_q}}^2 + \|\mathbf{w}\|_{\mathbf{Q}^{-1}}^2 + \|\mathbf{v}\|_{\mathbf{R}^{-1}}^2, \tag{4.27}$$

where $\mathbf{P_q} \in \mathbb{M}^{3n_w+2\times 3n_w+2}$, $\mathbf{Q}^{-1} \in \mathbb{M}^{3n_w+2\times 3n_w+2}$, and $\mathbf{R}^{-1} \in \mathbb{M}^{3n_w+3\times 3n_w+3}$ are weighting matrices, and $\mathbf{q}_l^-$ is the estimated flow rate from the previous time index. The first term in (4.27) is known as a *regularization* term, whose purpose is to include *a priori* information in the objective. Due to the non-convex nature of the pressure momentum constraints, we are running a risk of formulating an optimization problem with an unknown number of local solutions (possibly far apart). Including this first term is an attempt to encourage the NLP solver to choose solutions close to the solution from the previous iteration by penalizing deviations from the previously estimated liquid flow rates. The weights in $\mathbf{P_q}$ are selected relatively small; we do not want this term to affect the flow estimates, but merely discourage the solver from taking large leaps from one local optimum to another. The second term penalizes *model errors*. By driving this term towards zero, the reconciled pressures will agree more with the edge models. The third term penalizes *pressure measurement errors*. A small value for this term implies that the reconciled pressures agree with the pressure measurements. $\mathbf{w}$ and $\mathbf{v}$ are vectors of model errors and pressure measurement errors, respectively. These are described in Section 4.2. The weighting matrices $\mathbf{P_q}$, $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ decide the relative penalty placed on the flow regularization term, model error term and pressure measurement term, respectively. $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ thus represent our relative confidence in the system model and the available pressure measurements. The structure of $\mathbf{Q}^{-1}$ is as follows:

$$\mathbf{Q}^{-1} = \text{blkdiag}\left(\mathbf{Q}_{\text{ipr}}^{-1}, \mathbf{Q}_{\text{wpc}}^{-1}, \mathbf{Q}_{\text{chk}}^{-1}, Q_{\text{vlp}}^{-1}, \left(Q_{\text{chk}}^{t}\right)^{-1}\right) \tag{4.28}$$

where $\mathbf{Q}_{\text{ipr}} = \text{diag}(Q_{\text{ipr}}^0, \ldots, Q_{\text{ipr}}^{n_w-1})$, $\mathbf{Q}_{\text{wpc}} = \text{diag}(Q_{\text{wpc}}^0, \ldots, Q_{\text{wpc}}^{n_w-1})$ and $\mathbf{Q}_{\text{chk}} = \text{diag}(Q_{\text{chk}}^0, \ldots, Q_{\text{chk}}^{n_w-1})$ are $n_w \times n_w$ diagonal matrices of weights representing the penalty on each model error. The weights in $\mathbf{R} = \text{diag}(R^0, \ldots, R^{3n_w+3})$ represent each pressure error. Keep in mind that for the elements of these matrices, superscripts are indices, not exponents. Some strategies for selecting weights based on calibration data are presented in Chapter 6.

4.7.2 Variable bounds

The default bounds on each variable is $0 \leq x \leq \infty$, where x is some variable. However, there are a few exceptions:

- Variables involved in B-spline constraints are bounded according to the domain of the B-spline function. For instance, let $\Phi(\cdot)$ be a B-spline function sampled from $\underline{x}_1$ to $\overline{x}_1$ in the first variable, $\underline{x}_2$ to $\overline{x}_2$ in the second variable and so forth. Then $\Phi(\cdot)$ is a function

$$\Phi(\cdot) : [\underline{x}_1, \overline{x}^1] \times [\underline{x}_2, \overline{x}^2] \times \cdots \times [\underline{x}_n, \overline{x}_n] \to \mathbb{R}. \tag{4.29}$$

  The B-spline function and its derivatives can only be evaluated inside its domain. Hence, we must bound the variables involved so they stay inside the domain.

- Some variables must be allowed to attain negative values. This applies to pressure measurement errors, model errors and temperature measurements (some negative temperatures were observed in the OLGA simulations in Chapter 8).

4.7.3 Starting point

To obtain a good starting point $\mathbf{x}_{\text{init}}$ for the NLP solver, we use the solution $\mathbf{x}^*_{T-1}$ from the previous iteration together with the latest measurements $\mathbf{y}_T$. The reasoning behind this is the following:

1. The system dynamics are relatively slow compared to the estimation loop frequency, therefore we would not expect the system to evolve much from one iteration to the next. This means the solutions of Problem 4.1 for two subsequent iterations are expected to be close.

2. Variables associated with measurements (pressure measurements $\tilde{\mathbf{p}}$, temperature measurements $\tilde{\mathbf{T}}$ and choke positions $\tilde{\mathbf{u}}$) are subject to the equality constraint (4.26), which means we can remove some infeasibility in the starting point by updating them with the latest measurements.

To obtain a compact expression for $\mathbf{x}_{\text{init}}$, we introduce selection matrices $\mathbf{S}_\mathbf{x}$ ($n_\mathbf{x} \times n_\mathbf{x}$) and $\mathbf{S}_\mathbf{y}$ ($n_\mathbf{x} \times n_\mathbf{y}$) defined as

$$\mathbf{S}_\mathbf{x} = \left\{s_{x,ii}\right\}_{ii}, \quad s_{x,ii} = \begin{cases} 0, & \text{if variable } i \text{ in } \mathbf{x} \text{ is a measurement variable,} \\ 1, & \text{otherwise.} \end{cases} \tag{4.30}$$

$$\mathbf{S}_\mathbf{y} = \left\{s_{y,ij}\right\}_{ij}, \quad s_{y,ij} = \begin{cases} 1, & \text{if measurement } j \text{ in } \mathbf{y}_T \text{ corresponds to index } i \text{ in } \mathbf{x}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.31}$$

We want the starting point for the measurement variables to be updated with the latest measurements, while the starting point for remaining variables should come from the solution from the previous iteration. This leads to the following expression for the starting point:

$$\mathbf{x}_{\text{init}|T} = \mathbf{S}_\mathbf{x}\mathbf{x}^*_{T-1} + \mathbf{S}_\mathbf{y}\mathbf{y}_T, \tag{4.32}$$

where $\mathbf{x}_{\text{init}|T}$ is the starting point for the problem to be solved at time index T.

4.7.4 Summary - the static flow estimation NLP

Problem 4.1: Static flow estimation NLP

$$
\begin{array}{ll}
\underset{\mathbf{x}}{\text{minimize}}\ J & \\
\text{subject to} & \\
\text{objective constraint:} & J \geq \left\| \mathbf{q}_l^- - \mathbf{q}_l \right\|^2_{\mathbf{P_q}} + \|\mathbf{w}\|^2_{\mathbf{Q}^{-1}} + \|\mathbf{v}\|^2_{\mathbf{R}^{-1}} \\
 & \quad \text{Mass balance constraints:} \\
\text{mass balance oil:} & \mathbf{0} = \mathbf{M}_{\text{int}}\mathbf{q}_o \\
\text{mass balance gas:} & \mathbf{0} = \mathbf{M}_{\text{int}}\mathbf{q}_g \\
\text{mass balance water:} & \mathbf{0} = \mathbf{M}_{\text{int}}\mathbf{q}_w \\
\text{mass balance liquid:} & \mathbf{0} = \mathbf{M}_{\text{int}}\mathbf{q}_l \\
\text{well GORs:} & 0 = r^i_{go} q^i_o - q^i_g, \quad \forall\ i \in \mathcal{W} \\
\text{well water cuts:} & 0 = \dfrac{r^i_{wc}}{1 - r^i_{wc}} q^i_o - q^i_w, \quad \forall\ i \in \mathcal{W} \\
\text{well liquid rates:} & 0 = q^i_l - q^i_o - q^i_w, \quad \forall\ i \in \mathcal{W} \\
\text{pipeline GOR:} & 0 = q^{3n_w}_g - q^{3n_w}_o r^p_{go} \\
\text{pipeline water cut:} & 0 = q^{3n_w}_w - q^{3n_w}_l r^p_{wc} \\
 & \quad \text{Momentum balance constraints:} \\
\text{decreasing pressure:} & \mathbf{0} \geq \mathbf{M}^\top \mathbf{p} \\
\text{IPRs:} & w^i_{\text{ipr}} = q^i_l - \Phi^i_{\text{ipr}}(p^i), \quad \forall\ i \in \mathcal{W} \\
\text{rate WPCs:} & w^i_{\text{wpc}q} = q^i_l - \Phi^i_{\text{wpc}q}(p^{n_w+i}), \quad \forall\ i \in \mathcal{W} \\
\text{wellhead chk.:} & w^i_{\text{chk}} = p^{2n_w+i} - \Phi^i_{\text{chk}}(q^i_l, p^{n_w+i}, \overline{T}^i_{\text{chk}}, C^i_v), \quad \forall\ i \in \mathcal{W} \\
\text{pipeline:} & w_{\text{vlp}} = p^{3n_w+1} - \Phi_{\text{vlp}}(q^{3n_w}_l, p^{3n_w}, \tilde{T}^{3n_w}, r^p_{go}, r^p_{wc}) \\
\text{turret chk.:} & w^t_{\text{chk}} = p^{3n_w+2} - \Phi^t_{\text{chk}}(q^{3n_w+1}_l, p^{3n_w+1}, \overline{T}^t_{\text{chk}}, r^p_{go}, r^p_{wc} C^t_v) \\
 & \quad \text{Auxiliary constraints:} \\
\text{WHC flow coeff.:} & 0 = C^i_v - \Phi^i_{C_v}(u^i),\ \forall i \in \mathcal{W} \\
\text{WHC flow coeff.:} & 0 = C^t_v - \Phi^t_{C_v}(u^t) \\
\text{WHC avg. temp.:} & 0 = \overline{T}^i_{\text{chk}} - \dfrac{1}{2}\tilde{T}^{n_w+i} - \dfrac{1}{2}\tilde{T}^{2n_w+i}, \quad \forall\ i \in \mathcal{W} \\
\text{turret chk. avg. temp.:} & 0 = \overline{T}^t_{\text{chk}} - \dfrac{1}{2}\tilde{T}^{3n_w+1} - \dfrac{1}{2}\tilde{T}^{3n_w+2} \\
\text{pressure errors:} & \mathbf{v} = \mathbf{p} - \tilde{\mathbf{p}} \\
\text{measurements:} & \mathbf{0} = \mathbf{y}_T - [\tilde{\mathbf{p}}^\top, \tilde{\mathbf{T}}^\top, \mathbf{u}^\top]^\top \\
 & \quad \text{Variable bounds:} \\
 & \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}
\end{array}
$$

Chapter 5

The dynamic flow estimation problem

The second part of the modelling scope of this thesis is to derive an appropriate dynamic model for the flow network. In this chapter, the static flow network model is extended to include dynamics. In the assignment text, it is stated that one of the operational goals of the Skarv field is to push against constraints on flow rates and velocities. While the static estimator is well suited for estimating flow rates, it does not estimate the velocities. Multiphase velocities are complicated, because they depend on not only the mass flow rates of each phase and the pipe diameter, but also the liquid holdup and density profiles along the wells and pipelines. We will not include velocity in this model, but we will make a first step towards estimating velocity by including the liquid holdup. Ideally, we would estimate a holdup *profile* along each well and pipeline, because we then would be able to estimate the velocity profile. However, this would result in a rather complicated model, so we limit ourselves to estimating the total holdup, i.e. the ratio between the volume occupied by liquid and the total volume in each well and the pipeline.

The modelling in the chapter results in a Moving-Horizon Estimation (MHE) formulation. By allowing the wellbore volumes and the flowline volume to accumulate mass, we can introduce additional variables like masses of each individual phase and liquid holdup. This is accomplished by allowing the flowrates into and out of these volumes to be different, and introducing differential mass balance constraints. By doing this, we are assuming that the well (IPR/WPC) and choke models are capable of estimating *instantaneous* flow rates into and out of of the well, respectively. Also, we assume that the instantaneous inflow to the pipeline is the sum of the well outflows, and that the outflow is given by the turret choke model. This assumption is made based on results from the static estimator, and will be discussed further in Chapter 9.

Mass accumulation in the template tubing (manifold jumpers and the manifold itself) is ignored due to the relatively small volume of the pipe segments in this area. Hence, the algebraic mass balance constraints from the static model are kept for the manifold. The general idea is illustrated in Figure 5.1, where the well tubings and pipeline from Figure 1.4 have been replaced by "tanks" which accumulate mass.

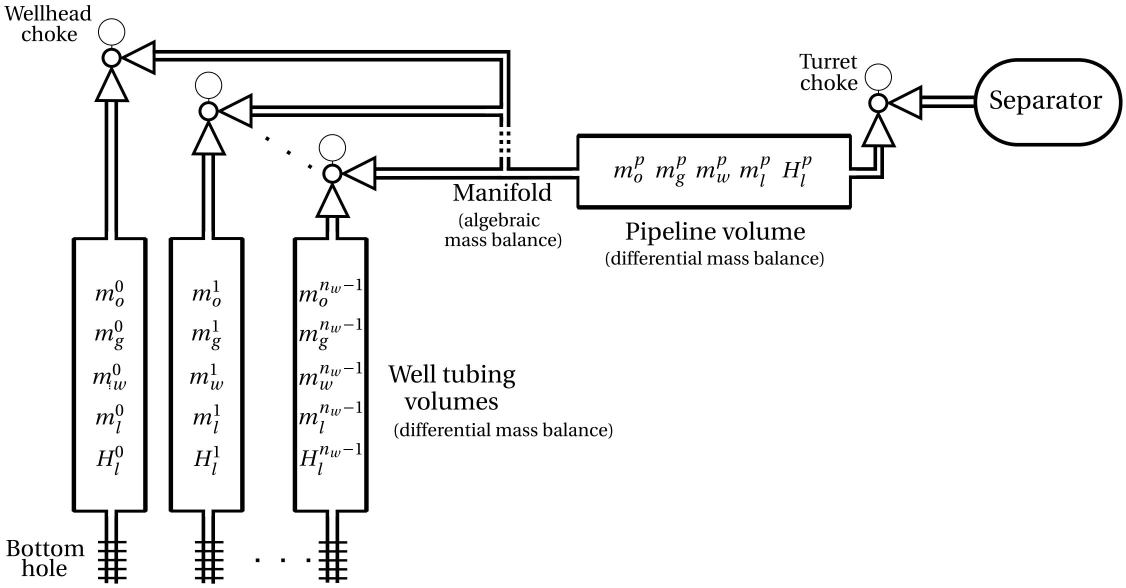

Wellhead choke
Turret choke
Separator
m_o^p m_g^p m_w^p m_l^p H_l^p
Manifold
(algebraic mass balance)
Pipeline volume
(differential mass balance)
m_o^0 m_g^0 m_w^0 m_l^0 H_l^0
m_o^1 m_g^1 m_w^1 m_l^1 H_l^1
$m_o^{n_w-1}$ $m_g^{n_w-1}$ $m_w^{n_w-1}$ $m_l^{n_w-1}$ $H_l^{n_w-1}$
Well tubing volumes
(differential mass balance)
Bottom hole

Figure 5.1: Structure of dynamic model.

5.1 Variable overview

The horizon length (the number of measurements to be considered in the problem) is denoted N. At time index T, we need to consider all the measurements from time index $T-N+1$ up to T. For notational convenience, we define the index set $\mathscr{H}$ as the entire estimation horizon, i.e. $\mathscr{H} = \{k\}_{k=T-N+1}^{T}$. In addition, we define $\mathscr{H}^- = \{k\}_{k=T-N+1}^{T-1}$ as the estimation horizon excluding the last time index, and $\mathscr{H}^+ = \{k\}_{k=T-N+2}^{T}$ as the estimation horizon excluding the first time index. Each time index in the horizon has an associated state variable, which means the complete state vector $\mathbf{x}_T$ is given as

$$\mathbf{x}_T = [\mathbf{x}_{T-N+1|T}^\top, \mathbf{x}_{T-N+2|T}^\top, \ldots, \mathbf{x}_{T|T}^\top]^\top, \tag{5.1}$$

where $\mathbf{x}_{k|T}$ is the state vector corresponding to time index k in the estimation problem to be solved at time index T. $\mathbf{x}_{k|T}$ is given as (the T is omitted to save some space)

$$\mathbf{x}_k = \left[\mathbf{q}_k^\top, \mathbf{p}_k^\top, r_{go,k}^p, r_{wc,k}^p, \tilde{\mathbf{p}}_k^\top, \tilde{\mathbf{T}}_k^\top, \tilde{\mathbf{u}}_k^\top \mathbf{w}_k^\top, \mathbf{v}_k^\top, \mathbf{C}_{v,k}^\top, \overline{\mathbf{T}}_{\text{chk},k}^\top, \overline{\mathbf{T}}_{V,k}^\top, \overline{\boldsymbol{\rho}}_k^\top, \mathbf{V}_{l,k}^\top, \mathbf{H}_{l,k}^\top, \mathbf{m}_k^\top, \mathbf{m}_{l,k}^\top, J_k\right]^\top. \tag{5.2}$$

Here, the variables $\mathbf{q}_k$ through $\overline{\mathbf{T}}_{\text{chk},k}$ and J_k are the same as the variables in the static estimation problem (except we now have N of them). Therefore, we refer to Table 4.3 for an explanation of the nomenclature here. The newly introduced variables $\overline{\mathbf{T}}_{V,k}$ through

$\mathbf{m}_{l,k}$ are explained in Table 5.1 below. The k-subscript is omitted in the table to prevent clutter. The superscript for these new variables denotes the volume; a number i denotes well tubing volume i, while a p denotes the pipeline volume.

Symbol	Definition	Description	Variables	Unit
$\overline{\mathbf{T}}_V$	$[\overline{T}^0, \overline{T}^1, \ldots, \overline{T}^{n_w-1}, \overline{T}^p]^\top$	Avg. temp. in well and pipeline volumes	$N \cdot (n_w + 1)$	[°C]
$\overline{\boldsymbol{\rho}}$	$[\overline{\boldsymbol{\rho}}_o^\top, \overline{\boldsymbol{\rho}}_g^\top, \overline{\boldsymbol{\rho}}_w^\top, \overline{\boldsymbol{\rho}}_l^\top, \overline{\boldsymbol{\rho}}_{tp}^\top]^\top$	Densities (see below)	-	[kg/m^3]
$\overline{\boldsymbol{\rho}}_o$	$[\overline{\rho}_o^0, \overline{\rho}_o^1, \ldots, \overline{\rho}_o^{n_w-1}, \overline{\rho}_o^p]^\top$:	Average oil densities in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg/m^3]
$\overline{\boldsymbol{\rho}}_g$	$[\overline{\rho}_g^0, \overline{\rho}_g^1, \ldots, \overline{\rho}_g^{n_w-1}, \overline{\rho}_g^p]^\top$	Average gas densities in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg/m^3]
$\overline{\boldsymbol{\rho}}_w$	$[\overline{\rho}_w^0, \overline{\rho}_w^1, \ldots, \overline{\rho}_w^{n_w-1}, \overline{\rho}_w^p]^\top$	Average water densities in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg/m^3]
$\overline{\boldsymbol{\rho}}_l$	$[\overline{\rho}_l^0, \overline{\rho}_l^1, \ldots, \overline{\rho}_l^{n_w-1}, \overline{\rho}_l^p]^\top$	Average liquid densities in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg/m^3]
$\overline{\boldsymbol{\rho}}_{tp}$	$[\overline{\rho}_{tp}^0, \overline{\rho}_{tp}^1, \ldots, \overline{\rho}_{tp}^{n_w-1}, \overline{\rho}_{tp}^p]^\top$	Average two-phase densities in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg/m^3]
$\mathbf{V}_l$	$[V_l^0, V_l^1, \ldots, V_l^{n_w-1}, V_l^p]^\top$	Volume occupied by liquid in well and pipeline volumes	$N \cdot (n_w + 1)$	[m^3]
$\mathbf{H}_l$	$[H_l^0, H_l^1, \ldots, H_l^{n_w-1}, H_l^p]$	Liquid holdup in well and pipeline volumes	$N \cdot (n_w + 1)$	[-]
$\mathbf{m}$	$[\mathbf{m}_o^\top, \mathbf{m}_g^\top, \mathbf{m}_w^\top]^\top$	Oil, gas and water mass variables	-	[kg]
$\mathbf{m}_o$	$[m_o^0, m_o^1, \ldots, m_o^{n_w-1}, m_o^p]^\top$	Oil masses in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg]
$\mathbf{m}_g$	$[m_g^0, m_g^1, \ldots, m_g^{n_w-1}, m_g^p]^\top$	Gas masses in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg]
$\mathbf{m}_w$	$[m_w^0, m_w^1, \ldots, m_w^{n_w-1}, m_w^p]^\top$	Water masses in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg]
$\mathbf{m}_l$	$[m_l^0, m_l^1, \ldots, m_l^{n_w-1}, m_l^p]^\top$	Liquid masses in well and pipeline volumes	$N \cdot (n_w + 1)$	[kg]

Table 5.1: Variable overview, dynamic flow estimation problem (remaining variables are explained in Table 4.3).

The exclusion of $\mathbf{m}_l$ from $\mathbf{m}$ is intentional; as we will see in the next section, the oil, gas and water masses in $\mathbf{m}$ are differential variables, while the liquid masses in $\mathbf{m}_l$ are algebraic.

The total number of variables in Table 5.1 is $12N \cdot (n_w + 1)$. In addition, the number of variables in Table 4.3 amount to $N \cdot (30n_w + 27)$ when we consider the estimation horizon. This gives a total of $n_{\mathbf{x}} = N \cdot (42n_w + 39)$ variables. This means the number of variables scales linearly with both the number of wells n_w and the estimation horizon N.

5.2 Differential constraints

5.2.1 Mass accumulation in well tubing and pipeline

The differential equations for mass accumulation follow from the conservation of mass, where the control volume is defined as the entire well tubing volume. The time derivative of the accumulated mass inside the tubing equals the difference between the mass inflow and the mass outflow. To convert standard condition volume flow (Sm^3/h) to mass flow (kg/s), the densities at standard conditions are used to obtain $w_o = \frac{\rho_o^{\text{std}}}{3600} q_o$, $w_g = \frac{\rho_g^{\text{std}}}{3600} q_g$ and $w_w = \frac{\rho_w^{\text{std}}}{3600} q_w$. Note that in this section, the w's denote mass flow rates and not model errors. Then, the differential equations become

$$\dot{m}_o^i = w_o^i - w_o^{i+n_w}, \ \ \forall\ i \in \mathcal{W} \qquad \text{[kg/s]} \qquad (5.3)$$

$$\dot{m}_g^i = w_g^i - w_g^{i+n_w}, \ \ \forall\ i \in \mathcal{W} \qquad \text{[kg/s]} \qquad (5.4)$$

$$\dot{m}_w^i = w_w^i - w_w^{i+n_w}, \ \ \forall\ i \in \mathcal{W} \qquad \text{[kg/s]} \qquad (5.5)$$

for the wells, and

$$\dot{m}_o^p = w_o^{3n_w} - w_o^{3n_w+1}, \qquad \text{[kg/s]} \qquad (5.6)$$

$$\dot{m}_g^p = w_g^{3n_w} - w_g^{3n_w+1}, \qquad \text{[kg/s]} \qquad (5.7)$$

$$\dot{m}_w^p = w_w^{3n_w} - w_w^{3n_w+1}, \qquad \text{[kg/s]} \qquad (5.8)$$

for the pipeline. The dot notation denotes the time derivative, i.e. $\dot{m} = \mathrm{d}m/\mathrm{d}t$. To ease implementation, a simple Euler discretization is performed to obtain a set of discrete-time algebraic constraints. For each time index k, there are $n_w + 1$ masses of each phase to keep track of (n_w well tubing volumes and one pipeline volume), so we recall the vectors

$$\mathbf{m}_{o,k} = \begin{bmatrix} m_{o,k}^0 \\ \vdots \\ m_{o,k}^{n_w-1} \\ m_{o,k}^p \end{bmatrix}, \quad \mathbf{m}_{g,k} = \begin{bmatrix} m_{g,k}^0 \\ \vdots \\ m_{g,k}^{n_w-1} \\ m_{g,k}^p \end{bmatrix}, \quad \text{and} \quad \mathbf{m}_{w,k} = \begin{bmatrix} m_{w,k}^0 \\ \vdots \\ m_{w,k}^{n_w-1} \\ m_{w,k}^p \end{bmatrix}, \qquad (5.9)$$

where $m_{o,k}^i$ is the mass of oil in well tubing i at time index k, $m_{o,k}^p$ is the mass of oil in the pipeline at time index k, and similarly for the gas and water phases. These three vectors are collected in a single mass vector $\mathbf{m}_k = [\mathbf{m}_{o,k}^\top, \mathbf{m}_{g,k}^\top, \mathbf{m}_{w,k}^\top]^\top$. We also define the vectors

$$\Delta\mathbf{q}_{o,k} = \begin{bmatrix} q_{o,k}^0 - q_{o,k}^{n_w} \\ \vdots \\ q_{o,k}^{n_w-1} - q_{o,k}^{2n_w-1} \\ q_{o,k}^{3n_w} - q_{o,k}^{3n_w+1} \end{bmatrix}, \Delta\mathbf{q}_{g,k} = \begin{bmatrix} q_{g,k}^0 - q_{g,k}^{n_w} \\ \vdots \\ q_{g,k}^{n_w-1} - q_{g,k}^{2n_w-1} \\ q_{g,k}^{3n_w} - q_{g,k}^{3n_w+1} \end{bmatrix}, \text{and } \Delta\mathbf{q}_{w,k} = \begin{bmatrix} q_{w,k}^0 - q_{w,k}^{n_w} \\ \vdots \\ q_{w,k}^{n_w-1} - q_{w,k}^{2n_w-1} \\ q_{w,k}^{3n_w} - q_{w,k}^{3n_w+1} \end{bmatrix}, \qquad (5.10)$$

where $\Delta \mathbf{q}_{o,k}$ is a vector of net oil (standard) volume inflows to the well and pipeline volumes (and similarly for gas and water). These are also collected in the vector $\Delta \mathbf{q}_k = [\Delta \mathbf{q}_{o,k}^\top, \Delta \mathbf{q}_{g,k}^\top, \Delta \mathbf{q}_{w,k}^\top]^\top$. Then, we can write the discretized mass accumulation constraints compactly as

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \Delta t \cdot \mathbf{D} \Delta \mathbf{q}_k, \ \forall \ k \in \mathscr{H}^-, \tag{5.11}$$

where

$$\mathbf{D} = \frac{1}{3600} \begin{bmatrix} \rho_o^{\text{std}} \mathbf{I}_{n_w+1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \rho_g^{\text{std}} \mathbf{I}_{n_w+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \rho_w^{\text{std}} \mathbf{I}_{n_w+1} \end{bmatrix} \tag{5.12}$$

is the matrix needed to convert standard volume flow rates (Sm^3/h) to mass flow rates (kg/s). The liquid masses are taken as the sum of the oil and water masses;

$$\mathbf{m}_{l,k} = \mathbf{m}_{o,k} + \mathbf{m}_{w,k}, \ \forall \ k \in \mathscr{H}, \tag{5.13}$$

where $\mathbf{m}_{l,k} = [m_{l,k}^0, \ldots, m_{l,k}^{n_w-1}, m_{l,k}^p]^\top$ is a vector of all the liquid masses in the well volumes and the pipeline volume. Note that the liquid masses are *not* differential variables.

5.3 Algebraic constraints

This section describes the algebraic constraints in the dynamic formulation. Some of the algebraic constraints are inherited from the static model, so for these the nomenclature will not be explained extensively (refer to Chapter 4).

5.3.1 Mass balance constraints

Wells

For all edges associated to one well, the GOR and water cut should be the same, and the liquid rate should be the sum of the oil and water rates. This is imposed by the following constraints:

$$q_{g,k}^{i+j} = r_{go}^i q_{o,k}^{i+j}, \qquad \forall \ i \in \mathscr{W}, j \in \{0, n_w, 2n_w\}, k \in \mathscr{H}, \tag{5.14}$$

$$q_{w,k}^{i+j} = r_{wc}^i q_{l,k}^{i+j}, \qquad \forall \ i \in \mathscr{W}, j \in \{0, n_w, 2n_w\}, k \in \mathscr{H}, \tag{5.15}$$

$$q_{l,k}^i = q_{o,k}^i + q_{w,k}^i, \qquad \forall \ i \in \{i\}_{i=0}^{2n_w-1}, k \in \mathscr{H}. \tag{5.16}$$

Manifold

Ordinary mass balance constraints are used to calculate the flow rates from the wells and into the manifold. This means we have to set the flow rates through the manifold jumpers equal to the flow rates through the choke valve:

$$q_{o,k}^{i+n_w} - q_{o,k}^{i+2n_w} = 0, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H} \tag{5.17}$$
$$q_{g,k}^{i+n_w} - q_{g,k}^{i+2n_w} = 0, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H} \tag{5.18}$$
$$q_{w,k}^{i+n_w} - q_{w,k}^{i+2n_w} = 0, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}. \tag{5.19}$$

Then we let the pipeline inflow be equal to the sum of the flow rates in the manifold jumpers:

$$q_{o,k}^{3n_w} = \sum_{i \in \mathscr{W}} q_{o,k}^{2n_w+i}, \qquad \forall\ k \in \mathscr{H}, \tag{5.20}$$
$$q_{g,k}^{3n_w} = \sum_{i \in \mathscr{W}} q_{g,k}^{2n_w+i}, \qquad \forall\ k \in \mathscr{H}, \tag{5.21}$$
$$q_{w,k}^{3n_w} = \sum_{i \in \mathscr{W}} q_{w,k}^{2n_w+i} \qquad \forall\ k \in \mathscr{H}. \tag{5.22}$$

Pipeline phase split

Having calculated the oil, gas and water rates for pipeline inflow through (5.20)-(5.22), we can calculate the pipeline GOR, water cut and liquid rate:

$$q_{g,k}^{3n_w} = r_{go,k}^{p} q_{o,k}^{3n_w}, \qquad \forall\ k \in \mathscr{H}, \tag{5.23}$$
$$q_{w,k}^{3n_w} = r_{wc,k}^{p} q_{l,k}^{3n_w}, \qquad \forall\ k \in \mathscr{H}, \tag{5.24}$$
$$q_{l,k}^{3n_w} = q_{o,k}^{3n_w} + q_{w,k}^{3n_w} \qquad \forall k \in \mathscr{H}. \tag{5.25}$$

The pipeline outflow is calculated in terms of liquid rate only, so to obtain a correct phase split through the turret choke, we must also include GOR, water cut and liquid rate constraints for the turret choke edge:

$$q_{g,k}^{3n_w+1} = r_{go,k}^{p} q_{o,k}^{3n_w+1}, \qquad \forall\ k \in \mathscr{H}, \tag{5.26}$$
$$q_{w,k}^{3n_w+1} = r_{wc,k}^{p} q_{l,k}^{3n_w+1}, \qquad \forall\ k \in \mathscr{H}, \tag{5.27}$$
$$q_{l,k}^{3n_w+1} = q_{o,k}^{3n_w} + q_{w,k}^{3n_w+1}, \qquad \forall\ k \in \mathscr{H}. \tag{5.28}$$

5.3.2 Well inflow and outflow

As seen in Figure 8.3, The IPR and WPC curves appear to give good predictions of well inflow, while the choke models seem to give good predictions of outflow. Therefore, these are used to calculate the liquid flow rate into and out of the well tubing. As with the static model, we express the constraints in terms of model error:

$$w^i_{\text{ipr},k} = p^i_k - \Phi^i_{\text{ipr}}(q^i_{l,k}), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.29}$$

$$w^i_{\text{wpc},k} = p^{n_w+i}_k - \Phi_{\text{wpc}}(q^i_{l,k}), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.30}$$

$$w^i_{\text{chk},k} = p^{2n_w+i}_k - \Phi^i_{\text{chk}}(q^{n_w+i}_{l,k}, p^{n_w+i}_k, \overline{T}^i_{\text{chk},k}, C^i_{v,k}), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}. \tag{5.31}$$

The choke model includes auxiliary constraints for the flow coefficient $C_{v,k}$ and the average temperature $\overline{T}_{\text{chk},k}$:

$$C^i_{v,k} = \Phi^i_{C_v}(\tilde{u}^i_k), \qquad \forall\ k \in \mathscr{H}, \tag{5.32}$$

$$\overline{T}^i_{\text{chk},k} = \frac{1}{2}\left(\tilde{T}^{n_w+i}_k + \tilde{T}^{2n_w+i}_k\right), \qquad \forall\ k \in \mathscr{H}. \tag{5.33}$$

These constraints are all inherited from the static model, so the nomenclature will not be explained here (refer to Sections 4.5.2 and 4.5.3).

5.3.3 Well tubing and pipeline pressure drop

For the well tubing and pipeline pressure drops, frictional and accelerational effects are assumed negligable. This assumption is founded on obervations during OLGA simulations of the Tilje template, where the hydrostatic contribution has consistently been dominant (> 95 % of the total pressure drop). Although this is not unreasonable for Tilje, we may not be so fortunate for other templates/fields. Therefore, an extended model which includes friction is presented in Appendix B. The well tubing hydrostatic pressure drop constraints are given by

$$w^i_{\Delta p,k} = (p^i_k - p^{i+n_w}_k) - \overline{\rho}^i_{tp,k} g h^i \cdot 10^{-5}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.34}$$

where $w^i_{\Delta p,k}$ is the model error, $(p^i_k - p^{i+n_w}_k)$ is the actual (reconciled) pressure drop, and $\overline{\rho}^i_{tp,k} g h^i$ is the hydrostatic pressure drop predicted by the model in Pascals. The 10^{-5} multiplier is necessary to convert the pressure drop from Pascals to bar. $\overline{\rho}^i_{tp,k}$ is the *two-phase* density, which describes the average density of the fluid mixture in the well tubing. $g = 9.81$ m/s^2 is the acceleration of gravity, and h^i is the total elevation change from the bottom hole to the wellhead. h^i can be calculated from the contents of the well geometry file from OLGA (see Appendix C.1). For the pipeline, the constraint becomes

$$w^p_{\Delta p,k} = (p^{3n_w}_k - p^{3n_w+1}_k) - \overline{\rho}^p_{tp,k} g h^p \cdot 10^{-5}, \qquad \forall\ k \in \mathscr{H}, \tag{5.35}$$

i.e. the bottom hole and wellhead pressures have been replaced by the manifold pressure and the pressure at the top of the riser. $w^p_{\Delta p,k}$ is the model error for the pipeline pressure drop model, $\overline{\rho}^p_{tp,k}$ is the two-phase density of the fluids in the pipeline, and h^p is the total elevation change of the pipeline. To calculate the two-phase densities, some additional constraints are necessary. These are described in the sections below.

Fluid properties

The pressure model uses fluid densities to determine the hydrostatic pressure drop. To simplify the model, average densities are used, which are obtained by evaluating the PVT tables at the average pressure and temperature in the well. The well average pressure is defined as

$$\overline{p}^i_k = \frac{k^i_{\overline{p}}}{2}\left(p^i_k + p^{i+n_w}_k\right), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.36}$$

where p^i_k is the bottom hole pressure and $p^{i+n_w}_k$ is the wellhead pressure. $k^i_{\overline{p}}$ is a constant determined by the well geometry, whose calculation is shown in Appendix C.1. The pipeline average pressure is taken as

$$\overline{p}^p_k = \frac{k^p_{\overline{p}}}{2}\left(p^{3n_w}_k + p^{3n_w+1}_k\right), \qquad \forall\ k \in \mathscr{H}. \tag{5.37}$$

Further, the well average temperature is taken as

$$\overline{T}^i_k = \frac{1}{2}\left(\tilde{T}^i_k + \tilde{T}^{i+n_w}_k\right), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}. \tag{5.38}$$

where $\tilde{T}^i_k$ is the measured bottom hole temperature and $\tilde{T}^{i+n_w}_k$ is the measured wellhead temperature, and the pipeline average temperature is taken as

$$\overline{T}^p_k = \frac{1}{2}\left(\tilde{T}^{3n_w}_k + \tilde{T}^{3n_w+1}_k\right), \qquad \forall\ k \in \mathscr{H}. \tag{5.39}$$

Using the average pressures and temperatures, the spline approximated PVT tables are evaluated to obtain the required fluid properties:

Average oil densities: $\overline{\rho}^i_{o,k} = \Phi_{\rho_o}(\overline{p}^i_k, \overline{T}^i_k) \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H},$ (5.40)

$\overline{\rho}^p_{o,k} = \Phi_{\rho_o}(\overline{p}^p_k, \overline{T}^p_k) \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}.$ (5.41)

Average gas densities: $\overline{\rho}^i_{g,k} = \Phi_{\rho_g}(\overline{p}^i_k, \overline{T}^i_k) \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H},$ (5.42)

$\overline{\rho}^p_{g,k} = \Phi_{\rho_g}(\overline{p}^p_k, \overline{T}^p_k) \qquad \forall\ k \in \mathscr{H},$ (5.43)

Average water densities: $\overline{\rho}^i_{w,k} = \Phi_{\rho_w}(\overline{p}^i_k, \overline{T}^i_k) \qquad \forall\ k \in \mathscr{H}.$ (5.44)

$\overline{\rho}^p_{w,k} = \Phi_{\rho_w}(\overline{p}^p_k, \overline{T}^p_k) \qquad \forall\ k \in \mathscr{H}.$ (5.45)

The average liquid densities are taken as weighted sums of the oil and water densities, using the water cuts:

$$\overline{\rho}^i_{l,k} = r_{wc}\overline{\rho}^i_{w,k} + (1 - r_{wc})\overline{\rho}^i_{o,k} \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.46}$$

$$\overline{\rho}^p_{l,k} = r^p_{wc,k}\overline{\rho}^p_{w,k} + (1 - r^p_{wc,k})\overline{\rho}^p_{o,k} \qquad \forall\ k \in \mathscr{H}. \tag{5.47}$$

Using the water cut for weighting is not entirely accurate, since the *in-situ* water cut may be different than r_{wc} which is defined at standard conditions. However, since the Tilje wells produce a relatively small amount of water, one would expect that the introduced error will be small.

Liquid holdups

The liquid holdup (liquid volume fraction) H_l is determined by first calculating the liquid volume present in the volume, V_l, using the accumulated liquid mass and the average liquid density:

$$\text{Wells:} \qquad \overline{\rho}^i_{l,k} V^i_{l,k} = m^i_{l,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.48}$$

$$\text{Pipeline:} \qquad \overline{\rho}^p_{l,k} V^p_{l,k} = m^p_{l,k}. \qquad \forall\ k \in \mathscr{H}. \tag{5.49}$$

Then, the well liquid holdup is calculated using (5.48) together with the total volume of the well tubing V^i_p:

$$H^i_{l,k} V^i_p = V^i_{l,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.50}$$

and the pipeline holdup is calculated using (5.49) with the total volume of the pipeline V^p_p:

$$H^p_{l,k} V^p_p = V^p_{l,k}, \qquad \forall\ k \in \mathscr{H}. \tag{5.51}$$

The volumes of the well tubings and pipeline are calculated from information about section lengths and diameters in the geometry files from OLGA.

Two-phase densities

The two-phase densities are calculated by using the liquid holdup to calculate a weighted sum of the average liquid and gas densities. Thus, it represents the average density of the mixture in the volume:

$$\text{Wells:} \qquad \overline{\rho}^i_{tp,k} = H^i_{l,k} \overline{\rho}^i_{l,k} + (1 - H^i_{l,k}) \overline{\rho}^i_{g,k}. \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{5.52}$$

$$\text{Pipeline:} \qquad \overline{\rho}^p_{tp,k} = H^p_{l,k} \overline{\rho}^p_{l,k} + (1 - H^p_{l,k}) \overline{\rho}^p_{g,k}. \qquad \forall\ k \in \mathscr{H}. \tag{5.53}$$

The two-phase densities are used in (5.34) and (5.35), respectively, to complete the pressure drop model.

5.3.4 Pipeline inflow/outflow

The pipeline inflow is given by the mass balance constraints, i.e. the sum of the well flow rates. The pipeline outflow is estimated by the turret choke model. Although the pipeline VLP could potentially be used to calculate pipeline in- or outflow, we will see in Chapter 8 that it is not capable of predicting flow rates well for the operational conditions at Skarv. Therefore, we choose to leave it out of the dynamic model. The turret choke model is given by

$$w^t_{\text{chk},k} = p_k^{3n_w+2} - \Phi^t_{\text{chk}}(q_{l,k}^{3n_w}, p_k^{3n_w+1}, \overline{T}^t_{\text{chk},k}, r^p_{go,k}, r^p_{wc,k}, C^t_{v,k}), \qquad \forall\; k \in \mathcal{H}, \tag{5.54}$$

and the auxiliary constraints

$$C^t_{v,k} = \Phi^t_{C_v}(\tilde{u}^t_k), \qquad \forall\; k \in \mathcal{H}, \tag{5.55}$$

$$\overline{T}^t_{\text{chk},k} = \frac{1}{2}\left(\tilde{T}_k^{3n_w+1} + \tilde{T}_k^{3n_w+2}\right), \qquad \forall\; k \in \mathcal{H}. \tag{5.56}$$

Refer to Section 4.5.3 for an explanation of nomenclature.

5.4 Formulation of the dynamic flow estimation NLP

5.4.1 Objective function

The objective function for the dynamic NLP to be solved at time index T is given by

$$\begin{aligned} J(\mathbf{x}) = {} & \left\|\mathbf{m}^-_{T-N+1} - \mathbf{m}_{T-N+1}\right\|^2_{\mathbf{P}_{\mathbf{m}(0)}} + \left\|\mathbf{q}_{l,T-N} - \mathbf{q}_{l,T-N+1}\right\|^2_{\mathbf{P}_{\mathbf{q}(0)}} \\ & + \sum_{k\in\mathcal{H}^+} \left\|\mathbf{q}_{l,k} - \mathbf{q}_{l,k-1}\right\|^2_{\mathbf{P}_{\mathbf{q}}} + \sum_{k\in\mathcal{H}} \left\{\|\mathbf{w}_k\|^2_{\mathbf{Q}^{-1}} + \|\mathbf{v}_k\|^2_{\mathbf{R}^{-1}}\right\} \end{aligned} \tag{5.57}$$

We will explain this term by term.

- The first term in (5.57) is the *prior weighing* term, which is there to include information gathered in previous iterations, i.e. an approximated arrival cost (cf. Section 2.4). $\mathbf{m}^-_{T-N+1}$ is an *a priori* estimate for the masses at the start of the estimation horizon, which is calculated based on the solution of the preceding estimation problem (the one solved at time index $T-1$). We use (2.15) to calculate $\mathbf{m}^-_{T-N+1}$ as
$$\mathbf{m}^-_{T-N+1} = \mathbf{m}_{T-N} + \Delta t \mathbf{D} \Delta \mathbf{q}_{T-N}, \tag{5.58}$$
where $\mathbf{m}_{T-N}$ is the estimated mass at time index $T-N$, and $\mathbf{q}_{l,T-N}$ are the estimated liquid flow rates at time index $T-N$ (both from of the solution of the estimation problem solved at time index $T-1$). The first term in (5.57) will thus penalize deviations from this *a priori* estimate. The weights in $\mathbf{P}_{\mathbf{m}(0)}$ decides how much deviation will be allowed; larger weights will allow less deviation from the *a priori* estimate.

- The second term in (5.57) is a flow regularization term similar to the first term in the static estimator objective function (4.27). $\mathbf{q}_{l,T-N}$ is taken as the liquid rate at the start of the trajectory from the solution of the estimation problem solved at time index $T-1$.

- The third term is also a flow regularization term, i.e. a continuation of the second term throughout the liquid rate trajectory. Penalizing the difference between "adjacent" liquid rates in the trajectory is (as discussed in Section 4.7.1) an attempt to discourage the solver to jump from one local optimum to another. In addition, the term can be used for *smoothing* the liquid rate trajectory if the weights in $\mathbf{P_q}$ are selected sufficiently large.

- The fourth and final term in (5.57) is equivalent to the second and third terms in the static estimator objective function; it penalizes model errors and pressure measurement errors. However, this objective function has one term for each time index in the estimation horizon (as opposed to the static estimator which only needs one). We will not go into detail here, as we have discussed these terms and the weighting matrices $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ in Section 4.7.1.

5.4.2 Measurements

Like the case was for the static estimator, we include measurements by using a set of linear equality constraints:

$$\mathbf{0} = \mathbf{y}_k - [\tilde{p}_k^\top, \tilde{\mathbf{T}}_k^\top, \tilde{\mathbf{u}}_k^\top]^\top, \;\; \forall k \in \mathscr{H}. \tag{5.59}$$

5.4.3 Starting point

The starting point $\mathbf{x}_{\text{init}}$ is taken similarly to the static estimator, except $\mathbf{x}_{\text{init}}$ is now a state *trajectory* as opposed to a single state vector. That is, the starting point for the problem to be solved at time index T is

$$\mathbf{x}_{\text{init}|T} = [\mathbf{x}_{\text{init},T-N+1}^\top, \mathbf{x}_{\text{init},T-N+2}^\top, \dots, \mathbf{x}_{\text{init},T}^\top]^\top. \tag{5.60}$$

A good starting point for the state vector at time index k, i.e. $\mathbf{x}_{\text{init},k}$, would be the corresponding state estimate from the previous time index, which we denote $\mathbf{x}_{k|T-1}^*$. As with the static estimator, we also update the starting point with the latest estimates and measurements through the selection matrices $\mathbf{S_x}$ and $\mathbf{S_y}$. This results in the following expression for $\mathbf{x}_{\text{init},k}$:

$$\mathbf{x}_{\text{init},k} = \mathbf{S_x}\mathbf{x}_{k|T-1}^* + \mathbf{S_y}\mathbf{y}_k, \;\; \forall \; k \in \mathscr{H}^-. \tag{5.61}$$

Note that $\mathbf{S_x}$ is now based on the new state vector $\mathbf{x}$ defined in (5.2). We have only provided starting points up to time index $T-1$ here, since the last state vector in the trajectory was not estimated in the previous time index. Therefore we reuse $\mathbf{x}_{T-1|T-1}$ as a starting point for $\mathbf{x}_{\text{init},T}$;

$$\mathbf{x}_{\text{init},T} = \mathbf{S}_{\mathbf{x}}\mathbf{x}^{*}_{T-1|T-1} + \mathbf{S}_{\mathbf{y}}\mathbf{y}_{T}. \tag{5.62}$$

5.4.4 Estimating the initial holdup and masses

Since the optimization problem now includes mass and holdup variables, we need to provide some initial values for these. This is accomplished by solving the *static* optimization problem (Problem 4.1) to obtain a set of flow rate estimates, and then solving the pressure model for masses and holdup based on these flow rates. This procedure is described in Appendix C.2.1. The obtained information is passed to the optimization problem through the starting point: Let $\mathbf{x}^{S}$ be the solution of Problem 4.1, and let $\mathbf{m}^{S}, \mathbf{H}_{l}^{S}$ be the calculated masses and holdups. Then, we choose the following starting point for the very first problem to be solved;

$$\mathbf{x}_{\text{init},k} = \mathbf{S}_{\mathbf{y}}\mathbf{y}_{T_0} + \mathbf{S}_{\mathbf{x}}^{S}\mathbf{x}^{S} + \mathbf{S}_{\mathbf{m}}\begin{bmatrix}\mathbf{m}^{S}\\ \mathbf{H}_{l}^{S}\end{bmatrix}, \ \forall\ k \in \mathscr{H}, \tag{5.63}$$

where $\mathbf{S}_{\mathbf{x}}^{S}$ and $\mathbf{S}_{\mathbf{m}}$ are selection matrices much like $\mathbf{S}_{\mathbf{x}}$ and $\mathbf{S}_{\mathbf{y}}$; $\mathbf{S}_{\mathbf{m}}$ maps the masses and holdups to the correct variables in $\mathbf{x}_{\text{init},k}$, while $\mathbf{S}_{\mathbf{x}}^{S}$ maps the available variables from $\mathbf{x}^{S}$ to $\mathbf{x}_{\text{init},k}$. T_0 is the very first time index, i.e. $\mathbf{y}_{T_0}$ is the first available measurement.

5.4.5 Summary - the dynamic flow estimation NLP

Problem 5.1: Dynamic flow estimation NLP

$$\underset{\mathbf{x}}{\text{minimize}}\ J$$

subject to

objective constraint:

$$J \geq \left\|\mathbf{m}^{-}_{T-N+1} - \mathbf{m}_{T-N+1}\right\|^2_{\mathbf{P}_{\mathbf{m}(0)}} + \left\|\mathbf{q}_{l,T-N} - \mathbf{q}_{l,T-N+1}\right\|^2_{\mathbf{P}_{\mathbf{q}(0)}} + \sum_{k\in\mathscr{H}^+} \left\|\mathbf{q}_{l,k} - \mathbf{q}_{l,k-1}\right\|^2_{\mathbf{P}_{\mathbf{q}}} + \sum_{k\in\mathscr{H}} \left\{\|\mathbf{w}_k\|^2_{\mathbf{Q}^{-1}} + \|\mathbf{v}_k\|^2_{\mathbf{R}^{-1}}\right\}$$

Differential mass balance constraints:

mass accumulation:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \Delta t \cdot \mathbf{D}\Delta\mathbf{q}_k,\ \forall\ k \in \mathscr{H}^-$$

$$\mathbf{m}_{l,k} = \mathbf{m}_{o,k} + \mathbf{m}_{w,k},\ \forall\ k \in \mathscr{H}$$

Algebraic mass balance constraints:

well GORs:

$$0 = q^{i+j}_{g,k} - r^i_{go} q^{i+j}_{o,k}, \forall\ i \in \mathscr{W}, j \in \{0, n_w, 2n_w\}, k \in \mathscr{H}$$

well water cuts:

$$0 = q^{i+j}_{w,k} - r^i_{wc} q^{i+j}_{l,k}, \forall\ i \in \mathscr{W}, j \in \{0, n_w, 2n_w\}, k \in \mathscr{H}$$

well liq. rates:

$$0 = q^i_{l,k} - q^i_{o,k} + q^i_{w,k}, \forall\ i \in \{i\}^{2n_w-1}_{i=0}, k \in \mathscr{H}$$

manifold jumpers:

$$0 = q^{i+n_w}_{o,k} - q^{i+2n_w}_{o,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

$$0 = q^{i+n_w}_{g,k} - q^{i+2n_w}_{g,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

$$0 = q^{i+n_w}_{w,k} - q^{i+2n_w}_{w,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

manifold mass balance:

$$0 = q^{3n_w}_{o,k} - \sum_{i\in\mathscr{W}} q^{2n_w+i}_{o,k}, \forall k \in \mathscr{H}$$

$$0 = q^{3n_w}_{g,k} - \sum_{i\in\mathscr{W}} q^{2n_w+i}_{g,k}, \forall k \in \mathscr{H}$$

$$0 = q^{3n_w}_{w,k} - \sum_{i\in\mathscr{W}} q^{2n_w+i}_{w,k}, \forall k \in \mathscr{H}$$

pipeline phase split:

$$0 = q^{3n_w+i}_{g,k} - r^p_{go,k} q^{3n_w+i}_{o,k}, \forall i \in \{0, 1\}, k \in \mathscr{H}$$

$$0 = q^{3n_w+i}_{w,k} - r^p_{wc,k} q^{3n_w+i}_{l,k}, \forall i \in \{0, 1\}, k \in \mathscr{H}$$

$$0 = q^{3n_w+i}_{l,k} - q^{3n_w+i}_{o,k} + q^{3n_w+i}_{w,k}, \forall i \in \{0, 1\}, k \in \mathscr{H}$$

Well in/outflow:

IPRs:

$$w^i_{\text{ipr},k} = p^i_k - \Phi^i_{\text{ipr}}(q^i_{l,k}), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

WPCs:

$$w^i_{\text{wpc},k} = p^{n_w+i}_k - \Phi_{\text{wpc}}(q^i_{l,k}), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

wellhead chokes:

$$w^i_{\text{chk},k} = p^{2n_w+i}_k - \Phi^i_{\text{chk}}(q^{n_w+i}_{l,k}, p^{n_w+i}_k, \overline{T}^i_{\text{chk},k}, C^i_{v,k}), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

$$C^i_{v,k} = \Phi^i_{C_v}(\tilde{u}^i_k), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

$$\overline{T}^i_{\text{chk},k} = \frac{1}{2}\tilde{T}^{n_w+i}_k + \frac{1}{2}\tilde{T}^{2n_w+i}_k, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

Dynamic flow estimation NLP (cont.)

Pipeline outflow:

turret choke:
$$w^t_{\text{chk},k} = p_k^{3n_w+2} - \Phi^t_{\text{chk}}(q_{l,k}^{3n_w}, p_k^{3n_w+1}, \overline{T}^t_{\text{chk},k}, r^p_{go,k}, r^p_{wc,k}, C^t_{v,k}), \forall k \in \mathscr{H}$$
$$C^t_{v,k} = \Phi^t_{C_v}(\tilde{u}^t_k), \forall\ k \in \mathscr{H}$$
$$\overline{T}^t_{\text{chk},k} = \frac{1}{2}\tilde{T}_k^{3n_w+1} + \frac{1}{2}\tilde{T}_k^{3n_w+2}, \forall\ k \in \mathscr{H}$$

Well and pipeline pressure drop models:

hydrostatic pressure drop:
$$w^i_{\Delta p,k} = (p^i_k - p_k^{i+n_w}) - \overline{\rho}^i_{tp,k} g h^i \cdot 10^{-5}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$w^p_{\Delta p,k} = (p_k^{3n_w} - p_k^{3n_w+1}) - \overline{\rho}^p_{tp,k} g h^p \cdot 10^{-5}, \forall\ k \in \mathscr{H}$$

well avg. pressures:
$$\overline{p}^i_k = k^i_{\overline{p}}\left(\frac{1}{2}p^i_k + \frac{1}{2}p_k^{i+n_w}\right), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

pipeline avg. pressure:
$$\overline{p}^p_k = k^p_{\overline{p}}\left(\frac{1}{2}p_k^{3n_w} + \frac{1}{2}p_k^{3n_w+1}\right), \forall\ k \in \mathscr{H}$$

well avg. temperatures:
$$\overline{T}^i_k = \frac{1}{2}\tilde{T}^i_k + \frac{1}{2}\tilde{T}_k^{i+n_w}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$

pipeline avg. temperature:
$$\overline{T}^p_k = \frac{1}{2}\tilde{T}_k^{3n_w} + \frac{1}{2}\tilde{T}_k^{3n_w+1}, \forall\ k \in \mathscr{H}$$

average oil densities:
$$\overline{\rho}^i_{o,k} = \Phi_{\rho_o}(\overline{p}^i_k, \overline{T}^i_k), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$\overline{\rho}^p_{o,k} = \Phi_{\rho_o}(\overline{p}^p_k, \overline{T}^p_k), \forall\ k \in \mathscr{H}$$

average gas densities:
$$\overline{\rho}^i_{g,k} = \Phi_{\rho_g}(\overline{p}^i_k, \overline{T}^i_k), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$\overline{\rho}^p_{g,k} = \Phi_{\rho_g}(\overline{p}^p_k, \overline{T}^p_k), \forall\ k \in \mathscr{H}$$

average water densities:
$$\overline{\rho}^i_{w,k} = \Phi_{\rho_w}(\overline{p}^i_k, \overline{T}^i_k), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$\overline{\rho}^p_{w,k} = \Phi_{\rho_w}(\overline{p}^p_k, \overline{T}^p_k), \forall\ k \in \mathscr{H}$$

average liquid densities:
$$\overline{\rho}^i_{l,k} = r_{wc}\overline{\rho}^i_{w,k} + (1 - r_{wc})\overline{\rho}^i_{o,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$\overline{\rho}^p_{l,k} = r^p_{wc,k}\overline{\rho}^p_{w,k} + (1 - r^p_{wc,k})\overline{\rho}^p_{o,k}, \forall\ k \in \mathscr{H}$$

liquid volumes:
$$0 = m^i_{l,k} - \overline{\rho}^i_{l,k} V^i_{l,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$0 = m^p_{l,k} - \overline{\rho}^p_{l,k} V^p_{l,k}, \forall\ k \in \mathscr{H}$$

liquid holdups:
$$0 = V^i_{l,k} - V^i_p H^i_{l,k}, \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$0 = V^p_{l,k} - V^p_p H^p_{l,k}, \forall\ k \in \mathscr{H}$$

two-phase densities:
$$\overline{\rho}^i_{tp,k} = H^i_{l,k}\overline{\rho}^i_{l,k} + (1 - H^i_{l,k}), \forall\ i \in \mathscr{W}, k \in \mathscr{H}$$
$$\overline{\rho}^p_{tp,k} = H^p_{l,k}\overline{\rho}^p_{l,k} + (1 - H^p_{l,k}), \forall\ k \in \mathscr{H}$$

Measurements:
$$\mathbf{0} = \mathbf{y}_k - [\tilde{p}_k^\top, \tilde{\mathbf{T}}_k^\top, \tilde{\mathbf{u}}_k^\top]^\top,\ \ \forall k \in \mathscr{H}$$

Variable bounds:
$$\underline{\mathbf{x}} \le \mathbf{x} \le \overline{\mathbf{x}}$$

Chapter 6

Model calibration

Both in the static and dynamic formulations, there will inevitably be some mismatch between reality and the model used for estimation, leading to an estimation error. *Calibration* is the process of identifying and correcting for these errors. The calibration process may also give some useful information about how much error there is in each model, which can be used for weighting "good" models more in the NLP objective function. The calibration methods described in this chapter are crude, but sufficient for their purpose. After an introductory section, we will discuss the MPFM and routing network, i.e. how the template must be routed to calibrate each well and the common pipeline/turret choke. Then, the next few sections will discuss the calibration of each model (IPR, WPC, choke, pipeline and turret choke). Finally, some ideas are presented for further taking advantage of calibration data in the optimization problem, by transferring some measure of model uncertainty to the model error weights.

6.1 Introduction

Terms like *model fitting, model tuning, parameter estimation*, etc., may all be viewed as some sort of calibration process - we seek to find/adjust a set of parameters to obtain a "best fit", e.g. in a least-squares sense, between the model and observed measurements. For mechanistic/first-principles models, these parameters will typically represent physical quantities such as densities, friction factors, heat loss coefficients and the like. Thus, when calibrating such models we can include *a priori* information about the parameters. For instance, if we are to calibrate a density parameter we may restrict our search to positive densities, possibly within some tolerance of our previous estimate. In contrast, parameters for empirical/data-driven models (e.g. the Fetkovich/C&n IPR, see Section 3.3.1) will typically not have any physical meaning - they are simply chosen to make the model match observed measurements.

The models we need to calibrate here are the B-spline approximated models used in Chapters 4 and 5. These B-spline models could in a sense be considered semi-mechanistic, since the B-splines were created by sampling and approximating semi-mechanistic models[1]. On the other hand, B-spline functions are represented by a set of coefficients (cf. Appendix A) which have no physical meaning, so it would probably be more correct to call them data-driven models. In any case, we must make a choice - should we calibrate the semi-mechanistic models *before* creating the B-spline approximations, or should we modify the B-spline coefficients directly? In this chapter, we choose the former approach; partly because it is more intuitive, but also because it is not trivial to work directly with B-spline coefficients, particularly for higher-dimensional B-splines like the ones representing the pipeline VLP and choke models. IPRs and WPCs are calibrated by adding a fixed pressure offset, while choke models are calibrated by a calibration factor included in the model. The pipeline VLP is calibrated with a model fitting tool in GAP before sampling. After calibration, new B-spline approximations are created based on the calculated parameters.

In this subsequent sections, we assume we have been provided with a set of m measured liquid rates, pressures, temperatures and choke positions at relevant points in the production system. Here, m denotes the number of tested rates. For calibration against a single-rate well test, we could set $m = 1$. We could also include old test data by setting $m > 1$, however, we may want to give old data less consideration than new data, so we introduce a set of weights $\{w_k\}_{k=0}^{m-1}$ to allow for this. For a multi-rate test with n tested rates, we could set $m = n$ (or $m > n$ if we also want to consider old data).

For the wells, we assume we have measured the individual phase flow rates in standard conditions, pressures in the bottom hole, wellhead and downstream the turret choke, the wellhead temperature and the wellhead choke position:

$$\text{Well calibration sets: } \left\{\tilde{q}^{i}_{o,k}, \tilde{q}^{i}_{g,k}, \tilde{q}^{i}_{w,k}, \tilde{p}^{i}_{k}, \tilde{p}^{n_w+i}_{k}, \tilde{p}^{2n_w+i}_{k}, \tilde{T}^{n_w+i}_{k}, \tilde{u}^{i}\right\}_{k=0}^{m-1}, \quad \forall\, i \in \mathscr{W}. \tag{6.1}$$

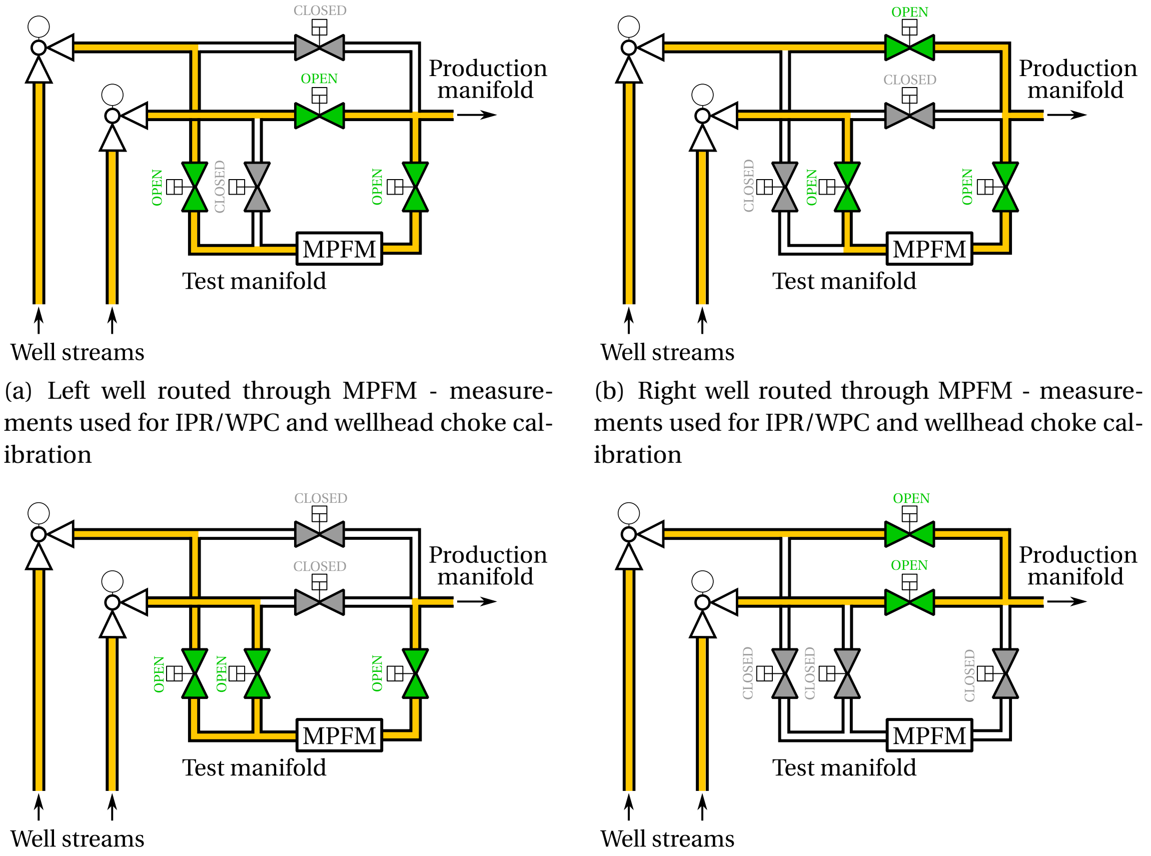
For the pipeline and turret choke, we have measured the pipeline individual phase flow rates in standard conditions and the pressures and temperatures in the manifold, upstream and downstream the turret choke, and the turret choke position:

$$\text{VLP/turret chk. calib. set: } \left\{\tilde{q}^{3n_w}_{l,k}, \tilde{q}^{3n_w}_{g,k}, \tilde{q}^{3n_w}_{w,k}, \tilde{p}^{3n_w}_{k}, \tilde{p}^{3n_w+1}_{k}, \tilde{p}^{3n_w+2}_{k}, \tilde{T}^{3n_w}_{k}, \tilde{T}^{3n_w+1}_{k}, \tilde{T}^{3n_w+2}_{k}, \tilde{u}^{t}\right\}_{k=0}^{m-1}. \tag{6.2}$$

[1] When we say *semi*-mechanistic, we mean that the models are based on physical principles, but still contain an empirical aspect in the sense that correlations are used to account for multiphase flow.

6.2 Template configuration

The Skarv wells are tested by routing the well stream through a multiphase flow meter (MPFM) fitted on the subsea template. This section briefly describes the routing system and how the template must be configured to obtain the data sets above. The data set for a well (6.1) is obtained by routing the well through the MPFM as shown in Figures 6.1(a) (for the first well) and 6.1(b) (for the second well). For the pipline and turret choke, i.e. data set (6.2), both wells are routed through the MPFM; this way the total flow rate through the pipeline and turret choke is measured. This is shown in Figure 6.1(c).



(a) Left well routed through MPFM - measurements used for IPR/WPC and wellhead choke calibration

(b) Right well routed through MPFM - measurements used for IPR/WPC and wellhead choke calibration

(c) Both wells routed through MPFM - measurements used for pipeline and turret choke calibration

(d) Both wells routed directly to production manifold - no calibration

Figure 6.1: Different template configurations for multiphase metering.

6.3 Well Calibration

With well calibration, we mean calibrating all parameters and models associated with a well, namely the GOR, water cut, IPR and WPC.

6.3.1 GOR and water cut adjustment

The GOR and water cut are calibrated using a simple weighted average of the measurements:

$$r^i_{go} = \frac{\sum_{k=0}^{m-1} w_k r^i_{go,k}}{\sum_{k=1}^{m-1} w_k}, \text{ where } r^i_{go,k} = \frac{\tilde{q}^i_{g,k}}{\tilde{q}^i_{o,k} + \tilde{q}^i_{g,k}}, \tag{6.3}$$

$$r^i_{wc} = \frac{\sum_{k=0}^{m-1} w_k r^i_{wc,k}}{\sum_{k=1}^{m-1} w_k}, \text{ where } r^i_{wc,k} = \frac{\tilde{q}^i_{w,k}}{\tilde{q}^i_{o,k} + \tilde{q}^i_{w,k}}. \tag{6.4}$$

Once calibrated, the GOR and water cut are assumed constant until the next well test. Consequently, any changes in the actual GOR and water cut will lead to errors in the estimated flow rates of each phase. Some reservoir conditions may lead to flow-rate dependent GOR and water cut (e.g. gas and/or water coning), therefore, a simple maximum difference check is performed to verify that the GOR and water cut are independent of flow rate;

$$\left| \max_{k=0,\dots,m-1} r^i_{go,k} - \min_{k=0,\dots,m-1} r^i_{go,k} \right| \le \epsilon_{r_{go}} \text{ and } \left| \max_{k=0,\dots,m-1} r^i_{wc,k} - \min_{k=0,\dots,m-1} r^i_{wc,k} \right| \le \epsilon_{r_{wc}}, \tag{6.5}$$

where $\epsilon_{r_{go}}$ and $\epsilon_{r_{wc}}$ are some tolerances. If these conditions are not satisfied for some i, this might indicate that the associated well has rate dependent GOR or water cut. Although this would lead to estimation errors for the estimators formulated in Chapters 4 and 5, it could be accounted by introducing GOR and water cut variables for each well and adding constraints describing the flow rate dependency, or removing the GOR and water cut constraints and calculating each flow rate independently, cf. Section 2.6.

6.3.2 IPR offset adjustment

The IPRs are calibrated by adding an offset to each curve and thereby lifting or lowering the entire curve to match observed data. The IPRs are calibrated to liquid rates, so the measurements used from (6.1) are the oil and water rates, and the bottom hole pressures. We seek to find the offset Δp^i which minimizes the least-squares error between the measured bottom hole pressures $\tilde{p}^i_k$ and the bottom hole pressures predicted by the (offset) B-spline approximated IPR from GAP, $p^i_k = \Phi_{\text{ipr}}(\tilde{q}^i_{l,k}) + \Delta p^i$, evaluated at the measured liquid rates $\tilde{q}^i_{l,k} = \tilde{q}^i_{o,k} + \tilde{q}^i_{w,k}$. This results in the following nonlinear least-squares problem:

Problem 6.1: IPR offset calibration

$$\begin{array}{ll} \underset{\Delta p^i}{\text{minimize}} & \sum_{k=0}^{m-1} w_k \left(\tilde{p}_k^i - p_k^i\right)^2 \\ \text{subject to} & p_k^i = \Phi_{\text{ipr}}(\tilde{q}_{l,k}) + \Delta p^i, \quad k = 0, \ldots, m-1. \end{array}$$

When an optimal offset $\Delta p^{i,*}$ has been found, a new B-spline approximation of the IPR curve is made, where the offset is added to the IPR curve from GAP, i.e. we make a B-spline approximation of $f_{\text{ipr}}^{i}{}^{-1}(q_l) + \Delta p^{i,*}$.

6.3.3 WPC offset adjustment

The WPC curves are calibrated in the same way the IPRs were, by adding an offset. This time, the offset is added to the wellhead pressure. The measurements used from (6.1) are the oil and water rates and the wellhead pressures. Similarly to the IPR, we now seek to find the offset Δp^{n_w+i} which minimizes the least-squares error between the measured wellhead pressures $\tilde{p}^{n_w+i}$ and the wellhead pressures predicted by the (offset) B-spline approximated WPC from GAP, $p_k^{n_w+i} = \Phi_{\text{wpc}}(\tilde{q}_{l,k}) + \Delta p^{n_w+i}$. The nonlinear least-squares problem thus becomes

Problem 6.2: WPC offset calibration

$$\begin{array}{ll} \underset{\Delta p^{n_w+i}}{\text{minimize}} & \sum_{k=0}^{m-1} w_k \left(\tilde{p}_k^{n_w+i} - p_k^{n_w+i}\right)^2 \\ \text{subject to} & p_k^{n_w+i} = \Phi_{\text{wpc}}(\tilde{q}_{l,k}^i) + \Delta p^{n_w+i}, \quad k = 0, \ldots, m-1. \end{array}$$

Again, we use the optimal offset $\Delta p^{nw+i,*}$ to create a new B-spline approximation of an offset WPC from GAP $f_{\text{wpc}}^{i}{}^{-1}(q_l) + \Delta p^{n_w+i,*}$.

One remark is necessary here; this calibration strategy does not reflect the structure of the WPC. The WPC can be considered a "composite" IPR/VLP; therefore it would make more sense to calibrate the IPR and VLP separately, and then creating a WPC based on these. The well VLP would typically be calibrated with separate hydrostatic and friction terms, which may be necessary when frictional pressure losses cannot be neglected. However, we have assumed that frictional pressure losses are not significant, which means the calibration method outlined above is sufficient for its purpose.

6.4 Choke Calibration

Recall the choke pressure drop function (4.18); $p^{ds} = f_{\text{chk}}(q_l, p^{us}, \overline{T}, C_v, r_{go}, r_{wc}, c)$. We seek to find a calibration factor c which matches this function to observed data. To accomplish this, we sample the function in a grid of calibration factors, liquid rates, upstream pressures and flow coefficients, and create a B-spline approximation which now becomes 4-dimensional. To reduce the number of variables in the B-spline, the average temperature, GOR and water cut are assumed fixed, and calculated from available data. For the wellhead chokes, we use the GOR calculated in (6.3) and the water cut calculated in (6.4). The average temperature is calculated as

$$\overline{T}^i = \frac{\sum_{k=0}^{m-1} w_k \overline{T}_k^i}{\sum_{i=0}^{m-1} w_k}, \tag{6.8}$$

where $\overline{T}_k^i = \frac{1}{2}\tilde{T}_k^{n_w+i} + \frac{1}{2}\tilde{T}_k^{2n_w+i}$ is the average of the upstream and downstream temperature in sample k. For the turret choke, we need to calculate an average pipeline GOR and water cut based on the measured flow rates in the pipeline, i.e.

$$\overline{r}_{go}^p = \frac{\sum_{k=0}^{m-1} w_k r_{go,k}^p}{\sum_{k=1}^{m-1} w_k}, \text{ where } r_{go,k}^p = \frac{\tilde{q}_{g,k}^{3n_w}}{\tilde{q}_{o,k}^{3n_w} + \tilde{q}_{g,k}^{3n_w}}, \tag{6.9}$$

$$\overline{r}_{wc}^p = \frac{\sum_{k=0}^{m-1} w_k r_{wc,k}^p}{\sum_{k=1}^{m-1} w_k}, \text{ where } r_{wc,k}^p = \frac{\tilde{q}_{w,k}^{3n_w}}{\tilde{q}_{o,k}^{3n_w} + \tilde{q}_{w,k}^{3n_w}}. \tag{6.10}$$

Similarly to the wellhead chokes, the average temperature in the turret choke is calculated as

$$\overline{T}^t = \frac{\sum_{k=0}^{m-1} w_k \overline{T}_k^t}{\sum_{i=0}^{m-1} w_k}, \tag{6.11}$$

where $\overline{T}_k^i = \frac{1}{2}\tilde{T}_k^{3n_w+1} + \frac{1}{2}\tilde{T}_k^{3n_w+2}$ is the average of the upstream and downstream temperature in sample k. The resulting B-splines obtained by sampling the choke pressure drop function for the fixed GOR, water cut and temperature as descibed above, are denoted $\Phi_{cc}^i(\cdot)$ for wellhead choke i and $\Phi_{cc}^t(\cdot)$ for the turret choke. The cc-subscript denotes *choke calibration*. This results in the following functions used for calibration:

$$p^{2n_w+i} = \Phi_{cc}^i(\tilde{q}_l^i, \tilde{p}^{n_w+i}, C_v^i, c^i), \tag{6.12}$$

and

$$p^{3n_w+2} = \Phi_{cc}^t(\tilde{q}_l^{3n_w}, \tilde{p}^{3n_w+1}, C_v^t, c^t). \tag{6.13}$$

Now, we find an optimal calibration factor c by solving a nonlinear least-squares problem similar to the ones for the IPRs and WPCs. We also make use of the B-spline approximated choke coefficient characteristic $\Phi_{C_v}(\cdot)$ for translating choke positions to flow coefficients. For the wellhead chokes, the problem becomes the following:

Problem 6.3: Wellhead choke calibration

$$\begin{array}{lll} \underset{c^i}{\text{minimize}} & \sum_{k=0}^{m-1} w_k \left(\tilde{p}_k^{2n_w+i} - p_k^{2n_w+i}\right)^2 & \\ \text{subject to} & p_k^{2n_w+i} = \Phi_{cc}^i(\tilde{q}_{l,k}^i, \tilde{p}_k^{n_w+i}, C_{v,k}^i, c^i), & k = 0,\dots,m-1 \\ & C_{v,k}^i = \Phi_{C_v}^i(\tilde{u}_k^i), & k = 0,\dots,m-1 \end{array}$$

For the turret choke, the problem is basically the same, except for the variables in the problem;

Problem 6.4: Turret choke calibration

$$\begin{array}{lll} \underset{c^t}{\text{minimize}} & \sum_{k=0}^{m-1} w_k \left(\tilde{p}_k^{3n_w+2} - p_k^{3n_w+2}\right)^2 & \\ \text{subject to} & p_k^{3n_w+2} = \Phi_{cc}^t(\tilde{q}_{l,k}^{3n_w}, \tilde{p}_k^{3n_w+1}, C_{v,k}^t, c^t), & k = 0,\dots,m-1 \\ & C_{v,k}^t = \Phi_{C_v}^t(\tilde{u}_k^t), & k = 0,\dots,m-1 \end{array}$$

The optimal calibration factors $c^0,\dots,c^{n_w-1},c^t$ are then used when sampling (4.18) to create the choke models.

6.5 Pipeline Calibration

Seeing as how the pipeline VLP is sampled upfront from GAP, it is difficult to derive a simple calibration strategy which makes much sense. Clearly, the simple approach used for the IPR and WPC is an alternative, but such a strategy does not reflect the structure of the pipeline model; ideally, the hydrostatic and friction terms should be calibrated separately. Since GAP allows the user to enter measurement data for model fitting, we choose to calibrate the pipeline VLP *before* sampling.

6.6 Model uncertainty and weighting

Obviously, calibration is essential to obtain accurate flow estimates. However, since we have allowed for weighting model errors in the flow estimation problem, we would clearly benefit from finding some measure of model uncertainty. For single-rate well tests ($m = 1$), this is difficult, since we can only measure the model error at the test point itself. On the other hand, if multi-rate tests are available, we can compare the calibrated model to our set of test points and calculate some form of model uncertainty. This section presents some ideas about how we can measure model uncertainty and incorporate this knowledge into the optimization problem.

6.6.1 Time-invariant weighting based on error variance

One idea is to measure the model uncertainty in terms of error variance. Assuming we have calibrated the models as described in the sections above, we can evaluate model errors at each test point, and calculate statistics like mean error and error variance. Take for instance the IPR. The *model error* at test point k is the difference between the measured bottom hole pressure $\tilde{p}^i_k$, and the bottom hole pressure predicted by the IPR, i.e. $p^i_k = \Phi^i_{\text{ipr}}(\tilde{q}^i_{l,k})$. Note that $\Phi_{\text{ipr}}(\cdot)$ is now calibrated, i.e. the offset found in Problem 6.1 has been included before the B-spline was created. We denote the model error for IPR i at test point k as $e^i_{\text{ipr},k}$, which becomes

$$e^i_{\text{ipr},k} = \tilde{p}^i_k - \Phi^i_{\text{ipr}}(\tilde{q}^i_{l,k}), \quad k = 0, \dots, m-1. \tag{6.16}$$

Repeating this for all k, we can calculate a *mean error* $\overline{e}^i_{\text{ipr}}$ as

$$\overline{e}^i_{\text{ipr}} = \frac{1}{m} \sum_{k=0}^{m-1} e^i_{\text{ipr},k}, \tag{6.17}$$

and an *error variance* $\left(\sigma^i_{\text{ipr}}\right)^2$ as

$$\left(\sigma^i_{\text{ipr}}\right)^2 = \frac{1}{m} \sum_{k=0}^{m-1} \left(e^i_{\text{ipr},k} - \overline{e}^i_{\text{ipr}}\right)^2. \tag{6.18}$$

A large error variance will now indicate that the model uncertainty is large; note that a perfect model would give $e^i_{\text{ipr},k} = 0$ for all k, and consequently an error variance of zero. If we follow this procedure for every model (WPCs, chokes, pipeline and turret choke), we can view the error variances as relative measures of model accuracy. For the remaining models, the procedure will be similar, except for the calculation of the model errors for test point k, i.e. (6.16), which will be different for each model. For clarity, this is demonstrated for each model in Table 6.1 below.

Model	Error at test point k	Note
IPR, well i	$e^i_{\text{ipr},k} = \tilde{p}^i_k - \Phi^i_{\text{ipr}}(\tilde{q}^i_{l,k})$	-
WPC, well i	$e^i_{\text{wpc},k} = \tilde{p}^{n_w+i}_k - \Phi^i_{\text{wpc}}(\tilde{q}^i_{l,k})$	-
Wellhead choke i	$e^i_{\text{chk},k} = \tilde{p}^{2n_w+i}_k - \Phi^i_{\text{chk}}(\tilde{q}^i_{l,k}, \tilde{p}^{n_w+i}_k, \overline{T}^i, C^i_{v,k})$	$C^i_{v,k} = \Phi^i_{C_v}(u^i_k)$
Pipeline VLP	$e_{\text{vlp},k} = \tilde{p}^{3n_w+1}_k - \Phi_{\text{vlp}}(\tilde{q}^{3n_w}_{l,k}, \tilde{p}^{3n_w}_k, \tilde{T}^{3n_w}, \overline{r}^p_{go}, \overline{r}^p_{wc})$	-
Turret choke	$e^t_{\text{chk},k} = \tilde{p}^{3n_w+2}_k - \Phi^t_{\text{chk}}(\tilde{q}^{3n_w}_{l,k}, \tilde{p}^{3n_w+1}_k, \overline{T}^t, \overline{r}^p_{go}, \overline{r}^p_{wc}, C^t_{v,k})$	$C^t_{v,k} = \Phi^t_{C_v}(u^t_k)$

Table 6.1: Definition of model errors.

Following the same approach as in (6.17) and (6.18), we can calculate mean errors and error variances for each model. Keeping in mind that we have calibrated the models by minimizing a least-squares error term which, aside from the constant $1/m$ is basically the same as (6.17), we would expect the mean errors to be close to zero. If it is not, this might indicate that something went wrong during the calibration, for instance, the solver may have converged to a suboptimal local minimum. This is not likely for the simple offset calibration of the IPR and WPC, but when experimenting with the choke calibration (Problems 6.3 and 6.4), it turned out the problems had several suboptimal local minima, and it was necessary to find a reasonable starting point for the calibration factor c in order to converge to the correct (global) minimum.

When we are done with calculating error variances for each model, we can include this information in the optimization problem through the weights in $\mathbf{Q}^{-1}$, by setting (with reference to Section 4.7.1)

$$\mathbf{Q}_{\text{ipr}} = \text{diag}\left(\left(\sigma^0_{\text{ipr}}\right)^2, \dots, \left(\sigma^{n_w-1}_{\text{ipr}}\right)^2\right), \tag{6.19}$$

$$\mathbf{Q}_{\text{wpc}} = \text{diag}\left(\left(\sigma^0_{\text{wpc}}\right)^2, \dots, \left(\sigma^{n_w-1}_{\text{wpc}}\right)^2\right), \tag{6.20}$$

$$\mathbf{Q}_{\text{chk}} = \text{diag}\left(\left(\sigma^0_{\text{chk}}\right)^2, \dots, \left(\sigma^{n_w-1}_{\text{chk}}\right)^2\right), \tag{6.21}$$

$$Q_{\text{vlp}} = \left(\sigma_{\text{vlp}}\right)^2, \tag{6.22}$$

$$Q^t_{\text{chk}} = \left(\sigma^t_{\text{chk}}\right)^2 \tag{6.23}$$

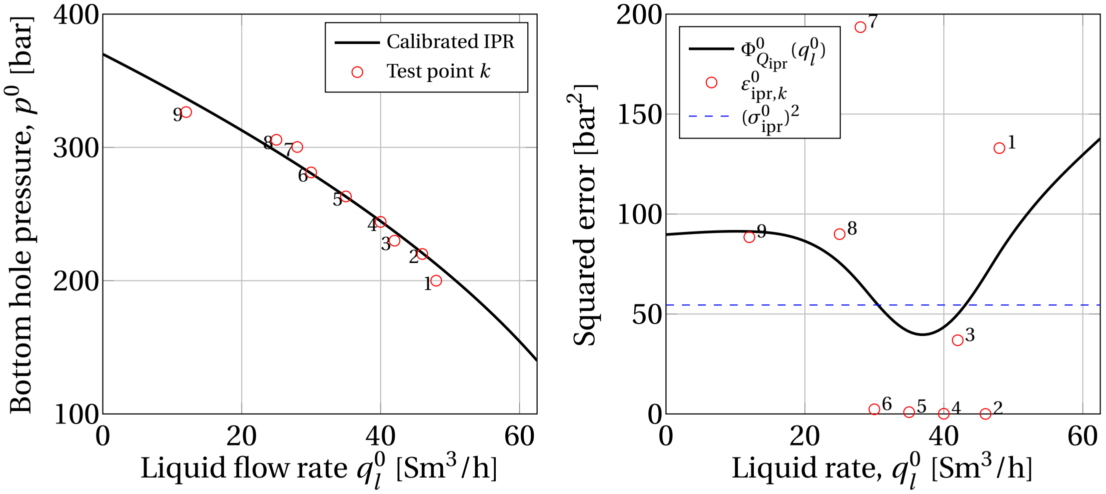
Note that the actual weights are inverses of the above, i.e. a large error variance leads to a small weight. Thus, a model with a large error variance (large model uncertainty) will be given less consideration in the optimization problem. Since the model error weights now are in terms of pressure error variances, the natural choice for the weights in **R** may be the measurement error variance for each pressure transmitter, which will typically be printed on the data sheet and/or calibration certificate of each transmitter.

6.6.2 Time-variant weighting

A slightly more ambitious idea is to assume that in some areas, a model will be good, while it will be bad in other areas. When the estimator is running, we will have a pretty good idea of the current operational conditions by looking at the estimate from the previous time index. Then, we may want to give the most weight to models which we know are good for the given conditions. We will not go into details here (the idea has not been implemented in this thesis), but simply illustrate with an example. Say we have performed a multi-rate well test on well 0, with $m = 10$, and calibrated the IPR curve by solving Problem 6.1. This may result in the calibrated IPR curve shown together with the 10 flow tests in Figure 6.2(a). In this case, we have relatively large errors for low and high flow rates, while the error is small in the middle region. If we define a squared error for test point k as

$$\varepsilon^0_{\text{ipr},k} = \left(e^0_{\text{ipr},k} - \bar{e}^0_{\text{ipr}}\right)^2, \tag{6.24}$$

this will indicate the model quality at the liquid flow rate at test point k. In Figure 6.2(b), we have plotted $\varepsilon^0_{\text{ipr},k}$ for the test rates in Fig. 6.2(a) (indicated with red circles). Since the points may be scattered, we may want to use e.g. a smoothing spline to get a more well-behaved relation between liquid flow rate and the squared error; in Fig. 6.2(b) we have shown such a spline in black and called it $\Phi^0_{Q_{\text{ipr}}}(q^0_l)$.



(a) Well tests

(b) Well test, squared pressure errors

Figure 6.2: Time-variant weighting strategy.

The idea is now to choose the weight for the IPR model error based on the estimated liquid flow rate from the previous time index. For instance, assume that the optimal solution of Problem 2.1 at time index T suggested a liquid flow rate from well 0 $q^{0,*}_{l,T}$.

Then, the IPR model error weight Q^0_{ipr} for the problem to be solved at time index $T+1$ is selected as

$$Q^0_{\text{ipr},T+1} = \Phi^0_{Q_{\text{ipr}}}(q^{0,*}_{l,T}). \tag{6.25}$$

Note that this results in *time-variant* weighting, i.e. the weights in Problem 2.1 vary over time. This does not lead to any additional constraints - the weights are calculated before the estimation problem is solved. For comparison, we have also shown the time-invariant Q^0_{ipr} that would result from the suggested method in Section 6.6.1, i.e. the error variance $(\sigma^0_{\text{ipr}})^2$.

Chapter 7

Implementation

In this chapter, we will outline how the flow estimation problems defined in Chapters 4 and 5 are used in a real-time flow estimation loop, and give some brief comments regarding the C++ code implementation.

7.1 Static flow estimation algorithm

Since the solution of Problem 4.1 provides flow rate estimates at a single time index only, we place the problem inside a loop which solves the problem each time a new set of measurements is available, i.e. every Δt seconds. This enables real-time estimation, provided the problem can be solved in less than Δt seconds. A coarse, but illustrative outline of the implemented static flow estimator is given as Algorithm 7.1 below.

Algorithm 7.1: Static flow estimation algorithm.

```
Data: Measurement stream $\mathbf{y}_0, \mathbf{y}_1, \ldots$
Result: Estimate stream $\mathbf{x}_0^*, \mathbf{x}_1^*, \ldots$
Build optimization problem 4.1;
$T \leftarrow 0$;
while true do
    $\mathbf{y}_T \leftarrow$ Read measurements at time $t_T$;
    if $T > 0$ then
        $\mathbf{x}_{\text{init},T} \leftarrow \mathbf{S}\mathbf{y}_T + (\mathbf{I} - \mathbf{S})\mathbf{x}_{T-1}^*$;
    else
        $\mathbf{x}_{\text{init},T} \leftarrow \mathbf{S}\mathbf{y}_T$;
    end
    $\mathbf{x}_T^* \leftarrow$ Solve Problem 4.1 with starting point $\mathbf{x}_{\text{init},T}$;
    wait until $t = t_T + \Delta t$
    $T \leftarrow T + 1$, $t_T \leftarrow t$;
end
```

Note that this algorithm resembles the moving-horizon estimator described in Section 2.4. The only difference is that the internal model is static as opposed to dynamic, which also means the estimation horizon N is 1. With respect to model calibration, this is done before we start the algorithm. In our case, we have calibrated the internal models by solving Problems 6.1 through 6.4. The obtained information is taken into account in the line "Build optimization problem 4.1", where the models (in the form of tabular data) from GAP are read and B-spline approximations are created based on the model tables and the calibration factors.

7.2 Dynamic flow estimation algorithm

Similarly to the static estimation algorithm above, we place Problem 5.1 inside a loop to provide real-time estimates. The resulting algorithm is a moving-horizon estimator, and is given as Algorithm 7.2 below.

Algorithm 7.2: Dynamic flow estimation algorithm (MHE).

```
Data: Measurement stream $\mathbf{y}_0, \mathbf{y}_1, \dots$
Result: Estimate stream $\mathbf{x}^*_{N|N}, \mathbf{x}^*_{N+1|N+1}, \dots$
Build optimization problem 4.1;
$T \leftarrow N$;
$\mathbf{y}_T \leftarrow$ Read measurements at time $t_T$;
$\mathbf{x}^S \leftarrow$ Solve Problem 4.1 with starting point $\mathbf{x}_{\text{init}} = \mathbf{S}_\mathbf{y}\mathbf{y}_T$;
$(\mathbf{m}^S, \mathbf{H}^S_l) \leftarrow$ Calculate masses and holdups using procedure in Appendix C.2.1;
Build optimization problem 5.1;
while true do
    $[\mathbf{y}^\top_{T-N+1}, \dots, \mathbf{y}^\top_T]^\top \leftarrow$ Read $N$ preceding measurements at time $t_T$;
    if $T > N$ then
        $\mathbf{x}_{\text{init}|T} \leftarrow$ as per (5.60)-(5.62);
    else
        $\mathbf{x}_{\text{init}|T} \leftarrow$ as per (5.60) and (5.63);
    end
    $\mathbf{x}^*_T \leftarrow$ Solve Problem 5.1 with starting point $\mathbf{x}_{\text{init}|T}$;
    Output last estimate in trajectory ($\mathbf{x}^*_{T|T}$) to estimate stream;
    wait until $t = t_T + \Delta t$
    $T \leftarrow T + 1$, $t_T \leftarrow t$;
end
```

There are two factors which make Algorithm 7.2 slightly more involved than Algorithm 7.1. The first is the fact that we have included mass and holdup variables, which require some intialization. Here, we solve the static estimation problem (4.1) and use the resulting flow rate estimates together with spline-approximated fluid property tables to provide the initial estimates. The second complicating factor is the dynamic model and estimation horizon N, which first and foremost makes the optimization problem larger, but also leads to a more involved calculation of a good starting point. Like the case was for Algorithm 7.1, we assume the internal models have been calibrated beforehand.

7.3 C++ code implementation

Algorithms 7.1 and 7.2 are implemented in C++, in the code framework CENSO (Convex ENvelopes for Spline Optimization) , which is a framework for global optimization of MINLP (Mixed-Integer Nonlinear Programming) problems with B-spline constraints (Grimstad & Sandnes, 2014). The choice of using CENSO is not based on its global optimization capabilities, nor its ability to handle integer variables; Problems 4.1 and 5.1 are NLPs which are to be solved locally. However, CENSO has a streamlined coding interface to the powerful interior-point solver IPOPT, and in addition it has all the required B-spline functionality needed for this thesis. Moreover, *if* we decided to extend the model in a way which called for integer or binary variables (e.g. well shut-in modelling), this would be straightforward in CENSO.

The most code-intensive parts of the implementation are not the flow estimation loops outlined in Algorithms 7.1 and 7.2, but rather the actual formulation of the optimization problems, calibration routines and file import routines. Some functionality with respect to reading GAP files was already implemented in relation to the case study by (Grimstad et al., 2014), however, new functionality for reading OLGA geometry files (.geo), PVTsim PVT tables (.tab) and OLGA/historian measurement data (.csv) has been added. In addition, the choke model described in Section 4.5.3 was implemented from scratch. Figure 7.1 shows the overall structure and information flow in the finished code project. The code project has been submitted to my co-supervisor Bjarne Grimstad, and is thus available for inspection or further work.

7.3.1 NLP solver

The (local) NLP solver used to solve Problems 4.1 and 5.1 is the open source interior-point solver IPOPT (Wächter & Biegler, 2006). IPOPT is a part of the COIN-OR project[1], and is a much used solver for large-scale nonlinear programming. In this thesis, it is set up with default settings for the most part, except for the parameter *bound_relax_factor*, which is set small (10^{-4}). This is to prevent the solver from venturing outside the bounds of the variables involved in B-spline constraints (The B-splines cannot be evaluated outside their bounds).

[1] See http://www.coin-or.org.

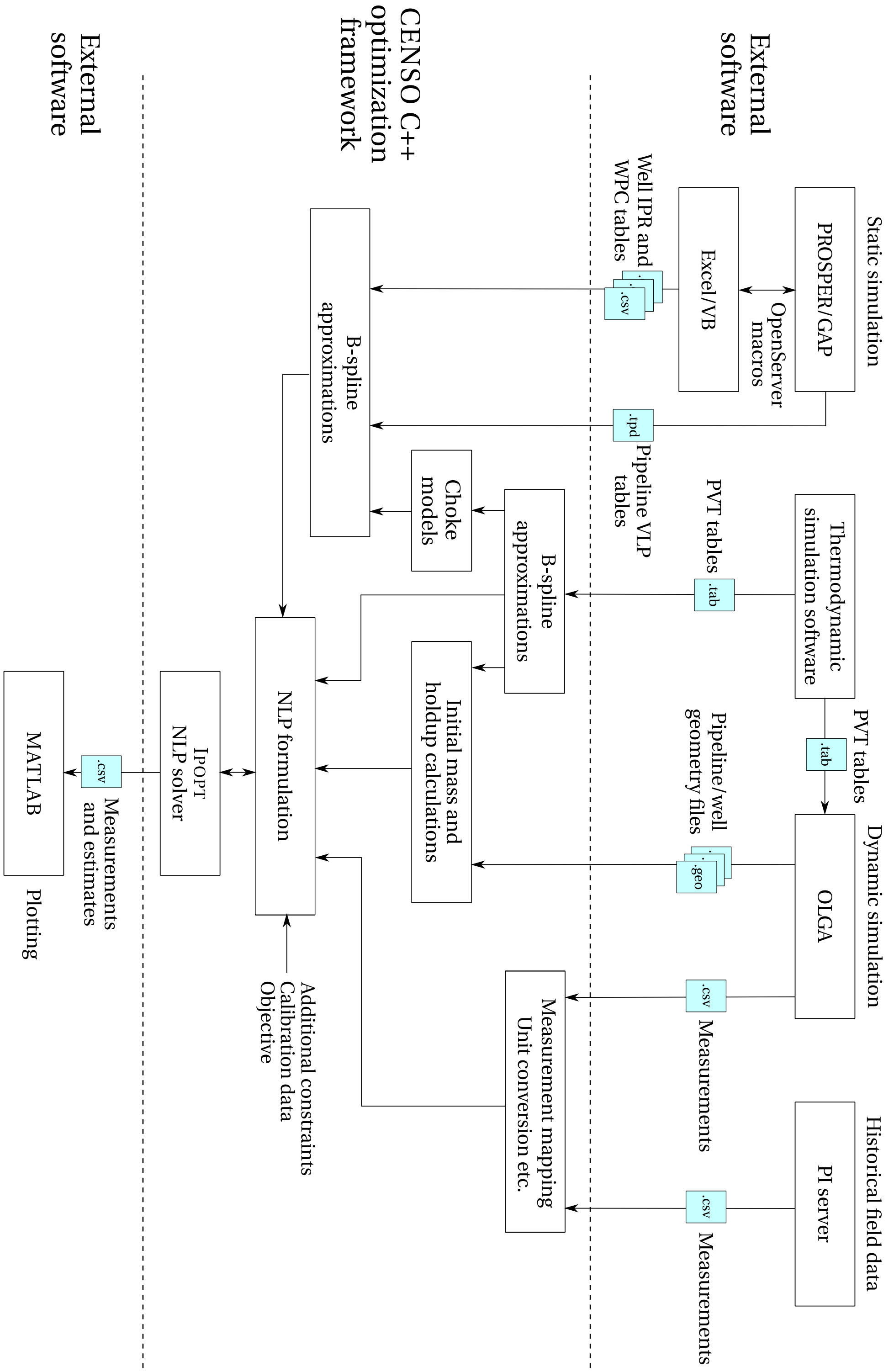


Figure 7.1: Estimation system structure and information flow.

Chapter 8

Simulation results

In this chapter, the estimators presented in Chapters 4 and 5 are tested on a set of typical (and not so typical) operating scenarios. To assess the performance of the estimators, a relatively detailed model of a two-well production network was implemented in OLGA, an industry standard multiphase flow simulator. The first section gives a short summary of the OLGA model, which is followed by a description of the measures to be used for evaluating the performance of the estimators. Then, two simulation cases are presented where the measurements from OLGA are compared with estimates generated by the static and dynamic estimators. The first case emulates a simple well adjustment, and the second is a riser slugging case. After the simulation section, a field data case is presented, where estimates are generated from field measurements supplied by BP. A section on solution times presents the CPU time used to generate the estimates, before we conclude the chapter by taking a closer look at the sensitivity properties of the pipeline VLP, which is seen to cause some problems in the first test case. In this chapter, *static estimation* refers to Algorithm 7.1, while *dynamic estimation* refers to Algorithm 7.2.

8.1 OLGA production network model

A two-well model was created in OLGA to run benchmark simulations. OLGA is a multiphase flow simulator which is used extensively in the oil industry. Typical uses include what-if analyses and simulations used for flow assurance. The model uses detailed well and pipeline geometries which correspond to the Tilje production template. In addition, the choke valves are equipped with the actual flow coefficient characteristics of the Tilje choke valves, which were obtained in a water test facility before the chokes were fitted. Figure 8.1 shows the network model graphically as it appears in OLGA. The PVT table used for compositional calculations was also supplied by BP, and represents the Tilje wells.

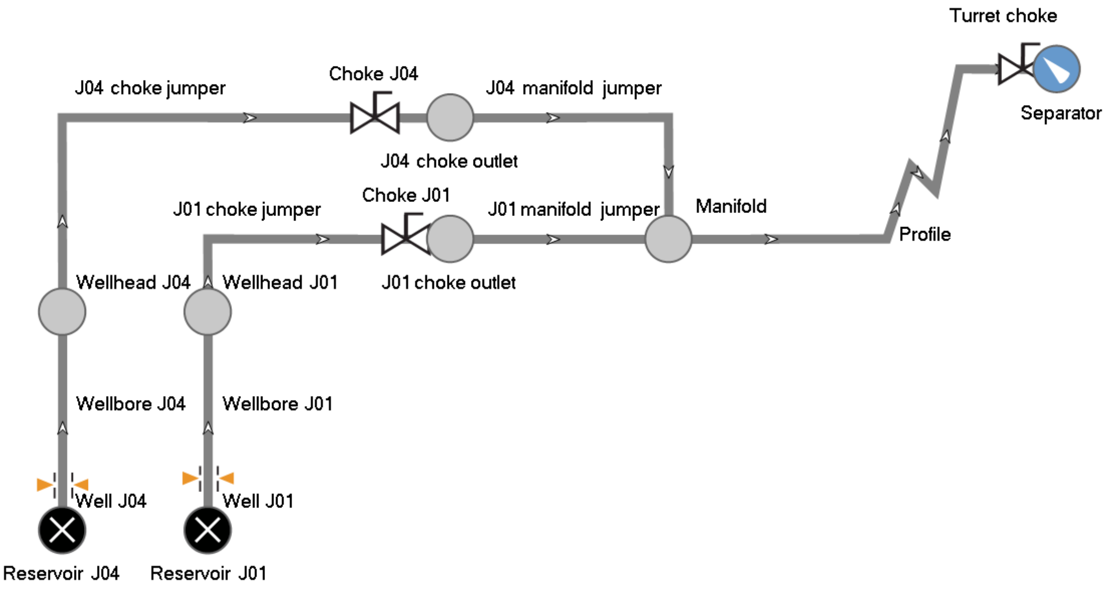


Figure 8.1: OLGA production network model.

Table D.1 in Appendix D.1 shows the most imporant settings used in the OLGA simulations. Refer to the OLGA user manual (SPT Group, 2013) for detailed descriptions.

8.2 Performance measures

To assess the performance of the different methods objectively, a set of performance measures are needed. The performance measures defined below are designed to judge the performance in terms of both the maximum estimation error and the total estimation error throughout the simulation. Since the system model is mainly formulated in terms of pressure-liquid rate relations, the performance measures are based on liquid rate estimation errors. Errors are evaluated at six locations; the in/outflow of both wells (4) and the in/outflow of the flowline (2). Table 8.1 defines the instantaneous error measures at time index T:

Symbol	Description	Definition
e^0_T	Well 0 inflow estimation error	$e^0_T = q^0_{l,T} - \tilde{q}^0_{l,T}$
e^1_T	Well 1 inflow estimation error	$e^1_T = q^1_{l,T} - \tilde{q}^1_{l,T}$
e^2_T	Well 0 outflow estimation error	$e^2_T = q^2_{l,T} - \tilde{q}^2_{l,T}$
e^3_T	Well 1 outflow estimation error	$e^2_T = q^3_{l,T} - \tilde{q}^3_{l,T}$
e^6_T	Pipeline inflow estimation error	$e^6_T = q^6_{l,T} - \tilde{q}^6_{l,T}$
e^7_T	Pipeline outflow estimation error	$e^7_T = q^7_{l,T} - \tilde{q}^7_{l,T}$

Table 8.1: Instantaneous error measures.

Errors are not evaluated for edges 4 and 5, since these flow rates will be roughly equal to the flow rates in edges 2 and 3. The notation in Table 8.1 follows the thesis standard notation: $q^i_{l,T}$ is the estimated liquid flow rate through edge i in the flow network at time index T, and $\tilde{q}^i_{l,T}$ is the corresponding liquid flow rate measurement (from OLGA). Since we are minimizing a least-squares objective, a natural error measure is the integrated squared estimation error (ISE) of each error, i.e. $\int_0^{T_{\text{sim}}} |(e^i(t))^2| \, \mathrm{d}t$, where T_{sim} is the total simulation time. Since the measurements and estimates are discrete, the ISE is approximated by E^i_{ise}, which is defined as

$$E^i_{\text{ise}} = \Delta t \sum_{T=0}^{n_T-1} (e^i_T)^2, \qquad \forall \; i \in \mathscr{E} \tag{8.1}$$

where Δt is the sampling time in seconds, n_T is the number of samples and $\mathscr{E} = \{0, 1, 2, 3, 6, 7\}$ is an index set for all the edges considered in the error evaluation. Another relevant measure is the maximum deviation between the estimated rates and the measured rates for the entire simulation. This is most conveniently expressed as a percentage, so we define the maximum relative estimation errors as

$$E^i_{\text{max}} = 100 \cdot \max_{T=0,\dots,n_T-1} \frac{|e^i_T|}{\tilde{q}_{\text{max}}}, \qquad \forall \; i \in \mathscr{E} \tag{8.2}$$

where $\tilde{q}_{\text{max}}$ is the largest measured flow rate during the simulation. Finally, a measure to summarize the ISEs is their sum, which we denote E_{tot} (total ISE):

$$E_{\text{tot}} = \sum_{i \in \mathscr{E}} E^i_{\text{ise}}. \tag{8.3}$$

For the field data case, the full set of performance measures described above cannot be calculated, since the true flow rate is not available for all the edges. However, by using the available measurements from the MPFM, errors can still be calculated for edges 2 and 6.

8.3 Simulation cases

Two simulation cases are presented in this section to assess the performance of the state estimators presented in Chapters 4 and 5. Before the cases are simulated, the model is calibrated through a simulated single-rate well test. In the first case, a well adjustment is performed, and in the second case the turret choke is gradually opened to provoke riser slugging behaviour. Thus, the first case includes both stable operation (before and after the adjustment) and some dynamic/transient behaviour, while the second case displays severe dynamic behaviour. In all cases, the simulation is started at steady-state. To save space, only selected plots (choke positions, liquid rates and holdups) of the simulation results are shown in this section. Additional plots are included in Appendix D, which include oil, gas and water rates, pressures and mass estimates.

8.3.1 Model calibration

A "pre-case" was run with no dynamics at all, with the aim of calibrating the model for the subsequent simulation cases. Both wells were set to produce at constant rates, and the choke valve positions were fixed throughout the simulation. Such a case may be typical when one or both wells are routed through the multiphase flow meter to determine the well flow rate for the current pressure and temperature conditions. The choke valves were positioned as shown in Table 8.2.

Choke	Position
Well 0	30 %
Well 1	20 %
Turret	30 %

Table 8.2: Choke settings for calibration run.

Twenty evenly spaced samples were extracted from the measurement time series and used for calibration, which amounts to solving Problems 6.1 (IPR), 6.2 (WPC) and 6.3 (choke) once for each well, and 6.3 for the turret choke, with $m = 20$. The calibration run resulted in IPR offsets of $\Delta p^0 = 6.99$ bar and $\Delta p^1 = 3.19$ bar, WPC offsets of $\Delta p^2 = 0.44$ bar and $\Delta p^3 = 6.46$ bar, and choke calibration factors of $c^0 = 1.55$, $c^1 = 1.21$ and $c^t = 1.60$. The GORs was found to be $r^0_{go} = 322.2$ Sm^3/Sm^3 and $r^1_{go} = 222.1$ Sm^3/Sm^3, while the water cuts were found to be $r^0_{wc} = 0.007$ and $r^1_{wc} = 0.0262$. No plots were generated from the calibration run, as all the variables are stationary.

8.3.2 Case 1: Well adjustment

Case 1 is designed to simulate a well adjustment. Wells are regularly adjusted to keep production optimal, so such a case may be relevant when a new set of optimal settings have been determined through production optimization. Here, the two wells are assumed to be producing at steady-state, where the wellhead chokes are set at 30 % and 20 %, respectively. Then, well adjustments are made such that choke 0 is ramped dowm from 30 % to 20 % and choke 1 is ramped up from 20 % to 30 %. The turret choke is set at 30 % throughout the case (see Fig. 8.2). These choke settings lead to an initial drop in total production as choke 0 is ramped down, followed by an increase as choke 1 is ramped up. The total length of the simulation is eight hours.

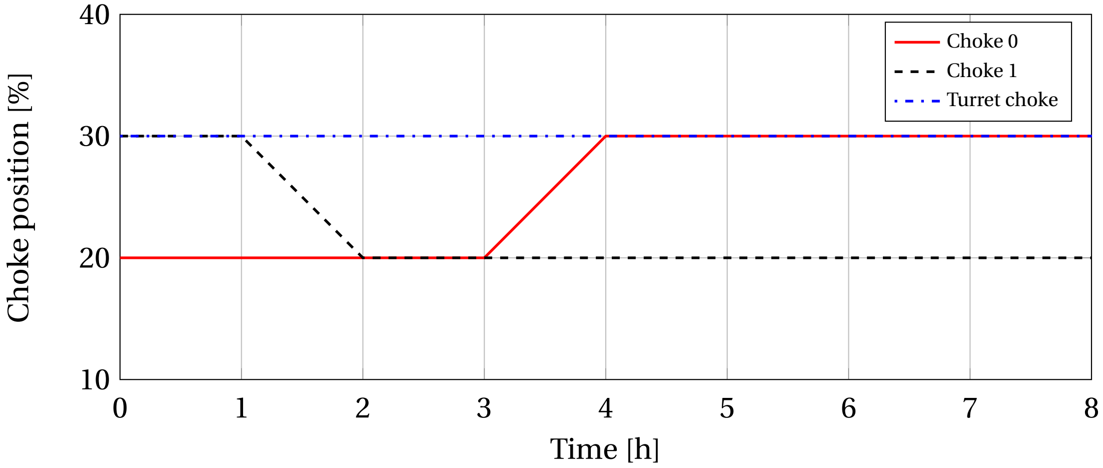
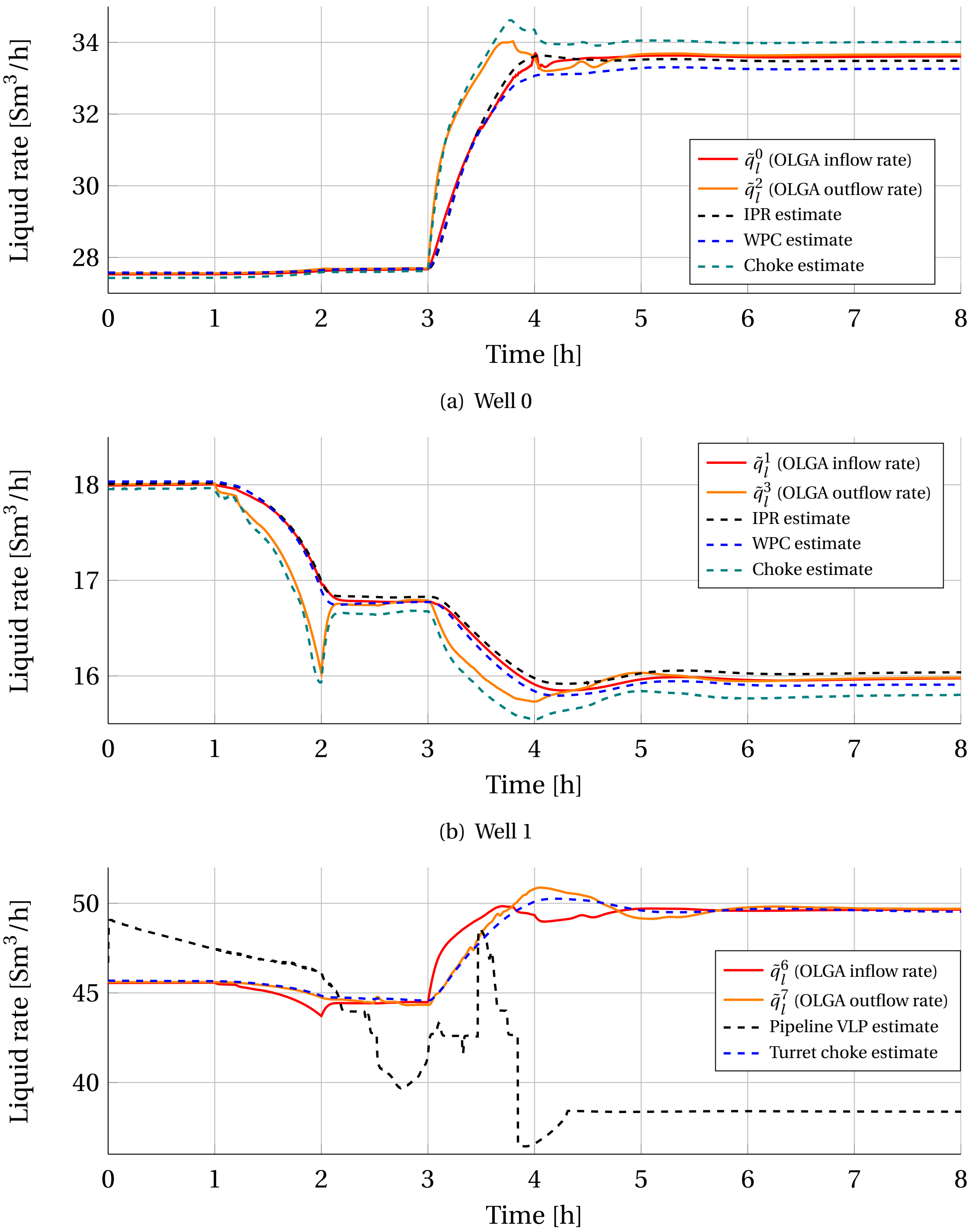


Figure 8.2: Choke positions, well adjustment case.

Static estimation - Naive model evaluation: As an introduction, and to get an initial feel about the quality of each individual model, it is useful to see how the liquid rates are estimated when a single model is used for estimation. This is comparable to estimating rates "manually", i.e. by simply taking the observed pressure and temperature conditions and evaluating each model to obtain the liquid rate estimate. This is accomplished by running several simulations in which the pressure measurement errors are weighted heavily and only one of the model errors is weighted. For example, to see the estimates generated by the WPC curves, the WPC curve model errors are weighted heavily while the rest of the model errors are not weighted at all. For the flowline VLP and turret choke models, the WPCs are weighted to obtain a reasonable estimate for pipeline GOR and water cut, but the mass balance constraints are relaxed to give each respective model complete control over the estimated rates. The resulting estimates are shown in Figure 8.3. Although the results are to be discussed in Chapter 9, some initial observations are made: We see that the choke models give reasonable estimates of ouflow (not surprisingly, since the chokes are placed at outlets), and the IPR/well performance curves give reasonable estimates of inflow. However, the pipeline VLP estimate is poor, which is due to high model sensitivity. This is discussed further in Section 8.6; for now, it suffices to say that the pipeline VLP is used with caution hereafter.



(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.3: Well adjustment case, static estimation with naive model evaulation - Measured and estimated liquid flow rates.

Static estimation: Two estimation runs were performed on the well adjustment case using the static estimator. Both runs were configured with large weights on the pressure measurement errors in an attempt to force the reconciled pressures close to the measured pressures[1]. The estimated model error variances from the calibration run were used for the model error weights[2], however, the pipeline VLP weight was divided by 10 due to the poor estimate seen in the previous test run. The only difference between the two runs was the configured weight on the wellhead choke models; the second run places extra weight on these. The point of running the case twice was merely to illustrate how the estimates are influenced by different weighting. The sampling time was $\Delta t = 10$ s, i.e. Problem 4.1 was solved every 10 seconds. For the first run, Figure 8.4 shows the measured and estimated liquid flow rates. Additional plots are shown in Figures D.1 (flow rates) and D.2 (pressures). For the second run, Figure 8.5 shows the measured and estimated liquid flow rates. Additional plots are shown in Figures D.3 (flow rates) and D.4 (pressures).

The first run (Fig. 8.4) gave good estimates of well inflow ($E^0_{\text{max}} = 0.62$ % and $E^1_{\text{max}} = 0.12$ %) compared to well outflow ($E^2_{\text{max}} = 3.93$ and $E^3_{\text{max}} = 1.79$ %). The flowline estimates have quite large errors compared to the well inflow estimates ($E^6_{\text{max}} = 3.31$ % and $E^7_{\text{max}} = 2.67$ %. However, as seen in Fig. 8.4, the steady-state estimates are reasonably close to the OLGA measurements. The second run (Fig. 8.5) gave good estimates of well outflow ($E^2_{\text{max}} = 1.12$ % and $E^3_{\text{max}} = 0.51$ %) compared to well inflow ($E^0_{\text{max}} = 3.69$ % and $E^1_{\text{max}} = 1.43$ %). The flowline inflow estimate is good compared to the previous case ($E^6_{\text{max}} = 0.98$ %), however, the pipeline outflow estimate is slightly worse ($E^7_{\text{max}} = 3.78$ %).

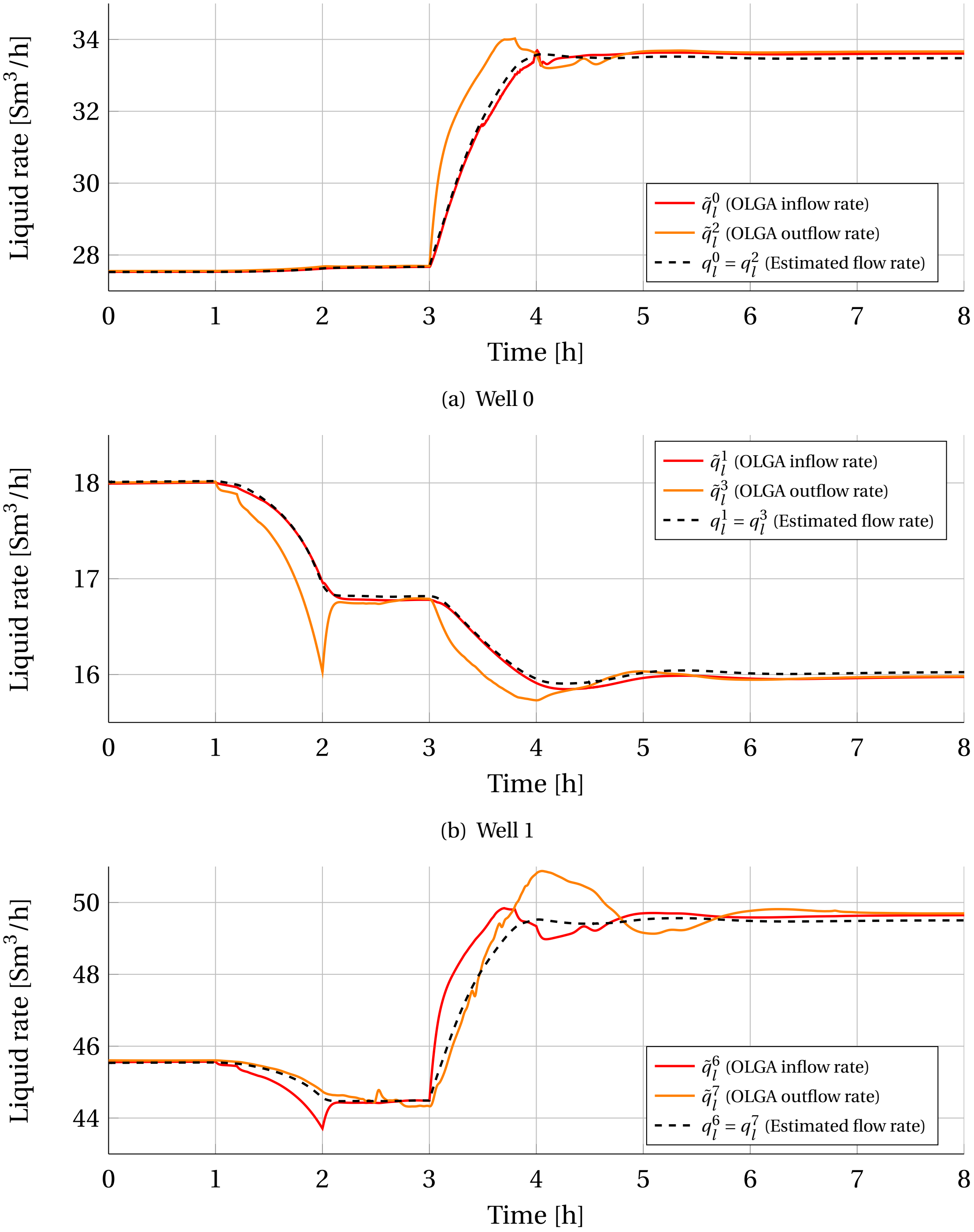
Dynamic estimation: A third run was made for the well adjustment case, this time with the dynamic estimator. The weighting was similar to the second static estimation run described in the previous section, i.e. with extra weight on the choke models. The well and pipeline pressure drop models were also assigned relatively large weights. The pipeline VLP was not weighted at all, since it is not part of the dynamic model. The time horizon was set to $N = 5$, which corresponds to a window of six minutes with the sampling time $\Delta t = 30$ s. In other words, Problem 5.1 was solved every 30 seconds, where 6 minutes of preceding measurements were taken into consideration. The estimated liquid rates and holdups are shown in Figures 8.6 and 8.7, respectively. Additional plots are shown in Figures D.5 (flow rates), D.6 (pressures) and D.7 (masses).

The maximum estimation errors were in general smaller than in the static estimation case ($E^0_{\text{max}} = 0.74$ %, $E^1_{\text{max}} = 0.17$ %, $E^2_{\text{max}} = 1.50$ %, $E^3_{\text{max}} = 0.62$ %, $E^6_{\text{max}} = 1.11$ % and $E^7_{\text{max}} = 3.56$ %). As seen in Fig. 8.6, the outflow estimates for the wells and flowline have more steady-state offset at the new operating point (> 6 h) than in the static case.

[1] This is sensible since we know that the measured pressures from OLGA are correct and noise-free.

[2] These error variances may be misleading, since we have only tested one flow rate. However, the weights seem to work well nonetheless, so we choose to keep them.

Consequently, the ISEs are relatively large ($E_{\mathrm{ise}}^{2} = 2609$, $E_{\mathrm{ise}}^{3} = 996$ and $E_{\mathrm{ise}}^{7} = 10997$). The liquid holdups (Fig. 8.7) are offset from the values from OLGA, however, the qualitative dynamic behaviour is captured nicely.

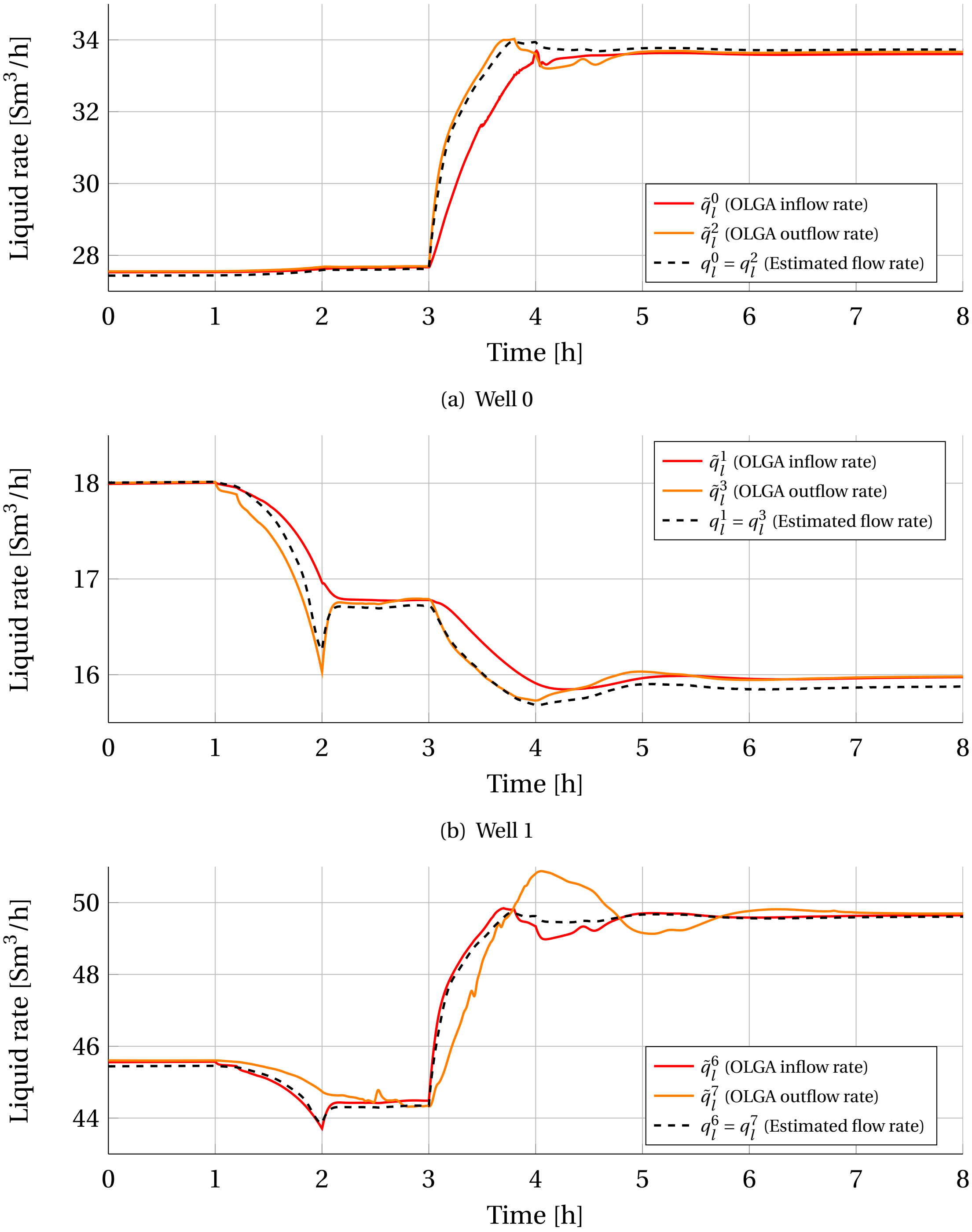


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.4: Well adjustment case, static estimation - Measured and estimated liquid flow rates.

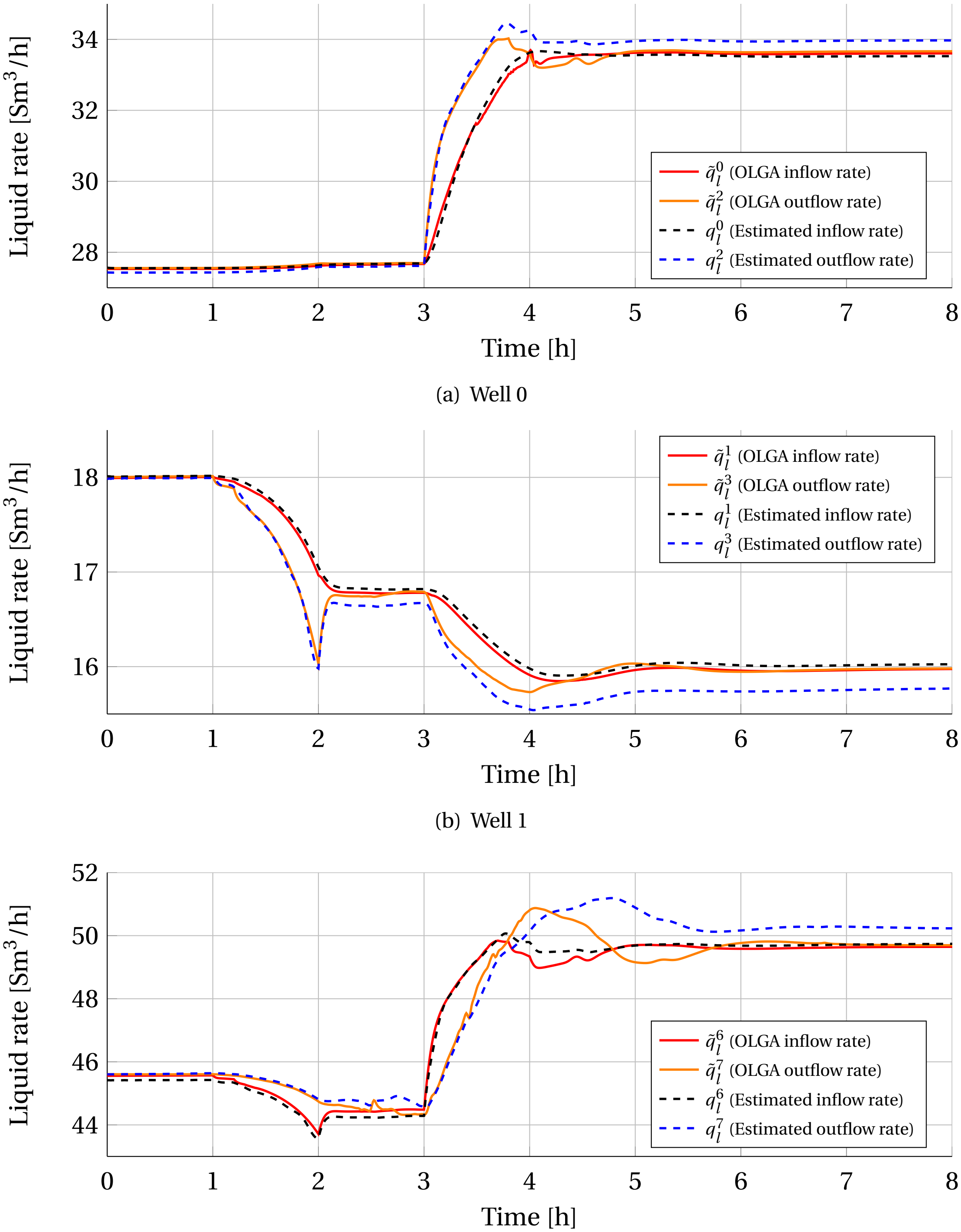


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.5: Well adjustment case, static estimation with extra weight on choke models - Measured and estimated liquid flow rates.

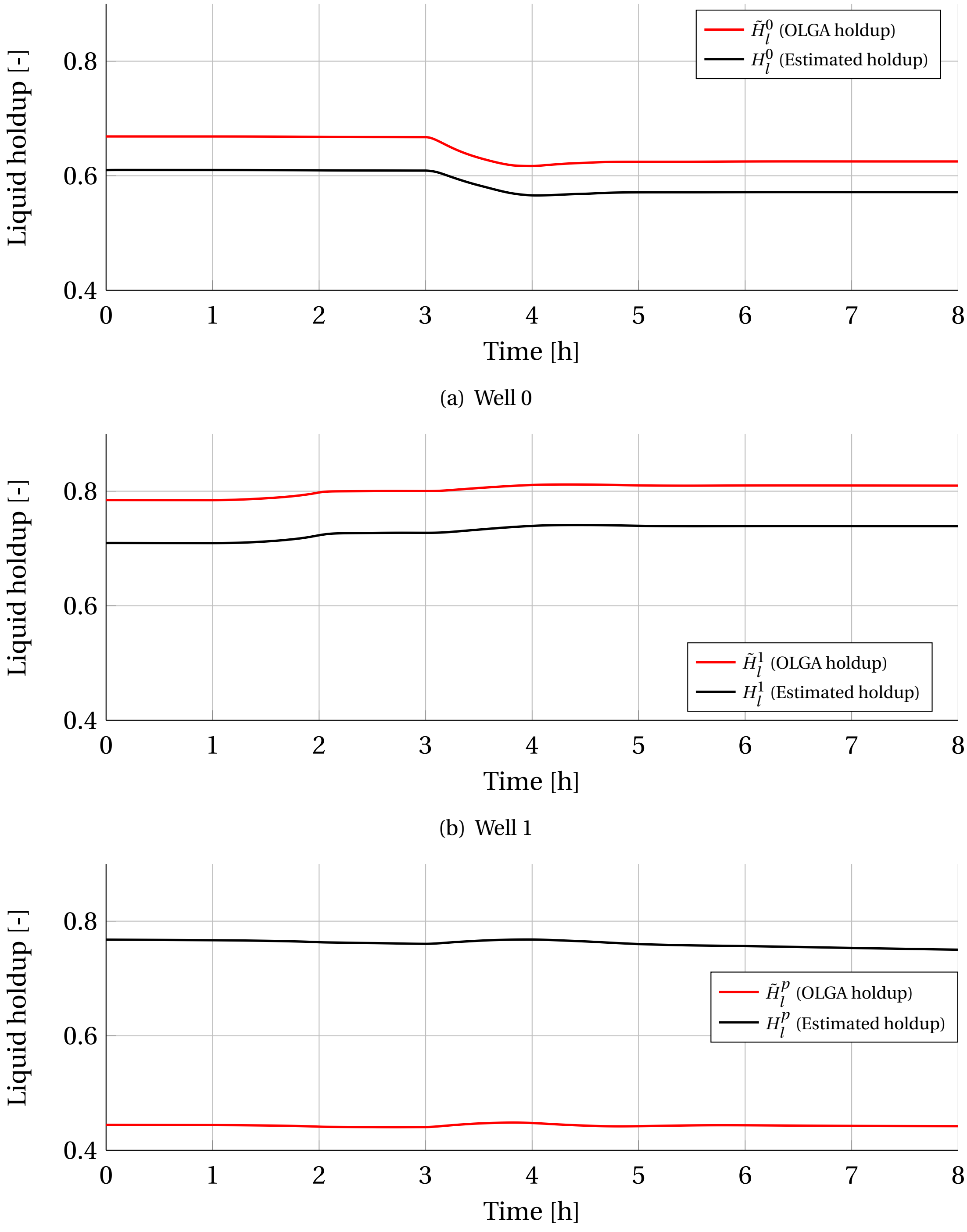


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.6: Well adjustment case, dynamic estimation - Measured and estimated liquid flow rates.



(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.7: Well adjustment case, dynamic estimation - Measured and estimated liquid holdups.

8.3.3 Case 2: Riser slugging

The second case is a riser slugging case. The term *riser slugging* refers to an irregular flow condition in the flowline/riser in which liquid blocks the entire cross-section of the riser low point. This blockage is known as a *slug*, and when it forms, gas pressure builds up behind it. When the pressure is great enough, the slug is blown out of the riser. Then, the process repeats itself as a new slug forms, leading to oscillating flow dynamics. Riser slugging, or *severe slugging*, is a serious challenge for flow assurance engineers as the liquid blowout can cause damage to downstream equipment. Hence, measures must be taken to prevent it, e.g. by feedback control (see e.g (Siahaan et al., 2005; Jahanshahi, 2013)). Although this is not a typical operating scenario, it would be interesting to see how the estimators perform when subjected to such a case. To induce slugging, the wellhead choke valves are left at the positions they were at the end of Case 1, and the turret choke is ramped up until slugging occurs. The choke positions are shown in Figure 8.8.

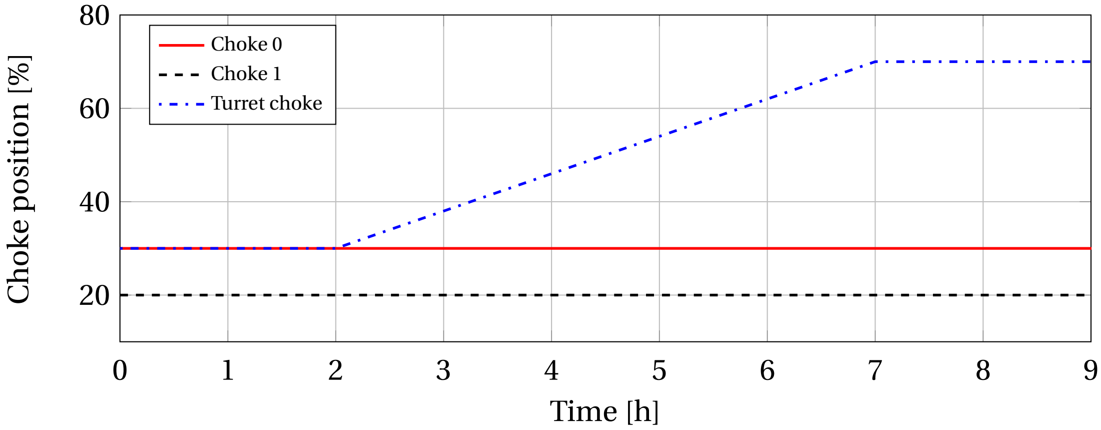


Figure 8.8: Choke positions, riser slugging case.

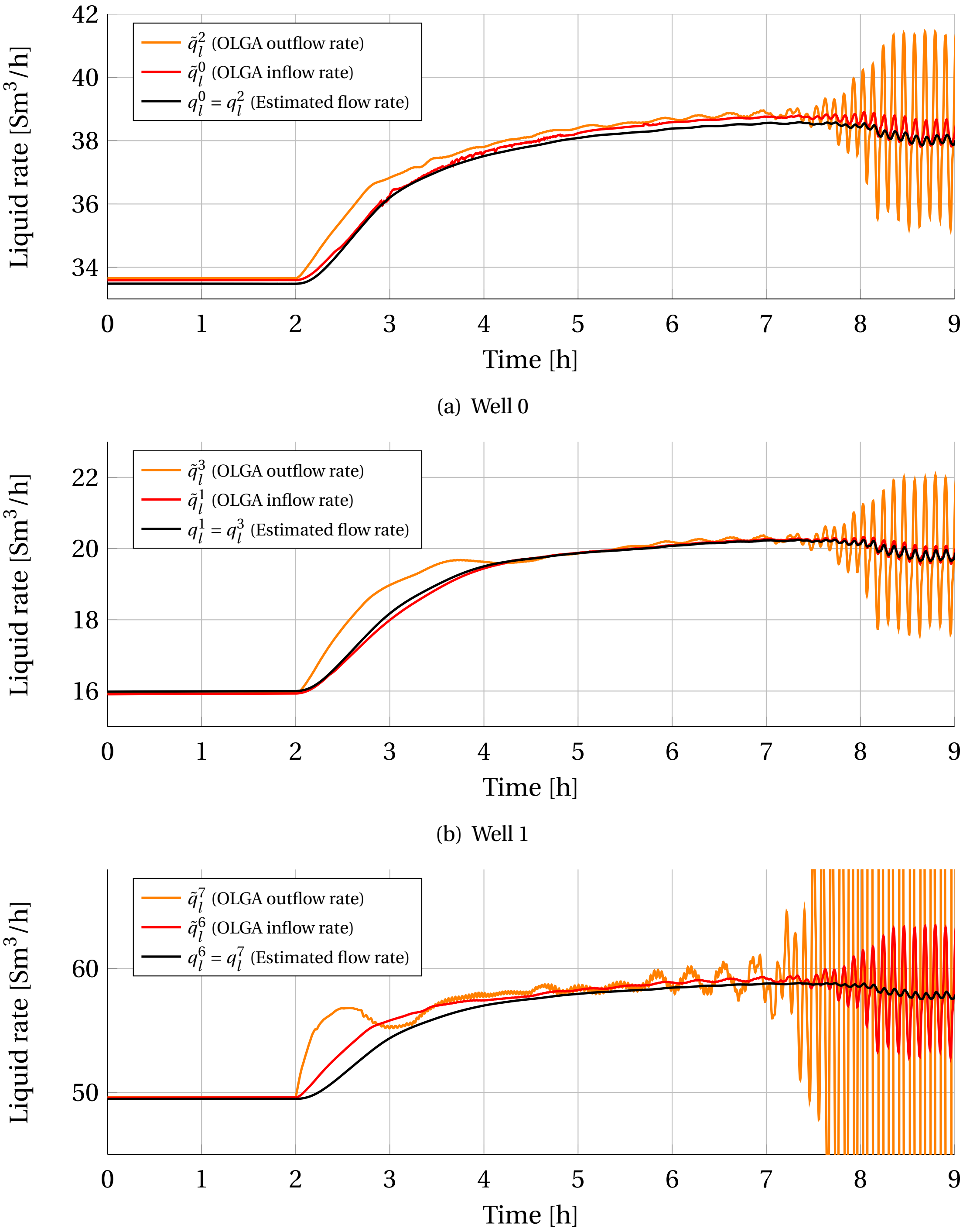
Static estimation: The static estimator was run with the same parameters as in the first static estimation run in Case 1, with one exception: The pipeline VLP was not weighted at all, as this seemed to deteriorate the estimates quite significantly. The sampling time was 10 seconds. Figure 8.9 shows the measured and estimated liquid flow rates. Additional plots are shown in Figures D.8 (flow rates) and D.9 (pressures).

For the wells, the rate estimates tend to follow the inflow measurements from OLGA ($E^0_{\max} = 0.29$ % and $E^1_{\max} = 0.14$ %) rather than the outflow measurements ($E^2_{\max} = 1.89$ % and $E^3_{\max} = 1.17$ %). The flowline rate estimate gives a fairly good prediction of the OLGA inflow until the onset of slugging at about 7 hours. Then, the estimated flowrate resembles a highly dampened version of the OLGA measurements. This results in relatively large estimation errors, especially for flowline outflow ($E^6_{\max} = 3.06$ % and $E^7_{\max} = 69.2$ %). We also note that even though the estimated flow rates do not keep up with the oscillations, they tend to follow the small drop in the "mean" flow rate seen as slugging starts to occur.

Dynamic estimation: The dynamic estimator was run with manually tuned parameters (see Table D.3), a sampling time of 30 seconds and a time horizon of 5. Measured and estimated liquid rates are shown in Figure 8.10, while measured and estimated liquid holdups are shown in Figure 8.11. Additional plots are shown in Figures D.10 (flow rates), D.11 (pressures) and D.12 (masses).

For the wells, the flow rate estimates are in general better than in the static case ($E^0_{\max} = 0.28$ %, $E^1_{\max} = 0.14$ %, $E^2_{\max} = 1.54$ % and $E^3_{\max} = 0.88$ %), and we see that the estimated outflow rate oscillates along with the measured OLGA rate as slugging occurs, albeit slightly dampened. For the flowline, the inflow estimate is comparable to the estimate from the static case, with similar behaviour and a slightly smaller maximum estimation error ($E^6_{\max} = 2.42$ %). The outflow estimate overpredicts the flow rate as the turret choke starts opening. Just before six hours have elapsed, the in/outflow estimates collapse, which happens when the flowline is emptied of gas (see Fig. D.12). When slugging occurs the outflow estimate oscillates along with the OLGA outflow rate, but reaches its upper bound of 100 Sm^3/h. Since the dynamic estimator is able to predict this oscillating flow, the maximum estimation error is smaller than in the static case ($E^7_{\max} = 47.7$ %). However, the aforementioned overshoot gives a larger ISE than in the static case ($E^7_{\mathrm{ise}} = 18.7 \cdot 10^6$ for the static case and $E^7_{\mathrm{ise}} = 37.7 \cdot 10^6$ for the dynamic case).

The liquid holdup estimates for the wells behave similarly to the well adjustment case, i.e. some offset but reasonable dynamic behaviour. The estimated flowline holdup in generally quite a bit larger than the measured holdup from OLGA, and also decreases much more than the measured holdup until the flowline is emptied of gas; then it remains stationary until slugging occurs. Some oscillation can be seen after 8 hours, which corresponds to the oscillations seen in the measured holdup.

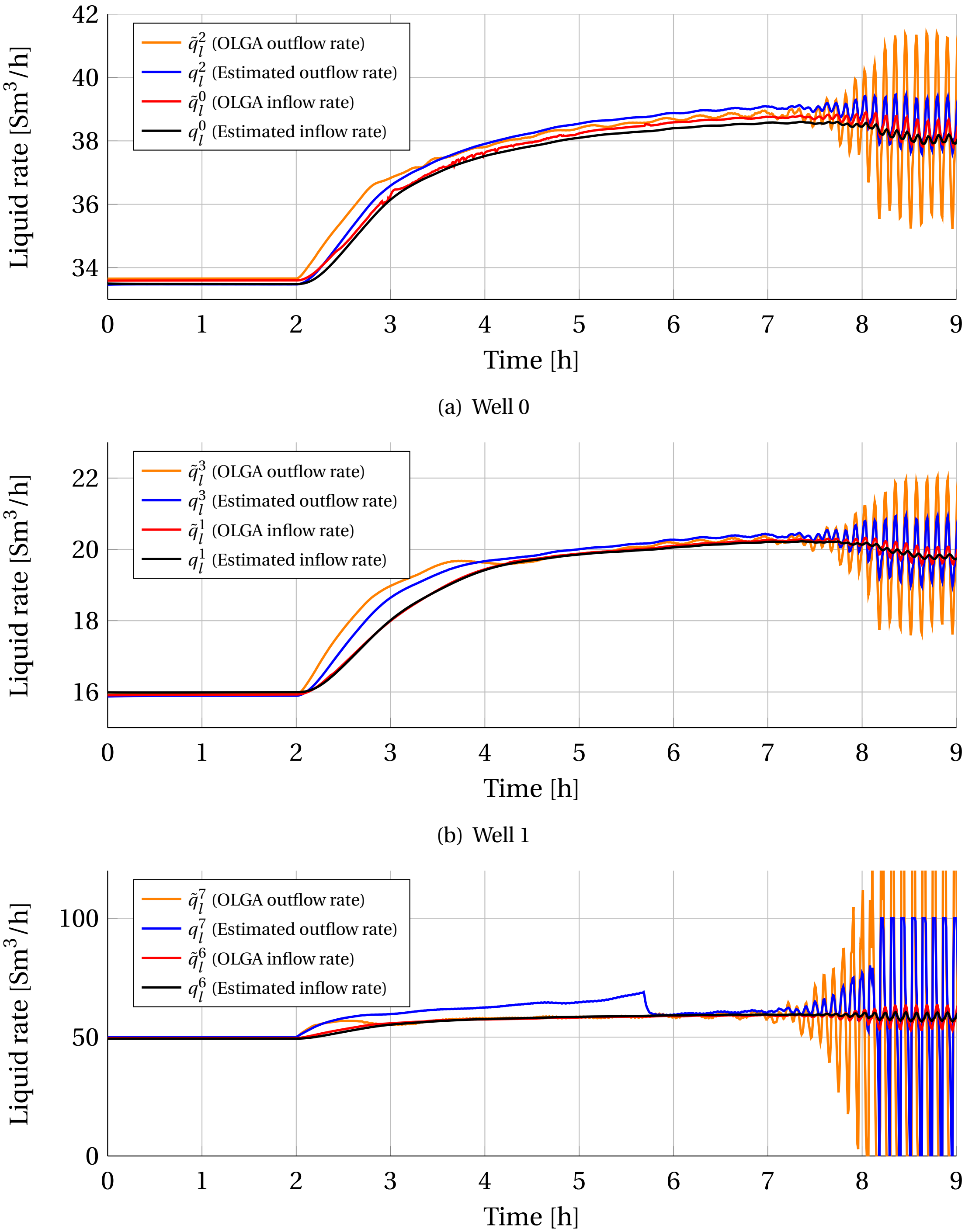


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.9: Riser slugging case, static estimation - Measured and estimated liquid flow rates.

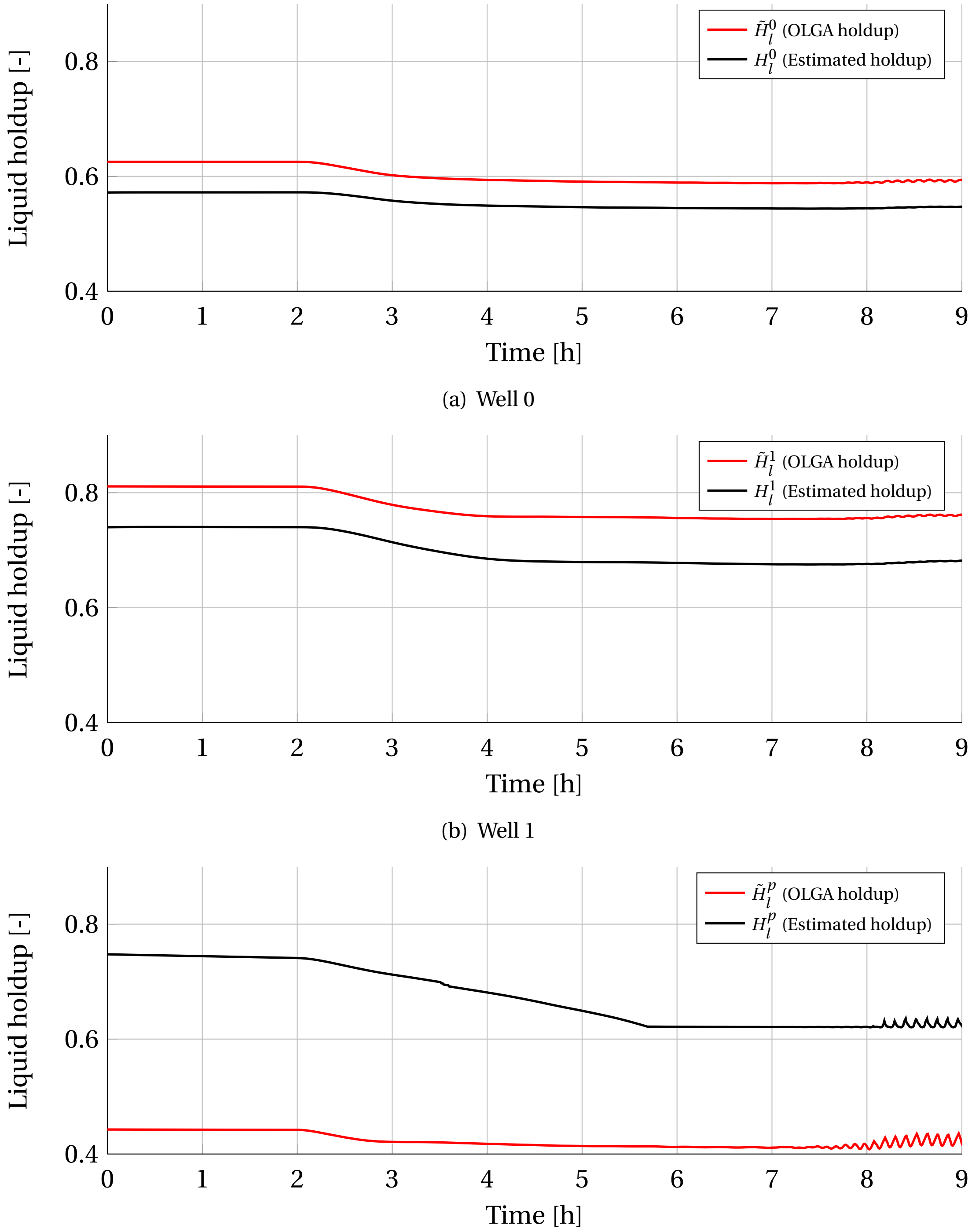


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.10: Riser slugging case, dynamic estimation - Measured and estimated liquid flow rates.



(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.11: Riser slugging case, dynamic estimation - Measured and estimated liquid holdups.

8.4 Field data case

As mentioned in the introduction, the Skarv field is equipped with multiphase flow metering systems (MPFM) on each template, which enables measurement of a single flow rate in the template (either a single well flow rate or the flow rate from both wells, as described in Section 6.2). BP have generously granted access to their database of historical measurements from the Skarv field, which allows us to test the performance of the estimators on a real-world case. A time series of 30 hours was found which includes positioning of all three chokes (see Fig. 8.12). About 20 hours into the time series, the routing into the MPFM is changed, meaning that two flow rates (well 0 and pipeline) can be compared to the estimates.

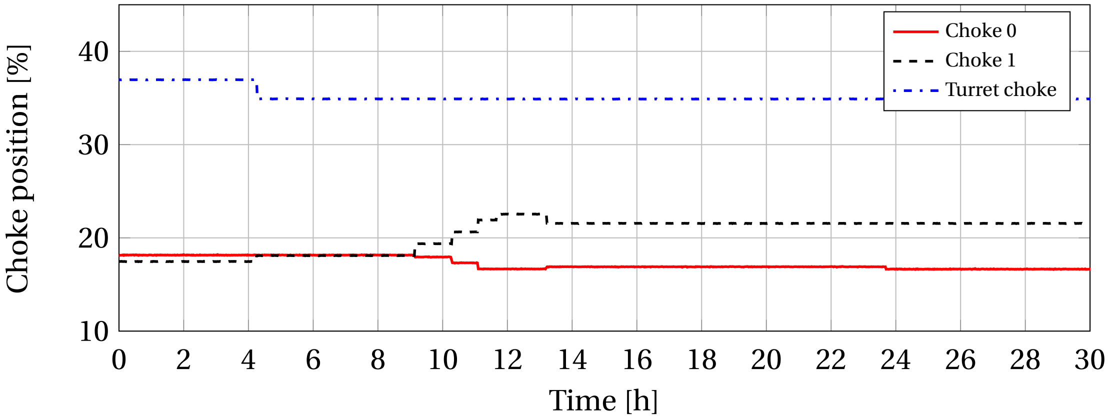


Figure 8.12: Choke positions, field data case.

Calibration was performed by using data from single-rate well tests (also supplied by BP), i.e. solving Problems 6.1 (IPR), 6.2 (WPC) and 6.3 (choke) once for each well, and 6.3 for the turret choke, with $m = 1$. In this case, well 0 was tested one day before the time series starts, Well 1 and Well 1/Well 0 (pipeline) were tested about a week and a half before the time series starts. The calibration resulted in the following parameters; $\Delta p^0 = -5,19$ bar, $\Delta p^1 = -1.52$ bar, $\Delta p^2 = -2.71$ bar, $\Delta p^3 = -9.24$ bar, $c^0 = 0.44$, $c^1 = 0.23$, $c^t = 2.11$, $r^0_{go} = 293.4$ Sm^3/Sm^3, $r^1_{go} = 193.5$ Sm^3/Sm^3, $r^0_{wc} = 0.015$, and $r^1_{wc} = 0.049$.

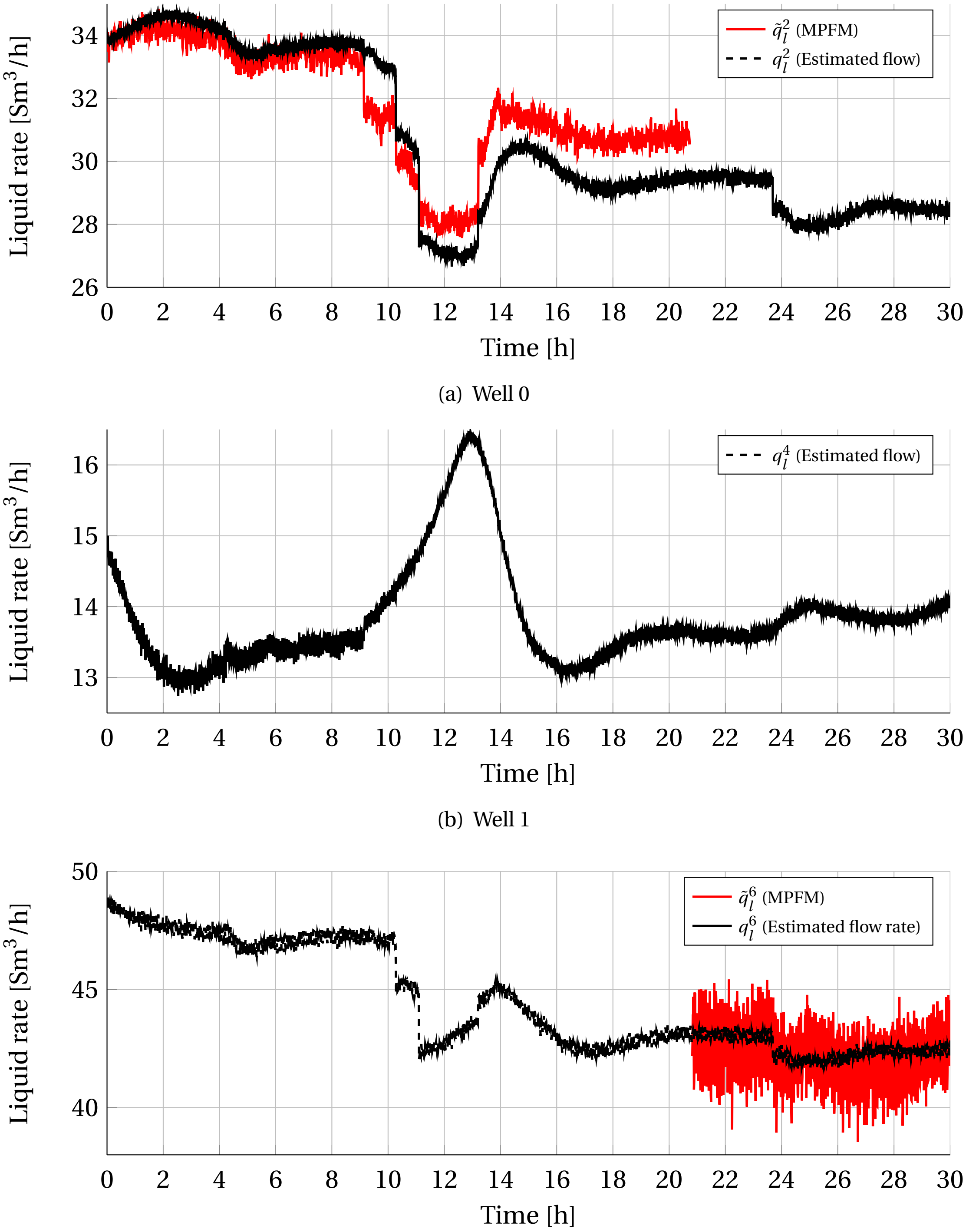
Static estimation: The static estimator was run with manual tuning, no weight on the pipeline VLP, and a sampling time of 30 seconds. Figure 8.13 shows the measured and estimated liquid flow rates. Additional plots are shown in Figures D.13 (flow rates) and D.14 (pressures).

At a glance, the flow rates correspond well with the available measurements from the MPFM. Both the estimates and measurements look noisy, but this is due to small pressure oscillations and a slightly deceptive time scale. In hindsight, it may have been a good idea to apply smoothing to the pressure measurements before feeding them to the estimation algorithm. The maximum estimation errors are $E^2_{\text{max}} = 5.60$ % (well 0) and $E^6_{\text{max}} = 8.54$ % (flowline). However, we see that by smoothing the data and looking at mean errors, these numbers would be smaller.

Dynamic estimation: The dynamic estimator was run with manual tuning, a sampling time of 90 seconds and an estimation horizon of 20. Before starting the estimation, $\tilde{p}^6$ (manifold pressure measurement) and $\tilde{p}^7$ (upstream turret choke pressure measurement) were preconditioned with a moving average filter with a window length of 25. This was necessary due to large pressure fluctuations (the effect of the filter can be seen by comparing Figs. D.14 and D.16). Measured and estimated liquid rates are shown in Figure 8.14, while estimated liquid holdups are shown in Figure 8.15. Additional plots are shown in Figures D.15 (flow rates), D.16 (pressures) and D.17 (masses).

In this case, the maximum estimation error for well 0 is marginally larger than in the static case ($E^2_{\text{max}} = 5.99$), and slightly smaller for the flowline ($E^6_{\text{max}} = 7.18$ %. Measurements are not available for the remaining rates, but from a strictly qualitative standpoint, they seem quite reasonable.

No holdup measurements are available for comparison.

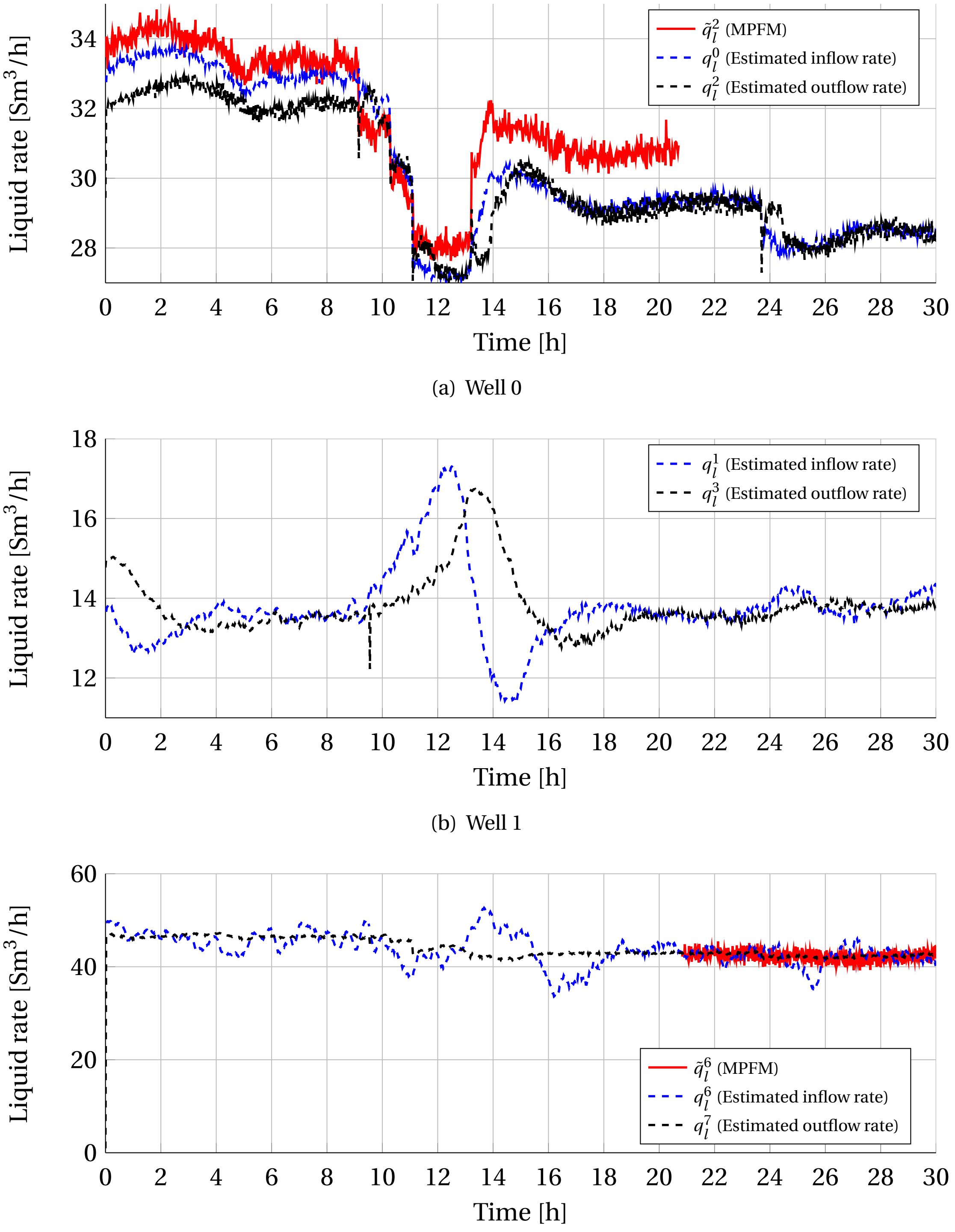


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.13: Field data case, static estimation - Measured and estimated liquid flow rates.

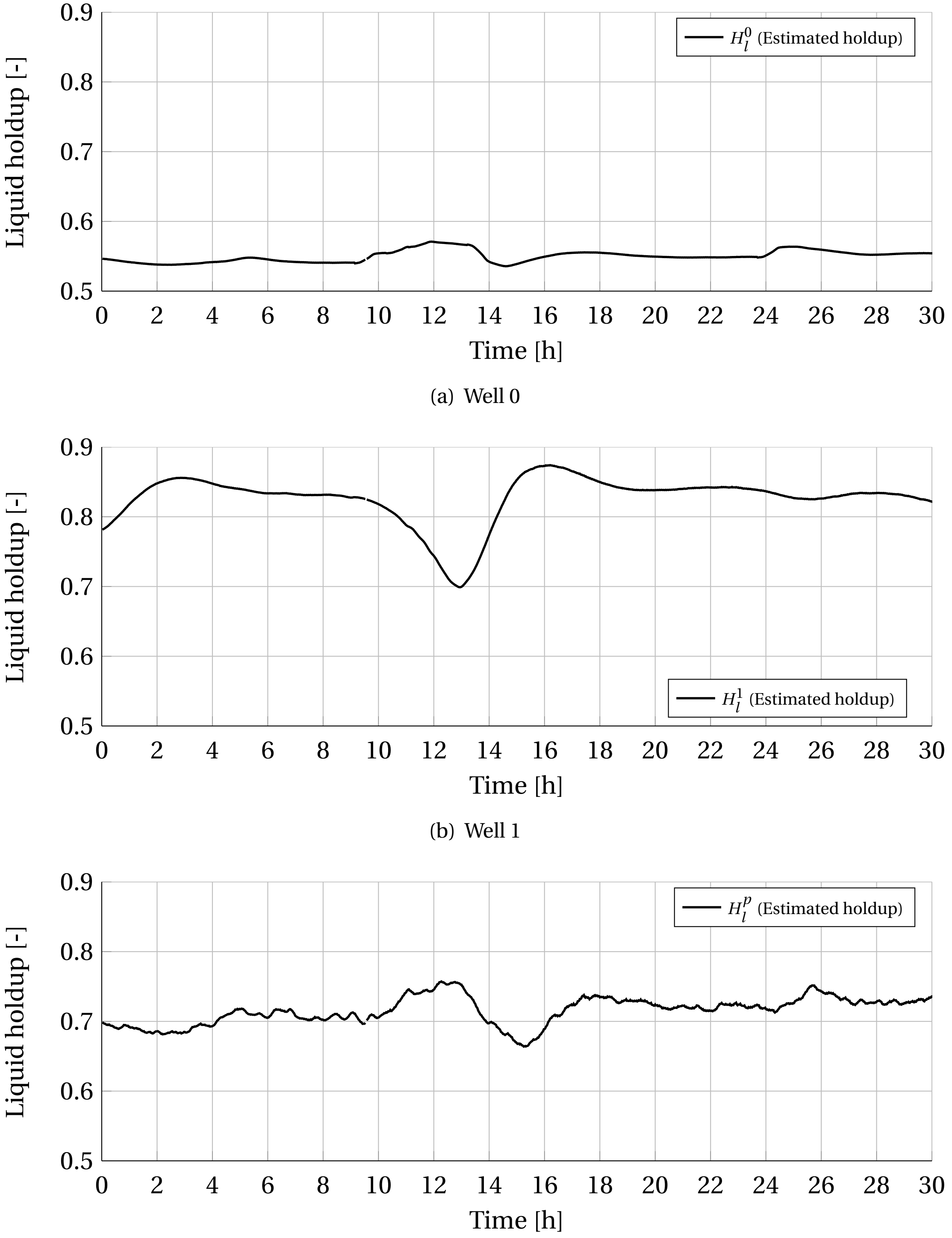


(a) Well 0

(b) Well 1

(c) Pipeline

Figure 8.14: Field data case, dynamic estimation - Measured and estimated liquid flow rates.



(a) Well 0

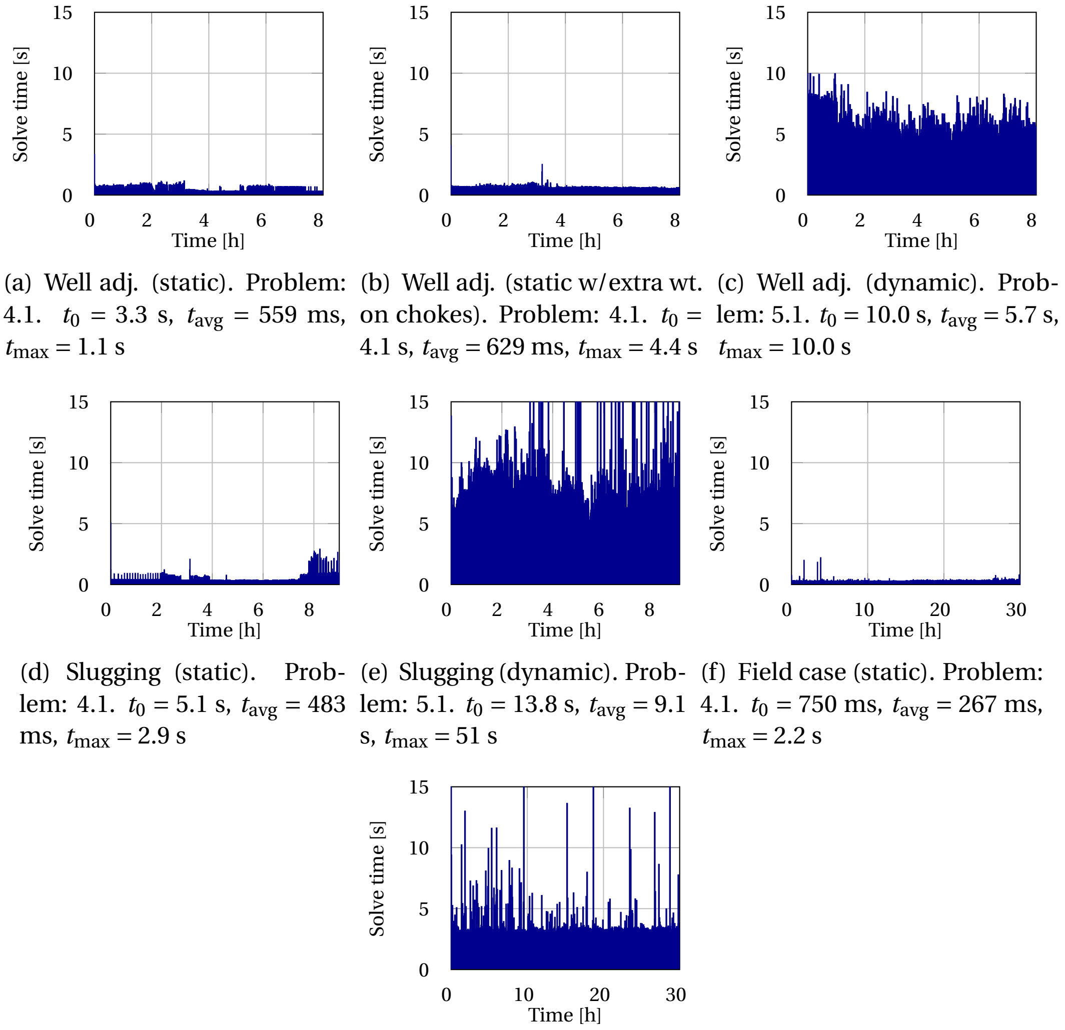
(b) Well 1

(c) Pipeline

Figure 8.15: Field data case, dynamic estimation - Measured and estimated liquid holdups.

8.5 Solution times

In this section, we present the times required to solve the flow estimation optimization problems (Problem 4.1 or 5.1) in Figure 8.16. Some important statistics are provided in each caption, namely t_0 (the solution time for the first iteration), t_{avg} (the average solution time), and t_{max} (the maximum solution time). The problems were solved on a Dell laptop with a 64-bit Ubuntu 13.10 operating system, a 2.7 GHz Intel Core i7-3740QM CPU (only one core was used) and 8 GB of RAM.



(a) Well adj. (static). Problem: 4.1. $t_0 = 3.3$ s, $t_{\text{avg}} = 559$ ms, $t_{\text{max}} = 1.1$ s

(b) Well adj. (static w/extra wt. on chokes). Problem: 4.1. $t_0 = 4.1$ s, $t_{\text{avg}} = 629$ ms, $t_{\text{max}} = 4.4$ s

(c) Well adj. (dynamic). Problem: 5.1. $t_0 = 10.0$ s, $t_{\text{avg}} = 5.7$ s, $t_{\text{max}} = 10.0$ s

(d) Slugging (static). Problem: 4.1. $t_0 = 5.1$ s, $t_{\text{avg}} = 483$ ms, $t_{\text{max}} = 2.9$ s

(e) Slugging (dynamic). Problem: 5.1. $t_0 = 13.8$ s, $t_{\text{avg}} = 9.1$ s, $t_{\text{max}} = 51$ s

(f) Field case (static). Problem: 4.1. $t_0 = 750$ ms, $t_{\text{avg}} = 267$ ms, $t_{\text{max}} = 2.2$ s

(g) Field (dynamic). Problem: 5.1. $t_0 = 184$ s, $t_{\text{avg}} = 3.6$ s, $t_{\text{max}} = 63$ s

Figure 8.16: Solution times for flow estimation optimization problems.

8.6 Pipeline VLP sensitivity

We saw in Figure 8.3 that each individual model was able to provide acceptable flow rate estimates, with one notable exception, namely the pipeline VLP. This is interesting, and we will explore this further in this section. To get a handle on what is going on, we look at a snapshot from the well adjustment case. Specifically, we look at the run with relaxed mass balance constraints and weighting on the pipeline VLP only, i.e. the flow rate in the pipeline is determined from the VLP. The objective function is in this case

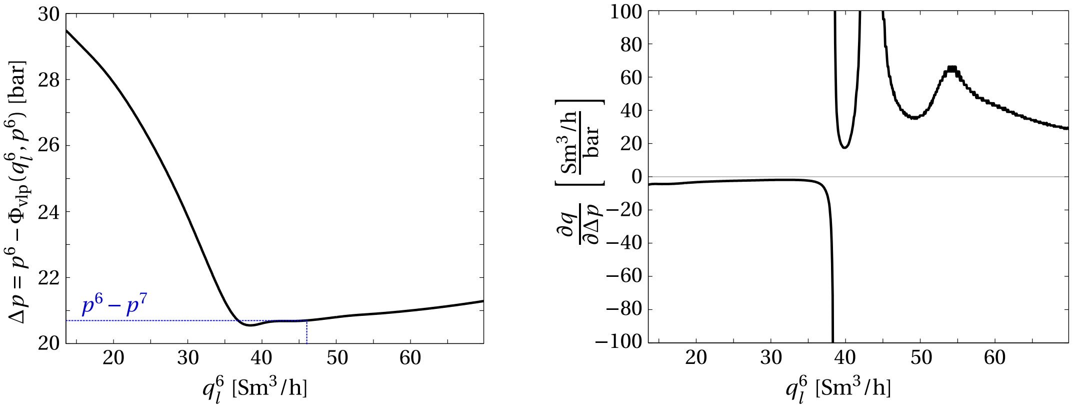
$$J(w_{\text{vlp}}, \mathbf{v}) = \left| w_{\text{vlp}} \right|^2 + \|\mathbf{v}\|^2_{\mathbf{R}^{-1}}, \tag{8.4}$$

subject to the pipeline VLP model (4.25). $\mathbf{R}^{-1}$ is selected with large weights such that the reconciled pressures follow the measurements closely (making $\|\mathbf{v}\|^2_{\mathbf{R}^{-1}} \approx 0$). Some additional constraints are in place to calculate the pipeline GOR and water cut using the WPCs, but the mass balance constraints are relaxed such that the flow rate in the pipeline is determined from (4.25) only. The snapshot is taken at 1 hour and 22 minutes, where the conditions are as shown in Table 8.3 below:

Symbol	Description	Value
q_l^6	Estimated liquid flow rate	46.0 Sm^3/h
p^6	Reconciled manifold pressure	114.20 bara
$\tilde{p}^6$	Measured manifold pressure	114.20 bara
p^7	Reconciled outlet pressure	93.46 bara
$\tilde{p}^7$	Measured outlet pressure	93.46 bara
r_{go}^p	Pipeline GOR	284.5 Sm^3/Sm^3
r_{wc}^p	Pipeline water cut	0.089
$\tilde{T}^6$	Manifold temperature	31.4 °C

Table 8.3: Snapshot of pipeline VLP run.

We now fix the manifold pressure, manifold temperature, GOR and water cut, and plot the pipeline pressure drop as a function of liquid flow rate. This is done by evaluating $\Phi_{\text{vlp}}(\cdot)$ at increasing flow rates to obtain the downstream pressure, and taking the pressure drop as $\Delta p = p^6 - \Phi_{\text{vlp}}(q_l^6, p^6)$ (p^6 and the remaining variables in $\Phi_{\text{vlp}}(\cdot)$ are fixed). This is shown in Figure 8.17(a).



(a) Pipeline pressure drop. The snapshot Δp and estimated flow rate is indicated in blue.

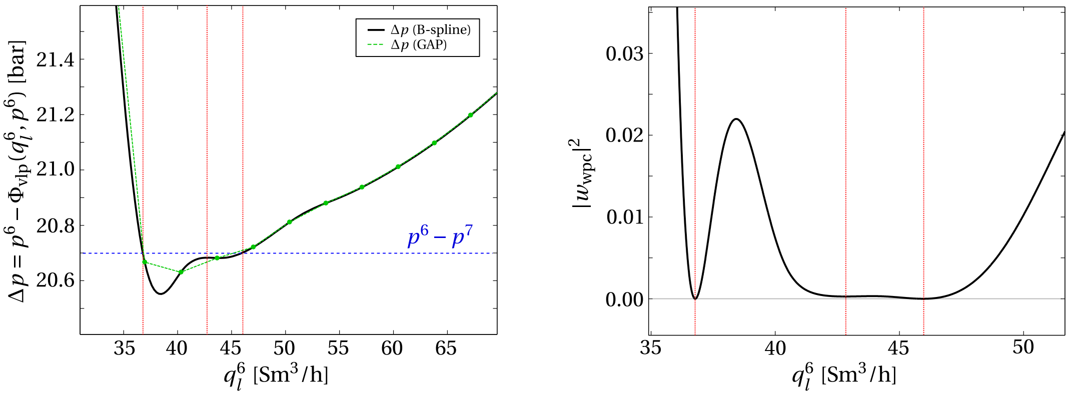
(b) Flow rate sensitivity.

Figure 8.17: Pipeline VLP sensitivity.

The resulting plot is not immediately intuitive - in a typical $q - \Delta p$ plot we would expect the pressure drop to increase with increasing flow, due to friction. However, as mentioned in Chapter 3, due to the relatively large GOR, the liquid is lifted by gas bubbles, which causes an initial drop as the flow increases. Eventually, at around 40 Sm^3/h, the pressure drop starts to increase again due to friction. In the relevant flow rate range, the frictional term does not become particularly significant; by inspection in GAP, it turns out that friction does not take proper hold until the rate reaches around 150 Sm^3/h. There are two potential problems here.

Firstly, the pressure drop curve is almost flat at our operating point. This means that a small error in differential pressure will result in an large error in flow rate. According to the VLP, we could increase the flow rate by approximately 30 Sm^3/h simply by increasing the pressure drop by 1 bar. In other words, the estimated flow rate will be *very* sensitive to pressure. This sensitivity ($\partial q / \partial \Delta p$) is shown graphically in Figure 8.6, and we see that it behaves well until the flat portion of the pressure drop curve, and then becomes very large. In this sense, we could say that the flow rate is "almost unobservable" at the operating point, since a large range of flow rates will "almost" satisfy the VLP model constraint.

Secondly, the measured pressure drop intersects the pressure drop curve at two different flow rates, which means the objective function of the estimation problem will have two global minima with respect to the VLP. This is illustrated in Figure 8.18, which shows (a) a closer look at the pressure drop curve and (b) the objective function value as a function of flow rate.



(a) Pipeline differential pressure

(b) Objective function value

Figure 8.18: Local optima for pipeline VLP. Flow rates which correspond to local minima of the objective are inidicated with red vertical lines.

This is a concrete example of non-convexity in the estimation problem which shows that an optimal solution of the estimation problem does not necessarily imply that a good flow rate estimate has been found. We also note that the objective function has a third *local* minimum, which is due to the small oscillations caused by the B-spline approximation of the VLP from GAP. Since the estimation problem is solved by a local NLP solver, there are thus *three* flow rate estimates which could potentially be solutions to the estimation problem. In this case, we would not benefit much from using a *global* solver either, since two of the solutions are in fact global minima.

The bottom line here is that the pipeline VLP is not particularly suitable to use as a constraint function *in our estimation problem* when the friction term is insignificant. This makes intuitive sense; the friction term is strongly linked to the flow rate, while the hydrostatic term is not (except for low flow rates when increased gas flow reduces the average density of the multiphase mixture). We would, however, expect the pipeline VLP to be useful in cases with more friction.

Chapter 9

Discussion

In this discussion chapter, we attempt to convey some of the experiences gained in the modelling and simulation parts of the thesis. The main goals in this study were to (1) find suitable models to include in a flow estimation problem, and (2) compare the performance of static and dynamic flow rate estimators with automatic rate control in mind. Based on the assignment text and literature study, the estimation methods of choice were a static weighted least-squares method for the static estimation problem, and moving-horizon estimation for the dynamic estimation problem. The first part of this discussion chapter will be about the choice of models, where we will discuss the choice of using B-splines as a model replacement and write up some general impressions about the practical considerations associated with them. Then, we will proceed to the main part of the discussion, where we discuss the static and dynamic estimators, both in terms of the simulation results from Chapter 8 and in terms of model complexity and robustness. After this, we discuss model calibration briefly. Finally, we give some additional remarks on various topics, before a short section at the end gives some concluding remarks.

An important practical consideration in this thesis has been that the models used for estimation should fit nicely into already existing workflows for e.g. production forecasting and optimization. Large organizations like BP employ teams of experts in every aspect of oil production, and it makes sense to incorporate their knowledge when deriving and maintaining models. In our case, a lot of this knowledge is summarized as well performance and VLP curves in the GAP model used for production optimization, and fluid property tables used by flow assurance and reservoir engineers. It seems only natural to include as much of this information as possible into the flow estimator. Having worked with B-splines before (Robertson, 2013), and considering the excellent results of (Sandnes, 2013; Grimstad & Sandnes, 2014; Grimstad et al., 2014), the choice of using B-splines to approximate the GAP models was clear.

As the GAP model included the necessary models for wells and pipelines only, we needed to find a suitable choke model. The multiplier model was selected for two main reasons. For one, it was simple and easy to understand. Secondly, it allowed us to include more information from in-house BP models by means of flow coefficient characteristics and fluid property tables. In an attempt to account for compressible flow, we used the Newton-iterative approach applied to pipelines in (Beggs & Brill, 1973) to evaluate the fluid property tables at the *average* pressure in the choke as opposed to the upstream pressure. This resulted in a model which could not be written in closed form. However, we could still sample pressure drops, much like we could with the GAP models. Therefore, we used the B-spline once more; by sampling and approximating the choke model, we were able to implement the model with ease, and test the effect of different multiphase multipliers without modifying too much of the code.

9.1 Using the B-spline as an approximation tool

Translating the models into usable B-spline approximations required some work. The B-spline implementation in CENSO requires the user to sample the function to be approximated in a grid structure. It then computes the B-spline approximation by solving a linear system of equations, where the number of unknowns is equal to the total number of sampled points. Ideally, we would sample the functions tightly to capture as much detail as possible. For one-dimensional functions (IPRs and WPCs) and two-dimensional functions (fluid properties) the total number of samples will be relatively small, and the resulting linear system is easy to solve. However, for higher-dimensional models such as pipeline VLPs (five-dimensional) and choke models (four-dimensional for the wellhead chokes and six-dimensional for the turret choke), the linear system quickly grows in size and becomes time-consuming to solve. Time is not really an issue, since the B-spline computation is only performed once (before the actual estimation starts). Once the B-spline has been created, evaluating its value for a given input is very fast. On the other hand, a very large number of samples may be problematic. This is easily illustrated by an example; say we wanted to sample the pipeline VLP in a grid of 20 flow rates, 10 upstream pressures, 10 upstream temperatures, 10 GORs and 10 water cuts (which is the maximum allowed in GAP). This would result in a total of $20 \cdot 10^4 = 200000$ samples, and a linear system with just as many unknowns. However, the laptop computer used in this thesis began struggling with memory capacity when the number of samples exceeded about 100000, thus limiting the number of sampling points.

To obtain a good B-spline approximation of a function, sufficiently tight sampling is necessary in areas with "sharp corners", i.e. areas where the Hessian of the function has large elements, or else the B-spline will display oscillating behaviour. This issue is discussed in some detail in both (Sandnes, 2013) and (Robertson, 2013), and a prime example is shown in Figure 4.4. To make the most of the limited amount of sampling points, two steps were taken. First, the output from the OLGA simulations/historical data were inspected to determine which variable ranges would have to be covered by the B-spline. Then, the functions were inspected in the relevant ranges to determine areas in which dense sampling was required, and where we could get away with sparse sampling. The end result of this process was good in the sense that the resulting B-spline functions represented the nonlinear models well, however, obtaining these B-splines required some insight into both the models themselves and the behaviour of B-splines. Moreover, the approximated models were valid in a limited domain. Consequently, such a process is difficult to automate for the general case; however we must still emphasize that the B-spline is an excellent tool for representing models given by tabular data.

Despite the issues mentioned above, the B-spline has proved itself as a remarkably powerful approximation tool. Rather than listing all the advantages of using the B-spline, we may ask; what possible alternatives do we have? One alternative is to go back to the black-box simulator. However, this approach does not exploit the structure of the model, model evaluation takes longer since the entire network model must converge, and gradients are not (in general) available[1]. A second alternative is to use some other approximation to replace the simulator, for instance the piecewise linear approximation used in (Kosmidis et al., 2005; Gunnerud & Foss, 2009). However, the latter paper reported a large increase in solution time and number of variables when refining the approximation, since the piecewise linearization is handled by introducing binary variables for each sample point, resulting in a large mixed-integer linear program (MILP). Moreover, piecewise linear functions have discontinuous gradients which can cause unpredictable steps in the solver. The (cubic) B-spline eliminates both these problems; it provides both first and second derivatives, and refining it does not increase the number of variables[2]. A third alternative is to directly implement analytic functions as constraints. It goes without saying that using complicated multiphase models directly in an optimization problem (and associated calibration problems) is slightly tedious. Nevertheless, this approach can be found in e.g. (Binder, 2012) for production optimization.

9.2 Comparison of static and dynamic estimators

In this section, we will compare the static and dynamic estimators to each other. We will start with discussing flow rate estimation, as this was the main goal. Later on, we will discuss other important matters, such as robustness and model complexity/solution times. As we mentioned in the introduction, these properties are important if we are to use the flow rate estimates for feedback in an automatic rate control system.

[1] Note that some simulator frameworks *do* provide gradients i.e. JModelica (Nalum, 2013)

[2] When CENSO is set to solve the optimization problem *globally*, B-spline refinement increases the number of auxiliary variables in the relaxed lower bound problems, see (Grimstad & Sandnes, 2014). However, this does not apply for us as we only seek a local solution.

9.2.1 On flow rate estimation

We saw in the Chapter 8 that the static and dynamic estimators performed comparably with respect to estimating flow rates. Estimation errors were about the same for both estimators in most cases, and it was also clear from the simulation plots in Chapter 8 and Appendix D that both estimators provided reasonable estimates. The riser slugging case was an exception, which illustrated that the dynamic estimator may perform slightly better when there is severe dynamic behaviour in the production network, but that this increased performance comes at the cost of some erratic estimates; in our case, the pipeline outflow estimate was quite bad. This is related to robustness, and we will discuss it further in the next section.

Before implementing and testing the estimators, one might expect that an estimator based on static models would perform well during stable operation and poorly during transients, i.e. when the system is moving from one operating point to another. An estimator equipped with a dynamic model would perhaps be expected to improve the transient behaviour, but we would most likely have to pay the price of a more complicated model. After putting the estimators to the test, some of these expectations have been fulfilled, while others have not. First of all, the static estimator does not necessarily do a bad job of estimating flow transients. This was seen in the very first test case (Fig. 8.3), where each model was assigned to estimate flow rates individually. For the wells, we saw that the IPR and WPC give good predictions of well inflow, while the choke models gave good predictions of outflow. For the pipeline, we saw that the turret choke model gave reasonable outflow estimates, while the pipeline VLP failed to produce good estimates. However, as we saw in Section 8.6, the latter was mostly due to an unlucky operating point for the VLP.

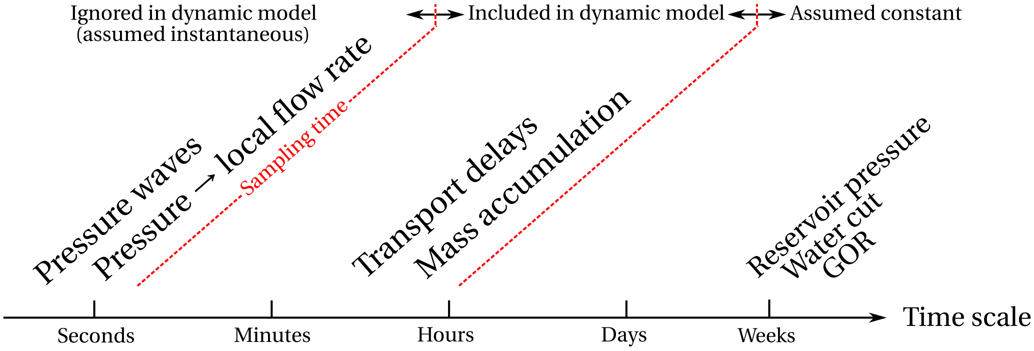


Figure 9.1: Time scales of dynamics in subsea production system.

From this, we can conclude that the steady-state well and choke models are fully capable of estimating transient flow *locally*, at least for the conditions encountered in the OLGA simulations. Although the production system has relatively slow transient behaviour with settling times ranging from a few minutes to one hour, these transients are mostly associated with transport delays and mass accumulation in the wells and pipeline. The instantaneous flow rate at a given point in the network was seen to depend mostly on the surrounding pressures. This indicated that the time constant of a local pressure-flow rate system was within, or at least not much larger than the sampling time of the system, and we could make the simplifying assumption that the local pressure-rate dynamics could be ignored (see Fig. 9.1).

This assumption directly influenced the approach taken for dynamic modelling; we assumed that the instantaneous flow rates into and out of the wells were given by the well models (IPR/WPC) and choke model, respectively. Similarly, we assumed that the pipeline inflow was given by the total outflow of the wells, while the outflow was determined by the turret choke model. As the same pressure/flow models were used for both the static and dynamic estimators, estimation accuracy was seen to depend mostly on each estimator's robustness properties with respect to calibrated ranges and poor tuning choices, as we will discuss in the next section.

9.2.2 Robustness properties

Although (nominal) estimation accuracy is obviously important, we must also consider other factors. As mentioned above, we need the estimator to be robust in the sense that it the estimation accuracy is not too fragile with respect to e.g. calibration ranges and poor tuning choices. This is particularly important if the estimator is to be used in a feedback loop for automatic rate control, which we have defined as our ultimate goal.

The riser slugging case (Fig. 8.10) gave a good indication of the robustness properties of the estimators. Although the dynamic estimator performed better in terms of predicting oscillating flow rates towards the end of the simulation, the flowline outflow estimate was way off for large parts of the simulation. The most likely cause for this is the turret choke model. The model was calibrated to a single-rate flow test where the turret choke was 30 % open. In the riser slugging case, the turret choke was ramped up to 70 %, which means it was quite far from its calibrated operating point. For the static estimator this had limited consequences, due to the rigid mass balance constraints. The mass balance constraints ensure that the pipeline outflow is calculated as a compromise between *all* the models. However, the dynamic estimator depends on the turret choke model and pipeline pressure drop model *only* to provide good estimates of pipeline outflow. In the riser slugging case, the system moved to an operating point in which the turret choke model was poor, which caused poor outflow estimates. Eventually, the large net outflow emptied the pipeline of gas, which forced the outflow to equal the inflow. Although this can be seen as a kind of "safety net", it is hardly redeeming. Still, it should be noted that the slugging case is very difficult since it requires the models to be valid over a large operational envelope, which is practically impossible to obtain through single-rate testing.

The lesson learned here is that robustness is increased by allowing the flow rates to be calculated as compromises between several models. In this sense, the static estimator can be seen as the "ultimate" configuration; in a sense, all the models are used to calculate all the rates, as models are rigidly connected by means of mass balance constraints. An important remark here is that all the models are formulated in terms of *liquid* rates, which means the estimator is robust with respect to calculating liquid rates. The quality of the oil, gas and water rate estimates depends on whether or our assumption of fixed GORs and water cuts is reasonable or not. This was not mentioned at all in Chapter 8, but as seen in the figures in Appendix D, the oil, gas and water rate estimates are in general very good, indicating that our assumption holds. We could, of course, estimate oil, gas and water rates separately due to the dense instrumentation of the Skarv field. This would enable us to *estimate* the GORs and water cuts and thus provide a means to detect gas and water breakthrough. However, a consequence of this would be that fewer models would be used to predict each flow rate, and we would expect the resulting estimates to deteriorate more as the system moves away from its calibrated operating point.

On the whole, the static estimator is clearly more robust than the dynamic estimator. Although not clear from the simulations in Chapter 8, the static estimator was easier to tune, and less fragile to poor tuning, in addition to being more accurate away from the calibration point. In contrast, the dynamic estimator was difficult to tune; for both simulation cases and the field data case, quite a few attempts were made before the performance was acceptable.

9.2.3 Model complexity and solution times

Aside from estimation accuracy and robustness, model complexity is also an important issue. A less complex model is easier to understand, easier to maintain, and less computationally demanding to solve. Obviously, the static estimator is the clear winner here. This is reflected in the solution times presented in Section 8.5, where we saw that the static flow estimation problem was typically solved in less than one second, while the dynamic estimation problem was typically solved in about ten seconds, depending on the estimation horizon. The longer solution time is hardly surprising when we look at Problem 2.2 (dynamic) compared to Problem 2.1 (static), but it is still very good if we keep in mind that we are solving a dynamic flow estimation case with advanced multiphase models.

In terms of closed-loop control, we need to ensure that the estimates are provided in a reasonable amount of time, since the solution time directly contributes to the time delay in the feedback loop. This time delay is (from the feedback controller point of view) no different than dead-time in the system, and imposes strict limitations on the achievable feedback performance (Skogestad & Postlethwaite, 2005). The solution times for the estimators in this thesis were in general more than fast enough, although the dynamic estimator in particular struggled from time to time. On the whole, both methods are feasible for real-time closed-loop control in terms of solution times.

Based on the sections above, we can conclude that the static estimator has in general performed better than the dynamic estimator. While the two estimators were comparable in terms of estimation accuracy in most cases, the static estimator performed better when we moved away from the calibration point, in addition to being simpler and providing estimates faster.

9.3 The importance of model calibration

It is clear that the estimates will be poor unless the system model is calibrated. In this thesis, we have not investigated in depth how poor calibration will affect the estimates, but rather made an effort to ensure that models are calibrated to the relevant operational conditions. We did, however, see in the very first test case (Fig. 8.3) that estimation accuracy for each individual model was degraded when we moved from one operating condition to another. While the models were able to accurately estimate the flow at the start of the time series (i.e. the calibration point), the estimates were slightly offset after the choke valves were repositioned and the flow rates changed. However, when all the models were used together in the static estimator (Fig. 8.4), this effect was largely suppressed. This can be explained by the fact that some models overestimate the flow rate, while others may underestimate it. Thus, the "composite" flow rate estimated by all the models at once is closer to the measured value. When the dynamic estimator was applied to the same case (Fig. 8.6), fewer models were used to estimate each flow rate, and some of the offset returned.

An extreme case was seen in the riser slugging case where the turret choke was repositioned from 30 % open to 70 % open. When we applied the dynamic estimator to this case (Fig. 8.10), this caused the turret choke model to significantly overestimate the flow rate as the system moved far away from the calibration point. Although this was not a realistic case (in practice, slugging is not provoked on purpose), it did illustrate what happens when we move far away from the operating point at which a model is calibrated, and as we already have discussed, it also illustrates how the static estimator is more robust with respect to operational changes.

Judging from both intuition and industry experience (e.g. (Heddle et al., 2012; Goh et al., 2007)), we would probably see better results if we had calibrated the models using multi-rate well tests, using e.g. the methods described in Chapter 6. Another idea may be to consider the B-spline as an interpolation method for data-driven calibration methods. With its flexibility and customizability with respect to interpolation and/or smoothing, its seems a natural choice for connecting more or less trusted measurement data from well tests.

9.4 Additional remarks

In this section, we will gather some additional remarks which are not directly related to flow estimation, but important nonetheless.

9.4.1 On transparent modelling

Both the static and dynamic estimators differ from "standard" data reconciliation techniques in the sense that both model errors and measurement errors are included in the objective function. This is possible since the model has been decomposed into simple network components as opposed to using a black-box model. The advantages of this formulation is that we can transfer our confidence in each individual model to the weights, or even better, measure the quality of each model through some form of calibration and use the results to give "good" models more weight than "bad" models. In addition, if we know the qualitative behaviour of each model, we can use the weights to customize the behaviour of the estimator. For example, in the well adjustment case, we saw that we could obtain good inflow estimates by giving the IPR and WPC models large weights, while large weights on choke models gave good outflow estimates. Like the black-box approach, this formulation also allows a lot of flexibility with respect to our trust in the pressure measurements. If we are confident the pressure measurements are correct, we can configure $\mathbf{R}^{-1}$ with large weights compared to $\mathbf{Q}^{-1}$, which will result in a solution with relatively large model errors ($\mathbf{w}$). By inspecting these model errors, we can identify models which "disagree" with the other models, which might indicate e.g. poor calibration or the need for recalibration (i.e. a new well test). On the other hand, if we suspect one or more pressure measurements are drifting or noisy, we can account for this by reducing the appropriate weights.

Another advantage of the transparent modelling approach is the easy access to the variables - this enables us to easily add constraints in the optimization problem without affecting the solution times significantly (unless the constraint is particularly complicated).

9.4.2 On mass and holdup estimation

The dynamic model and associated MHE formulation was described in Chapter 5. It was largely based on the static model; dynamics were introduced by relaxing certain mass balance constraints to allow the well and pipeline volumes to accumulate mass. This approach is both simple and intuitive, and we saw that it was able to provide reasonable predictions of holdup (Fig. 8.7) and liquid mass dynamics (Fig. D.7), albeit with some offset.

However, the dynamic model was seen to have some flaws. We saw both in the well adjustment case and the riser slugging case that the holdup estimates had some offset compared to the measurements from OLGA. Since the PVT calculations of the estimator are the roughly same as the ones in OLGA (due to the B-spline approximations), it is reasonable to assume that most of the error is due to the simplification made with regard to average pressure and temperature. While the relevant densities needed for the holdup calculation is based on an approximated average pressure in the estimator, OLGA uses a spatial discretization and keeps track of individual holdups in a large number of pipe segments. Recall the hydrostatic pressure drop in a well or pipeline was given as $\Delta p_{\text{hs}} = \overline{\rho}_{tp} g h$. The only unknown here is the two-phase density $\overline{\rho}_{tp}$, since the gravitational acceleration g and elevation change h are known constants. An expression for the *actual* two-phase density would involve an integral along the entire pipe length (of length L), i.e.

$$\overline{\rho}_{tp} = \int_0^L H_l \rho_l + (1 - H_l)\rho_g \, \mathrm{d}x, \tag{9.1}$$

where the holdup $H_l = H_l(x)$, the liquid density $\rho_l = \rho_l(p(x), T(x))$ and the gas density $\rho_g = \rho_g(p(x), T(x))$ are all functions of the spatial variable x. By replacing the rough approximations (5.52) and (5.53) with a better approximation of (9.1), we would probably obtain a better holdup estimate. However, this would require a spatial discretization which would increase the number of variables in the optimization problem and make it more difficult and time-consuming to solve. In addition, we would need sufficiently accurate pressure and temperature profiles along the pipeline.

Another problem with the dynamic formulation was seen clearly in the riser slugging case. From Figure D.12 we see that the gas mass in the pipeline is depleted as the outflow is much higher than the inflow. At the same time, the pipeline GOR remains at about 280, which clearly does not make physical sense (zero gas should give a GOR of zero). Tuning the pressure drop model weights proved to be difficult; insufficient weighting resulted in accumulation/depletion of mass (as seen in the aforementioned example), while too much weighting gave oscillating behaviour.

The model formulation does not ensure that the inflow of a well or pipeline equals the outflow at steady-state. In other words, the pressure drop models must be weighted sufficiently to prevent them from accumulating/emptying large amounts of mass.

9.4.3 The estimation horizon

In the dynamic estimator, the estimation horizon N determines how far the estimator is able to see into the past. For the algebraic constraints (IPR, WPC, choke etc.) this is not significant, since the estimated flow rate at time index k depends only on the pressures and temperatures at time index k. However, the pressure drop models require a sufficiently large estimation horizon to function properly. This is clear from intuition; in order to increase or decrease the hydrostatic pressure drop, the net inflow to the well or pipeline must be nonzero *over time*. Hence, we must allow the pressure drop model to detect that it has control over the pressure drop by means of controlling the net inflow. For the well adjustment case and the riser slugging case, we had the turret choke model to assist the pressure drop model in predicting flowline outflow. In these cases the estimation horizon was set to $N = 5$, and the pressure drop model was primarily used for mass and holdup estimation. However, in the field data case, the turret choke model was not good enough to be used, and the outflow was estimated by using the pressure model only. In this case, we needed a longer estimation horizon ($N = 20$) to obtain a reasonable result.

9.4.4 On velocity estimation and control

We have not considered velocity estimation in this thesis. However, velocity estimation would definitely be useful for fields like Skarv, where erosion caused by high gas velocity is an operational concern. To determine the gas velocity at a given point in the flow network, we would require (1) the *in-situ* volumetric flow rate of gas, and (2) the cross-sectional area occupied by gas at the given location, which depends on the liquid holdup. In other words, we would require in-depth knowledge of the *in-situ* conditions at the point in which we wanted to calculate the velocity. Usually, the most interesting point is the point with the largest gas velocity, which is typically the exit point (separator) of the network, where the pressure is low and a lot of gas is present. Since we are equipped with both pressure and temperature measurements, and fluid properties in the form of B-spline approximated PVT tables, we may be able to calculate reasonable velocity estimates, which in turn could be used for simple velocity control, e.g. using a PID controller. Note that advanced controllers like MPC would be more difficult, since this would require us to *predict* pressures and temperatures in the network, which the models described in this thesis are not designed to do. In any case, velocity estimation would be a natural and useful extension to the flow estimators developed in this thesis.

9.5 Concluding remarks

To conclude this discussion, we sum up a few key points. For an automatic rate control application, the static estimator described in Chapter 4 has been seen to be a feasible choice for flow estimation. The dynamic estimator from Chapter 5 has some challenges, but could potentially be useful if these challenges were addressed. In any case, good well testing and calibration routines are crucial for estimation accuracy.

Finally, we remind the reader of the potential benefits of flow estimation methods such as the ones described in this thesis. In addition to providing a means to supervise and control flow rates, a decent flow estimation method could be the cornerstone of several useful tools, such as (but not limited to) flow assurance systems, systems for automatic reporting of production rates, and systems for automatic updating of reservoir models. Flow estimation is thus a key part of the Integrated Operations/Field of the Future mindset which is becoming more and more dominant in the petroleum industry.

Chapter 10

Conclusion

In this thesis, we have compared two optimization-based flow estimation methods for subsea production systems; one with static models, and one with dynamic models. The derived models were based on the Tilje two-well template in the BP-operated Skarv field, however they were generalized to represent any cluster of oil wells tied back through a single pipeline. We used OLGA and field data from Tilje to compare the performance of the two methods with automatic rate control in mind.

Based on the results in Chapter 8 and the discussion in Chapter 9, the main conclusion of this thesis is that *the estimator with static models is best suited for automatic rate control*. While both methods were able to provide good flow rate estimates in most cases, the static estimator was superior with respect to steady-state accuracy, robustness and solution times, which are all important factors for any estimator to be used for feedback control. In addition, the static estimator was seen to perform surprisingly well with respect to estimating flow rate *transients*, as the flow rate dynamics were captured in the available pressure measurements. Although not investigated in depth, it has also been clear that none of the methods will work well unless the models are properly calibrated against well test data. The relatively simple nonlinear least-squares method used to calibrate the models in this thesis was sufficient in most cases, however, since the models were calibrated to a single-rate test, model accuracy deteriorated somewhat as the operational conditions changed.

The extra effort put into the dynamic model did not pay off in terms of improving flow rate estimates. However, it did provide estimates of important variables like liquid holdup, which could be useful for e.g. predicting slug flow. When applied to a slugging case, it did not perform particularly well in terms of estimating flow rates, but it *did* clearly indicate slug flow, even though the system was operating far from the point at which the model was calibrated. Rather than writing the method off completely, the author believes that some clever modifications could potentially make the dynamic model suitable for virtual flow metering applications.

To the author's knowledge, the methods applied in this thesis differ from traditional virtual flow metering systems in two respects. Firstly, the optimization problems were formulated with transparent models rather than black-box models, which enabled easier implementation of model constraints, more intuitive tuning and easier incorporation of information about model uncertainty. Secondly, to achieve the aforementioned, the black-box models were approximated by B-spline functions. Consequently, the solver could work *directly* with well, choke and pipeline models which all provided gradients, leading to fast solution times, i.e. less than one second for the static estimator and ten seconds for the dynamic estimator, which is clearly within the limits required for real-time feedback control for the system in question. The use of B-splines was motivated by promising results in production optimization (Sandnes, 2013; Grimstad et al., 2014), and they were seen to work equally well for the flow estimation problem. The usefulness of the B-spline was further illustrated by approximation of PVT tables, which eliminated the need for equation-intensive compositional models in the optimization problem.

10.1 Summary

In this short summary, we will list the (in the author's opinion) most important contributions of this thesis.

- We have compared two optimization-based flow estimation methods, and shown that the method with static models is best suited for automatic rate control.

- We have shown that the methods used in this thesis (particularly the static estimator) are relatively simple and robust ways of estimating flow rates in subsea production systems, in the sense that a simple yet functional prototype was successfully developed and tested on field data in less than six months.

- We have extended the use of B-splines from the production optimization problem to the flow estimation problem, and seen that this leads to fast solution times and easy integration of PVT tables.

- We have seen that breaking up the black-box model and introducing model error variables allows for easier tuning and incorporation of model uncertainty.

A conference paper based on this work is currently underway, and is intended for presentation at the 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production on May 27-29, 2015 in Florianópolis, Brazil.

10.2 Further work

To conclude, we present some ideas on how the estimators presented in this thesis could be improved. Although the list is rather long, it could probably be longer; the items below can be viewed as necessary steps to be taken before our methods could be considered fully fledged virtual flow metering systems.

- Although we have presented some ideas on how multi-rate tests could be used to quantify model uncertainty and incorporate this in the flow estimation problem, we have not tested them in practice. It would be interesting to both see the ideas implemented in practice, and to see a more rigorous treatment of model uncertainty and weighting strategies.
- As mentioned in the discussion, the dynamic model has some flaws which should be addressed to improve robustness and performance.
- The models derived in this thesis are restricted to a rather specific network structure. In order to apply the method to other types of flow networks, the methods should be further generalized. Important considerations here would be how the methods scale with different network structures and how this affects solution times.
- We have restricted our investigation to oil wells; consequently, single-phase gas wells cannot be handled without some additional measures. Thus, one idea for further work may be to modify or extend the models to support this possibility.
- We have assumed fixed GORs and water cuts, and thus estimated oil, gas and water rates based on liquid rate estimates. It would be interesting to see how an approach which estimates oil, gas and water rates separately (and thus let the GORs and waters cuts be free variables in the optimization problem) compares to the estimators in this thesis.
- Investigate the B-spline more closely. Preprocess data to ascertain where tight sampling is needed (in an automatic way based on Hessians). Look more closely into model sensitivity to see which variables can be dropped in order to reduce the dimension.
- For fields like Skarv where MPFMs are installed, we could potentially include some of the flow rates in the objective function (i.e. a flow error term). If our formulation included several templates tied back to a common separator system with fiscal metering for each phase, we could also reconcile the total flow rate from all the templates against the fiscal measurements.
- Other improvements may include e.g. improved choke modelling, velocity estimation strategies, well shut-in modelling, inclusion of injection wells, etc.

References

Alessandri, A., Baglietto, M., & Battistelli, G. (2008). Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44(7), 1753–1765.

Bakken, A., Grimstad, B., & Larsen, M. (2011). Life of field tool for optimal subsea design, condition monitoring, virtual metering, and flow assurance advice using a common field model. In *Proceedings of Subsea Controls DownUnder conference, held in Perth, Australia, 17.-19. October 2011.*

Beggs, D. & Brill, J. (1973). A study of two-phase flow in inclined pipes. *Journal of Petroleum Technology*, 25(5), 607–617.

Beggs, H. D. (2003). *Production Optimization using NODAL analysis.* OGCI and Petroskills Publications, second edition.

Bendiksen, K. H., Maines, D., Moe, R., & Nuland, S. (1991). The dynamic two-fluid model OLGA. Theory and application. *Society of Petroleum Engineers*, 6(2), 171–180.

Bieker, H. P., Slupphaug, O., & Johansen, T. A. (2007). Real-time production optimization of oil and gas production systems: A technology survey. *SPE Production & Operations*, 22(4), 382–391.

Binder, B. J. T. (2012). Production optimization in a cluster of gas-lift wells. Master's thesis, Norwegian University of Science and Technology.

BP (2014). The Skarv field. Web: http://www.bp.com/no_no/norge/om-bp-norge/hva-vi-gjoer/bp-opererte-felt/skarv.html. Accessed 07.06.2014.

Bringedal, B., Storkaas, E., Dalsmo, M., Aarset, M., & With, H. M. (2010). Recent developments in control and monitoring of remote subsea fields. In *SPE Intelligent Energy Conference and Exhibition held in Utrecht, The Netherlands, 23–25 March 2010.*

Brown, R. G. & Hwang, P. Y. C. (2012). *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc., 4 edition.

Çengel, Y. A. & Cimbala, J. M. (2010). *Fluid Mechanics Fundamentals and Applications.* McGraw Hill Education.

Chen, C.-T. (1995). *Linear System Theory and Design.* Oxford University Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms.* MIT Press.

Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control*, 9(1), 5–12.

Cybernetica (2014). Cybernetica CENIT booklet. Web: http://www.cybernetica.no/v3/products/CENIT/CENIT.pdf Accessed 03.03.2014.

Demneh, F. A. & Mesbah, A. (2008). The effect of kinetic energy change on flow in gas pipelines. *Hydrocarbon processing.*

Dempf, D. & List, T. (1998). On-line data reconcilation in chemical plants. *Computers & Chemical Engineering*, 22(1), 1023–1025.

Duns, H. & Ros, N. (1963). Vertical flow of gas and liquid mixtures in wells. In *Proceedings of 6th World Petroleum Congress, 19-26 June, Frankfurt am Main, Germany.*

Egeland, O. & Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control.* Marine Cybernetics AS.

Foss, B. (2012). Process control in conventional oil and gas field - challenges and opportunities. *Control Engineering Practice*, 20(10), 1058–1064.

Goh, K.-C., Moncura, C. E., Overschee, P. V., & Briers, J. (2007). Production Surveillance and Optimization with Data Driven Models. In *Proceedings of the International Petroleum Technology Conference, 4-6 December 2007, Dubai, U.A.E.*

Grimstad, B., Foss, B., Heddle, R., & Woodman, M. (2014). A framework for global production optimization of multiphase networks - theory and application on a real subsea field case. Manuscript draft.

Grimstad, B. & Sandnes, A. (2014). Global optimization with spline constraints. Submitted paper.

Guennebaud, G., Jacob, B., et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Gunnerud, V. (2011). *On decomposition and piecewise linearization in petroleum production optimization.* PhD thesis, Norwegian University of Science and Technology.

Gunnerud, V. & Foss, B. (2009). Oil production optimization - a piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, 34(11), 1803 – 1812.

Haseltine, E. L. & Rawlings, J. B. (2005). Critical evaluation of Extended Kalman Filtering and Moving-Horizon Estimation. *Industrial & Engineering Chemistry Research*, 44(8), 2451–2460.

Hauge, J. & Horn, T. (2005). The Challenge of Operating and Maintaining 115 Subsea Wells on the Troll Field. In *Proceedings of The Offshore Technology Conference held in Houston, TX, USA, 2-5 May 2005.*

Heddle, R., Foot, J., & Rees, H. (2012). ISIS Rate&Phase - Delivering Virtual Flow Metering for 300 Wells in 20 Fields. In *Proceedings of he SPE Intelligent Energy International held in Utrecht, The Netherlands, 27–29 March 2012.*

Hermann, R. & Krener, A. J. (1977). Nonlinear controllability and observability. *IEE Transactions on Automatic Control*, 22(5), 728–740.

Holmås, K. & Løvli, A. (2011). FlowManager^{TM} Dynamic: A multiphase flow simulator for online surveillance, optimization and prediction of subsea oil and gas production. *BHR Group.*

Huang, Y.-F., Werner, S., Huang, J., Kashyap, N., & Gupta, V. (2012). State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. *Signal Processing Magazine, IEEE*, 29(5), 33–43.

Imsland, L., Kittilsen, P., & Schei, T. S. (2010). Model-based optimizing control and estimation using Modelica models. *Modeling, Identification and Control*, 31(3), 107–121.

Jahanshahi, E. (2013). *Control Solutions for multiphase flow.* PhD thesis, Norwegian University of Science and Technology.

Johansen, T. A. (2011). *Selected Topics on Constrained and Nonlinear Control*, chapter Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation, (pp. 1–53). STU/NTNU.

Julier, S. J. & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3 (pp. 3–2).

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35–45.

Kandepu, R., Foss, B., & Imsland, L. (2008). Applying the unscented Kalman filter for nonlinear state estimation. *Journal of Process Control*, 18(7-8), 753–768.

Kopp, R. E. & Orford, R. J. (1963). Linear regression applied to system identification for adaptive control systems. *Aiaa Journal*, 1(10), 2300–2306.

Kosmidis, V. D., Perkins, J. D., & Pistikopoulos, E. N. (2005). A mixed integer optimization formulation for the well scheduling problem on petroleum fields. *Computers & Chemical Engineering*, 29(7), 1523–1541.

Larsen, E. & Hocking, P. (2012). Skarv and Valhall Re-development - A Journey to the Second Generation of Digital Oilfields. In *Proceedings of the SPE Intelligent Energy International, held in Utrecht, the Netherlands, 27-29 March 2012.*

Lerma, P. G., Simonton, C., & Wadle, T. (2006). Allocation Process Modeling for Deep Water Production. In *Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference, held in Abu Dhabi, U.A.E., 5-8 November 2006.*

Maquin, D., Adrot, O., & Ragot, J. (2000). Data reconcilation with uncertain models. *ISA Transactions*, 39(1), 35–45.

Mayinger, F. & Kiederle, G. (1993). Pressure loss in valves during horizontal two-phase flow. In *Proceedings of the National Heat Transfer Conference, Atlanta* (pp. 101–107).

Melbø, H., Morud, S. A., Bringedal, B., van der Geest, R., & Stenersen, K. (2003). Software that enables flow metering of well rates with long tiebacks and with limited or inaccurate instrumentation. In *Proceedings of the Offshore Technology Conference held in Houston, TX, USA, 5-8 May, 2003.*

Moriari, M. & Lee, J. H. (1999). Model predictive control: Past, present and future. *Computers & Chemical Engineering*, 23(4-5), 667–682.

Nalum, K. (2013). Modeling and dynamic optimization in oil production. Master's thesis, Norwegian University of Science and Technology.

Narasimhan, S. & Jordache, C. (1999). *Data Reconciliation & Gross Error Detection - An intelligent use of process data.* Gulf Proffesional Publishing.

Ptil (2009). Audit of fabrication of flexible risers for the Skarv field. Web: http://www.ptil.no/news/audit-of-fabrication-of-flexible-risers-for-the-skarv-field-article5773-878.html. Accessed 18.02.2014.

Qin, S. J. & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Rao, C. V., Rawlings, J. B., & Lee, J. H. (2001). Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10), 1619–1628.

Rawlings, J. B. & Mayne, D. Q. (2013). Model Predictive Control: Theory and Design. Nob Hill Publishing. Electronic book available for download from http://jbrwww.che.wisc.edu/. Accessed 26.02.2014.

Robertson, D. G., Lee, J. H., & Rawlings, J. B. (1996). A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8), 2209–2224.

Robertson, P. (2013). B-spline approximations in an optimization framework. Project assignment, Norwegian University of Science and Technology.

Sandnes, A. (2013). Solving a network flow decision problem with sampled nonlinearities. Master's thesis, Norwegian University of Science and Technology.

Schüller, R., Solbakken, T., & Selmer-Olsen, S. (2003). Evaluation of Multiphase Flow Rate Models for Chokes Under Subcritical Oil/Gas/Water Flow Conditions. *SPE Production & Facilities*, 18(3), 170–181.

Siahaan, H. B., Aamo, O. M., & Foss, B. A. (2005). Suppressing riser-based slugging in multiphase flow by state feedback. In *Proceedings of 44th IEEE Conference on decision and control, held in Seville, Spain, December 12.-15. 2005.*, volume 44 (pp. 452).

Simon, D. & Chia, T. L. (2002). Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1), 128–136.

Skogestad, S. & Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. Wiley, 2nd edition.

SPE (1982). The SI Metric System of Units and SPE Metric Standard.

SPT Group (2013). *OLGA 7 User Manual*.

Sui, D., Nybø, R., Gola, G., Roverso, D., & Hoffmann, M. (2011). Ensemble methods for process monitoring in oil and gas industry operations. *Journal of Natural Gas Science and Engineering*, 3(6), 748 – 753. Artificial Intelligence and Data Mining.

Vogel, J. V. (1968). Inflow performance relationships for solution-gas drive wells. *Journal of Petroleum Technology*, 20(01), 83–92.

Wächter, A. & Biegler, L. T. (2006). On the implenentation of an interior-point filter line-seach algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Appendix A

Function approximation with B-splines

This appendix is a highly truncated introduction to B-splines, based on material from my project assignment last semester (Robertson, 2013). We will limit this discussion to univariate functions, but the methods described here generalize to any dimension. B-spline functions are representations of piecewise polynomials. A univariate B-spline function b of degree p is given as a linear combination of n B-spline basis functions:

$$b : D \to \mathbb{R}, \quad x \mapsto \sum_{i=1}^{n} c_i B_{i,p}(x). \tag{A.1}$$

Here, $D \subset \mathbb{R}$ is the function domain, which is assumed to be a closed interval defined by upper and lower bounds on x, i.e. $D = [\underline{x}, \overline{x}]$ where $\underline{x}$ is the lower bound on x and $\overline{x}$ is the upper bound on x. $\{B_{i,p}\}_{i=1}^{n}$ are piecewise polynomial *B-spline basis functions* of degree p, and $\{c_i\}_{i=1}^{n}$ is a set of real-valued coefficients. For a given degree, the B-spline basis functions are determined entirely from a nondecreasing sequence of real numbers known as the *knot sequence*. To generate n basis functions, a knot sequence of length $n+p+1$ is needed, so we define the knot sequence as

$$T = \{t_j\}_{j=1}^{n+p+1} \tag{A.2}$$

Given a polynomial degree p and a knot sequence T, the resulting basis functions can be shown to be linearly independent, and they span a linear space consisting of piecewise polynomial functions defined on D. A basis function of degree p is defined by $p+2$ adjacent knots, and can be written out as an explicit function of these knots (and x):

$$B_{i,p}(x) = f(x, t_i, t_{i+1}, \ldots, t_{i+p+1}). \tag{A.3}$$

(A.3) is given by the *Cox-de Boor recursion*:

$$B_{i,p}(x) = \frac{x - t_i}{t_{i+p} - t_i} B_{i,p-1}(x) + \frac{t_{i+p+1} - x}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1}(x) \tag{A.4a}$$

$$B_{i,0}(x) = \begin{cases} 1, & x \in [t_i, t_{i+1}), \\ 0, & \text{otherwise.} \end{cases} \tag{A.4b}$$

As seen from (A.4), a B-spline of degree p is constructed from two overlapping B-splines of degree $p-1$. Its shape depends on the distribution of the knots, but in general its shape resembles a bell. Outside the interval $x \in [t_i, t_{i+p+1}]$, $B_{i,p}(x)$ is identically zero.

To approximate a function $y = f(x)$ represented by a set of m data points $\{x_i, y_i\}_{i=1}^{m}$, we first define that our approximation $\hat{f}$ of f should be a B-spline function, i.e.

$$\hat{f}(x) = \sum_{i=1}^{n} c_i B_i(x). \tag{A.5}$$

Function approximation with B-splines amounts to selecting the c_i's which satisfy a given set of conditions. This could be e.g. interpolating all the points in the data set, or other variants which may include smoothing of data or other properties. The simplest case is interpolation, so we will give a short introduction here; to interpolate all the data points, we require the following:

$$y_i = \hat{f}(x_i) = \sum_{j=1}^{n} c_j B_j(x_i), \quad \forall\ i \in \{1, \ldots, m\}. \tag{A.6}$$

This is a set of linear equations, which can be written as

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} B_1(x_1) & B_2(x_1) & \ldots & B_n(x_1) \\ B_1(x_2) & B_2(x_2) & \ldots & B_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_1(x_m) & B_2(x_m) & \ldots & B_n(x_m) \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}}_{\mathbf{c}}, \tag{A.7}$$

or $\mathbf{y} = \mathbf{Bc}$ where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{B} \in \mathscr{M}^{m\times n}$. For given choices of the knot sequence, e.g. the *free* knot sequence

$$T_F = \{ \underbrace{x_1, \ldots, x_1}_{p+1 \text{ repetitions}}, x_3, \ldots, x_{m-2}, \underbrace{x_m, \ldots, x_m}_{p+1 \text{ repetitions}} \} \tag{A.8}$$

we have $m = n$ and we can calculate the coefficients as $\mathbf{c} = \mathbf{B}^{-1}\mathbf{y}$. The resulting B-spline function will now interpolate all the data points. As mentioned above, other methods for calculating the coefficients exist, which may include smoothing, constraints on the properties of the resulting B-spline function and/or its derivatives, etc. (see (Robertson, 2013)). In this thesis, we have used both interpolating B-splines (well models, pipeline VLP and choke models) and a variant of smoothing B-splines known as the P-spline (PVT tables). The calculation of the coefficients is done within the CENSO framework using the sparse LU linear system solver from the EIGEN library (Guennebaud et al., 2010).

Appendix B

Extended pressure model

In this appendix, we will present an extension to the dynamic model presented in Chapter 5, which accounts for frictional pressure losses in the well and pipeline volumes. This model *was* in fact implemented, but the additional variables and constraints slowed down solution times without improving estimates. This was due to that the calculated frictional pressure losses were very small, which was consistent with the OLGA simulations of the Tilje template. First, we present the additional variables necessary for the model extension:

Symbol	Definition	Description	Variables	Unit
$\overline{\boldsymbol{\mu}}_o$	$[\overline{\mu}_o^0, \overline{\mu}_o^1, \dots, \overline{\mu}_o^{n_w-1}, \overline{\mu}_o^p]^\top$	Average oil viscosities	$N \cdot (n_w + 1)$	$[\mathrm{N/sm}^2]$
$\overline{\boldsymbol{\mu}}_w$	$[\overline{\mu}_w^0, \overline{\mu}_w^1, \dots, \overline{\mu}_w^{n_w-1}, \overline{\mu}_w^p]^\top$	Average water viscosities	$N \cdot (n_w + 1)$	$[\mathrm{N/sm}^2]$
$\overline{\boldsymbol{\mu}}_l$	$[\overline{\mu}_l^0, \overline{\mu}_l^1, \dots, \overline{\mu}_l^{n_w-1}, \overline{\mu}_l^p]^\top$	Average liquid viscosities	$N \cdot (n_w + 1)$	$[\mathrm{N/sm}^2]$
$\overline{\mathbf{w}}_l$	$[\overline{w}_l^0, \overline{w}_l^1, \dots, \overline{w}_l^{n_w-1}, \overline{w}_l^p]^\top$	Average liquid mass flow rates	$N \cdot (n_w + 1)$	[kg/s]
$\overline{\mathbf{q}}_l$	$[\overline{q}_l^0, \overline{q}_l^1, \dots, \overline{q}_l^{n_w-1}, \overline{q}_l^p]^\top$	Average in-situ liquid volumetric flow rates	$N \cdot (n_w + 1)$	$[\mathrm{m}^3/\mathrm{s}]$
$\overline{\mathbf{v}}_{sl}$	$[\overline{v}_{sl}^0, \overline{v}_{sl}^1, \dots, \overline{v}_{sl}^{n_w-1}, \overline{v}_{sl}^p]^\top$	Average in-situ liquid superficial velocities	$N \cdot (n_w + 1)$	[m/s]
$\mathbf{N}_{\mathrm{Re}}$	$[N_{\mathrm{Re}}^0, N_{\mathrm{Re}}^1, \dots, N_{\mathrm{Re}}^{n_w-1}, N_{\mathrm{Re}}^p]^\top$	Reynolds numbers	$N \cdot (n_w + 1)$	[-]
$\boldsymbol{\lambda}_f$	$[\lambda_f^0, \lambda_f^1, \dots, \lambda_f^{n_w-1}, \lambda_f^p]^\top$	Darcy friction factors	$N \cdot (n_w + 1)$	[-]
$\Delta\mathbf{p}_f$	$[\Delta p_f^0, \Delta p_f^1, \dots, \Delta p_f^{n_w-1}, \Delta p_f^p]^\top$	Frictional pressure losses	$N \cdot (n_w + 1)$	[bar]

Table B.1: Variable overview, extended pressure model w/friction (remaining variables are explained in Tables 4.3 and 5.1).

This gives $9N(n_w + 1)$ additional variables, so the total number of variables is now $n_{\mathbf{x}} = 9N(n_w + 1) + N(42n_w + 39) = N(51n_w + 48)$.

Recall from Section 3.5 that the frictional pressure gradient in a pipeline is given as $\mathrm{d}p_{\mathrm{fr}}/\mathrm{d}L = \lambda_m \rho_m v_m^2 / 2gD$. To keep the model simple, we will base the friction factor λ_m, the density ρ_m and the velocity v_m on *liquid* properties. This assumes that the flow regime is dominated by liquid. Moreover, we use averaged values, i.e. based on the fluid properties evaluated at average pressures and temperatures in each volume, much like the hydrostatic pressure model in Section 5.3.3. First, we evaluate the PVT table to obtain the average viscosities of oil and water:

$$\text{Wells:} \qquad \overline{\mu}^i_{o,k} = \Phi_{\mu_o}(\overline{p}^i_k, \overline{T}^i_k), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.1}$$

$$\overline{\mu}^i_{w,k} = \Phi_{\mu_w}(\overline{p}^i_k, \overline{T}^i_k), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.2}$$

$$\text{Pipeline:} \qquad \overline{\mu}^p_{o,k} = \Phi_{\mu_o}(\overline{p}^p_k, \overline{T}^p_k), \qquad \forall\ k \in \mathscr{H}, \tag{B.3}$$

$$\overline{\mu}^p_{w,k} = \Phi_{\mu_w}(\overline{p}^p_k, \overline{T}^p_k), \qquad \forall\ k \in \mathscr{H}. \tag{B.4}$$

The average liquid viscosities are obtained by using the water cuts in a similar way to the average densities defined in (5.46):

$$\overline{\mu}^i_{l,k} = r^i_{wc}\overline{\mu}^i_{w,k} + (1 - r^i_{wc})\overline{\mu}^i_{o,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.5}$$

$$\overline{\mu}^p_{l,k} = r^p_{wc,k}\overline{\mu}^p_{w,k} + (1 - r^p_{wc,k})\overline{\mu}^p_{o,k}, \qquad \forall\ k \in \mathscr{H}. \tag{B.6}$$

We now calculate the average liquid mass flow rates using a simple average of inflow and outflow. Standard densities are used to convert from standard volumetric flow to mass flow, like we did in the differential mass balance constraints (Section 5.2.1);

$$\overline{w}^i_{l,k} = \frac{1}{2} \cdot \frac{\rho_l^{\text{std}}}{3600}(q^i_{l,k} + q^{n_w+i}_{l,k}), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.7}$$

$$\overline{w}^p_{l,k} = \frac{1}{2} \cdot \frac{\rho_l^{\text{std}}}{3600}(q^{3n_w}_{l,k} + q^{3n_w+1}_{l,k}), \qquad \forall\ k \in \mathscr{H}. \tag{B.8}$$

The average in-situ liquid volumetric flow rates are calculated from the mass flow rates and the (in-situ) liquid densities:

$$\overline{\rho}^i_{l,k}\overline{q}^i_{l,k} = \overline{w}^i_{l,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.9}$$

$$\overline{\rho}^p_{l,k}\overline{q}^p_{l,k} = \overline{w}^p_{l,k}, \qquad \forall\ k \in \mathscr{H}. \tag{B.10}$$

The average liquid *superficial velocity* is the velocity which would result from the liquid occupying the entire cross-sectional area of the pipe. This differs from the "normal" velocity, since the actual cross-sectional area of the pipe occupied by liquid is smaller due to the presence of gas. The average liquid superficial velocities are calculated as

$$\overline{v}^i_{sl,k} = \frac{\overline{q}^i_{l,k}}{A^i_p} = \frac{4}{\pi(D^i_p)^2}\overline{q}^i_{l,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.11}$$

$$\overline{v}^p_{sl,k} = \frac{\overline{q}^p_{l,k}}{A^p_p} = \frac{4}{\pi(D^p_p)^2}\overline{q}^p_{l,k}, \qquad \forall\ k \in \mathscr{H}, \tag{B.12}$$

where the A_p's and D_p's are the pipe cross-sectional areas and diameters, respectively. We now go on to calculate the friction factors. The first thing we must to is to calculate the Reynolds numbers. The Reynolds number is a dimensionless number which is a measure of the ratio between interial and viscous forces acting on the fluid (Çengel & Cimbala, 2010). It is perhaps the most used quantity for determining the flow regime for single-phase flow (laminar/transitional/turbulent). We will calculate our Reynolds numbers as

$$\overline{\mu}^i_{l,k} N^i_{\text{Re},k} = D^i_p \overline{\rho}^i_{l,k} \overline{v}^i_{sl,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.13}$$

$$\overline{\mu}^p_{l,k} N^p_{\text{Re},k} = D^p_p \overline{\rho}^p_{l,k} \overline{v}^p_{sl,k}, \qquad \forall\ k \in \mathscr{H}. \tag{B.14}$$

The Darcy friction factor λ_f depends on the Reynolds number N_{Re} of the flow and the pipe relative roughness, ϵ_p / D_p. For low Reynolds numbers ($N_{\text{Re}} < 2300$), the flow is considered laminar, and the friction factor is given by $\lambda_f = 64/N_{\text{Re}}$. For large Reynolds numbers ($N_{\text{Re}} > 4000$), the flow is considered turbulent and the friction factor is governed by the implicit Colebrook equation:

$$\frac{1}{\sqrt{\lambda_f}} = -2.0 \log_{10}\left(\frac{\epsilon_p / D_p}{3.7} + \frac{2.51}{N_{\text{Re}} \sqrt{\lambda_f}} \right). \tag{B.15}$$

Clearly, this is not easy to incorporate as constraints in an optimization problem. When plotting the friction factor as defined above in a log-log plot, we obtain the *Moody chart* shown in Figure B.1.

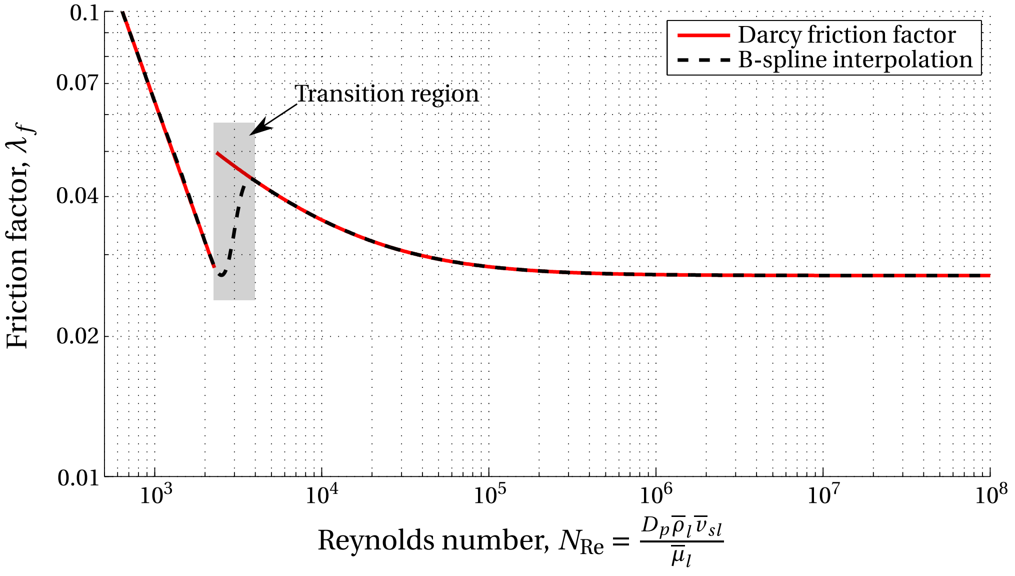


Figure B.1: The Moody chart showing the Darcy friction factor as a function of the Reynolds number, and its B-spline approximation ($\frac{\epsilon_p}{D_p} = 3.29 \cdot 10^{-3}$).

Again, we make use of B-splines and approximate the Moody chart by sampling it for a set of Reynolds numbers and creating an interpolating B-spline Φ_{λ_f} (shown in dashed black in Fig B.1). The relative roughnesses for the wells and the pipeline are obtained from OLGA geometry files. Then, we can add the following constraints to calculate the friction factors:

$$\lambda^i_{f,k} = \Phi_{\lambda_f}(N^i_{\text{Re},k}), \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.16}$$

$$\lambda^p_{f,k} = \Phi_{\lambda_f}(N^p_{\text{Re},k}), \qquad \forall\ k \in \mathscr{H}. \tag{B.17}$$

We now have all the variables required to calculate the frictional pressure drop. We keep it simple and multiply the pressure gradient by the entire pipe length L_p to obtain the total frictional pressure drop. As recommended in (Beggs, 2003), the liquid holdup H_l is included as a multiplier to incorporate knowledge about the actual amount of liquid present in the well or pipe. We also include a 10^{-5} multiplier which is necessary for converting from SI units (Pascals) to our unit of choice (bar).

$$\Delta p^i_{f,k} = \frac{10^{-5} L^i_p}{2D^i_p} H^i_{l,k} \lambda^i_{f,k} \overline{\rho}^i_{l,k} \left(\overline{v}^i_{sl,k}\right)^2, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.18}$$

$$\Delta p^p_{f,k} = \frac{10^{-5} L^p_p}{2D^p_p} H^p_{l,k} \lambda^p_{f,k} \overline{\rho}^p_{l,k} \left(\overline{v}^p_{sl,k}\right)^2, \qquad \forall\ k \in \mathscr{H}. \tag{B.19}$$

Now that we have calculated the frictional pressure drop, we can replace (5.34) and (5.35) with new constraints accounting for friction;

$$w^i_{\Delta p,k} = (p^i_k - p^{i+n_w}_k) - \overline{\rho}^i_{tp,k} g h^i \cdot 10^{-5} - \Delta p^i_{f,k}, \qquad \forall\ i \in \mathscr{W}, k \in \mathscr{H}, \tag{B.20}$$

$$w^p_{\Delta p,k} = (p^{3n_w}_k - p^{3n_w+1}_k) - \overline{\rho}^p_{tp,k} g h^p \cdot 10^{-5} - \Delta p^p_{f,k}, \qquad \forall\ k \in \mathscr{H}, \tag{B.21}$$

which completes the friction model. On a side note: The observant reader may have noted that some of the constraints above are not simple linear/bilinear constraints, which are the relevant constraint types supported in CENSO. To implement these constraints, a set of auxiliary variables are defined, and the constraints are constructed by means of a series of linear and bilinear constraints.

Appendix C

Calculations

C.1 Average pressure

This appendix section describes an *ad hoc* method for calculating a multiplier $k_{\overline{p}}$ intended to obtain a better approximation of the average pressure $\overline{p}$ in a pipeline than the "standard" $\overline{p} = \frac{1}{2}(p^{us} + p^{ds})$, where p^{us} is the upstream pressure and p^{ds} is the downstream pressure. We rather calculate the average pressure as

$$\overline{p}' = \frac{k_{\overline{p}}}{2}(p^{us} + p^{ds}), \tag{C.1}$$

where $k_{\overline{p}}$ is intended to factor in the pipeline geometry. In this thesis, the average pressure in a well or pipeline is calculated by using information from the flow path (well/pipeline) geometry files (.geo) from OLGA. Each flow path is divided into a number of segments, which we will denote n. The geometry files contain information about the lengths $\left\{L^i_{\text{seg}}\right\}_{i\in\mathscr{S}}$, the elevation changes $\left\{\Delta h^i_{\text{seg}}\right\}_{i\in\mathscr{S}}$, the diameters $\left\{D^i_{\text{seg}}\right\}_{i\in\mathscr{S}}$ and the roughnesses $\left\{\epsilon^i_{\text{seg}}\right\}_{i\in\mathscr{S}}$ of each segment. Here, we have defined the index set $\mathscr{S} = \{1,\dots,n\}$ as all the segments. We calculate a pressure profile along the pipeline under the assumption that the pressure profile is decided by hydrostatic pressure loss, i.e. the elevation profile. First, we calculate the elevation profile $\{h^i\}_{i=1}^{n+1}$ as

$$h^1 = 0, \quad h^{i+1} = h^i + \Delta h^i_{\text{seg}}, \ \ \forall\ i \in \mathscr{S} \tag{C.2}$$

where h^i is the elevation at the boundary between segment i and segment $i-1$, with respect to the reference 0 set at the inlet of the first segment. The total elevation change for the pipe is then h^{n+1}. We now calculate a pressure profile $\left\{p^i_{seg}\right\}_{i=1}^{n+1}$ as

$$p^1_{\text{seg}} = p^{us}, \quad p^i_{\text{seg}} = p^{us} - (p^{us} - p^{ds})\frac{h^i}{h^{n+1}}, \ \ \forall\ i \in \{1,\dots,n+1\}. \tag{C.3}$$

Now p^i_{seg} is the pressure at the inlet of segment i (and $p^{n+1} = p^{ds}$ is the pressure at the outlet of the last segment). We are now able to calculate the average pressure in each segment $\overline{p}^i_{\text{seg}}$ as

$$\overline{p}^i_{\text{seg}} = \frac{1}{2}(p^i_{\text{seg}} + p^{i+1}_{\text{seg}}), \;\; \forall\; i \in \mathscr{S}. \tag{C.4}$$

The modified average pressure in the pipeline $\overline{p}'$ is now taken as a weighted average of all the segment average pressures, using the segment lengths as weights;

$$\overline{p}' = \frac{\sum_{i\in\mathscr{S}} L^i_{\text{seg}}\overline{p}^i_{\text{seg}}}{\sum_{i\in\mathscr{S}} L^i_{\text{seg}}}. \tag{C.5}$$

Then, the multiplier $k_{\overline{p}}$ is taken as the ratio between the modified average pressure $\overline{p}'$ and the "standard" average pressure $\overline{p}$:

$$k_{\overline{p}} = \frac{\overline{p}'}{\overline{p}} = \frac{\overline{p}'}{\frac{1}{2}(p^{us} + p^{ds})}. \tag{C.6}$$

The idea is that now we can calculate the modified average pressure $\overline{p}'$ by means of the simple calculation (C.1), which can easily be included in our optimization problem without performing a spatial discretization which would lead to a large number of additional variables. The values of $k_{\overline{p}}$ for the wells and pipelines in Tilje all turned out to be roughly 1.1.

C.2 Initial mass and holdup estimates for dynamic flow estimator

Before the (dynamic) estimation algorithm is started, it must be provided with an initial estimate of the liquid holdup in each volume, and the masses of each phase. This is done by first solving the static estimation problem to find estimated flow rates through each volume, and subsequently solving the pressure drop model for liquid holdup and mass.

C.2.1 Standard pressure model

The following procedure is used for calculating mass and holdup estimates for the pressure model used in Problem 5.1:

1. Given a measurement vector $\mathbf{y}$, solve Problem 4.1 to obtain the solution $\mathbf{x}^*$.

2. The following values from $\mathbf{x}^*$ are used ($*$-superscripts omitted); volumetric flow rate of each phase $\mathbf{q}_o$, $\mathbf{q}_g$, $\mathbf{q}_w$, pressure measurements $\tilde{\mathbf{p}}$, temperature measurements $\tilde{\mathbf{T}}$ and pipeline water cut r^p_{wc}.

3. Calculate average pressures in well and pipeline volumes:

$$\text{Wells:} \qquad \overline{p}^i = \frac{k^i_{\overline{p}}}{2}\left(p^i + p^{i+n_w}\right), \ \forall\ i \in \mathcal{W} \qquad \text{[bara]} \tag{C.7}$$

$$\text{Pipeline:} \qquad \overline{p}^p = \frac{k_{\overline{p}^p}}{2}\left(p^{3n_w} + p^{3n_w+1}\right). \qquad \text{[bara]} \tag{C.8}$$

$k^i_{\overline{p}}$ and $k^p_{\overline{p}}$ are constants related to well/pipeline geometry (their calculation is shown in Appendix C.1).

4. Calculate average temperatures in well and pipeline volumes:

$$\text{Wells:} \qquad \overline{T}^i = \frac{1}{2}\left(\tilde{T}^i + \tilde{T}^{i+n_w}\right), \ \forall\ i \in \mathcal{W} \qquad [^\circ\text{C}] \tag{C.9}$$

$$\text{Pipeline:} \qquad \overline{T}^p = \frac{1}{2}\left(\tilde{T}^{3n_w} + \tilde{T}^{3n_w+1}\right). \qquad [^\circ\text{C}] \tag{C.10}$$

5. Evaluate the spline-approximated PVT tables at the average pressure and temperature to obtain the average oil, gas and water densities;

$$\text{Wells:} \qquad \overline{\rho}^i_o = \Phi_{\rho_o}(\overline{p}^i, \overline{T}^i), \ \forall\ i \in \mathcal{W} \qquad [\text{kg/m}^3] \tag{C.11}$$

$$\overline{\rho}^i_g = \Phi_{\rho_g}(\overline{p}^i, \overline{T}^i), \ \forall\ i \in \mathcal{W} \qquad [\text{kg/m}^3] \tag{C.12}$$

$$\overline{\rho}^i_w = \Phi_{\rho_w}(\overline{p}^i, \overline{T}^i), \ \forall\ i \in \mathcal{W} \qquad [\text{kg/m}^3] \tag{C.13}$$

$$\text{Pipeline:} \qquad \overline{\rho}^p_o = \Phi_{\rho_o}(\overline{p}^p, \overline{T}^p), \qquad [\text{kg/m}^3] \tag{C.14}$$

$$\overline{\rho}^p_g = \Phi_{\rho_g}(\overline{p}^p, \overline{T}^p), \qquad [\text{kg/m}^3] \tag{C.15}$$

$$\overline{\rho}^p_w = \Phi_{\rho_w}(\overline{p}^p, \overline{T}^p). \qquad [\text{kg/m}^3] \tag{C.16}$$

6. Calculate average liquid densities based on water cuts (the well water cuts r^i_{wc} are constants while the pipeline water cut r^p_{wc} is taken from $\mathbf{x}^*$);

$$\text{Wells:} \qquad \overline{\rho}^i_l = r^i_{wc}\overline{\rho}^i_w + (1 - r^i_{wc})\overline{\rho}^i_o, \ \forall\ i \in \mathcal{W}, \qquad [\text{kg/m}^3] \tag{C.17}$$

$$\text{Pipeline:} \qquad \overline{\rho}^p_l = r^p_{wc}\overline{\rho}^p_w + (1 - r^p_{wc})\overline{\rho}^p_o. \qquad [\text{kg/m}^3] \tag{C.18}$$

7. Calculate the hydrostatic pressure drops based on differential pressure (assuming zero frictional pressure loss);

$$\text{Wells:} \qquad \Delta p^i_{\text{hs}} = p^i - p^{i+n_w}, \ \forall\ i \in \mathcal{W}, \qquad \text{[bar]} \tag{C.19}$$

$$\text{Pipeline:} \qquad \Delta p^p_{\text{hs}} = p^{3n_w} - p^{3n_w+1}. \qquad \text{[bar]} \tag{C.20}$$

8. Solve pressure drop model (see Eqn. (5.34)) for two-phase densities;

$$\text{Wells:} \qquad \overline{\rho}^{i}_{tp} = \frac{\Delta p^{i}_{\text{hs}}}{g h^{i}} \cdot 10^{5}, \;\; \forall\, i \in \mathscr{W} \qquad [\text{kg/m}^3] \tag{C.21}$$

$$\text{Pipeline:} \qquad \overline{\rho}^{p}_{tp} = \frac{\Delta p^{p}_{\text{hs}}}{g h^{p}} \cdot 10^{5}. \qquad [\text{kg/m}^3] \tag{C.22}$$

9. Solve (5.52)/(5.53) for liquid holdup;

$$\text{Wells:} \qquad H^{i}_{l} = \frac{\overline{\rho}^{i}_{tp} - \overline{\rho}^{i}_{g}}{\overline{\rho}^{i}_{l} - \overline{\rho}^{i}_{g}}, \;\; \forall\, i \in \mathscr{W}, \qquad [\text{-}] \tag{C.23}$$

$$\text{Pipeline:} \qquad H^{p}_{l} = \frac{\overline{\rho}^{p}_{tp} - \overline{\rho}^{p}_{g}}{\overline{\rho}^{p}_{l} - \overline{\rho}^{p}_{g}}. \qquad [\text{-}] \tag{C.24}$$

10. Calculate masses using the liquid holdup and densities. Here, V_o, V_g, V_w are volumes occupied by oil, gas and water, respectively, and V_p is the total volume of the well/pipeline:

$$\text{Wells:} \qquad m^{i}_{o} = \overline{\rho}^{i}_{o} V^{i}_{o} = \overline{\rho}^{i}_{o} H^{i}_{l} (1 - r^{i}_{wc}) V^{i}_{p}, \;\; \forall\, i \in \mathscr{W}, \qquad [\text{kg}] \tag{C.25}$$

$$m^{i}_{g} = \overline{\rho}^{i}_{g} V^{i}_{g} = \overline{\rho}^{i}_{g} (1 - H^{i}_{l}) V^{i}_{p}, \;\; \forall\, i \in \mathscr{W}, \qquad [\text{kg}] \tag{C.26}$$

$$m^{i}_{w} = \overline{\rho}^{i}_{w} V^{i}_{w} = \overline{\rho}^{i}_{w} H^{i}_{l} r^{i}_{wc} V^{i}_{p}, \;\; \forall\, i \in \mathscr{W}, \qquad [\text{kg}] \tag{C.27}$$

$$m^{i}_{l} = m^{i}_{o} + m^{i}_{w}, \;\; \forall\, i \in \mathscr{W}, \qquad [\text{kg}] \tag{C.28}$$

$$\text{Pipeline:} \qquad m^{p}_{o} = \overline{\rho}^{p}_{o} V^{p}_{o} = \overline{\rho}^{p}_{o} H^{p}_{l} (1 - r^{p}_{wc}) V^{p}_{p}, \qquad [\text{kg}] \tag{C.29}$$

$$m^{p}_{g} = \overline{\rho}^{p}_{g} V^{p}_{g} = \overline{\rho}^{p}_{g} (1 - H^{p}_{l}) V^{p}_{p}, \qquad [\text{kg}] \tag{C.30}$$

$$m^{p}_{w} = \overline{\rho}^{p}_{w} V^{p}_{w} = \overline{\rho}^{p}_{w} H^{p}_{l} r^{i}_{wc} V^{p}_{p}, \qquad [\text{kg}] \tag{C.31}$$

$$m^{p}_{l} = m^{p}_{o} + m^{p}_{w},. \qquad [\text{kg}] \tag{C.32}$$

C.2.2 Extended pressure model with friction

When the friction model is included, we solve the pressure drop model for liquid holdup using a Newton-iterative approach. To avoid confusion, we do this for well i only, however, the procedure is valid for the pipeline volume as well (the only difference is the pressures and temperatures used in the calculations):

1. Complete Steps 1-6 in the procedure above (Section C.2.1).

2. Evaluate the spline-approximated PVT tables at the average pressure and temperature to obtain the average oil and water viscosities ($\overline{\mu}_o$ and $\overline{\mu}_w$):

$$\overline{\mu}^{i}_{o} = \Phi_{\mu_o}(\overline{p}^{i}, \overline{T}^{i}) \qquad [\text{N/sm}^2] \tag{C.33}$$

$$\overline{\mu}^{i}_{w} = \Phi_{\mu_w}(\overline{p}^{i}, \overline{T}^{i}) \qquad [\text{N/sm}^2] \tag{C.34}$$

3. Use the obtained values to calculate the average liquid viscosities:

$$\overline{\mu}_l^i = r_{wc}^i \overline{\mu}_w^i + (1 - r_{wc}^i)\overline{\mu}_o^i, \ \ \forall\ i \in \mathscr{W}, \qquad [\mathrm{N/sm^2}] \qquad (C.35)$$

4. Calculate the liquid mass flow rate $\overline{w}_l^i$ using (B.7), the liquid superficial velocity $\overline{v}_{sl}^i$ using (B.5), the Reynolds number N_{Re}^i using (B.13), and the Darcy friction factor λ_f^i using (B.16). These values are precalculated since they do not change from one Newton iteration to the next.

5. Assume a frictional (and accelerational) pressure loss of zero. Then, (3.15) reduces to $\Delta p^i = \Delta p_{hs}^i$, where $\Delta p^i = \tilde{p}^i - \tilde{p}^{n_w+i}$ is given by the pressure measurements. This can be used to calculate an initial guess $\Delta\hat{p}_{hs,0}^i$ of the hydrostatic pressure loss:

$$\Delta\hat{p}_{hs,0}^i = \tilde{p}^i - \tilde{p}^{n_w+i}. \qquad [\mathrm{bar}] \qquad (C.36)$$

6. Solve (5.34) for two-phase density using the estimate for hydrostatic pressure loss (assuming $w_{\Delta p}^i = 0$ in (5.34)):

$$\overline{\rho}_{tp}^i = \frac{\Delta\hat{p}_{hs,k}^i}{gh^i} \cdot 10^5. \qquad [\mathrm{kg/m^3}] \qquad (C.37)$$

7. Solve (5.52) for liquid holdup. Note that this is also an initial guess, since the "true" hydrostatic pressure loss is slightly smaller than our guess due to friction.

$$\hat{H}_{l,k}^i = \frac{\overline{\rho}_{tp}^i - \overline{\rho}_g^i}{\overline{\rho}_l^i - \overline{\rho}_g^i}. \qquad [-] \qquad (C.38)$$

8. Use this liquid holdup to calculate the frictional pressure loss using (B.18):

$$\Delta p_f^i = 10^{-5} \frac{\hat{H}_{l,k}^i \lambda_f^i \overline{\rho}_l^i \left(\overline{v}_{sl}^i\right)^2}{2D_p^i} L_p^i. \qquad [\mathrm{bar}] \qquad (C.39)$$

9. Use the calculated frictional pressure loss to generate a new estimate for the hydrostatic pressure drop:

$$\Delta\hat{p}_{hs,k+1}^i = \tilde{p}^i - \tilde{p}^{n_w+i} - \Delta p_f^i \qquad [\mathrm{bar}] \qquad (C.40)$$

10. If $|\Delta\hat{p}_{hs,k+1}^i - \Delta\hat{p}_{hs,k}^i| > \epsilon$, where ϵ is some tolerance, increment k and repeat steps 6 through 9.

11. Take $\hat{H}_{l,k}^i \triangleq H_l^i$ as the estimated liquid holdup. Calculate the mass of each phase using step 10 in the procedure for the static model (Section C.2.1).

C.3 Choke valve pressure drop

Algorithm C.1 below shows the evaluation of the function (4.18) in Section 4.5.3.

Algorithm C.1: Calculation of choke pressure drop using a multiplier model and PVT splines.

```
Data: $q_l, p^{us}, \overline{T}, C_v, r_{go}, r_{wc}, c$
Result: $p^{ds}$
Guess an initial pressure drop of zero: $p^{ds} = p^{us}$
repeat
    Set downstream pressure estimate: $\hat{p}^{ds} = p^{ds}$
    Calculate average pressure: $\overline{p} = \frac{1}{2} p^{us} + \frac{1}{2} \hat{p}^{ds}$
    Calculate volumetric flow rates at standard conditions:
    $q_w = r_{wc} q_l$ (water)
    $q_o = q_l - q_w$ (oil)
    $q_g = r_{go} q_o$ (gas)
    Calculate mass flow rates at standard conditions:
    $w_o = \rho_o^{\mathrm{std}} q_o$ (oil)
    $w_g = \rho_g^{\mathrm{std}} q_g$ (gas)
    $w_{hc} = w_o + w_g$ (hydrocarbons)
    Get fluid properties from spline evaluation:
    $\alpha_g = \Phi_{\alpha_g}(\overline{p}, \overline{T})$ (gas mass fraction)
    $\rho_g = \Phi_{\rho_g}(\overline{p}, \overline{T})$ (gas density)
    $\rho_o = \Phi_{\rho_o}(\overline{p}, \overline{T})$ (oil density)
    $\rho_w = \Phi_{\rho_w}(\overline{p}, \overline{T})$ (water density)
    $\alpha_g = \Phi_{\alpha_g}(\overline{p}, \overline{T})$ (gas mass fraction)
    Calculate in-situ mass and volumetric flows:
    $w_g^{\mathrm{is}} = \alpha_g w_{hc}$ (in-situ gas mass flow)
    $w_o^{\mathrm{is}} = w_{hc} - w_g$ (in-situ oil mass flow)
    $q_o^{\mathrm{is}} = w_o / \rho_o$ (in-situ oil volumetric flow)
    $q_l^{\mathrm{is}} = q_o^{\mathrm{is}} + q_w$ (in-situ liquid volumetric flow)
    Calculate liquid density:
    $\rho_l = (1 - r_{wc})\rho_o + r_{wc}\rho_w$
    Calculate multiphase multiplier (Morris with Chisholm slip correlation):
    $k = \sqrt{\alpha_g \frac{\rho_l}{\rho_g} + (1 - \alpha_g)}$
    $\Psi_{lo}^2 = \left[\alpha_g \frac{\rho_l}{\rho_g} + k(1 - \alpha_g)\right] \left[\alpha_g + \frac{(1-\alpha_g)}{k} + \left(1 + \frac{(k-1)^2}{\sqrt{\rho_l/\rho_g}}\right)\right]$
    Calculate pressure drop:
    $\Delta p = c \cdot \Psi_{lo}^2 \frac{\rho_l}{\rho_w^{\mathrm{std}}} \left(\frac{q_l^{\mathrm{is}}}{C_v}\right)^2.$
    Calculate new estimate for downstream pressure:
    $p^{ds} = p^{us} - \Delta p$
until $|p^{ds} - \hat{p}^{ds}| \leq \epsilon_p$
return $p^{ds}$
```

C.4 Extrapolating the WPC zero-crossing point

As the wellhead pressure increases, the liquid rate from the well will eventually reach zero. When sampling WPC curves in GAP with a fixed interval between the wellhead pressures, this zero-crossing point may occur in between samples. The location of the zero-crossing point is important, since it determines the wellhead pressure at which a well will no longer produce. One alternative to obtain a more accurate zero-crossing point is to inspect the WPC and sample more tightly as the liquid rate approaches zero. However, this requires some know-how and is difficult to automate, especially if the sampled WPC is to be calibrated at a later stage. Therefore, the last three samples are used to extrapolate the zero-crossing point using a second-order Taylor expansion based on approximated derivatives. This approximation will not be particularly accurate in terms of predicting the correct zero-crossing point, but it will make sure the zero-rate is included in the feasible set of the NLP, and it will prevent oscillations in the B-spline approximation of the WPC. Moreover, it may be performed *after* an offset calibration, so the zero rate is defined even when the WPC has been lifted. For the rate WPC, let $\{(p_k^{wh}, q_{l,k})\}_{k=1}^{3}$ be the last three WPC samples obtained from GAP (or some other software), and let $(p_4^{wh}, 0)$ be the desired zero-crossing point. Further, we use finite differences to define the following approximated first derivatives at $p^{wh} = p_2^{wh}$ and $p^{wh} = p_3^{wh}$:

$$\left.\frac{\mathrm{d}q_l}{\mathrm{d}p^{wh}}\right|_{p_2^{wh}} \approx \partial_2 = \frac{q_{l,2} - q_{l,1}}{p_2^{wh} - p_1^{wh}} \quad \text{and} \quad \left.\frac{\mathrm{d}q_l}{\mathrm{d}p^{wh}}\right|_{p_3^{wh}} \approx \partial_3 = \frac{q_{l,3} - q_{l,2}}{p_3^{wh} - p_2^{wh}} \tag{C.41}$$

Then, we approximate the second derivative at $p^{wh} = p_3^{wh}$ using the approximated first derivatives from (C.41):

$$\left.\frac{\mathrm{d}^2 q_l}{(\mathrm{d}p^{wh})^2}\right|_{p_3^{wh}} \approx \partial_3^2 = \frac{\partial_3 - \partial_2}{p_3^{wh} - p_1^{wh}}. \tag{C.42}$$

A second-order Taylor expansion of the WPC curve around $p^{wh} = p_3^{wh}$ is given as

$$\tilde{f}_{\text{wpc}}(p^{wh}) = \tilde{f}_{\text{wpc}}(p_3^{wh}) + \left.\frac{\mathrm{d}q_l}{\mathrm{d}p^{wh}}\right|_{p_3^{wh}} (p^{wh} - p_3^{wh}) + \frac{1}{2}\left.\frac{\mathrm{d}^2 q_l}{(\mathrm{d}p^{wh})^2}\right|_{p_3^{wh}} (p^{wh} - p_3^{wh})^2 \tag{C.43}$$

$$\approx q_{l,3} + \partial_3 (p^{wh} - p_3^{wh}) + \frac{\partial_3^2}{2}(p^{wh} - p_3^{wh})^2. \tag{C.44}$$

Inserting the zero-crossing point as $\tilde{f}_{\text{wpc}}(p_4^{wh}) = 0$, we obtain the following quadratic equation, which can be solved for p_4^{wh} using the quadratic formula:

$$0 = q_{l,3} + \partial_3 q_{l,3}(p_4^{wh} - p_3^{wh}) + \frac{\partial_3^2}{2}(p_4^{wh} - p_3^{wh})^2 \tag{C.45}$$

$$= \frac{1}{2}\partial_3^2 (p_4^{wh})^2 + (\partial_3 - \partial_3^2 p_3^{wh}) p_4^{wh} + q_{l,3} + \frac{1}{2}\partial_3^2 (p_3^{wh})^2. \tag{C.46}$$

Appendix D

Simulation settings and results

D.1 Key settings for OLGA model

	Keyword	Setting	Description
Case level	MINDT	10^{-9} sec	Minimum time step during simulation
	MAXDT	10 sec	Maximum time step during simulation
	TEMPERATURE	WALL	Enables the flow path walls to transfer and store heat.
	COMPOSITIONAL	OFF	Uses the PVT tables for fluid property calculations.
	FLASHMODEL	WATER	Enables mass transfer between gas/oil and gas/water.
	NOSLIP	OFF	Enables slip between phases, i.e. phases are allowed to travel at different velocities along the flow paths.
	HYDSLUG	ON	Enables hydrodynamic flow regimes (hydrodynamic slug flow and dispersed bubble flow).
Wells	PRODOPTION	VOGELS	Vogel's IPR curve from (Vogel, 1968) is used for inflow calculation.
	ISOTHERMAL	YES	Assumes isothermal inflow.
Valves	MODEL	HYDROVALVE	The Hydro valve model (described in (Schüller et al., 2003)) is used for flow calculation.
	EQUILIBRIUMMODEL	FROZEN	Assumes no mass transfer (flashing) inside the choke.
	THERMALPHASEEQ	NO	Gas is expanded isentropically through the valve, while the liquid is isothermal.
	C_v model	-	Taken from tables obtained in water test facility (PHASE = LIQUID).

Table D.1: Key settings for OLGA model.

D.2 Estimation errors and solve times

	i	**Well adj.** (s)	**Well adj.** (s)*	**Well adj.** (d)	**Slugging** (s)	**Slugging** (d)	**Field data** (s)	**Field data** (d)
E^i_{max} [%]	0	0.62	3.69	0.74	0.29	0.28	-	-
	1	0.12	1.43	0.17	0.10	0.14	-	-
	2	3.93	1.12	1.50	1.89	1.54	5.60	5.99
	3	1.79	0.51	0.62	1.17	0.88	-	-
	6	3.31	0.98	1.11	3.06	2.42	8.54	7.18
	7	2.67	3.78	3.56	69.2	47.7	-	-
E^i_{ise}	0	347	6609	284	1072	2251	-	-
	1	55	806	74	178	507	-	-
	2	7212	866	2609	21490	48723	84024	164908
	3	1137	322	996	10906	15322	-	-
	6	5271	628	861	60870	117829	73358	60810
	7	5125	8836	10997	18662352	37695379	-	-
E_{tot}		19147	18066	15822	18756869	37880011	157382	225719
$\bar{t}_{\text{solve}}$ [ms]		559	629	5739	483	9068	267	3636

Table D.2: Errors and solve times for simulation cases. (s) - static estimation, (d) - dynamic estimation, * - extra weight on choke models.

D.3 Parameters for simulation cases

Parameter	**Index**	**Well adj.** (s)	**Well adj.** (s) *	**Well adj.** (d)	**Slugging** (s)	**Slugging** (d)	**Field data** (s)	**Field data** (d)
$\mathbf{P_{m0}}(i,i)$	0-9	-	-	100	-	100	-	100
$\mathbf{P_{q0}}(i,i)$	0	-	-	1	-	1	-	1
	1	-	-	1	-	10	-	10^2
	2	-	-	1	-	1	-	1
	3	-	-	1	-	10	-	10^2
	4	-	-	1	-	1	-	1
	5	-	-	1	-	1	-	1
	6	-	-	1	-	1	-	1
	7	-	-	1	-	10^2	-	10^3
$\mathbf{P_{q}}(i,i)$	0	1	1	1	1	1	1	1
	1	1	1	1	1	10	1	10^4
	2	1	1	1	1	1	1	1
	3	1	1	1	1	10	1	10^4
	4	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1
	7	1	1	1	1	10^2	1	10^4
$\mathbf{Q}^{-1}(i,i)$	0 (IPR A)	0.567	0.567	0.567	0.567	10^2	10^3	10
	1 (IPR B)	1.031	1.031	1.031	1.031	10^2	10^2	10
	2 (WPC A)	1.119	1.119	1.119	1.119	10^2	10^3	10
	3 (WPC B)	3.115	3.115	3.115	3.115	10^2	10^2	10
	4 (Chk. A)	27.78	2778	2778	27.78	10^3	$2 \cdot 10^4$	10^2
	5 (Chk. B)	10^3	10^5	10^5	10^3	10^3	10^4	10^2
	6 (VLP)	0.093	0.093	0	0	0	0	0
	7 (T. chk.)	9.8	9.8	9.8	9.8	10^2	10^2	0
	8 (Well A Δp)	-	-	10^5	-	10^4	-	10^5
	9 (Well B Δp)	-	-	10^5	-	10^4	-	10^4
	10 (Pipeline Δp)	-	-	10^5	-	10^4	-	10^7
$\mathbf{R}^{-1}(i,i)$	0-8	10^6	10^6	10^6	10^6	10^5	10^6	10^5
N		-	-	5	-	5	-	20
Δt		10 s	10 s	30 s	10 s	30 s	30 s	90 s

Table D.3: Parameters for simulation cases. (s) - static estimation, (d) - dynamic estimation, * - extra weight on choke models.

D.4 Additional figures

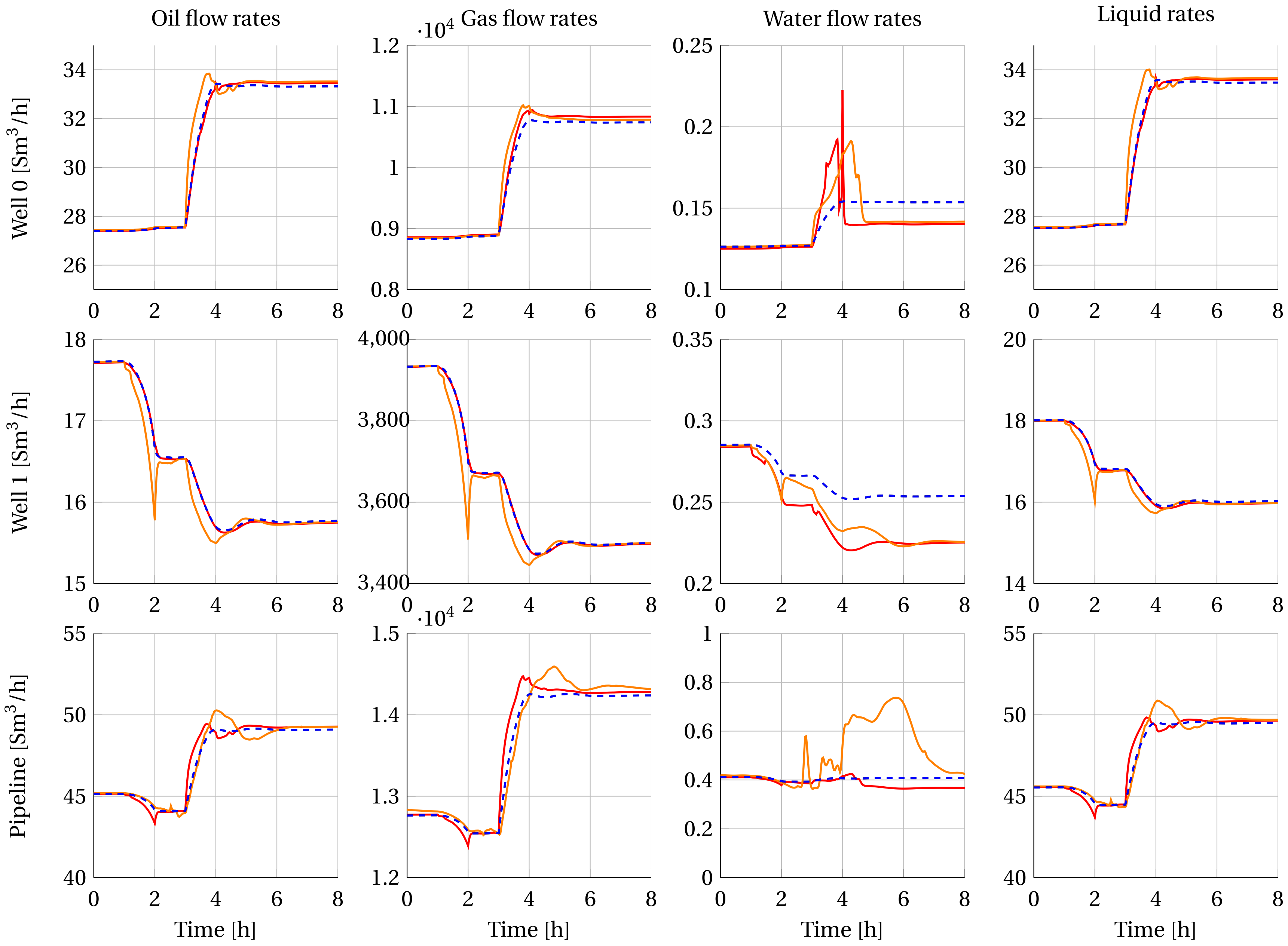


Figure D.1: Flow rates for well adjustment case (static estimation). Legend: Red; Measured inflow rates (from OLGA), Orange; Measured outflow rates (from OLGA), Dashed blue; estimated flow rates. All flow rates in [Sm^3/h].

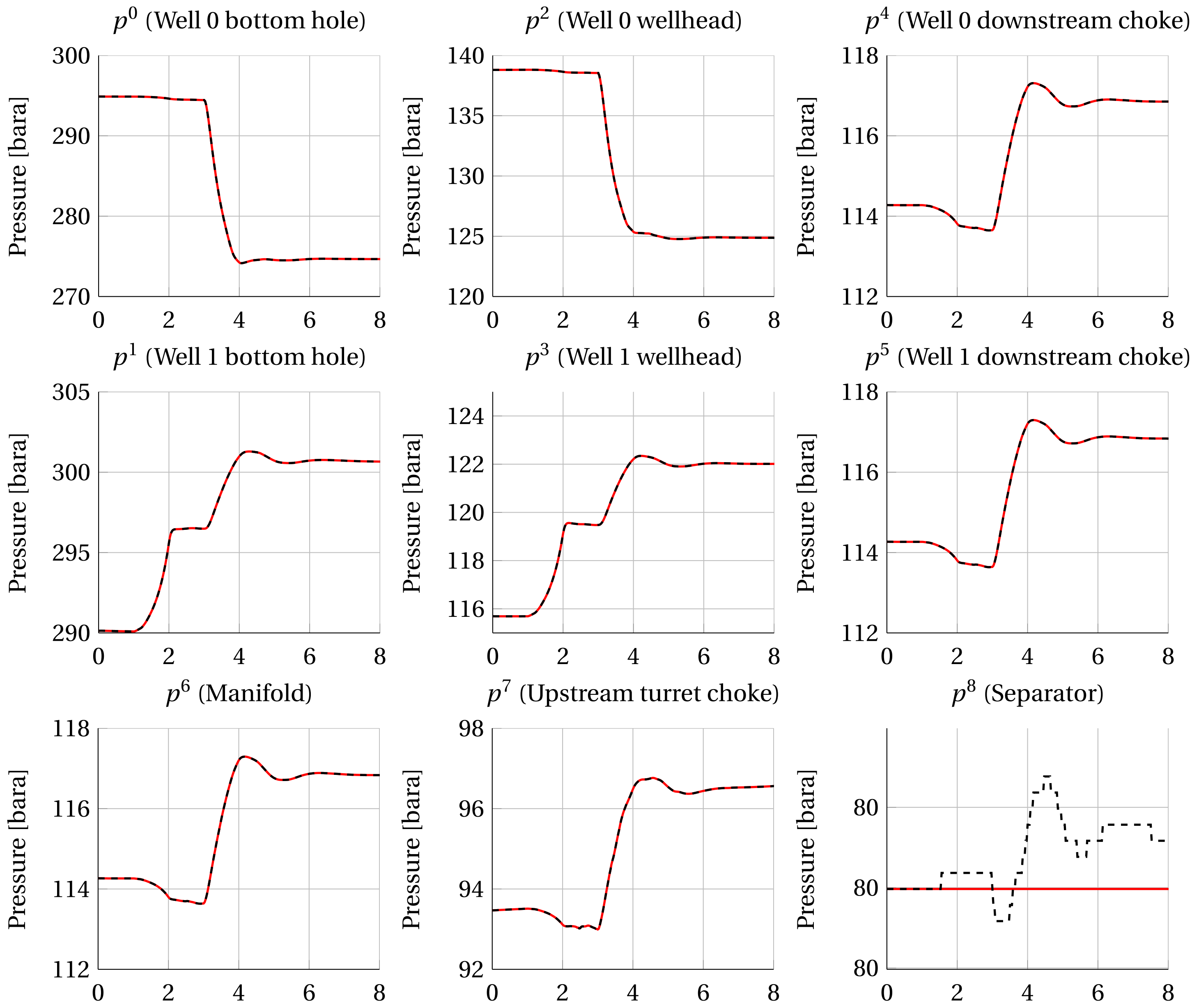


Figure D.2: Pressures for well adjustment case (static estimation). Legend: Red; Measured pressures (from OLGA), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

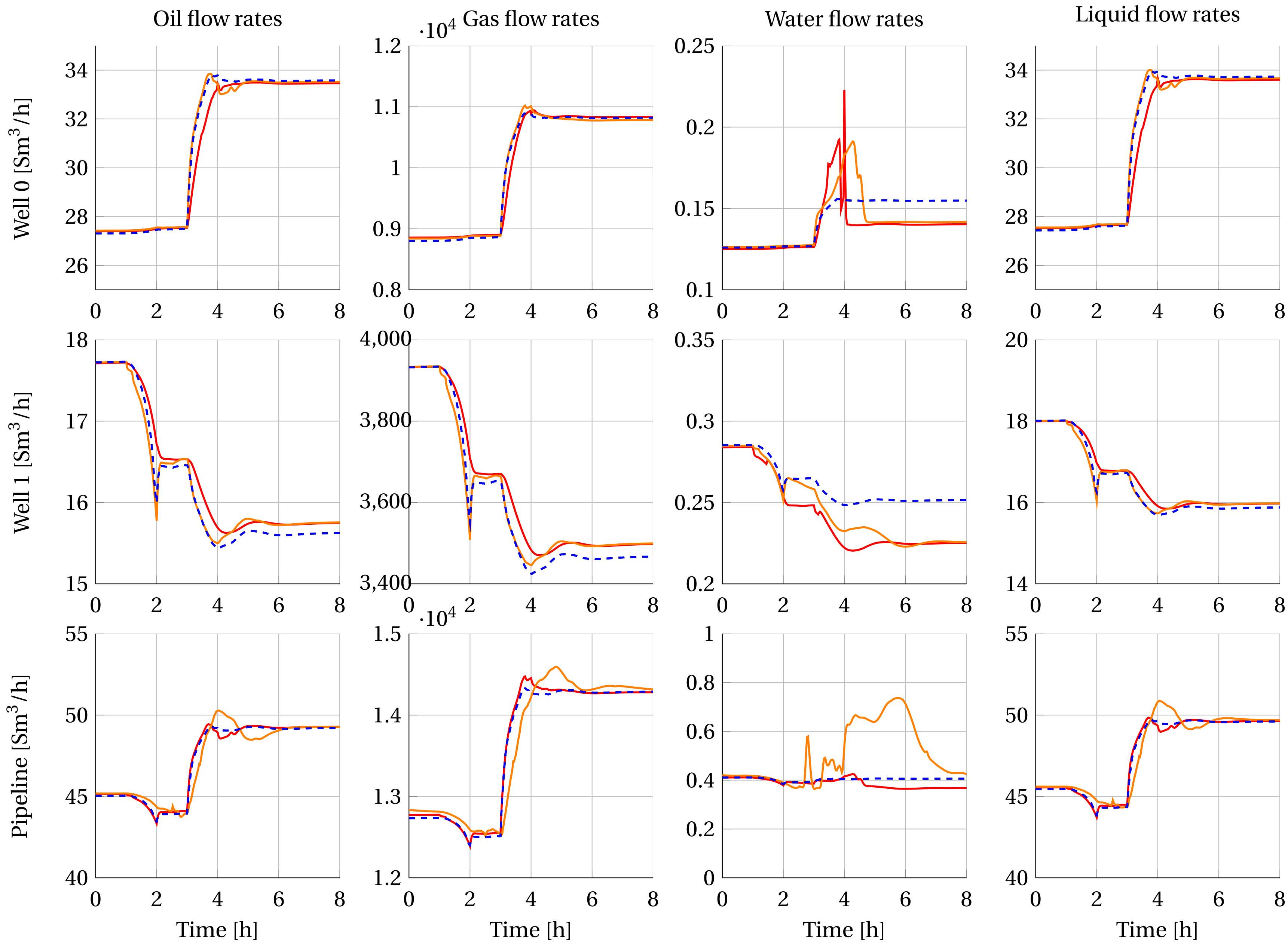


Figure D.3: Flow rates for well adjustment case (static estimation with extra weight on choke models). Legend: Red; Measured inflow rates (from OLGA), Orange; Measured outflow rates (from OLGA), Dashed blue; estimated flow rates. All flow rates in [Sm^3/h].

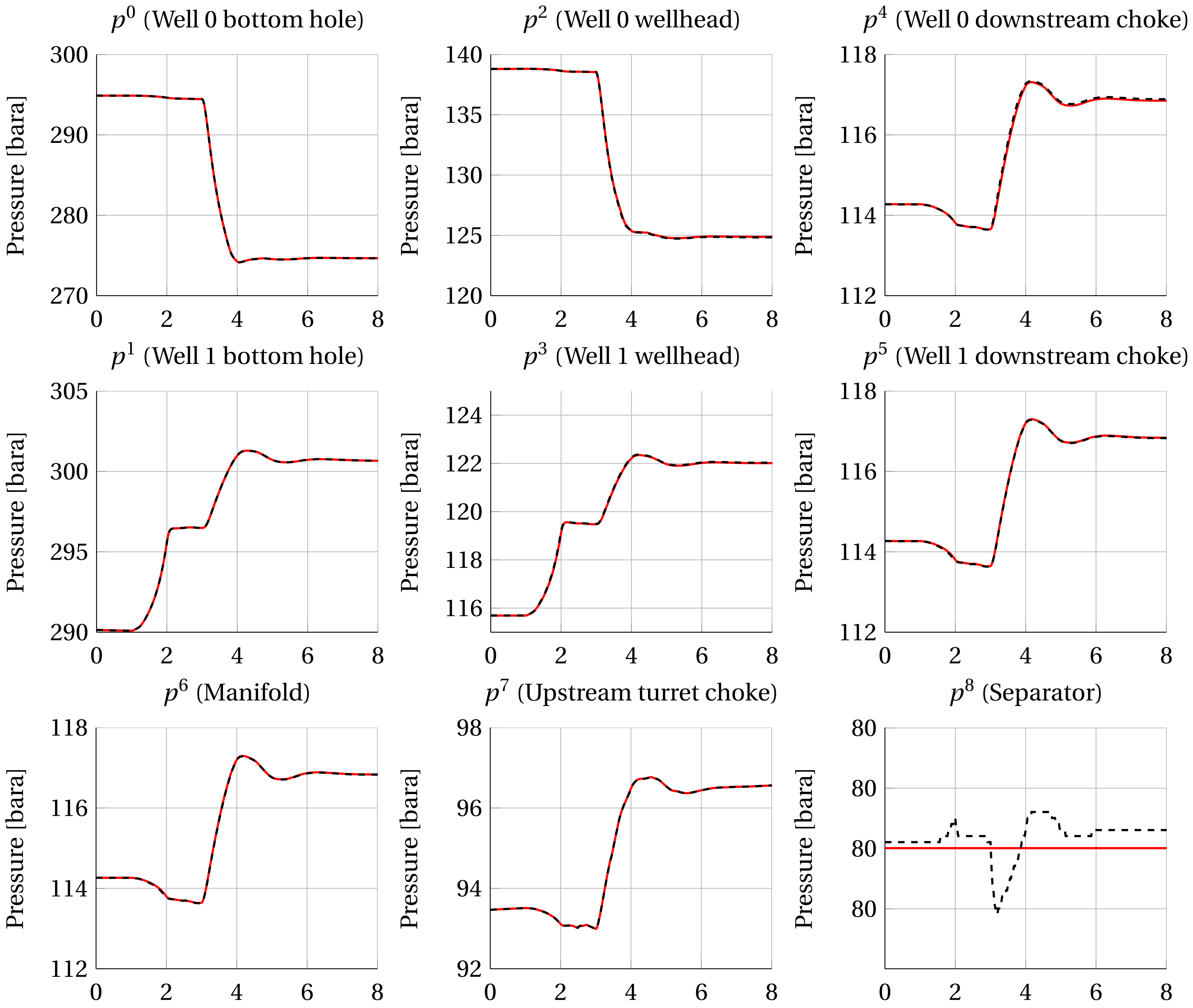


Figure D.4: Pressures for well adjustment case (static estimation with extra weight on choke models). Legend: Red; Measured pressures (from OLGA), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

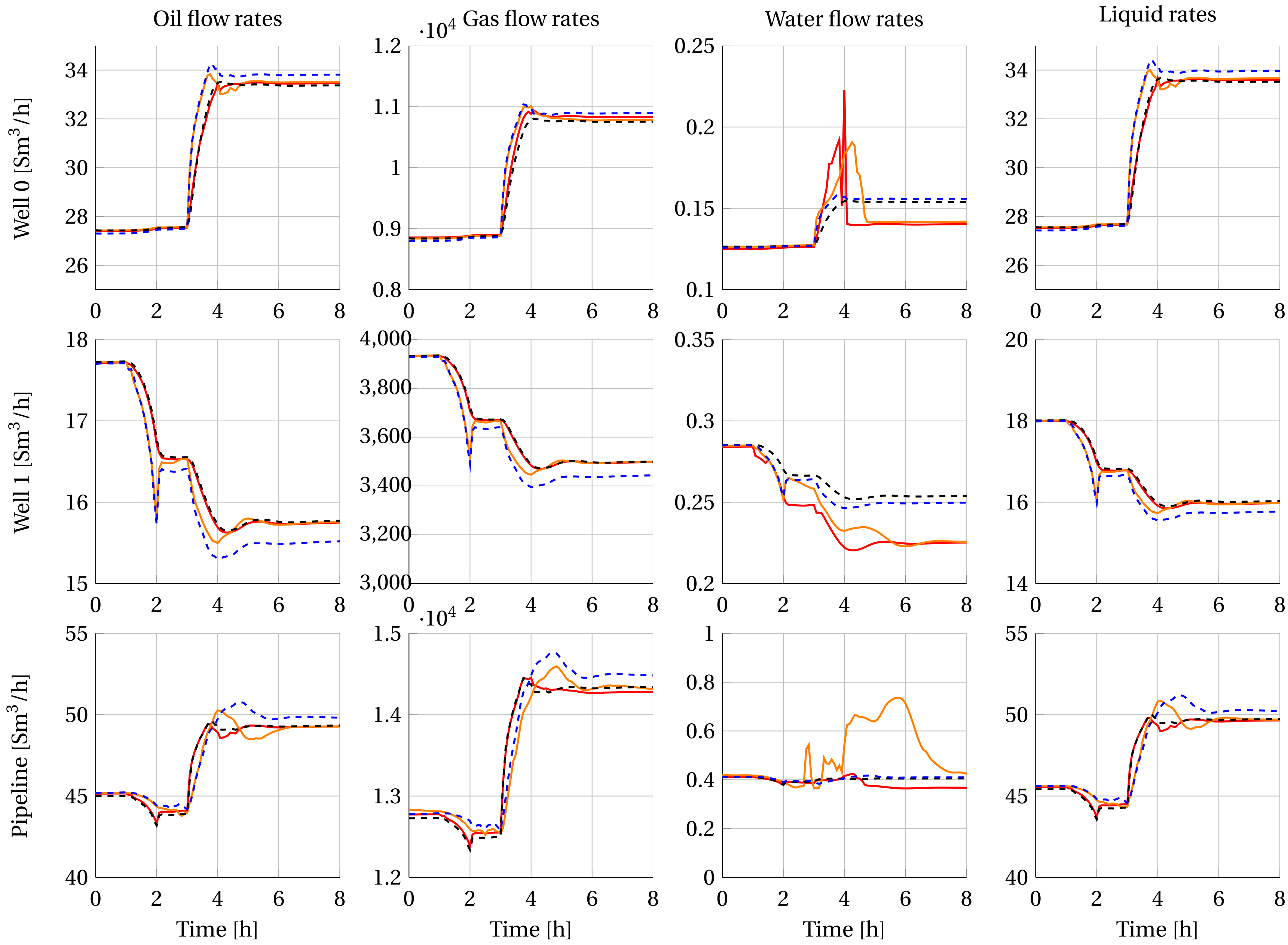


Figure D.5: Flow rates for well adjustment case (dynamic estimation). Legend: Red; Measured inflow rates (from OLGA), Orange; Measured outflow rates (from OLGA), Dashed black; estimated inflow rates, Dashed blue; estimated outflow rates. All flow rates in [Sm^3/h].

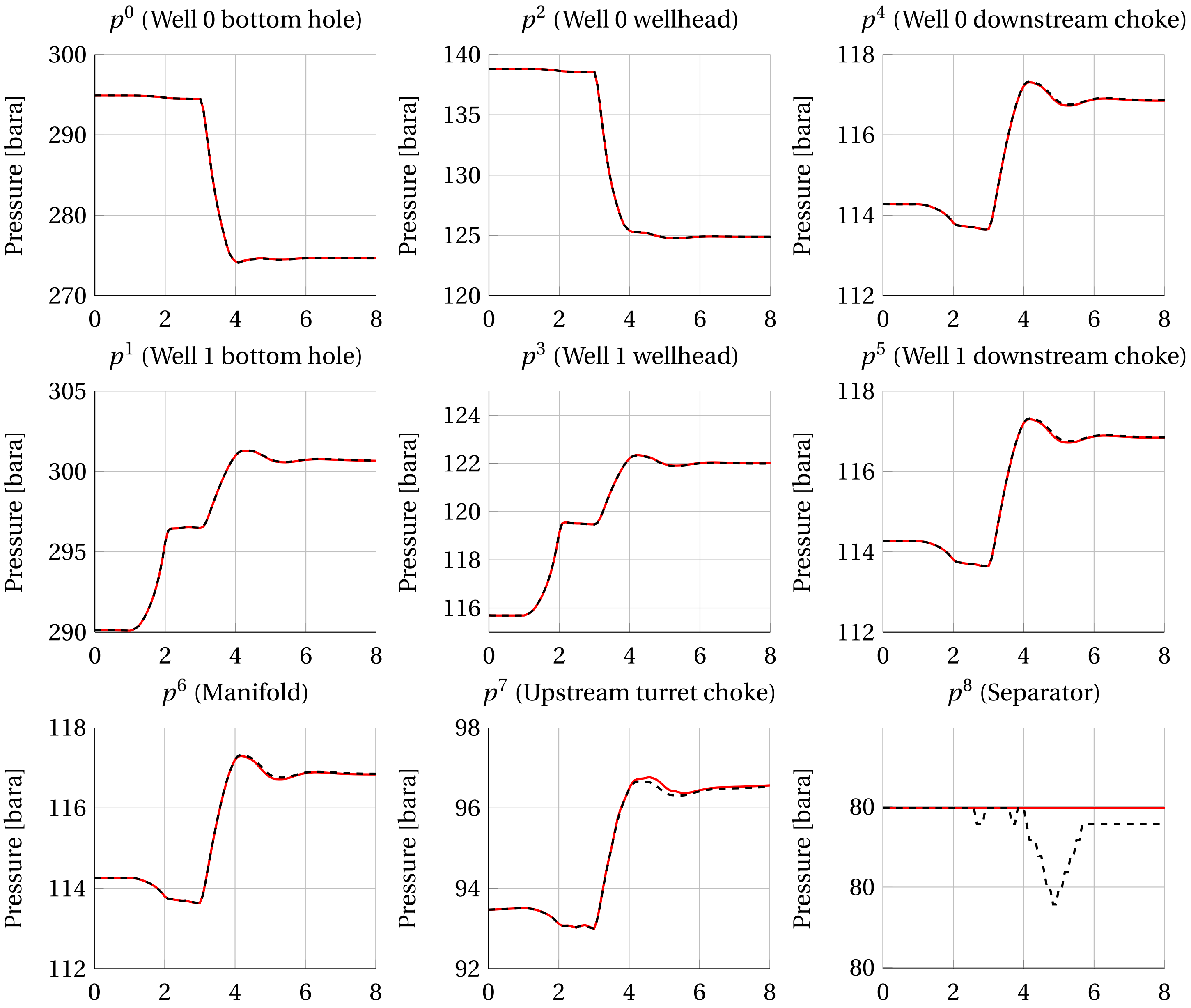


Figure D.6: Pressures for well adjustment case (dynamic estimation). Legend: Red; Measured pressures (from OLGA), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

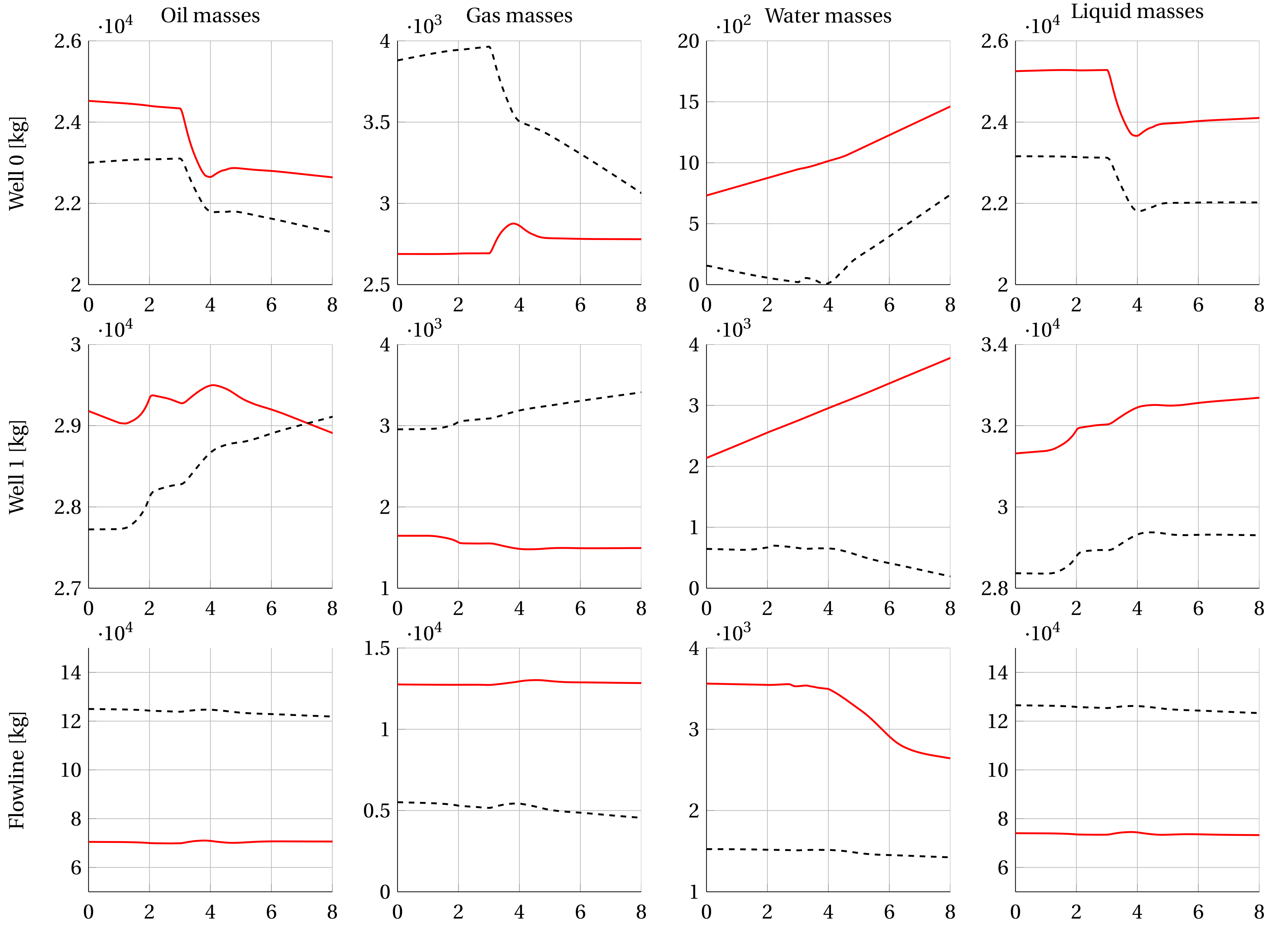


Figure D.7: Masses for well adjustment case (dynamic estimation). Legend: Red; Measured masses (from OLGA), Dashed black; estimated masses. All masses in [kg]. x-axis: Time [h].

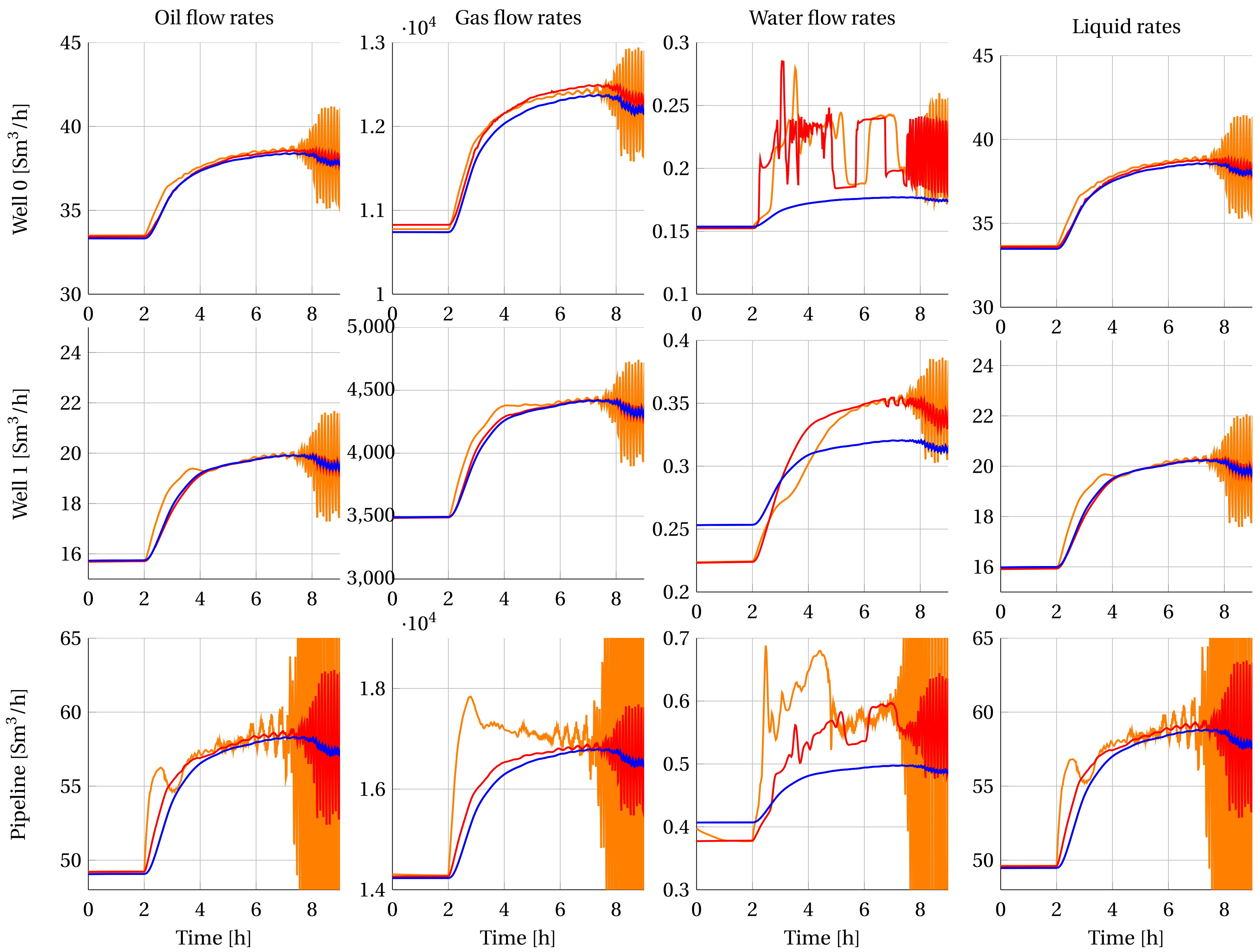


Figure D.8: Flow rates for riser slugging case (static estimation). Legend: Red; Measured inflow rates (from OLGA), Orange; Measured outflow rates (from OLGA), Blue; estimated flow rates. All flow rates in [Sm^3/h].

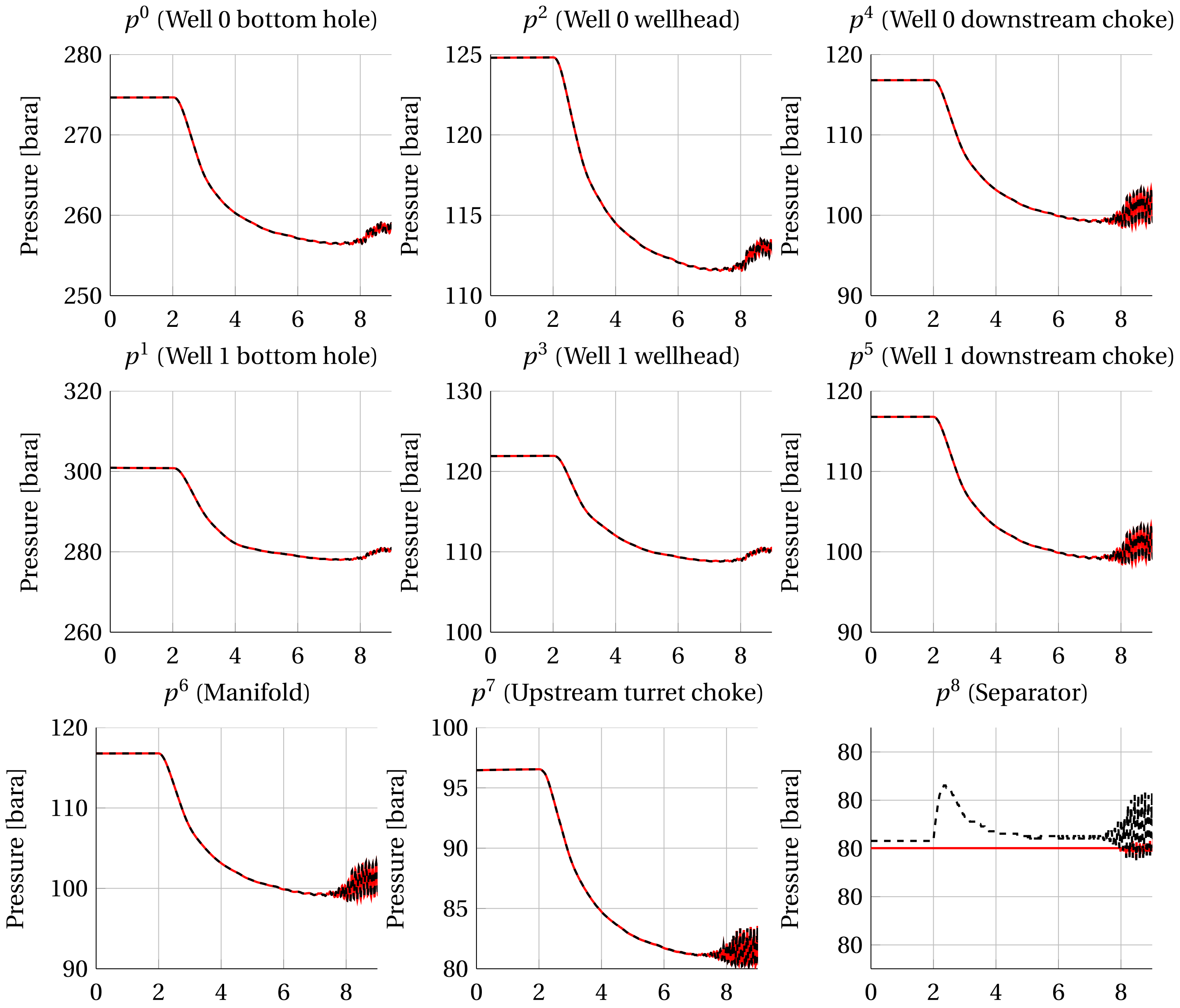


Figure D.9: Pressures for riser slugging case (static estimation). Legend: Red; Measured pressures (from OLGA), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

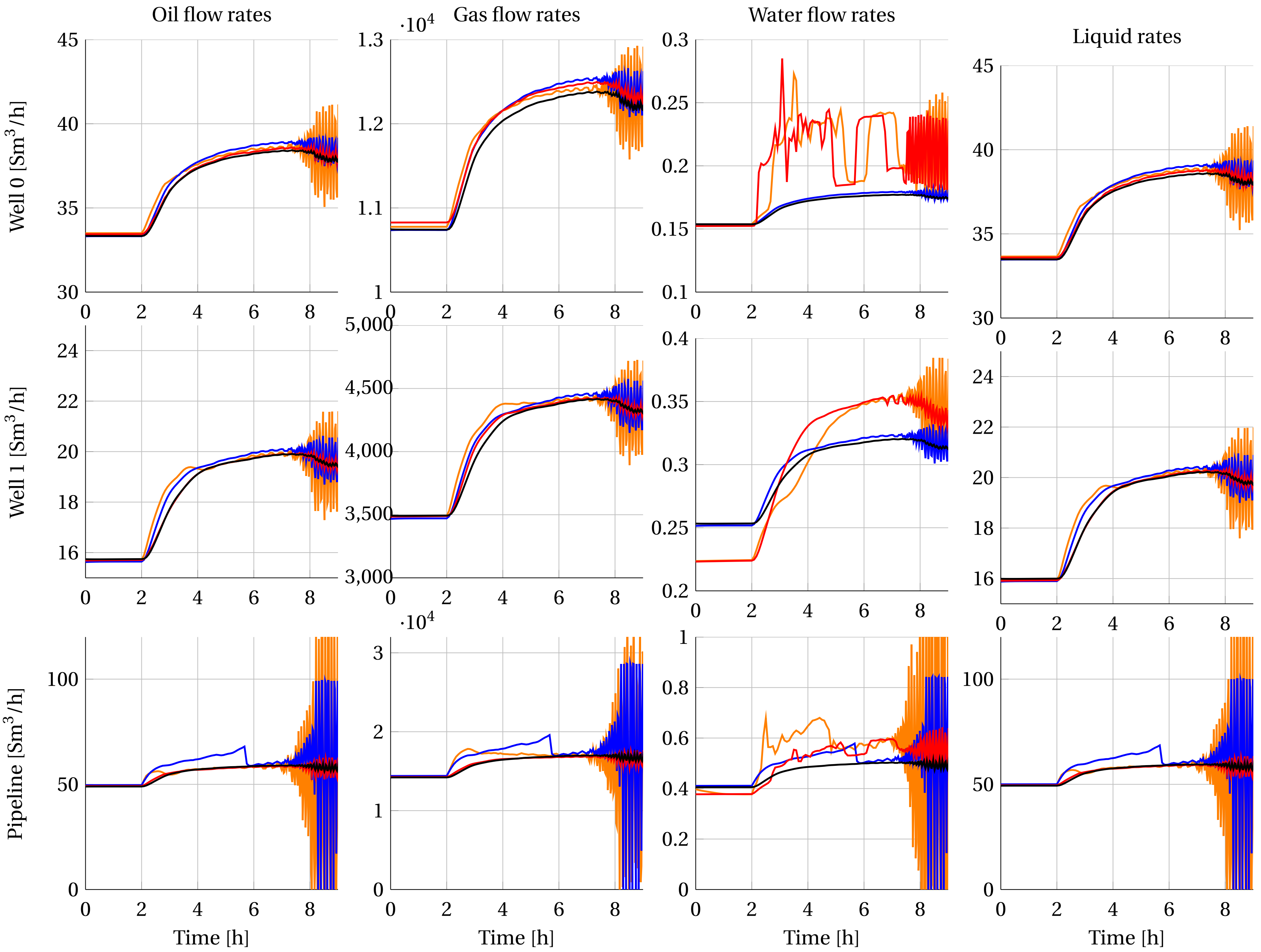


Figure D.10: Flow rates for riser slugging case (dynamic estimation). Legend: Red; Measured inflow rates (from OLGA), Orange; Measured outflow rates (from OLGA), Black; estimated inflow rates, Blue; estimated outflow rates. All flow rates in [Sm^3/h].

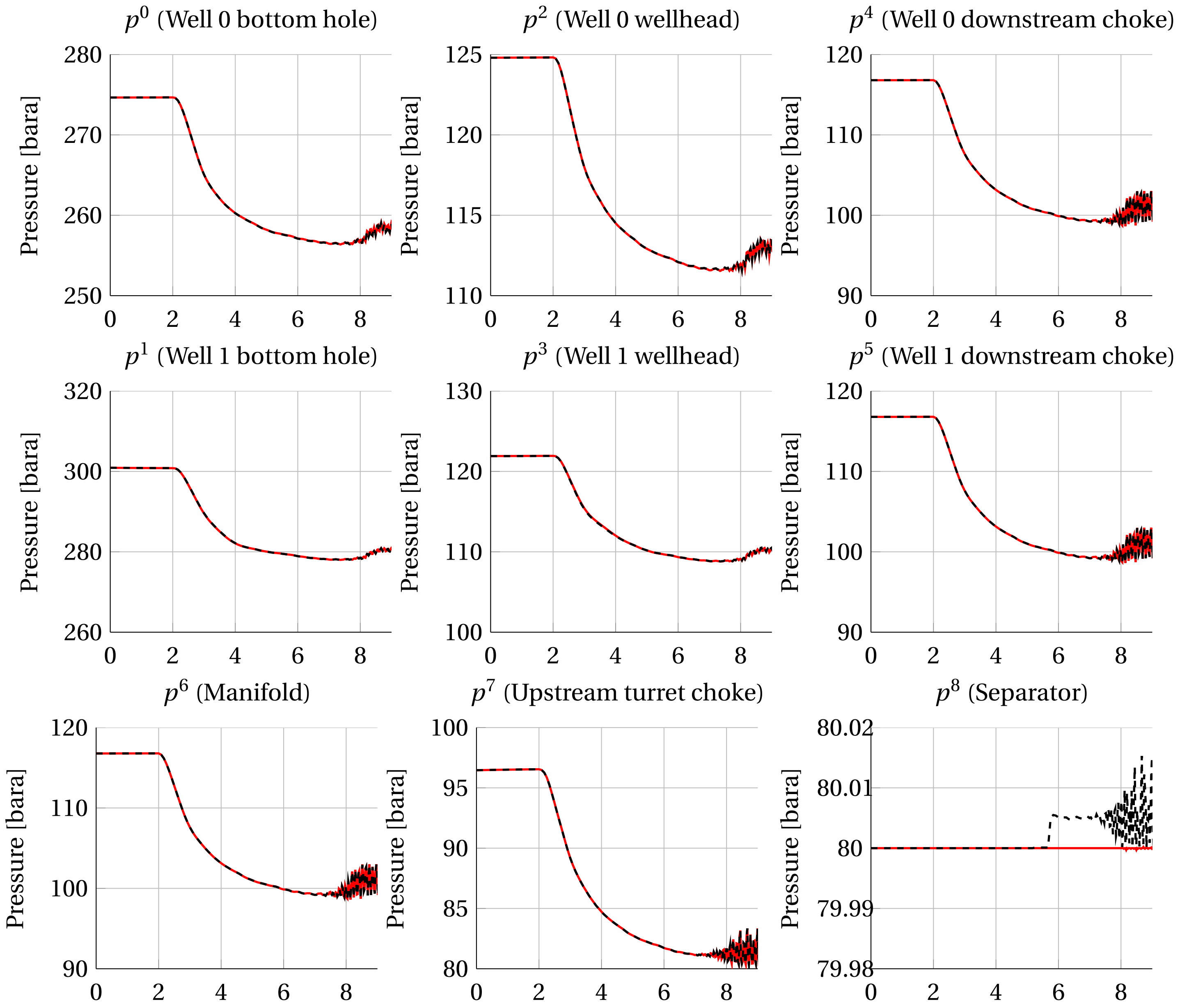


Figure D.11: Pressures for riser slugging case (dynamic estimation). Legend: Red; Measured pressures (from OLGA), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

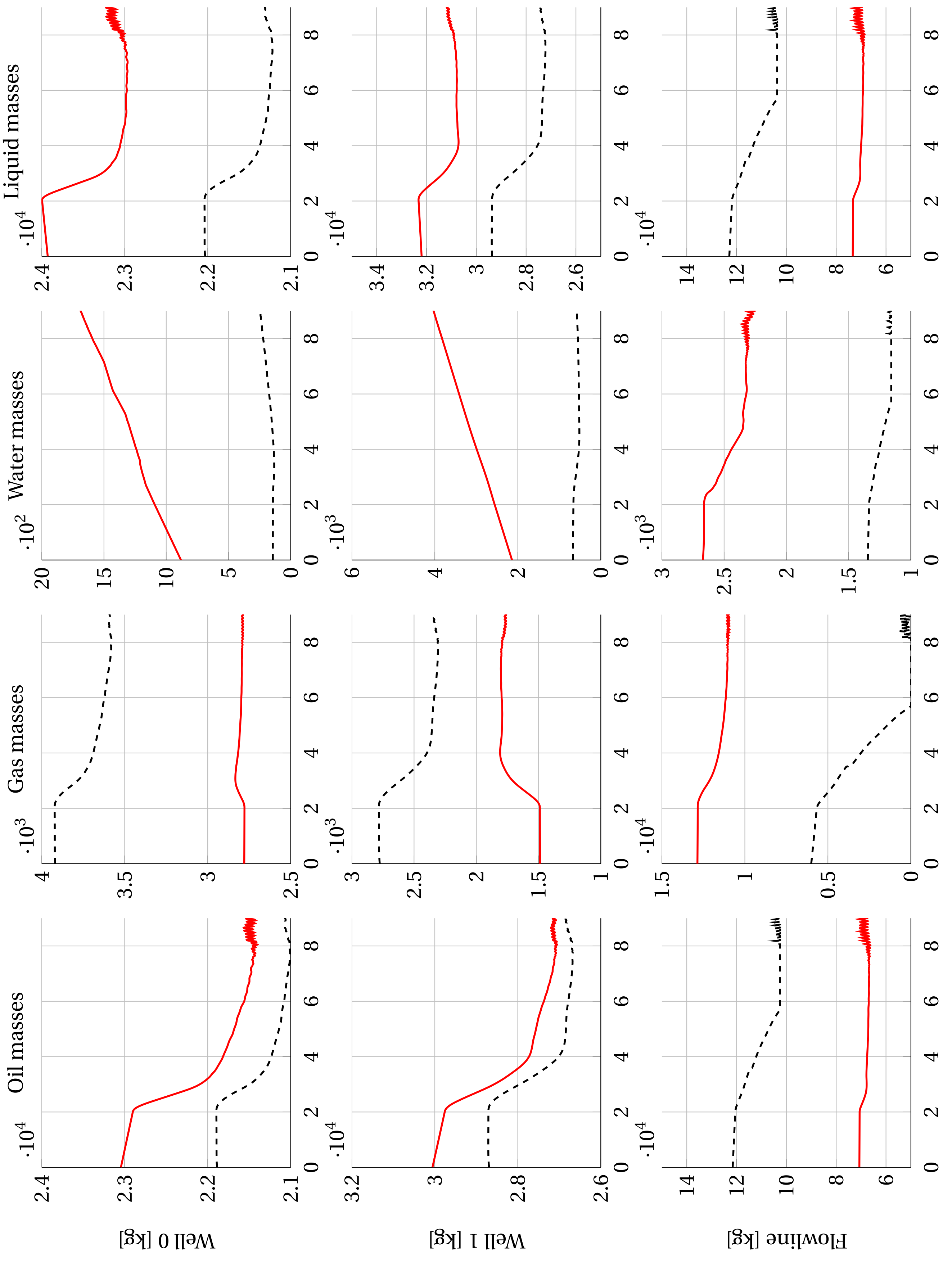


Figure D.12: Masses for riser slugging case (dynamic estimation). Legend: Red; Measured masses (from OLGA), Dashed black; estimated masses. All masses in [kg]. *x*-axis: Time [h].

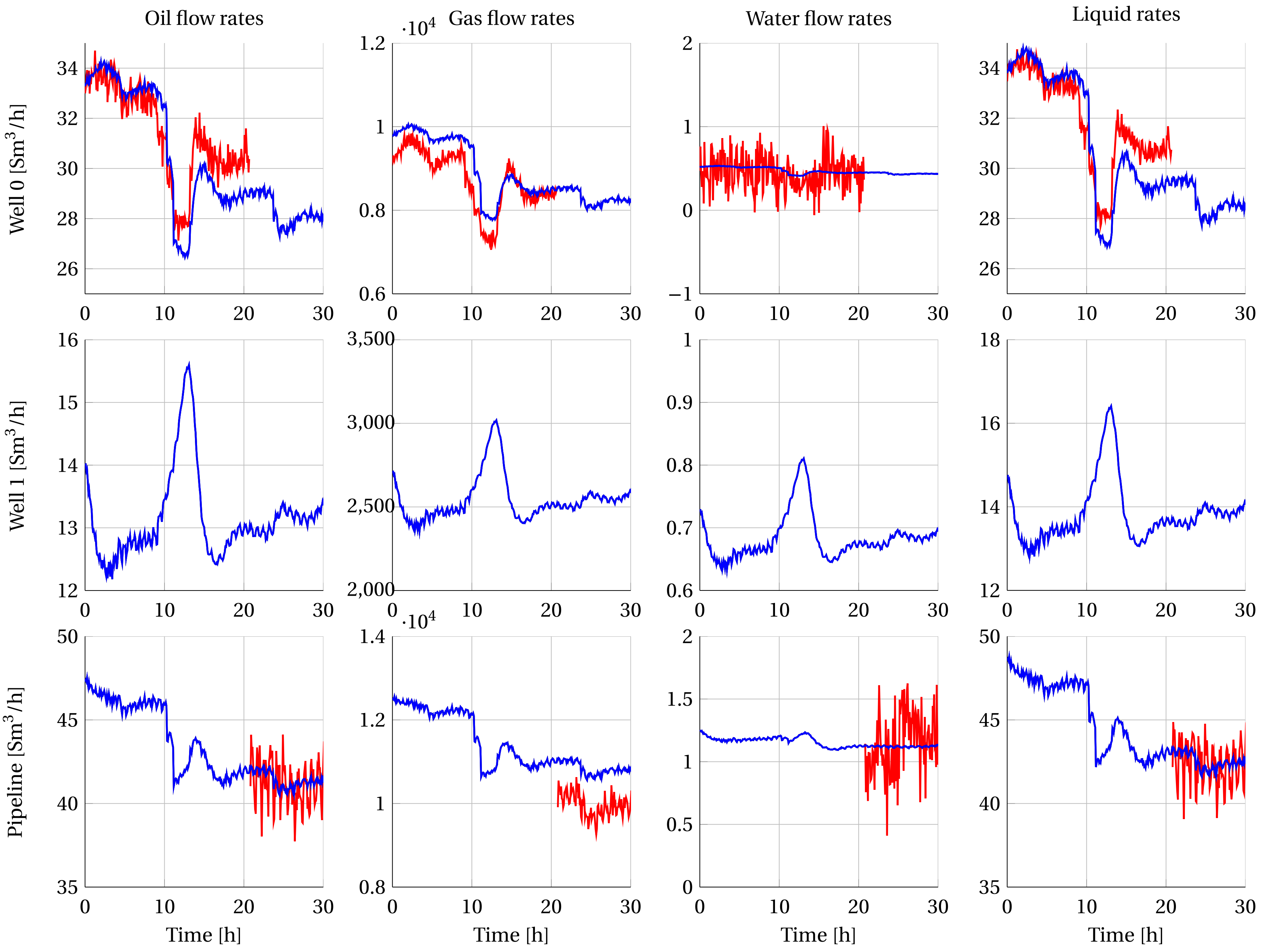


Figure D.13: Flow rates for field data case (static estimation). Legend: Red; Measured flow rates (from MPFM), Blue; estimated flow rates. Note that from time 0 to about 20 h, well A is routed through the MPFM. After this, both wells are routed through the MPFM. All flow rates in [Sm^3/h].

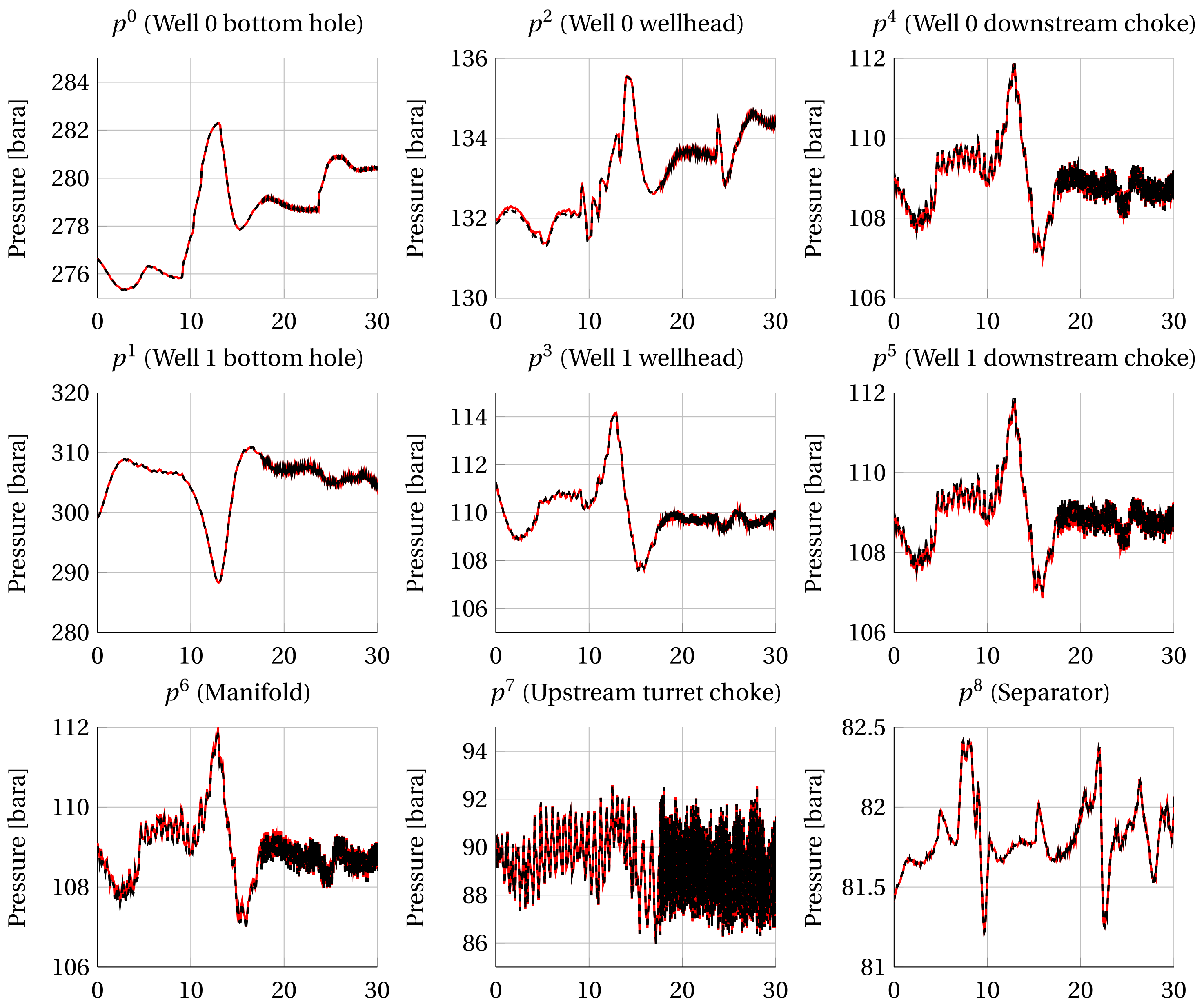


Figure D.14: Pressures for field data case (static estimation). Legend: Red; Measured pressures (from pressure transmitters), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

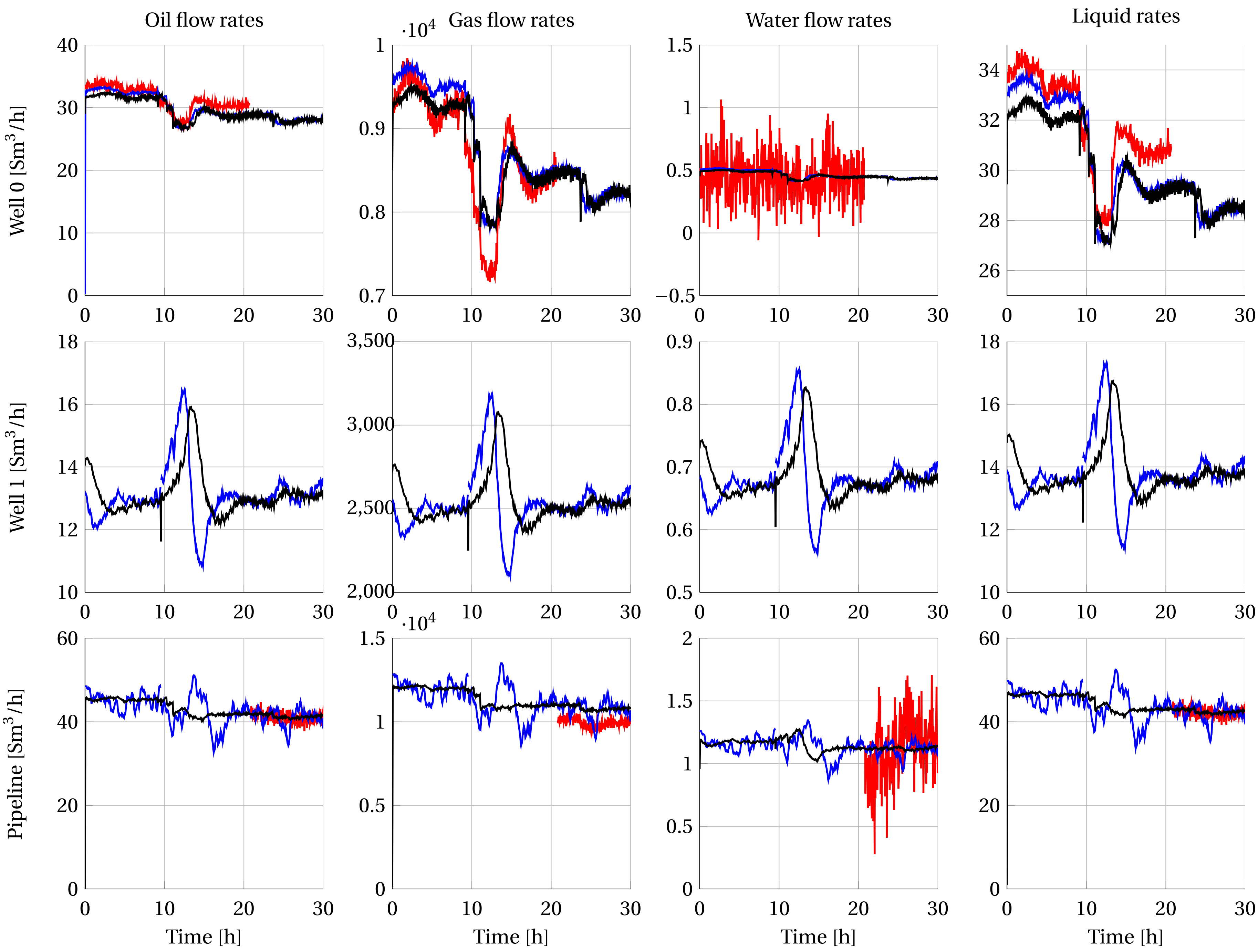


Figure D.15: Flow rates for field data case (dynamic estimation). Legend: Red; Measured flow rates (from MPFM), Black; estimated inflow rates, Blue; estimated outflow rates. Note that from time 0 to about 20 h, well A is routed through the MPFM. After this, both wells are routed through the MPFM. All flow rates in [Sm^3/h].

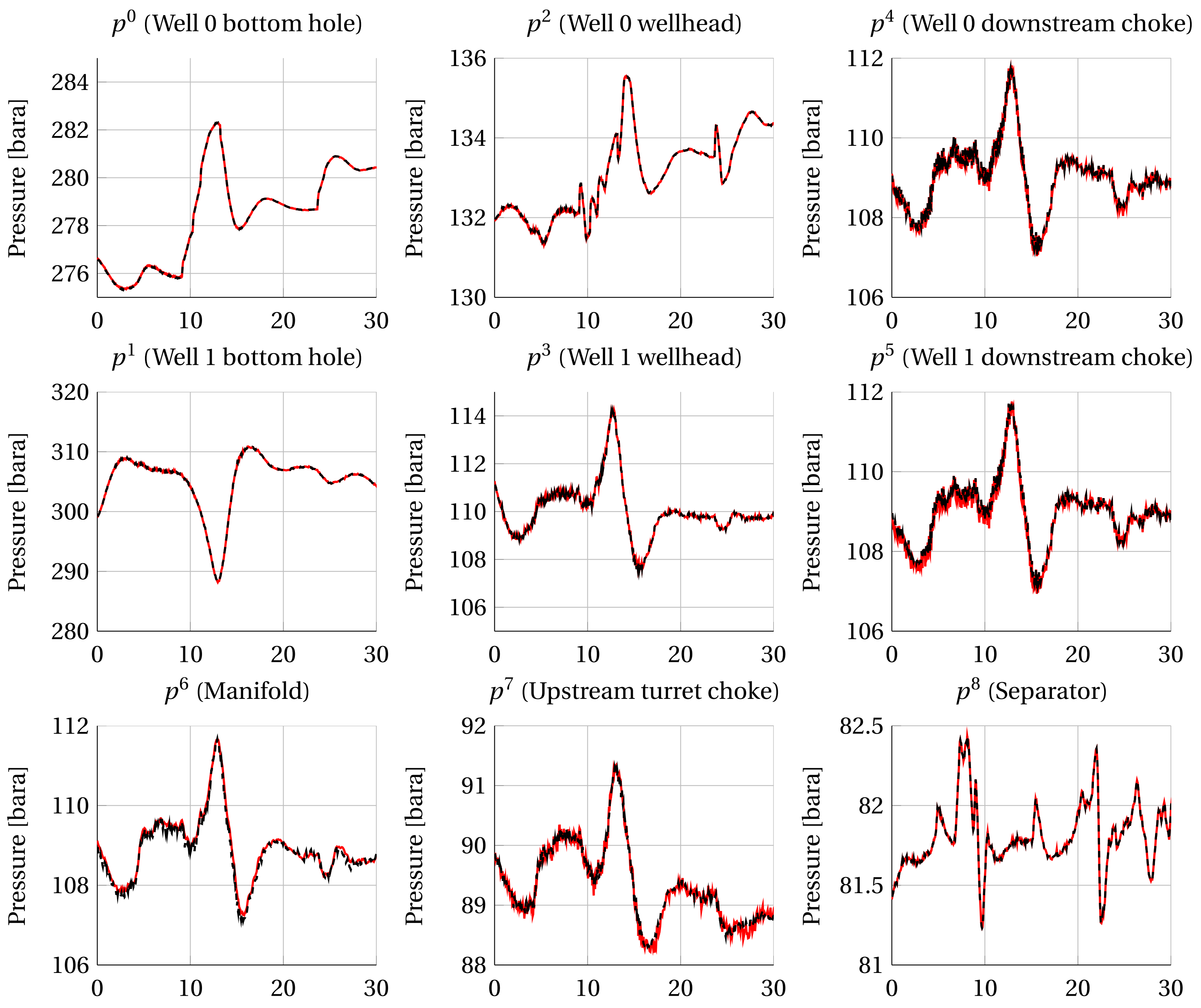


Figure D.16: Pressures for field data case (dynamic estimation). Legend: Red; Measured pressures (from pressure transmitters), Dashed black; reconciled (estimated) pressures. All pressures in [bara]. x-axis: Time [h].

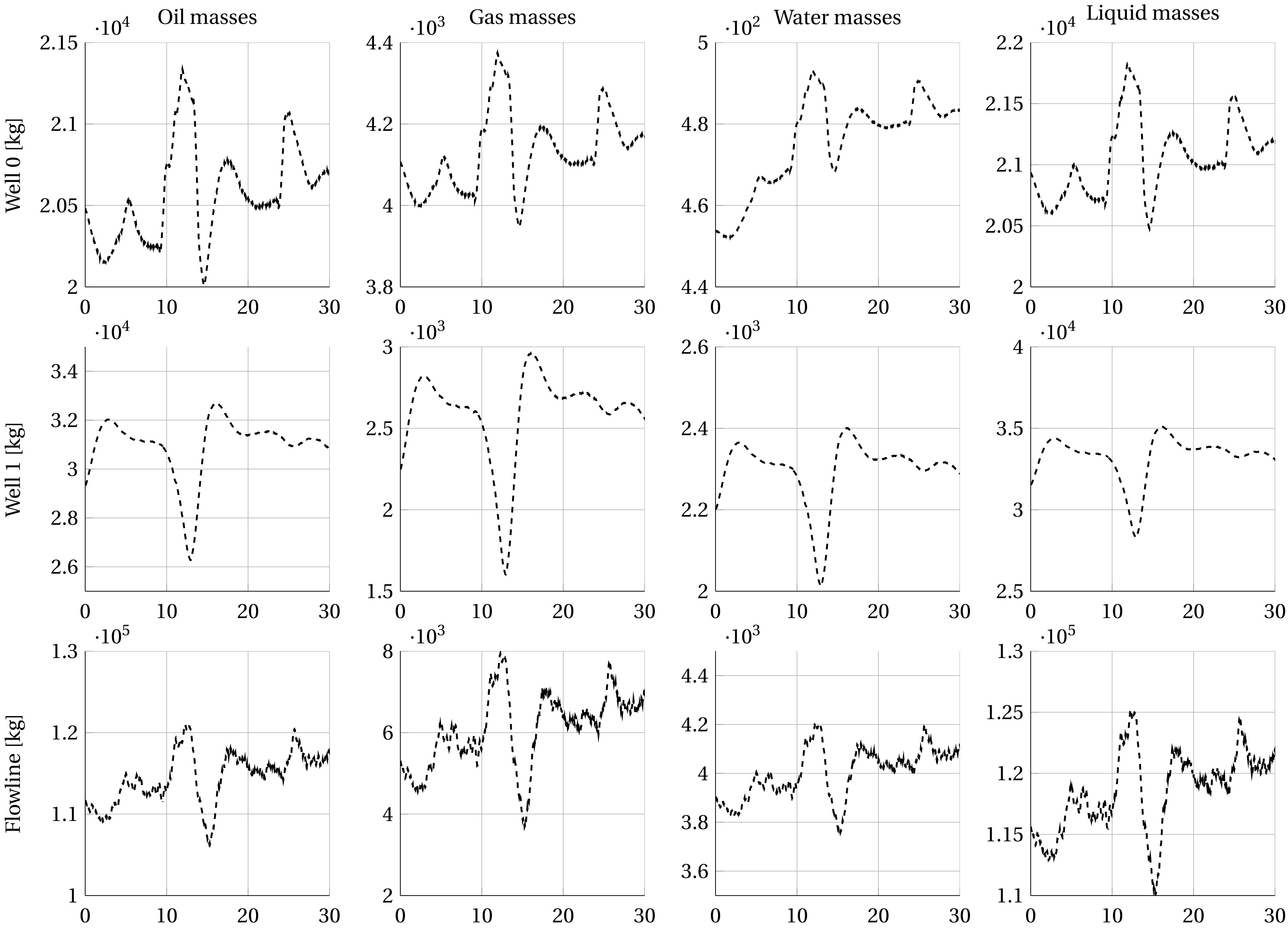


Figure D.17: Masses for field data case (dynamic estimation). Legend: Dashed black; estimated masses. All masses in [kg]. x-axis: Time [h].