



NTNU – Trondheim
Norwegian University of
Science and Technology

Automatic Detection of Poorly Calibrated Models in State Estimation Applied to Oil and Gas Production Systems

Håkon Skibeli

Master of Science in Cybernetics and Robotics

Submission date: June 2015

Supervisor: Bjarne Anton Foss, ITK

Co-supervisor: Bjarne Grimstad, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Automatic detection of poorly calibrated models in state estimation applied to oil and gas production systems

Candidate: Håkon Skibeli, Supervisor: Bjarne Foss, Co-supervisor: Bjarne Grimstad

Department of Engineering Cybernetics (ITK), NTNU, Norway

Project start-end date: 12.01.2015 - 08.06.2015.

June 7, 2015

1 Problem Description

Modern subsea production systems have pressure and temperature sensors installed throughout the network. Flow meters, being more expensive and less reliable, are less frequently installed. The modern subsea production system is monitored by feeding these measurements to a virtual flow metering system that estimates the flows. Flow estimation is the enabling technology for advanced monitoring systems such as flow assurance systems and condition performance systems.

The performance of a virtual flow metering system is highly correlated with the amount of uncertainty in the system. The uncertainty is normally categorized as measurement noise, process disturbances, and model error. In subsea systems, engineers are mostly concerned with measurement noise and model errors. As measurement noise (and sensor failure) are easy for a human to characterize, model error is not. Unfortunately, models are infrequently calibrated due to the expense of well testing. Thus, there is considerable uncertainty in the models (which increases with time since the last well test). In any virtual flow metering system it would be highly beneficial if poorly calibrated models were identified and weighed less in the estimation problem.

1.1 Estimation

Consider the following flow estimation problem formulation. Let $\mathbf{y} \in \mathbb{R}^{n_y}$ be a vector of variables to be reconciled with corresponding measurements $\bar{\mathbf{y}} \in \mathbb{R}^{n_y}$. We denote the difference between the reconciled and measured values with \mathbf{v} , i.e. $\mathbf{v} = \mathbf{y} - \bar{\mathbf{y}}$. Furthermore, we denote with $\mathbf{x} \in \mathbb{R}^{n_x}$ the unmeasured variables that we want to estimate. To estimate \mathbf{x} we solve the following nonlinear programming problem

$$\begin{aligned} & \underset{\mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{w}}{\text{minimize}} && \|\mathbf{v}\|_{\mathbf{M}}^2 + \|\mathbf{w}\|_{\mathbf{N}}^2 \\ & \text{subject to} && \mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{w} \\ & && \mathbf{y} - \bar{\mathbf{y}} = \mathbf{v} \\ & && \mathbf{x} \in \mathbf{X} \end{aligned} \tag{P}$$

where $\mathbf{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^m$ are m maps between the reconciled (measured) variables y and the unmeasured variables \mathbf{x} ; for example, \mathbf{g} may contain difference equations resulting from discretization of a continuous model. In general, \mathbf{g} is a vector of nonlinear functions and, hence, $\mathbf{g}(\cdot) = \mathbf{w}$ describes a nonconvex constraint set. The variables $\mathbf{w} \in \mathbb{R}^m$ represent model errors; in the case of a perfect model $\mathbf{w} = \mathbf{0}$. The set \mathbf{X} is a convex polytope which may include linear constraints on the estimated variables \mathbf{x} . Note that the measurements $\bar{\mathbf{y}}$ are not considered variables in **P**. The vectors \mathbf{v} and \mathbf{w} are called residuals.

The objective of \mathbf{P} is a weighted least-squares quadratic function defined by the norms $\|\mathbf{v}\|_{\mathbf{M}}^2 = \mathbf{v}^T \mathbf{M} \mathbf{v}$ and $\|\mathbf{w}\|_{\mathbf{N}}^2 = \mathbf{w}^T \mathbf{N} \mathbf{w}$, representing penalties on measurement and modelling errors, respectively. The matrices \mathbf{M} and \mathbf{N} can be thought of as the inverse covariance matrices for the measurement noise and model errors. Note that in a moving horizon estimation (MHE) formulation the objective is typically augmented with an *arrival cost* term [5].

1.2 Fault detection

Statistical methods may be used on the residuals \mathbf{v} and \mathbf{w} from \mathbf{P} to detect “faults” (sensor or model errors). These methods are usually formulated as an hypothesis test, with predefined thresholds determining the outcome of the test. One such test is the generalized likelihood ratio (GLR) test. Refer to [3, 2] for more information on fault detection algorithms.

1.3 Modelling

Consider a model of a subsea production systems with four topside wells connected to a separator – modelled as in [1]. Assume that pressure (and temperature) sensors are located in the downhole and wellhead of each well. In addition, pressure and production rate of all phases (fluid components) are measured at the separator. Realistic characterizations of the measurement uncertainty can be found in the literature [4, 6].

2 Assignment

The candidate is given the following assignments:

- Perform a literature study on state estimation with focus on moving horizon estimation for nonlinear systems.
- Perform a brief literature study on fault detection and in particular hypothesis testing using the generalized likelihood ratio test.
- Model the production system previously described. Characterize the measurement noise of the sensors. Specify 2-3 test cases where one or several of the well models change during the case to induce model error (for example by changing the oil density parameter).
- Implement a MHE scheme for the production system model. Test it on the specified cases. Discuss how the model error affects the estimator performance.
- Implement the preferred fault detection algorithm (for example GLR) for testing the hypothesis that the models are accurate (or contrarily, not accurate). At selected time instances, run the algorithm on the residuals resulting from the MHE to identify poorly calibrated well models. Use the attained information to calibrate the model in \mathbf{P} . Assert the change in estimator performance with and without fault detection. Comment on the difficulty of using algorithms for hypothesis testing.

References

- [1] Ole M Aamo, GO Eikrem, HB Siahhan, and Bjarne A Foss. Observer design for multi-phase flow in vertical pipes with gas-lift—theory and experiments. *Journal of process control*, 15(3):247–257, 2005.
- [2] Mogens Blanke and Jochen Schröder. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.
- [3] Rolf Isermann. Model-based fault-detection and diagnosis—status and applications. *Annual Reviews in control*, 29(1):71–85, 2005.
- [4] Béla G Lipták. *Instrument Engineers' Handbook, Volume One: Process Measurement and Analysis*, volume 1. CRC press, 2003.
- [5] Christopher V Rao, James B Rawlings, and David Q Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *Automatic Control, IEEE Transactions on*, 48(2):246–258, 2003.
- [6] John G Webster and Halit Eren. *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*, volume 1. CRC press, 2014.

Abstract

In modern oil and gas industry, there is an increasing use of instrumentation. This lead to a huge flow of information, which typically is not utilized to its full potential. By the use of increasingly more complex Virtual Flow Metering (**VFM**) solutions, the gap between the amount of data available and the amount of data utilized is reduced. VFM can contribute to operational awareness and increased efficiency of the operations, which are qualities that are becoming more and more important for the operators. The performance of a virtual flow meter is, however, highly correlated with the accuracy of the system models. Unfortunately these system models are infrequently calibrated, and with increasingly more complex VFM models, these calibrations become more and more difficult to perform. To further develop the field of VFM, this process of maintaining the models needs to be improved. In this thesis, a system that can potentially simplify this process is investigated.

This thesis presents an investigation on the use of state of the art fault diagnosis techniques, to detect and identify poorly calibrated models used in virtual flow metering. With the help of the information gathered from the fault diagnosis, an operator can potentially pinpoint when and where maintenance of the model is needed. If successful this can lead to a shift from recalibrating the entire system, to focusing on the parts of the model that have been determined as weak links. By going straight for the weak link, the operator can save substantial amounts of time and money, while the reliability of the system models is simultaneously increasing. The investigation has been conducted by running test scenarios on a simulator created during this thesis. The tests on this simulator were performed using state of the art virtual flow meters, together with the fault diagnosis tools developed in this thesis.

The results show that this idea has good potential, and should be further investigated. The results showed this by successfully detecting and identifying poorly calibrated models for simple test scenarios. That bein said, both of the developed fault diagnosis systems, still have some drawbacks and unresolved issues, which makes them less suited for real applications. Several suggestions are, however, posted on how these issues can be resolved, and recommendations are given regarding the direction of future investigations.

Sammendrag

I dagens olje og gass industri, ser man en økt bruk av instrumentering. Dette gir en stor strøm av informasjon, som ofte ikke blir brukt til sitt fulle potensiale. Ved bruk av Virtual Flow Metering (VFM) løsninger kan noe av denne forskjellen mellom data tilgjengelig og data brukt bli minket. VFM kan bidra til oversikt og økt produktivitet, som er egenskaper med større og større betydning for operatørene av produksjonssystemene. Ytelsen til VFM er dog nært knyttet til kvaliteten på systemmodellene. Disse modellene er dessverre sjelden kalibrert, og ettersom VFM løsningene blir mer komplekse, blir også kalibreringen vanskeligere å gjennomføre. For å videreutvikle feltet VFM, så må metoder bli funnet for å effektivisere denne prosessen. Denne oppgaven ser nærmere på dette problemet.

Opgaven presenterer forskning på bruken av moderne feildiagnostikk verktøy for deteksjon og identifisering av svakt kalibrerte deler av modellene brukt i VFM. Ved hjelp av informasjonen samlet av feildiagnostikk verktøyet, så kan operatøren potensielt plassere når og hvor vedlikehold av modellen bør gjøres. Sett at dette er vellykket, så kan dette lede til ett skifte fra dagens situasjon, der man gjerne gjør en full systemrekalibrasjon, til å fokusere på de delene av modellen som er påvist å være dårlig kalibrert. På det viset spares både tid og penger for operatøren, i tillegg til å gi økt tillit til VFM løsningen. Forskningen i denne oppgaven er utført ved å kjøre tester på en simulator med moderne VFM løsninger og utviklede feildiagnostikkverktøy.

Resultatene viser at dette er et felt som viser godt potensiale, og bør forskes videre på. Resultatene viser dette fordi, deteksjon og identifikasjon ble vellykket gjennomført, for enkle scenarier. De implementerte verktøyene, har i dag flere problemer og svakheter som ville gjort dem mindre egnet for bruk i en praktisk setting. Når det er sagt, er det ingen grunn til å tro at disse problemene ikke kan bli løst, og mange forslag til hvordan dette kan gjøres samt forslag til videre arbeid innen feltet har blitt presentert i denne oppgaven.

Preface

This paper is written as the master thesis at Engineering Cybernetics, NTNU. In total, 21 working weeks has been spent on this thesis, from early January until early June 2015. A paper to be published at an international conference is also under planning. This paper would compactly present the preliminary results of this thesis.

I would like to give a special thanks to my co-supervisor Bjarne Grimstad, who constructed this assignment, and has provided me with valuable input and guidance throughout the work with this master thesis. I would also like to thank my supervisor Professor Bjarne Foss. A special thanks goes out to everyone who has helped and supported me during this thesis work. In particular, I would like to thank my classmates at masters office GG44, my father and my girlfriend, for helping me at various stages during this semester.

Table of Contents

Summary	i
Sammendrag	ii
Preface	iii
Table of Contents	vii
List of Tables	ix
List of Figures	xii
Abbreviations	xiii
1 Introduction	1
1.1 Background	1
1.1.1 Oil and gas production, control and monitoring	1
1.1.2 Virtual flow metering	3
1.2 Motivation and research objective	4
1.3 Scope and interpretation of the assignment	5
1.4 Presentation of the system	5
1.5 Structure of the thesis	8
2 Literature Review	9
2.1 State estimation	9
2.1.1 Kalman filtering	10
2.1.2 Moving horizon estimation	13
2.1.3 Comparison of Kalman and MHE	16
2.2 Fault diagnosis	17
2.2.1 Introduction to fault diagnosis	18
2.2.2 Fault detection using GLRT	20
2.2.3 Fault identification	24

2.3	Measurement error, sensitivity and its characteristics	27
2.3.1	Nomenclature in the field of measurement error	27
2.3.2	The elements of measurement error	28
2.3.3	Typical uncertainties of measurement sensors in the market today	31
3	Modeling	33
3.1	Modeling of the system	33
3.1.1	System equations	34
3.1.2	Measurement noise.	38
3.2	Implementation	43
3.2.1	Validation of the model	45
4	Estimation	49
4.1	Extended Kalman filter equation	49
4.2	Moving horizon estimation	51
4.2.1	MHE with slack for model error	51
4.3	Moving horizon estimation with parameter estimation	53
4.3.1	The choice of MHE parameters	54
5	Fault Diagnosis	59
5.1	Fault detection	60
5.2	Implementation of fault diagnosis using \mathbf{w} as test-variables	62
5.2.1	Implementation of the identification signatures	62
5.3	Implementation of fault diagnosis using \mathbf{p} as test-variables	64
5.3.1	Determining a good value for the threshold (h)	65
6	Numerical Study	67
6.1	Presentation of the test cases	67
6.1.1	Fault diagnosis using \mathbf{w} as test-variables	68
6.1.2	Fault diagnosis using \mathbf{p} as test-variables	70
6.2	Results	74
6.2.1	Fault diagnosis using \mathbf{w} as test-variables	74
6.2.2	Fault diagnosis using \mathbf{p} as test-variables	75
6.3	Runtime analysis	83
7	Discussion	85
7.1	Estimation	85
7.2	Fault diagnosis	86
7.2.1	Fault diagnosis using \mathbf{w} as test-variables, discussion of the test cases	87
7.2.2	Fault diagnosis using \mathbf{p} as test-variables, discussion of the test cases	88
7.2.3	Fault diagnosis, discussion of the most important discoveries	91
8	Conclusion	93
8.1	Future work	94
	Bibliography	95

A	General background theory	101
A.1	Reservoir fluid composition	101
A.1.1	Properties of hydrocarbon fluids	101
B	Implementation of estimators, supplement	103
B.1	Moving horizon estimator implementation	103
C	Additional Plots	105
C.1	Validation of simulator model	105
C.2	GLRT with parameters as test variable, results	106
D	Matlab scripts and functions	109

List of Tables

2.1	Some nonlinear Kalman filters, as presented in Brown and Hwang (2012)	12
2.2	Pros and cons EKF and MHE	17
2.3	Various approaches for presenting accuracy (RDP group, 2005)	29
2.4	Measurement accuracy from instrument data sheets	32
3.1	Variables in the model	35
3.2	Std deviation of real measurement noise	40
3.3	Main results from Section 3.1.2	40
3.4	Parameters used in the simulations	43
3.5	Variables chosen for steady state validation of well model	46
3.6	Steady state validation	46
3.7	Dynamical Validation table	47
4.1	Explanation of variable in the MHE formulation	52
5.1	Basis for the Γ matrix (in the fault identification algorithm)	63
5.2	Basis for the weighting matrix K_D (in the fault identification algorithm)	63
5.3	Chosen values for σ , based on the analysis from Figure 5.3	65
5.4	Distribution parameters for $g(k)$	66
5.5	Selection of h given various P_{FA}	66
6.1	Presentation of the test cases performed in Chapter 6	68
6.2	Test case 1: parameters	69
6.3	Test case 2: parameters	71
6.4	Test case 3: parameters	72
6.5	Test case 4: parameters	73
6.6	A short summary of the results presented in Section 6.2.	74

List of Figures

1.1	An illustration of the three main segments of the petroleum chain	2
1.2	Multi-level control hierarchy (Foss, 2012).	2
1.3	Illustration of the production system	7
2.1	Block diagram of a normal system with state observer.	10
2.2	Simple illustration of a basic Kalman filter (Brown and Hwang, 2012).	11
2.3	Illustration of the effect of nonlinearity on the distribution of $f(x)$	13
2.4	Arrival cost illustration	15
2.5	Simple illustration of a normal fault diagnosis system (Blanke et al., 2006).	18
2.6	Illustrations, choice of threshold h , for GLRT	22
2.7	Illustration of example 1, smearing	26
2.8	Illustration of the fault identification algorithm	27
2.9	The components of measurement error	30
2.10	Illustration of error from nonlinearity	31
3.1	Simplified illustration of one well	35
3.2	Measurement noise analysis, main figure	39
3.3	Measurement noise multi phase flow measurements	41
3.4	Dynamical validation plot	48
4.1	An illustration of how the time constant was determined.	55
4.2	Performance of the state and parameter estimator as a function of N_H	56
4.3	Performance of the state and parameter estimator as a function of μ_p	56
4.4	Performance of the state and parameter estimator as a function of μ	57
5.1	Illustration of the setup for the data flow used in most of this thesis	60
5.2	A schematic illustrations of the steps in the fault diagnosis system used in this thesis.	62
5.3	Distribution of the parameter estimation test-variable	64
5.4	Distribution of the test statistic $g(k)$, in GLRT	65

6.1	Test case 1: Illustration of input sequence	69
6.2	Test case 2: Illustration of input sequence. $p :=$ the parameter that is being tested.	70
6.3	Test case 2d: Illustration of input sequence, $p :=$ the parameter that is being tested	71
6.4	Test case 3: Illustration of input sequence. $p_{1,2} :=$ the two parameters that is being tested.	72
6.5	Test case 4: Illustration of input sequence	73
6.6	Test case 1: plot	75
6.7	Test case 2a: plot	76
6.8	Test case 2b: plot	76
6.9	Test case 2c: plot	77
6.10	Test case 2c: supplement plot	78
6.11	Test case 2: supplement plot	79
6.12	Test case 2d: plot	80
6.13	Test case 3a: plot	81
6.14	Test case 4: plot	82
6.15	Test case 4: supplement plot	82
6.16	Runtime analysis of the estimators	83
7.1	Global and local minimum	91
A.1	Approximate illustration of the compressibility factor of methane at 25 °C	102
C.1	Dynamical validation from section 3.2.1. This time with change in reservoir pressure.	106
C.2	Test case 3b: Fault in both C_c and GLR parameter of well one. No change in u	107
C.3	Test case 3c: Fault in both C_c and GLR parameter of well one. No change in u	107
D.1	A general description of the Matlab code used in this thesis	110

Abbreviations

Abbreviation = definition

General Oil and Gas, state estimation

EKF = Extended Kalman Filter
MHE = Moving Horizon Estimation
VFM = Virtual Flow Metering
EOR = Enhanced Oil Recovery
GLR = Gas Liquid Ratio
WC = Water Cut
GOR = Gas Oil Ratio

Fault diagnosis and statistics

PDF = Probability Distribution Function
PSD = Power Spectral Density
IID = Independent and Identically Distributed
FDI = Fault Detection and Identification
GLRT = Generalized Likelihood Ratio Test

Introduction

This thesis describes the work done by the author on research for improving the existing virtual flow metering techniques used in the oil and gas industry today. To motivate the research done in this paper, this chapter presents some general background theory in the fields of oil and gas production, and Virtual Flow Metering (**VFM**). Furthermore the chosen scope of the assignment is presented, along with a presentation of the test system used in this thesis, and a simple presentation of the structure of this paper.

1.1 Background

1.1.1 Oil and gas production, control and monitoring

This section is added to give the reader sufficient background information, to fully comprehend the problem that is being addressed in this thesis. It consists of two main elements, namely a presentation of oil and gas production, with its control systems and monitoring aspects, and a short introduction to the field of virtual flow metering. Before discussing the production, and its control and monitoring aspects, a small paragraph will be devoted to placing the production system in the complete oil and gas industry chain. The oil and gas industry can generally be said to consist of three main segments, namely "upstream", "midstream" and "downstream" (Petroleum Services Association of Canada, 2015), as illustrated in figure 1.1.

- **"Upstream"** which consists of the exploration and extraction of the oil from the reservoir. This is the segment that includes the oil and gas production system
- **"Midstream"** which usually is said to be the processing, and potentially storage of the reservoir fluids
- **"Downstream"** part of the industry which consists of, the transportation, final refining of the hydrocarbons, and the marketing of the final product

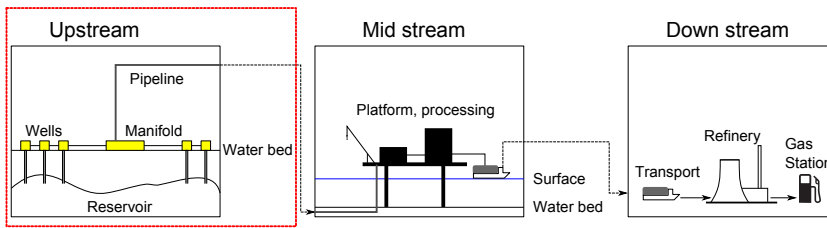


Figure 1.1: An illustration of the three main segments of the petroleum industry chain as given by Petroleum Services Association of Canada (2015). Note that the upstream element is marked in red, as it is this segment of the chain that is of primary interest in this thesis.

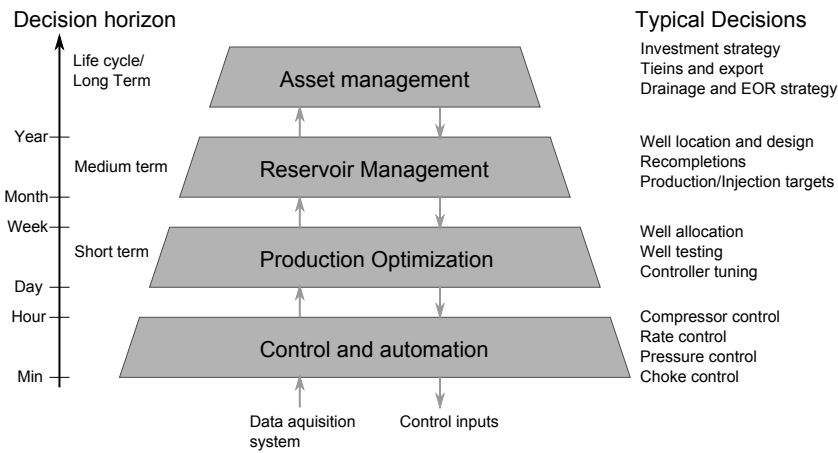


Figure 1.2: Multi-level control hierarchy (Foss, 2012).

Oil and gas production can simply put be explained as the section located between the reservoir and the platform, safely transporting the reservoir fluids from the reservoir to the processing oil platform. This fluid transportation is driven by the large pressure difference between reservoir pressure and platform pressure. For oil fields with decreasing reservoir pressure this transportation can be further assisted by gas or water injection into the reservoir (Gunnerud and Foss, 2010), or by additional gas lift in the wells (Aamo et al., 2005). After the reservoir fluids enter the platform they are separated through a sequence of separators, compressors, scrubbers, etc, to almost clean water, oil and gas.

Operating these production systems can be thought of as an optimization problem – to continuously utilize the maximum capacity of the processing plant, and hence producing the maximum amount of hydrocarbons. The challenge consists of several elements, including

- Planning the layout of the production system, and long term operation strategy
- Allocating good locations for the production and injection wells

- Allocation of production from each well
- Optimally run the processing system

as explained in Foss (2012). This is illustrated in Figure 1.2.

Provided below is a list containing three of the four levels illustrated in Figure 1.2. In this list the importance of knowing the well flow rates is pinpointed at each of these levels.

1. Control and Automation:

This level has the shortest time horizon and consists of several controllers for many different instruments. A common denominator for many of these controls is that the controlled property is the flow rate. For the controller to have any closed loop feedback, this flow needs to be given as information to the controller.

2. Production Optimization:

Is the level that determines the optimal flow rate from each well, and how to set the production valves to achieve this. To properly choose the valve openings, there is a need for knowledge regarding the flow rate in each well with the current valve opening.

3. Reservoir Management:

In this layer it is decided how to place any new wells, that is introduced to the system. This is mostly based on information regarding the reservoir, and the flow of hydrocarbons within it. Intuitively this level does not benefit much from real time estimation or measurement of flow, however as stated in Heddle et al. (2012), this information can contribute to more accurate reservoir simulation models leading to more accurate information for depletion planning.

Furthermore, it should be noted that for the purpose of condition monitoring of the equipment, measurements- or estimations of flow can be a strongly contributing factor (Bringedal et al., 2010).

1.1.2 Virtual flow metering

In the previous section, the importance of knowing the well flow rates was stressed. In this section, different approaches to estimating this flow is presented. Traditionally the flow in the different wells were found through frequent well testing, and the assumption that the conditions remained approximately unchanged between each well test (Melbøet al., 2003). In addition to being poor approximations of the flow, this approach also required temporary interruption of the production which is very costly.

It is also a procedure that becomes increasingly more costly as many of the new fields are connected with existing production facilities through long tiebacks¹. Well tests with a longer tieback are costly, because it requires the reservoir fluid to be transported further, leading to a more time consuming production interruption (Melbøet al., 2003). Another

¹The connection between a new remote cluster of wells, and an existing production facility.

approach is to implement a multi phase flow meter (**MF**M), which has the ability to measure the flow rate subsea with separate measurements for each component (gas, water and oil). However these MFMs are very costly, and prone to numerous faults, and failures that would demand subsea intervention (Bringedal et al., 2010). In addition it is known that the MFMs provide measurements with very poor accuracy. This is also tested in section 3.1.2.

There are numerous extensions to these two approaches and other approaches to get approximations of the flow, however in this paper Virtual Flow Metering (**VFM**) is used (Bringedal and Phillips, 2006). VFM is a technology that estimates the flow based on information provided by the measurements already available in the system. It can range from any kind of model based methods to empirical black box methods (Cramer et al., 2009). VFM has, from the perspective of the operator, low cost, with ease of implementation as there is no need for extra hardware, only software simulations that use the measurements already available (Melbøet al., 2003). A big drawback to VFM, however, is the cost of maintaining an accurate model of the system (Bringedal and Phillips, 2006). This drawback forms the motivation for the extension of VFM investigated in this thesis.

1.2 Motivation and research objective

As stated above, a major drawback with the current VFM systems is the cost of maintaining the models (in terms of time and money) . This maintenance is necessary, as the VFM estimates its flow rates based on measurements and the models that relates these measurements to the flow rates. This maintenance of the models is often conducted by doing well tests, for the purpose of recalibrating the models, which is an approach that is both costly and time consuming. This leads to large costs, or models that are not maintained properly, which ultimately can make the VFM into a flow estimator that can not be relied upon.

In this thesis the theory of fault diagnosis using statistical detection algorithms, is attempted transferred to the field of model validation, by detecting and identifying poorly calibrated models in the VFM. This addition to the VFM can potentially give an added reliability to the VFM, as the algorithm will notify the user when the model no longer seems to be properly calibrated, which implies that unless the user has been notified, one can with a much greater probability assume that the models in the VFM are still valid. Furthermore, if the fault diagnosis can identify which part of the model that seems to be poorly calibrated. It could save the user a lot of time and money, by reducing the need for recalibrating the entire model. The time and money, could instead be focused on recalibrating the parts of the model that are identified to be poorly calibrated. All in all, the goal is that with a working addition like this to VFM, one could improve reliability and accuracy, and hence increase the value added by VFM systems.

Goal of the assignment:

To improve current VFM solutions by introducing complementary and unintrusive software for self-diagnosis via detection – and identification of poor models. These findings will in turn be used to alert the user of the VFM that estimates may be inaccurate and that model calibration should be considered.

1.3 Scope and interpretation of the assignment

The goal of this thesis has been summarized by the table presenting the "Goal of the assignment" in the previous section. In addition a literature study on state estimation related to nonlinear observers, and fault detections are to be conducted. Furthermore a simulator, a MHE and a fault diagnosis algorithm is to be implemented.

For the purpose of keeping the thesis fairly concise, the following scope has been put on the assignment:

- State estimators are implemented for the purpose of working as the VFM in the tests, and to provide the fault diagnosis with necessary test variables for the diagnosis (see section 2.2). This thesis will, however, not provide further development in the field of state estimation for VFM. Tests will for that reason not be conducted with respect to evaluating these estimators and their ability to estimate the states
- The goal of this thesis is to test the possibility – and potential of introducing statistical change detection algorithms as an extension to VFM. For the purpose of testing, different versions of a fault diagnosis algorithm will therefor be created. However the scope of these algorithms are purely for testing of the concept and will not be created as a finished product.
- Any comprehensive tuning of either the state estimators or the fault diagnosis systems are defined to be outside of the scope for this thesis.

In short the scope of this thesis is not to create new and improved Virtual Flow Metering algorithms, nor is it to create a fault diagnosis system to supplement an existing VFM. It is a thesis that investigates the possibilities, difficulties and the different approaches to such an extension of Virtual Flow Metering.

1.4 Presentation of the system

As a test case for this thesis the system illustrated in figure 1.3 was created. It is a simplified model of an oil and gas production system consisting of 4 well heads, and one separator.

To keep the implementation simple, the wellheads are placed topside, removing the need for a subsea manifold. The production system is equipped with pressure- and temperature sensors located as illustrated in Figure 1.3, and flow rate measurements located after the separator. These flow rate measurements can be assumed to be fairly accurate as they are the equivalents to the export measurements on a real platform. The tests on this system will at all times consider the processing/separation of the reservoir fluids in this system to be instantaneous and perfectly executed. Thus removing the need to model the processing system. For the test scenarios, the well chokes will be adjusted to excite the system. Furthermore the model parameters defining the inflow from the reservoir (PI), the Gas to Liquid Ratio (GLR) and valve equation coefficient (C_c), will be artificially changed – creating the desired poor models in the estimators.

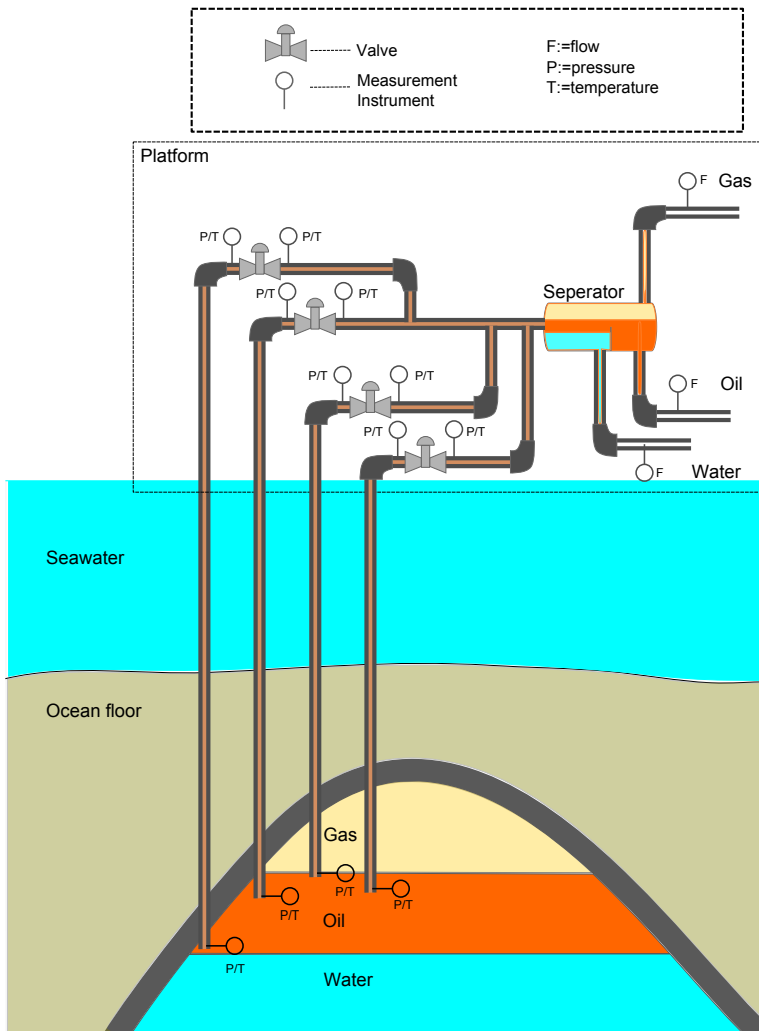


Figure 1.3: Illustration of the production system to be utilized as a test system. 4 wells with topside wellheads and 1 separator.

1.5 Structure of the thesis

To better understand how to go about reading this thesis a short guide is presented in this section. It portrays what you can expect to find in each chapter, the objectives of each chapter and how they are all related.

This thesis begins with the introduction to give some insight into the the problems that are investigated in this thesis, and a short presentation of the test system. **Chapter 2** is a literature study that takes a closer look into the theory already present in the field of state estimation and fault diagnosis. In addition a section on measurement errors, its characteristics and normal accuracies have been added. **Chapter 3** describes the system which is used as a test case scenario, and presents how the mathematical model of the simulator is constructed. The equations used in this simulator are also the same equations used in the state estimators developed in Chapter 4. **Chapter 4** presents the developed estimators in the thesis. **Chapter 5** is the equivalent to Chapter 4 for the development of the fault diagnosis system. **Chapter 6** present the motivation and results of the tests conducted to investigate the implemented state estimators and fault diagnosis system. These results are then analyzed in **Chapter 7**, together with a general discussion of virtual flow metering with the extension of the fault diagnosis system. **Chapter 8** concludes the thesis, and provides suggestions for future work in this field.

In addition to these main chapters, it should be noted that a useful abbreviations table is provided immediately preceding the introduction chapter, and there is an appendix consisting of some general theory that might be useful background information for some parts of the thesis. The appendix does also consist of a short documentation of the code attached with this thesis, and a section that includes the plots that was not found important enough to be included in the actual report.

Literature Review

This chapter presents some of the theory present in the fields of state estimation, fault diagnosis and measurement error. The latter of which can be viewed as a side note to the overall goal of the assignment, but it was included as the measurement errors plays a vital role when estimating the states based on the measurements.

2.1 State estimation

This section presents some of the research in the field of state estimation and specifically nonlinear observers. To limit the scope of this section, only Kalman filter based observers and Moving Horizon Estimators (**MHE**) have been reviewed. It should, however, be noted that there are numerous other alternatives to nonlinear observers, amongst others:

- Lyapunov-based observers (Gauthier et al., 1992)
- Extended Luenberg observers (Zeitz, 1987)
- Techniques based on transforming the nonlinear system to a suitable observer canonical form where the observer design problem can be solved (Bestle and Zeitz, 1983).

For the course of this section, the system that is discussed will be formulated as shown in (2.1).

$$\begin{aligned}x_{k+1} &= f_k(x_k, u_k) + w_k \\ y_k &= h_k(x_k, u_k) + v_k\end{aligned}\tag{2.1}$$

Any linear system would simply be a subcategory of this system, and any continuous system can be discretized into this framework.

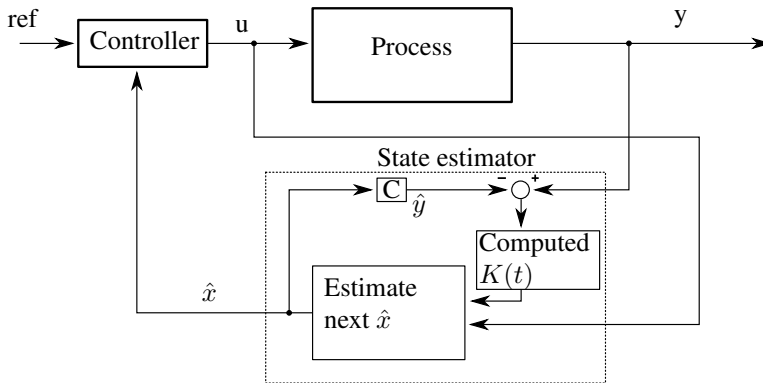


Figure 2.1: Block diagram of a normal system with state observer. The state observer is drawn with a time varying $K(t)$ in mind. This $K(t)$ can be the Kalman gain computed based on the equations in this section.

2.1.1 Kalman filtering

The Kalman filter (Kalman, 1960) is the best known filter for recursive state estimation in the industry today (Kang et al., 2013). It is an optimal recursive Bayesian estimator that has the property of being optimal for linear Gaussian problems. It is built up by a set of recursive equations that results in the optimal "Kalman gain", which theoretically minimizes the mean square error between the real states x and the estimated \hat{x} .

The Kalman filter theory is based on a system of linear equations without inequality constraints, which places itself somewhat outside the scope of this thesis. There are however numerous extensions to the Kalman filter theory, which make it applicable to nonlinear equations. In such cases, the Kalman gain is no longer necessarily optimal, but it still might be a good observer gain. This section will begin with a short introduction of the theory of normal linear Kalman filter – moving onwards to an introduction to the extended Kalman filter (which is applied in this thesis). In Section 2.1.3 the information in this literature review, and in the literature review of MHE will be compared with respect to their strengths and weaknesses.

$$x_{k+1} = A_k x_k + w_k \quad (2.2a)$$

$$y_k = C_k x_k + v_k \quad (2.2b)$$

Given a linear system like (2.2), with measurement noise v_k and process disturbance w_k (both of which, white noise Gaussian signals) that corrupt the signal - one can statistically determine an optimal observer gain $K(t)$ for a state estimator like the one shown in Figure 2.1. This optimal gain is the so called Kalman gain, and it is given in equation 2.3. The derivation of these equations and further information on general Kalman filter can be found

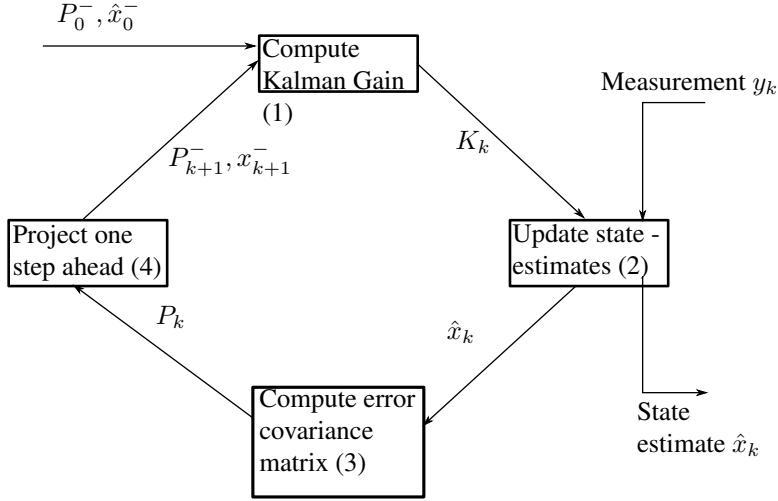


Figure 2.2: Simple illustration of a basic Kalman filter (Brown and Hwang, 2012).

in Brown and Hwang (2012).

$$K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \quad (2.3)$$

$$P_k = (I - K_k C_k) P_k^- \quad (2.4)$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \quad (2.5)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C_k \hat{x}_k^-) \quad (2.6)$$

where

$$Q_k = E [w_k w_k^T]$$

$$R_k = E [v_k v_k^T]$$

$$P_k = E [e_k e_k^T]$$

Note that the elements on the diagonal of P_K forms the squared estimation error

$$e_k^2 = (x_k - \hat{x}_k)^2$$

which is minimized with the Kalman gain. Figure 2.2 illustrates how these equations are combined to form the linear Kalman filter.

For a nonlinear system of equations, no such universally optimal recursive Kalman equations are known. However the theory has been expanded to nonlinear systems by the use of some alterations to the equations. In Table 2.1, 4 different estimators for nonlinear systems based on the Kalman filter has been presented. The first two filters (**Extended** and **Linearized**) are both based on fitting the real nonlinear system into the linear assumption of the Kalman filter. These kind of Kalman filter extension have been widely used, especially the extended Kalman filter, however the result from using the linearized version of the system is strongly dependent on the degree of nonlinearity in the system (Brown and

State estimator	Basic idea
Linearized Kalman filter	Linearize your entire system of equations
Extended Kalman filter	Linearize continuously about the current point in the trajectory.
Ensemble Kalman filter	Estimating a collection(ensemble) of possible state values, and withdraw statistical properties from the collection.
Unscented Kalman filter	Estimating a chosen set of possible state values, and withdraw statistical properties from the collection.

Table 2.1: Some nonlinear Kalman filters, as presented in Brown and Hwang (2012)

Hwang, 2012).

The **Ensemble** and **Unscented** Kalman filter, represents another approach in using the Kalman way of thinking for state estimators. The main difference is that instead of computing the error covariance matrix P_k algebraically, one approximates it based a set of state estimates which is estimated at each iteration (Brown and Hwang, 2012).By doing it this way the computing power needed for the algebraic operation is kept low, which can be important for very large systems. Furthermore the probability distribution of the state vector, is estimated through empirically sampling around the operating point, instead of estimating it based on the assumption of local linearity around the operating point, which can be very misleading.

The extended Kalman filter

"Extended Kalman filter is probably still the most robust and practical approach for most problems" (Kang et al., 2013). This quote was taken from a survey on observers for nonlinear dynamical system – it feels appropriate to discuss the Extended Kalman filter (**EKF**) . In addition to being fairly easy to implement, the EKF can computationally speaking be fairly cheap, especially if the Jacobian of the system is known, relieving the estimator of the task of numerically linearizing the system at each point. That being said, the EKF does have some drawbacks. In areas with a high degree of nonlinearity, state estimation from the EKF will be poor (see Figure 2.3), furthermore convergence can generally not be guaranteed, and for systems that cannot be supplied with analytical derivatives, it is computationally more demanding than some of the alternatives (Kang et al., 2013).

The EKF is as explained above an extension to the Kalman filter which linearizes the system of equations around each iterative estimation \hat{x}_k of the state. This way it approximates local linear behavior around the operating point. With regards to the implementation of the Kalman filter, this extension has no effect on the equations (2.3),(2.4) or (2.5). However,

as the system can no longer be formulated as (2.2) but rather as (2.1), C_k and A_k are no longer available. These matrices are instead approximated through linearization at each iteration as described in (2.7) and for the update of \hat{x}_k in (2.6) $-C_k\hat{x}_k^-$ is replaced with $h(\hat{x}_k^-, u)$. Furthermore, $\hat{x}_{k+1}^- = f(\hat{x}_k^-, u_k)$ instead of $\hat{x}_{k+1}^- = A_k\hat{x}_k^-$

$$A_k = \left. \frac{\partial f_k(x, u)}{\partial x} \right|_{\hat{x}_k^{mh}} \quad (2.7a)$$

$$C_k = \left. \frac{\partial h_k(x, u)}{\partial x} \right|_{\hat{x}_k^{mh}} \quad (2.7b)$$

$$(2.7c)$$

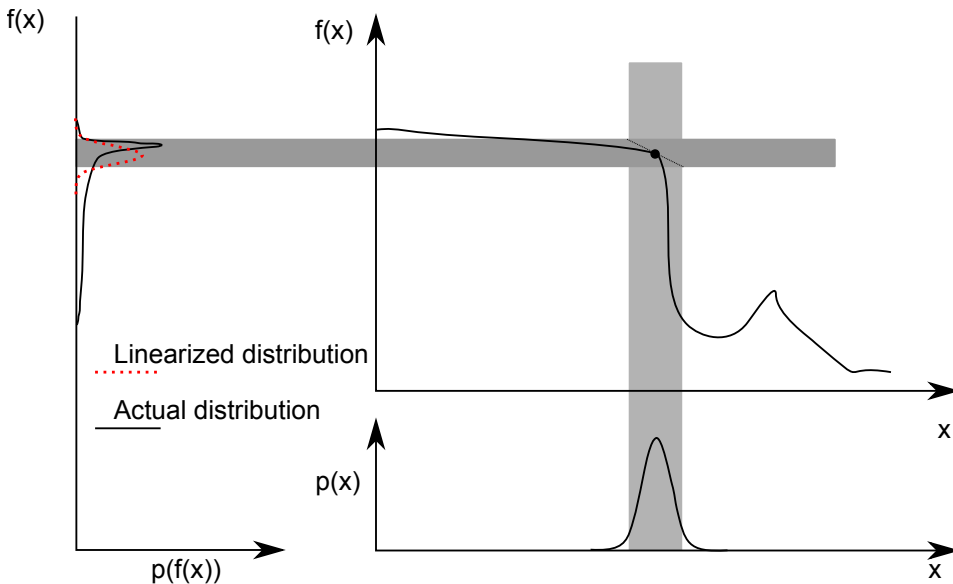


Figure 2.3: Illustration of the effect of nonlinearity on the distribution of $f(x)$. Notice how the distribution of $f(x)$ with the given distribution of x differs depending on whether or not $f(x)$ is linearized. Inspired by Thrun et al. (2005)

2.1.2 Moving horizon estimation

A Moving Horizon Estimator (**MHE**) is an optimization-based observer not unlike the Kalman filter Kang et al. (2013). The similarities, differences and pros and cons, with these two methods are discussed in Section 2.1.3. In this section an introduction to MHE is presented.

General introduction to moving horizon estimation

Moving Horizon Estimation mainly consist of three elements, *stage cost*, *arrival cost* and *constraints*. Where the cost elements form the objective function, and the constraints represents the relations given by the mathematical model, and possibly further constraints that can contribute to restrict the size of the solution space. Together they form the general optimization formulation given in (2.8) (Rao et al., 2003).

$$\Phi_T^* = \min \sum_{k=T-N+1}^T L_k(w_k, v_k) + \mathcal{Z}_{T-N}(z) \quad (2.8a)$$

s.t.

$$x_{k+1} = f_k(x_k, u_k) + w_k \quad (2.8b)$$

$$y_k = h_k(x_k, u_k) + v_k \quad (2.8c)$$

$$x_k \in \mathbb{X}, v_k \in \mathbb{V}, w_k \in \mathbb{W} \quad (2.8d)$$

$$\forall k \in (T - N + 1) \dots T \quad (2.8e)$$

with the $f(x)$ and $h(x)$ stemming from the formulation posted in (2.1).

Stage cost ($L(w_k, v_k)$):

The stage costs (L_w and L_v) are the elements in the objective functions that put cost on any error at each time step that is estimated and measured (Rao et al., 2003). How this stage cost function is chosen can differ, as long as it is positive definite. A normal choice is to choose a weighted sum of square function as the stage cost, which is shown below:

$$L = x^T Q x = \|x\|_Q^2 \quad (2.9)$$

This is also the stage cost function chosen in this project:

$$L_w = \sum_k \|w_k\|_N^2 \text{ and } L_v = \sum_k \|v_k\|_M^2$$

Note: it is this kind of cost that the Kalman filter optimizes as well.

Arrival cost ($\mathcal{Z}(z)$):

Optimally one would want to utilize the information from all the previous measurements in the estimation of the states, which is known as a Full Information Estimator (**FIE**) (Rao et al., 2001). In MHE, however, only a finite set of measurements (y_k) are directly included in the optimization formulation to reduce the computation cost, and improve the scalability of the estimator (Rao et al., 2001). Arrival cost¹ is the element that is added to the objective function to summarize the effect of the discarded data (measurements prior to $k = T - N$) on the current states (Qu and Hahn, 2009). In other words the arrival cost summarizes the total cost that would accumulate from every stage $k = \{0, \dots, T - N - 1\}$ for a given choice of x_{T-N} as illustrated in Figure 2.4. Notice how including the arrival cost in the optimization in Figure 2.4, changes the optimal state estimates to fit better with the complete set of measurements.

¹Also known as *Cost to come*, and *Cost to arrive* (Rao et al., 2003)

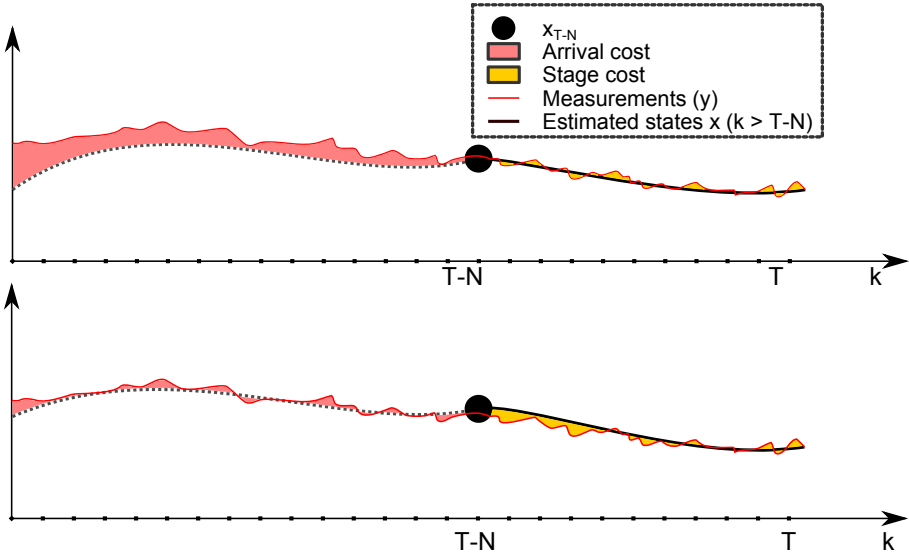


Figure 2.4: Arrival cost illustration. The arrival cost is illustrated by the pink area. In the first graph x_{T-N} is chosen to minimize the stage cost (yellow area), and in the second the x_{T-N} is chosen so that the sum of stage and arrival cost is minimized. Note that neither the dotted line nor the measurements before $t = T - N$ are known or estimated by an MHE.

By including an optimal arrival cost, the **MHE** would be equivalent to the **FHE**. A general analytical expression for the arrival cost is, however, rarely available for nonlinear or constrained systems (Qu and Hahn, 2009). Which means that an approximation of the arrival cost is needed.

There are several ways this approximation can be done, and two approaches are discussed in this section. The most discussed method in the literature is based on the recursive algebraic covariance matrix (P_k in (2.4)) of a Kalman filter based estimator. This is an approach that stems from the fact that this kind of Kalman approach would give an exact arrival cost for a linear system without inequality constraints, which gives reason to believe that an approach like this is a good approximation for the arrival cost of a nonlinear system with inequality constraints as well.

This approximation of the arrival cost is implemented as shown in equation (2.10), (Rao et al., 2003).

$$\hat{Z}_{T-N}(z) = (z - \hat{x}_{T-N}^{mh})^T \Pi_{T-N}^{-1} (z - \hat{x}_{T-N}^{mh}) + \hat{\Phi}_{T-N} \quad (2.10a)$$

$$\Pi_T = Q + A_T \Pi_{T-1} A_T^T - A_T \Pi_{T-1} C_T^T (R + C_T \Pi_{T-1} C_T^T)^{-1} C_T \Pi_{T-1} A_T^T \quad (2.10b)$$

Where:

- z : is the chosen initial value in the current MHE.

- \hat{x}_{T-N}^{mh} : is the optimal x_{T-N} estimated by moving horizon based on data before $t = T - N$.
- Π_T : is the covariance matrix (P_{T+1}) from Section 2.1.1. Found by combining equations (2.3), (2.4) and (2.5)
- $\hat{\mathcal{Z}}$: is the arrival cost
- Φ : is the objective function (as given in (2.8))
- $\hat{\Phi}_{T-N}$: is the objective function, found to be optimal at the MHE iteration when $t = T - N$. This term is added as it plays a big role in the approximation of the total arrival cost \mathcal{Z} . However, in the optimization problem it is independent of all of the decision variables, and hence works as a constant in the objective function. This implies that in an actual implementation of the arrival cost, this element can be omitted.

Note: Q, R, A_T, C_T are equivalent to the matrices described in Section 2.1.1.

A simpler approach to the arrival cost approximation is to assume that the previous estimate of x_{T-N} , is a sufficiently good estimation of the real x_{T-N} . We name this previous estimate \bar{x}_{T-N} . The arrival cost is then approximated as $\hat{\theta}_{T-N}(z) = \mu \|z - \bar{x}_{T-N}\|$ (Alessandri et al., 2008). Where μ is a scalar, and z is the current estimate for x_{T-N} .

Constraints:

These constraints consists of two main elements, namely the constraints that describe the system (hereafter, named system constraints, e.g. $x_{k+1} = f(x_k)$), and additional constraints to shrink the solution space. These additional constraints can be either hard constraints that cannot be broken, or asserted constraints to help the optimization solver find the the correct solution, and/or find the solution faster. The system constraints are usually equality constraints, while the additional constraints are inequality constraints.

2.1.3 Comparison of Kalman and MHE

Kalman filter and MHE has many of the same properties, qualities and disadvantages. They are both optimization-based observers, which attempt to minimize the state estimation error ($e_k = \hat{x}_k - x_k$). Usually, this is also achieved by minimizing the mean square error of e_k , however while this choice of cost function is predefined for the Kalman filter, it is optional for the MHE. For a linear system without constraints (except for the system "constraints") the optimal solution of the MHE is equivalent with the solution of the Kalman filter – however "formulating the system with inequality constraints, prevents recursive solutions such as Kalman filtering" (Rao et al., 2001). The same statement applies with the introduction of nonlinear behavior of the system, one can then no longer apply such an guarantied optimal recursive solution. Do note, however, that there are methods based on the Kalman filter for nonlinear set of equations, there are also methods for adding constraints to Kalman-based estimators (Kandepu et al., 2008a), (Kandepu et al., 2008b),

though it is somewhat less intuitive than what it is for MHE.

Which of the estimators EKF or MHE that is best for all purposes cannot be stated. However some advantages and disadvantages are stated in table 2.2.

+ EKF	+ MHE
Computationally cheaper than MHE (Haseltine and Rawlings, 2003)	Intuitive handling of constraints, (can potentially be handled by Kalman-based approaches as well, but less intuitive (Kandepu et al., 2008b))
Simple implementation for problems without constraints	Versatile, can easily add additional nonlinearity, additional constraints etc.
Very much knowledge on EKF available in the literature	Possibly better with local nonlinearities.(Haseltine and Rawlings, 2003) Provides more intuitive model error parameters, or model parameters for the fault diagnosis.

Table 2.2: Pros and cons EKF and MHE. Note that this is not meant to be any conclusion or indication to which of the estimators that is best, or even best for this thesis, it is simply a comparison based on the available literature.

For the remainder of this thesis, there will be a primary focus on MHE, as it is this estimator that was chosen to be the main estimator from the assignment text. For the purpose of virtual flow metering, however, there is no reason to think, based on the research done in this thesis, that a Kalman-based filter could not work just as well.

2.2 Fault diagnosis

Fault diagnosis is a field that employs various techniques of data handling to detect – and analyze a fault² that occur in a system. It is a science that is becoming increasingly more relevant as more and more systems become autonomous, and the number of employed personnel to monitor the systems is decreasing. This leads to large systems that cannot possibly be monitored simply by human inspection of the measurement data. Add to this challenge an ever increasing demand for improved performance, safety and reliability, paints a good picture of why automatic detection and analysis of faults are getting increasingly more attention (García and Frank, 1997).

²A fault is defined by Isermann and Ballé (1997) as: An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition

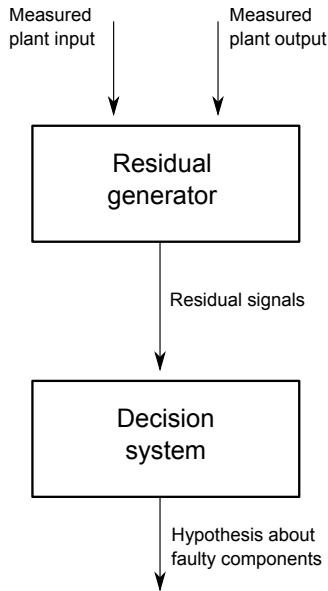


Figure 2.5: Simple illustration of a normal fault diagnosis system (Blanke et al., 2006).

In this paper the fault diagnosis is not going to be used in the traditional sense of detecting physical faults that inflict the system. Rather, the same theory will be attempted utilized to search for especially faulty parts of a mathematical model of the system. However to get a general impression of the fault diagnosis techniques described in the literature this section will cover the basic theory of general fault diagnosis applied to "normal" physical faults.

To limit the scope of this section, it includes only a short introduction of the elements in fault diagnosis, a mention of some of the techniques in the literature, and finally a more thorough introduction to Generalized Likelihood Ratio (**GLR**) test, which is the technique utilized for fault detection in this paper.

2.2.1 Introduction to fault diagnosis

The overall concept of fault diagnosis consists of three central elements as stated in Blanke et al. (2006)

- **Fault Detection:** Decide whether or not a fault has occurred.
- **Fault Isolation:** Find where or in what component the fault has occurred.
- **Fault Identification and Estimation:** Identify the fault, and estimate the magnitude of that fault.

All these elements does not necessarily need to be contained in the fault diagnosis, however it shows the underlying idea, that a fault is detected and then analyzed.

Furthermore for any fault diagnosis system, three steps should be performed (Frank, 1996)

- **Residual generation:** Generation of signals that can reflect the fault.
- **Residual evaluation:** Evaluate the signals provided by the residual generator. Detecting fault, and determine time of occurrence.
- **Fault analysis:** Identification of fault, in addition to possibly determine magnitude and cause of fault.

See Figure 2.5 for an illustration of the steps, note that in this illustration the two last steps are combined into one decision system.

Residual generation

Residual generation is the process of creating good signals that can reflect the faults, and at the same time not react to any unknown disturbances in the system which are not faults. The techniques used for residual generation can be divided into three groups which are all closely related (Frank, 1996)

- **Parity space approach:** Based on a so called parity check of the consistency between the model and the measurements from the system
- **Observer-based approach:** Based on the residuals between estimated measurements and actual measurements
- **Parameter estimation approach:** Based on estimating the system parameters, which often can be good physical indicators of what kind of error that has occurred

Note: these different techniques might have different strengths depending on the situation, and that the most efficient approach often is to use different residual generators in combination (Frank, 1996). For more on these residual generators the reader is referred to [(Delmaire et al., 1994) and (Chow and Willsky, 1984)] for the parity approach, [(Ding et al., 1993)(time domain) and (Patton et al., 1989)(frequency domain)] for the observer-based approach and (Isermann, 1993) for parameter estimation-based approaches.

Residual evaluation and fault analysis

Residual evaluation and fault analysis is the process of determining whether – and where a fault has occurred. This detection and analysis needs to be able to determine that a fault has occurred without making false alarms when a fault has not occurred. To achieve this, choose appropriate thresholds for detection. It can be done in several different ways (Frank, 1996), below some approaches are listed

- Threshold logic, (see Ding and Frank (1991))

- Statistical decision theory (see Basseville and Nikiforov (1993))
- Fuzzy decision making (see Sauter et al. (1994))
- Neural networks (see Sorsa and Koivo (1993))

In this paper the statistical decision theory is the only approach applied, and the only one that is discussed. For further information on the alternatives, the reader is referred to the citations in the bullet list above.

2.2.2 Fault detection using GLRT

Generalized Likelihood Ratio Test (**GLRT**) is simply put a method that determines if the measurements and inputs are more likely to correlate with an alternative model better than the original null-hypothesis \mathcal{H}_0 model.

Independent and Identically Distributed residuals

For the statistical change detection algorithms related to GLRT, there is a general prerequisite that the residuals which are diagnosed are Independent, and Identically Distributed (**IID**). Which more or less means that each residual sample $\varepsilon[k]$ should be uncorrelated with the previous samples $\varepsilon[k - \tau]$. However, this can in many cases be a fairly rough approximation, and it should be noted that with increasing correlation in the residuals, the quality of the fault detection algorithms are decreasing. To reduce the effect of this correlation, a technique called white filtering can be applied on the residuals as explained in Hansen and Blanke (2012). This works as a preprocessing of the signals to improve the performance of the fault detection algorithm. This white filtering attempts to remove the deterministic element of the residuals, leaving only white noise, which has no correlation with previous residual samples. One approach for creating such a white filter can be a Finite Impulse Response representation of a linear estimator

$$\tilde{\varepsilon}[k] = \varepsilon[k] - \sum_{j=1}^J a_j \varepsilon[k - j]$$

With a chosen size of the filter J , which in practice usually does not have to be too large Hansen and Blanke (2013)

For the remainder of this thesis, the residuals are assumed IID.

Log likelihood ratio

The likelihood ratio forms the basis for the GLRT and is a statistical property that compares consistency of the decision variables z with the "null model" and an alternative model. Where the "null hypothesis" \mathcal{H}_0 , is that z is described by the null model, and the alternative hypothesis \mathcal{H}_1 is that the alternative model is a better fit for z . The models are

from this point and on described by their model parameters Θ_0 and Θ_1 .

The log likelihood ratio is found by the following relation:

$$D = \log \frac{P(z; \Theta_1)}{P(z; \Theta_0)} \quad (2.11)$$

Which has the following properties:

- if z is more likely to belong to Θ_0 , then $D < 0$
- if z is more likely to belong to Θ_1 , then $D > 0$

Cumulative sum

Cumulative Sum (**CUSUM**) is based on the log likelihood ratio, and forms the foundation of GLRT (Blanke et al., 2006). Unlike GLRT this method needs an alternative model as proposed in the log likelihood ratio section. It is built up by the following function

$$S(k) = \sum_{i=1}^k \ln \frac{P(z(i); \Theta_1)}{P(z(i); \Theta_0)} \quad (2.12)$$

Which is simply the sum of the log likelihood ratio at each sample z . This $S(k)$ is then used in the test-statistic $g(k)$

$$g(k) = S(k) - m(k) = \sum_{i=j}^k \ln \frac{P(z(i); \Theta_1)}{P(z(i); \Theta_0)} \quad (2.13)$$

$$m(k) = \min_{1 \leq j \leq k} S(j) \quad (2.14)$$

Which is used to decide whether or not to keep \mathcal{H}_0

$$\begin{aligned} \text{if: } g(k) \leq h & \text{ accept } \mathcal{H}_0 \\ \text{if: } g(k) > h & \text{ accept } \mathcal{H}_1 \end{aligned} \quad (2.15)$$

To lessen the computational cost of this likelihood test, a horizon ($N_D :=$ horizon of detection algorithm) is usually added to $S(k)$. This is done by choosing j in (2.14) in the range of $[k - N_D + 1, k]$ instead of $[1, k]$.

Determine a good threshold h

To determine a good threshold h in (2.15), there are several approaches as described in Section 2.2.1. The approach presented in this section is inspired by Willersrud et al. (2014a).

There is no obvious way to intuitively choose a good h , based on a desired behavior of the hypothesis-test. However, given a PDF ($f(x)$) one can scientifically calculate a good threshold h based on a desired sensitivity of the hypothesis test. More precisely, one can

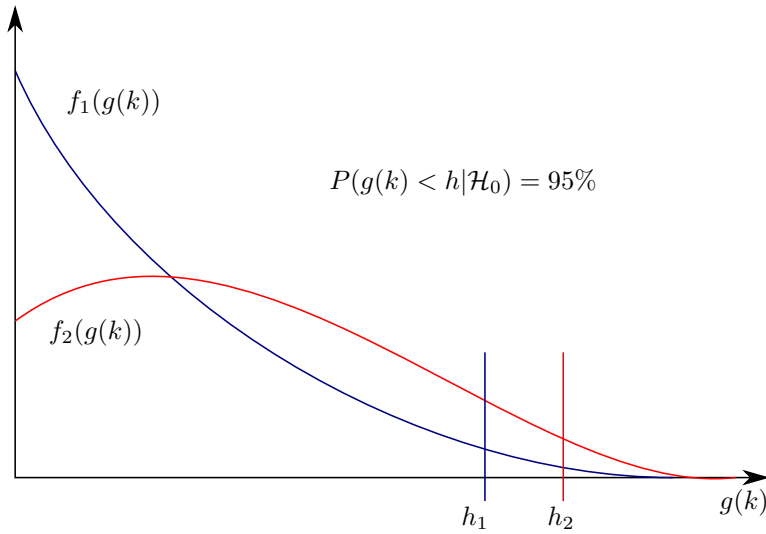


Figure 2.6: Two different possibilities for a distribution, illustrating how the distribution can significantly change the choice of h , given a desired $P_{FA} = 0.05 = 5\%$

choose how low one wants the probability for accepting \mathcal{H}_1 given \mathcal{H}_0 ($P_{FA} := P$ false alarm), and calculate the exact h that assures this probability. Without using any specific PDF, the idea is shown in equation (2.16) which is solved for h .

$$1 - F(h, \Theta) = P(g > h | \mathcal{H}_0) = P_{FA} \quad (2.16)$$

$$F(h, \Theta) = \int_0^h f(g, \Theta) dg = P(g < h | \mathcal{H}_0)$$

$F(h, \Theta)$ is the cumulative PDF of $g(k)$ given \mathcal{H}_0 , $\Theta :=$ distribution parameters (e.g. μ and σ in the normal distribution (Walpole, 2012)).

GLRT

It is above illustrated how one can use the log likelihood ratio to detect if the samples $z(i)$ seem to have a better fit with Θ_1 than Θ_0 , when the alternative model Θ_1 is known. This section describes how one can adjust the CUSUM method to work when the alternative model is not known.

The element of the CUSUM method that cannot be utilized when the alternative model is not known is Θ_1 . An estimated $\hat{\Theta}_1$ is instead implemented. This $\hat{\Theta}_1$ is not set prior to the test, but chosen at each iteration to best fit the model with the data in the the test window N_D . In other words it is a method that determines how well the data fits the "null model"

Θ_0 compared to the best possible alternative model $\hat{\Theta}_1$ ³. This is implemented as

$$S_j^k(\theta_1) = \sum_{i=j}^k \log \frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))} \quad (2.17)$$

$$g(k) = \max_{k-N_D+1 \leq j \leq k} \max_{\theta_1} S_j^k(\theta_1) \quad (2.18)$$

as given by Blanke et al. (2006).

General implementation of a GLR-test

1. Determine the "test-variable"

This implies choosing good test-variables/residuals to be taken into the GLR-test as explained in 2.2.1.

2. Find a good probability distribution to describe the random "test-variable"

To determine the probability that a certain sample $z(i)$ of the random variable \mathcal{Z} are consistent with a model presented by Θ . The model needs to have a known/estimated probability distribution.

3. Choose a desired probability for false alarms to occur P_{FA}

Choose the probability for the test to accept \mathcal{H}_1 , given it is \mathcal{H}_0 . In other words choose how certain one wish to be before rejecting the null hypothesis and accept the alternative hypothesis.

4. Choose a good fit for the probability distribution of the test statistic $g(k)$

To be able to scientifically choose a good threshold h based on a desired P_{FA} , a distribution of the test statistic $g(k)$ is needed. Figure 2.6 illustrates the importance of knowing the distribution of $g(k)$ when one attempts to choose a good h .

5. Based on P_{FA} and the chosen distribution of $g(k)$, calculate the corresponding h .

When one has the equations for the probability distribution of $g(k)$ and the needed certainty to accept \mathcal{H}_1 , the threshold h can be found.

6. Insert the distribution of the decision variables into equation (2.17). From the previous points a density function for the null model and alternative model has been created. This should be inserted in equation (2.17). Furthermore optimal Θ_1 should be found algebraically if possible, and hence removing the \max_{Θ_1} element of (2.18).

7. Set up the comparison of the test statistic $g(k)$ and threshold h

As done in (2.15)

³it should be noted that if the "null model" is a perfect model, then the alternative model $\hat{\Theta}_1$ would be equal to Θ_0

Execution of GLRT

Depending on the dimension of the decision variables and their distribution, the actual execution, and the functions will vary somewhat. However looking behind the specific details, the GLRT execution can be described by the algorithm 1.

Algorithm 1: Basic idea of a simple single variate GLR-test (Blanke et al., 2006)

Given:

A sequence of data $z(1), \dots, z(k)$ with probability density function $p_\theta(z)$ depending on the scalar parameter θ ;

A threshold h ;

Compute: $g(k)$ using (2.17) and (2.18) ;

Decide to:

if $g(k) \leq h$ **then**

 | accept \mathcal{H}_0 ;

else

 | accept \mathcal{H}_1 ;

end

2.2.3 Fault identification

This part is about using the statistically optimal μ_1 of the PDF $f(x, \mu, S)$ found in the fault detection algorithm, to determine what the fault is, as opposed to only knowing, that there is a fault. In the case of this project, this forms the difference between simply detecting that the model is poor, to hopefully isolating which part of the model that is poor. The approach to do this identification/isolation, is based on a technique shown in Willersrud et al. (2014b).

Smearing

Before introducing the fault identification algorithm, a concept known as smearing needs to be introduced (Narasimhan and Jordache, 1999). Optimally one could wish that if a fault occurred in one part of the model (indexed with i), then, that corresponding residual signal w_i would increase respectively. Given a system like this, it would be simple to always identify the fault. However because smearing this is not the case. The fault from one of part of the system will be distributed to all of the residuals that has correlation with the faulty part of the system. The concept of smearing is illustrated with Example 1.

Example 1 Given an optimization system on the form

$$\begin{aligned} J^* &= \min_{w_1, w_2} w_1^2 + w_2^2 \\ &s.t. \\ &g_1(x) = w_1 \\ &g_2(x) = w_2 \end{aligned}$$

Assume $g_2(x^{true}) = 0$ and that $g_1(x^{true}) = 1$. Which means that the true solution implies $w_1^{true} = 1$. However because w_1 and w_2 are related through x , some of the "error" is moved from w_1 through w_2 . By analyzing g_1 and g_2 it is found that the relationship between w_1 and w_2 is given such that any decrease in w_1 entails an increase of twice the magnitude in w_2

$$w_1 = w_1^{true} - \xi \implies w_2 = 2\xi$$

This leads to:

$$\begin{aligned} w_1^* &= 0.8 \\ w_2^* &= 0.4 \end{aligned}$$

This shows that any model error will naturally be "smeared" / distributed to other model error parameters in the system. Even in the cases like this where the decrease in the current model error parameter w_1 are smaller than the increase of the other (w_2). An illustration of this example is shown in Figure 2.7.

Fault identification based on residual vector direction.

If the distribution of the alternative model were known, a CUSUM test could have been used to detect which element of the model that is "faulty". This is not the case in this project, however something might be found that can distinguish the possible "faulty" components of the model from each other. The technique presented here is based on the assumption that given a certain fault – something can be said about the expected direction of the complete vector $\mu_1 - \mu_0$. Using this, one can potentially identify the fault.

The identification process is done by solving

$$i_D^* = \arg \max_i \frac{D_i^T (\mu_1^D - \mu_0)}{D_i^T D_i} \quad (2.19)$$

$$D_i := \frac{K_D \Gamma_{D,i}}{\|K_D \Gamma_{D,i}\|} \quad (2.20)$$

Where $\Gamma_{D,i}$ represents any knowledge of the direction to which the elements of θ_1 change from θ_0 given a certain fault, and K_D is a weighting matrix that tells the optimization

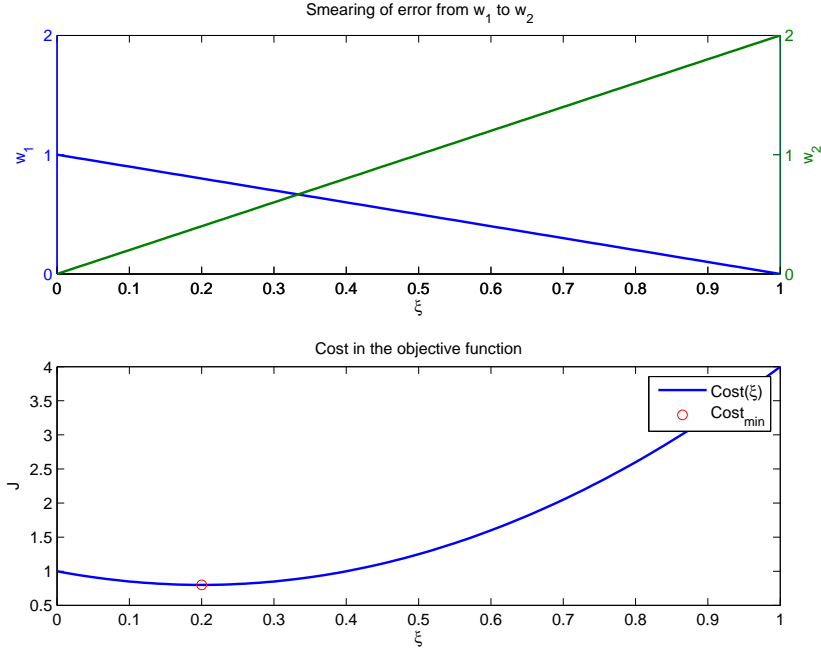


Figure 2.7: Illustration of example 1. There will always be some distribution (smearing) of the error related to one function in the model.

formulation approximately how much each element of θ changes given a fault (compared to each other). This concept has been illustrated with Figure 2.8, and through example 2. Note that the identification algorithm used in this thesis and illustrated in Figure 2.8, is extended from (Willersrud et al., 2014b), by the addition of making the vector comparison between D_i^T and $(\mu_1^D - \mu_0)$ independent of sign. This is done by reformulating (2.19) to:

$$i_D^* = \arg \max_i \frac{\|D_i^T(\mu_1^D - \mu_0)\|}{D_i^T D_i}$$

Example 2 Given a $\theta = [\theta_1 \ \theta_2 \ \theta_3]$, and a set of possible faults $i \in \{1, 2, 3\}$, with the knowledge that

- given fault number 1, then all the elements of θ will increase, which implies $\Gamma_{D,1} = [1 \ 1 \ 1]^T$.
- given fault number 2, then θ_2 will increase and θ_1 will decrease, and the direction of θ_3 is not known. Which implies $\Gamma_{D,2} = [-1 \ 1 \ 0]^T$
- given fault number 3, $\Gamma_{D,3} = [-1 \ -1 \ 1]^T$
- Furthermore, although it is hard to determine the relative magnitude of the change K_D , it is approximately known that changes in θ_1 and θ_2

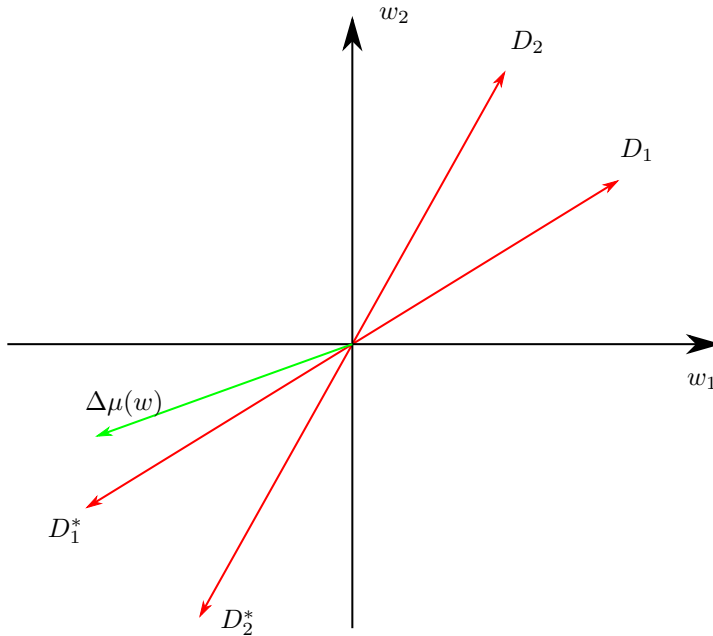


Figure 2.8: Illustration of the fault identification algorithm, in this case, $i_D^* = 1$, because the test-variable vector $\Delta\mu(w)$ fits best with D_1^* , which is the negative of D_1 .

are approximately equal in magnitude, whilst θ_3 changes about 100 times more in magnitude than the other two. This leads to $K_D = [1 \ 1 \ \frac{1}{100}]^T$

2.3 Measurement error, sensitivity and its characteristics

In state estimation, there is as previously stated two main sources for error, namely measurement error and model error. Although model error is the main focus of this paper, it makes sense to cover some theory about the measurement error as well. This section will cover the various elements that together form the total measurement error, the nomenclature in the field of measurement error, and a short summary of normal accuracies of some measurement sensors⁴.

2.3.1 Nomenclature in the field of measurement error

This section is added to better understand the terms that are used to describe the measurement error, both in this paper and more importantly in the data sheets of measurement sensors.

⁴also called transmitters, transducers and indicators

Terms related to the uncertainty of measurements, are often used interchangeably, although they in fact represents different aspects of uncertainty (Webster and Eren, 2014). This should be avoided as it creates confusion and sometimes completely wrong statements. To clear this up, a list of some general expressions are defined below. All these explanation-s/definitions are based on Webster and Eren (2014).

- *Accuracy*: is the closeness of a measurement to the value defined to be the true value. An accuracy of $\pm 5\%$ generally implies that almost all of the time (e.g 99% of the time) its indication is within 5% of the "true value"
- *Uncertainty*: is the negative equivalent of Accuracy, the distance of the measurement from the true value. Note however that Uncertainty is unambiguous, while Accuracy is not, (e.g what is twice the accuracy of $\pm 2\%$? Is it $\pm 1\%$ or $\pm 4\%$?)
- *Sensitivity* : An expression of the influence an error source has on a test or measured result.
- *Precision*: is a term that too often is misused as a synonym to accuracy, which is not the case. Precision describes the repeatability of the measurements, which can be seen as the variance of the measurements. It differs as a term from accuracy and uncertainty because the mean of the measurements does not necessarily represent the "true value". This is explained further in Section 2.3.2 and Figure 2.9
- *Discrimination*: The discrimination is the smallest increment possible from the sensor. (E.g, discrimination of 0.1, would mean there is no output between 2.4 and 2.5, $x = 2.432$ becomes $x = 2.4$)
- *Resolution*: A synonym to discrimination
- *Error*: The difference between the measured and true value

Understanding the data sheets that presents the accuracy of the sensors is important as there can be massive differences depending on how you understand the numbers presented. This is explained in Table 2.3, by some examples of different accuracy description from different suppliers and data sheets.

Note: the Best Fit Straight Line (**BFSL**) description of accuracy (shown in Table 2.3) can be misleading. It essentially describes only one part of the uncertainty, namely the linearity (which is described in the section below). However given an accuracy of $\pm x\%$ BFSL, one should usually understand it as a measure of the total accuracy, and interpret it as $\pm x\%$ FS with the linearity approximated by BFSL. This should however be verified by reading an explanation in the respective data sheet (if present). The reason as to why BFSL is added is probably because the linearity aspect of the uncertainty often plays a dominating role in the total uncertainty.

2.3.2 The elements of measurement error

Some terms are previously defined to help understand the description of measurement error provided in the data sheets of instruments. In this section the different elements that

Presentation	Description	Comment
$\pm x$ [unit]	Absolute value	
$\pm x\%$	Accuracy depending on the magnitude of the measured value.	
$\pm x\%$ FS	% of Full Scale (Lipták, 1995). Indicates accuracy as $\pm \frac{x \times 100\%}{x_{max}}$ where x_{max} is the maximum value in the range of the sensor.	Note that there is often a significant difference between $\pm x\%$ and %FS. Note also that this is often related to linearity
$\pm x\%$ FS(O/R)	% of Full Scale (Output/Range). Exactly the same as $\pm x\%$ FS	
$\pm x\%$ BFSL	Best Fit Straight Line Usually same as %FS with the added information of how the nonlinear relationship is approximated, see sec 2.3.2	
$\pm x\%$ of calibrated span	You can adjust the range of your transmitter yourself, which opens up for significantly more accurate measurements	

Table 2.3: Various approaches for presenting accuracy (RDP group, 2005)

contribute to the measurement error is described.

The actual error of a measurement can be divided into two main elements, namely systematic error (bias), and random error (noise). Where the precision presented in the list above only describes the the random error part of the total error. For an illustration of this concept see Figure 2.9.

The systematic error generally results from poor calibration, from degradation of the sensor over time, or from changes in the condition under which the sensor operates. It is therefore important to periodically recalibrate the sensor to keep this systematic error at a minimum.

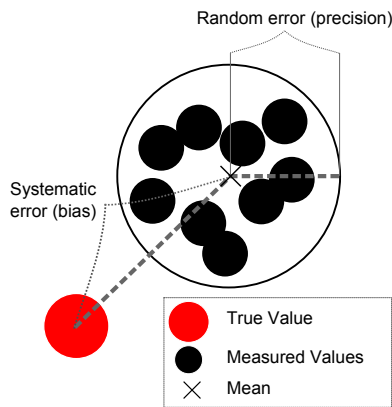


Figure 2.9: Illustration of the two elements of a measurement error. Adapted from Webster and Eren (2014)

For now the different elements of the actual measurement error has been described. This next part will present the elements that are responsible for the error in the first place. Several sources for error exist. To name some; **error of nonlinearity, error from hysteresis, zero stability, discrimination, saturation** and **drift** (Webster and Eren, 2014). However, only the error of nonlinearity is described further, as this can generally contribute most to the total uncertainty. It is also often the only source of error that is highlighted when the accuracy is described in the data sheets of measurement instruments.

When a measurand is being measured, what happens, is in fact that an instrument detects and reacts in some way on the property being measured, the instrument will then send out some kind of signal (normally current or voltage⁵). This signal is then interpreted to give us information on the property that is being measured. The relationship between the measured property and the electrical signal, however, is rarely as nice as one would like it to be. This relationship and the desired relationship forms the foundation of Error of

⁵depending on whether it is a transmitter or transducer

nonlinearity⁶. The interpretation of the transmitted signal are generally assumed linear with the measured property. However this is often not the case, and the deviation from approximated linear relationship to the actual (nonlinear) relationship is called Error of nonlinearity. See Figure 2.10 for illustration of the phenomena.

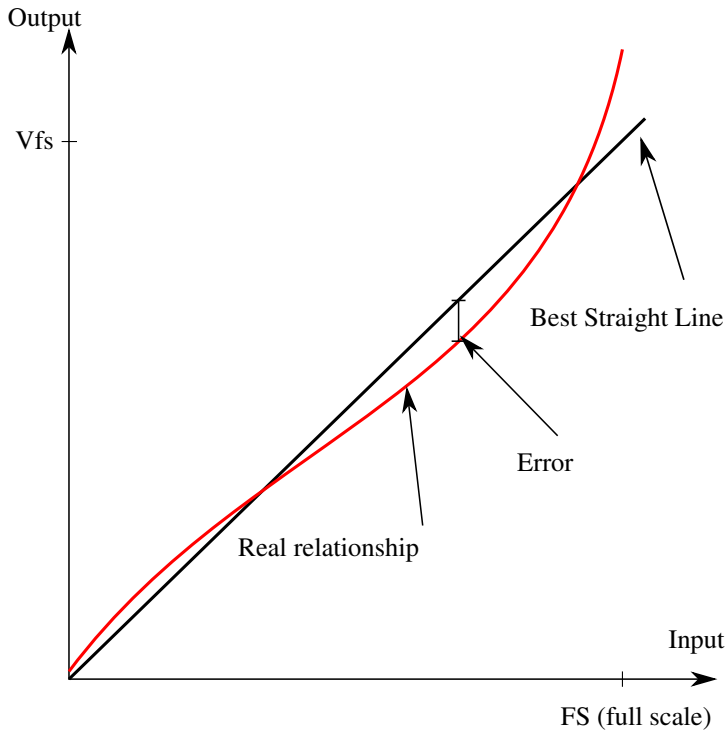


Figure 2.10: Illustration of error from nonlinearity, and best fit straight line. Adapted from RDP group (2005)

2.3.3 Typical uncertainties of measurement sensors in the market today

This section consists of a short survey on the accuracies of measurement instruments in the market today. It is based on a chosen selection of instruments used in the oil and gas industry (or at least can be used in the oil and gas industry). The data is shown in Table 2.4, and all the presented information is based on the data sheets for the respective sensors.

The instruments presented in Table 2.4 are all directly relevant for this project, and covers all the measurements used in this thesis, namely topside single phase flow measurements, pressure transmitters and temperature transmitters (both for topside and subsea).

⁶also called Linearity

Measurand	Accuracy	Range (selected)	Supplier (Model)	Note
Pressure	$\pm 0.25\%$ FS (BFSL)	0... {138, 207 or 345} bar	PSI-Tronix (PT-L1)	
Pressure	$\pm 0.2\%$ BFSL	0... {350 or 400} bar	Honeywell (425)	Particularly usefull for drilling, but can potentially be used in production as well.
Pressure	$\pm 0.25\%$ of calibrated span	Adjustable up to 350bar	Rosemount (4600)	Note that this percentage is of calibrated span which can be significantly better than full scale.
Pressure	$\pm 0.1\%$ FS	0 – 690bar	Emerson (220-30-020)	Especially created for subsea monitoring .
Temperature	$\pm 0.3^\circ\text{C}$	calibrated between –40, 125	Roxar (Sen-Corr PTPT)	Specifically intended for subsea measuring. Also a pressure transmitter
Temperature	$\pm 0.05\%$ of calibrated span	Adjustable up to 850°C	ABB (TTF300)	Note that the minimal span is of 10°C
Temperature	$\pm 0.02\%$ of calibrated span	Adjustable up to 850°C	Rosemount (3144P)	Note that the minimal span is of 10°C
Liquid mass flow	$\pm 0.1\%$ of actual measured value	950 – 430000kg/h	Honeywell VersaFlow Coriolis	Both gas and liquid
Gas mass flow	$\pm 0.5\%$ of actual measured value	950 – 430000kg/h	Honeywell VersaFlow Coriolis 1000	Both gas and liquid
Liquid mass flow	$\pm 0.1\%$ of actual measured value	Tube diameter between 3mm – 50mm Not sure about kg/h	Foxboro Coriolis CFT51	Both gas and liquid
Gas mass flow	$\pm 0.5\%$ of actual measured value	Tube diameter between 3mm – 50mm Not sure about kg/h	Foxboro Coriolis CFT51	Both gas and liquid

Table 2.4: Given accuracies from some suppliers. Note that there is often a large selection of ranges on the transmitters, but only the relevant ranges has been picked out in this table.

Modeling

This chapter presents the modeling and implementation of the system that was introduced in Section 1.4. The first section of this chapter presents the modeling of the system, including the equations chosen, and a description of the phenomena that have been included and those which have been omitted. Furthermore, a section on modeling of the measurement noise is added as an extension to the theory described in Chapter 2. The modeling of this system is adapted from the models of Aamo et al. (2005).

- Test case:
In this project, an oil production system as shown in Figure 1.3 is used as a test case. It consists of 4 wells and pipelines running through chokes to a separator. It is a test case that is simplified by disregarding elements such as subsea manifold, risers, etc. It is simply 4 wells that goes directly to 4 topside (surface) wellheads and from there merge into one pipeline which leads to the separator.
- Scenario:
This experiment will be performed by artificially introducing realistic measurement errors, and more importantly, artificially introducing model errors. The model errors are introduced by changing some of the model parameters of the simulator to inflict errors in the models used in the state estimators. The task is then to detect that the model has become poorly calibrated, and identify which part(s) of the model that has (have) become poorly calibrated. The specific scenarios that will be tested are presented in Chapter 6.

3.1 Modeling of the system

For a system like this there will always be additions that can be made to further increase the complexity of the simulator. However to simplify the model, and to more easily highlight the effects that are to be investigated in this project, some natural components on a subsea production system have been omitted. The phenomena that is added have been listed below:

- Inflow to the well from the reservoir
- Pressure loss due to elevation in pipeline
- Mass conservation in the pipeline from bottom hole to choke valve at the surface
- Pressure loss over choke valve
- Pressure after choke valve is equal for all flows that merge into the separator

The assumptions relevant for the test case have been listed below

- Constant and known **GOR** and **WC** for all wells
- Linear relationship between pressure difference $P^{res} - P^{BH}$ and inflow to the well
- No manifold, no subsea wellheads, to simplify the model, and decrease the number of "sub models" and variables.
- Negligible pipeline between choke valve (wellhead) and separator
- Instant separation of oil, gas and water in the separator
- Any pressure loss due to friction in the pipelines is incorporated into the valve equation of the wellhead
- Ideal gas law valid for the gas component of the reservoir fluid

3.1.1 System equations

In this section the mathematical modeling of the system is presented. To ease the understanding of the buildup of this model, Figure 3.1 is presented below.

Variables

p^{res}	"Known"	Reservoir
p^{BH}	Measured	Bottom Hole
p^{WH}	Measured	Wellhead (before choke)
p^{sep}	Measured/Known	Separator
GOR	"Known" and constant	Gas Oil Ratio
WC	"Known" and constant	Water Cut
$q_{o,w,g}^{BH}$	Unknown	Inflow Bottom Hole
$q_{o,w,g}^{WH}$	Unknown	Flow out from wellhead
u	Known	Input to the choke valves from control system
PI	"Known" and constant	Inflow parameter
C_c	"Known" and constant	Valve coefficient

Table 3.1: The variables in the model have been listed in this table, to give an overview of what is known, and what is not. Notice that a distinction has been made between "Known" and Known. This distinction was made because the parameters that are characterized as Known, is in fact known and can be controlled by the user, while the ones that are characterized with "Known", are only assumed to be known, but in a real situation they would only be approximations.

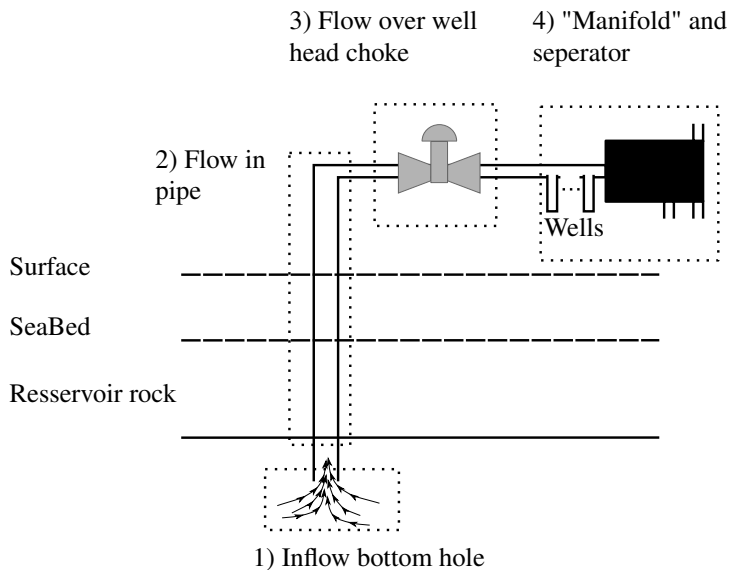


Figure 3.1: Simplified illustration of one well, and how this thesis separates the entire system into submodules (1,2,3 and 4). These submodules are presented with equations in Section (3.1.1).

Note: the water and oil are hereafter are consider to be one identity, namely liquid - represented below by x_2 and $q_l^{BH/WH}$. For that reason it is no longer relevant to talk about WC and GOR as properties of the flow¹ - and instead a new property is introduced, namely Gas to Liquid Ratio (**GLR**). Below the relations with GLR are stated:

$$\begin{aligned} \text{GLR} &= \frac{q_g}{q_l} = \frac{q_g}{q_o(1 + \frac{\text{WC}}{100 - \text{WC}})} = \text{GOR} \frac{100 - \text{WC}}{100} \\ &= \text{GOR} \left(1 - \frac{\text{WC}}{100}\right) \end{aligned} \quad (3.1a)$$

$$q_g = \text{GLR} \times q_l \quad (3.1b)$$

$$q_g = \frac{\text{GLR}}{\text{GLR} + 1} q_c \quad (3.1c)$$

$$q_l = \frac{1}{\text{GLR} + 1} q_c \quad (3.1d)$$

Note: GLR, GOR and WC are all mass fractions in this thesis, as opposed to the norm which seems to be volume fraction (at standard conditions). This is a choice made, because this thesis always work with the fluids in mass, hence removing the need for conversion between volume fraction and mass fraction.

Inflow bottom hole:

$$q_l^{BH} = PI(p^{res} - p^{BH}) \quad (3.2a)$$

$$q_g^{BH} = \text{GLR} \times q_l^{BH} \quad (3.2b)$$

- PI : is a constant parameter
- q_c : is the total mass flow (including gas, oil and water)

Flow in pipes:

The following equations are inspired by Aamo et al. (2005), however adjustments have been made, as there is no added gas lift in this system.

$$\dot{x}_1 = q_g^{BH} - q_g^{WH} \quad (3.3a)$$

$$\dot{x}_2 = q_l^{BH} - q_l^{WH} \quad (3.3b)$$

where

$$q_l^{BH} = q_o^{BH} + q_w^{BH}$$

$$q_g^{WH} = \frac{x_1}{x_1 + x_2} q_c^{WH}$$

$$q_l^{WH} = \frac{x_2}{x_1 + x_2} q_c^{WH}$$

- q_c : total mass flow rate
- x_1, x_2 : accumulated mass in the well - gas and liquid respectively.

¹For more on these properties look to Appendix 1.1

Flow over wellhead choke:

For the wellhead choke, a simple, but widely used valve equation is used. It should be noted, however, that there are many potential issues not accounted for in this model of the valve - such as: cooling over choke, critical flow and turbulence in the gas downstream of the valve.

$$q_c^{WH} = C_c^{WH} \sqrt{\rho_m (p^{WH} - p^{sep})} u \quad (3.4a)$$

$$p^{WH} \geq p^{sep}$$

where

$$\rho_m = \frac{x_1 + x_2}{L_t A_t} \quad (3.4b)$$

$$p^{WH} = \frac{RT_t}{M} \frac{x_1}{L_t A_t - \nu_l x_2} 10^{-5} \quad (3.4c)$$

$$p^{BH} = p^{WH} + \rho_m g L_t 10^{-5} \quad (3.4d)$$

- ρ_m := density of oil/gas/water mix at the wellhead.
- L_t, A_t := the length and cross section of the well
- p^{WH} : pressure at the top of the tubing, directly before choke valve
- p^{sep} : pressure by the separator which assumed equal to that of the pressure directly after the choke valve
- ν_l : specific volume of the liquid
- M : molar weight of the gas
- R : gas constant
- T_W : Temperature in the well
- u : opening of valve, in the range $[0, 1]$

Note that (3.4c) and (3.4d) are not directly related to the flow over the choke, but stems from pressure loss due to elevation, and for (3.4c), pressure found through the ideal gas law.

Collection of flow from all wells, and through separator:

Because this thesis assumes no manifold, and instant separation of the oil/gas/water mixture in the separator – this submodule is fairly easy, it is simply a sum of the inflows.

$$q_g^l = \sum_{i=1}^4 q_g^{WH,i} \quad (3.5)$$

$$q_l^l = \sum_{i=1}^4 q_l^{WH,i}$$

- $n := \#$ wells
- $q_{l/g}^{WH,i} :=$ the flow through the choke of well i
- $WC_i :=$ the water cut of the liquid entering well i
- $q_{g/l}^l :=$ flow of gas/liquid leaving the separator.

3.1.2 Measurement noise.

In Table 2.4 a survey on the measurement accuracies declared by the suppliers were presented. The instruments were all presented with ranges within which the suppliers guarantee that the measurements will be (95% (or 99.7%) of the time). It did, however, not proclaim how the error was distributed, and whether or not it was white noise. For the remainder of this thesis the measurement error is modeled as Gaussian zero mean white noise. The standard deviation of the measurement noise inserted into the simulator is also set based on the provided accuracies in Table 2.4.

In this section these measurement error models are tested, by analyzing measurements provided from an operative oil platform. The results found in this measurement analysis will also be compared to the provided accuracy in Table 2.4. It should, however, be noted that the instruments used to provide the measurements in this analysis are not the same as the instruments shown in the table.

Results

Provided with this analysis are two tables and two figures. Table 3.2 presents the standard deviation of the (approximately) normally distributed measurement errors, and Table 3.3 summarizes the most important results from this analysis. Figure 3.2 shows a subset of the measurements presented in Table 3.2, namely

- Temperature measurements wellhead
- Pressure bottom hole
- Oil flow, exported
- Gas flow, exported.

The first column presents the measurement signal and the estimated true values (based on moving horizon filter) which is subtracted from the measurements to obtain the noise component of the measurements. The second column presents the distribution of the residual $y - \hat{y}$, where y is the provided measurements, and \hat{y} is the estimated true value. Along with the distribution is a fitted normal distribution to illustrate the closeness of the measurement residual distribution to a normal distribution. The third column provides the power spectral density (PSD) of $y - \hat{y}$, which is provided to analyze the assumption that the measurement error is white.

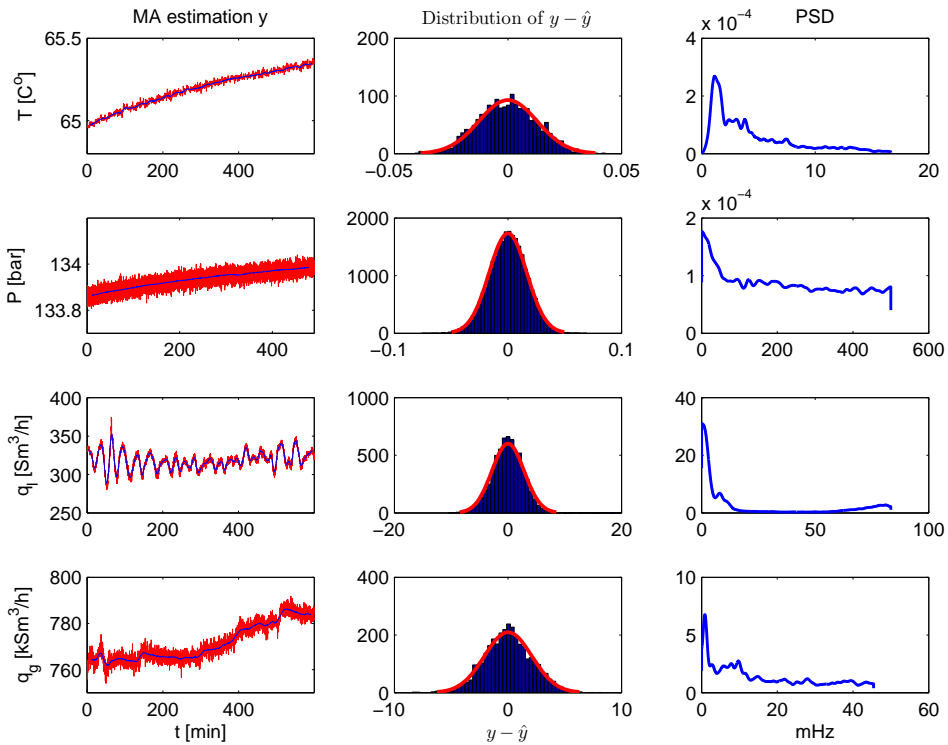


Figure 3.2: Analysis of operational measurement data, for the purpose of determining the characteristics of the noise. The actual standard deviations are presented in Table 3.2. MA := Moving Average, PSD := Power Spectral Density. Note that to save space, only the bottom plots have an xlabel, these xlabels are valid for every plot in that column. The different lengths of the PSD's comes as a result of different samplings rate.

Figure 3.3 is more of a side note that illustrates the accuracy of flow measurements taken subsea, with a multi phase meter. Measurements like these are not used in this thesis, and is for that reason not relevant with respect to choosing good artificial measurement noise introduced to the simulator. It is, however, of interest as these measurements measures the same states that are estimated in this thesis, (as discussed in the introduction).

Discussion of measurement error analysis

In this section the results shown in Section 3.1.2 are analyzed and discussed. Note: If there is a bias error in the measurements, this bias will be removed together with the true value when this method of extracting the measurement error is used. Which means that the error extracted from these signals is in fact only the precision (random error) component of the measurement error - not the complete measurement error (see Section 2.3.2). However, as nothing more is known about the system than the measurement signals provided, the

	T [°C]	P [bar]	Gas Flow [Sm ³ /h]	Liq Flow [Sm ³ /h]
Bottom Hole	0.0006	0.0022		
Upstream WH	0.0170	0.016		
DownStream WH	0.0166	0.0689		
Annulus	0.0684	0.0008		
Well			751.7 (≈ 2.2%)	0.97 (≈ 9.7%)
Export			2087 (≈ 0.3%)	1.81 (≈ 0.6%) ^a

Table 3.2: Standard deviation of measurement noise, found through analysis of data provided from a operational platform. See Figure 3.2. Note that the percentages are percent of the actual flow.

^aNote that this signal had a lot of process dynamics in the measurement - making it harder to extract the noise from the signal see Figure 3.2. The noise has been attempted withdrawn, however the quality of this extraction is poor, and it is reason to believe that the σ found in Table 3.2 for liquid (oil) flow export is artificially large.

- All the measurements seem to have distributions similar to a normal distribution.
- It is difficult to verify that the noise act as white noise, but it seams reasonable to believe that, at least the temperature and pressure measurements, can be modeled as white - which is discussed further in 3.1.2
- Small random error in the pressure- and temperature measurements. Somewhat larger for flow measurements, but the random error found for these measurements are also more uncertain.

Table 3.3: Main results from Section 3.1.2

random error is the only thing that can be extracted, and hence the only component used in the analysis that is conducted in this section.

This discussion is separated into three main part, namely

- A standard deviation comparison with the values provided in Table 2.4
- A discussion on the assumption of Gaussian white noise
- A small discussion of the measurements done on subsea multi phase flow measurements.

The standard deviation (σ) of the distributed measurement error describes the precision of the measurements, however, the accuracies presented in Table 2.4 are not presented in standard deviation. Instead they are presented with a range within which most of the measurement errors will be distributed. Assuming that this range is defined with a 95% certainty (which it is defined to be in some of the cases) then each end of this range represents $\pm 2\sigma$. It should also be noted that the accuracy provided in the data sheets also accounts for error from bias, which is not accounted for in the results of this section.

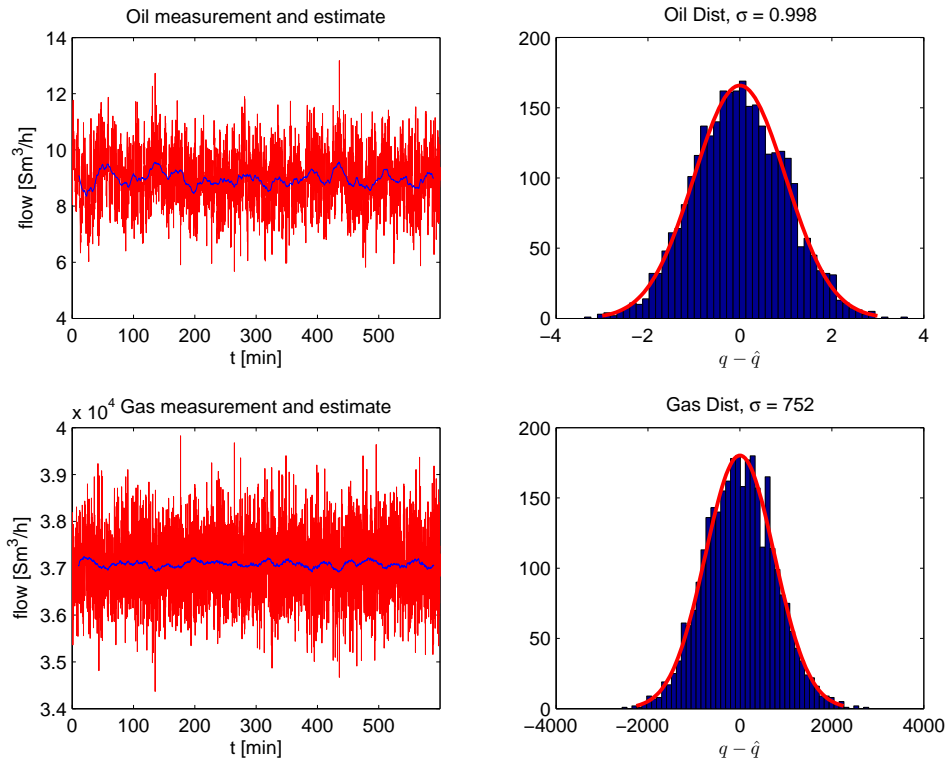


Figure 3.3: Measurement noise of gas and oil measurement conducted subsea with a multiphase flow measurement system.

From the accuracy found in the data sheets, the different suppliers (except Rosemount) only guarantees measurement error with standard deviation of $\sigma = 0.2\% * FS / 2 \approx 0.4 \text{ bar}$. This is considerably higher uncertainty than all of the pressure measurements provided indicate. In the case of Rosemount, the accuracy is decided based on the calibrated range of the pressure measurements, which could potentially lead to accuracies in the same range as the measurements provided in this test indicate. The temperature measurements are generally said to have good accuracy by the suppliers, this corresponds well with the measurements analyzed in this thesis. For the uncertainties in the flow, only the exported flow measurement are relevant. For the exported flow, it can be seen that the standard deviation of the gas flow is within reasonable magnitude compared to that of the accuracy stated by the suppliers for similar instruments. For the oil flow on the other hand, the accuracy seems to be considerably worse than that of the accuracy stated by the suppliers. The σ found through this analysis, however, might be drastically overestimated as the extraction of the noise from the measurements were particularly difficult for the oil flow. This because there were considerably more activity in the actual (estimated) true flow than what was the case in any of the other estimates (which can be seen from Figure 3.2).

This analysis also covers the characteristics of the measurement noise, through the distribution plots and power spectral density plots in Figure 3.2. Based on the data, it can be observed that, there seems to be no reason to think that the measurement noise from any of the instruments are better characterized by any other distribution than the Gaussian distribution. The minor deviations from the Gaussian distribution in the temperature measurement and gas flow can be defended by the fact that they have a smaller sample size. For the oil flow the lack of consistency can be explained by a poor estimation \hat{y} , and hence a poor extraction of the noise.

The Power Spectral Density (PSD) of a zero mean white noise signal is expected to be a constant line. Any deviation from a constant line would be an indication that the samples of the noise are correlated. In practice no finite sequence of samples will give a constant level over all the frequencies in the PSD, however, the degree to which there seem to be a lack of any peaks in the PSD gives a strong indication on the whiteness of the signal. The measurement errors analyzed in this section, all seem to have slight peaks towards 0 Hz.

Whether or not these peaks stem from dynamics in the process that was not successfully removed during the noise extraction, or from actual correlations in the noise is no easy task to determine. However, it is not unreasonable to believe that some of the correlations in the measurement noise comes as a result of process dynamics transferred to the estimated measurement noise. This is especially relevant for the flow measurements as the extraction of the measurement noise was harder with these measurements. Assuming that the peak towards zero can be defended by remaining process dynamics still present in the estimated noise ($y - \hat{y}$), one can for the temperature and pressure measurements safely approximate white noise characteristics. In the case of the flow measurements, however, the analysis done in this thesis does not form enough of a basis to conclude in any direction whether or not the measurement noise can be characterized as white noise.

Figure 3.3, presents the measurements of flow done by a subsea multiphase flow meter (Falcone and Hewitt, 2002). It provides the user with the same information as the VFM, but based on actual measurements from an instrument placed physically somewhere in the production line between the reservoir and the platform (usually by manifolds (High et al., 1995), but can also be placed by each single wellhead.). The field of multiphase flow meters (**MF**M) was briefly presented in Section 1.1, in this presentation the MF M were presented as an alternative to VFM, it should, however, be stressed that it can also be used as a significant asset to a VFM system.

Looking at Figure 3.3, it is evident that the measurements from these kind of instruments are imprecise. For the oil flow presented in the figure it shows that one must expect random error up to $\pm 3S_{m^3/h} = \pm 30\%$ of the true flow. This is not counting possible additional bias error. The precision of the gas flow seems to be a little bit better, but one would still have to expect random error up to $\pm 6\%$ of the true flow. Although a plot is not added for the PSD, tests have been conducted, and both seem to be fairly consistent with white noise.

A_T	0.12 m
L_T	2000 m
r_t	61.8 mm
C_c	0.5060
p^{sep}	50 bar
p^{res}	[2.5, 2.55, 2.60, 2.45] 10^2 bar
R	8.3145
M	0.020 06 kg/mol
g	9.81 m/s ²
T_T	350 K
ρ_o	900 kg/m ³ (Jahn et al., 2008)
ρ_w	1000 kg/m ³

Table 3.4: Parameters used in the simulations. Chosen arbitrarily in the scope of reasonable values.

3.2 Implementation

There will be implemented two simulators based on the model presented above.

- Standard approach: using the equation of Section 3.1.1, in a differential solver in matlab (in this project *ode45*)
- Alternative approach: solving a feasibility problem, with the constraints being the equations of Section 3.1.1.

In both of the implementations, the parameters are given as shown in Table 3.4.

With the standard implementation there cannot be any direct inequalities, which means that equation (3.4a) is changed to

$$q_c^{WH} = C_c^{WH} \sqrt{\max(\rho_m(p^{WH} - p^{sep}), 0)} u$$

The standard implementation is validated in Section 3.2.1 with respect to steady state calculations of the equations, and intuitive understanding of the behavior in a well.

Simulation using optimization

The alternative implementation is to simulate the system using an optimization approach without objective function, (also called a feasibility problem). With this approach, the equations of the model is implemented as constraints to the feasibility problem. The differential equations ((3.3b) and (3.3a)) are implemented in the feasibility problem constraints by the use of Euler's method² ($x_{k+1} = x_k + dt f(x_k)$) (Egeland and Gravdahl, 2002). This way of implementing the model is validated by comparing the simulation results with the results of the standard implementation. The implementation as a optimization problem is given as in (3.6)

²Also called Euler's standard method or forward method.

min \mathbf{x} 0,

$$\mathbf{x} = \left[q_{g/l,k}^{BH,i}; q_{c/g/l,k}^{WH,i}; p_k^{WH,i}; p_k^{BH,i}; x_{1,k+1}^i; x_{2,k+1}^i; \rho_{m,k}^i; q_{g/l,k}^l \right]$$

$$\forall i \in \{1, 2, 3, 4\}, k \in 0 \dots N - 1$$

s.t

$x_{1,0}^i$ and $x_{2,0}^i$ is given $\forall i$

$$q_{l,k}^{BH,i} = PI^i(p^{res,i} - p_k^{BH,i}), \forall i, k \in 0 \dots N - 1 \quad (3.6a)$$

$$q_{g,k}^{BH,i} = GLR q_{l,k}^{BH,i}, \forall i, k \in 0 \dots N - 1 \quad (3.6b)$$

$$q_{c,k}^{WH,i} = C_c^{WH,i} \sqrt{\rho_m(p_k^{WH,i} - p^{sep})} u_k^i, \forall i, k \in 0 \dots N - 1 \quad (3.6c)$$

$$q_{g,k}^{WH,i} = \frac{x_{1,k}^i}{x_{1,k}^i + x_{2,k}^i} q_c^{WH,i}, \forall i, k \in 0 \dots N - 1 \quad (3.6d)$$

$$q_{l,k}^{WH,i} = \frac{x_{2,k}^i}{x_{1,k}^i + x_{2,k}^i} q_c^{WH,i}, \forall i, k \in 0 \dots N - 1 \quad (3.6e)$$

$$\rho_{m,k}^i = \frac{x_{1,k}^i + x_{2,k}^i}{L_t A_t}, \forall i, k \in 0 \dots N - 1 \quad (3.6f)$$

$$p_k^{WH,i} = \frac{RT_t^i}{M} \frac{x_{1,k}^i}{L_t A_t - \nu^i x_{2,k}^i} 10^{-5}, \forall i, k \in 0 \dots N - 1 \quad (3.6g)$$

$$p_k^{BH,i} = p_k^{WH,i} + \rho_m g L_t 10^{-5}, \forall i, k \in 0 \dots N - 1 \quad (3.6h)$$

$$x_{1,k+1}^i = x_{1,k}^i + (q_{g,k}^{BH,i} - q_{g,k}^{WH,i}) * \Delta t, \forall i, k \in 0 \dots N - 1 \quad (3.6i)$$

$$x_{2,k+1}^i = x_{2,k}^i + (q_{l,k}^{BH,i} - q_{l,k}^{WH,i}) * \Delta t, \forall i, k \in 0 \dots N - 1 \quad (3.6j)$$

$$q_g^l = \sum_{i=1}^4 q_{g,k}^{WH,i} \quad (3.6k)$$

$$q_l^l = \sum_{i=1}^4 q_{l,k}^{WH,i} \quad (3.6l)$$

$$p_k^{WH,i} \geq p^{sep}, \forall i, k \in 0 \dots N - 1$$

$$p^{res,i} \geq p_k^{BH,i}, \forall i, k \in 0 \dots N - 1$$

$$x_{1,k}^i \geq 0, \forall i, k \in 0 \dots N - 1$$

$$x_{2,k}^i \geq 0, \forall i, k \in 0 \dots N - 1$$

Note that the following syntax is used in this paper

$$\begin{aligned} \mathbf{x} &= [x_k^i], \quad \forall i \in 1, \dots, I, \quad k \in 1, \dots, N \\ &= [x_1^1, \dots, x_N^1, x_1^2, \dots, x_N^2, \dots, x_1^I, \dots, x_N^I]^T \end{aligned}$$

Where $k \in 1 \dots N$, are the time steps of the problem. For the sake of simulation, this time horizon can be set to $k = 0$, implying that the simulation is done iteratively by solving (3.6) at each time step. However the optimization problem is formulated with this time horizon as the same equations will be utilized in the estimators, which do need a time horizon.

Equations ((3.6a),(3.6b),(3.6f),(3.6h),(3.6i),(3.6j),(3.6l),(3.6k)) are all linear and are implemented by $Ax = b$, the inequalities ((3.6c)), (3.6d),(3.6e),(3.6g)) are implemented by $Ax \leq b$. The remaining equations are nonlinear equalities, and are implemented as they are with ($c(x) = 0$). Additionally, these equations are differentiated and supplemented to the solver³. The derivatives are given by the Jacobian matrix

$$J(\mathbf{x}) = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \end{bmatrix}$$

$$J_1(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \frac{-\frac{1}{2}C_c \rho m u}{\sqrt{\rho(p^{WH} - p^{sep})}} & 0 & 0 & 0 & \frac{-\frac{1}{2}C_c(p^{WH} - p^{sep})u}{\sqrt{\rho(p^{WH} - p^{sep})}} & 0 & 0 \end{bmatrix}$$

$$J_2(\mathbf{x}) = \begin{bmatrix} 0 & 0 & \frac{-x_1}{x_1+x_2} & 1 & 0 & 0 & 0 & \frac{-q_c^{WH}}{x_1+x_2} + \frac{x_1 q_c^{WH}}{(x_1+x_2)^2} & \frac{x_1 q_c^{WH}}{(x_1+x_2)^2} & 0 & 0 & 0 \end{bmatrix}$$

$$J_3(\mathbf{x}) = \begin{bmatrix} 0 & 0 & \frac{-x_2}{x_1+x_2} & 0 & 1 & 0 & 0 & \frac{x_2 q_c^{WH}}{(x_1+x_2)^2} & \frac{-q_c^{WH}}{x_1+x_2} + \frac{x_2 q_c^{WH}}{(x_1+x_2)^2} & 0 & 0 & 0 \end{bmatrix}$$

$$J_4(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{-RT_t}{M(L_t A_t - \nu_l x_2)} 10^{-5} & \frac{-RT_t \nu_l x_1}{M(L_t A_t - \nu_l x_2)^2} 10^{-5} & 0 & 0 & 0 \end{bmatrix}$$

3.2.1 Validation of the model

In this thesis, no real data are available for performing the validation. This means that the validation will be more of a test to check that the model has been implemented correctly and that one gets the expected results. The validation is essentially split up in two categories; dynamical and steady state validation.

- Steady state validation is a test that is conducted to check that the steady state value is correct. This is something that can be compared to the model as one can quite easily find steady state values from the equations in the model.
- Dynamic validation is tests that investigates whether or not any transients and dynamical behavior is as expected from our intuitive understanding of the system.

Note: This validation is performed on the model of one single well. This is done because the part of merging the flows from the wells is simply a matter of adding the flows together - which is considered to be a simple operation, and hence not requiring validation.

³supplemented to drastically decrease the run time

Variable	chosen value
p^{res}	250 bar
GOR	0.22
WC	70%

Table 3.5: Variables chosen for steady state validation of well model

Variable	Matlab implementation	Maple calculation	difference (%)
p^{BH}	197.91	197.91	0
p^{WH}	95.502	95.502	0
$q_c^{WH} = q_c^{BH}$	3.8987	3.8987	0
ρ_m	521.95	521.95	0
x_2	1.1744	1.1744	0

Table 3.6: Steady state validation. Comparing results from calculation of the steady state equations (performed in maple), and the results from the simulated Matlab implementation.

Steady state validation

As stated earlier, there is no real data, with which to compare the model. This validation will cover whether or not the implemented dynamical model gives the same steady state values, as the equations in the mathematical model presented in this section.

To analyze the steady state value of the set of equations given in this model, \dot{x}_1 and \dot{x}_2 is set to zero, leaving $q_g^{BH} = q_g^{WH}$ and $q_l^{BH} = q_l^{WH} \implies q_c^{BH} = q_c^{WH}$.

Furthermore in steady state the relationship between x_1 and x_2 is known and constant, which means that x_1 should not be counted as an individual independent variable, but rather as a function of x_2 .

$$x_1 = x_2 \cdot \text{GLR}$$

For the steady state analysis 5 variables (p^{BH} , p^{WH} , q_c^{BH} , ρ_m , x_2) calculated from equations ((3.2a), (3.4c), (3.4d), (3.4a), (3.4b)) are compared with the corresponding values found from simulation of the implemented model. For the steady state analysis, the parameters are given as in Table 3.4 and 3.5.

The results of the steady state comparison is shown in Table 3.6. These results cannot verify the equations, but the steady state behavior of the implemented dynamical simulator is now validated compared to the steady state solution of the equations in 3.1.

Dynamical validation

In the dynamic validation, two scenarios are tested (decrease in valve opening, and decrease in reservoir pressure). The comparison between the simulated- and expected are

Change in conditions	Expected outcome	Actual outcome from the simulation
Ramp decrease in valve opening. Total decrease of 50%	Greater pressure drop over valve \implies increase in p^{WH} . Decrease in q_c^{WH} , and similar decrease in q_c^{BH} with some delay. Increase of p^{BH} expectation (because of equation (3.2a)). Increase of $x_2 \implies$ increase of ρ_m , (because of equation (3.4c))	Everything did correspond with the expectations, as can be seen from Figure 3.4
Ramp decrease in reservoir pressure. Total decrease of 10%	Decrease in p^{BH} , decrease in q_c^{BH} \implies decrease in q_c^{WH} decrease in $x_2 \implies$ decrease in ρ_m . Decrease in p^{WH} (because of (3.4c))	Everything did correspond with the expectations. The results are shown in Figure C.1, which can be found in the appendix.

Table 3.7: Dynamical validation, expected dynamical effect, and actual effect from simulation

discussed in Table 3.7, and the simulated results are shown in Figure 3.4 and C.1. To conclude, it can be said that with this implementation of the model it was not observed any unexpected phenomena, hence it seems reasonable to state that the dynamics of this simulator has been validated.

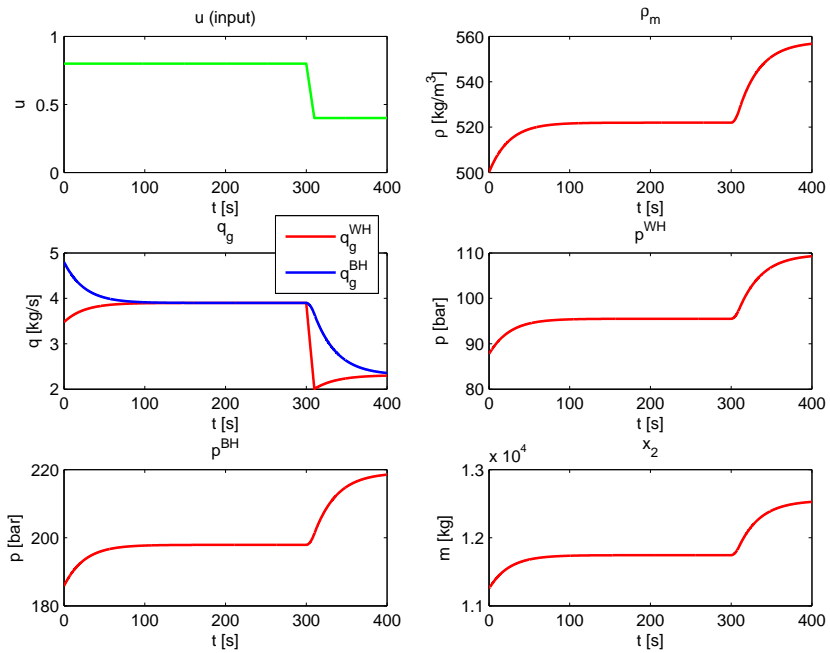


Figure 3.4: Dynamical validation. The results from simulation of the implemented model.

Estimation

Buildup of the simulator was explained in the previous chapter. This simulator is meant to represent a real life system, and is created to produce realistic measurement data for testing of the state estimators and fault diagnosis tools that are created in this thesis. This chapter describes the buildup of the observers created for this thesis, with a primary focus on the Moving Horizon Estimator (**MHE**).

The system is defined with the following measurements

$$\mathbf{y}_k = \left[p_k^{WH,i}; p_k^{BH,i}; q_{g/l,k}^l \right] \quad (4.1)$$

In addition, temperatures can usually be measured for such a system. This is, however, not used in the equations of our model, and hence not included in our measurement vector \mathbf{y} .

4.1 Extended Kalman filter equation

This thesis has a focus on using MHE as state estimator and using some of the information from the output of that estimator to detect model errors using fault diagnosis algorithms. For the purpose of showing that any state estimator, also state estimators not based on MHE, can be applied with the fault diagnosis algorithm, an Extended Kalman filter has been implemented. EKF is a natural choice for the alternative estimator as it is relatively easily implemented, and it is a widely known state estimator (Kang et al., 2013).

In the implementation of the Kalman filter in this thesis it makes sense to separate the states into two categories – algebraic and differential states (respectively $\boldsymbol{\xi}_k$ and \mathbf{x}_k) (Kühl et al., 2011). Where the algebraic states ($\boldsymbol{\xi}$) are only set based on algebraic relations, while the differential states (\mathbf{x}_k) enters differential equations as well. The system can then be

presented with the following formulation

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \boldsymbol{\xi}_k, u) \\ 0 &= g(\mathbf{x}_k, \boldsymbol{\xi}_k, u) \\ \mathbf{y}_k &= h(\mathbf{x}_k, \boldsymbol{\xi}_k, u) \end{aligned} \quad (4.2)$$

In the formulation of (3.6) this would imply that

$$\mathbf{x}_k = [x_{1,k}^i; x_{2,k}^i]$$

$$f = \begin{bmatrix} \text{eq (3.6i)} \\ \text{eq (3.6j)} \end{bmatrix}, \quad \boldsymbol{\xi}_k = \begin{bmatrix} q_{g/l,k}^{BH,i} \\ \bar{w}^{WH,i} \\ q_{c/g/l,k} \\ \bar{w}^{WH,i} \\ p_k^{BH,i} \\ p_k^i \\ \rho_{m,k}^i \\ q_{g/l,k}^l \end{bmatrix}, \quad g = \begin{bmatrix} \text{eq (3.6a)} \\ \text{eq (3.6b)} \\ \text{eq (3.6c)} \\ \text{eq (3.6d)} \\ \text{eq (3.6e)} \\ \text{eq (3.6f)} \\ \text{eq (3.6g)} \\ \text{eq (3.6h)} \\ \text{eq (3.6k)} \\ \text{eq (3.6l)} \end{bmatrix}, \quad h = \begin{bmatrix} p_k^{WH,i} \\ p_k^{BH,i} \\ q_{g,k}^l \\ q_{l,k}^l \end{bmatrix}$$

From Section 2.1.1 it is known that an EKF is built up with the following set of equations

$$\begin{aligned} K_k &= P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \\ \hat{x}_k &= K_k (y_k - c(x_k)) \\ P_k &= (I - K_k C_k) P_k^- \\ P_{k+1}^- &= A_k P_k A_k^T + Q_k \\ \hat{x}_{k+1}^- &= f(x_k, \boldsymbol{\xi}_k(x_k), u) \end{aligned}$$

Where R_k and Q_k are chosen covariance matrices (chosen time invariant in this case) for the measurement noise v_k and process noise w_k respectively

A_k and C_k is given as in (2.7), which for this system makes

$$\begin{aligned} A_k &= \left. \frac{df(\mathbf{x}_k, \boldsymbol{\xi}_k(\mathbf{x}_k), u_k)}{d\mathbf{x}_k} \right|_{\hat{x}_k} \\ &= \begin{bmatrix} 1 + \left(\frac{dq_{g,k}^{BH}}{dx_1} - \frac{dq_{g,k}^{WH}}{dx_1} \right) \cdot \Delta t & \left(\frac{dq_{g,k}^{BH}}{\partial x_2} - \frac{dq_{g,k}^{WH}}{\partial x_2} \right) \cdot \Delta t \\ \left(\frac{dq_{l,k}^{BH}}{dx_1} - \frac{dq_{l,k}^{WH}}{dx_1} \right) \cdot \Delta t & 1 + \left(\frac{dq_{l,k}^{BH}}{\partial x_2} - \frac{dq_{l,k}^{WH}}{\partial x_2} \right) \cdot \Delta t \end{bmatrix} \\ C_k &= \left. \frac{dh(\mathbf{x}_k, \boldsymbol{\xi}_k(\mathbf{x}_k))}{d\mathbf{x}_k} \right|_{\hat{x}_k} \end{aligned}$$

This set of equations are then implemented in a similar way to the loop in Figure 2.2.

4.2 Moving horizon estimation

Moving Horizon Estimation (**MHE**) is a strong optimization-based estimation technique, which was presented in the literature review (Section 2.1). In this thesis two variations of MHE has been implemented.

- The first, which is also the one that was indicated in the problem task, is a MHE where the system of equations has been given some degree of freedom by introducing parameters for process noise (model error). The MHE will then minimize the sum of the estimated measurement error and the estimated model error.
- The second approach comes as a response to the wish for a more intuitive approach to the fault identification discussed in Section 2.2. It is a MHE that estimates the parameters¹ in the system along with the states.

These parameter estimates, work in the same way as the model error parameters in the sense that they provide some slack in the estimation for potential errors in the model. The difference is how the slack is implemented in the two. These two implementations are presented in the remainder of this section.

4.2.1 MHE with slack for model error

The first MHE in this project is implemented with the general formulation given as

$$\min_{\mathbf{x}, \mathbf{y}^{MH}, \mathbf{v}, \mathbf{w}, \mathbf{x}_0} \|\mathbf{v}\|_M^2 + \|\mathbf{w}\|_N^2 + \mathcal{Z}(\mathbf{x}_0) \quad (4.3a)$$

s.t.

$$\mathbf{g}(\mathbf{x}, \mathbf{x}_0) = \mathbf{w} \quad (4.3b)$$

$$\mathbf{y}^{MH} = \mathbf{h}(\mathbf{x}, \mathbf{x}_0)$$

$$\mathbf{y}^{MH} - \mathbf{y} = \mathbf{v} \quad (4.3c)$$

$$\mathbf{x} \in \mathbf{X} \quad (4.3d)$$

The formulation given as in (4.3) is a general formulation that is posted as help for understanding the MHE. The actual implementation is in fact very similar to the implemented simulator in Section 3.1, with some extension that is discussed below.

The difference between the optimization formulation of Section 3, and this MHE can be summarized with the following additions:

- Addition of measurements and measurement error: which is achieved in the MHE by the following addition to the problem formulation of (3.6)

$$\mathbf{y}^{MH} - \mathbf{y} = \mathbf{v} \quad (4.4)$$

where \mathbf{y}^{MH} are the estimated measurements, and \mathbf{y} are the actual measurements.

¹parameters that were given by the operator in the first MHE approach

$\mathbf{x} \in \mathbb{R}^{N_{states} \cdot N_H}$	States given as in (3.6)
$\mathbf{y}^{MH} \in \mathbb{R}^{N_y \cdot N_H}$	Estimated measurements given as in (4.1)
$\mathbf{v} \in \mathbb{R}^{N_y \cdot N_H}$	Measurement error
$\mathbf{w} \in \mathbb{R}^{5 \cdot N_{wells} \cdot N_H}$	Model error
$\mathbf{x}_0 \in \mathbb{R}^{N_{wells} \cdot 2}$	The chosen initial value, separated from \mathbf{x} because of its unique position in the arrival cost.
$\mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{x}_0) \in \mathbb{R}^{N_{eqs} \cdot N_H}$	The model equations given in (3.6) where \mathbf{x}_0 is a constant in (3.6), but a variable in the MHE
$\mathcal{Z}(\mathbf{x}_0) \in \mathbb{R}^1$	Arrival cost
N_H	Horizon of the MHE
N_{states}	# of states, in this case is $10 \cdot N_{wells} + 2$.
N_y	# of measurements, in this case $2 \cdot N_{wells} + 2$.
N_{eqs}	# of equations, in this case $10 \cdot N_{wells} + 2$.

Table 4.1: Explanation of variable in (4.3), note that these vectors each include every time step of the estimation horizon.

- Addition of the model error parameters $w_{j,k}^i$: where $i :=$ well number, $j :=$ equation number, and $k :=$ time instant. It is implemented as shown in the modified (3.6c) below

$$q_{c,k}^{WH,i} = \frac{1}{10} C_c^{WH,i} \sqrt{\rho_m 10^2 (10^2 p_k^{WH,i} - p^{sep})} u_k^i + w_{1,k}^i, \quad \forall i, k \in (t - N_H + 1), \dots, t \quad (4.5)$$

Equal additions of the model error parameters ($w_{2,k}^i, w_{3,k}^i, w_{4,k}^i, w_{5,k}^i$) are done with equation ((3.6g), (3.6a), (3.6b) and (3.6h)) respectively. These equations are chosen to have the slack provided by w as they have the possibilities for mis-calibrations, and/or are approximations, while the remaining relationships (equations) are considered to be exact relationships, and any slack variables on these relationships are considered to be an unnecessary and incorrect addition of degrees of freedom in the system.

- Addition of an objective function, and of initial x_0 as an estimated state:

The objective function implemented in this MHE consists of three elements, namely stage cost on the measurement error (\mathbf{v}), stage cost on the model error (\mathbf{w}) and arrival cost on \mathbf{x}_0 . The arrival cost can, as discussed in the literature review, be approximated in many different ways, the simplest of which being $\mathcal{Z}(\mathbf{x}_0) = \mu \|\mathbf{x}_0 - \mathbf{x}_{T-N_H+1}^{MH}\|$, with $\mu \in \mathbb{R}^1$, and $\mathbf{x}_{T-N_H}^{MH}$ is the estimated state from the previous iteration of the MHE, that correspond to the same time instant as \mathbf{x}_0 . Despite the fact that using (2.10) will most likely lead to a better arrival cost and consequently an increase in the performance of the MHE, the simple formulation is implemented in this thesis. This is done due to the ease of implementation, and the fact that the performance of the state estimator is not of primary concern in this paper. This arrival cost together with the stage costs result in the following objective function that is to be minimized

$$(\Phi_T^* = \min_{\mathbf{z}} \Phi_T)$$

$$\Phi_T = \|\mathbf{v}\|_M^2 + \|\mathbf{w}\|_N^2 + \mu \|\mathbf{x}_0 - \mathbf{x}_{T-N_H}^{MH}\| \quad (4.6)$$

With

$$\mathbf{z} = [\mathbf{x}; \mathbf{x}_0; \mathbf{w}; \mathbf{v}]$$

$$\mathbf{x}_0 = [x_{1,0}^i; x_{2,0}^i]$$

$$\forall i \in \{1, 2, 3, 4\}$$

$$\mathbf{w} = [w_{j,k}^i; w_{11,k}; w_{12,k}]$$

$$\forall i \in \{1, 2, 3, 4\}, k \in L, \dots, t, j \in 1, \dots, 10$$

$$L := t - N_H + 1, \quad t := \text{current iteration of the estimation},$$

$$N_H := \text{length of MHE Horizon}$$

and M and N are chosen weighting matrices for \mathbf{v} and \mathbf{w} respectively.

Furthermore it should be noted that for numeric reasons the equations have been scaled to be of approximately the same magnitude.

4.3 Moving horizon estimation with parameter estimation

The MHE formulation presented in this section is, as previously indicated, not created primarily for its ability to estimate the states (\mathbf{x}), but rather the adjoining parameter estimates (\mathbf{p}). The system parameters estimated with this approach, is experienced by the author to give a more intuitive approach to the detection- and identification of faults that is presented in Chapter 5. The theory that this estimator is based on is given in Kühl et al. (2011).

This formulation of the MHE has many common elements with the formulation stated in Section 4.2.1. The main distinctions are; removal of \mathbf{w} both from the constraint equations ((3.6c), (3.6g), (3.6a), (3.6b), (3.6h)) and from the objective function (4.6), reformulation of the parameters from given constants to estimated "states", (which also entails a small addition to the objective function to smoothen the parameter estimations (Kühl et al., 2011)).

Note: the removal of all the model error parameters (\mathbf{w}) is not necessarily the best approach (Kühl et al., 2011). Nevertheless it is the approach applied in this thesis. This was chosen to limit the number of parameters that the model error could be distributed to. However, there might be possible to achieve better results by including some additional slack through \mathbf{w} , but this has not been tested in this thesis.

And the actual changes from the optimization formulation presented in 4.2.1 are:

- Removal of all the \mathbf{w} 's as stated earlier

- Addition of \mathbf{p} as an optimization variable in equation (3.6c), (3.6a) and (3.6b), as presented below:

$$q_{c,k}^{WH,i} = \frac{1}{10} C_c^{WH,i} \sqrt{\rho_m 10^2 (10^2 p_k^{WH,i} - p^{sep})} u_k^i$$

$$q_{l,k}^{BH,i} = \frac{1}{10} P I^i (p^{res,i} - 10^2 p_k^{BH,i}), \quad \forall i, k \in L, \dots, t \quad (4.7a)$$

$$q_{g,k}^{BH,i} = \text{GLR}^i 10 q_{l,k}^{BH,i} \quad \forall i, k \in L, \dots, t \quad (4.7b)$$

$$(4.7c)$$

- Corresponding change in the Jacobian function, which for the purpose of keeping the paper somewhat tidy, has been placed in appendix B.1
- Addition of \mathbf{p} to the objective function:

$$\Phi_T = \|\mathbf{v}\|_M^2 + \mu_p \|\mathbf{p} - \mathbf{p}^{MH}\| + \mu \|\mathbf{x}_0 - \mathbf{x}_{T-N_H}^{MH}\| \quad (4.8)$$

Where \mathbf{p}^{MH} is the previously estimated \mathbf{p}

The new optimization variables \mathbf{z} are given as

$$\mathbf{z} = [\mathbf{x}; \mathbf{x}_0; \mathbf{p}; \mathbf{v}]$$

$$\mathbf{p} = [C_c^{WH,1}, \dots, C_c^{WH,N_w}, \text{PI}^1, \dots, \text{PI}^{N_w}, \text{GLR}^1, \dots, \text{GLR}^{N_w}]^T$$

$$N_w := \# \text{ of wells}$$

with $\mathbf{p} \in \mathbb{R}^{N_w \cdot 3}$.

4.3.1 The choice of MHE parameters

Choosing the weighting matrices and scalars in the objective function of the MHE's is essentially a tuning problem. As the actual performance of the the estimators are outside the scope of this thesis, this tuning has been conducted fairly roughly. However this section presents how the MHE parameters N_H , μ and μ_p for the MHE with parameter estimation were chosen.

- The estimation horizon: N_H determines how far back one evaluates the measurements². The longer the horizon is, the more information is used as basis for the state and parameter estimation, which in turn usually implies better estimations. However increased N_H leads to increased computation time, and it weakens the assumption of constant model parameters (\mathbf{p}) during the estimation horizon.

As a rule of thumb for a MPC (Model Predictive Control) prediction horizon, the horizon need to be at least 2-3 times the size of the dominant time constant (Love, 2007). Due to the similarities of the optimization problems in MHE an MPC, this rule of thumb is assumed transferable to the MHE problem. Based on this the dominant time constant for the system were found in to be approximately $\tau_{max} = 50\text{s}$

²The lost information from the any time step before this, is attempted summarized in the arrival cost.

(see Figure 4.1). As can be seen from Figure 4.2, however, it seems that a smaller horizon is prudent for this particular case. For the purpose of the experiments conducted in this thesis, N_H is chosen to be:

$$N_H = 20s$$

- The weights μ and μ_p : These two parameters are indicators on how certain one is that the last estimate of x_0 and p were correct, and the cost to deviate from these estimates. Figure 4.3 and 4.4 show an analysis of these parameters which led to the following choice for parameters $\mu := 1$, $\mu_p := 2$. Tests were not conducted with such a scientific approach for the weighting matrix M , it was chosen based on the degrees of importance of the different measurements in the system.

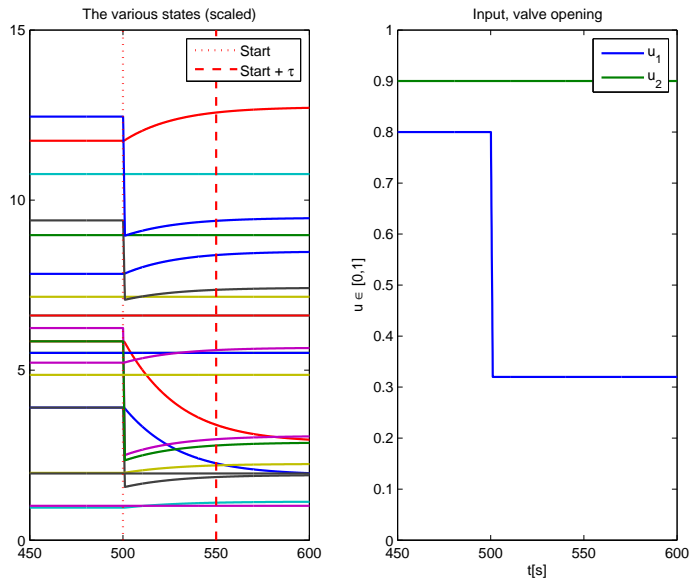


Figure 4.1: An illustration of how the time constant was determined.

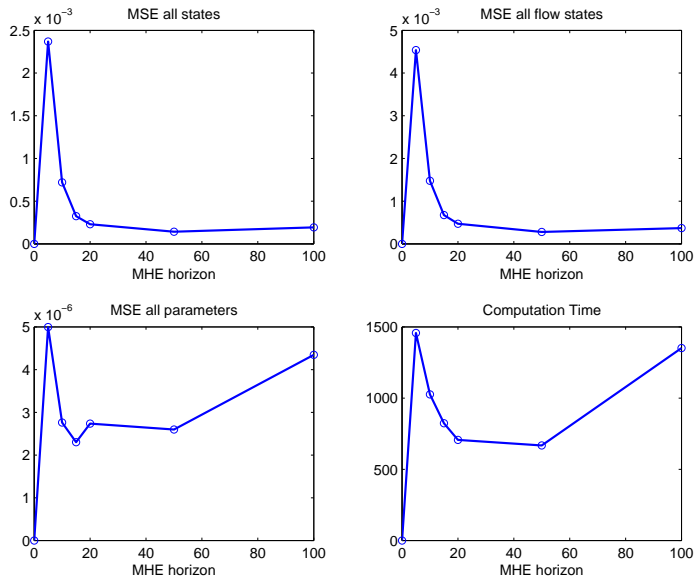


Figure 4.2: Plots showing the performance of the state and parameter estimator as a function of N_H , given good initial guesses on the parameters and x_0 , and some added "measurement" noise.

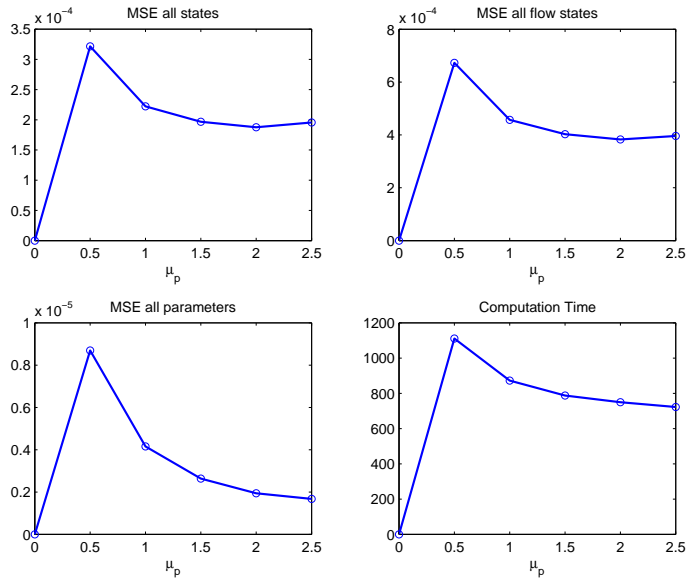


Figure 4.3: Plots showing the performance of the state and parameter estimator as a function of μ_p , given good initial guesses on the parameters and x_0 , and some added "measurement" noise.

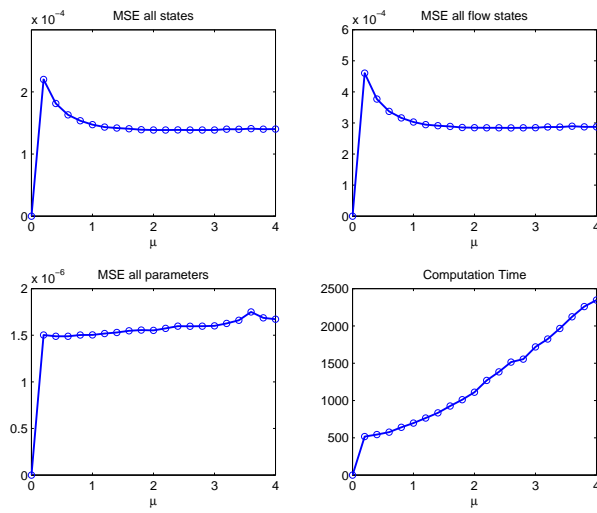


Figure 4.4: Plots showing the performance of the state and parameter estimator as a function of μ , given good initial guesses on the parameters and x_0 , and some added "measurement" noise.

Chapter 5

Fault Diagnosis

This chapter presents the fault diagnosis systems used in this thesis. Two approaches to the fault diagnosis are developed, where the main difference is the choice of test-variables¹, as discussed in Chapter 4. The assignment text was originally stated with an MHE consisting of model error parameters (\mathbf{w}) as described in Chapter 4. Through the development of this thesis, however, it was decided by the author that it is more intuitive to use parameter estimates as test-variables. This is further discussed in Section 7.1. Both approaches are developed and described in this section, however the implementation of the parameter-based fault diagnosis system, is conducted with more care, as this is the primary fault diagnosis system of this thesis. Figure 5.1 illustrates how the fault diagnosis system is implemented in the complete VFM system. Note that in this illustration, the residual(test-variable) generator is defined as the parameter estimator, however the approach with model error parameters (\mathbf{w}) as test-variables, is implemented with the VFM in exactly the same way.

¹also described as residuals

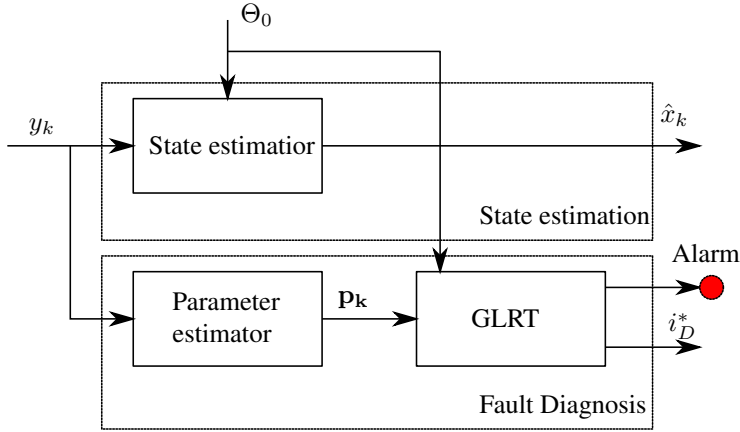


Figure 5.1: Illustration of the setup for the data flow used in most of this thesis. Notice how the state estimation is parallel to (independent of) the fault diagnosis despite the fact that the parameter estimator is usually a state estimator as well. $i_D^* :=$ fault ID, $\Theta_0 :=$ parameters provided to the state estimator by the operator.

5.1 Fault detection

Fault detection is, as explained in Section 2.2, the process of determining whether or not there is reason to think that a fault has occurred. To do this, a test statistic $g(k)$ is calculated based on a "known" distribution of the test-variables (model error parameters (\mathbf{w}) or system parameters (\mathbf{p})). In this paper, the test-variables (both \mathbf{p} and \mathbf{w}) are approximated to have a multivariate normal distribution², which has the PDF:

$$f(x, \mu, S) = (2\pi)^{-k/2} |S|^{-1/2} e^{-1/2(x-\mu)^T S^{-1}(x-\mu)} \quad (5.1)$$

Using this in the equation (2.18) and solving for \max_{θ_1} one gets the equation for $g(k)$:

$$g(k) = \max_{k-N_D+1 \leq j \leq k} \sum_{i=j}^k (z(i) - \mu_1)^T S^{-1} (z(i) - \mu_1) - \sum_{i=j}^k (z(i) - \mu_0)^T S^{-1} (z(i) - \mu_0) \quad (5.2)$$

$$\mu_1 = \frac{1}{k-j+1} \sum_{i=j}^k z(i) \quad (5.3)$$

Derivation. of equation (5.2) and (5.3)

²Note that if one wish to approximate the test-variables as a t-distribution, look to Willersrud et al. (2014a)

From 2.18 it is known that:

$$S_j^k(\theta_1) = \sum_{i=j}^k \log \frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}$$

$$g(k) = \max_{k-N_D+1 \leq j \leq k} \max_{\theta_1} S_j^k(\theta_1)$$

inserting $f(x, \mu, S)$ from (5.1), for $p_{\theta}(z(i))$ in equation (2.17), with the changing component of the distribution being μ . This gives:

$$g(k) = \max_{k-N_D+1 \leq j \leq k} \log \frac{\max_{\mu_1} \prod_{i=j}^k f(z(i), \mu_1, S)}{\prod_{i=j}^k f(z(i), \mu_0, S)}$$

$$= \max_{k-N_D+1 \leq j \leq k} \log \frac{\max_{\mu_1} e^{-\frac{1}{2} \sum_{i=j}^k (z(i) - \mu_1)^T S^{-1} (z(i) - \mu_1)}}{e^{-\frac{1}{2} \sum_{i=j}^k (z(i) - \mu_0)^T S^{-1} (z(i) - \mu_0)}} \quad (5.4)$$

Furthermore the property of \max_{μ_1} can be found analytically by the following relationship:

$$\max_{\mu_1} e^{-\frac{1}{2} \sum_{i=j}^k (z(i) - \mu_1)^T S^{-1} (z(i) - \mu_1)}$$

$$\implies \min_{\mu_1} \sum_{i=j}^k (z(i) - \mu_1)^T S^{-1} (z(i) - \mu_1)$$

$$\implies \mu_1^* = \frac{1}{k-j+1} \sum_{i=j}^k z(i)$$

Using this μ_1 one has eliminated the optimization property \max_{μ_1} from (5.4), and after further simplification through the use of logarithmic properties:

$$\log \frac{e^a}{e^b} = \log(e^a) - \log(e^b) = a - b$$

One gets

$$g(k) = \max_{k-N_D+1 \leq j \leq k} \sum_{i=j}^k (z(i) - \mu_1)^T S^{-1} (z(i) - \mu_1)$$

$$- \sum_{i=j}^k (z(i) - \mu_0)^T S^{-1} (z(i) - \mu_0)$$

$$\mu_1 = \frac{1}{k-j+1} \sum_{i=j}^k z(i)$$

as stated in (5.2). □

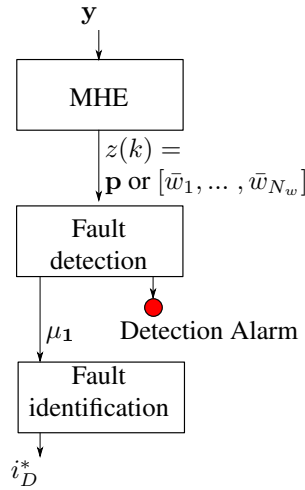


Figure 5.2: A schematic illustrations of the steps in the fault diagnosis system used in this thesis.

5.2 Implementation of fault diagnosis using w as test-variables

The choice of test-variables does not change the actual implementation of the fault detection and identification much. The detection is implemented with the test statistic from (5.4), combined with the test variables $z(t) = \bar{w}$, where $\bar{w}_i :=$ mean of the model errors over the horizon of the MHE, at the i -th iteration of the moving horizon estimator. A threshold h for the hypothesis test were found through trial and error. A more elegant approach for finding h is explained in 2.2.2 and implemented for the parameter estimation approach. Due to limited time, however, this was not done with this approach. The covariance matrix S of w is approximated to be a diagonal matrix.

Note: this approximation of zero covariance between the model errors is far from the case³, but a simplification to simplify the implementation.

5.2.1 Implementation of the identification signatures

As stated above, it was decided that using test-variables from parameter estimation, would provide a more intuitive fault identification process. It does, however, not imply that fault identification cannot be done with \bar{w} as test-variables. This section presents the signatures in \bar{w} from the different "faults". These relations are found through tests, by asserting the faults on the system, and documenting the corresponding test-variables \bar{w} . The signatures which are used to create Γ and K_D for (2.19), are posted in table 5.1 and 5.2.

³due to smearing, as discussed in 2.2.3

	ΔPI	ΔGLR	ΔC_C
w_1^1	+	+	+
w_1^2	+	+	+
w_2^1	+	+	-
w_2^2	-	-	-
w_3^1	+	o	+
w_3^2	o	o	+
w_4^1	o	+	-
w_4^2	+	-	-
w_5^1	+	+	-
w_5^2	-	o	-

Table 5.1: Direction of change given an increase of the respective parameters (changes in well 1). In the case of decrease, this table is negated (approximately true, however, it does not seem to be the truth always). Note that this table comes as a result of tests. This table is to be used for the creation of Γ_D in (2.19)

w_i	weighting
w_1^1	0.5
w_1^2	0.5
w_2^1	1
w_2^2	1
w_3^1	10
w_3^2	10
w_4^1	10
w_4^2	1
w_5^1	1/2
w_5^2	1/2

Table 5.2: Approximate relative weighting, based on test runs with faults. This table is to be used for the creation of K_D in (2.19)

5.3 Implementation of fault diagnosis using \mathbf{p} as test-variables

In this section the implementation of the fault diagnosis system using test-variables from parameter estimation is presented. The difference between the two approaches are quite limited. Both are based on fault detection through GLRT, and fault identification through the technique presented in Section 2.2.3. However, a more thorough analysis are done with regards to determine the distribution of the test-variables \mathbf{p} and test-statistic $g(k)$, for this approach (see Section 7.1 for a discussion on why this implementation of fault diagnosis has been prioritized). The results from this distribution analysis are shown in Figure 5.3 and 5.4, and are applied in the choice of S in (5.4) and in the calculation of h from (2.16). Figure 5.3 illustrates the distribution of the error in the estimated parameters,

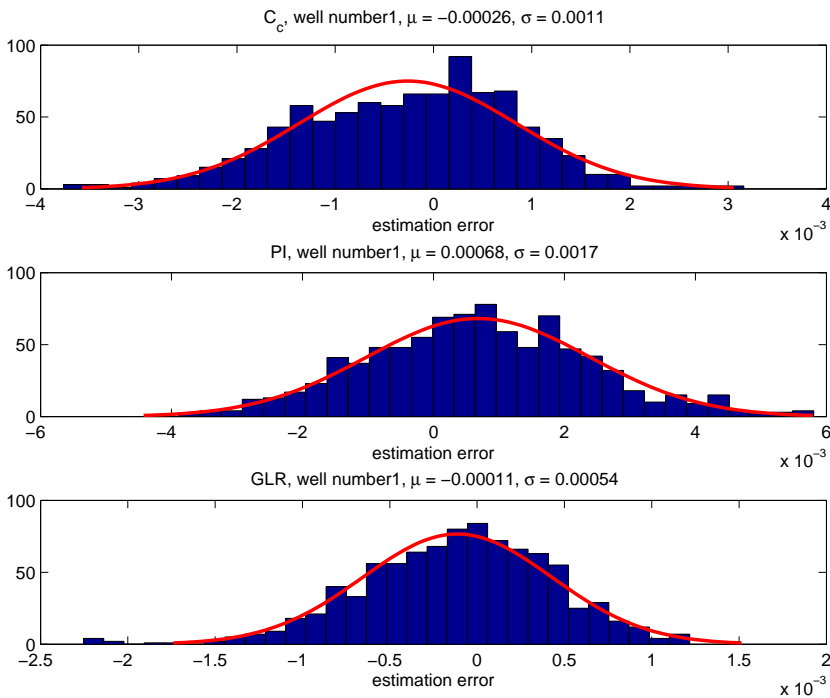


Figure 5.3: Distribution of $\tilde{p} = p - p^*$ (the parameter estimation error). This distribution is found through 5 runs of the of parameter estimation with fairly good initial guess of parameters and normally distributed measurement noise.

found through several simulations with added measurement error. These distributions were the best fit to the datasets retrieved from the simulations, however, for a nicer distribution it is approximated as zero mean normal distribution ($\mu_{\tilde{p}} = 0$) and the standard deviations (σ) are assumed to be a bit bigger than what is given in the figure. The chosen σ 's are shown in Table 5.3.

	C_{pc}^i	PI^i	GLR^i
σ	0.0015	0.002	0.0006

Table 5.3: Chosen values for σ , based on the analysis from Figure 5.3

5.3.1 Determining a good value for the threshold (h)

This section is added to support the choice of the threshold (h). The threshold is important as it defines how sensitive the GLRT is. This threshold is found through the probability distribution of $g(k)$ and a desired P_{FA} . Where P_{FA} is the probability of false alarm, and the probability distribution of $g(k)$ is approximated through Monte Carlo simulations, as shown in Figure 5.4.

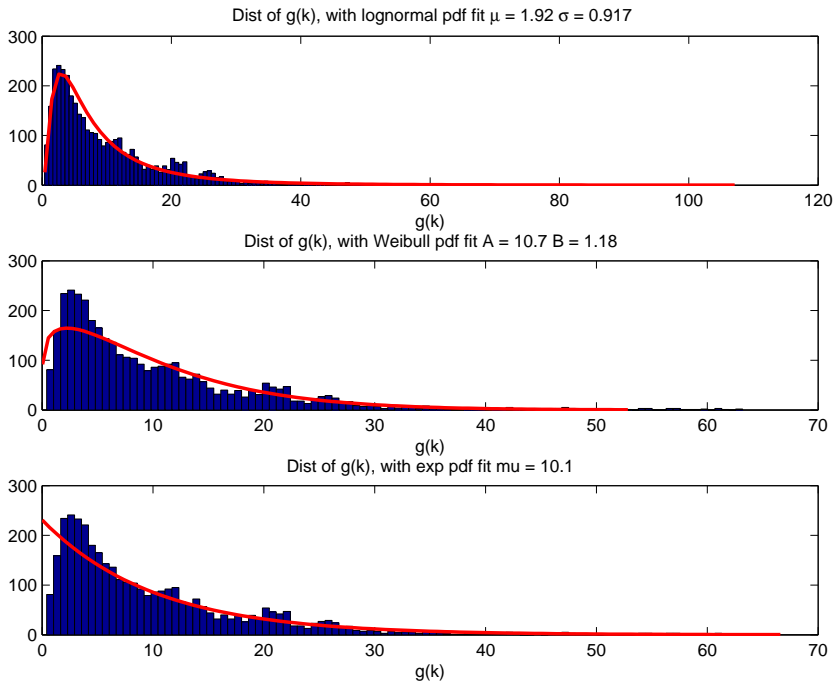


Figure 5.4: Distribution of $g(k)$, which is used to determine a good threshold (h). Log-normal distribution was found to be the best fit.

The test statistic $g(k)$ was found to be distributed similar to a log-normal probability distribution, with the parameters presented in Table 5.4. For a log-normal distribution, the

μ	σ
1.92	0.92

Table 5.4: Parameters chosen for the log-normal PDF of the test statistic $g(k)$, given a GLRT horizon (N_D) of $N_D = 10$ (see Section 2.2.2).

P_{FA}	h
5%	30.9
1%	57.7
0.1%	116

Table 5.5: Selection of h given various P_{FA}

PDF is given as: (Walpole, 2012)

$$f(x; \mu, \sigma) = \begin{cases} \frac{1}{\sqrt{2\pi} \sigma x} e^{-\frac{1}{2\sigma^2} [\ln(x) - \mu]^2}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Unlike with a Weibull distribution, where one can determine an explicit function for h by inserting $f(x; \mu, \sigma)$ into (2.16) (Willersrud et al., 2014a), it is not possible to do the same with a log normal distribution as $F(x; \mu, \sigma)$ is a function of the Gauss error function ($-\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$). However, h still has a unique value that solves equation (2.16), and this value is found in this thesis through the Matlab function $icdf(PD, 1 - P_{FA})^4$. Some samples that was found for h at different P_{FA} , are presented in Table 5.5.

⁴With, PD, being a probability distribution object.

Numerical Study

This chapter covers the experiments performed to test the theory in this thesis. It consists of two main sections, one to describe the different test cases and the motivation behind them. The other section presents the results of those tests. To summarize the information in this rather long chapter Table 6.1 and Table 6.6 have been created for the test case section and the result section respectively.

6.1 Presentation of the test cases

This thesis consists of two main topics, namely

- State estimation for the purpose of virtual flow metering
- Statistical fault diagnosis for the purpose of detecting poorly calibrated models.

The work done with state estimation in this thesis is not new (Grimstad et al., 2015). For that reason, testing the performance of the state estimators created in this thesis would give little new insight – and hence, no such tests have been included in this thesis. The focus of this chapter has instead been to test the use of statistical fault detection and identification techniques to detect poorly calibrated parts of the model. This is a field that, to the authors knowledge, has not been done with virtual flow metering for subsea production systems before.

All the tests in this chapter, except for test case 1, were performed on fault diagnosis using MHE with parameter estimation (see Chapter 5 for more information). Test case 1 was performed on fault diagnosis using MHE with model error parameters (w), however this approach was decided to entail a less intuitive approach for fault identification. Only one test was therefore performed, to test the possibility of doing fault diagnosis with this approach. The rest of the tests were performed using MHE with parameter estimation as residual generator.

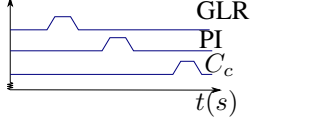
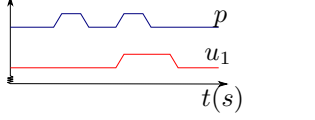
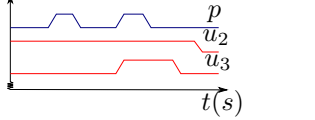
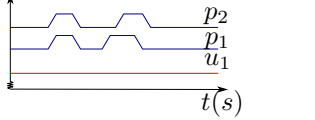
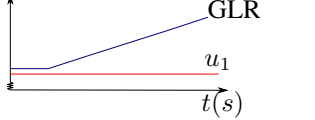
CaseId	Description	Input Sequence
1	Fault diagnosis using w as test-variable	
2a 2b 2c	Only error in well 1 <ul style="list-style-type: none"> • $p_{err} = \text{GLR}$ • $p_{err} = \text{PI}$ • $p_{err} = C_c$ 	
2d	Only error in well 1 <ul style="list-style-type: none"> • C_c 	
3a 3b 3c	Error in well 1 <ul style="list-style-type: none"> • $p_1, p_2 = C_c, \text{GLR}$ • $p_1, p_2 = \text{PI}, C_c$ • $p_1, p_2 = \text{GLR}, \text{PI}$ 	
4	Realistic use case, slow increase of GLR well 1. With re-calibration 4 times.	

Table 6.1: Presentation of the test cases performed in Chapter 6

See Section 7.2.1, for further information on why the approach of test case 1 has been considered to be less intuitive.

Note that throughout this chapter, all changes in the parameters are termed a fault. What's being tested is actually the possibility to detect poorly calibrated models. The reason why a change in a parameter is labeled as a fault, is that such a change in a parameter implies that parts of the model in the state estimator, become less accurate, and hence needs to be detected as poorly calibrated.

6.1.1 Fault diagnosis using w as test-variables

This section presents a superficial study of fault diagnosis using w (from the formulation in (4.2)) as test-variable (see Chapter 5). This approach to fault diagnosis on an oil and gas production system, has been given less priority in this thesis as presented above. This is mainly due to the fact that, for this approach to work, a fault signature matrix Γ_D , must be selected. No viable approaches have, however, been identified for finding such a signature matrix Γ_D .

The one test that was performed using this approach, has been given a Γ_D that was determined through testing (as described in Section 5.2.1), which is a viable approach when one is dealing with a simulator, but not as viable for a real system. (This is discussed further

in Section 7.2.1)

Test case 1

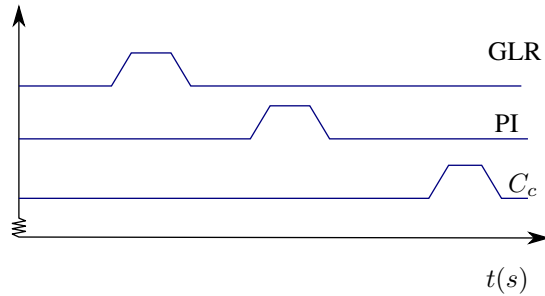


Figure 6.1: Test case 1: Illustration of input sequence

What: This test is a collection of sequential errors happening on the parameters in the system (see figure in Table 6.1). As this approach to fault diagnosis has not been the focus of this thesis, some simplifications have been done to this test.

1. It is assumed previously determined that the fault occurred in well 1. This assumption is based on the fact that, although it has not been done by the author, it seems reasonable to believe (based on a superficial analysis of w) that it is possible to pinpoint the well in which the incident occurred.
2. To simplify the system the number of wells has been reduced from 4 to 2.
3. No changes in the valve openings (u_i) during the test.

This test has been executed with the parameters shown in Table 6.2, and Γ_D , K_D derived from Table 5.1 and 5.2 (see Section 5.2).

Parameter	Value
Error	-50%
N_D	10
N_{wells}	2
h	0.1

Table 6.2: Test case 1: Parameters chosen. Note that the fractional error on the parameters are equal (-50% on all of them). Error presented in relative change from nominal parameter value.

Why: This test is performed to get an indication on whether or not this approach can be used on a system like this, given that a good signature matrix is provided (which as described above, is not easy). The test consists of three sequential "faults" (one for each parameter [C_c PI GLR]) to test the performance of this approach on each type of fault.

6.1.2 Fault diagnosis using p as test-variables

In this section several tests are performed to test the ability of the fault diagnosis systems based on the estimated parameters $\mathbf{p} = [C_c \text{ PI} \text{ GLR}]$. Using this approach to fault diagnosis, several tests are performed to analyze the algorithm's ability to detect and identify the faults. The tests are set up to test and highlight different properties and potential flaws in the implemented fault diagnosis system. Note that the goal of these tests are not to evaluate the performance of this specific implementation, but to map the possibilities, flaws, and pitfalls of implementing a fault diagnosis algorithm on a system like this. In this section there are three main categories of tests performed, each with a specific purpose:

2. Simple tests with one fault, with or without changes in u
3. More faults at the same time
4. A realistic use case to illustrate the possibility of fault detection and identification.

Test case 2: Basic tests

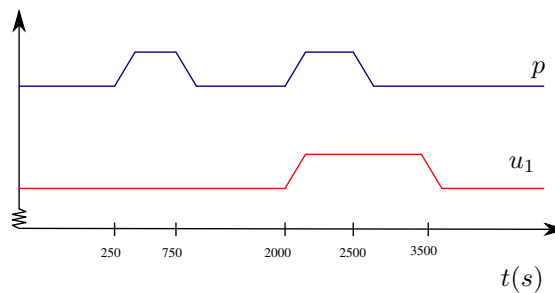


Figure 6.2: Test case 2: Illustration of input sequence. p := the parameter that is being tested.

What: Test case 2 is a set of 3 cases (2 a,b,c) that tests the three respective parameters $[GLR \text{ PI} \text{ } C_c]$. The cases are set up with three main points of interest to test basic capabilities of the complete fault diagnosis system:

1. Test the algorithm's capability of detecting and identifying a fault occurring without any other changes to the system. Time span $250 < t < 750$
2. Test the effect of changes in u (wellhead valve opening) on the test statistic $g(k)$. Time span $3500 < t < 3600$
3. Test the algorithms capability of detecting and identifying a fault occurring while u changes simultaneously. This part is especially important as it is reasonable to think that most changes in the parameters happens together with dynamics in the system. Time span $2000 < t < 2500$

Why: This test investigates the use of the fault diagnosis on fairly basic test cases. The reason for starting with basic tests is the idea that one should start with simple tests, and increase the complexity of the tests until the system no longer performs up to the expected standards – hereby determining what's working, and what is not.

Parameter	Value
Error	+20%
N_D	10
N_{wells}	4
h	116

Table 6.3: Test case 2: Parameters chosen, equal for all of the tests in test case 2. Error presented in relative change from nominal parameter value

Test case 2d: Slightly less basic test

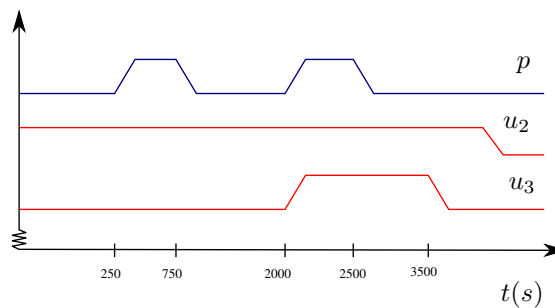


Figure 6.3: Test case 2d: Illustration of input sequence, p := the parameter that is being tested

What: Test case 2d, is a supplement of test case 2, that changes 1 parameter in one well. It differs from 2a-2c, however, as it includes changes in the valve opening of both well 2 and well 3, and not in well 1. In this exact test case, C_c in well 1 is changed simultaneously as a change in the valve of well 2, as shown in Figure 6.3. In addition the valve of well 3 is changed at the end of the horizon.

Why: Several tests with small additions to complicate the basic tests that were performed in Section 6.1.2. This was done to search for any further complication in the fault diagnosis. All of these tests are not presented in this section, but this test is added as it involves some interesting results.

Test case 3: Several faults simultaneously

What: Test case 3 is an extension of test case 2, with similar changes in the parameters. The property that is being tested in this case is how the fault diagnosis algorithm handles several faults in the system at the same time. For the scope of this case, only 2 parameters have been changed simultaneously. This test case is set up with three main points of interest:

1. Simultaneous change in two parameters, Time span $250 < t < 750$

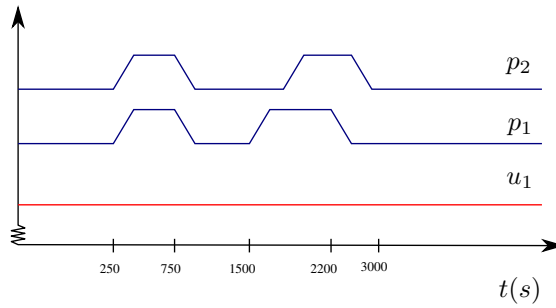


Figure 6.4: Test case 3: Illustration of input sequence. $p_{1,2}$:= the two parameters that is being tested.

2. Change in one parameter followed by a change in the other parameter (before the first parameter is changed back to nominal state), Time span $1500 < t < 2200$
3. Two active "faults" (changed parameters), when one of the parameters is reset to nominal state, leaving only one active "fault". Time span $2200 < t < 3000$

Why: It is already known that the identification algorithm will only identify 1 fault as it simply finds the one fault that seems to be most consistent with the changes in the test variables ($p = [\text{GLR}_i \text{ PI}_i \text{ C}_{c,i}]$). The result of this test is therefore not expected to be an identification of both faults, but rather an experiment to see which one, if any, of the two faults that is identified as the fault.

Parameter	Value
Error	+20%
N_D	10
N_{wells}	4
h	116

Table 6.4: Test case 3: Parameters chosen. Error presented in relative change from nominal parameter value.

Test case 4: Realistic use case simulation

What: The case is based on a continuously increasing parameter (in this case GLR in well 1). This change in the parameter is not known by the state estimator (which assumes the parameters Θ_0 to be true) and hence there will be a degradation of the flow estimates¹. Because of the fault diagnosis system, however, an operator will at some points be informed that there is a high likelihood that the expected parameters Θ_0 are no longer correct. The operator in this example will then follow Procedure 2, to adjust these parameters.

¹ See illustration of how the state estimator in VFM are assumed decoupled from the state- and parameter estimator used in the fault diagnosis system. Figure 5.1

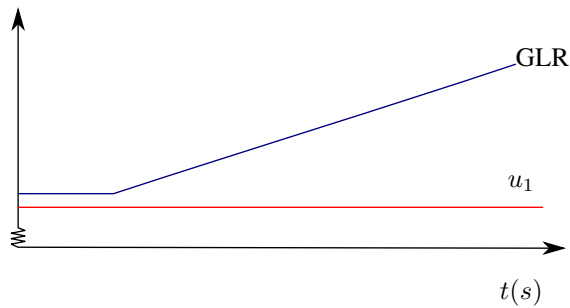


Figure 6.5: Test case 4: Illustration of input sequence

Why: Test case 4, is an example created to test a more realistic use of the fault diagnosis algorithm on a virtual flow metering (VFM) system – as opposed to the previous cases where the tests have been performed to determine weaknesses and strengths of the fault diagnosis system. This test case is more of an example to show the potential of this fault diagnosis addition to VFM, and to test how the system performs in a more realistic situation.

Procedure 2: OperatorProcedure

- 1) Operator is alarmed that one- or several models are probably poorly calibrated;
 - 2) Gets information on which parameter that is poorly calibrated, and an estimated value for that parameter;
 - 3) Waits until the fault diagnosis algorithm has produced 30 consecutive time steps with the same identified parameter;
 - 4) Re-calibrates the parameters Θ_0 with the new estimate on the poorly calibrated parameter (in this case GLR well 1);
-

Parameter	Value
End error	+50%
N_D	10
N_{wells}	4
h	116×1.5

Table 6.5: Test case 4: Parameters chosen. Error presented in the total change from start until the end in percentage of nominal parameter value. Note that the threshold is slightly increased in this test to reduce the frequency of recalibration.

The state estimator applied for the VFM in this case is an Extended Kalman Filter, while the state- and parameter estimator used for the fault diagnosis system is a MHE. This is done to illustrate that the actual implementation of the state estimator is irrelevant to the fault diagnosis system. The state estimator could also have been a MHE.

6.2 Results

This section presents the results of the tests presented in Section 6.1. For a summary of the results presented in this section see Table 6.6.

Test	Main result
1	Detects and identifies all the faults.
2	<ul style="list-style-type: none"> • Faults from changes in GLR are detected and identified correctly • Faults from changes in PI are detected and identified almost correctly, with a drawback that it takes time to "undetect" the fault • Faults from changes in C_c are detected and identified almost correctly, with the exception that it never "undetected" the fault that occurred simultaneously with changes in u • Faults from changes in C_c are detected correctly, but misidentified when there is a simultaneous change in the valve opening of another well. The fault is, furthermore, never "undetected" similar to that of test case 2c.
3	Only one fault identified. Mainly identifies one of the correct parameters that are "faulty". Always return to nominal state after both the parameters are reset to nominal state.
4	Fault diagnosis system detects and identifies fault. This is used efficiently to keep the residual between real and estimated flow at a relatively speaking low level (below 10% of true flow).

Table 6.6: A short summary of the results presented in Section 6.2.

6.2.1 Fault diagnosis using w as test-variables

Test case 1: Test with w as test variables

Figure 6.6 presents the performed test, where $g(k)$ is the test statistic presented in Section 5.1, which eventually detects the faults. It shows that the faults are detected, and identified. There are some delay both before the faults are detected, and from the fault is removed until the algorithm no longer detects a fault. It can also be observed that "faults" in GLR and PI have a much smaller effect on $g(k)$, than that of C_c .

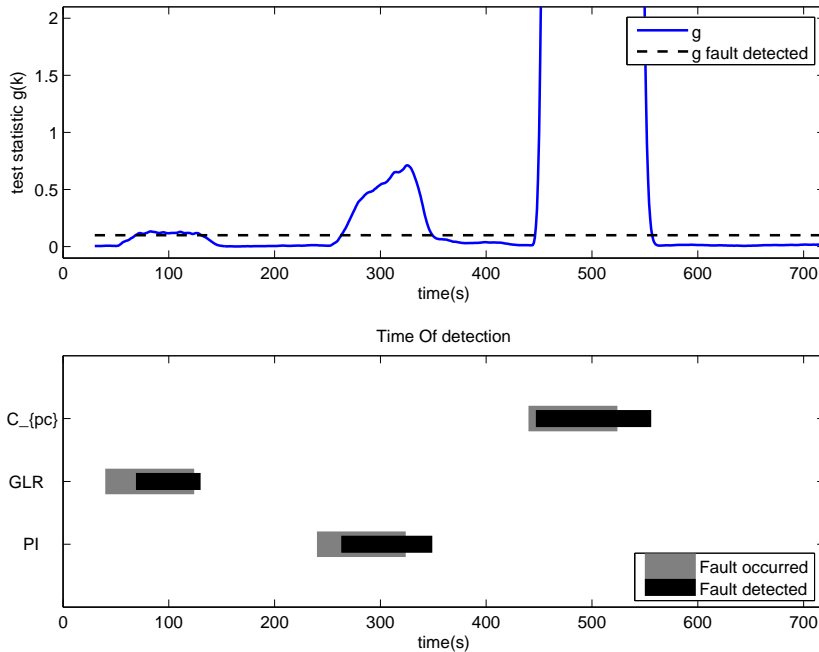


Figure 6.6: Test case 1: A GLR-test run with w as input and the fault identification as presented in Section 2.2.3. Knowledge that the fault is in well number one, is assumed apriori known, in the identification algorithm.

6.2.2 Fault diagnosis using p as test-variables

This section presents the results from test case 2,3,4 as they are all performed based on GLRT with the parameters $p = [C_c \quad PI \quad GLR]$ as test variables.

Test case 2: Basic tests

The results from test case 2 are presented in Figure 6.7 to 6.9, with the addition of Figure 6.10 and 6.11 to further illustrate the results from the tests.

Figure 6.7, 6.8 and 6.9 present plots showing the general performance of the fault diagnosis algorithm, on faults from changes in the three parameters. It is observed that the algorithm in this case detects and identifies faults from changes in GLR well, and fairly well on faults from changes in PI. It also correctly detects and identifies the faults from changes in C_c , but after the simultaneous change in C_c and u the algorithm never "undetected" the fault. It is also observed that the detection algorithm generally is not affected much by changes in u , with the exception of situations when it is combined with changes in C_c . It is also observed that for changes in C_c and PI, the time for "undetected" is fairly long and that in this period the identification algorithm is somewhat unpredictable.

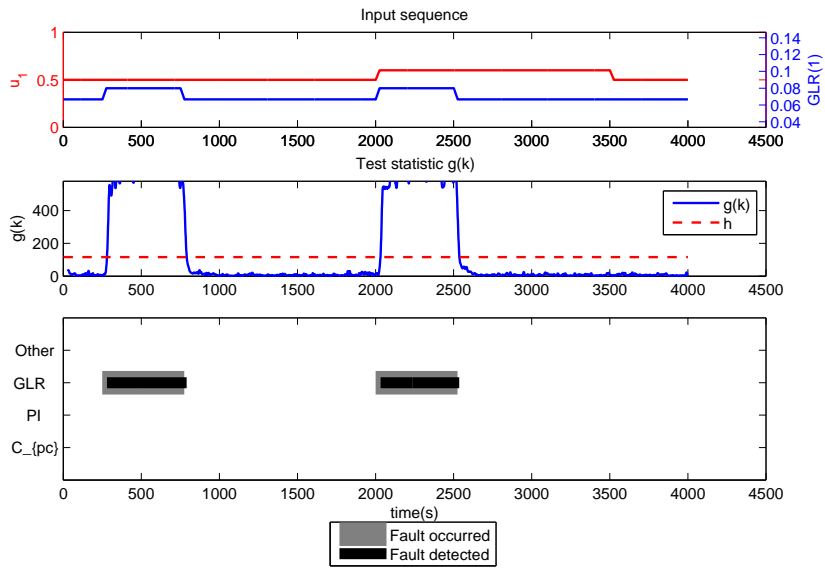


Figure 6.7: Test case 2a: fault in GLR of well one. Note: "other" refers to a detected error in another well.

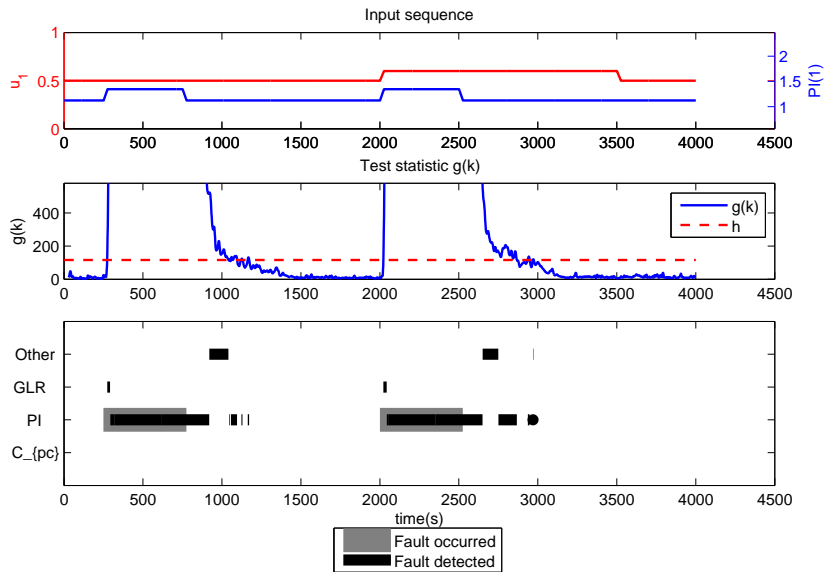


Figure 6.8: Test case 2b: fault in PI of well one. Note: "other" refers to a detected error in another well.

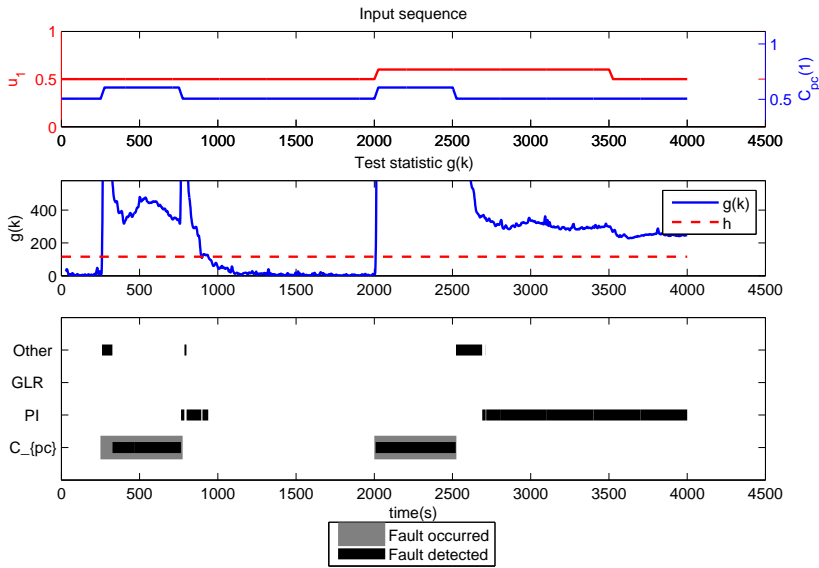


Figure 6.9: Test case 2c: fault in C_c of well one. Note: "other" refers to a detected error in another well.

Figure 6.10 shows that the estimated parameters PI and C_c (of well 1) never return to the real parameters after the simultaneous change in C_c and u . This observation can safely be stated as the reason for the constant fault detection in test case 2c. Figure 6.11 shows how changes in one of the actual system parameters, is reflected upon the estimated system parameters. The main observation from this plot is the correlation between the estimated parameters, and how changes in GLR do not seem to have any significant impact on the estimates of PI, and C_c .

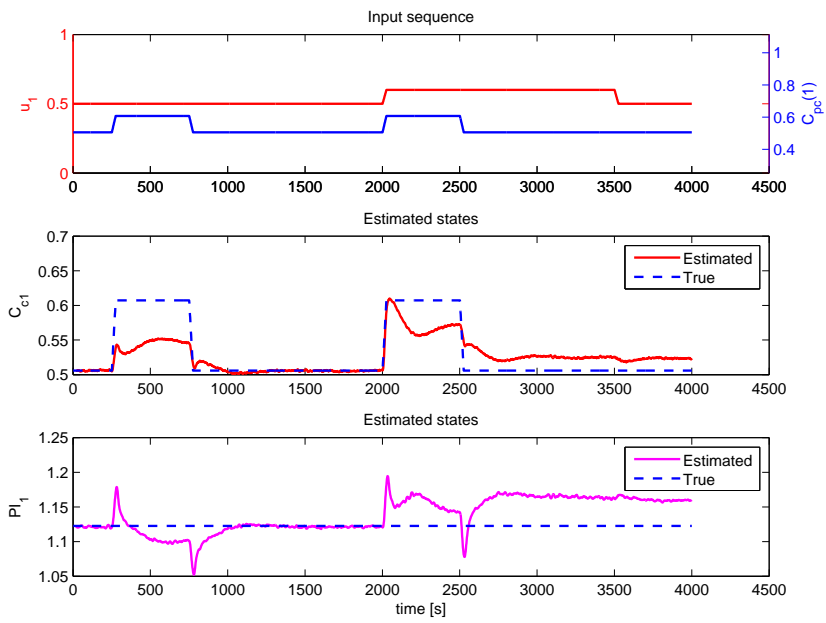


Figure 6.10: Test case 2c supplement: constant error in parameter estimation after simultaneous changes in u and C_c

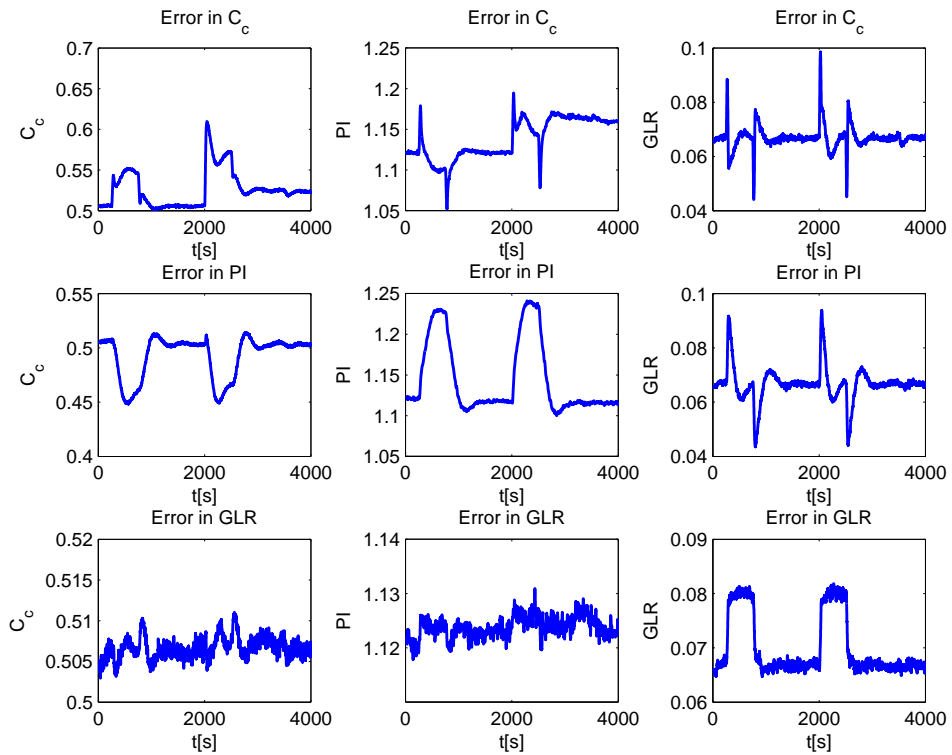


Figure 6.11: Test case 2 supplement: Illustration of the correlation between the estimated parameters

Test case 2d: slightly more complicated addition to test case 2

Figure 6.12 illustrates the slightly more complex test case 2d, which is complicated with additional changes in u in other wells. This exact test has been chosen because it is the only scenario that presents a situation where the fault is in fact completely mis-identified. Several similar tests were also performed, but not included in this thesis, because they did not bring much new information. This test, shows that the identified fault is never unidentified (similar to the results of 2c), in addition it identifies the fault to be associated with the wrong well.

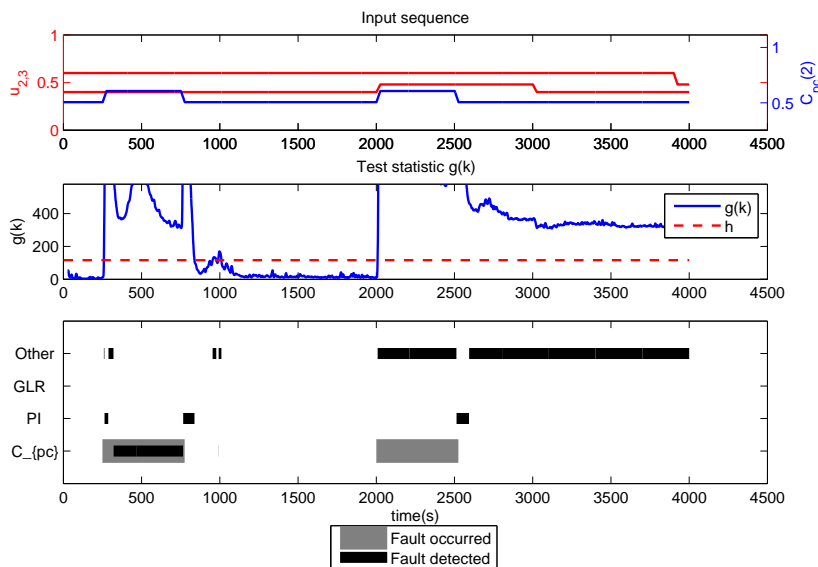


Figure 6.12: Test case 2d: fault in C_c of well 2.

Test case 3: Several faults simultaneously

In this case three different tests have been run (3a, 3b and 3c). Each of the tests did, however, not give much new insight – thus to limit the scope of this section, only the figure illustrating 3a is included in this section (see Appendix C.2 for the other two figures).

From all the figures it can be derived that a common denominator is the fact that the identification algorithm for the most part chooses one of the two correct faults. It is also observed that the fault diagnosis system always return back to nominal state after both the faults are reset.

Going a bit more into detail, it can be seen that the identification algorithm, as it is implemented now seems to have problems identifying a fault as GLR – especially as long as the effect from some other changed parameter is still present. This becomes especially

evident in Figure C.3, where the the PI parameter is reset to nominal state, while the GLR parameter remains changed/(in a fault state). In that case it took a long time (approx 250 s) until the algorithm identified the fault as changed GLR, instead of changed PI or another parameter.

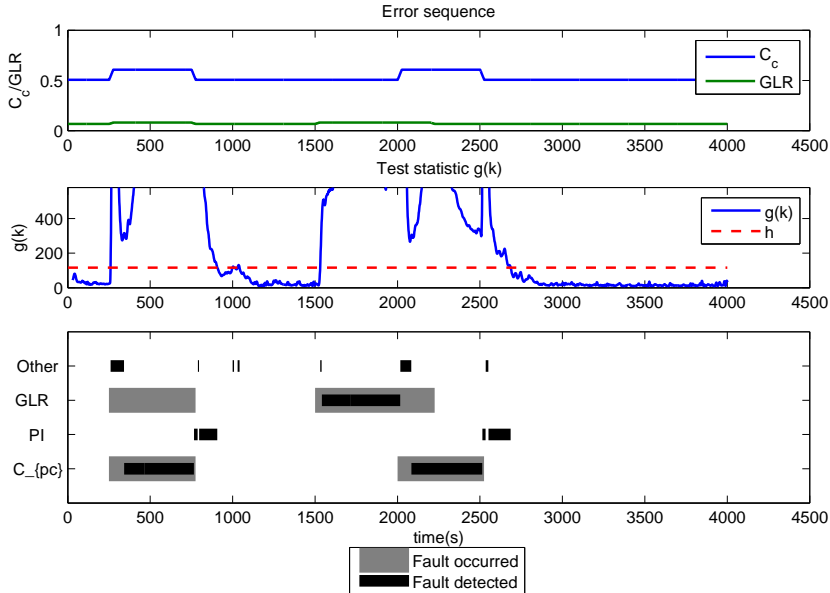


Figure 6.13: Test case 3a: Fault in both C_c and GLR parameter of well one. No change in u .

Test case 4: Realistic use case simulation

Figure 6.14 shows the simulation of test case 4. The figure illustrates how the increasing GLR parameter leads to increasing error in VFM of the bottom hole gas flow (q_g^{BH}), up until a point (shown with the dashed red line) where the GLR parameter is re-calibrated with the estimated GLR parameter. This procedure repeats itself 4 times as the GLR parameter simply continues to increase.

The results show a working fault detection algorithm that is able to provide the operator with the detection of the fault, identification of the fault and provide a good estimate of the magnitude. It also shows that after recalibration has been performed the test-statistic goes back to nominal state, and the alarm is turned off again. An additional plot is also added to this section, namely Figure 6.15, which presents the corresponding $g(k)$ for the simulation shown in Figure 6.14. This figure is added to illustrate the uncertainty related to this fault detection with the current implementation. This issue will be discussed in Section 7.2.3, with a focus on how Independent and Identically Distributed test variables, could contribute to improving the variance of $g(k)$.

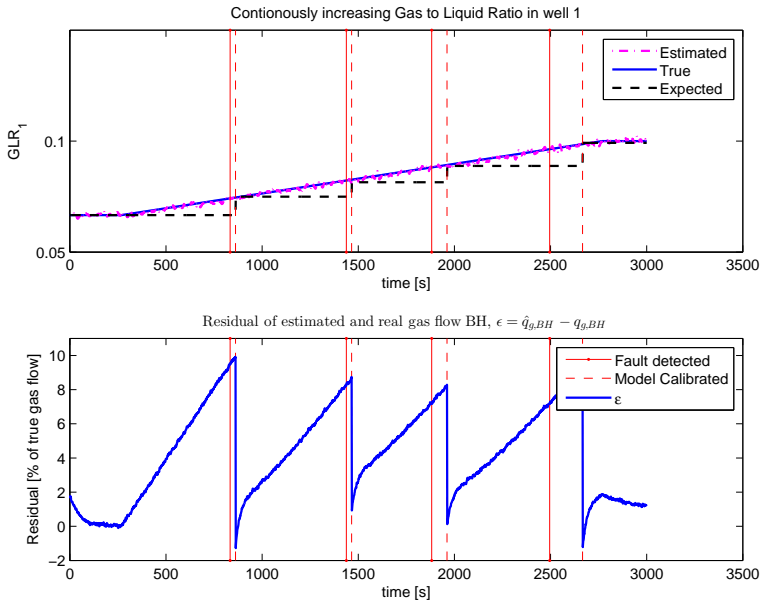


Figure 6.14: Test case 4: Illustration of a realistic use case. Note that in reality the time span of this would be considerably longer.

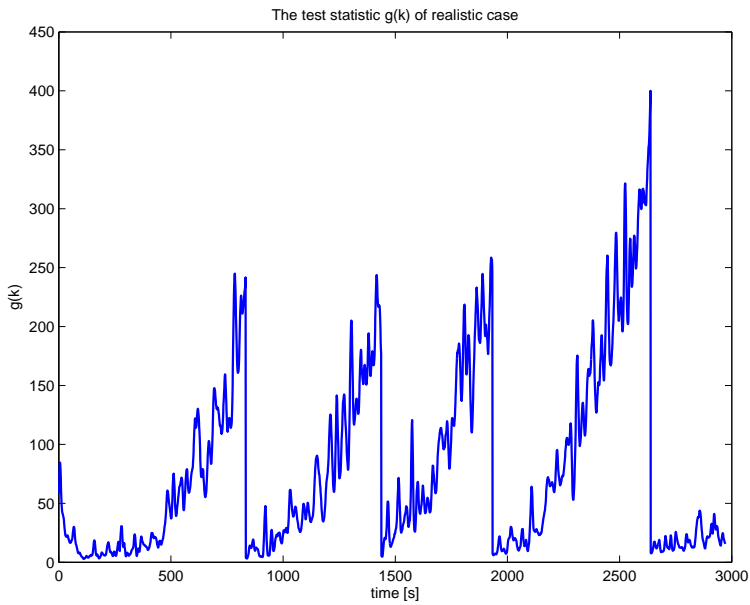


Figure 6.15: Test case 4: supplement plot, showing the variance and non-whiteness of $g(k)$

6.3 Runtime analysis

This section includes a brief presentation of the computation time, which is added for the purpose of illustrating the scalability of the estimators, and the general processing time needed for the different state estimators. Whether the estimators were used as state estimators in the VFM or for residual generation in the fault diagnosis system is irrelevant for this section.

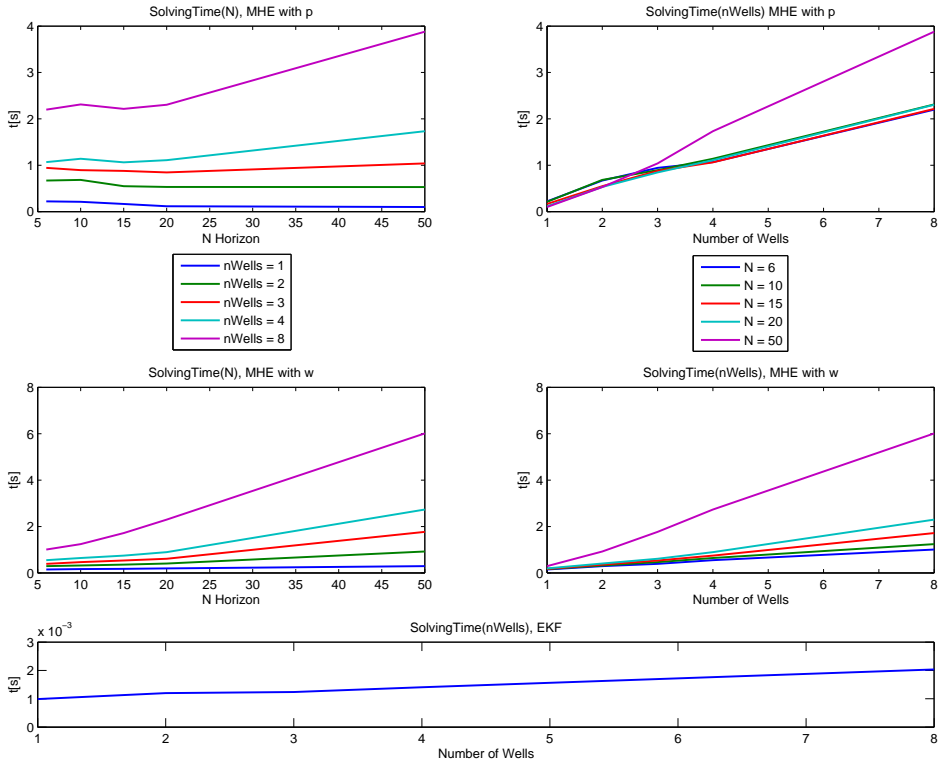


Figure 6.16: Runtime analysis of the estimators: Notice how the Kalman filter is significantly faster. The times referred to in the y-label is the average time of estimation for each time step.

The runtime analysis shown in Figure 6.16, shows the runtime as a function of "number of wells" and of "estimation horizon" (for the MHE estimators).

- The top row presents the runtimes for the MHE state and parameter estimator from Section 4.3.1
- The second row presents the runtimes for the MHE with model error parameters from Section 4.2.1

- The third row presents the runtimes of the extended Kalman filter as a function of the number of wells in the system

From observing the figure it becomes evident that:

- The runtime of the EKF is considerably lower than that of the two MHE approaches. Furthermore, it can be seen that the increase in runtime with increase in number of wells, seems to be a slowly increasing linear function.
- The runtime of the two MHE's, both seem to be in the same range of runtimes, but MHE with parameter estimation seems to have some problems when the horizon is too short. Apart from this phenomena, both of the MHE's seem to increase with increases in both horizon time and number of wells. The increase seems to be linear.

Discussion

The main goal of this thesis was to investigate the possibility of expanding the field of Virtual Flow Metering (VFM) with statistical change detection algorithms based on information from state of the art moving horizon estimators. The outcome of this investigation is discussed in this chapter.

7.1 Estimation

In this thesis, no performance tests were performed on the state estimators created (extended Kalman filter, MHE with- or without parameter estimation). No comparison of the performance between the respective state estimators are therefore discussed in this section. The experience obtained throughout the work with these estimators will, however, be discussed with respect to the estimators working as state estimators and as residual generator for the fault diagnosis system.

In Section 2, a table showing some of the pros and cons associated with EKF and MHE were presented from a theoretical viewpoint. These points form the framework for the discussions in this section.

Some of the points in Table 2.2, were related to the performance and will not be discussed in this thesis, but the following three points will be discussed further based on the experience gained during this work.

- **Computationally cheaper with EKF:** Looking to Section 6.3, this statement seems to be truthful, at least for the implementations in this thesis. It becomes very clear in Figure 6.16, that there is a significantly longer processing time when using the MHE's than it is with EKF. Furthermore, it seems that this difference is increasing with increasing dimension of the system (increase in number of wells). This can be an issue as real systems tend to be larger than the test case used in this thesis. That being said, there has not been put enough effort into making these approaches

efficient, that one can rule out MHE as a sufficiently efficient estimator. With some additional solving tools, the MHE's could potentially perform with a computational efficiency that competes with the efficiency of EKF (Kühl et al., 2011).

- **Simple implementation for unconstrained problems with EKF:** The system shown in this thesis can be estimated without the use of inequality constraints, which has been done in the implementation of the EKF. Based on the authors experience, this point has proven to be true. There is a significantly smaller effort needed for implementation of the EKF than what can be said for the MHE. Especially when the derivatives of the functions need to be provided (which is the case in this thesis, to achieve an acceptable computation time). This is mainly because the MHE (implemented in this thesis) needs to get the linear constraints and the Jacobian as matrices. These matrices become very large, and although they are handled by Matlab through the use of sparse matrices, it becomes fairly difficult for the user to keep track of where in the matrix all the functions need to be placed.
- **Test-variables from the MHE might be more intuitive for the implementation of the fault diagnosis system:** This property has not been investigated properly, as there has been no attempt to utilize the EKF as a residual generator (see Figure 2.5 for information on residual generators). It is the author's opinion, however, that there is no reason to think that EKF (or any other Kalman based estimator) cannot be utilized as a residual generator. Especially because any Kalman based estimator can in fact be used as a parameter estimator (Van Der Merwe and Wan, 2001) in a similar way to how the MHE was applied in this thesis.

To conclude this section of the analysis, the recommendation of the author is that future investigations in fault diagnosis used for model error detection, is focused towards Kalman filter based estimators. This is mainly because the MHE demands more effort into the implementation and that the increasing computing time might become an issue as the dimension of the system is increasing. In return there does not seem to be any particular advantages with MHE (for this purpose) that can outweigh these disadvantages.

7.2 Fault diagnosis

Fault diagnosis is usually used to detect physical faults in the system, which is done by detecting deviations from the "normal" behavior as a result of faults in the physical system. In this thesis however, the "fault" is in fact errors in the model and not in the physical system. This section discusses the results from testing the fault diagnosis systems created for this thesis – both with regards to testing the performance of the fault diagnosis systems, and with respect to its performance as a supplement to the VFM.

This section has become quite extensive as there are a lot of results to discuss. Accordingly the section is divided into several subsections.

- A short discussion on the fault diagnosis using w as test-variables

- Discussion on the fault diagnosis using \mathbf{p} as test-variables.
- An additional section that highlights the most important phenomena

Note: the system used for testing in this thesis is fairly simple, and hence the fault diagnosis would be significantly more difficult in a real life application.

7.2.1 Fault diagnosis using \mathbf{w} as test-variables, discussion of the test cases

The analysis in this section is somewhat limited because only one test was performed on the fault diagnosis system using model error parameters (\mathbf{w}) as test-variables. This comes as a result of the fact that, defining the necessary fault signature matrix Γ_D (described in Section 2.2.3) has proven to be problematic. Had this problem been solved, this approach would still be less intuitive than the fault diagnosis system using the system parameters (\mathbf{p}) as test-variables, hence the choice was taken to prioritize fault diagnosis based on \mathbf{p} .

This section will start by explaining and discussing this decision in more detail. For the fault identification algorithm to work, there is a need to identify a certain fault signature that relates the different faults to the test variables. This signature is presented in Γ_D as explained in Section 2.2.3. It has, however, not been determined any viable approaches to set a good Γ_D for the test-variables \mathbf{w} . As long as there is no good way to determine a sufficiently good signature matrix, this approach of fault diagnosis cannot be applied, as it would not be able to identify the poorly calibrated model.

The technique used to find Γ_D with \mathbf{w} as test-variables in this thesis, was to run tests on the simulator that introduced changes in the various parameters, and then documenting the effect on \mathbf{w} . This is a viable approach for a test on a simulator, but in a real system it is less viable.

If a Γ_D could be found without too much effort from the user, then this approach could probably perform just as well as any other approach. Some suggestions as to how this Γ_D could be found in a real system includes:

- Determining it analytically, by investigating the equations
- Determining it based on tests on a simulator for the respective system

Both of these methods have, however, serious drawbacks.

- Doing it analytically seems very difficult due to a considerable degree of smearing (as described in 2.2.3) in the error parameters (\mathbf{w})
- Doing it by simulation, depends on a very precise simulator to provide a sufficiently good Γ_D

It is also worth noting that the signatures of faults in GLR and PI, seem to be very similar – creating the need for a very precise Γ_D . It is also not unreasonable to believe that the

true Γ_D is dependent on the conditions of the system, implying that Γ_D can vary over time.

Despite these issues, one test were performed to investigate the possibilities of this approach, given that one had already managed to provide a good Γ_D . This test, analyzed whether or not the algorithm was able to detect and identify the fault when one of the system parameters was changed. Test case 1 showed good results, as the algorithm managed to detect and identify the faults. Especially the results for detecting and identifying the poorly calibrated C_c 's showed to be good. Based on the fact that there were many simplifications and assumptions present in this test, however, one should not conclude that using w as test-variables is the best approach. The results rather state that there might be possibilities and strengths associated with this approach that can be utilized if one was able to efficiently determine a good Γ_D .

7.2.2 Fault diagnosis using p as test-variables, discussion of the test cases

The tests performed in this section are all based on the parameters $p = [C_c \text{ PI } \text{GLR}]$ as test-variables in the fault diagnosis algorithm. This approach embodies the advantage that it is more intuitive in its implementation of the fault identification algorithm and the disadvantage that an already nonlinear optimization problem becomes more nonlinear. Which is the case because the two linear equations (3.6a) and (3.6b) in the original MHE formulation become nonlinear in the MHE with parameter estimation formulation. This challenge might also lead to a higher degree of non-convexity in the optimization problem, which the phenomena in test case 2 c and d might be an indicator of.

Test case 2: Basic tests

This test case illustrates many important results relevant for the discussion.

1. Working detection and identification of the faults with and without changes in u :

Although there are some minor mis-identifications, and a fairly long time for "undetection", (especially for the changes in PI and C_c), it is apparent that the algorithm for the most part manages to detect and correctly identify the source of the fault (mis calibration). For more complex problems, it is the author's opinion that the identification algorithm could be improved correspondingly to still achieve good results. This improvement can for instance be to include more information like measurement error, known correlations between the parameters, and possibly more functions and measurements (e.g. temperature relationships). These possible improvements have not been tested in this thesis, but constitute, in the author's opinion, possible upgrades with great potential.

2. Steady state deviation between real and estimated parameters after simultaneous changes in C_c and u :

In test case 2c, a situation occurred in which the parameters PI, and C_c never returned to their original state, leaving the fault (changed C_c) constantly detected even

after the fault had been removed (C_c were changed back). This phenomena is further discussed in 7.2.3. In addition to the fact that the optimization algorithm seems to find a local solution instead of the correct solution, it is interesting to consider when this happens. It only seem to happen when there is a change in the valve opening u simultaneously with change in the parameter C_c . This is interesting as changes in u did not seem to have any measurable effect on the parameter estimates, when there were no simultaneous change in C_c . Looking at this from a physical point of view, it can possibly be explained by the closeness of the two "variables" in equation (4.5). A change in C_c would give the exact same effect on q_c^{WH} as an identical change in u . When there is no simultaneous change in C_c , however, no such effect on the C_c estimate can be detected from changes in u .

3. Performance of detection and identification of faults from changes in GLR compared to changes in PI and C_c :

From the tests performed in this section it is clear by looking at Figure 6.11 that there is a considerable difference between how the changes in the different parameters affect each other's estimates. It becomes evident that changes in the real PI and C_c affect the estimates of the other parameters considerably more than changes in GLR¹. This result indicates that this fault diagnosis approach is especially good at detecting and identifying changes in GLR. This might come as a result of the fact that both C_c and PI directly affects the flow in the well, while GLR affects the composition of the fluid flowing, making PI and C_c more tightly connected.

Note that the tests performed in this case are all performed with changes in well 1. This is only done for convenience, the changes could have been in any of the other wells as well. Furthermore, unlike in test case 1, the fault diagnosis system has no apriori knowledge regarding the location of the changed parameters. In fact, each time the algorithm indicates that the fault occurred in the parameter "other" (in the figures) it implies that the fault has been (wrongly) identified to be in one of the other wells.

Test case 2d: changes in different valve openings

This test was performed to determine the effect of changes in valve openings in different wells than well 1. For the most part it showed (through tests that are not attached in this report) that the changes in u had negligibly effect. In the case of simultaneous changes in C_c of well 1 and changes in u of well 2, however, a mis-identification of the fault was observed. The fault was identified to arise from the same well that changed the valve opening. This is an interesting observation, as it highlights the difficulties with changes in u in parallel with changes in C_c . The connection between C_c and u seems to create significant issues for the fault diagnosis system.

Test case 3: Several faults simultaneously

This scenario tests the response to introducing several faults at the same time in the fault diagnosis system. As stated earlier it is already known that with the identification algorithm

¹ Note that this distribution of the error can be described as smearing (presented in Section 2.2.3)

implemented in this thesis, it is not possible to identify two different faults simultaneously. Hence, this test was not created to test if both the faults could be identified, but rather to see how the system reacted to a scenario with more than one fault.

From the results it is determined that the fault diagnosis system handled several faults fairly well. Most of the time it correctly identified one of the faults that was present, as opposed to an alternative result that could have been that the combination of the two faults led to a set of test-variables that was more consistent with a third type of fault (hence identifying a fault that was not present at all.) Furthermore, the faults were detected and "undetected" within a reasonable time, similar to the tests in Section 7.2.2.

Test case 4: Realistic use case simulation

This test case was created to illustrate the fault diagnosis algorithm in a more realistic scenario, both to illustrate the possibilities of the fault diagnosis system combined with VFM and to investigate a more realistic scenario. From this test-case, three main observations can be discussed

- **Provides significant help to an operator:**

As a test to illustrate an actual use case of the fault diagnosis system, it can safely be said to have given the imaginary operator valuable help to decide when and where re-calibration was needed, which in turn helped to keep the state estimation within a reasonable degree of uncertainty. It should, however, be noted that this was a fairly simple example, performed on a scenario with changes in GLR (known from previous tests to be the parameter, of the 3 parameters, that is best detected and identified).

- **Simple case, but still considerable uncertainty in the test statistic $g(k)$:**

Looking at Figure 6.15, it is evident that there are large oscillations in the test statistic that is used to detect the fault. These spikes would all signal for a fault, but because the operator delays any decision until there are 30 consecutive steps with the same identified fault, these spikes do not effect the actual use case shown in Figure 6.14. This is nevertheless not a desirable behavior and should be addressed. Why this happens and possible solutions are discussed further in Section 7.2.3.

- **Difficulties in correctly identifying the fault:**

In this test there was an additional barrier present - that the operator needed 30 consecutive time steps with the same fault identification to do recalibration. This barrier made sure that the correct parameters were at all instances re-calibrated. However it should be noted that the system detected the fault long before there seemed to be enough of a parameter residual ($\mathbf{p}^* - \hat{\mathbf{p}}$) to correctly identify the fault. This led to many different temporary "identifications" before, it was correctly identified after 30 consecutive time-steps with identical identifications. For the purpose of an actual implementation, this would be too much of an uncertainty and possible improvements would have to be evaluated and performed. Such improvements could potentially be expansion of the identification algorithm as suggested in Section 8.1.

7.2.3 Fault diagnosis, discussion of the most important discoveries

This section presents three phenomena revealed in the test cases of Chapter 6, that are considered to be of special interest. These phenomena are therefore highlighted, and described in more detail in this section.

Lack of IID

As stated in Chapter 2, a prerequisite for optimal performance of the fault detection technique used in this thesis is that the test-variables are IID (Independent and identically distributed). This is not the case in this thesis, which becomes especially apparent in test case 4. In that test case there were large oscillations which lead to a big variance in the test statistic, as observed in Figure 6.15. Optimally with no correlation between the test-variables at the current time step and the previous, such large and deterministic oscillations would disappear, leading to a more stable fault diagnosis algorithm. For future implementations, this is considered by the author to be one of the more crucial elements for improvement. The reader is referred to Hansen and Blanke (2012) for more information on white filtering for the purpose of making the test-variables IID.

Possibility for local solutions, due to non convexity

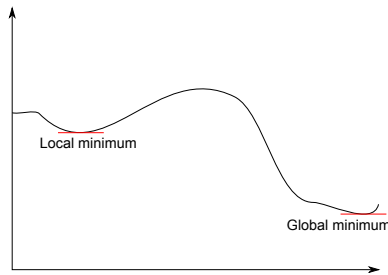


Figure 7.1: Illustration of the possibility for local minimum, which can be drastically different from the global minimum.

In test case 2c and 2d, it was observed that a problem arose, when the change in C_c occurred simultaneously with changes in u . This problem was caused by the fact that the estimated parameter used in the fault diagnosis system did not return to their correct value after the change in C_c had been reset. It seemed as though the MHE found a different optimal solution for the parameters. This is a problem that can potentially be a massive uncertainty with this kind of fault diagnosis. If it cannot be guaranteed that such a false solution does not occur, it makes it harder to trust the output of the algorithm. There can be two possible reasons for this alternative solution provided by the MHE.

- One is that there might be more than one equally good solution as the problem is over determined

- The other and more likely reason is that the optimization has found a local solution instead of the global solution (see Figure 7.1), which can happen for these kind of non convex problems.

Globally solving non-convex optimization problems is not a simple task. In fact, as stated by Murty and Kabadi (1987), finding a global minimum of a non-convex NLP is a NP-complete problem, implying that there are no techniques that solve such a problem in polynomial time. There are techniques, however, that can help improve the the probability of finding the global solution, e.g. use of heuristic methods for choosing various initial guesses provided to the optimization algorithm. No such techniques have been deployed in this thesis, but it is the author's opinion that this is one of the more crucial points for future implementations, as such local solutions procure false alarms, which degrade the performance of the fault diagnosis system.

Weaknesses and strength in the two approaches

Through the tests performed in this thesis there seem to be a clear indication that the two approaches for fault detection have different strengths and weaknesses. Fault diagnosis based on the model error parameters \mathbf{w} , seemed to be good at detecting and identifying poorly calibrated C_c . While the fault diagnosis system based on the system parameters \mathbf{p} , seemed to be especially good at detecting and identifying poorly calibrated GLR, it did also seem to be fairly good at detecting and identifying poorly calibrated PI. The fact that they embody different strengths can potentially be a good thing. If both of these systems were to be combined (as stated in Section 2), it could make for a much better fault diagnosis system. This combination requires, however, an efficient way of determining Γ_D for the fault diagnosis system using \mathbf{w} as test-variable.

That being said, with the current issues with the implementation of an identification algorithm. The fault diagnosis system based on \mathbf{p} is the recommended approach, of the two MHE's, for future investigations.

Smearing of the test-variables

As stated earlier, there is a significant degree of smearing in both the approaches of residual generation. This is a factor that complicates the process of identifying the poorly calibrated models. This smearing is essentially what makes the process of determining Γ_D for fault diagnosis using \mathbf{w} so complicated, and a big part of the uncertainty in the fault identification process using \mathbf{p} as test-variables. Learning how the model error is distributed in the "slack variables" (\mathbf{p} and \mathbf{w}), would significantly help the process of identifying the faults. Furthermore it could help to pinpoint which "faults" that can be identified with certainty, and which "faults" that have greater uncertainty when identified.

Conclusion

From the tests conducted in this thesis, it can be concluded that both of the fault diagnosis systems have potential. Test cases demonstrated that it is possible to detect and identify poorly calibrated models for simple test cases. Both of the approaches have, however, some drawbacks and unresolved issues. Because of these limitations, the implementations could not be used in a real system as of now, nor is it known for a fact that the issues can be resolved in the future. This leads to suggestions for future improvements and implementations, which in many ways, can be seen as the main results of this thesis. These improvements have been presented in section 8.1 and 7.2.3, and it is not considered unlikely that the issues can be resolved by the help of some of these suggestions.

Operators or VFM vendors that evaluate this supplement to VFM products will find that more research is needed to improve its robustness/reliability. If the fault diagnosis system was to be implemented, it would be for the purpose of increasing the reliability of the models. If the fault diagnosis system is unreliable itself, however, this would not lead to any improvement of the VFM. With the current implementation many false alarms would be raised and many mis-identification occur – all of which contributing to make the fault diagnosis system unreliable, and unfit for an actual implementation.

To conclude, no fault diagnosis system, satisfactory for implementation in a production environment has been created, nor has it been proven that such a system can be made. However positive results have been found, indicating the potential of the fault diagnosis systems, and suggestions have been posted for further investigation of this concept.

To continue this work, and preserve the knowledge obtained during this thesis work. Bjarne Grimstad and the author have started to draft a paper with preliminary results to be published in an international conference.

8.1 Future work

This section lists the possibilities for future work, that are considered by the author to be most promising:

- Investigate the possibilities of using a Kalman filter-based residual generator as opposed to the MHE.
- If investigations are continued with MHE as residual generator: focus on MHE with parameter estimation, or focus on how to determine sufficiently good Γ_D for the approach with \mathbf{w} as test-variables. To continue with testing on the approach with \mathbf{w} as test-variable without the knowledge that, Γ_D can be found efficiently, gives little contribution to the development of this VFM expansion.
- Improve the identification algorithm with improved signature matrix Γ_D and possibly expanded test variables to include measurement error v . Potentially also expanded by introducing new functions related to temperature etc.
- Improve the test-variables (residuals) by pre-processing them with a whitening filter as explained in section 2.2.2, and discussed in 7.2.3.
- Get some external simulator, or measurements from a real scenario, with parameters that are known to have been re-calibrated. This is crucial to provide more reliability to the test results.
- Develop more robust solvers with respect to avoiding local solutions, as discussed in 7.2.3. Note that the techniques do not necessarily need to be applied constantly, but could possibly be an extension to check the solution in the case of fault detection. This way one would save computation time and possibly still avoid false alarms. If the solution was found to be local and not global, no alarm would be raised, and the fault diagnosis system do not loose credibility.

References

- Aamo, O., Eikrem, G., Siahaan, H., and Foss, B. (2005). Observer design for multiphase flow in vertical pipes with gas-lift - theory and experiments. *Journal of Process Control*, 15(3):247–257.
- Alessandri, A., Baglietto, M., and Battistelli, G. (2008). Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44:1753–1765.
- Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs.
- Bestle, D. and Zeitz, M. (1983). Canonical form observer design for non-linear time-variable systems. *International Journal of control*, 38(2):419–431.
- Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., and Schröder, J. (2006). *Diagnosis and fault-tolerant control*, volume 2. Springer.
- Bringedal, B. and Phillips, A. (2006). Application of Virtual Flow Metering as a Backup or Alternative to Multiphase Flow Measuring Devices. In *Subsea Controls and Data Acquisition 2006: Controlling the Future Subsea*, pages 17–24. Society of Underwater Technology.
- Bringedal, B. O., Storkaas, E., Aarset, M. F., and Dalsmo, M. (2010). Recent Developments in Control and Monitoring of Remote Subsea Fields. In *SPE Intelligent Energy Conference and Exhibition*, number March, pages 23–25. Society of Petroleum Engineers.
- Brown, R. G. and Hwang, P. Y. C. (2012). *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises, 4th Edition*. CourseSmart Series. Wiley.
- Chow, E. and Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection systems. *Automatic Control, IEEE Transactions on*, 29(7):603–614.
- Cramer, R., Goh, K. C., Dolan, M., and Moncur, C. (2009). Data driven surveillance and optimization. *SPE paper*, 122554.

-
- Currie, J. and Wilson, D. I. (2012). OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In Sahinidis, N. and Pinto, J., editors, *Foundations of Computer-Aided Process Operations*, Savannah, Georgia, USA.
- Delmaire, G., Cassar, J.-P., and Staroswiecki, M. (1994). Comparison of identification and parity space approaches for failure detection in single input single output systems. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pages 865–870. IEEE.
- Ding, X. and Frank, P. M. (1991). Frequency Domain Approach and Threshold Selector for Robust Model-Based Fault Detection and Isolation. In *Preprint of IFAC/IMACS Symp. SAFEPROCESS'91*, volume 1, pages 307–312, Baden-Baden.
- Ding, X., Guo, L., and Frank, P. M. (1993). A frequency domain approach to fault detection of uncertain dynamic systems. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 1722–1727. IEEE.
- Egeland, O. and Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics.
- Falcone, G. and Hewitt, G. (2002). Multiphase flow metering: current trends and future developments. *Journal of Petroleum Technology*, 54(April):77–84.
- Foss, B. (2012). Process control in conventional oil and gas fields-Challenges and opportunities. *Control Engineering Practice*, 20(10):1058–1064.
- Frank, P. M. (1996). Analytical and qualitative model-based fault diagnosis—a survey and some new results. *European Journal of control*, 2(1):6–28.
- García, E. A. and Frank, P. M. (1997). Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5(5):663–670.
- Gauthier, J. P., Hammouri, H., and Othman, S. (1992). A simple observer for nonlinear systems applications to bioreactors. *Automatic Control, IEEE Transactions on*, 37(6):875–880.
- Grimstad, B., Robertson, P., and Foss, B. (2015). Virtual flow metering using b-spline surrogate models. In *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, Florianópolis, Brazil.
- Gunnerud, V. and Foss, B. (2010). Oil production optimization-A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, 34(11):1803–1812.
- Hansen, S. and Blanke, M. (2013). Control surface fault diagnosis with specified detection probability - Real event experiences. *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, pages 526–531.
- Hansen, S. r. and Blanke, M. (2012). In-flight fault diagnosis for autonomous aircraft via low-rate telemetry channel. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 8(PART 1):576–581.

-
- Haseltine, E. L. and Rawlings, J. B. (2003). A Critical Evaluation of Extended Kalman Filtering and Moving Horizon Estimation. *Technical Report TWMCC*, pages 2451–2460.
- Heddle, R., Foot, J., Rees, H., et al. (2012). Isis rate & phase: Delivering virtual flow metering for 300 wells in 20 fields. In *SPE Intelligent Energy International*. Society of Petroleum Engineers.
- High, G., Frantzen, K.-H. H., Marshall, M., High, G., Frantzen, K.-H. H., Marshall, M., and Hess, A. (1995). On-Line Subsea Multiphase Flow Measurement. (May):1–4.
- Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing-tutorial paper. *Automatica*, 29(4):815–835.
- Isermann, R. and Ballé, P. (1997). Trends in the application of model based fault detection and diagnosis of technical processes. *Control Eng. Practice*, 5(5):709–719.
- Jahn, F., Cook, M., and Graham, M. (2008). *Hydrocarbon Exploration and Production*. Developments in petroleum science. Elsevier.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.
- Kandepu, R., Foss, B., and Imsland, L. (2008a). Applying the unscented Kalman filter for nonlinear state estimation. *Journal of Process Control*, 18(7-8):753–768.
- Kandepu, R., Imsland, L., and Foss, B. a. (2008b). Constrained state estimation using the Unscented Kalman Filter. *2008 16th Mediterranean Conference on Control and Automation*, pages 1453–1458.
- Kang, W., Krener, A. J., Xiao, M., and Xu, L. (2013). A Survey of Observers for Non-linear Dynamical Systems. In Park, S. K. and Xu, L., editors, *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. II)*, volume II, pages 1–25. Springer, Berlin, Heidelberg.
- Kühl, P., Diehl, M., Kraus, T., Schlöder, J. P., and Bock, H. G. (2011). A real-time algorithm for moving horizon state and parameter estimation. *Computers and Chemical Engineering*, 35(1):71–83.
- Lipták, B. G. (1995). *Process measurement and analysis*. Butterworth Heinemann.
- Love, J. (2007). *Process Automation Handbook: A Guide to Theory and Practice*. Springer.
- Melbø, H., Morud, S., Geest, R., Bringeda, B. r., and Stenersen, K. (2003). Software That Enables Flow Metering of Well Rates With Long Tiebacks and With Limited or Inaccurate Instrumentation. *Proceedings of Offshore Technology Conference*, pages 1–7.
- Murty, K. G. and Kabadi, S. N. (1987). Some NP-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129.

-
- Narasimhan, S. and Jordache, C. (1999). *Data reconciliation and gross error detection: An intelligent use of process data*. Gulf Professional Publishing.
- Nithya V S (2009). EKF. <http://www.mathworks.com/matlabcentral/fileexchange/24855-extended-kalman-filter--ekf->. Accessed: 2015-01-28.
- Patton, R. J., Frank, P. M., and Clarke, R. N. (1989). *Fault diagnosis in dynamic systems: theory and application*. Prentice-Hall, Inc.
- Petroleum Services Association of Canada (2015). Petroleum, industry-overview. <http://www.pzac.ca/business/industry-overview>. Accessed: 2015-05-26.
- Qu, C. C. and Hahn, J. (2009). Computation of arrival cost for moving horizon estimation via unscented Kalman filtering. *Journal of Process Control*, 19(2):358–363.
- Rao, C. V., Rawlings, J. B., and Lee, J. H. (2001). Constrained linear state estimation – a moving horizon approach. *Automatica*, 37(10):1619–1628.
- Rao, C. V., Rawlings, J. B., and Mayne, D. Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *Automatic Control, IEEE Transactions on*, 48(2):246–258.
- RDP group (2005). White Paper 1: Understanding Transducer Characteristics and Performance Understanding Linearity. Technical report, RDP group, Wolverhampton.
- Sauter, D., Mary, N., Sirou, F., and Thieltgen, A. (1994). Fault diagnosis in systems using fuzzy logic. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pages 883–888. IEEE.
- Sorsa, T. and Koivo, H. N. (1993). Application of artificial neural networks in process fault diagnosis. *Automatica*, 29(4):843–849.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
- Van Der Merwe, R. and Wan, E. A. (2001). The square-root unscented Kalman filter for state and parameter-estimation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 6, pages 3461–3464. IEEE.
- Wachter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.
- Walpole, R. E. (2012). *Probability & statistics for engineers & scientists*. Number 9th ed. Pearson, Boston, Mass.
- Webster, J. G. and Eren, H. (2014). *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement*, volume 1. CRC press.

-
- Willersrud, A., Blanke, M., Imsland, L., and Pavlov, A. (2014a). Drillstring Washout Diagnosis using Friction Estimation and Statistical Change Detection. *IEEE Trans. Control Syst. Technol. (submitted)*, pages 1–14.
- Willersrud, A., Blanke, M., Imsland, L., and Pavlov, A. (2014b). Fault diagnosis of down-hole drilling incidents using adaptive observers and statistical change detection. *Journal of Process Control (submitted)*.
- Zeitz, M. (1987). The extended Luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2):149–156.

General background theory

This chapter contains some general background theory to support the theory presented in the principal part of this paper.

A.1 Reservoir fluid composition

The fluid that is streaming up from the reservoir will not be pure, gas, oil or water. It will be a mixture of the three. When considering the mixture of fluids that originates from the reservoir, two parameters are usually considered to fundamentally describe the composition. Water Cut (**WC**) and Gas to Oil Ratio (**GOR**).

- **WC**: Water cut describes the fraction of water in the liquid comprised of water and oil (Jahn et al., 2008). Whether the fraction is in mass fraction, or volumetric fraction at standard conditions is relevant and needs to be decided to keep consistency. In this thesis, **WC** is chosen to be the mass fraction. In other words

$$\text{WC} = \frac{q_w \text{ [kg/s]}}{(q_w + q_o) \text{ [kg/s]}} \times 100\% \quad (\text{A.1})$$

- **GOR**: Gas Oil Ratio refers to the relationship between the gas produced and oil produced (Jahn et al., 2008). This parameter is like **WC** chosen to present the mass fraction, and not the volumetric fraction at standard conditions.

$$\text{GOR} = \frac{q_g \text{ [kg/s]}}{q_o \text{ [kg/s]}} \quad (\text{A.2})$$

A.1.1 Properties of hydrocarbon fluids

Density of the fluid is a central element in the modelling of the system. For this project there are 3 important densities that stands out.

-
- $\rho_w \approx 1000 \text{ kg/m}^3$, assumed incompressible, which means that it is not effected by temperature or pressure (not strictly true, but a decent approximation)
 - $\rho_o \approx 900 \text{ kg/m}^3$, assumed incompressible as well, the density does however vary much dependent of the kind of oil. The assumption of incompressible fluid is much weaker with the oil than with the water.
 - $\rho_g = \frac{MP}{zRT}$, where z is the compressibility factor (also known as gas-deviation factor). Is the most uncertain aspect, but it will be in the range of $[0.9 \ 1]$ for the conditions that are relevant in our case. In the case of standard conditions, and approximately methane as the gas we have $\rho_g \approx 0.7 \text{ kg/m}^3$

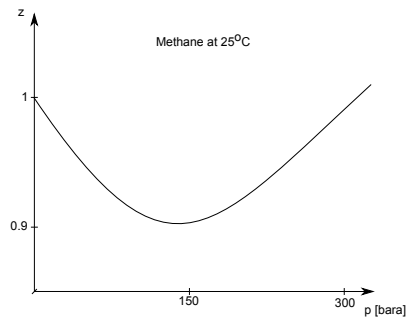


Figure A.1: Approximate illustration of the compressibility factor of methane at 25 °C

Appendix **B**

Implementation of estimators, supplement

B.1 Moving horizon estimator implementation

The implementation is explained in Section 4.3. This section presents the Jacobian function provided with this implementation, which were excluded from the Section 4.3 to save

limit the size of that section.

$$\mathbf{z} = [\mathbf{x}; \mathbf{x}_0; \mathbf{p}; \mathbf{v}]$$

$$J(\mathbf{z}) = \begin{bmatrix} J_{1,x} & J_{1,x_0} & J_{1,(p,v)} \\ J_{2,x} & J_{2,x_0} & J_{2,(p,v)} \\ J_{3,x} & J_{3,x_0} & J_{3,(p,v)} \\ J_{4,x} & J_{4,x_0} & J_{4,(p,v)} \\ J_{5,x} & J_{5,x_0} & J_{5,(p,v)} \\ J_{6,x} & J_{6,x_0} & J_{6,(p,v)} \end{bmatrix}$$

$$J_{1,x}(\mathbf{z}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \frac{-\frac{1}{2}C_c\rho_m u}{\sqrt{\rho(p^{WH}-p^{sep})}} & 0 & 0 & 0 & \frac{-\frac{1}{2}C_c(p^{WH}-p^{sep})u}{\sqrt{\rho(p^{WH}-p^{sep})}} & 0 & 0 \end{bmatrix}$$

$$J_{2,x}(\mathbf{z}) = \begin{bmatrix} 0 & 0 & \frac{-x_1}{x_1+x_2} & 1 & 0 & 0 & 0 & \frac{-q_c^{WH}}{x_1+x_2} + \frac{x_1 q_c^{WH}}{(x_1+x_2)^2} & \frac{x_1 q_c^{WH}}{(x_1+x_2)^2} & 0 & 0 & 0 \end{bmatrix}$$

$$J_{3,x}(\mathbf{z}) = \begin{bmatrix} 0 & 0 & \frac{-x_2}{x_1+x_2} & 0 & 1 & 0 & 0 & \frac{x_2 q_c^{WH}}{(x_1+x_2)^2} & \frac{-q_c^{WH}}{x_1+x_2} + \frac{x_2 q_c^{WH}}{(x_1+x_2)^2} & 0 & 0 & 0 \end{bmatrix}$$

$$J_{4,x}(\mathbf{z}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{-RT_t}{M(L_t A_t - \nu_l x_2)} 10^{-5} & \frac{-RT_t \nu_l x_1}{M(L_t A_t - \nu_l x_2)^2} 10^{-5} & 0 & 0 & 0 \end{bmatrix}$$

$$J_{5,x}(\mathbf{z}) = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 10\text{PI} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$J_{6,x}(\mathbf{z}) = [1 \quad -10\text{GLR} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$J_{1,(p,v)}(\mathbf{z}) = \begin{bmatrix} \mathbf{C}_p \mathbf{c} & \text{PI} & \text{GLR} & \mathbf{v} \\ -\frac{1}{10} \sqrt{\rho_m} 10^2 (10^2 p^{WH} - p_{sep}) u & 0 & 0 & 0 \end{bmatrix}$$

$$J_{2,(p,v)}(\mathbf{z}) = [0 \quad 0 \quad 0 \quad 0]$$

$$J_{3,(p,v)}(\mathbf{z}) = [0 \quad 0 \quad 0 \quad 0]$$

$$J_{4,(p,v)}(\mathbf{z}) = [0 \quad 0 \quad 0 \quad 0]$$

$$J_{5,(p,v)}(\mathbf{z}) = [0 \quad -\frac{1}{10}(p^{res} - 10^2 p^{BH}) \quad 0 \quad 0]$$

$$J_{6,(p,v)}(\mathbf{z}) = [0 \quad 0 \quad -10q_l^{BH} \quad 0]$$

Appendix C

Additional Plots

This section presents plots, that were not considered to be as relevant as the ones that were included in the principal part of this thesis. The plots are presented within section that indicate which section they are meant to supplement.

C.1 Validation of simulator model

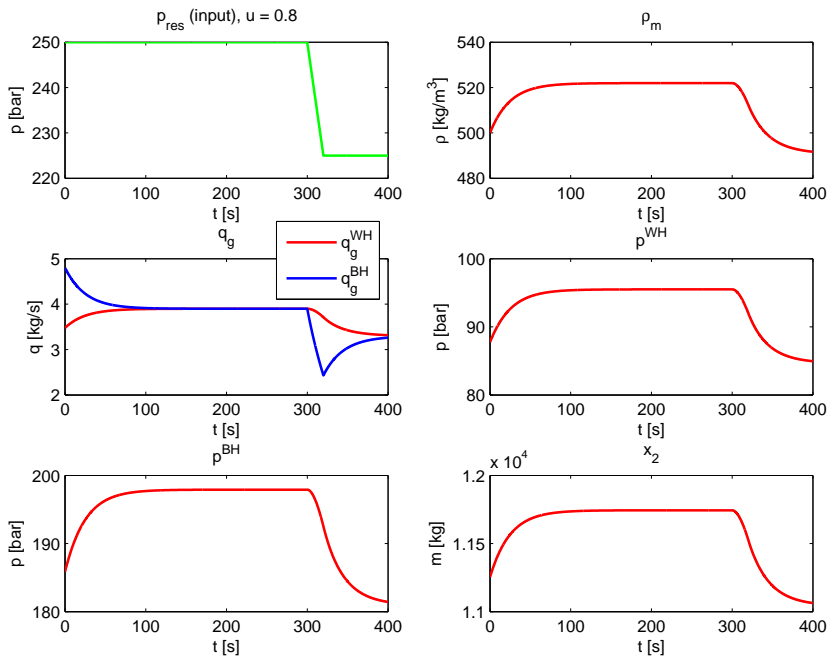


Figure C.1: Dynamical validation from section 3.2.1. This time with change in reservoir pressure.

C.2 GLRT with parameters as test variable, results

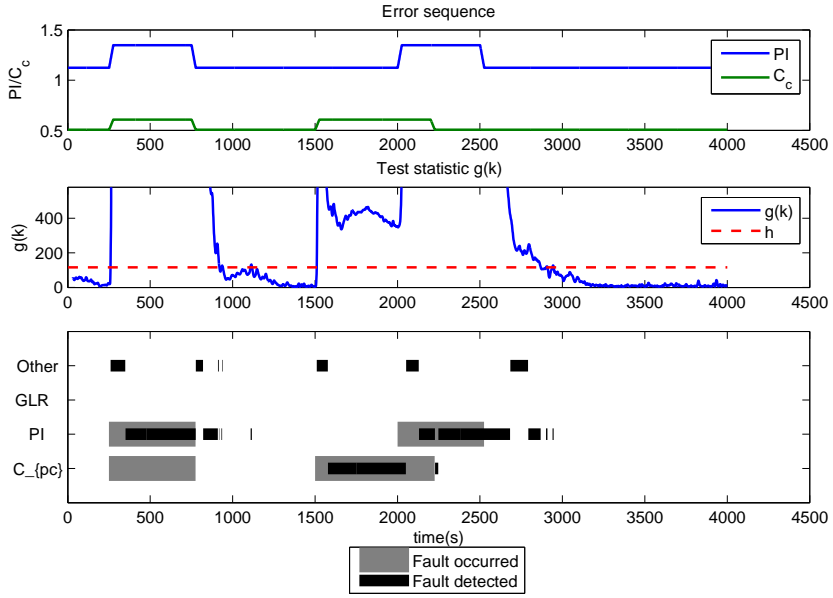


Figure C.2: Test case 3b: Fault in both C_c and GLR parameter of well one. No change in u .

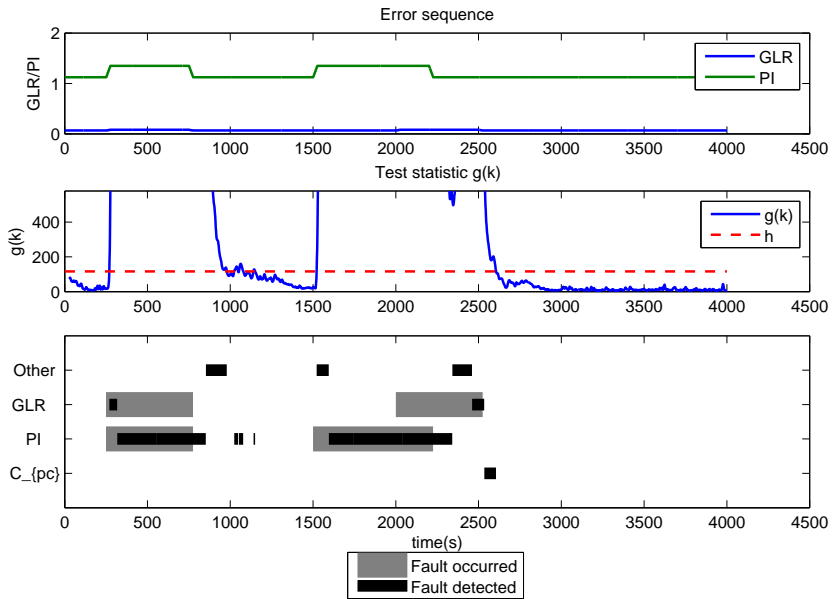


Figure C.3: Test case 3c: Fault in both C_c and GLR parameter of well one. No change in u .

Matlab scripts and functions

This chapter is added to give a brief overview on how the code used in this thesis was built up. It will not contain any scripts or function (this can be found in a separate zip-file handed in along with this thesis.). See figure D.1 for a brief summary of the complete code. In the item list below a short description of each box is given. For further information, look at the code handed, and the documentation for each script/function.

- **Main script/ test cases:** For every test that is run, some input was needed for the estimators, fault diagnosis algorithm and for the simulator. This information is created in the main script (in reality there are a set of different scripts, but they serve the same purpose.). The output from the simulators, estimators and fault diagnosis, is also analyzed in this box.
- **Simulator ode45:** A simple simulator that is not used much, but created mostly to verify the simulator based on optimization (the one explained in section 3).
- **Simulator based on optimization:** A simulator of the system that is implemented (as the one explained in section 3). The actual optimization is done through "opti-toolbox".
- **Opti-toolbox:** An external toolbox (Currie and Wilson, 2012), that works as a framework for utilizing open source optimization solvers. In this thesis IPOPT (Wachter and Biegler, 2006) has been chosen as the solver.
- **MHE:** The two different implementations of MHE described in section 4.
- **Fault diagnosis:** The algorithm that includes "Generalized Likelihood Ratio Test" and the fault identification algorithm explained in section 2.2.3.
- **Kalman Filter:** An interface to use the EKF program that is provided externally. This Box also include all the functions related to this system
- **EKF:** Externally created EKF framework (Nithya V S, 2009).

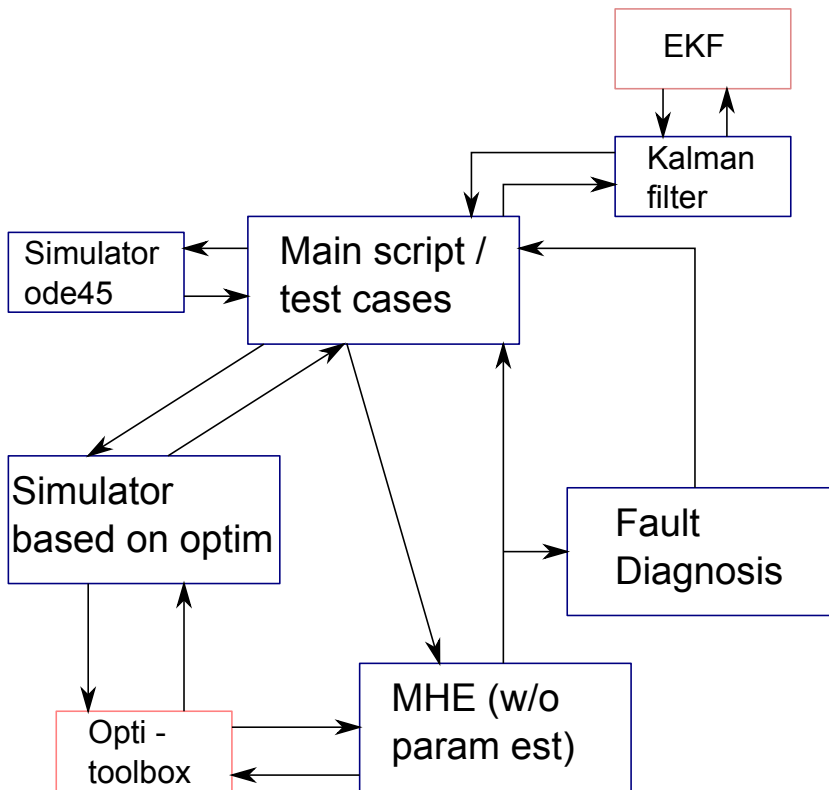


Figure D.1: A general description of the Matlab code used in this thesis. Note that the boxes in red ("EKF" and "opti-toolbox") are both external programs.