NTNU – Trondheim
Norwegian University of
Science and Technology

Production Optimization in a Cluster of Gas-Lift Wells

Benjamin Julian Tømte Binder

Master of Science in Engineering Cybernetics
Submission date: June 2012
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Vidar Gunnerud, IO-senteret

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem description

In an oil and gas production system, where wells are connected to manifolds with multiple pipelines, a complex optimization problem arises. The objective is often to maximize the oil production while respecting constraints on the system, such as limited capacity of processing facilities. In this work production optimization of one manifold, which is connected to 7 wells and which resembles a manifold and wells on the Marlim field in the Campos Basin outside Brazil, is studied. The field is operated by Petrobras.

The project will compare the use of static models for production optimization with the alternative use of dynamic models to account for transients. The goal is thus to assess the gain of a dynamic approach vs. optimization with static models. To make the study as realistic as possible, data from operations of one manifold will be used to evaluate the models, simulator and scenarios as close as possible.

The research is a continuation of previous work: [Binder 2011a, Binder 2011b]

Task description

1. Literature review on methods for optimization and control of dynamic systems. This is an extensive area so the review should be limited to methods of particular interest to the current study.
2. Develop and implement suitable static and dynamic models for one specific template with wells. Assume that the downstream boundary is defined by a constant pressure. Fit the model to one template at the Marlim field using available data and other information.
3. Define a set of typical operating scenarios; some with predominantly stable operation and others with extensive variability.
4. Develop, implement and test methods for static optimization and dynamic optimal control.
5. Assess and compare the performance of static optimization with dynamic optimal control for the operating scenarios defined in item 3.
6. The optimization methods require hardware capabilities in terms of instrumentation and actuators. Compare this with available technologies at the Marlim field and discuss the need for an upgrade to facilitate production optimization as discussed above.

Assignment given: January 23rd, 2012

Supervisor: Prof. Bjarne Anton Foss, ITK

Co-supervisors: Vidar Gunnerud, NTNU, and Alex Teixeira, Petrobras SA

Abstract

Subsea petroleum extraction systems may be large and complex, and many decisions affect the production. Maintaining high production levels is not a trivial task. As decisions are made based on available information and experience, better decisions come with better information. Decision support tools may provide essential information to achieve better production levels.

In this master thesis, different methods are proposed as decision support tools. The aim is to increase the production from a part of a subsea production system, consisting of a manifold with seven producing wells and two flowlines, given certain system constraints. The methods are based on well models and numerical optimization, and both static and dynamic optimization is considered. The well models are non-linear, and binary decisions are also present. The problems that arise are complex MINLP problems, and are solved by combining 'brute force', 'Branch & Bound', and a nonlinear solver. The solution of the problems is implemented in MATLAB, and tested on predefined test scenarios, with no, little or extensive dynamics present. The performance is assessed by simulations, and by calculating the resulting average production.

It was found that static optimization to decide the well settings, such as valve openings and flowline routing, has a great potential to increase the oil production from the system. The results when applying a dynamic approach to the system were not conclusive, but the methods proposed showed no indications of any major performance increase, relative to applying only static optimization.

Sammendrag

Undervanns petroleumproduksjonssystemer er ofte store og komplekse, og krever mange avgjørelser som har betydning for produksjonen. Å opprettholde høye produksjonsnivåer er ikke en triviell oppgave. Siden avgjørelser tas basert på tilgjengelig informasjon, samt erfaring, vil bedre avgjørelser kunne tas med bedre informasjon tilgjengelig. Beslutningsstøtteverktøy kan bidra med avgjørende informasjon for å oppnå bedre produksjonsnivåer.

I denne masteroppgaven er forskjellige metoder foreslått som beslutningsstøtteverktøy. Målet er å øke produksjonen fra en del av et undervannsproduksjonssystem, som består av en manifold med sju produksjonsbrønner, og to produksjonsrørledninger. Metodene er basert på brønnmodeller og numerisk optimalisering, og både stasjonær og dynamisk optimering blir vurdert. Brønnmodellene er ulineære, og binære beslutninger er også til stede. Problemene som oppstår er komplekse MINLP-problemer, og blir løst ved å kombinere 'brute force', 'Branch & Bound' og en ulineære problemløser. Løsningen av problemene er implementert i MATLAB, og testet på forhåndsdefinerte testscenarioer, med ingen, lite eller mye dynamikk. Ytelsen er vurdert ved hjelp av simuleringer, og ved å regne ut gjennomsnittlig produksjon.

Resultatene viser at stasjonær optimering for å avgjøre innstillingene i systemet, slik som ventilåpninger til brønnene, og ruting av produksjonslinjene, har et stort potensial for å øke oljeproduksjonen fra systemet. Resultatene av en dynamisk tilnærming var ikke konkluderende, men metodene som er foreslått viste ingen indikasjon til å gi noen større forbedring av ytelsen til systemet, sammenlignet med å kun implementere stasjonær optimering.

Preface

This master thesis was written during the final semester of the five-year study program 'Master of Science in Engineering Cybernetics' at the Norwegian University of Science and Technology.

This work has been done in collaboration with the IO center at NTNU, and Petrobras' research center CENPES in Rio de Janeiro, Brazil. I began this work when I was employed by the IO center, as an intern at CENPES, from June-August 2011, and continued during my specialization project at NTNU during the fall of 2011. This master thesis is a further continuation of this work.

I would like to use this opportunity to thank my supervisor, professor Bjarne A. Foss, both for his insight, encouragement, and support. I would also like to thank my co-supervisor, PhD Vidar Gunnerud, both for his friendship and support, and his useful advice during the entire project. I would also like to thank Alex Teixeira for the stay at CENPES, and the information and ideas he has provided. I am also grateful for the opportunity I got from the IO center, to go to Rio de Janeiro, and for the interesting project work that followed. Finally, I would like to thank my fiancée, Siri Tesaker, for her continued encouragement and support.

Trondheim, June 22, 2012

Benjamin J. T. Binder

Contents

1 Introduction **1**

1.1 Background 1

1.1.1 Petroleum production 1

1.1.2 The offshore industry 2

1.1.3 Subsea production systems 2

1.1.4 Artificial lift 3

1.1.5 Production optimization 5

1.1.6 Petroleum in Brazil 6

1.1.7 The Marlim field 8

1.2 Scope 10

1.3 Report outline 12

2 Literature study **13**

2.1 Control system structure 13

2.2 Dynamic optimization 15

2.3 Introduction to MPC 16

2.4 Linear MPC 17

2.5 Nonlinear MPC 18

2.6 Alternative MPC formulations 18

3 Modeling **21**

3.1 Deriving the well model . . . 21

3.1.1 Mass . . . 21

3.1.2 Mass flow . . . 22

3.1.3 Density . . . 25

3.1.4 Pressure . . . 26

3.2 Simplifications and assumptions . . . 29

3.3 Modifications . . . 29

3.4 Summary of the well model . . . 32

3.5 Model simulations . . . 33

3.5.1 Test simulations . . . 33

3.6 Manifold model . . . 36

3.7 Static model . . . 36

3.7.1 Precalculation of static model . . . 37

4 Methods **39**

4.1 Static optimization . . . 39

4.1.1 Objective function . . . 39

4.1.2 Constraints . . . 39

4.1.2.1 Stability regions . . . 40

4.1.3 Problem summary . . . 42

4.1.4 Solution . . . 43

4.2 Dynamic optimization . . . 45

4.2.1 Problem definition . . . 45

4.2.2 Different formulations (methods) . . . 46

4.2.3 Linear constraints . . . 47

4.2.4 Method summary . . . 48

4.2.5 Implementation . . . 48

4.3 Optimization free strategy . . . 48

5 Performance assessment **51**

5.1 Scenarios 51

5.2 Optimization results 53

5.2.1 Scenario 0 53

5.2.2 Scenario 1 55

5.2.3 Scenario 2 57

5.3 Solution times 58

5.4 Performance summary 60

6 Discussion **61**

6.1 Optimization and control 61

6.2 Modeling 61

6.3 Optimization methods and results 62

6.3.1 Static optimization 62

6.3.2 Dynamic optimization 62

6.3.3 Solvers 63

6.4 Hardware requirements 64

7 Conclusion **65**

8 Further work **67**

A Control system design **69**

A.1 Basic concepts and terminology 69

A.2 Modeling 70

A.3 Feedback control 72

A.4 Feedforward control 73

A.5 Decentralized control 74

A.6 Cascaded control 74

A.7 Compensators 75

A.8 Estimators 75

A.9 Optimal controllers 76

B Optimization **77**

B.1 Basic concepts . . . 77

B.2 Mathematical formulation . . . 78

B.3 Problem classification . . . 79

B.4 Linear Programming (LP) . . . 80

B.5 Quadratic Programming (QP) . . . 80

B.6 Nonlinear Programming (NLP) . . . 81

B.7 Branch and Bound . . . 81

C Alternative pressure model **83**

D Implementation **85**

D.1 Model implementation . . . 85

D.2 Static model implementation . . . 87

D.3 Stable regions . . . 89

D.4 Optimization free strategy . . . 91

D.5 Static optimization . . . 95

D.6 Dynamic optimization . . . 103

E Simulations of the scenarios **109**

E.1 Scenario 0 . . . 109

E.2 Scenario 1 . . . 111

E.3 Scenario 2 . . . 114

F Parameters and nomenclature **117**

List of Figures

1.1 Well with gas-lift [Eikrem 2008] 4

1.2 The Santos, Campos and Espirito Santo basins [Rigzone web b] 7

1.3 Development of the Brazilian oil production. [de Luca 2003] 8

1.4 Location of the Marlim field [Ribeiro 2005] 8

1.5 P-35 Production system [Storvold 2011] 11

1.6 Manifold 11

2.1 The control hierarchy 14

2.2 Applications with a control layer and a RTO layer 14

2.3 Introducing optimization in the control structure 15

2.4 MPC controller principles 17

3.1 Test simulation 34

3.2 Test simulation Eikrem model 35

3.3 Precalculated data ($m_{gt,ss}$, well 1) 37

4.1 Stability regions with proposed linear limits 41

4.2 Oil production as a function of well inputs [Binder 2011b] 43

4.3 Exponential weighting functions 47

5.1 Inputs and stability regions, scenario 0 54

5.2 Inputs and stability regions, scenario 1 56

5.3 Inputs and stability regions, scenario 2 57

5.4 Function value and step size as functions of iteration, method 2, scenario 1 . . 58

5.5 Function value and step size as functions of iteration, method 3, scenario 2 . . 59

5.6 Function value and step size as functions of iteration with a reduced step size tolerance, method 3, scenario 2 59

A.1 Feedback control 72

A.2 Feedforward controller 73

A.3 Feedforward from reference signal 74

A.4 Example of a cascaded control configuration 75

A.5 Estimator for a LTI system 76

B.1 Local versus global solution 78

E.1 Simulation of scenario 0 using dynamic otimization method 1 109

E.2 Simulation of scenario 0 using dynamic otimization method 3 110

E.3 Simulation of scenario 1 using optimization free method 111

E.4 Simulation of scenario 1 using static optimization method 111

E.5 Simulation of scenario 1 using dynamic optimization method 1 112

E.6 Simulation of scenario 1 using dynamic optimization method 2 112

E.7 Simulation of scenario 1 using dynamic optimization method 3 113

E.8 Simulation of scenario 2 using optimization free method 114

E.9 Simulation of scenario 2 using static optimization method 114

E.10 Simulation of scenario 2 using dynamic optimization method 1 115

E.11 Simulation of scenario 2 using dynamic optimization method 2 115

E.12 Simulation of scenario 2 using dynamic optimization method 3 116

List of Tables

4.1 Values for linear constraints 43

5.1 Routing, scenario 0 53

5.2 Inputs, scenario 0 53

5.3 Average production, scenario 0 54

5.4 Routing, scenario 1 55

5.5 Inputs, scenario 1 55

5.6 Average production, scenario 1 55

5.7 Inputs, scenario 2 57

5.8 Average production, scenario 2 57

5.9 Solution times (seconds) 58

5.10 Performance summary (average oil production, [bpd]) 60

5.11 Oil production at steady-state [bpd] 60

5.12 Oil production increase at steady-state using static optimzation (percent) . . 60

5.13 Oil production increase using dynamic optimization (percent) 60

F.1 General parameters 117

F.2 Well parameters 117

F.3 System constraints 118

F.4 Some conversion factors 118

Chapter 1

Introduction

This chapter first provides a background for the work described in this master thesis. The scope for this master thesis is defined, and an outline of the report is provided.

1.1 Background

This section gives an introduction to the petroleum industry, the history of offshore production, subsea production systems, artificial lift and production optimization. Also, a brief history of the Brazilian petroleum industry is provided, and a description of the Petrobras operated Marlim field, which is used as a case in this project. This should provide a foundation for the work described in this master thesis.

1.1.1 Petroleum production

This section gives a brief introduction to some basic concepts of petroleum production, and is mainly based on [Gunnerud 2011].

Petroleum is deposited in underground reservoirs, in porous rock formations, capped by non-permeable formations that prevent the hydrocarbons to escape to the surface. To produce oil and gas, wells are drilled into the reservoir. Driven by the pressure in the reservoir, fluids (usually a composition of gas, oil and water) flow through the well, usually through pipelines (flowlines), to a surface processing facility. The fluids are separated and treated. Hydrocarbon products are sold, water is treated and deposited for instance into the sea. Gas and water may be re-injected into the reservoir, to keep the pressure, and increase the recovery of oil from the reservoir.

Oilfields and their potential are usually discovered by seismic and geological surveys, in combination with exploration drilling. After discovery and initial information gathering, decisions need to be made as to how to produce the reservoir(s). This includes choice of technologies, and a long-term investment and production plan. There are large financial investments involved in petroleum exploration and production, and early return of investment is very important.

Typically, over time, the wells produce more gas and water compared to oil, due to depletion effects. This may to some extent be compensated for by drilling new wells, based on new knowledge of the reservoir from existing wells or seismic surveys. As the reservoir is produced over time, the pressure in the reservoir drops, and the production decreases. Injecting gas or water may help to preserve the reservoir pressure, and artificial lift methods (see section 1.1.4) may be implemented to increase the production and enhance the recovery from the reservoir. There are also other options to enhance the recovery from reservoirs, like chemical injection, microbial injection and thermal recovery, but this is not considered in this report.

1.1.2 The offshore industry

This section gives a brief introduction to the history of offshore oil production. The information is mainly gathered from [NOIA web].

A large amount of the world's oil and gas reserves are located in waters. One could say that the exploitation of these resources started at the end of the 1800's, when some "early oilmen" in California started building piers extending from land into the ocean, placing drilling rigs on them to drill wells in the water. Such piers after a while stretched as far as 350 meters into the ocean. The following years, the internal combustion engine boosted the consumption of gasoline, and lots of new technologies were developed to produce oil, from steel cables replacing ropes, to the development of modern seismology in 1926.

In this period, concrete platforms, artificial islands, steel barges, and fixed platforms were used to drill oil wells in shallow waters, close to land. The first oil discovery out-of-sight of land was made in the Gulf of Mexico (GoM) in 1947. This marked the beginning of the modern offshore industry, and by 1949, 11 fields were found in GoM.

The industry developed rapidly during the following decades, and by the end of the 1970's, there were more than 800 platforms in the GoM. New records of drilling in increasing water depths were continuously made. During the last decades, new technologies, such as 3D seismic and horizontal drilling, has laid the ground for new discoveries and exploitations.

As of February 2012, there were 671 rigs (barges, ships, jackups, platforms, semisubs, submersibles, and tenders) available in the total international competitive rig fleet [Rigzone web a], and oil is produced from water depths close to 3000 meters [Shell web].

Today, major offshore fields are found in the Gulf of Mexico, the North Sea, the Campos Basin and Santos Basin offshore Brazil, off Newfoundland and Nova Scotia in Canada, offshore West Africa, in South-East Asia, off Sakhalin in Russia and in the Persian Gulf.

1.1.3 Subsea production systems

This section briefly explains the basic concept of a subsea production system, based on [Sangesland 2007].

A subsea production system is a system that consists of subsea completed wells (in contrast to surface completed wells), and subsea equipment and control facilities to operate the well. The wells are remote from hands-on access, making maintenance complex and often very expensive. Control of the wells is achieved through umbilicals (hydraulic or electric control lines/cables), and produced fluids travel through subsea flowlines and risers to a surface production unit.

Subsea completion is usually preferred in large water depths, where rigid platforms and tension leg platforms (TLP) are not feasible solutions, due to construction requirements and costs. However, platform is the default solution for many wells in shallow water. Subsea completions may also be used:

- to extend the reach of existing platforms
- in marginal fields that cannot economically justify a platform
- to provide early production (subsea systems can often be installed faster than a conventional system)
- to provide information about the performance of a reservoir with a early and relatively low cost subsea development.

As an example, to gain an early assessment of the fields' productivity, Petrobras developed the first offshore fields temporarily, using subsea completions and flexible risers taking the produced fluids to floating production units. Fixed platforms were installed later. The general engineering concept was developed from models tested in the North Sea. [de Luca 2003]

1.1.4 Artificial lift

This section gives a brief introduction to artificial lift techniques, used to increase the production and enhance the recovery from reservoirs.

If the reservoir pressure is not sufficiently high to provide acceptable flow rates in the wells, the flow may be increased using artificial lift methods. There are mainly two kinds of artificial lift methods used; pump-assisted lift and gas-lift. Pump-assisted lift basically means installing pumps to increase the flow from the reservoir. (Offshore, this could be e.g. Electric Submersible Pumps (ESP), Hydraulic Submersible Pumps (HSP) or Jet Pumps.) The gas-lift technology, on the other hand, is based on injecting gas into the lower part of the production tubing, to decrease the hydrostatic pressure drop in the well. Both methods (gas-lift and pump-assisted lift) seek to reduce the backpressure in the wellbore caused by flowing fluids in the production tubing, and in this way increase the inflow from the reservoir, and increase the production. [Hu 2004]

Figure 1.1 shows an oil well with gas-lift. Gas flows through a gas-lift choke valve, into the casing-tubing annulus. Close to the reservoir is an injection valve. This is a one-way valve, and gas is only injected when the pressure in the annulus is higher than in the production tubing. The injected gas flows together with the produced fluids through the production tubing, and through the production choke valve. The injected gas in the production tubing (which is basically just a vertical pipe) causes a decrease in the average density of the fluids in the tubing, and thus a reduced hydrostatic pressure drop. The effect is a lower bottomhole pressure in the tubing, which causes a higher inflow rate of fluids from the reservoir, and thus an increased production.

Gas-lift can be used both with naturally flowing wells, to increase the production, and with dead wells (wells that do not have any natural flow driven by reservoir pressure), to make them produce. It is not the most efficient artificial lift technique, but it is often more technically and economically feasible [Lorenzatto 2004, Hu 2004]. Some advantages of gas lift are [Hu 2004, PSC web]:

- Gas-lift requires few moving parts, and therefore is suitable also when solids (such as sand) are produced.
- Gas-lift works well in a well with a multi-inclination trajectory, where installing a bottomhole pump may be difficult,.
- Gas-lift wells have downhole equipment with low cost and long service life. The major equipment is the gas compressor, which is located on the surface (offshore: on the production unit), which allows for easy maintenance, while the downhole equipment mainly consists of valves.

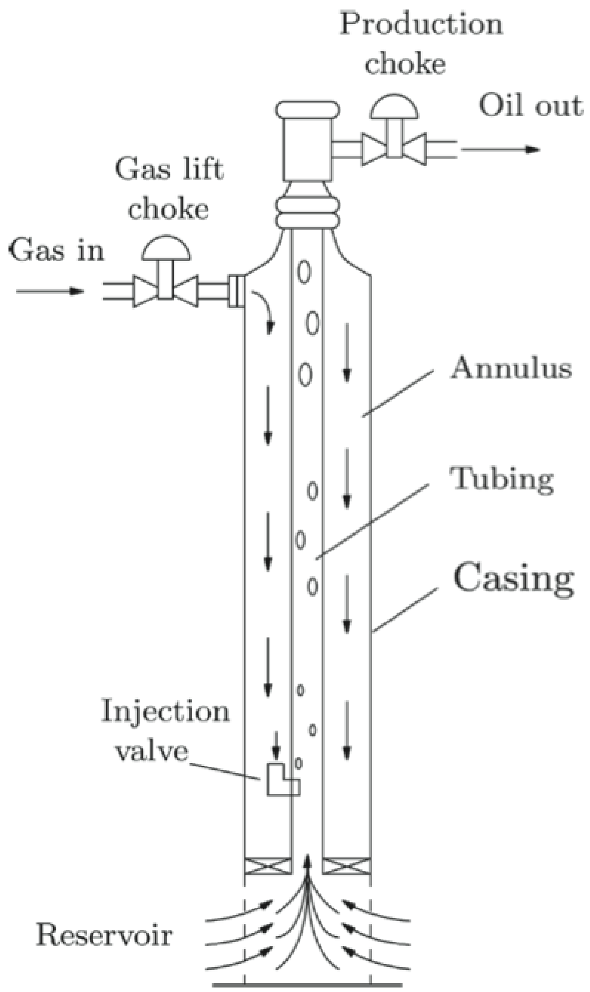


Figure 1.1: Well with gas-lift [Eikrem 2008]

- Gas-lift is very flexible to changes in well conditions and production rates.

There are also some problems related to using gas-lift for artificial lift. First, gas is circulated through the wells, which requires a high gas compressor capacity, as more gas is compressed than what is produced. There is also a problem related to the stability of the production from the wells, known as casing heading instability. This is a problem where wells with gas-lift show an oscillatory behavior, where production levels vary greatly with time. The chain of events in this case may be explained as follows:

1. No gas is injected into the tubing, as the annulus pressure is lower than the tubing pressure.
2. Gas is injected into the annulus to build up the annulus pressure.
3. When the annulus pressure supersedes the tubing pressure, gas is injected into the tubing.
4. The pressure in the tubing decreases significantly, as the injected gas causes a lower hydrostatic pressure drop in the tubing.
5. The pressure drop in the tubing causes even more gas to be injected into the tubing.

6. The pressure in the annulus also drops, as more gas is injected into the tubing than what is injected into the annulus.
7. As the annulus pressure drops, less gas is injected into the tubing
8. This causes the tubing pressure to build up, and even less gas is injected into the tubing.
9. Finally, the tubing pressure supersedes the annulus pressure, and no gas is injected into the tubing, and the cycle repeats itself.

This instability causes problems for downstream processing equipment, and also causes a loss of production compared to a stable production. The phenomenon occurs only in some wells, and only at low gas-lift rates in combination with a high production choke opening. This phenomenon, and using a controller to stabilize the wells, was investigated in my previous work [Binder 2011b], and is also discussed in [Xu 1989, Hu 2004, Eikrem 2008].

Gas-lift is widely used in Brazil. More than 80 % of the wells in the Campos basin employ gas-lift [Pinto 2003]. However, there are several ongoing research- and pilot projects on pump-assisted lift. According to [Pinto 2003], this "represents a complete cultural change".

1.1.5 Production optimization

This section puts production optimization into a context, and points out some necessary decisions to optimize the production from a field. This section is based on [Gunnerud 2011].

There are many decisions and factors that affect the production from a petroleum reservoir, a production system, or a well. This includes reservoir geology, placement and geometry of wells, production system type and structure, subsea or surface completion, processing facilities and capacities, choke openings, and so on. Decisions also need to be made whether or not to use artificial lift, or any other technique, to increase the production and/or enhance the recovery of petroleum from the reservoir. Such decisions are made based on available information about the field, which may be gathered from seismic explorations, exploratory wells/test producers, and from production over time.

Development of a field requires planning on different time scales. On a long-term time horizon, strategic decisions are made. This includes choice of technology (e.g. subsea installations), export options (pipelines or tankers), investment strategies and recovery strategies. Typically, analyses and development plans seek to maximize the net present value (NPV) of the field. Over time, it may be necessary (or profitable) to consider new investments in the production system, based on new knowledge or technology. The production or recovery from the reservoir may be improved by installing a subsea separator, drilling new wells, invest in new processing equipment, installing pumps, etc.

On a medium term time horizon (months to a couple of years), production rates (and possibly injection rates) are decided. Depending on the stage of the field development, drilling programs may also be an issue, including placement and completion of wells. Artificial lift may also be considered. Reservoir models are often important tools on this time horizon.

On a short term time horizon, days to weeks, decisions need to be made as to how much one should produce from each well, determined by the opening of the production choke valves located at the wellheads. If gas-lift is used to increase the production from the wells (see section 1.1.4), one needs to determine how much gas to inject into each well. These are non-trivial decisions, as the capacities for the processing equipment may impose limits as to how much gas, water or liquid that may be produced, and capacities of compressors may limit the availability of gas available for injection. The routing of produced fluids through the production and processing system also must be decided [Binder 2011b]. Decisions on this time-scale rely on well data, while the reservoir, production system and processing facilities form boundary conditions and constraints on the decisions. Well parameters, such as water cut, gas-to-oil ratio, and so on, may vary with time, or with different production rates. To make good decisions, it is important to monitor the wells, using available continuous measurements, and performing well tests, but also to find useful and relevant information from these data.

The amount of data in a production system may be enormous, and today, not all of these data are exploited to make good decisions. Models and methods to use this information are therefore very welcome in the industry. Formulating and solving mathematical optimization problems may be useful tools for making good decisions, on each of the above mentioned time horizons.

1.1.6 Petroleum in Brazil

This section gives an introduction to the history of the Brazilian oil industry. The main content of this section is found in [de Luca 2003], where a more extensive and detailed history is presented.

The first effort of finding petroleum in Brazil dates back to 1897, and was done by a farmer/landowner. After drilling a well to a depth of 488 meters, only two barrels of oil were produced, and no further efforts were made. In the period of 1921-1933, about 65 exploration wells were drilled by the Brazilian Geologic and Mineral Service, without any results. The state of São Paulo also made some efforts of petroleum exploration in the period of 1927-1929, but this did not result in any findings either. Some small, private petroleum firms also made some efforts, but likewise without any positive results. In 1930, the Provisional Government decreed the end of state activities in petroleum exploration.

However, in 1934, the Brazilian Constitution established that all underground resources belonged to the Federal Government, who re-initiated (limited) exploratory works. From 1934 to 1938, only six wells were drilled. In 1938, the National Petroleum Council (CNP) was established, taking responsibility of establishing and carrying out the country's petroleum policy.

In 1941, the first commercial Brazilian oilfield, the Candeias field, was discovered in the Recôncavo basin in the state of Bahia, at the east coast of Brazil. In the following years, several other fields were discovered in the same region, and Recôncavo became the first oil province in Brazil, with 15 million barrels of oil registered. From 1938 to 1954, 52 exploratory wells were drilled, mostly in the Recôncavo district, but also in other regions.

In 1953, the Brazilian Government established the company Petróleo Brasileiro S.A. (Petrobras), which received all assets belonging to CNP, and had the monopoly of oil production in Brazil. With Petrobras, the oil industry became more business oriented, and many foreign professionals and managers were contracted. The known fields in Recôncavo were developed, and further explorations were made in other districts in the following years. Further onshore discoveries in different regions were made in the following decades; the Carmópolis field in Sergipe in 1963, Uburana and Mossoró in the state of Rio Grande do Norte in 1974 and Urucu in the Amazon region in 1988, to mention some.

Petrobras started their offshore explorations in 1967. At that time, the water depth limit for drilling was around 100 meters. Starting from 1968, some offshore discoveries were made, mostly of very small fields. Production from the first offshore field started in 1973. In the seventies, several discoveries were made in the Campos basin, on the continental shelf of the state of Rio de Janeiro. In 1988, several offshore fields were also discovered in the Santos basin offshore São Paulo, at water depths of less than 200 meters. Production from these fields started in 1991. The location of the Campos and Santos basins are shown in figure 1.2.

Figure 1.2: The Santos, Campos and Espirito Santo basins [Rigzone web b]

From 1984, using a dynamic-positioning drillship, Petrobras started exploring larger water depths in the Campos basin, up to 1000 meters. Three giant oilfields were discovered, among them the Marlim field, Brazil's largest oil field, discovered in 1985. These fields required large investments and advanced technology to be developed. The Petrobras Research and Development Center (CENPES) started technology programs aiming to enable production in such water depths, first up to 1 000 meters, later up to 2 000 meters. In 1994, Petrobras completed a well at 1 027 meters water depth, a new world record. Petrobras continuously set new records, and completed a well at 1 709 meters already in 1997.

A law from 1997 ended Petrobras' monopoly on exploration, production and transportation of petroleum in Brazil, and the National Petroleum Agency (ANP) was created as a regulatory body. Leasing attracted around 50 new companies to Brazil the first five years after this transition.

In 1938, Brazil imported all oil consumed in the country, 38 000 bpd (barrels per day) in total. In 1961, they produced 90 000 bpd, which then accounted for 40% of the nation's consumption. During the early seventies, the total onshore production reached 160 000 bpd. With the offshore explorations and developments, the production grew rapidly, especially with the developments in deep water, reaching 1.5 million bpd in 2003 [de Luca 2003]. Brazil attained self-sufficiency of oil in 2005 [Johann 2011]. In 2011, Brazil produced 2.52 million bpd on average [CIPEG web]. Figure 1.3 shows the development of petroleum production in Brazil from 1940 to 2002, divided into four phases:

1. Early phase, with CNP as developer
2. Petrobras activities onshore
3. Offshore activities in shallow waters
4. Offshore activities in deep and ultra deep waters

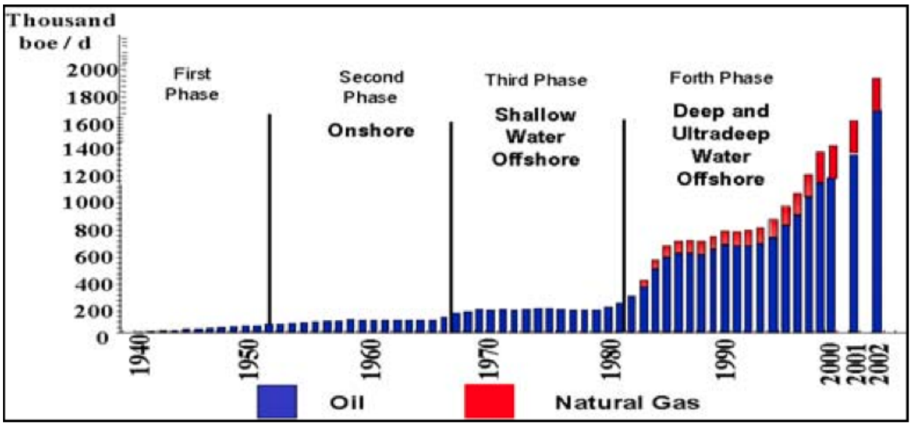


Figure 1.3: Development of the Brazilian oil production. [de Luca 2003]

Furthermore, in 2006, Petrobras announced major discoveries in the Santos basin, in the so-called pre-salt region. Extremely large possible accumulations there have been estimated, from 50 to over 300 billion barrels of recoverable oil, which alone yields enormous potential for the future of Brazilian petroleum production. [Jones 2011]

1.1.7 The Marlim field

This section describes the Marlim field, which is used as a case in this project.

The Petrobras operated Marlim field was discovered in February 1985. It is the largest producing field in Brazil [de Luca 2003], located in the northeastern part of the Campos basin, neighboring the Marlim Leste (Marlim East) and Marlim Sul (Marlim South) fields, about 110 km offshore the state of Rio de Janeiro (see fig. 1.4). The field covers about 145 km^2, and the water depth ranges from 600 to 1 200 meters [Bampi 2010].

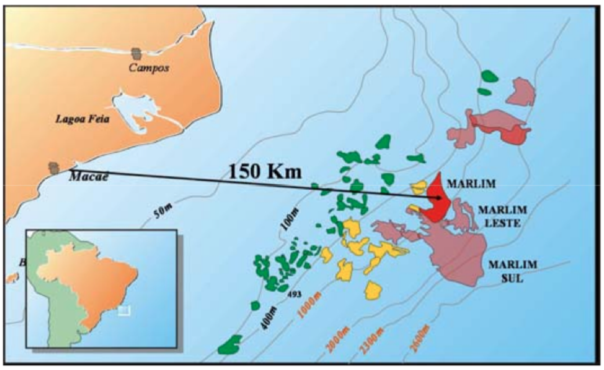


Figure 1.4: Location of the Marlim field [Ribeiro 2005]

The first oil from the field was produced in March 1991, and an economic life of 20 years, with a peak production of 430 000 bpd (barrels of oil per day), were estimated for the field [Oliveira 1989]. In 2005, 8 floating production units, with 132 wells and 5 subsea manifolds, reached a peak total production of 650 000 bpd [Ribeiro 2005]. In April 2010, oil production averaged 282 000 bpd, with a cumulative production of 2 billion barrels, representing an oil recovery of 32 % [Bampi 2010].

Even though the initial estimate of the economic life of the field was 20 years, Petrobras seeks to extend this to 40 years. Among other important aspects, this relies heavily on technological development. [Lorenzatto 2004]

Due to the large water depth, the Marlim field is developed using floating production units, subsea manifolds and satellite wells. Today, 4 semi-submersible platforms, and 5 Floating Production Storage and Offloading vessels (FPSO's) are used to produce the field. [Lorenzatto 2004] Seawater is injected to maintain reservoir pressure, thus increasing the recovery. The water injection started in 1994, three years after the first oil was produced from the field. The field has more than 200 wells, of which 125 are in operation, and there are 1.85 producers per injector. [Bampi 2010]

More than 80 % of the wells in the Campos basin employ gas-lift [Pinto 2003]. It is not the most efficient artificial lift method, but it is considered the most technically and economically feasible. Pump-assisted lift has also been considered for the Marlim field in the recent years, with research and pilot projects. [Lorenzatto 2004]

1.2 Scope

The work in this master project is based on previous work, described in [Binder 2011a] and [Binder 2011b]. These reports concentrate on modeling and simulation of casing heading, implementation of PI-controllers to stabilize casing heading, and implementation of a non-linear static optimization algorithm to increase the production in a specific production system. The reader is referred to [Binder 2011b] for more information on previous work.

This project work focuses on using mathematical modeling and optimization as tools to increase the production from a production system. In particular, static and dynamic optimization is compared, to investigate possible benefits of introducing dynamic optimization-based algorithms in the production system. The problem to be solved throughout this project is to find the routing and input settings (choke openings and gas-lift rates for each well) that yield the highest rate of oil production, while respecting physical limitations and production constraints. The methods are tested on certain test scenarios to assess the performance of the methods.

The production system considered in this project is inspired by a part of the production system producing to the FPSO (Floating Production, Storage and Offloading unit) P-35 (Petrobras 35) in the Marlim field. As seen in figure 1.5, this production system consists of two subsea manifolds, 15 subsea completed wells, and surface processing equipment on the FPSO, including one surface manifold, and three separators. 6 of the wells are satellite wells, producing to the surface manifold. 7 wells produce to one subsea manifold, and 2 wells produce to the other. All wells produce with gas-lift (see section 1.1.4).

The considered system in this project consists of the subsea manifold shown as Subsea Manifold 1 in figure 1.5, and the seven wells producing to it. In the manifold, the production from each well is controlled using choke valves, one for each well. There are two flowlines taking produced fluids from the manifold to the surface, and each well may be routed to either one of these flowlines using on/off valves. Compressed gas from the surface, for gas-lift, is also distributed to the wells using choke valves, one for each well. Figure 1.6 shows the considered manifold with three of the wells.

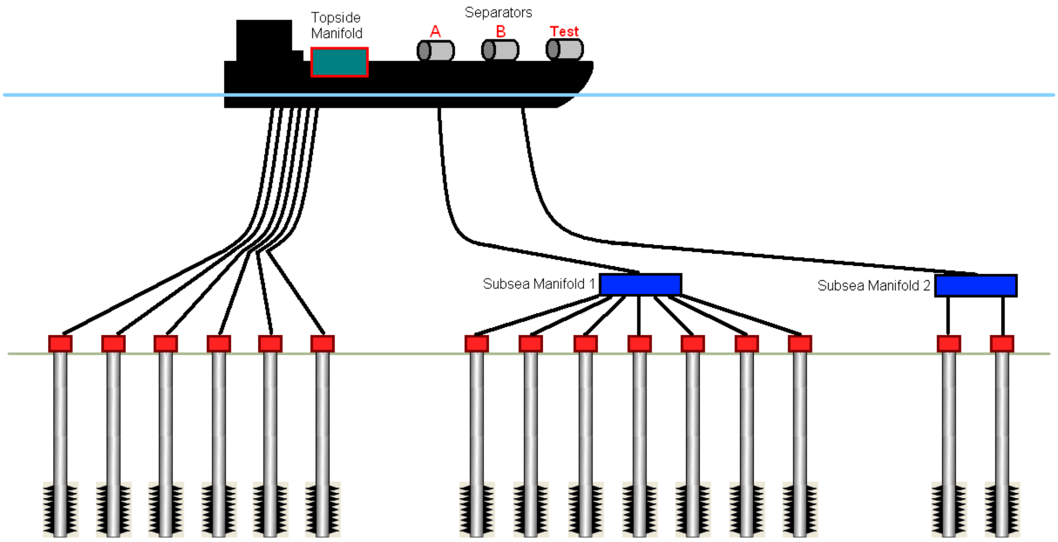


Figure 1.5: P-35 Production system [Storvold 2011]

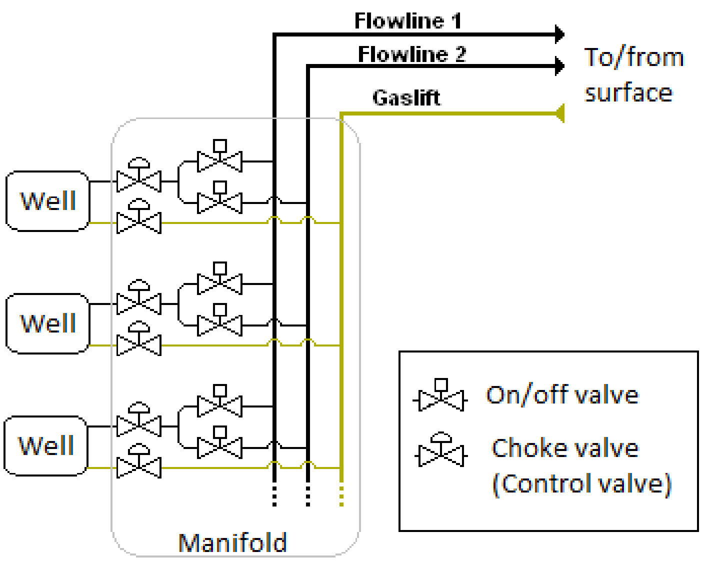


Figure 1.6: Manifold

1.3 Report outline

This report is organized as follows:

- **Chapter 2** presents a summary of a literature study on control systems and optimization.
- In **chapter 3**, a dynamic well model is derived, and a model for the considered system (the manifold) is presented.
- The proposed production optimization methods are formulated and presented in **chapter 4**.
- The assessment of the optimization results is given in **chapter 5**.
- **Chapter 6** provides some comments on hardware requirements for production optimization.
- The results are discussed in **chapter 7**, a conclusion is made in **chapter 8**, and further work is proposed in **chapter 9.**

Chapter 2

Literature study

This chapter provides a literature review on methods of introducing optimization in the control of dynamic systems. The main objective in a control problem is to maintain acceptable operation of the system/plant. This may be in terms of safety, load on operators, environmental impact and so on. Secondly, optimizing the performance of the system (often economical) is of concern [Skogestad 2005]. Optimization techniques, discussed in appendix B, may prove useful in many aspects of the control problem. The study in this chapter is limited to methods of particular interest to this master project. Further background on the topics may be found in appendices A and B, and in literature on control theory and optimization, such as [Skogestad 2005] and [Nocedal 2006].

2.1 Control system structure

In existing production facilities and similar dynamic systems, the control system structure is typically divided into layers, defined by frequency and type of decisions. This is called a “control hierarchy”, and is illustrated in figure 2.1. In a control hierarchy, a layer’s goal is defined by the layer above, and a layer sends commands or setpoints to the layer below. The name, tasks and frequency of each layer vary a lot in the literature, and from plant to plant, but the concept of a division into layers is commonly found in many applications [Maciejowski 2002, Skogestad 2005].

Many applications are structured with a layer referred to as the control layer. The main purpose of the control layer is typically to stabilize the plant, and to track setpoints given by the above layer. The control layer is typically populated by conventional PID controllers and logic, but may also embed MPC controllers (see section 2.3) for setpoint tracking. The control layer may also be sub-divided into a supervisory and a regulatory control layer, where the regulatory control typically is performed by PID controllers, and the supervisory control is based on more advanced control, e.g. MPC.

The setpoints for the control layer are decided in the above layer, by plant operators, optimization software or a combination of these. This layer is often referred to as the real-time optimization (RTO) layer. Optimization in this layer is typically based on steady-state models of the plant, i.e. models of the system’s long-term, stable response to the input signals.

The general structure with a control layer and a RTO layer is shown in figure 2.3b on page 15. Figure 2.2 shows two examples of control structures with a control layer and a RTO layer.

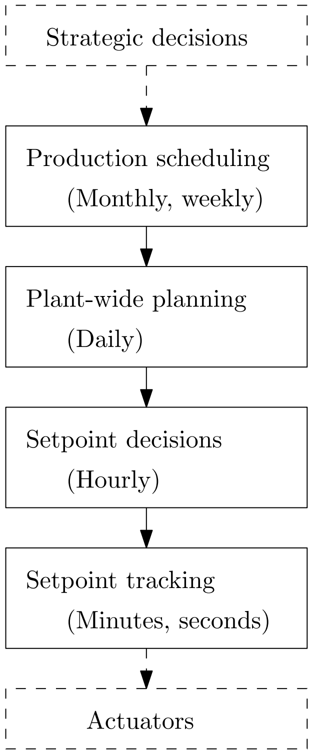


Figure 2.1: The control hierarchy

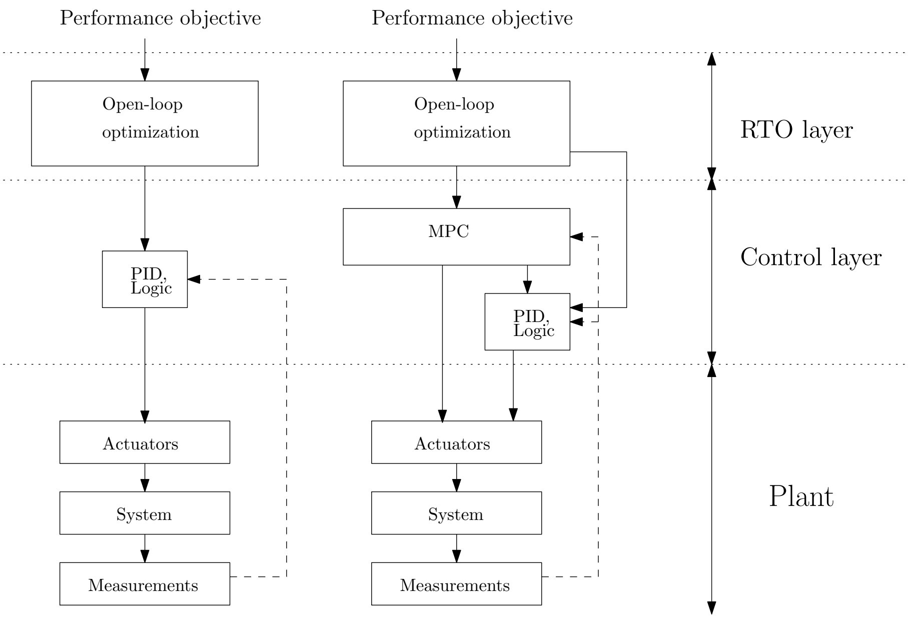
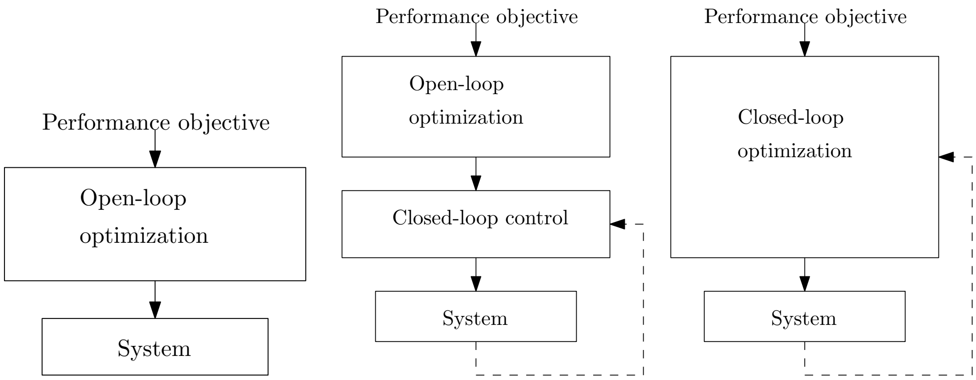


Figure 2.2: Applications with a control layer and a RTO layer

There are, of course, other possible structures. In some processes, feedback control may not be necessary at all, and a structure as shown in figure 2.3a may be implemented. This kind of optimization is usually based on steady-state system models. However, due to model errors and unmeasured disturbances, this usually does not yield acceptable performance. To handle this, feedback control is often implemented in a control layer, as discussed above and shown in figure 2.3b. Note that with such a control structure, feedback is only introduced in the control layer, not in the optimization. (However, the models used in the optimization layer may be updated using data gathered from process measurements.) A third option is to implement an optimizing controller as shown in figure 2.3c. With this structure, the control layer and the above optimization layer are combined. The performance objective is embedded in the controller, as are the dynamic system models and measurements from the system. Theoretically, this yields the more optimal performance, as all control actions are coordinated and optimized with respect to the real performance objective. However, this structure is rarely used, due to the modeling effort required, the complexity of controller design, difficult modification and maintenance, robustness problems, operator acceptance, and the lack of computing power [Skogestad 2005].



(a) Open-loop optimization (b) Open-loop optimization with closed-loop control (c) Closed-loop optimization

Figure 2.3: Introducing optimization in the control structure

The division into layers is common, and is probably a result of the different time scales the decisions are made on. For example, for logistic purposes, it may be necessary to plan what products to produce weeks ahead, while it is of no interest what a valve's position will be in a few weeks. The most obvious benefit of this layering is that the complex problem of optimizing the plant's performance is divided into smaller subproblems, which by themselves are less complex to solve. Another important benefit is that the performance of each layer is quite independent from the other layers, as they are divided by frequency. However, the overall performance may suffer from dividing the overall performance optimization problem into sub-problems. In general, dividing an optimization problem into subproblems may yield a suboptimal solution for the original problem. This can be summarized as a conflict of interest between simplicity and performance. [Maciejowski 2002, Skogestad 2005]

2.2 Dynamic optimization

As mentioned in section 2.1, it is common to divide the control structure into a control layer, and an optimization layer. Optimization is then typically based on steady-state models of the plant, i.e. models of the system's long-term, stable response to the input signals. The control layer handles the dynamics of the system. This was shown in figure (2.3b). Alternatively, dynamic models may be included in the optimization layer, so that a predicted optimal input trajectory may be calculated and implemented on the system, substituting the control layer. This is often referred to as dynamic optimization. Although the procedure may be repeated when new information is available, feedback is in general not included in a dynamic optimization structure. Introducing feedback, and what is known as the receding horizon principle, leads to the MPC controller, discussed in the following sections.

2.3 Introduction to MPC

This section is based on [Maciejowski 2002, Imsland 2007].

Model Predictive Control (MPC) is said to be the only advanced control method widely used in the industry. So far, it is mainly applied to petrochemical industries, but it is increasingly applied to other industries. The basic principle of MPC is formulating an optimization problem based on dynamic system models and real-time measurements to find the optimal inputs to the system. The procedure of an MPC controller is divided into three steps, as follows:

1. Based on dynamic system models and available measurements, calculate the optimal future inputs to the system, subject to constraints on the system
2. Implement the first input (or inputs in a multivariable system) of this sequence
3. Wait for new measurements, shift the time ($k = k + 1$, receding horizon), then repeat from step 1

The main benefits of a MPC controller are:

- It is highly performance oriented, as it applies optimization to find the optimal inputs
- It handles multivariable control problems naturally
- It handles constraints, both on inputs and outputs
- Feedback is naturally included, and feedforward may also be included

Because the MPC controller handles constraints, it often allows for operation closer to certain limitations regarding safety or quality requirements, compared to conventional feedback control. This may often lead to more profitable operation. However, there are also some significant disadvantages with MPC, including:

- Solving the optimization problem requires time and computational power, and is performed on-line
- Modeling effort is required to formulate reliable system models, and the performance is sensitive to uncertainties
- The performance is sensitive to tuning parameters in the controller, which do not necessarily have a local or direct effect, thus may be difficult to understand
- It is not as intuitive as conventional feedback control, maintenance and modifications may require skilled personnel

Because of the time and computational power required by the MPC controller, it is traditionally implemented in relatively slow dynamic systems. However, because of the increased computational power of modern hardware and improving solution algorithms, MPC is implemented for increasingly faster processes.

As mentioned, the MPC controller solves an optimization problem based on system models, and a performance objective. The performance objective is usually to track a reference setpoint, or a reference trajectory. To assess the future performance, the system model is used to predict the future behavior of the system as a function of the future inputs, which are the decision variables in the controller's optimization problem, hence the name. The predictions are made for a predefined amount of time into the future, known as the *prediction horizon*.

The MPC principle is illustrated in figure 2.4. Here, a change in the setpoint has recently occurred. The MPC controller has calculated the future inputs that yield the best response on the prediction horizon, according to some evaluation criteria given by the objective function. The historical data are shown to the left, and the prediction horizon with predictions to the right. Note that the prediction of the output starts at the most recent measurement. This is always the case, and in this way, feedback is introduced in the controller.

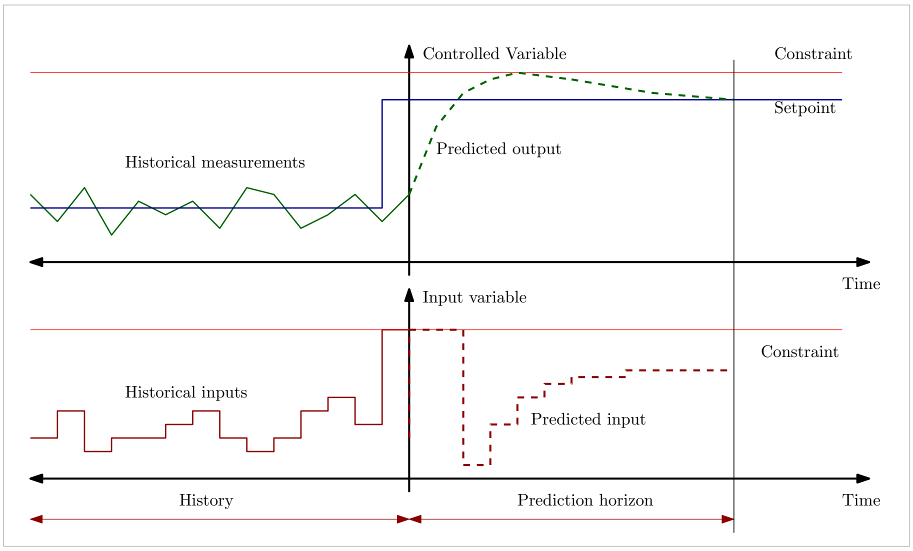


Figure 2.4: MPC controller principles

The MPC controller is run at fixed time intervals (time steps). Even though the predictions are made for the entire prediction horizon, only the first predicted input is implemented. At the next time step, it is assumed that a new measurement is available, and the prediction horizon is shifted one time step, before the optimization is run again. This is referred to as the *receding horizon* principle.

2.4 Linear MPC

This section is based on [Maciejowski 2002, Imsland 2007].

There are two classes of MPC controllers, classified by the internal models used; linear and nonlinear MPC. Linear MPC is most common. In linear MPC controllers, the model is a linear discrete-time model, e.g. on the form:

$$x_{k+1} = Ax_k + Bu_k \tag{2.4.1}$$

where x_k and u_k are vectors. Assuming the goal is to control the variables in the vector x to a constant setpoint at zero, while using the input as little as possible (because there is a cost related to the usage of the input variable), the objective function to be minimized by the controller may be formulated as:

$$\min_{u_k} \sum_{k=0}^{N-1} x_{k+1}^\top Q x_{k+1} + u_k^\top R u_k \tag{2.4.2}$$

where Q and R are weighting matrices, used to describe the relative importance of each variable. These are tuning variables, but they may be related to cost or profit. N is the number of time steps in the prediction horizon. The model, given by equation (2.4.1), enters the optimization problem as a constraint, together with a measurement of x at $k = 0$ (x_0 is given). In addition, there may be (linear) constraints on the inputs and outputs, e.g. on the form:

$$\begin{array}{rcll} D_x x_k & \leq & d_x, & k = 1, \dots, N \\ D_u u_k & \leq & d_u, & k = 0, \dots, N-1 \end{array} \tag{2.4.3}$$

These three equations constitute the optimization problem to be solved by the controller at each time step. This can be shown to be a quadratic programming (QP) problem, and algorithms for solving QP problems are implemented in linear MPC controllers. (For a description of QP problems and the solution of these, see appendix B.5.)

2.5 Nonlinear MPC

This section is also based on [Maciejowski 2002, Imsland 2007].

If there are severe nonlinearities in a plant, linearization of the model and a linear MPC controller may provide a poor performance, as the optimization problem solved is a poor representation of the real problem. The solution to this may be to use a nonlinear model in the controller. This is referred to as nonlinear MPC (or NMPC). NMPC is based on the same principles as linear MPC described in section 2.4. However, the model given by equation (2.4.1) is replaced by a nonlinear function on the more general form

$$x_{k+1} = f(x_k, u_k) \tag{2.5.1}$$

Furthermore, the constraints on inputs and outputs in NMPC are not necessarily linear, and the objective function does not have to be quadratic.

The advantage of a NMPC controller is that the optimization problem solved each iteration is a better approximation of the real system, than with a linear model. However, due to the nature of the arising optimization problem, convexity is usually lost, and it can no longer be solved as a QP problem, but must be solved as a NLP problem, using e.g. the SQP algorithm. (Solving NLP problems is discussed in appendix B.6.) Also, it is very difficult to predict the amount of time required to solve the optimization (though it is usually increased significantly), or if the solution found will be local (suboptimal) or global, or even whether it will terminate at a solution at all. NMPC formulations are difficult, or even impossible, to analyze, but has led to perfectly acceptable performance in many practical applications.

A compromise solution between linear and nonlinear MPC is to re-linearize the model if the operating point of the plant changes significantly, and use the latest linear model as the internal model. An extension to this is to use a time-varying linear model, if the operating point is expected to shift due to e.g. setpoint changes.

2.6 Alternative MPC formulations

In this section, the degrees of freedom in the formulation of the MPC controller is discussed, mainly based on [Maciejowski 2002, Imsland 2007].

Although all MPC formulations share the structure with an objective function and a system model as constraint, and a receding horizon, there is a wide range of alternative formulations of the MPC problem, other than the ones presented in section 2.4 and 2.5. In order to be solved by a QP algorithm, the linear MPC is required to take on a quadratic form, i.e. the objective function must be quadratic, while all the constraints, including the model, must be linear. However, there are still many degrees of freedom, including:

- Variables in the model
- Variables in the objective function
- Weighting of variables
- Prediction horizon
- Additional constraints
- Sampling frequency (time step size)
- Input blocking
- etc...

There are many alternatives when choosing the model variables. Variables of no interest to the performance of the system may not have to be included in the MPC formulation at all, depending on the system model. Alternatively, the corresponding elements of the weighting matrix may be set to zero. If variables of interest are not measured, an estimator may be implemented, and estimated variables included in the problem. Fast dynamics in the system may be controlled by e.g. a PI controller, and the setpoint for the PI controller rather than the actual input may be included in the MPC formulation. Feed-forward may be introduced by adding a disturbance model in the formulation. An unstable system may be stabilized by a basic feedback controller, and the decision variables in the MPC formulation could then simply be the deviance from the input calculated by such a feedback controller.

The choice of variables in the objective function will affect the controller performance. The objective function does not have to include every variable in every time step. Some special formulations, such as mean-level control, deadbeat control and "perfect" control, are based on choosing only a few (or even just one) variables in the objective function formulation. The change of the inputs can be implemented as decision variables instead of or in addition to the actual input value. In this way, the rate of change may be constrained and/or penalized. To reduce the number of decision variables in the optimization problem, the input variables may be constrained to be constant during certain intervals of the prediction horizon. This is referred to as input blocking.

Weighting of the variables in the objective may have a great impact on the system's performance. Variables with a low weight will be prioritized less by the controller than variables that are more heavily weighted. The weighting is often, but does not have to be, chosen to be constant over the prediction horizon.

The prediction horizon may influence both the performance and the stability properties of the controller. Stability is ensured with an infinite prediction horizon. (An infinite prediction horizon is enabled by using the Lyapunov equation.)

It is also possible to add additional constraints to the optimization problem. For example, to ensure stability of the controller, one can simply add a "terminal constraint". This requires the predicted state to take on a certain value at the end of the prediction horizon

The sampling frequency should be chosen based on the underlying system's dynamics. It should be chosen fast enough to predict the system dynamics accurately, but slow enough to enable the controller to solve the optimization problem between each sample. Increasing the sampling frequency will increase the number of decision variables in the optimization problem.

The nonlinear MPC does not have the same constraints on the formulation as the linear MPC, and thus have further degrees of freedom, both in the formulation of the objective function and the constraints. However, the solution of the optimization problem is more complex, and usually more time consuming.

Chapter 3

Modeling

The system defined in this project consists of one subsea manifold, and seven wells (see section 1.2). Each of the wells employ gas-lift for artificial lift (see section 1.1.4 and figure 1.1 on page 4). Eikrem et. al. has developed a non-linear model for gas lift wells, presented in [Eikrem 2008]. A modified version of this model was used in [Binder 2011b], and a further modified version is also used in this project. In [Eikrem 2008], the model is only presented as a final result. In this report, the derivation of the model is explained in detail. The model derivation with assumptions and simplifications is found in section 3.1, some major simplifications are further discussed in section 3.2, the exact modifications made to the 'Eikrem model' is discussed in section 3.3, a summary of the well model is found in section 3.4, a description of how simulations of the model are performed, and a test simulation of the model is shown in section 3.5, a model for the manifold is described in section 3.6, and the implementation of a static model based on the dynamic model is described in section 3.7.

3.1 Deriving the well model

This section explains how the model for the well with gas-lift shown in figure 1.1 is derived. A summary of the model is found in section 3.4, and a complete nomenclature is found in appendix F.

3.1.1 Mass

Considering mass balance, the change of mass in a defined volume equals the total mass flow into the volume, minus the total mass flow out [Egeland 2002]:

$$\frac{d}{dt}(\rho V) = w_{in} - w_{out} \tag{3.1.1}$$

Using $m = \rho V$, this is written:

$$\dot{m} = w_{in} - w_{out} \tag{3.1.2}$$

where the dot notation $\dot{}$ denotes differentiation with respect to time.

This model has three state variables: the mass of gas in the annulus volume (m_{ga}), mass of gas in the tubing volume (m_{gt}), and mass of liquid (oil and water) in the tubing volume

(m_{lt}). The annulus volume V_a and the tubing volume V_t are defined as the respective volumes *above the point of gas injection*, i.e. where the injection valve is installed. (See figure 1.1 on page 4.) Using the mass balance in equation (3.1.2), the dynamics of the state variables are modeled:

$$\dot{m}_{ga} = w_{gl} - w_{gi} \tag{3.1.3}$$

$$\dot{m}_{gt} = w_{gr} + w_{gi} - w_{gp} \tag{3.1.4}$$

$$\dot{m}_{lt} = w_{lr} - w_{lp} \tag{3.1.5}$$

Here, w_{gl} is the mass flow of lift gas into the annulus (through the gas-lift choke valve), w_{gi} is the mass flow of gas from the annulus into the tubing (through the injection valve), w_{gr} is the mass flow of gas from the reservoir into the tubing, w_{gp} is the mass flow of gas produced from the tubing (through the production choke valve), w_{lr} is the mass flow of liquid (oil and water) from the reservoir into the tubing , and w_{lp} is the mass flow of liquid produced from the tubing (through the production choke valve).

In addition, the total mass in the tubing volume is defined as:

$$m_t = m_{gt} + m_{lt} \tag{3.1.6}$$

and the total mass flow through the production choke valve is defined as:

$$w_p = w_{gp} + w_{lp} \tag{3.1.7}$$

3.1.2 Mass flow

This section describes how the models for the mass flows w_{gi}, w_{gl}, w_{gp}, w_{gr}, w_{lp} and w_{lr} are derived.

The volume flow q through a orifice in a valve is generally turbulent, and is given by:

$$q = C_d A_o \sqrt{\frac{2}{\rho}\Delta p} \tag{3.1.8}$$

where A_o is the cross section area of the orifice, Δp is the pressure drop through the valve, ρ is the density of the fluid flowing through the valve, and C_d is a constant discharge coefficient, normally in the range between 0.6 and 0.9 [Egeland 2002]. Using the relationship $w = \rho q$, the mass flow through a valve may then be modeled as:

$$w = \sqrt{2} C_d A \sqrt{\rho \Delta p} \tag{3.1.9}$$

Assuming the orifice area is given by a valve specific function ($f(u)$) of the choke setting (u) and the maximum orifice area (A_{max}):

$$A = A_{max} f(u) \tag{3.1.10}$$

and defining a valve specific constant:

$$C_v = \sqrt{2} C_d A_{max} \tag{3.1.11}$$

the mass flow through a valve is modeled as:

$$w = C_v \sqrt{\rho \Delta p}\, f(u) \tag{3.1.12}$$

The choke setting must be in the range [0,1], and the function $f(u)$ must satisfy:

$$\begin{array}{rcl} f(u) & \in & [0,1] \quad \forall u \in [0,1] \\ f(0) & = & 0 \\ f(1) & = & 1 \end{array} \tag{3.1.13}$$

For a linear valve characteristic, we have:

$$f(u_{lin}) = u_{lin} \tag{3.1.14}$$

Alternatively, if the nonlinear characteristic is known, the choke setting as the input variable may be replaced by a linearizing mapping $u = g(u_{lin})$ so that $f\left(g\left(u_{lin}\right)\right) = u_{lin}$, and the choke opening u_{lin} may be used as input variable. In both cases, the mass flow through a valve can be modeled as:

$$w = C_v \sqrt{\rho \Delta p}\, u_{lin} \tag{3.1.15}$$

where u_{lin} is the linear choke opening input variable.

It is assumed that the valve characteristics for the production choke valve, gas-lift choke valve and injection valve are known, and that either the characteristics are linear, or linearized by a mapping function as described above. The mass flow through these valves, respectively, are then modeled as:

$$w_p = C_{pc} \sqrt{\rho_p \max\{0, (p_p - p_m)\}}\, u_{pc} \tag{3.1.16}$$

$$w_{gl} = C_{gl} \sqrt{\rho_{gl} \max\{0, (p_{gl} - p_a)\}}\, u_{glc} \tag{3.1.17}$$

$$w_{gi} = C_{iv} \sqrt{\rho_{gi} \max\left\{0, (p_{ai} - p_{ti})\right\}} \tag{3.1.18}$$

where ρ_p, ρ_{gl} and ρ_{gi} are the densities of the fluid mixtures flowing through the respective valves, p_p is the pressure in the tubing at the production choke valve, p_m is the pressure in the flowlines at the manifold, p_{gl} is the pressure of the lift gas from the platform at the manifold, p_a is the pressure in the annulus at the gas lift choke valve, p_{ai} is the pressure in the annulus at the injection valve, and p_{ti} is the pressure in the tubing at the injection valve. u_{pc} is the production choke valve opening, and u_{glc} is the gas-lift choke valve opening. C_{pc}, C_{gl} and C_{iv} are the valve specific constants for the production choke valve, the gas-lift choke valve, and the injection valve, respectively. It is assumed that the flows through the valves are restricted to be positive (one-directional), thus Δp in the valve model given by equation (3.1.15) is replaced by $\max\left\{0, \Delta p\right\}$. It is also assumed that the injection valve is either fully open or fully closed. u in the valve model (3.1.15) is thus replaced by and 1 in the injection valve model (3.1.18).

To simplify the model, a cascaded control structure (see appendix A.6) is assumed, where a flow controller for the gas-lift is implemented, so that the gas-lift choke valve opening u_{glc} may be replaced by the mass flow setpoint u_{gl} as an input variable. The lift gas mass flow model (3.1.17) may then simply be replaced by:

$$w_{gl} = u_{gl} \tag{3.1.19}$$

where u_{gl} is the setpoint for the gas-lift flow controller.

Assuming that the ratio of produced gas an liquid equals the ratio of gas and liquid in the tubing, respectively, we have:

$$\frac{w_{gp}}{w_p} = \frac{m_{gt}}{m_t} \tag{3.1.20}$$

and:

$$\frac{w_{lp}}{w_p} = \frac{m_{lt}}{m_t} \tag{3.1.21}$$

The produced gas and liquid are then given by:

$$w_{gp} = \frac{m_{gt}}{m_t} w_p \tag{3.1.22}$$

and:

$$w_{lp} = \frac{m_{lt}}{m_t} w_p \tag{3.1.23}$$

where w_p is given by equation (3.1.16).

The liquid flow from a reservoir into a well may be modeled using Vogel's equation:

$$w_{lr} = \rho_l Q_{max} \left(1 - (1-C)\left(\frac{p_{bh}}{p_r}\right) - C\left(\frac{p_{bh}}{p_r}\right)^2 \right) \tag{3.1.24}$$

where ρ_l is the density of the liquid flowing from the reservoir, Q_{max} is an empirical constant representing the theoretical absolute open flow (AOF) potential, i.e. the liquid flow rate when the wellbore pressure is zero, C is a constant parameter usually approximated to 0.8, p_{bh} is the bottom hole pressure (BHP) in the wellbore and p_r is the reservoir pressure. Vogel's equation was developed in 1968, and is an empirical equation to model inflow from two-phase reservoirs, and serves the purpose as a reference IPR (inflow performance relationship) curve. It is widely used in the industry, for example in commercial simulators like Pipesim, and is considered more precise than e.g. a PI (productivity index), which is only accurate in single-phase wells. [Vogel 1968, Ahmed 2006]

The water cut (often abbreviated to WC) is defined as:

$$r_{wc} = \frac{w_{wr}}{w_{lr}} = \frac{w_{wp}}{w_{lp}} \tag{3.1.25}$$

where w_{wr} is the inflow of water from the reservoir, and w_{wp} is the mass flow of produced water. The inflow and production of water are then modeled as:

$$w_{wr} = r_{wc} w_{lr} \tag{3.1.26}$$

$$w_{wp} = r_{wc} w_{lp} \tag{3.1.27}$$

and the inflow and production of oil are equivalently modeled as:

$$w_{or} = (1 - r_{wc}) w_{lr} \tag{3.1.28}$$

$$w_{op} = (1 - r_{wc}) w_{lp} \tag{3.1.29}$$

Defining the gas-to-oil ratio (often abbreviated to GOR):

$$r_{gor} = \frac{w_{gr}}{w_{or}} \tag{3.1.30}$$

the inflow of gas from the reservoir is modeled as:

$$w_{gr} = r_{gor} w_{or} \tag{3.1.31}$$

Alternatively, the gas-to-liquid ratio may be defined as:

$$r_{glr} = \frac{w_{gr}}{w_{lr}} = (1 - r_{wc}) r_{gor} \tag{3.1.32}$$

and the inflow of gas from the reservoir may then be modeled as:

$$w_{gr} = r_{glr} w_{lr} \tag{3.1.33}$$

The water cut, gas-to-oil ratio and gas-to-liquid ratio are assumed to be constant or slowly varying, and are thus treated as constants in this model.

3.1.3 Density

Density of produced fluid

The ideal gas law may be written as:

$$pV = nRT \tag{3.1.34}$$

where p is the pressure in a volume V, n is the number of moles of gas in that volume, R is the universal gas constant ($R = 8.3145\ J/(K \cdot mol)$), and T is the temperature in the volume. The number of moles may be derived from the relationship:

$$n = \frac{m}{M} \tag{3.1.35}$$

where M is the molecular weight of the gas, and m is the mass of the gas. Using this, and the general relationship:

$$\rho = \frac{m}{V} \tag{3.1.36}$$

the ideal gas law is written as:

$$p = \rho\frac{RT}{M} \tag{3.1.37}$$

or equivalently:

$$\rho = \frac{M}{RT}p \tag{3.1.38}$$

[Skogestad 2009]. This is used to model the density of the gas in the annulus, which is injected into the tubing through the injection valve:

$$\rho_{gi} = \frac{M_g}{RT_a}p_{ai} \tag{3.1.39}$$

where M_g is the molecular weight of the lift gas in the annulus, T_a is the temperature in the annulus, and p_{ai} is the pressure in the annulus at the injection valve.

The fluid in the tubing is a mixture of gas and liquid. The density at the production choke valve is given by a weighted sum:

$$\rho_p = r_{gp}\rho_{gp} + r_{lp}\rho_l \tag{3.1.40}$$

where

$$r_{gp} = \frac{A_{gp}}{A_t} \tag{3.1.41}$$

is the ratio of cross sectional area occupied by gas relative to the total area of the tubing, and

$$r_{lp} = \frac{A_{lp}}{A_t} \tag{3.1.42}$$

is the ratio of cross sectional area occupied by liquid relative to the total area of the tubing, both at the production choke valve. Using these definitions, the following relationship must be satisfied:

$$r_{gp} + r_{lp} = 1 \tag{3.1.43}$$

The fluid in the tubing is assumed to be perfectly mixed at any time, so that the ratio between produced liquid and gas equals the ratio between liquid and gas in the tubing (as given by equations (3.1.20) and (3.1.21)). According to this, at the production choke valve, the following equation must be satisfied:

$$\frac{r_{lp}\rho_l}{r_{gp}\rho_{gp}} = \frac{m_{lt}}{m_{gt}} = K \tag{3.1.44}$$

$$\Rightarrow r_{lp}\rho_l = K r_{gp}\rho_{gp} \tag{3.1.45}$$

Inserting this into equation (3.1.40), we have:

$$\rho_p = (K+1) r_{gp}\rho_{gp} \tag{3.1.46}$$

The density of the gas is given by the ideal gas law in equation (3.1.38):

$$\rho_{gp} = \frac{M_g}{RT_t} p_p \tag{3.1.47}$$

From this and equation (3.1.45), we have:

$$r_{lp}\rho_l = K r_{gp}\rho_{gp} = K r_{gp}\frac{M_g}{RT_t} p_p \tag{3.1.48}$$

Inserting equation (3.1.43) yields:

$$(1 - r_{gp})\rho_l = K r_{gp}\frac{M_g}{RT_t} p_p \tag{3.1.49}$$

$$\Rightarrow (K M_g p_p + \rho_l R T_t)\, r_{gp} = \rho_l R T_t \tag{3.1.50}$$

$$\Rightarrow r_{gp} = \frac{\rho_l R T_t}{\rho_l R T_t + K M_g p_p} \tag{3.1.51}$$

Equation (3.1.46) then yields:

$$\rho_p = (K+1) r_{gp}\rho_{gp} = (K+1)\frac{\rho_l R T_t}{\rho_l R T_t + K M_g p_p}\frac{M_g}{RT_t} p_p \tag{3.1.52}$$

$$\Rightarrow \rho_p = (K+1)\frac{\rho_l M_g p_p}{\rho_l R T_t + K M_g p_p} \tag{3.1.53}$$

Inserting equation (3.1.44), the density of the produced fluid is thus modeled as:

$$\rho_p = \frac{\rho_l M_g p_p m_t}{\rho_l R T_t m_{lt} + M_g p_p m_{gt}} \tag{3.1.54}$$

Liquid density

Assuming ρ_w is the density of the produced water, and ρ_o is the density of the produced oil, and assuming that the two liquids are perfectly mixed, then the density of the liquid in the tubing is given by:

$$\rho_l = r_{wc}\rho_w + (1 - r_{wc})\rho_o \tag{3.1.55}$$

Note that this is a constant parameter as the water cut is assumed constant, and the liquids are assumed to be incompressible, thus the densities are also constant.

3.1.4 Pressure

Annulus

Both the pressure and the density in the annulus will vary with height, according to the ideal gas law in equation (3.1.37) and the hydrostatic pressure formula:

$$dp = -\rho g dh \tag{3.1.56}$$

where g is the acceleration of gravity ($g = 9.81$ $[m/s^2]$), and a positive dh means a height difference in the opposite direction of the earth's gravity (upwards). A model for the pressure in the annulus based on this is derived in appendix C. However, the model used in this project is simplified by approximating the density at the mass center of the annulus as the average density in the annulus. We then have:

$$p_0 = \bar{\rho}_a \frac{RT_a}{M_g} = \frac{m_{ga}}{V_a}\frac{RT_a}{M_g} \tag{3.1.57}$$

where $\bar{\rho}_a$ is the average density in the annulus, and p_0 is the pressure at the mass center of the annulus. Then the pressures p_a and p_{ai} are given by subtracting or adding the pressure from the weight of half the gas in the annulus:

$$p_a = p_0 - \frac{m_{ga}g}{2A_a} \tag{3.1.58}$$

$$p_{ai} = p_0 + \frac{m_{ga}g}{2A_a} \tag{3.1.59}$$

$$\Downarrow$$

$$p_a = \left(\frac{RT_a}{V_a M_g} - \frac{g}{2A_a}\right) m_{ga} \tag{3.1.60}$$

$$p_{ai} = \left(\frac{RT_a}{V_a M_g} + \frac{g}{2A_a}\right) m_{ga} \tag{3.1.61}$$

Tubing

In the tubing, both the pressure, density and distribution between gas and liquid will vary with depth. By doing some approximations, the density and relative distribution of gas and liquid in the mass center of the tubing are assumed to equal the average density and distribution in the tubing, so that the following equation will be satisfied:

$$\bar{\rho}_t A_t = \rho_l \bar{A}_l + \bar{\rho}_g \bar{A}_g \tag{3.1.62}$$

Here, $\bar{\rho}_t$ is the average density in the tubing, $\bar{\rho}_g$ is the average density of gas in the tubing, A_t is the cross sectional area of the tubing, $\bar{A}_l$ is the average cross sectional area occupied by liquid in the tubing, and $\bar{A}_g$ is the average cross sectional area occupied by gas. The average cross sectional area of the tubing occupied by liquid is found through the relationship $m = \rho V$:

$$m_{lt} = \rho_l V_{lt} = \rho_l \bar{A}_l L_t \Rightarrow \bar{A}_l = \frac{m_{lt}}{\rho_l L_t} = \frac{\nu_l m_{lt}}{L_t} \tag{3.1.63}$$

where $\nu = 1/\rho$. Through the relationship:

$$\bar{A}_l + \bar{A}_g = A_t \tag{3.1.64}$$

the average area of gas in the tubing is found:

$$\bar{A}_g = A_t - \bar{A}_l = A_t - \frac{\nu_l m_{lt}}{L_t} \tag{3.1.65}$$

Inserting this yields:

$$\bar{\rho}_t A_t = \rho_l \frac{m_{lt}}{\rho_l L_t} + \bar{\rho}_g \left(A_t - \frac{\nu_l m_{lt}}{L_t}\right) \tag{3.1.66}$$

Inserting

$$\bar{\rho}_t = \frac{m_t}{V_t} \tag{3.1.67}$$

yields:

$$\frac{m_t}{V_t}A_t = \frac{m_{lt}}{L_t} + \bar{\rho}_g\left(A_t - \frac{\nu_l m_{lt}}{L_t}\right) \tag{3.1.68}$$

$$\Rightarrow m_t - m_{lt} = \bar{\rho}_g\left(V_t - \nu_l m_{lt}\right) \tag{3.1.69}$$

$$\Rightarrow \bar{\rho}_g = \frac{m_{gt}}{V_t - \nu_l m_{lt}} \tag{3.1.70}$$

The pressure at the mass center is then found using the ideal gas law from equation (3.1.37):

$$p_0 = \bar{\rho}_g \frac{RT_t}{M_g} \tag{3.1.71}$$

$$\Rightarrow p_0 = \frac{RT_t m_{gt}}{M_g V_t - M_g \nu_l m_{lt}} \tag{3.1.72}$$

where T_t is the temperature in the tubing. The pressures at the production choke valve and the injection valve are then approximated by adding and subtracting the pressure from the weight of half of the fluids in the tubing:

$$p_p = p_0 - \frac{gm_t}{2A_t} \tag{3.1.73}$$

$$p_{ti} = p_0 + \frac{gm_t}{2A_t} \tag{3.1.74}$$

$$\Rightarrow p_{ti} = p_p + \frac{gm_t}{A_t} \tag{3.1.75}$$

The pressures are thus modeled as:

$$p_p = \frac{RT_t m_{gt}}{M_g V_t - M_g \nu_l m_{lt}} - \frac{gm_t}{2A_t} \tag{3.1.76}$$

$$p_{ti} = \frac{RT_t m_{gt}}{M_g V_t - M_g \nu_l m_{lt}} + \frac{gm_t}{2A_t} \tag{3.1.77}$$

Bottom hole pressure

The bottom hole pressure in the wellbore (BHP) is modeled by adding the hydrostatic pressure from the fluids in the tubing below the point of injection to the pressure at the injection point:

$$p_{bh} = p_{ti} + \rho_f g L_w \tag{3.1.78}$$

where ρ_f is the average density of the fluids in the wellbore, and L_w is the vertical length of the tubing from the point of injection to the reservoir.

The average density of fluids in the well bore below the point of gas injection is modeled as:

$$\rho_f = \frac{\rho_l + r_{glr}\bar{\rho}_{gw}}{1 + r_{glr}} \tag{3.1.79}$$

where $\bar{\rho}_{gw}$ is the average density of the gas in the wellbore, modeled using the ideal gas law and an approximation of the average pressure in the wellbore:

$$\bar{\rho}_{gw} = \frac{M_g}{RT_t}\frac{p_{ti} + p_{bh}}{2} \tag{3.1.80}$$

From this, we have:

$$p_{bh} = p_{ti} + \frac{\rho_l + r_{glr}\frac{M_g}{RT_t}\frac{p_{ti}+p_{bh}}{2}}{1 + r_{glr}} gL_w \tag{3.1.81}$$

$$\Rightarrow (1 + r_{glr})p_{bh} = (1 + r_{glr})p_{ti} + \rho_l g L_w + \frac{r_{glr} M_g g L_w}{2RT_t} p_{ti} + \frac{r_{glr} M_g g L_w}{2RT_t} p_{bh} \tag{3.1.82}$$

$$\Rightarrow \left(1 + r_{glr} - \frac{r_{glr} M_g g L_w}{2RT_t}\right) p_{bh} = \left(1 + r_{glr} + \frac{r_{glr} M_g g L_w}{2RT_t}\right) p_{ti} + \rho_l g L_w \tag{3.1.83}$$

The bottom hole pressure is thus modeled as:

$$p_{bh} = \frac{\left(1 + r_{glr} + \frac{r_{glr} M_g g L_w}{2RT_t}\right) p_{ti} + \rho_l g L_w}{1 + r_{glr} - \frac{r_{glr} M_g g L_w}{2RT_t}} \tag{3.1.84}$$

3.2 Simplifications and assumptions

To derive this model, naturally many assumptions and simplifications have been made. Especially two significant simplifications should be noted [Binder 2011b]:

- Friction is not considered in this model at all. Pressure drop due to friction must be considered to obtain a more precise model of the fluid flow and pressures in the tubing. This is probably significant at high flow rates.
- It is assumed that the masses of the three fluids in the tubing is distributed without any transport delay. Multiphase flow, with the resulting distribution of oil, water and gas, is not considered. In general, the behavior of a system with multiphase flowlines depends on the flow regime (or flow pattern), which makes the mathematical modeling very complicated. Since it is not considered in this project, the reader is referred to other literature on the subject, such as [Arubi 2011] or [Valle 1998].

Other assumptions and simplifications are described in the model derivation in section 3.1.

According to [Eikrem 2008], the 'Eikrem model' catches the essential dynamics of gas-lift wells (the casing heading instability in particular), in spite of the complexity of multiphase flow, and the simplifications made to derive the model. The model used in this project is based on the Eikrem model (with some changes, see section 3.3), and shows very similar dynamic behavior (see section 3.5).

3.3 Modifications

Although the model used in this project work is based on the 'Eikrem model' presented in [Eikrem 2008], some changes have been made. This section highlights these changes. However, much of the reasoning behind these modifications are found in section 3.1.

Oil, water and liquid The Eikrem model considers only a two-phase system, producing oil and gas. The model used in this project includes water. The state variable representing mass of oil in the tubing in the Eikrem model is replaced by mass of liquid in the tubing, and the oil flow rates in the Eikrem model are replaced by liquid flow rates in this model. Also, a gas-to-liquid ratio is defined, replacing the gas-to-oil ratio in the Eikrem model:

$$r_{glr} = (1 - r_{wc})r_{gor} \tag{3.3.1}$$

and a liquid density is introduced using the water cut parameter r_{wc}:

$$\rho_l = r_{wc}\rho_w + (1 - r_{wc})\rho_o \tag{3.3.2}$$

The mass flow of produced water and oil, respectively, is also modeled using the water cut parameter:

$$w_{wp} = r_{wc} w_{lp} \tag{3.3.3}$$

$$w_{op} = (1 - r_{wc}) w_{lp} \tag{3.3.4}$$

Lift gas In the Eikrem model, the lift gas is modeled as a constant:

$$w_{gl} = Constant \tag{3.3.5}$$

while in this project, the lift gas is considered a controlled variable, and is used as an input variable to the system:

$$w_{gl} = u_{gl} \tag{3.3.6}$$

Valve models The model for the injection valve is the same, but the Eikrem model includes an (undefined) valve function in the model for the production choke:

$$w_p = C_{pc} \sqrt{\rho_p \max\{0, (p_p - p_m)\}}\, f_{pc}(u) \tag{3.3.7}$$

In this project, this valve function is replaced by the input variable representing the choke opening:

$$f_{pc}(u) = u_{pc} \tag{3.3.8}$$

and the flow through the valve is modeled as:

$$w_p = C_{pc} \sqrt{\rho_p \max\{0, (p_p - p_m)\}}\, u_{pc} \tag{3.3.9}$$

Inflow model In the Eikrem model, the inflow of oil from the reservoir is modeled as an (unspecified) function of the pressure difference between the reservoir and the bottom-hole pressure in the wellbore:

$$w_{or} = f_r \left(p_r - p_{bh}\right) \tag{3.3.10}$$

In this project, this is replaced by the inflow of liquid from the reservoir, and is modeled using Vogel's equation:

$$w_{lr} = \rho_l Q_{max} \left(1 - (1 - C)\left(\frac{p_{bh}}{p_r}\right) - C\left(\frac{p_{bh}}{p_r}\right)^2\right) \tag{3.3.11}$$

The inflow of gas is modeled using the gas-to-liquid ratio r_{glr} instead of the gas-to-oil ratio r_{gor} which is used in the Eikrem model, so that the model becomes:

$$w_{gr} = r_{glr} w_{lr} \tag{3.3.12}$$

where

$$r_{glr} = (1 - r_{wc}) r_{gor} \tag{3.3.13}$$

Densities As in the Eikrem model, the density of the gas in the annulus is modeled using the ideal gas law. However, in the Eikrem model, the density in the tubing at the production choke valve is modeled as the average density in the tubing:

$$\rho_p = \frac{m_t}{V_t} \tag{3.3.14}$$

Because the density in the tubing will vary with height due to the pressure difference, this is a very coarse approximation. In this project, the density is modeled using the ideal gas law, hydrostatic pressure gradients, and the assumption that the ratio between produced gas and liquid equals the ratio between the gas and liquid in the tubing (see section 3.1.3). This results in the following model:

$$\rho_p = \frac{\rho_l M_g p_p m_t}{\rho_l R T_t m_{lt} + M_g p_p m_{gt}} \tag{3.3.15}$$

Pressures In the Eikrem model, the pressure in the annulus at the gas-lift choke valve seems to be modeled using the ideal gas law with the total mass of gas in the annulus and the total annulus volume:

$$p_a = \frac{m_{ga} R T_a}{M_g V_a} \tag{3.3.16}$$

The pressure at the point of injection is then modeled by adding the weight of the gas:

$$p_{ai} = p_a + \frac{m_{ga} g}{A_a} \tag{3.3.17}$$

so that the model becomes:

$$p_{ai} = \left(\frac{RT_a}{M_g V_a} + \frac{g}{A_a} \right) m_{ga} \tag{3.3.18}$$

However, the ideal gas law will in fact provide an estimate of the average pressure in the annulus, not the pressure in the top of the annulus. In this project, this pressure is therefore modeled by adding only half the weight of the gas to the pressure found using the ideal gas law:

$$p_{ai} = \left(\frac{RT_a}{V_a M_g} + \frac{g}{2A_a} \right) m_{ga} \tag{3.3.19}$$

See section 3.1.4 for a detailed derivation of this model.

In the Eikrem model, the pressure in the tubing at the production choke valve is modeled using the ideal gas law and the total volume of gas in the tubing, modeled as the volume of the tubing not occupied by oil. With the same argumentation as with the annulus model, this is a very coarse approximation. In this project, this pressure is modeled in a similar way, but by subtracting half the weight of the mass in the tubing from the pressure found using the ideal gas law in this way:

$$p_p = \frac{RT_t m_{gt}}{M_g V_t - M_g \nu_l m_{lt}} - \frac{g m_t}{2A_t} \tag{3.3.20}$$

The pressure in the tubing at the injection valve is found relative to the the pressure at the choke valve in the same way as in the Eikrem model, by adding the weight of the mass in the tubing:

$$p_{ti} = p_p + \frac{g m_t}{A_t} \tag{3.3.21}$$

Even though this equation remains unchanged, this pressure will be different because of the change in the pressure p_p.

In the Eikrem model, the bottom-hole pressure in the wellbore is modeled by adding the weight of oil in the well below the point of injection:

$$p_{bh} = p_{ti} + \rho_o g L_w \tag{3.3.22}$$

The inflow from the reservoir is thus assumed to consist only of oil. However, in the model used in this project, it is assumed a three-phase flow from the reservoir, and the BHP is thus modeled using

$$p_{bh} = p_{ti} + \rho_f g L_w \tag{3.3.23}$$

where ρ_f is the average density of the fluids in the tubing below the point of gas injection. The model then becomes:

$$p_{bh} = \frac{\left(1 + r_{glr} + \frac{r_{glr} M_g g L_w}{2RT_t}\right) p_{ti} + \rho_l g L_w}{1 + r_{glr} - \frac{r_{glr} M_g g L_w}{2RT_t}} \tag{3.3.24}$$

See section 3.1.4 for a detailed derivation of this model.

3.4 Summary of the well model

A summary of the well model is given in this section. The MATLAB implementation of this model is found in appendix D.1.

State models (masses):

$$\dot{m}_{ga} = w_{gl} - w_{gi} \tag{3.4.1}$$
$$\dot{m}_{gt} = w_{gr} + w_{gi} - w_{gp} \tag{3.4.2}$$
$$\dot{m}_{lt} = w_{lr} - w_{lp} \tag{3.4.3}$$

Mass flows:

$$w_{gl} = u_{gl} \tag{3.4.4}$$
$$w_{gi} = C_{iv}\sqrt{\rho_{gi}\max\{0,(p_{ai}-p_{ti})\}} \tag{3.4.5}$$
$$w_p = C_{pc}\sqrt{\rho_p \max\{0,(p_p-p_m)\}}\, u_{pc} \tag{3.4.6}$$
$$w_{gp} = \frac{m_{gt}}{m_t} w_p \tag{3.4.7}$$
$$w_{lp} = \frac{m_{lt}}{m_t} w_p \tag{3.4.8}$$
$$w_{wp} = r_{wc} w_{lp} \tag{3.4.9}$$
$$w_{op} = (1-r_{wc}) w_{lp} \tag{3.4.10}$$
$$w_{lr} = \rho_l Q_{max}\left(1-(1-C)\left(\frac{p_{bh}}{p_r}\right)-C\left(\frac{p_{bh}}{p_r}\right)^2\right) \tag{3.4.11}$$
$$w_{gr} = r_{glr} w_{lr} \tag{3.4.12}$$
$$r_{glr} = (1-r_{wc}) r_{gor} \tag{3.4.13}$$

Densities:

$$\rho_{gi} = \frac{M_g}{RT_a} p_{ai} \tag{3.4.14}$$
$$\rho_p = \frac{\rho_l M_g p_p m_t}{\rho_l R T_t m_{lt} + M_g p_p m_{gt}} \tag{3.4.15}$$
$$\rho_l = r_{wc}\rho_w + (1-r_{wc})\rho_o \tag{3.4.16}$$

Pressures:

$$p_{ai} = \left(\frac{RT_a}{V_a M_g} + \frac{g}{2A_a}\right) m_{ga} \tag{3.4.17}$$
$$p_p = \frac{RT_t m_{gt}}{M_g V_t - M_g \nu_l m_{lt}} - \frac{g m_t}{2A_t} \tag{3.4.18}$$
$$p_{ti} = p_p + \frac{g m_t}{A_t} \tag{3.4.19}$$
$$p_{bh} = \frac{\left(1 + r_{glr} + \frac{r_{glr} M_g g L_w}{2RT_t}\right) p_{ti} + \rho_l g L_w}{1 + r_{glr} - \frac{r_{glr} M_g g L_w}{2RT_t}} \tag{3.4.20}$$

3.5 Model simulations

Simulations of the model throughout this report are performed using the first order numerical integration scheme known as Euler's method. This is a simple integration scheme, where the numerical integration of

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{3.5.1}$$

is calculated using:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k)\Delta t \tag{3.5.2}$$

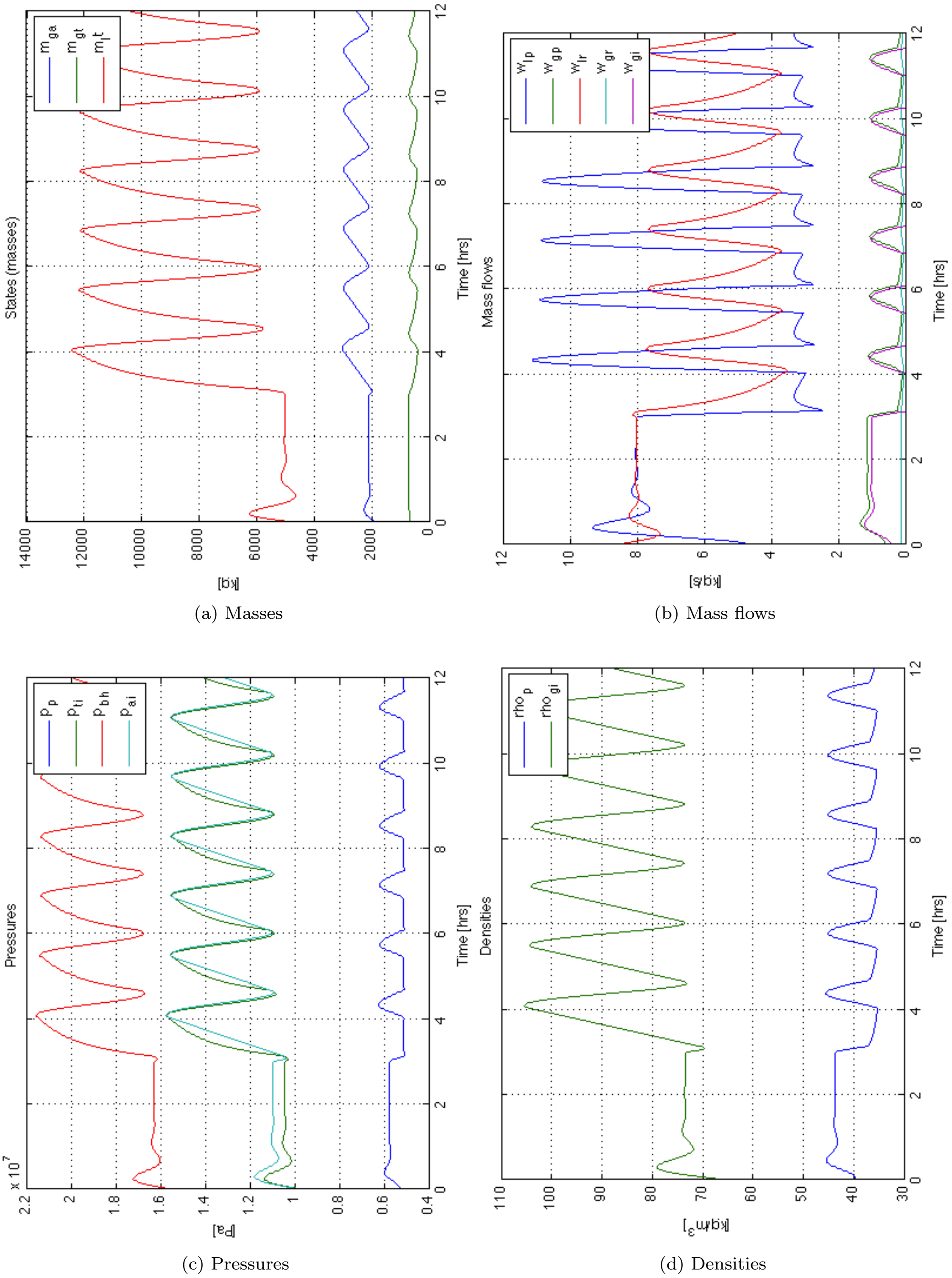
where Δt denotes the time step. Euler's method provides a more accurate integration for smaller Δt, and less accurate for larger Δt. [Egeland 2002]

In this report, Δt is set to 0.5 seconds for simulations, which is very small compared to the system's dynamics. Euler's method with this small time steps is assumed to be sufficiently accurate, where the numerical errors are insignificantly small compared to model errors.

Other integration methods were also tried, like the built-in ode45 method in MATLAB, which is based on the explicit Runge-Kutta (4,5) method (see [Egeland 2002]). This was a lot slower, without gaining much accuracy, and even some numerical problems occured, which was not seen with Euler's method.

3.5.1 Test simulations

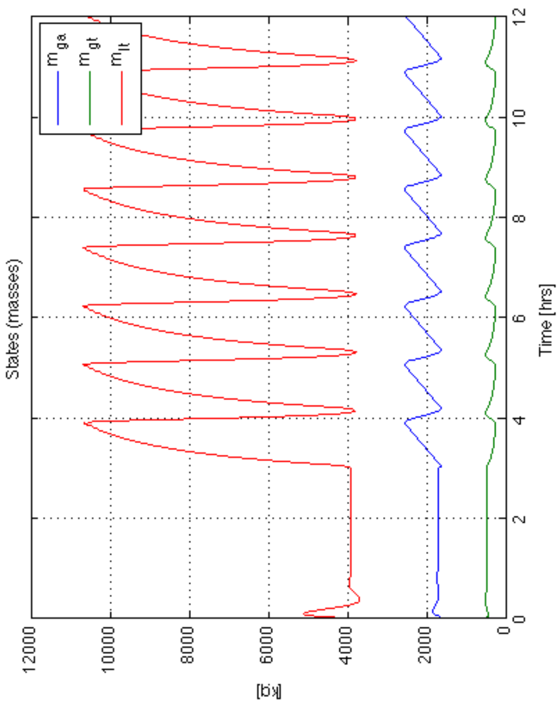
Figure 3.1 on the next page shows a test simulation of the model with the parameters of well 6. The simulation starts close to (but not at) steady-state with $u_{pc} = 1$ and $u_{gl} = 1\,kg/s$, a stable operating point. After 3 hours, the gas-lift is changed to $u_{gl} = 0.3\,kg/s$. The well is then clearly at an unstable operating point (see section 4.1.2.1), and large oscillations are observed. For comparison, the same simulation is performed with the 'Eikrem model', shown in figure 3.2 on page 35.



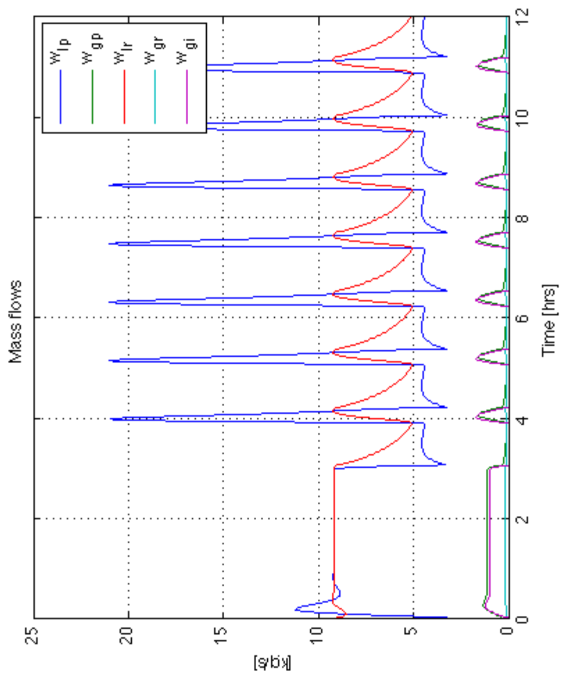
(a) Masses (b) Mass flows

(c) Pressures (d) Densities

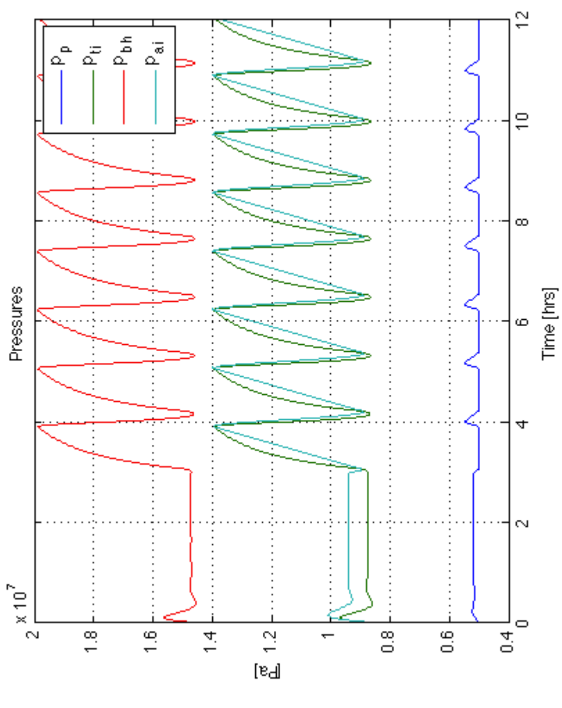
Figure 3.1: Test simulation



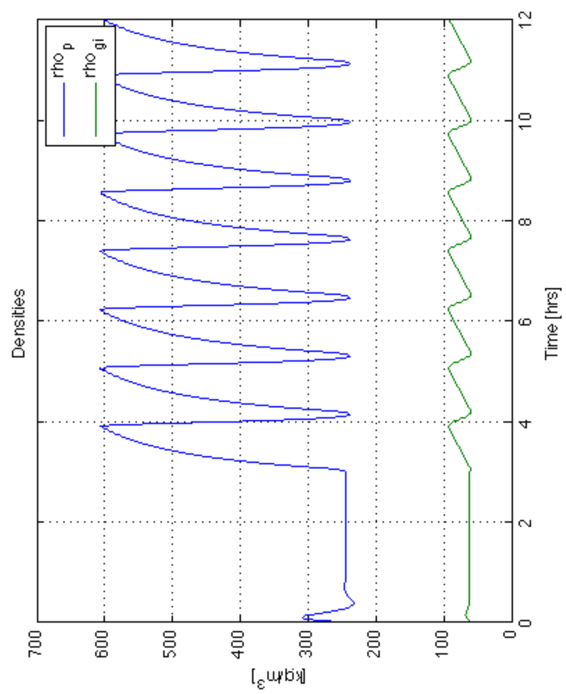
(a) Masses



(b) Mass flows



(c) Pressures



(d) Densities

Figure 3.2: Test simulation Eikrem model

3.6 Manifold model

As described in section 1.2, the seven wells are producing to one manifold. There are two flowlines taking the produced fluids from the manifold to the surface facility (the FPSO). In the manifold, the production from each well is routed to either one of these flowlines using on/off valves. The pressure in the flowlines at the manifold is considered a boundary condition to the system, and is assumed constant. The flow in each flowline is merely the sum of the flows from each well routed to that flowline.

The manifold pressure used in this project is given by the parameter p_m in table F.1 in the appendix.

3.7 Static model

A static model of the wells is developed, based on the steady-state of the dynamic model presented in section 3.4, to find a model on the form

$$m_{ss} = f(u_{pc}, u_{gl}) \tag{3.7.1}$$

where m_{ss} is the masses (state variables) at steady-state. The well model is at steady-state if $\dot{m}_{ga}$, $\dot{m}_{gt}$ and $\dot{m}_{lt}$ all equal zero. As the model depends on the inputs, the steady-state also varies with the inputs. Once the steady-state masses is known, the well model may be used to find all the other relevant information, such as production rates, pressures, etc.

Due to the complexity and non-linearity of the model, finding the steady-state is a non-trivial task. In this project, the steady-state is found numerically. At steady-state, the mass flow of gas into the annulus must equal the mass flow of gas into the tubing, $w_{gi} = w_{gl}$. Using this, the steady-state is found in two steps, which increases the speed of calculation significantly:

1. The steady-state masses in the tubing is found with the assumption $w_{gi} = w_{gl} = u_{gl}$
2. The steady-state mass in the annulus is found with the tubing masses already known

The states in the tubing is found using the `fsolve` function from the MATLAB optimization toolbox, which is used to solve problems on the form $f(x) = 0$, where x is a vector.

The procedure to find the steady-state mass of gas in the annulus is based on the observation that the injected gas flowing from the annulus into the tubing will increase with pressure in the annulus, which again increases with mass of gas in the annulus. The steady-state mass of gas in the annulus $m_{ga,ss}$ is characterized by $\dot{m}_{ga}(m_{ga,ss}) = u_{gl} - w_{gi}(m_{ga,ss}) = 0$. ($\dot{m}$ is a function of m, see the model in section 3.4.) With this in mind, the following procedure is applied:

1. Based on knowledge of the system, or trial and error, find two estimates of m_{ga}, a_0 and a_2, such that $\dot{m}_{ga}(a_0) > 0$ and $\dot{m}_{ga}(a_2) < 0$. This implies that $a_0 < m_{ga,ss} < a_2$
2. Use the average $a_1 = \frac{1}{2}(a_0 + a_2)$ as an estimate of $m_{ga,ss}$
3. If $\dot{m}_{ga}(a_1) > 0$, the estimate is too low, set $a_0 = a_1$
4. Else, $\dot{m}_{ga}(a_1) < 0$, and the estimate is too high, set $a_2 = a_1$

5. While $a_2 - a_0 > tol$ (a predefined tolerance), repeat from step 2.

This search procedure is based on the bisection method, described on pp. 552-555 in [Hillier 2010], and proved to be a lot faster than `fsolve` or a simulation-based approach, which were also tried.

This is implemented as a MATLAB function, providing the steady-state masses as a function of the inputs u_{pc} and u_{gl} and the well parameters. The main content of this function is shown in appendix D.2.

3.7.1 Precalculation of static model

Because the static model is used in optimization, its calculation speed is of great importance. The method described above is relatively slow, as it itself contains numerical searches. To increase the calculation speed, instead of using the above function directly, the function is used to generate precalculated data. An example of such precalculated data is shown in figure 3.3, where mass of gas in the tubing of well 1 is calculated for data points with u_{pc} in the range 0 to 1, and u_{gl} in the range 0 to 6. (A much higher resolution was used in the actual optimization.)

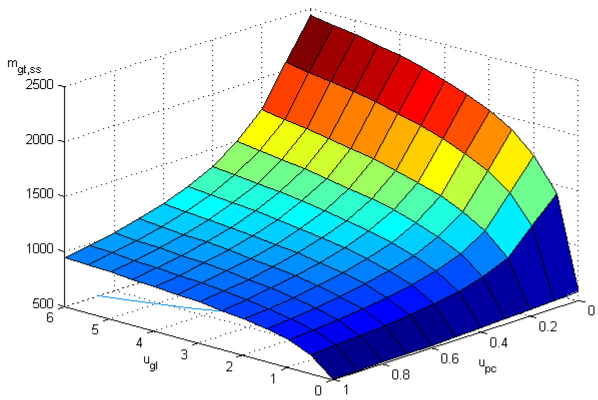


Figure 3.3: Precalculated data ($m_{gt,ss}$, well 1)

In the optimization, linear interpolation of the precalculated data is used to estimate the steady-state solutions for each set of inputs. With a high resolution of the precalculated data, this is quite accurate, and a lot faster. Precalculation increased the speed of optimization by a factor of 10-100.

Chapter 4

Methods

In this chapter, the methods developed to optimize the oil production are presented.

4.1 Static optimization

The problem to be solved using static optimization is to find the routing and inputs for the seven wells (gas-lift rates and production choke openings), that yield the highest possible oil production rate, on a long time horizon (i.e. a time horizon where all the wells settle to steady-state, and where dynamics in the system are negligible). The system system's processing capacities and other limitations also must be respected. To solve this problem, numerical optimization based on the static model described in section 3.7 is implemented. The formulation and solution of this problem is discussed in this section.

4.1.1 Objective function

The objective function for the static optimization problem is formulated as

$$\max_{u_{pc,i},u_{gl,i},u_{r,i}} \quad \sum_{i=1}^{7} w_{op,i}(u_{pc,i}, u_{gl,i}) \tag{4.1.1}$$

The produced oil is given by the static model described in section 3.7, and is a function of the production choke opening and gas-lift rate for each well. These are the decision variables in the optimization, as well as a routing decision variable, $u_{r,i}$ as described in section 4.1.2.

4.1.2 Constraints

There are several constraints for the static optimization problem. Obviously, due to physical considerations, the input variables are constrained by:

$$0 < u_{pc,i} < 1 \quad \forall i \in \{1..7\} \tag{4.1.2}$$

$$u_{gl,i} > 0 \quad \forall i \in \{1..7\} \tag{4.1.3}$$

$$u_{r,i} \in \{0, 1\} \quad \forall i \in \{1..7\} \tag{4.1.4}$$

The downstream processing equipment is assumed to have a limited capacity for liquid separation, water treatment and gas compression. This is introduced in the considered system by imposing constraints on the gas, water and liquid production for each flowline. Indexing the flowlines with 0 and 1, these constraints can be formulated as follows:

$$\sum_{i=1}^{7} w_{gp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq G_0 \tag{4.1.5}$$

$$\sum_{i=1}^{7} w_{gp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq G_1 \tag{4.1.6}$$

$$\sum_{i=1}^{7} w_{wp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq W_0 \tag{4.1.7}$$

$$\sum_{i=1}^{7} w_{wp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq W_1 \tag{4.1.8}$$

$$\sum_{i=1}^{7} w_{lp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq L_0 \tag{4.1.9}$$

$$\sum_{i=1}^{7} w_{lp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq L_1 \tag{4.1.10}$$

where G_j, W_j and L_j are the gas, water and liquid production constraints, respectively, for flowline j, and $u_{r,i}$ is a binary routing decision variable for each well. $u_{r,i} \in \{0, 1\}$, where $u_{r,i} = 0$ means that well i is routed to the flowline with index $j = 0$, and $u_{r,i} = 1$ means that well i is routed to the flowline with index $j = 1$. Note that the production rates are functions of the production choke valve and gas-lift input variables, given by the well model from section 3.4.

It is also assumed that there is a limited amount of gas available for gas-lift in the system, so that the constraint:

$$\sum_{i=1}^{7} u_{gl,i} \leq GL \tag{4.1.11}$$

is imposed on the system.

The values for the system constraints used in this report are given in table F.3 in the appendix.

4.1.2.1 Stability regions

Casing heading instability, as described in section 1.1.4, is not wanted when producing from the wells, as this both causes problems for downstream equipment, and causes a production loss. Casing heading occurs with low gas-lift rates and high production choke openings, and is also dependant on well parameters. (Simulations of the model show that the instability increases with higher water cut, lower Q_{max} or lower gas-to-oil ratio.) To find an estimate of the stable and unstable regions, the wells are simulated from close to steady-state, with varying production choke openings and gas-lift rates, while detecting whether the variations in the production levels increase or decrease with time. The gas-lift rate which is at the limit of stability for a given production choke opening is found using a search based on the bisection method (see [Hillier 2010]). This is implemented in a MATLAB script presented in appendix D.3.

After an estimate of the stabilty regions for the seven wells were found by this script, additional 'manual' simulations were performed to find sufficiently stable operating points for each well. Based on these stable operating points, linear stability region limits to be used in the optimization were defined. This is shown in figure 4.1.

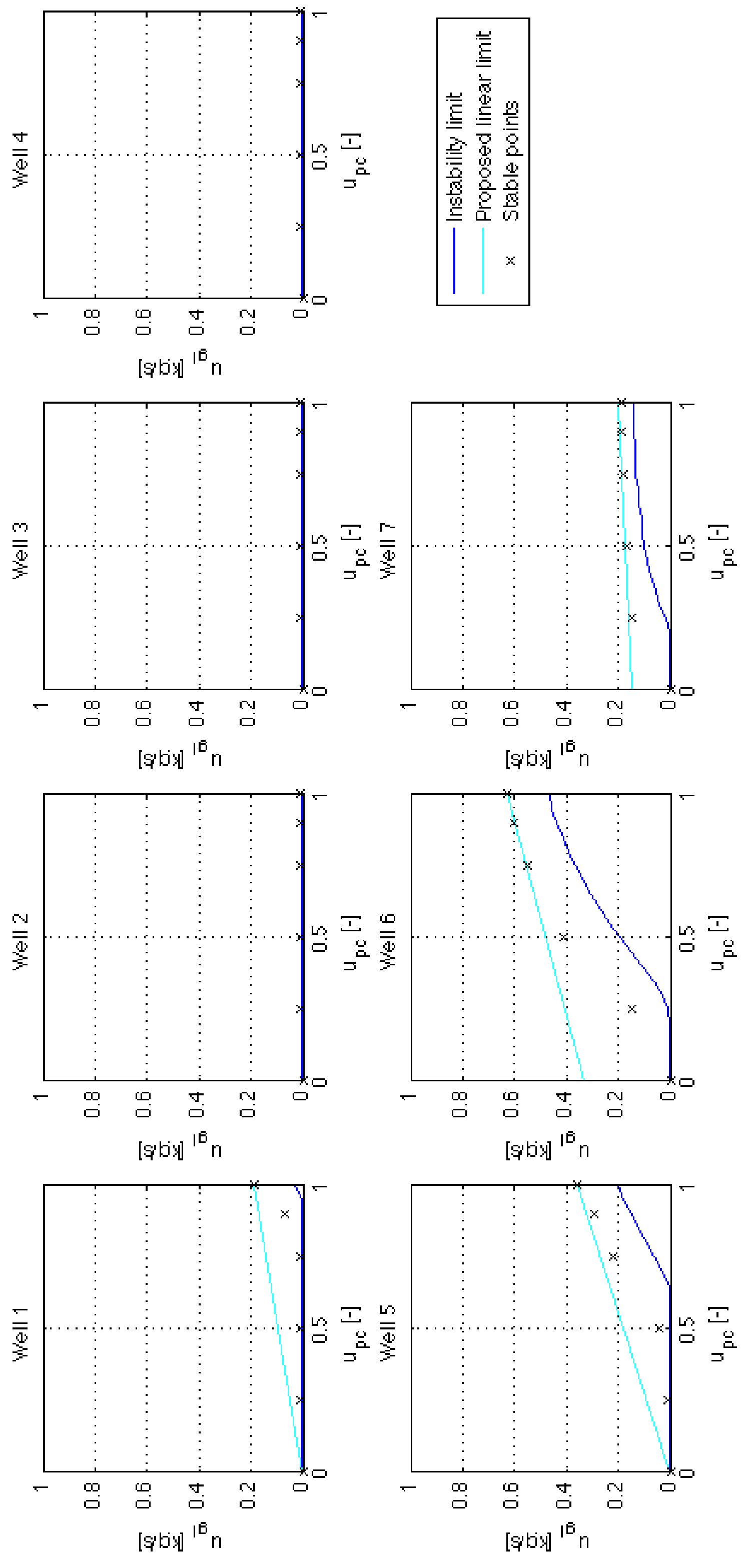


Figure 4.1: Stability regions with proposed linear limits

Note that with $u_{gl} = 0$, the well is also stable (as there will be no casing heading without any gas-lift). Based on this, the following constraints are added to the optimization problem:

$$\left.\begin{array}{c} u_{gl,i} \geq a_i u_{pc,i} + b_i \\ or \\ u_{gl,i} = 0 \end{array}\right\} \qquad \forall i \in \{1..7\} \tag{4.1.12}$$

The values used for a_i and b_i are given in table 4.1 on the next page. Casing heading does not occur for any inputs in wells 2, 3 or 4. The above constraint is thus reduced to $u_{gl} \geq 0$ for these wells, with $a_i = b_i = 0$.

4.1.3 Problem summary

The static optimization problem to find the optimal routing and input settings for the wells may be summarized as follows:

$$\max_{u_{pc,i}, u_{gl,i}, u_{r,i}} \quad \sum_{i=1}^{7} w_{op,i}(u_{pc,i}, u_{gl,i}) \tag{4.1.13}$$

subject to:

$$0 < u_{pc,i} < 1, \quad i \in \{1..7\} \tag{4.1.14}$$

$$u_{gl,i} > 0, \quad i \in \{1..7\} \tag{4.1.15}$$

$$u_{r,i} \in \{0, 1\}, \quad i \in \{1..7\} \tag{4.1.16}$$

$$\sum_{i=1}^{7} w_{gp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq G_0 \tag{4.1.17}$$

$$\sum_{i=1}^{7} w_{gp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq G_1 \tag{4.1.18}$$

$$\sum_{i=1}^{7} w_{wp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq W_0 \tag{4.1.19}$$

$$\sum_{i=1}^{7} w_{wp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq W_1 \tag{4.1.20}$$

$$\sum_{i=1}^{7} w_{lp,i}(u_{pc,i}, u_{gl,i}) \cdot (1 - u_{r,i}) \leq L_0 \tag{4.1.21}$$

$$\sum_{i=1}^{7} w_{lp,i}(u_{pc,i}, u_{gl,i}) \cdot u_{r,i} \leq L_1 \tag{4.1.22}$$

$$\sum_{i=1}^{7} u_{gl,i} \leq GL \tag{4.1.23}$$

$$\left.\begin{array}{c} u_{gl,i} \geq a_i u_{pc,i} + b_i \\ or \\ u_{gl,i} = 0 \end{array}\right\} \qquad \forall i \in \{1..7\} \tag{4.1.24}$$

The production ($w_{gp,i}$, $w_{lp,i}$ and $w_{wp,i}$) as functions of the input variables have a shape as shown in figure 4.2, and thus are concave in the region of interest. Because of this, the production constraints give a non-convex feasible set with respect to the input variables, and the resulting problem is non-convex. The problem also contains binary decisions; routing and stability region constraints. Thus, this is a non-convex MINLP problem.

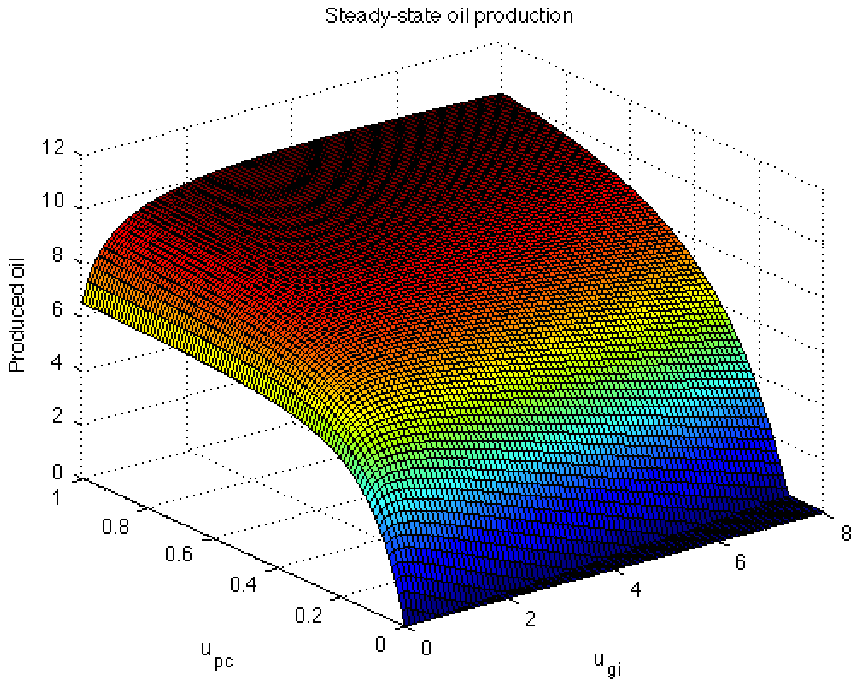


Figure 4.2: Oil production as a function of well inputs [Binder 2011b]

Well (i)	1	2	3	4	5	6	7
a_i	0.19	0	0	0	0.36	0.30	0.05
b_i	0	0	0	0	0	0.33	0.15

Table 4.1: Values for linear constraints

4.1.4 Solution

The solution of the resulting optimization problem is found by combining a 'brute force' approach to find the optimal routing, an implementation of a Branch-and-Bound solution approach to ensure stability of the wells, and using the SQP algorithm in the MATLAB function `fmincon` to solve the resulting NLP problem in each node/branch. (Branch and Bound is explained in appendix B.7.)

As mentioned, the routing is decided using a 'brute force' approach, i.e. all the possible combinations are tried. This involves solving the optimization problem with the input variables $u_{pc,i}$ and $u_{gl,i}$ as decision variables, for 2^n different sets of routing variables $u_{r,i}$, where n is the number of wells. However, there is symmetry in the routing options, as the two flowlines have the exact same properties. Therefore, only 2^{n-1} routing options are investigated for n wells, instead of 2^n.

Because this is a non-convex optimization problem, different starting points may result in different optimal solutions. Therefore, different starting points should be implemented to find the best solution, but this increases the computational time significantly. In this report, it was decided to use three different starting points in the optimization for each routing. These starting points are defined by:

1. $u_{gl} = 0$ and $u_{pc} = 0.1$ for all the wells.
2. $u_{gl} = 0$ and $u_{pc} = 0.5$ for all the wells.
3. $u_{pc} = 1$ and $u_{gl} = GL/n$ for all n wells, where GL is the available lift-gas.

Different and more varying starting points were also tested, but while this increased the computational time severely, the optimal solution was not significantly increased in the scenarios tested, compared to the three starting points described above.

The stability constraints for the wells, given by (4.1.24), is ensured using a Branch & Bound approach. The problem is first solved without this constraint. If one well violates this constraint, the constraints for this well is added in two different branches, one with $u_{gl,i} \geq a_i u_{pc,i} + b_i$ for that well, and the other with $u_{gl,i} = 0$ for that well. (If more than one well violates the constraints, the branching is performed on the well furthest away from the stable region.)

The solution approach is implemented in MATLAB using a recursive function, and may be outlined as follows:

1. For each possible routing
   (a) For each starting point
      i. Find a solution to the optimization problem without imposing the linear stability region constraints
      ii. If the solution of the unconstrained problem is not better than a previously found feasible solution (given by a bound), return without any feasible solution found
      iii. Else, if one or more of the stability region constraints are violated
         A. Find the well number with the greatest violation
         B. Branch on the stability constraints for this well. (Solve the optimization problem again adding the constraint $u_{gl,i} = 0$ in one branch, and $u_{gl,i} \geq a_i u_{pc,i} + b_i$ in the other branch. (The branching is done in a recursive function, which starts in 1.(a).i.)
         C. Compare the solutions from each branch, return the best solution
      iv. Else, return the solution as a possible optimal solution to the problem
   (b) Compare the solutions for each starting point, and keep the best solution
2. Compare the solutions for each routing, and keep the best solution

The MATLAB implementation for the solution approach is found in appendix D.5.

4.2 Dynamic optimization

The static optimization described in section 4.1 finds the routing and input settings that are optimal on a long time horizon, where system dynamics are assumed to be negligible. In this section, a dynamic optimization approach is developed, to optimize the production also during dynamic phases of the system, which occurs e.g. when wells are opened, closed or re-routed. The goal is to improve the production on a time horizon where system dynamics are significant.

4.2.1 Problem definition

Simulations of the wells show that they normally reach steady-state within 2-3 hours after a change in the input settings, depending on where the new operation point is, relative to the stability regions. The wells settle faster when they are far from the unstable regions. The linear stability region constraints may be defined further away from the unstable regions to decrease the time the wells need to reach steady-state, but this would decrease the feasible region for the optimization problem, and thus may reduce the oil production.

Using only the static optimization approach, when a planned event occurs, new well inputs, and possibly new routings, that are calculated using static optimization, are implemented at a certain time, and the well inputs are constant after the event, also during the 2-3 hours where system dynamics are significant. The idea with the dynamic approach is to build on the static optimization solution, by allowing the inputs to vary during this time horizon, introducing new degrees of freedom to the system, to increase the performance of the system.

Based on these considerations, the dynamic optimization problem is defined as follows:

- An event is defined as a change in the operational conditions of the system, e.g. when a well for some reason needs to be closed, or becomes available after being closed, or a re-routing needs to be performed, for example to test a well.
- The dynamic optimization should be performed whenever an event is planned, and implemented when the event occurs.
- When the event occurs, the wells are assumed to be producing at optimal steady-state settings, found using static optimization.
- All actions defining an event is assumed to occur simultaneously. (E.g., if two wells are re-routed, this is assumed to happen at the same time.)
- The time horizon for the dynamic optimization problem is set to 2 hours, as the main dynamics of the system occurs within this time interval.
- After the dynamic optimization's time horizon, the wells should produce at (or very close to) steady-state, at the new optimal settings found using static optimization.
- During the time horizon of 2 hours, the well routing is assumed to remain unchanged, and the well inputs are allowed to vary, only constrained by physical limitations.
- To reduce the number of decision variables, the well inputs are allowed to change only at $t = \{0, 5, 10, 15, 30, 45, 60, 75, 90, 105\}$ minutes after the occurrence of an event. At $t = 120$ minutes (after 2 hours), the optimal inputs calculated using static optimization is implemented.
- The dynamic optimization should provide a solution that improves the performance of the production system, compared to implementing the static optimal solution at $t = 0$.

4.2.2 Different formulations (methods)

The performance of the system is not easily measured. The main goal is to produce as much oil as possible, but system constraints should be respected. Also, varying production rates should be avoided, as this could cause problems in downstream processing equipment, in addition to added wear and tear on the system components. The dynamic optimization problem formulation should take all this into consideration. The static optimization solves this problem only on a long-term perspective. Short-term, when implementing the static optimal solution directly, the dynamics in the system could cause greatly varying production rates, and constraints may very well be violated.

In order to solve this problem, three different approaches are proposed:

Method 1 Define the objective function to maximize the oil production on the time horizon (2 hours), on the form:

$$\max_u \int_0^2 w_{op}(u,t)\, dt \tag{4.2.1}$$

(with t measured in hours). To respect the system constraints, and to ensure that the wells settle at the static optimization solution, the following constraints are added:

- Production constraints:
  - Average gas, liquid and water production in each flowline should not exceed the production limits
  - Maximum flow rates during the time horizon should not exceed the production limits by more than 10%. (The flowlines produce to separators, which also have a buffering function. Therefore, it is assumed that a relatively small violation of the production limits for a short period of time is not a problem, as long as the average productions over time are below the limits.)
- End constraint:
  - The system, defined by the state variables of the wells, should be at steady-state (static optimization solution) at the end of the time horizon

Method 2 Define the objective function to minimize the difference between the state variables and the steady-state masses on the time horizon, i.e. an objective function on the form:

$$\min_u \int_0^2 (m(t) - m_{ss})^2\, dt \tag{4.2.2}$$

(with t measured in hours). As it is assumed that this objective function would lead the system to steady-state relatively fast, no constraints are added to the problem. The idea behind this approach is that the steady-state solution provides the best solution on a long time horizon, and by using this objective function, the system should converge to this solution faster, providing a better, more stable production. However, the deviance in the beginning of the time horizon will always be relatively large, and will dominate this objective function. This may cause excessively aggressive control, and greatly varying production rates. One alternative to this is to replace m_{ss} by a time varying reference trajectory $m_{ref}(t)$, which leads the system from the initial state to the final steady-state. Another alternative is to introduce a time-varying weighting function in the objective function, so that the states towards the end of the time horizon are given more consideration than in the beginning. This idea is introduced in the third approach:

Method 3 Define the objective function to maximize the oil production on the time horizon as well as to minimize the deviance from steady-state, with an exponential time-varying weight, with an objective function on the form:

$$\min_{u} - \int_0^2 ce^{\left(\frac{-at}{2}\right)} w_{op}(u,t)\, dt + \int_0^2 e^{\left(\frac{bt}{2}-b\right)} (m(t) - m_{ss})^2\, dt$$

with t measured in hours, and where a, b and c are adjustment parameters. The exponential weighting functions, with $a = 10$, $b = 2$ and $c = 1$, are shown in figure 4.3.

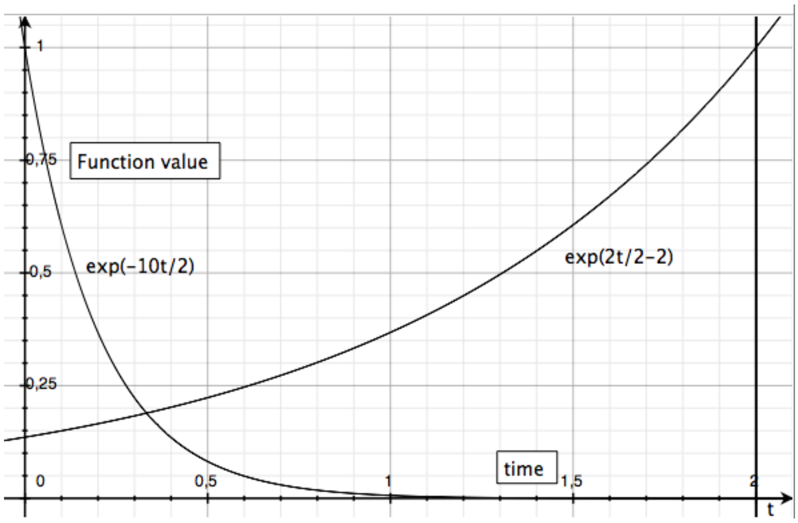


Figure 4.3: Exponential weighting functions

As with the second approach, the idea with this approach is to lead the system faster to steady-state. But in the beginning of the time horizon, the oil production rather than the deviance from steady-state is given consideration. The deviance is weighted more towards the end, where the system should stabilize at the optimal steady-state solution. The relative weight between these two goals is adjusted with the parameter c.

4.2.3 Linear constraints

All three methods have bounds on the inputs, and linear constraints implemented for the available lift-gas. Also, a linear constraint to avoid using a lot of gas-lift in one well if the production choke opening is small is implemented. Without this constraint, situations could occur where gas is injected into the reservoir. This is not wanted, and in this case would cause numerical errors in the optimization, as this is not considered in the model.

The linear constraints are:

$$0 < u_{pc,i} < 1, \qquad i \in \{1..7\} \tag{4.2.3}$$

$$u_{gl,i} > 0, \qquad i \in \{1..7\} \tag{4.2.4}$$

$$\sum_{i=1}^{7} u_{gl,i} \leq GL \tag{4.2.5}$$

$$u_{gl,i} \leq 10u_{pc,i}, \qquad i \in \{1..7\} \tag{4.2.6}$$

4.2.4 Method summary

The three dynamic optimization methods may be summarized as follows:

- Minimize the method's objective function, subject to:
  - A given routing
  - The initial state of the system
  - The linear constraints
  - Production and end constraints (method 1 only)

The decision variables are the wells' input variables, which may only change at

$$t = \{0, 5, 10, 15, 30, 45, 60, 75, 90, 105\}$$

minutes.

4.2.5 Implementation

The three dynamic optimization methods are all implemented in MATLAB, using the SQP algorithm in the function 'fmincon'. The routing is kept constant, and only the static optimization solution is used as a starting point for the optimization algorithm, so the problem only needs to be solved one time (as opposed to the static optimization, which is run $3 \cdot 2^{n-1}$ times for n wells). However, the problem now has $20 \cdot n$ decision variables for n wells (compared to $2 \cdot n$ decision variables in the static optimization), and thus is a lot more time consuming to solve.

The objective function value and the nonlinear production and end constraint values depend on the well dynamics. Therefore, the objective function and the nonlinear constraint function both run a simulation of the system at each function call. The simulations are performed using Euler's method (see section (3.5)), but with $dt = 20$. This is less accurate than with $dt = 0.5$, which is used in other simulations, but it is also 40 times faster.

The MATLAB implementation is found in appendix D.6.

4.3 Optimization free strategy

For comparison and performance assessment, a strategy to decide the routing and inputs without the use of numerical optimization is developed.

With this approach, the routing is decided first, based on the well parameters Q_{max}, r_{gor} and r_{wc}. The potential production of oil, water, liquid and gas, respectively, from each well are estimated as:

$$w_{o,pot} = (1 - r_{wc})Q_{max} \tag{4.3.1}$$

$$w_{w,pot} = r_{wc}Q_{max} \tag{4.3.2}$$

$$w_{l,pot} = Q_{max} \tag{4.3.3}$$

$$w_{g,pot} = r_{gor}(1 - r_{wc})Q_{max} \tag{4.3.4}$$

The potential production of oil, gas and water is then distributed as equally as possible between the two flowlines. This is solved by "brute force" in a MATLAB function (i.e. all the possible combinations are tried and compared, and the best solution is selected).

Then the distribution of lift-gas is decided. The lift-gas is distributed equally between the flowlines and equally between the wells routed to each flowline, as long as all the wells are within the stable regions, and the production constraints are not violated (see section 4.1.2). If not all the wells are stable, the gas-lift for the well furthest away from the stable region is set to zero, and the lift-gas is distributed between the remaining wells routed to the same flowline. If a production constraint for one flowline is violated, the amount of lift-gas is reduced for that flowline until the constraint is satisfied. The MATLAB implementation of this method is shown in appendix D.4.

This may be considered a sophisticated approach to the problem, but it does not involve numerical optimization methods. With this approach, one should expect quite decent and stable production levels.

Chapter 5

Performance assessment

In this chapter, the different production optimization methods described in chapter 4 are tested on different test scenarios. The performance of each method is assessed using simulations, and measures of average production.

5.1 Scenarios

To test and compare the different methods, three test scenarios are defined, described in this section. One scenario is without any dynamics, one has some dynamics, and one has extensive variability.

A scenario is defined as follows:

- A scenario lasts 2 hours
- A scenario is characterized by one or more events and a start condition
- An event is defined as a change in the operational conditions of the system, e.g. when a well for some reason needs to be closed, or becomes available after being closed, or a re-routing is performed, for example to test a well.
- The events of each scenario are assumed to be planned, so there is time to calculate optimal inputs and routing settings before the event occurs.
- Before the event occurs, the wells are assumed to be producing at optimal settings according to the production optimization method used.
- All actions defining an event is assumed to occur simultaneously. (E.g., if two wells are re-routed, this is assumed to happen at the same time.)
- During the time horizon of 2 hours, the routing is assumed to remain unchanged, but the well inputs may vary.

These are the scenarios defined:

Scenario 0

Description: A basic scenario where no changes occur

Identifier: Sce0

Name: No changes

Start condition: All wells produce at steady-state with optimal routing and input settings

Event: None

Scenario 1

Description: A scenario where one well is closed (e.g. due to maintenance)

Identifier: Sce1

Name: Single closing

Start condition: All wells produce at steady-state with optimal routing and input settings

Event: Well 1 is closed at $t = 0$

Scenario 2

Description: A scenario where one flowline is reserved for one well, e.g. during a well test

Identifier: Sce2

Name: Well test

Start condition: All wells produce at steady-state with optimal routing and input settings

Event: At $t = 0$, all wells except well 3 is routed to one flowline, while well 3 is routed to the other flowline

Other scenarios were also tested, including the reverse of scenarios 1 and 2, and a scenario combining a closing and an opening of a well at different times. A presentation of the results from those scenarios is omitted from this report, as they provided very little new information beyond the presented results. However, the omitted scenarios did show a consistency with the presented results, which may also be valuable information.

5.2 Optimization results

In this section, the main results from the different optimization methods applied to the test scenarios are presented. Simulations of the solutions may be found in appendix E.

5.2.1 Scenario 0

First, results from the optimization free method and the static optimization method are presented. The optimal routing found using these methods are shown in table 5.1. The optimal inputs are shown in table 5.2. In figure 5.1, the inputs are shown together with the stability regions.

The first dynamic method could not find a feasible solution, even though the initial point was the solution of the static optimization, which is feasible. The (infeasible) solution found with this method is presented as a simulation of the system, shown in figure E.1 in the appendix. Note the violation of the water production constraint in flowline 1, and the variations in the production in flowline 0 after the end of the scenario (after 2 hours). For comparison, the solution of the third dynamic optimization method is also shown, in figure E.2 in the same appendix. This is very close to constant, as expected when applied to this scenario. (In these figures, dashed lines represent constraints/limits.) No simulations are shown of the optimization free method, static optimization method, or the second dynamic optimization method, as these were all constant.

The average production for each method applied to this scenario is presented in table 5.3.

Well	1	2	3	4	5	6	7
Flowline (u_r)	0	0	0	1	1	0	1

(a) Opt. free method

Well	1	2	3	4	5	6	7
Flowline (u_r)	0	1	0	1	0	1	0

(b) Static optimization

Table 5.1: Routing, scenario 0

Well	1	2	3	4	5	6	7
u_{pc} $[-]$	1	1	1	1	1	1	1
u_{gl} $[kg/s]$	0.7954	0.7954	0.7954	0.7954	0.7954	0	0.7954

(a) Opt. free method

Well	1	2	3	4	5	6	7
u_{pc} $[-]$	1	1	1	1	1	0.7721	0.9974
u_{gl} $[kg/s]$	0.7000	0.6910	0.9200	1.2500	0.4500	0.5616	0.2000

(b) Static optimization

Table 5.2: Inputs, scenario 0

	Opt. free	Static opt.	Dyn. opt. 1*	Dyn. opt. 2	Dyn. opt. 3
Oil [bpd]	24 031	24 621	24 646*	24 621	24 621
Gas [Sm3/day]	834 260	843 560	844 590*	843 560	843 550
Water [bpd]	20 705	22 686	22 582*	22 686	22 687
Liquid [bpd]	44 736	47 307	47 228*	47 307	47 307

* The dynamic optimization method 1 did not find a feasible solution.

Table 5.3: Average production, scenario 0

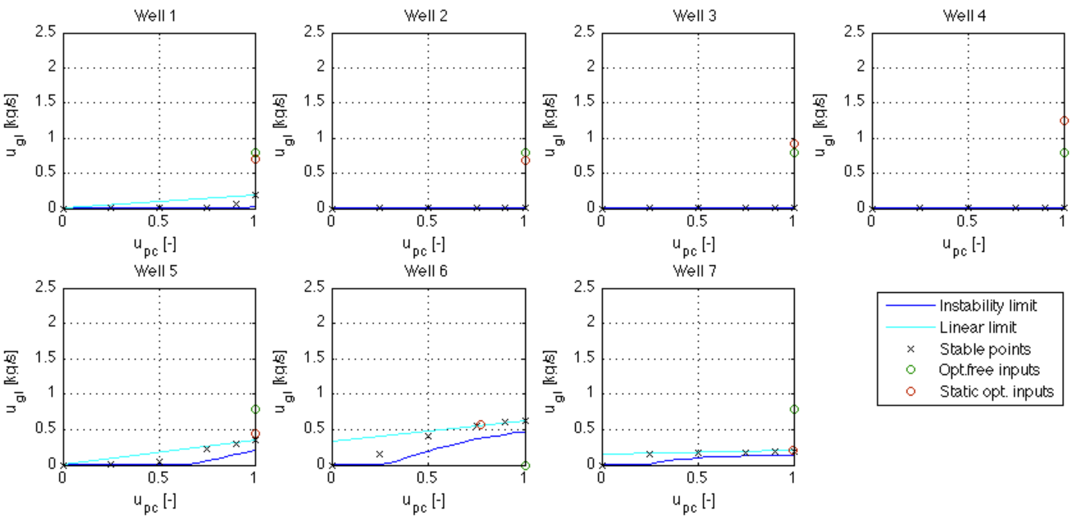


Figure 5.1: Inputs and stability regions, scenario 0

5.2.2 Scenario 1

The optimal routing found using the optimization free and the static optimization methods are shown in table 5.4. The optimal inputs are shown in table 5.5. In figure 5.2, the inputs are shown together with the stability regions.

The results from both the optimization free, the static optimization, and the dynamic optimization methods are simulated, and the simulations can be found in appendix E.2. The average production during the scenario (2 hours) for each method is presented in table 5.6.

Well	1	2	3	4	5	6	7
Flowline (u_r)	-	0	0	1	0	1	0

(a) Optimization free method

Well	1	2	3	4	5	6	7
Flowline (u_r)	-	0	0	1	1	0	0

(b) Static optimization

Table 5.4: Routing, scenario 1

Well	1	2	3	4	5	6	7
u_{pc} $[-]$	0	1	1	1	1	1	1
u_{gl} $[kg/s]$	0	0.5966	0.5966	1.1931	0.5966	1.1931	0.5966

(a) Opt. free method

Well	1	2	3	4	5	6	7
u_{pc} $[-]$	0	1	1	1	1	0.9143	0.9973
u_{gl} $[kg/s]$	0	0.8504	1.1000	1.5179	0.5000	0.6043	0.2000

(b) Static optimization

Table 5.5: Inputs, scenario 1

	Opt. free	Static opt.	Dyn. opt. 1	Dyn. opt. 2	Dyn. opt. 3
Oil	19 945	20 711	20 419	20 705	20 707
Gas	744 230	786 060	782 010	784 790	784 970
Water	19 949	20 676	20 500	20 667	20 669
Liquid	39 894	41 386	40 919	41 372	41 376

Table 5.6: Average production, scenario 1

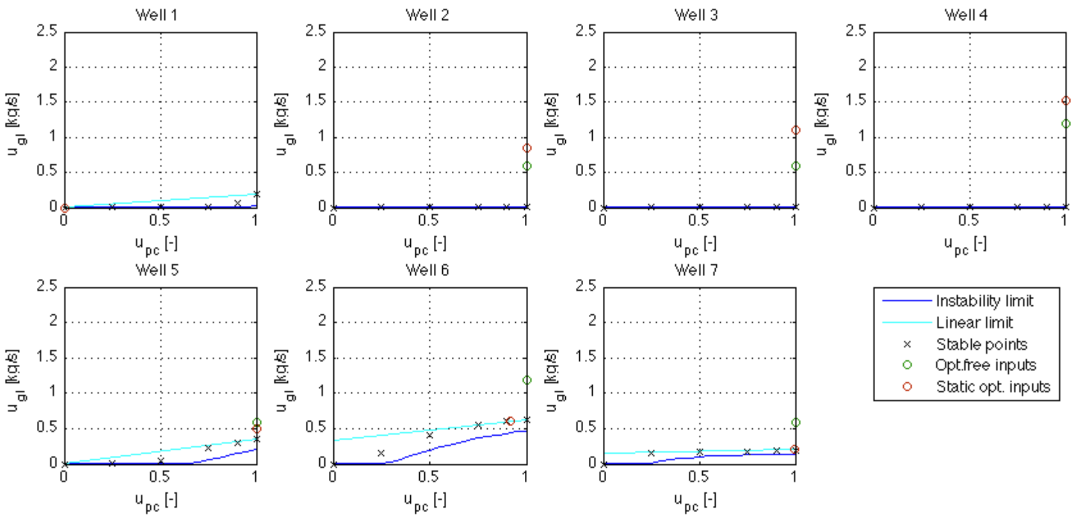


Figure 5.2: Inputs and stability regions, scenario 1

5.2.3 Scenario 2

The routing is the same for all methods in this scenario; all wells are routed to flowline 0, except from well 3, which is routed to flowline 1. The optimal inputs found using the optimization free and the static optimization methods are shown in table 5.7. In figure 5.3, the inputs are shown together with the stability regions. The average production during the scenario (2 hours) for each method is presented in table 5.8.

The results from all of the methods are simulated, and the simulations can be found in appendix E.3.

Well	1	2	3	4	5	6	7
$u_{pc}\,[-]$	1	1	1	1	1	1	1
$u_{gl}\,[kg/s]$	0	0.1731	2.3863	0.1731	0	0	0

(a) Opt. free method

Well	1	2	3	4	5	6	7
$u_{pc}\,[-]$	1	0.3996	1	1	1	0	0.9974
$u_{gl}\,[kg/s]$	0.8500	0	2.1769	1.1856	0.3600	0	0.2000

(b) Static optimization

Table 5.7: Inputs, scenario 2

	Opt. free	Static opt.	Dyn. opt. 1 *	Dyn. opt. 2	Dyn. opt. 3
Oil	20 692	22 761	22 889*	22 797	22 806
Gas	592 110	820 140	754 550*	800 940	801 160
Water	16 447	16 279	18 815*	16 814	16 813
Liquid	37 139	39 040	41 704*	39 611	39 619

Table 5.8: Average production, scenario 2

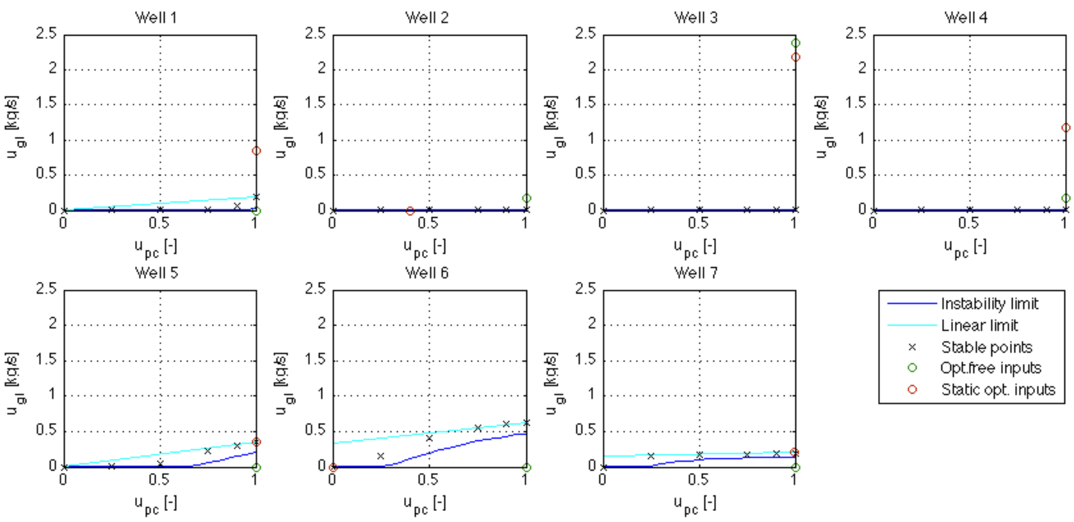


Figure 5.3: Inputs and stability regions, scenario 2

5.3 Solution times

The solution times required for the MATLAB implementations of the methods are presented in table 5.9.

	Opt. free	Static opt.	Dyn. opt. 1	Dyn. opt. 2	Dyn. opt. 3
Sce0	< 1	3744	101	15	768
Sce1	< 1	1846	252	876	716
Sce2	< 1	107	541	32	49

Table 5.9: Solution times (seconds)

The solution time for the static optimization is dependent on the number of possible routings and the number of open wells. Only one of the (two) CPU cores was utilized, however, there is a great potential for multithreading in the static optimization method, both for the different routings, the different starting points, and the different branches in the Branch and Bound search tree.

With the dynamic optimization methods, the solution times were greatly varying, from 15 seconds, to almost 15 minutes, without any apparent pattern. Method 2 applied to scenario 1 terminated after 102 iterations and 876 seconds. The objective function value, and the step size, as functions of the iteration count is shown in figure 5.4. After 50 iterations, the objective function value is less than 0.4 % above the final value. In other cases (the ones with the shortest solution times), the only improvement is seen in the first iteration, as is in figure 5.5, which shows the iterations of method 3, scenario 2. This method terminated after 3 iterations, as the step size was smaller than the step size tolerance, set to 10^{-6}. Changing this tolerance did not seem to affect the solution much. The same method with a step size tolerance of 10^{-10} terminated after 5 iterations, and the objective function value was less than 0.05 % smaller. This is shown in figure 5.6.

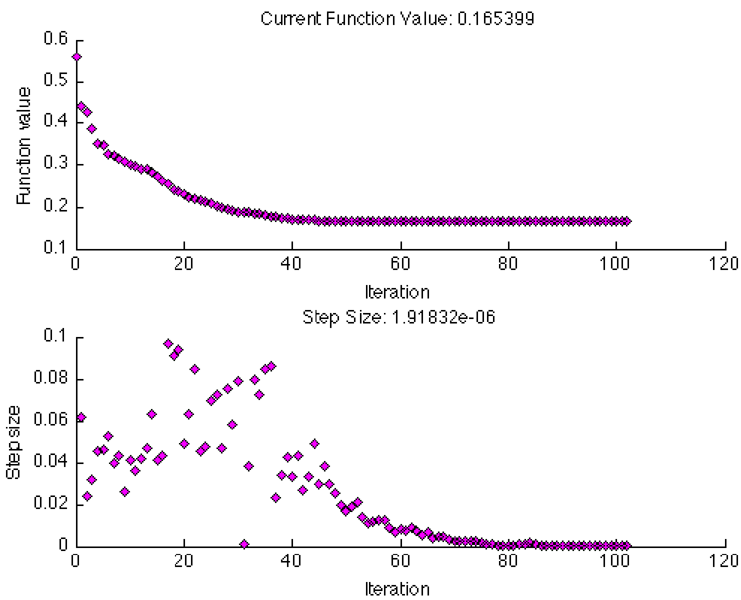


Figure 5.4: Function value and step size as functions of iteration, method 2, scenario 1

The solutions were calculated on a MacBook Pro, with a 2.53 GHz CPU (Intel Core 2 Duo), and 4 GB of RAM (DDR3).

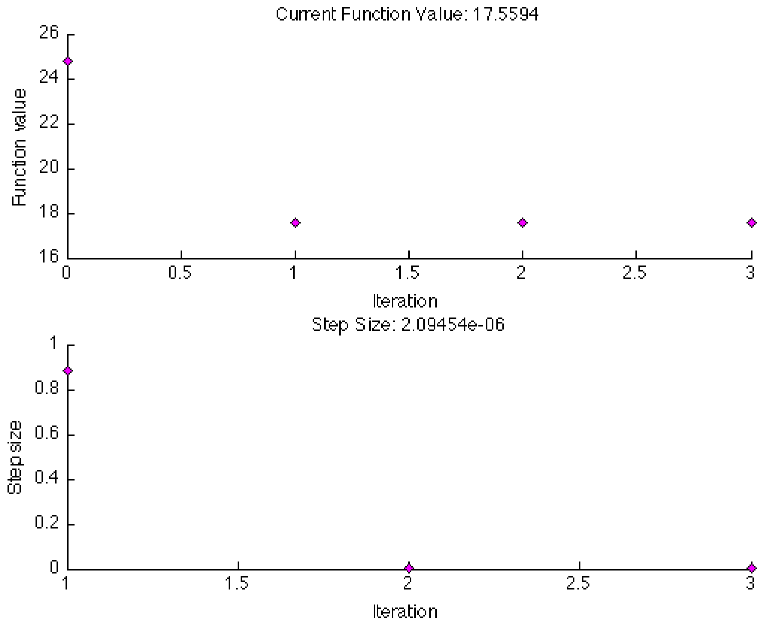


Figure 5.5: Function value and step size as functions of iteration, method 3, scenario 2

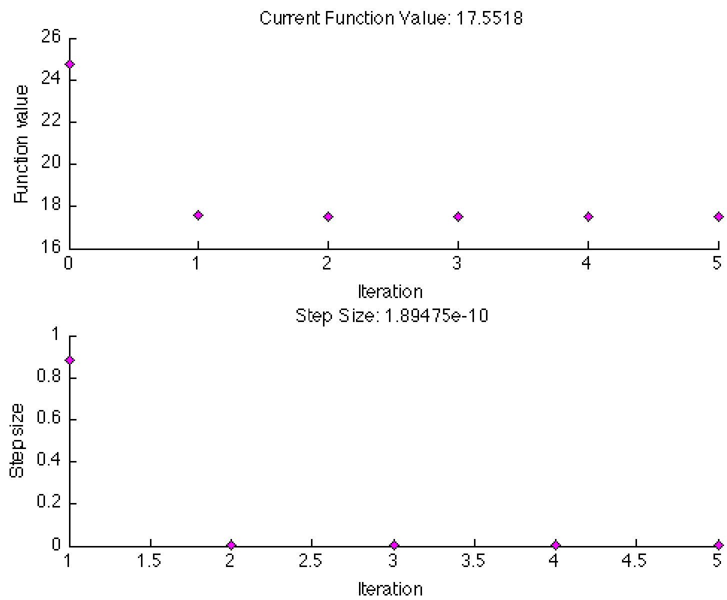


Figure 5.6: Function value and step size as functions of iteration with a reduced step size tolerance, method 3, scenario 2

5.4 Performance summary

Table 5.10 shows the average oil production rates during the three different scenarios, using the different production optimization methods. Note that these numbers differs from steady-state results, i.e. when the wells have reached steady-state after the end of the scenarios, which are presented in table 5.11.

Table 5.12 shows the oil production increase from applying static optimization, compared to the optimization free method, at steady-state, which may be the most interesting on a long-term point of view.

Table 5.13 shows the oil production increase (or decrease) during the scenarios, from applying the dynamic optimization methods, compared to static optimization. These numbers should be put in context by considering the simulations in appendix E. Note for example that the dynamic optimization method 1 did not provide feasible solutions for scenario 0 and scenario 2, and simulations of the (infeasible) solutions show a poor performance regarding variability of the flow rates, and violations of the production constraints.

	Opt. free	Static opt.	Dyn. opt. 1	Dyn. opt. 2	Dyn. opt. 3
Sce0	24 031	24 621	24 646*	24 621	24 621
Sce1	19 945	20 711	20 419	20 705	20 707
Sce2	20 692	22 761	22 889*	22 797	22 806

* Not feasible solutions

Table 5.10: Performance summary (average oil production, [bpd])

	Opt. free	Static opt.
Sce0	24 031	24 621
Sce1	20 157	20 695
Sce2	20 753	22 633

Table 5.11: Oil production at steady-state [bpd]

Scenario	Increase (%)
Sce0	2.46
Sce1	2.67
Sce2	9.06

Table 5.12: Oil production increase at steady-state using static optimzation (percent)

	Method 1	Method 2	Method 3
Sce0	0.10*	0	0
Sce1	-1.41	-0.03	-0.02
Sce2	0.56*	0.16	0.20

*Not feasible solutions

Table 5.13: Oil production increase using dynamic optimization (percent)

Chapter 6

Discussion

6.1 Optimization and control

Optimization and control has made it's way into many aspects of the petroleum industry, especially in onshore processing facilities (refineries), where e.g. MPC controllers are widely implemented. Offshore, however, little optimization or automatic control has been applied, thus having all the more potential.

The benefits of optimization may be severe. The amount of data in a production system may be enormous, and there is a large potential for gathering more information from these data, to make better decisions. Formulating and solving mathematical optimization problems may form the foundation for various decision support tools, for decisions made on any time horizon. However, this may require some adjustment in the mindset of both production engineers and offshore personnel. (See also section 1.1.5.)

6.2 Modeling

The analytical well model used in this report is based on the 'Eikrem model', presented in [Eikrem 2008]. A slightly modified version of this model was used in [Binder 2011b], and the model has been further modified in this report. The exact modifications compared to the Eikrem model are discussed in section 3.3, but as may be seen in the test simulations in section 3.5, the dynamics of the model are quite similar to those of the Eikrem model. However, some differences may be observed in pressure levels, and variations especially in liquid production are far less severe in the modified model. The density of the produced fluid is also radically different from the Eikrem model, which is a natural result of the underlying assumptions in the modeling.

The modified model is, of course, not a perfect model, which is not a reasonable goal. Some severe simplifications have been made regarding friction and multiphase flow, as discussed in section 3.2, and other more or less minor assumptions and modifications as discussed during the model derivation in section 3.1. A lot of effort could be put into adding friction and/or multiphase flow into the model. This would probably yield a more accurate model, but a more accurate model would also be a more complex one, and the benefits are uncertain.

However, [Eikrem 2008] claims that the 'Eikrem model' catches the essential dynamics of gas-lift wells. The author of this project report is reluctant to believe that the modifications made in this report have made the model less accurate, and would rather believe the opposite. This is, however, not documented by any comparison with real well data or laboratory experiments, or even by simulations using advanced multiphase flow simulation software, like OLGA. If this model should be used in any way for production optimization purposes, the model should be validated and/or fitted to the real system e.g. by applying system identification techniques to real production data, which was not readily available during this project work.

6.3 Optimization methods and results

6.3.1 Static optimization

Table 5.12 shows an oil production increase of 2.46 % from applying static optimization, when all wells and flowlines in the manifold are available (scenario 0), which may be considered as the main benchmark. This is a very promising result. The same table shows an increase of 9.06 % when one flowline is reserved for a well test (Scenario 2). This number may also indicate a weakness in the optimization free method, but nevertheless points out how difficult it may be to make good decisions in a complex production system, without the use of numerical optimization.

Simulations of the dynamic behavior during the scenarios are shown in appendix E. Considering these, and comparing table 5.10 and 5.11, the static optimization method seems to provide a better dynamic performance in addition to an increased steady-state oil production, compared to the optimization free method, but this may just be a coincidence.

Especially in scenario 2, some production constraint violations are observed in flowline 0. In practice, it may be necessary to introduce a safety margin between the limits used in the optimization and the real system constraints, to avoid such violations. Alternatively, one needs to reduce the production by some other means, e.g. by choking down the production from a flowline, reducing the gas-lift, or switching between separators. However, one needs to consider the entire production system when making such decisions. The constraint violations observed may be insignificant compared to other factors in the production system.

The results in this report are based on a almost perfect knowledge of the wells' behavior and performances. A more realistic situation would be where the production from each well, as functions of the well inputs, were approximated based on real production data, and/or simulations using state-of-the-art simulators, like OLGA. Still, the results reveal a great potential for applying optimization techniques to increase the oil production in a field. The increase seen in table 5.11 from 24031 bpd to 24621 bpd, an increase of 590 bpd, at an oil price above $80 a barrel, yields an increased profit of about $50 000 per day, a decent annual salary in Norway. And this is just one subsea manifold. Applying the same ideas to entire production systems, with several manifolds and satellite wells, introducing even more degrees of freedom, may have an even greater potential. This is investigated by previous and ongoing related projects, e.g. in [Dzubur 2011].

6.3.2 Dynamic optimization

Looking at production data from one well in the Campos basin, provided by Petrobras, the well input settings change only about 10 times per month (every 3 days on average). This may or may not be the typical case, but has been the only reference available in this project. Assuming this is typical, that wells operate at the same operating point for several days at a time, the steady-state performance is of far greater concern than dynamic behavior resulting from a change in operating conditions, which simulations show that lasts only for a couple of hours at most (see e.g. simulations in appendix E). Therefore, the improvements from using dynamic optimization need to be even greater to justify the efforts and investments needed to implement it.

However, the results in chapter 5 are not very promising. Table 5.13 shows only a slight oil production increase, of less than 0.2 %, as the best result from applying the proposed dynamic optimization methods on the defined test scenarios, compared to applying the static optimization method's solution directly. However, this may not be the only measure of performance. It is also important to keep stable production rates, and to respect system constraints. Considering the simulations presented in appendix E, dynamic optimization method 2 seems to achieve this better than the other methods. (Using this method, the oil production is not set as an objective directly, but indirectly through the steady-state solution, which is the control objective in this method.) With this method, the oil production remains unchanged in scenario 0, which is expected as no changes occur. In scenario 1, the oil production is reduced by 0.03 %, but the simulation shows a fast stabilization at steady-state. In scenario 2, the oil production is increased by 0.16 %. However, the simulation shows that the liquid constraint in flowline 0 is slightly violated.

Comparing the three dynamic optimization methods, weighting the oil production in the objective function seems to only cause more unstable production and more severe violations of system constraints. Weighting the steady-state solution seems to stabilize the production, with very little impact on the oil production.

It proved difficult to find feasible solutions to the problem with method 1. Only in scenario 1, a feasible solution was found, but the result was an oil production reduced by 1.41 %, and a very poor performance. This shows that it may be difficult to formulate the dynamic optimization problem in a robust way, and that proposed solutions need to be evaluated before they are implemented.

The results and simulations do not indicate a major performance increase, neither with respect to oil production, variations in the production rates, or constraint violations, with any of the proposed dynamic optimization methods. However, other formulation may exist that provide better results. But, considering that this is a very ideal case, where the system dynamics are completely known without any uncertainties, one should expect even poorer results in practice, where models are incorrect, and disturbances and uncertainties may affect the system. In practice, it could be necessary to introduce feedback, e.g. by implementing a moving horizon or shrinking horizon controller, to achieve acceptable results. Applying these methods would be even more complex, and would require an even more active use of subsea valves, which in many cases is not desirable, due to maintenance.

6.3.3 Solvers

The time required to solve the optimization problems may be of great concern, as 'events' (defined in section 5.1) may occur on short notice. A lot of effort was put into solving the optimization problems fast. However, fmincon in MATLAB is probably not the fastest alternative when it comes to solvers, and probably another solver should be used if one of the methods should be implemented in practice, e.g. BONMIN (Basic Open-Source Nonlinear Mixed Integer programming), distributed through the COIN-OR project (Computational Infrastructure for Operations Research).

6.4 Hardware requirements

As item 6 in the task description for this master thesis suggests, optimization and control methods often require certain (accurate) measurements and actuators. If feedback control is implemented, it is important to have reliable feedback from the system, in order to control it.

However, none of the proposed production optimization methods in this report suggests implementing feedback control, thus no real-time measurements are necessary. The exception is that the well model assumes that the gas-lift flow rate for each well is controlled, and used as an input variable. This may be achieved using real-time measurements of the lift gas pressure, and the pressure in the annulus of each well, and using the gas-lift choke valve as an actuator. Alternatively, the model could be formulated differently, using the gas-lift choke valve opening as an input variable. This would increase the complexity of the model, and make the formulation of the optimization problem more difficult, e.g. when formulating the total available lift-gas constraint, as the relation between the gas-lift choke valve opening and the gas-lift flow rate depends on the pressures.

Measurements of the pressures in the well heads are available in the production system considered in this project, but a general reluctance to use subsea valves as control actuators is observed, mainly due to maintenance issues. Replacing a subsea valve may be very expensive, both the operation itself, and the resulting loss of production.

The dynamic optimization methods require more active use of the subsea choke valves. However, this may be limited using a proper input blocking. (The methods proposed in this report limits the control actions to 10 actions per valve per event.) To decide this, one would need to weigh the performance increase versus the increase in maintenance, to find a compromise.

The model requires well parameters that may vary with time (the gas-to-oil ratio r_{gor}, open-flow potential Q_{max} and water cut r_{wc}). To obtain these parameters, well tests need to be performed on a regular basis. However, this is already common practice. The model may also need to be fitted with real production data, but this is assumed to be available for production engineers that may use this approach.

Currently, in the manifold studied in this project, one flowline is used for well tests, while the other is used as a production flowline. However, according to the performance assessment, when well tests are not necessary, both flowlines should be used for production. (Compare scenario 0 and scenario 2 in chapter 5.)

Chapter 7

Conclusion

Whereas optimization and control has made great success in many industries, also in onshore segments of the petroleum industry, it is not widely applied offshore. This report shows that there is a great potential for using numerical optimization to increase the profits from offshore production systems. Results presented show an increase in oil production of 2.5 % from applying a static optimization method to decide the routing and input settings of seven wells producing to one subsea manifold with two flowlines. This is consistent with previously presented results (in [Binder 2011b]), which also indicate that this number could be even higher if the casing heading instability is not experienced in the production system (whether from natural causes, or by implementing stabilizing PI-controllers). This is not discussed in this thesis, but nevertheless deserves consideration.

The main goal for this project work was to investigate whether or not dynamic optimization could improve the performance of the production system even more. The results from the three proposed dynamic optimization methods show little indications of any major performance improvement. The improvements observed would hardly justify the effort and investments required to implement the methods in practice. Even though one can not rule out that there may exist other problem formulations and solution methods that may yield better results, this report shows that formulating and solving a dynamic optimization problem to increase the oil production may be quite complicated, and difficult to implement in a robust way. One should also consider that the proposed methods were tested with perfect knowledge of the system and the well dynamics, which is not realistic. In practice, dynamic optimization would be even more difficult to implement than what was experienced during this project work, and the achieved results would probably be inferior to the results presented in this report.

Regarding hardware requirements, none of the methods proposed in this project requires any change in the hardware, but they may require a change of attitude towards how the hardware should be used.

Chapter 8

Further work

Even though this report indicates a great potential for using static optimization to increase the oil production, there is a long way to actually implement this in practice. First of all, the static model needs to be verified against and/or fitted to real production data. A way must also be found to be able to use the gas-lift flow rate as input variable, instead of the choke opening. Alternatively, a different model may be implemented, e.g. based on simulations using PipeSim or OLGA (which has been done in other recent and ongoing projects, related to this one through the IO-center and Petrobras, like [Dzubur 2011]). Real data from well tests and/or logging data should also be utilized, to form a reliable static system model.

Different solvers should also be considered, and the utilization of multi-threading on multi-core processors, to speed up the optimization.

The idea of stabilizing casing heading instabilities using PI controllers, as discussed in [Binder 2011b], should be further investigated, both to see how this could be done in practice, and what the benefits would be.

The system considered in this report only consists of one subsea manifold, and a constant pressure in the manifold is defined as a boundary condition to the other parts of the production system. It was necessary to define a relatively small and simple system to investigate dynamic optimization, but with static optimization, it is more interesting to look at the entire production system, like in [Dzubur 2011].

Even though the dynamic optimization methods in this report show little promising results, there may be other ideas or problem formulations that would provide a better performance. This may be given further consideration. If such methods are developed, in practice, open-loop dynamic optimization methods (like the ones proposed in this report) would probably not provide acceptable performance, due to uncertainties and unknown/unmodeled disturbances. It could be interesting to investigate the possibilities and benefits of implementing an MPC controller (or some other kind of feedback controller) for the system. However, this could require other measurements than the ones available, and estimators or adaptive control may be required.

Appendix A

Control system design

This appendix provides an introduction to some basic control theory concepts, such as modeling, feedback control, decentralized control, and other related topics. This is mainly based on the literature [Chen 1999, Egeland 2002, Balchen 2003, Skogestad 2005].

The goal when constructing a controller is usually to satisfy given system specifications, but may also be to improve the performance of existing systems. The benefits of control may e.g. be increased productivity or efficiency of the system, better quality of product(s), or less environmental impact. There are many kinds of controllers, from simple gains, to advanced decision algorithms based on optimization or even heuristics. The controller type and structure will always depend on the system that is to be controlled, and adapt to local conditions.

A.1 Basic concepts and terminology

A *system* (also often referred to as a *plant*) in this report refers to a defined set of units that influence each other, and together have a function. A *dynamic system* is a system where certain conditions/states change with time. A system's dynamics may be described mathematically, and this description is then referred to as a system *model*. A system's *states* are internal variables in a system model, often denoted by the state vector $\mathbf{x}$ (or the state variable x for monovariable systems).

The concept of *stability* is of fundamental importance in control theory, and a system's stability properties may be investigated using an appropriate system model. A point $\mathbf{x_e}$ is said to be an *equilibrium point* of a system if the system stays in $\mathbf{x} = \mathbf{x_e}$ whenever the system starts in $\mathbf{x_e}$. The equilibrium point $\mathbf{x_e}$ is said to be stable if the system stays close to $\mathbf{x_e}$ whenever it starts close to it. It is said to be asymptotically stable if it is stable, and the states converges to it. All these definitions apply to systems that are not influenced by external variables.

Control, or *dynamic control*, refers to using available degrees of freedom (*actuators, inputs*) to manipulate certain variables (*measurements, outputs*) of a dynamic system, to achieve acceptable or desired behavior of the system (given by a *reference* or *setpoint*). This is done by a *controller.* A system may not be able to perform acceptably without active control (continuous measurements and actions) due to e.g. external influences on the system (*disturbances*), or stability issues. *Tuning* of a controller refers to adjusting internal parameters in the controller, to achieve acceptable control.

Two main principles of controlling dynamic systems are *feedback* control and *feedforward* control. Feedback refers to control using measurements of the system. Feedforward refers to using measurements of disturbances or the reference signal when controlling the system. Because the signal flow in a feedback controlled system forms a loop, this is also referred to as *closed-loop* control, as opposed to *open-loop* control where no feedback is implemented in the controller.

When desired measurements are not available, *estimators* (also called *observers*) may be implemented. Estimators use system models to estimate or predict the value of process variables or disturbances, and function as extra measurements.

A.2 Modeling

To control a dynamic system, the behavior (dynamics) of the system should be described mathematically (*modeled*), and a controller designed based on this information. The *model* could e.g. be in the form of mathematical differential equations or transfer functions. The models may be derived based on concepts from physics, such as Newton's laws, balance equations (conservation of mass, energy, momentum, electrical charge), kinematics, thermodynamics, fluid mechanics, etc. In addition, empirical knowledge of the system and/or its components may be necessary (e.g. the relationship between applied force and deformation of a spring). Even though sufficient knowledge of a system is present to formulate e.g. a differential equation model of the system, certain parameters (properties of the system) may be unknown. Such parameters may be found using system identification techniques, based on measurements of the system (see e.g. [Ioannou 1996]). Some control applications are even based on empirical response models derived solely from measurements of identification experiments on the system.

System models will always be an approximation, as it is impossible to describe any physical system exactly. The main function or goal of the model is usually to enable an approximate prediction of a system's response to certain actions. It is therefore only necessary to describe the dominant dynamics of the system, i.e. dynamics with relatively large significance to the performance objective of the system.

There are many kinds of systems, and mathematical models describing them. One important characteristic of a model is the number of inputs, outputs and states of the system. A system with a single input and a single output is called a SISO system, while a system with multiple inputs and outputs is called a MIMO system. The latter is often also referred to as a multivariable system. The internal state variables, together with the inputs, contain all necessary information to calculate the outputs. If the number of states is finite, the system is called a lumped system. Otherwise, it is called a distributed system.

Another important aspect is if the system is modeled as a continuous-time or discrete-time system. Physical systems are continuous-time systems, while computers run on discrete-time (decisions are made in fixed time steps). Formulating a discrete-time system model may therefore be very useful to implement a controller on a computer.

A system can be linear or non-linear. A linear system is a system where the superposition property holds, a combination of additivity and homogeneity. (A non-linear system is a system where the superposition property does not hold.) A system is said to be time invariant if the dynamics of the system does not change with time. (See [Chen 1999] for more precise definitions of these properties.) Linear Time Invariant (LTI) systems hold both these properties, and form an important class of systems in control theory, as many physical systems can be modeled as LTI systems. Every lumped LTI system can be modeled on the form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \tag{A.2.1}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are constant matrices, $\mathbf{x}(t)$ is the state vector of the system, $\mathbf{u}(t)$ is the input vector, $\mathbf{y}(t)$ is the output vector, and $\dot{\mathbf{x}}(t)$ denotes the continuous time derivative of the state vector. ($\mathbf{x}$, $\dot{\mathbf{x}}$, $\mathbf{u}$ and $\mathbf{y}$ are variables of time, though the notation with the time variable (t) often is omitted.) This representation is called a state-space representation. It may also be applied for discrete-time systems, then on the form:

$$\begin{aligned} \dot{\mathbf{x}}_{k+1} &= \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C_d}\mathbf{x}_k + \mathbf{D_d}\mathbf{u}_k \end{aligned} \tag{A.2.2}$$

where the integer subscript index k refers to the time $t_0 + kT$, where t_0 is some initial time, and T is the sampling time interval. The system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ will not be the same in the discrete time state-space representation as in the continuous time representation, thus the subscript $\mathbf{d}$ for the discrete time matrices. The state-space representation may be used to easily investigate properties like stability, controllability and observability of the system.

Most physical systems are nonlinear and time varying, and can not be described using the state-space representation given in equation (A.2.1), but on the more general state-space form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \end{aligned} \tag{A.2.3}$$

where $\mathbf{f}$ and $\mathbf{h}$ are possibly nonlinear functions of their input variables. However, the dynamics of many such systems can be approximated by linear equations on the form (A.2.1) using partial differentiation. This is called linearization, and is often used in practice. The system is then linearized around one operating point, $(\mathbf{x}_0, \mathbf{u}_0)$, that the system is expected to operate close to. This is often an equilibrium point of the system (so that $\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = 0$), but not necessarily. This approximation will in general be most precise close to the point of linearization. [Chen 1999, Egeland 2002]

Transfer functions are important tools in the study of dynamic systems. A transfer function of a system enables the use of important analysis and design methods in the Laplace description (including stability analysis), and serves as a good starting point for frequency analysis of the system. A system's transfer function is derived by a Laplace transform of the system model. If the Laplace transform of the system can be written as:

$$y(s) = H(s)u(s), \quad \Leftrightarrow \quad \frac{y(s)}{u(s)} = H(s) \tag{A.2.4}$$

where $y(s)$ is the Laplace transform of the output, and $u(s)$ of the input, then $H(s)$ is said to be the system's transfer function. (If the system is MIMO (multivariable), y and u are vectors, and the transfer function $H(s)$ is a matrix.) The transfer function's poles and zeros (singularities in it's denominator and numerator, respectively) contain a lot of information about the system's dynamic properties. The poles are closely related to the system's eigenvalues. The transfer function also enables frequency analysis of a system, which is a collection of graphical analysis methods, based on magnitude and phase of the transfer function inserted $s = j\omega$ (equivalent to the Fourier transform of the system). Bode diagrams and Nyquist diagrams are two well-known and widely used frequency analysis methods. A main advantage of frequency response analysis is that it can be given a direct physical interpretation; the system's response to a sinusoidal input signal.

A.3 Feedback control

As mentioned in section A.1, feedback control refers to using measurements of the system when deciding actions to be performed by the actuators. Feedback control is generally used in the presence of signal uncertainty (e.g. unknown disturbances), model uncertainty or unstable plants. This is the case in a large variety of dynamic systems, and feedback controllers are widely used in a variety of practical applications. There are also many different kinds of feedback controllers, with varying complexity. This section explains the concepts for some basic feedback controllers.

Figure A.1 shows a common control structure for a single input single output (SISO) system, and will illustrate the discussion in this section. The input to the controller is the error (e), which is calculated by subtracting a measurement (y) from a reference setpoint (r). The input signal to the process (u) is the output of the controller, and is calculated by some rule in the controller. The process output is the measurement (y). The process is also affected by a disturbance (v), but this is not measured directly.

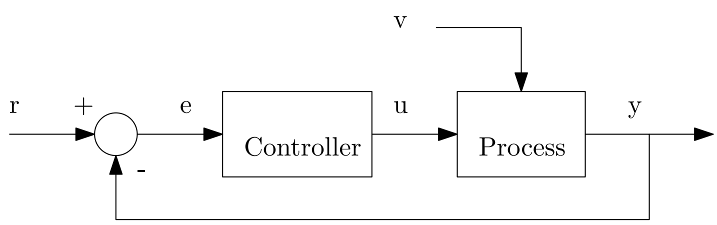


Figure A.1: Feedback control

The controller in figure (A.1) could be a very basic feedback controller. There are four commonly used basic feedback controllers:

- Proportional controller (P controller)
- Proportional and derivative controller (PD controller)
- Proportional and integral controller (PI controller)
- Proportional, integral and derivative controller (PID controller)

The most basic feedback controller is probably the proportional controller (P controller). With a P controller, the input signal (u) is simply calculated as the error (e) multiplied with a constant (a gain), here denoted K_p:

$$u(t) = K_p e(t) \tag{A.3.1}$$

Even though a P controller may in some cases provide sufficient control of a process, it is most often not considered an acceptable controller solution. The other basic feedback controllers most often are able to provide faster, more precise control. A common problem with the P controller (as well with the PD controller) is that the setpoint/reference in some cases is never reached at steady-state (when the system is stabilized), even without disturbances, and with perfect measurements. This is referred to as steady-state/stationary deviance/offset, and whether this problem is present depends on the process itself.

The PD controller introduces derivative action in addition to the proportional gain found in the P controller. The input signal u may then be calculated as:

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \tag{A.3.2}$$

The PD controller may often provide faster, more precise control than the P controller. However, at low frequencies (where only very small changes are present in the error signal e), the PD controller is basically just a P controller, and the same problem with steady-state deviance may be present.

The PI controller eliminates the problem of steady-state offset by introducing integral action in the controller, in addition to the proportional gain. In a PI controller, the input signal u may be calculated as:

$$u(t) = K_p e(t) + K_i \int_{t_0}^{t} e(\tau)\, d\tau \tag{A.3.3}$$

where K_i is the integrator gain. This controller is versatile, and widely used for many different applications, especially in the process industry.

The PID controller combines proportional, integral and derivative action. The input signal may be calculated as:

$$u(t) = K_p e(t) + K_i \int_{t_0}^{t} e(\tau)\, d\tau + K_d \frac{d}{dt} e(t) \tag{A.3.4}$$

The resulting controller may in many cases provide fast, precise control without steady-state offset, and this controller is also used in many different applications.

The formulations of the controllers given in this section capture the basic principles of these controllers. However, there are many different formulations and modifications to the controllers not mentioned here, including transfer function formulations, limited integral or derivative action and anti integral windup modifications. (See e.g.[Balchen 2003, Skogestad 2005] for a more extensive description on these subjects.)

There are several other, more advanced alternatives to the basic feedback control described in this section. This is discussed in section 2.3.

A.4 Feedforward control

Whenever a disturbance to the process is measured, the effect of the disturbance on the process may be modeled, and a feedforward controller designed to take advantage of this information. This could reduce the disturbance's (unwanted) impact on the process before it occurs, and thus increase the performance of the system. This is especially useful at frequencies where feedback is not effective. Feedforward is typically used in combination with feedback, as shown in figure (A.2).

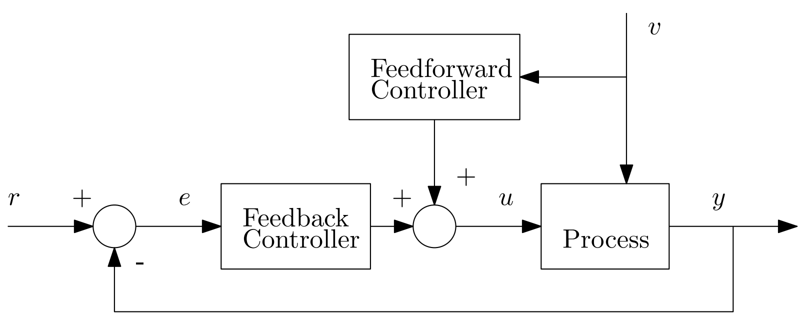


Figure A.2: Feedforward controller

Feedforward may also be used in systems with time-varying reference, to reduce the effect of the dynamics in the system. This is shown in figure (A.3). Such a controller should be designed based on a model of the process, to “invert” the process dynamics, so that the output y will follow the reference r as closely as possible.

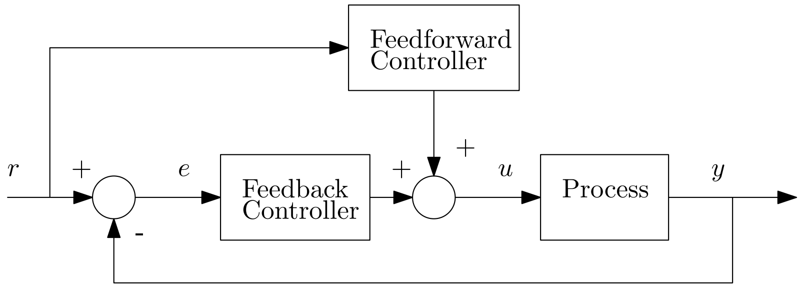


Figure A.3: Feedforward from reference signal

A.5 Decentralized control

In a MIMO system (multi-input multi-output system, a system with several inputs and outputs), there are many possible control configurations, i.e. possible ways to interconnect inputs and outputs of the system. One possibility is to implement a multiple of individual feedback controllers, each using one output to control one or more input, and where each input and each output of the system is used by at most one such controller. This is referred to as decentralized (or diagonal) control. This is considered the simplest approach to multivariable control, and it works well in systems with few or minor interactions. However, in a plant with severe interactions (where one input affects several outputs, or one output is affected by several inputs), the performance may be poor, as the interactions are not handled by the controllers.

A key element in decentralized control is to choose a good pairing of the inputs and outputs, primarily to minimize the effects of interactions in the controlled system. Secondly, the controllers must be designed and tuned. This may be done fully coordinated, independently, or sequentially. For more detail, see [Skogestad 2005], where an extensive explanation and discussion on decentralized control is found.

A.6 Cascaded control

Cascaded control is a special case of feedback control, sometimes used when there are more outputs available than inputs. A cascaded control structure is decomposed into subcontrollers. The input for a cascaded controller is the output from another controller. Usually, the subsystem controlled by a cascaded controller has much faster dynamics than the main control objective.

Example As an illustrative example, consider a system containing a tank with an uncontrolled and unmeasured flow of liquid into the tank, and a control valve deciding the flow out of the tank. Assume that the goal is to control the tank level to a constant setpoint, and that the tank level and the flow rate through the valve are measured variables. The controller could be designed as shown in figure (A.4).

Here, the valve is considered a subsystem, where the flow rate is controlled by the controller K_2. The setpoint for K_2 is the output from the controller K_1, which controls the tank level.

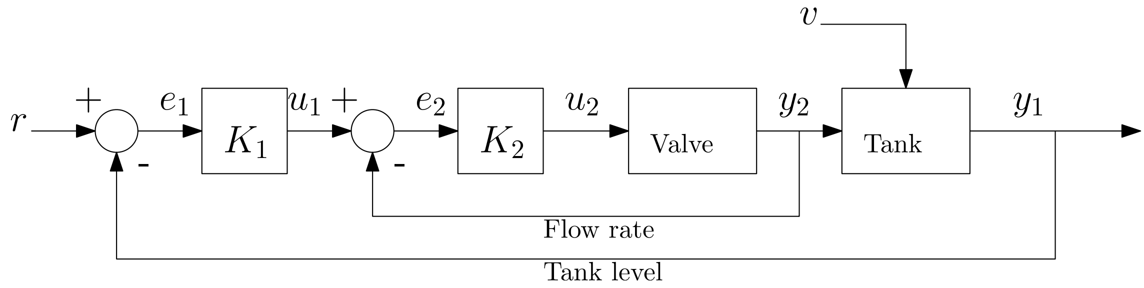


Figure A.4: Example of a cascaded control configuration

Here, K_2 is a cascaded controller. Note that in this system, there are more measurements than inputs, and the controllers use separate measurements. □

This is far from the only possible control structure for such systems, but it benefits from the extra measurements. (In the example, without this measurement, the tank level would have to be controlled directly by the valve opening.) The benefits of this structure are that internal dynamics and possibly nonlinear characteristics are removed by adding the cascaded controller. For the outer control loop, these dynamics appear linear. This will probably make tuning of the controller (adjusting the controller parameters) easier. The inner control loop may operate at a faster frequency, and thus increase the performance of the controlled subsystem. Disturbances arising within the inner control loop will not affect the outer loop, as these are handled internally. Also, this control structure may in some cases be more intuitive to a plant operator. [Skogestad 2005]

A.7 Compensators

In a system with severe interactions, one approach is to design a "compensator". A pre-compensator is designed based on system models, with the purpose to filter or shape the input signals to counteract interactions in the system. A system with a pre-compensator implemented may be viewed as a new plant, which is easier to control with a decentralized controller.

Decoupling results when the pre-compensator is chosen such that the "new system" is completely without interactions, at all or at some frequencies. However appealing, this idea is not possible or feasible for all systems. Decoupling may be very sensitive to modeling errors and uncertainties, it may yield a poor disturbance rejection, and it may even in some cases introduce stability problems.

A.8 Estimators

In systems where it is practically difficult to obtain desired measurements of a certain variable, a system model may be used to estimate the variable, by implementing what is referred to as a (state) estimator or (state) observer. An estimator is a system model run in parallel with the system, with the same inputs as the real system. The output of the estimator model will provide an estimate of the system's real states, and thus create an "artificial measurement", which may be used in a feedback control loop. A common estimator structure for a linear time-invariant (LTI) system is shown in figure A.5.

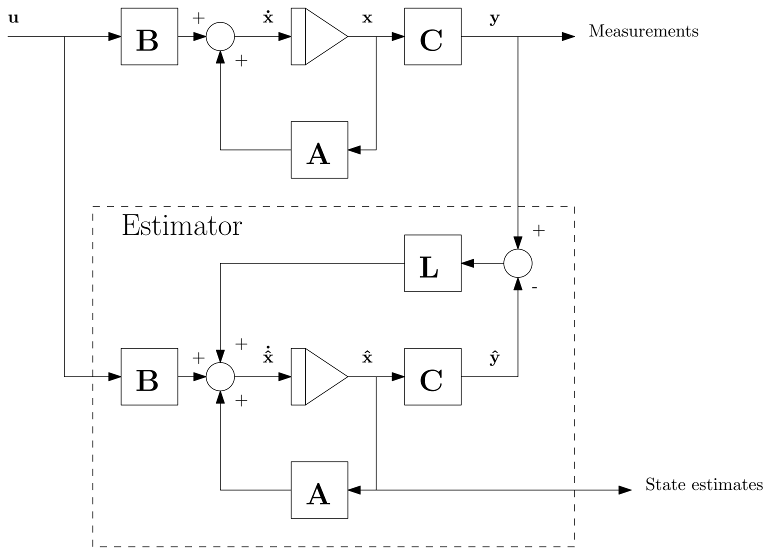


Figure A.5: Estimator for a LTI system

This estimator uses available measurements to correct the state estimates produced by the estimator. (Notice the feedback through the matrix **L**.) There exists theory on how to find the optimal matrix **L**, known as Kalman filtering, see e.g. [Brown 1997].

A.9 Optimal controllers

In section A.3, only the basic P, PD, PI and PID controllers were discussed. However, there are several alternatives. As mentioned in section A.7, methods exist to shape the system to make it easier to control. There are other similar methods, such as loop shaping and closed-loop shaping, where internal model control (IMC) is an analytical approach for the latter. A more signal based approach leads to the linear quadratic Gaussian (LQG) control, the Wiener-Hopf ($\mathcal{H}_2$ norm) design method, and the $\mathcal{H}_\infty$ optimal control. These methods all use optimization to design a feedback controller that is optimal in some sense. Numerical optimization techniques are also used to optimize more directly the true performance objective(s). All these approaches are off-line methods; they produce a predesigned feedback controller that is later implemented in the system. (See [Skogestad 2005] for further description of these methods.) In addition, there exists an on-line optimization approach, known as model predictive control (MPC), discussed in section 2.3.

Appendix B

Optimization

Optimization is an important tool for decision making, also in control theory. This appendix provides an introduction to the concepts of optimization, the formulation and solution of optimization problems, and some specific types of problems and the solution methods involved to solve these. This appendix is based on the litterature [Nocedal 2006, Hillier 2010].

B.1 Basic concepts

Optimization of a system refers to making decisions to maximize a quantitative measurement of performance, an *objective*, possibly subject to some given *constraints*. In optimization, making a decision equals assigning a value to a *decision variable*. The performance objective could be economical profit, production quantity or quality, energy, or anything that can be quantified, depending on the underlying system. The objective must be quantified by a single number, to enable the comparison of two possible solutions (sets of decisions), to determine which one is the better.

In order to find the (optimal) solution to the optimization problem, this needs to be formulated mathematically. The objective, decision variables and constraints must be identified (or defined), and then quantified. This process is known as modeling. (As with the modeling of dynamic systems, described in section A.2, modeling here refers to describing a system mathematically. However, in control theory, the focus is on dynamic behavior, while in optimization theory, the emphasis is on defining the objective, decision variables and constraints.)

When modeled, an optimization problem (or optimization program, as it is often referred to) is solved by some method, called an optimization algorithm. There are many such algorithms, and they are often implemented on computers and solved numerically. The kind of optimization algorithm needed to solve the problem depends on the problem itself. There are many optimization algorithms available, each tailored to a specific type of problems.

A solution may in many cases be shown to be the optimal solution to the problem (i.e. to the model of the real problem) by the use of optimality conditions, a set of mathematical expressions. An optimal solution can either be global or local. A global optimal solution is the best solution in the entire set of feasible points, while a local optimal solution may only the best possible solution among all nearby points. This is illustrated in figure B.1 for a function of one decision variable.

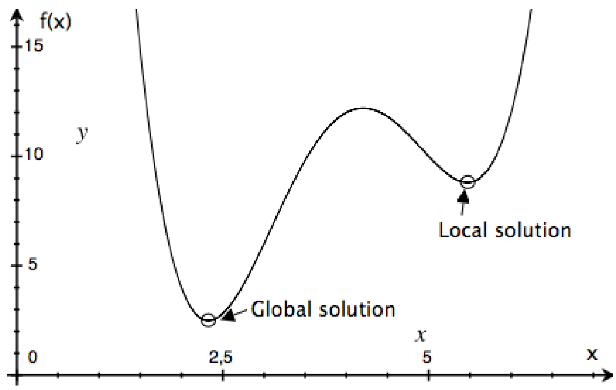


Figure B.1: Local versus global solution

In general, it may be very difficult to locate the global solution, or even determine whether a solution is local or global. Some optimization algorithms seek to find the global solution, while others may terminate at a local solution.

The solution of the problem, and properties of the problem itself, may be analyzed by a set of techniques known as *sensitivity analysis*, which provides information on how the solution would change by changing the model.

Applying the scientific methods of optimization to management problems (such as optimal distribution of resources in an organization) is the foundation for the scientific field of "operations research" (OR). (The term is of military origin, "research of (military) operations.") The term OR is often used synonymously with optimization. However, OR also includes practical issues, like management of organizations, and may be said to have a broader viewpoint than optimization. Modeling and solving optimization problems is still a fundamental part of operations research.

B.2 Mathematical formulation

The three basic components of an optimization problem are the objective, the decision variables and the constraints. The objective must be formulated as a function of the decision variables, relating the values of the decision variables to a measure of performance. This quantitative measure must be a single number to enable the comparison of the performance of two different possible solutions.

The decision variables are representations of some physical quantity. A decision variable could be the amount to invest in a project, the number of one type of items to produce in a production facility, the valve opening in a petroleum production system, and so on.

The decision variables are often subject to some constraints, which may or may not be related to physical limitations. The constraints could e.g. be a non-negativity requirement when the physical quantity represented may not be negative, a maximum limit on a valve opening due to the physical size of the valve, a maximum limit on total investment due to the amount available for investments, and so on.

A general formulation of an optimization problem may look like this:

$$\begin{array}{cc} \max_{x \in \mathbb{R}^n} & f(x) \\ s.t. & \\ c_i(x) = 0, & i \in \mathcal{E} \\ c_i(x) \leq 0, & i \in \mathcal{I} \end{array} \tag{B.2.1}$$

The first line, $\max_{x \in \mathbb{R}^n} \quad f(x)$, indicates that x is a vector of n decision variables, and $f(x)$ is the objective to be *maximized*, as a function of these decision variables. (This could equivalently be formulated as $\min_{x \in \mathbb{R}^n} \quad -f(x)$.) $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively, and $c_i(x)$ are constraint functions of the decsion variables.

There are many different notations, but the general structure of an optimization problem is the same.

B.3 Problem classification

Problems can be classified by their objective function, constraints and decision variables. Some classifications are described below:

Constrained optimization problems is a quite general class of optimization problems with at least one constraint on the decision variables.

Unconstrained optimization problems do not have any constraints on the decision variables. These problems arise naturally in many practical applications.

Integer programming (IP) problems require the decision variables to only take on integer values (e.g. the number of cars produced by one assembly line), or even binary values (typically decisions to do or not to do something). This is also referred to as discrete optimization.

Mixed integer programming (MIP) problems are problems with both continuous and integer (or binary) decision variables.

Stochastic optimization problems are problems using quantification of uncertainties and probabilities in the model.

Deterministic optimization problems are problems where the model is completely known. (As opposed to stochastic problems.)

Convex optimization problems have the important property that any local solution is also the global solution. (Global and local solutions are explained in section B.1.) For a problem to be convex, the objective function must be convex for minimization problems (concave for maximization problems), and in addition, the feasible region (the set of all points that satisfy all constraints) must be convex. (See e.g. [Nocedal 2006] for definitions of convex functions and sets.)

Linear programming (LP) problems have a linear objective function and linear constraints. LP problems are convex problems.

Nonlinear programming (NLP) problems are problems with either a nonlinear objective function, nonlinear constraints, or both.

Quadratic programming (QP) problems are NLP problems with a quadratic objective function and linear constraints.

B.4 Linear Programming (LP)

[Hillier 2010] ranks the development of linear programming "among the most important scientific advances of the mid-20th century, [...] that has saved many thousands or millions of dollars for many companies..." It is widely used for many applications, especially for allocation of limited resources to competing activities in an optimal way.

Linear programming (LP) problems have both a linear objective function and linear constraints. Any LP problem may in general be formulated to fit the following model:

$$
\begin{array}{lrcl}
\max\limits_{x_i} & Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n & & \\
 & \textit{subject to}: & & \\
 & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \leq & b_1 \\
 & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \leq & b_2 \\
 & & \vdots & \\
 & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \leq & b_m \\
 & x_i \geq 0, \quad i = 1...n & &
\end{array}
\tag{B.4.1}
$$

or equivalently the more compact matrix form:

$$
\begin{array}{lrcl}
\max & c^\top x & & \\
\textit{subject to} & & & \\
 & Ax & \leq & b \\
 & x & \geq & 0
\end{array}
\tag{B.4.2}
$$

where x, b and c are vectors, and A is a matrix.

LP problems are convex problems, as a linear objective function is both concave and convex, and linear constraints form a convex set of feasible points.

LP problems are most often solved by the optimization algorithm known as the simplex method. It was developed by George Dantzig in 1947, and is probably the most well known optimization algorithm of them all. It is an algebraic procedure, based on geometric concepts, and there are several free and commercial computer implementations of the algorithm (and variants of it). There are also other algorithms developed to solve LP problems, based on the interior-point approach developed in 1984, or the active-set method. An extensive discussion of the formulation and solution of LP problems may be found in e.g. [Hillier 2010].

In MATLAB, LP problems may be solved using the function `linprog` in the optimization toolbox.

B.5 Quadratic Programming (QP)

The Quadratic programming (QP) problem is similar to the LP problem in the way that all the constraints are required to be linear. However, in the QP model, the objective function is allowed to include quadratic terms, i.e. terms on the form x_i^2 and x_ix_j $(i \neq j)$. The resulting problem is thus a nonlinear programming (NLP) problem. QP problems can be formulated using a model quite similar to (B.4.2):

$$
\begin{array}{lrcl}
\min_x & \frac{1}{2}x^\top Qx + c^\top x & & \\
\textit{subject to} & & & \\
 & Ax & \leq & b \\
 & x & \geq & 0
\end{array}
\tag{B.5.1}
$$

Here, Q and A are matrices, and x, c and b are vectors.

QP problems on this form are convex if $Q \geq 0$ (Q is positive semi-definite).

There are several algorithms to solve convex QP problems, including a modification to the simplex method, active-set methods and interior-point methods. Active-sets methods have been widely used, and are effective for small- and medium-sized problems. Interior-point methods are more recent, and well suited for large problems, but may not be the most effective choice if a series of related problems are to be solved.

In MATLAB, QP problems may be solved using the function `quadprog` in the optimization toolbox.

B.6 Nonlinear Programming (NLP)

Nonlinear programming (NLP) problems are problems with either a nonlinear objective function, nonlinear constraints, or both. NLP problems can be formulated in the general form shown in equation (B.2.1) on page 78.

It may be difficult to prove the convexity of NLP problems, and thus to decide whether an optimal solution is local or global. Most solution algorithms for NLP problems terminate if a local solution is found.

NLP problems may be solved using interior-point, active-set, trust-region, penalty or augmented Lagrangian algorithms, each with their advantages and disadvantages.

The SQP (Sequential Quadratic programming) algorithm approximates the problem (B.2.1) as a QP problem at each iterate, and finds the next iterate as the solution to the QP problem. SQP is considered one of the most effective methods to solve nonlinearly constrained problems. There are both active-set and trust-region SQP algorithms.

In MATLAB, NLP problems may be solved using the function `fmincon`, which has four algorithm options: 'interior-point', 'sqp', 'active-set' or 'trust-region-reflective'.

B.7 Branch and Bound

Integer Programming (IP) problems, where some or all variables are restricted to only take on integer (or binary) values, typically have a finite number of feasible solutions. The intuitive approach to solve an IP problem may therefore be to compare all the feasible solutions, to find the optimal solution. (Often referred to as a 'brute force' approach.) However, the number of possible solutions may be, and often is, very large. Then it is necessary to only investigate a small fraction of those feasible solutions, to find the optimal solution sufficiently fast (or at all). One approach that does this is the branch-and-bound (B&B) technique.

The idea behind B&B is to divide the set of feasible solutions into smaller and smaller sets, until one feasible optimal solution is found within each set. Once such a solution is found, a bound is set, saying that the optimal solution is at least as good as this bound. Using this bound, entire sets of feasible solutions may be disregarded as the optimization search progresses.

B&B, and the application to IP problems, is thoroughly discussed in [Hillier 2010], but the general idea may be illustrated by an example:

Example Consider a problem where only one (of many) decision variables is required to be an integer variable, and name this variable x_I. A B&B approach would be to first solve the problem while considering x_I as a continuous variable. Say the optimal solution found has an objective function value of 5, with $x_I = 3.4$. The next step would be to solve the same problem with added constraints. First, solve the problem with the constraint $x_I \leq 3$ (branch 1), then solve the problem with $x_I \geq 4$ (branch 2). This creates two new branches in the search tree. (Each branch may also be referred to as a node.) Say branch 1 provides a feasible solution with function value 4.5 and with $x_I = 3$. This is a feasible solution, and 4.5 is thus set as a lower bound for the optimal solution. Say branch 2 provides an optimal solution for the subset with function value 4.3 and with $x_I = 5.8$. The value of 4.3 is then an upper bound on any feasible solution in this subset of the original problem. (No better solution may be found by adding constraints.) Then this second branch would be cut off, as 4.3 is less than the lower bound at 4.5. As all branches now are investigated, the optimal solution is found to be 4.5 with $x_I = 3$. $\square$

Appendix C

Alternative pressure model

An alternative model for the pressure in the annulus is derived in this appendix. See also section 3.1.4.

Both the pressure and the density in the annulus will vary with height due to gravity, according to the barometric formula:

$$dp = -\rho g dh \tag{C.1}$$

where a positive dh means a height difference in the opposite direction of the earth's gravity (upwards). The ideal gas law given by equation (3.1.34) may be rewritten as:

$$p = \frac{mRT}{MV} \tag{C.2}$$

Inserting

$$\rho = m/V \tag{C.3}$$

yields

$$p = \rho \frac{RT}{M} \Rightarrow \rho = \frac{pM}{RT} \tag{C.4}$$

Inserting this into equation (C.1) yields

$$\frac{dp}{dh} = -\frac{pMg}{RT} \tag{C.5}$$

Assuming a constant temperature, we then have:

$$\int_{p(h_0)}^{p(h_1)} \frac{dp}{p} = -\frac{Mg}{RT} \int_{h_0}^{h_1} dh \tag{C.6}$$

$$\Rightarrow \ln\left(\frac{p_1}{p_0}\right) = -\frac{MG}{RT}(h_1 - h_0) \tag{C.7}$$

$$\Rightarrow p(h) = p_0 \exp\left(-\frac{Mg}{RT}(h - h_0)\right) \tag{C.8}$$

The according density is found using the linear relationship from equation (C.4) :

$$\Rightarrow \rho(h) = \rho_0 \exp\left(-\frac{Mg}{RT}(h - h_0)\right) \tag{C.9}$$

For the annulus, this becomes:

$$p_{ai} = p_a \exp\left(\frac{L_a M_g g}{RT_a}\right) \tag{C.10}$$

$$\rho_{ai} = \rho_a \exp\left(\frac{L_a M_g g}{RT_a}\right) \tag{C.11}$$

$$\rho_a = \frac{M_g}{RT_a} p_a \tag{C.12}$$

$$\rho_{ai} = \frac{M_g}{RT_a} p_{ai} \tag{C.13}$$

where L_a is the vertical height difference between the gas-lift choke valve and the injection valve. Still more information is needed to find the pressures and densities. Knowing the total mass of gas in the annulus (the state variable m_{ga}), we have

$$m_{ga} = \int_{-L_a}^{0} \rho(h) A_a \, dh \tag{C.14}$$

Inserting equation (C.9) with $h_0 = 0$ defined at the height of the gas-lift choke valve, we have

$$\Rightarrow m_{ga} = -A_a \rho_a \int_{0}^{-L_a} \exp\left(-\frac{M_g g}{RT_a} h\right) dh \tag{C.15}$$

$$\Rightarrow m_{ga} = \frac{A_a RT_a}{M_g g} \left(\exp\left(\frac{L_a M_g g}{RT_a}\right) - 1\right) \rho_a \tag{C.16}$$

Defining

$$C_a = \exp\left(\frac{L_a M_g g}{RT_a}\right) \tag{C.17}$$

the model for the densities and pressures in the annulus then becomes

$$\rho_a = \frac{M_g g}{A_a RT_a (C_a - 1)} m_{ga} \tag{C.18}$$

$$p_a = \frac{g}{A_a (C_a - 1)} m_{ga} \tag{C.19}$$

$$\rho_{ai} = \frac{M_g g C_a}{A_a RT_a (C_a - 1)} m_{ga} \tag{C.20}$$

$$p_{ai} = \frac{g C_a}{A_a (C_a - 1)} m_{ga} \tag{C.21}$$

This model was considered too complex to model these variables, and an approximation was therefore used (see section 3.1.4).

Appendix D

Implementation

The implementation of the model and the optimization performed in this project work is done in MATLAB, and the following sections provide the MATLAB code (scripts and functions).

D.1 Model implementation

The well model developed in chapter 3 is implemented as a MATLAB function, providing the differentiated masses $\dot{m}_{ga}$, $\dot{m}_{gt}$ and $\dot{m}_{lt}$ as functions of the masses m_{ga}, m_{gt} and m_{lt}, the inputs u_{pc} and u_{gl}, and the well parameters. The main part of this MATLAB function is shown in the box below.

```
function [mdot y] = well_model(m,u,well)

% Get parameters, states and inputs
%   (Global variables and from input)
% ...

% Calculate extra parameters
r_glr = (1-r_wc)*r_gor;
rho_l = r_wc*rho_w + (1-r_wc)*rho_o;
nu_l  = 1/rho_l;


% Well model:

% Pressures
p_p = R*T_t*m_gt/(M_g*V_t - M_g*nu_l*m_lt) - g*m_t/2/A_t;
p_ti = p_p + g*m_t/A_t;
a = r_glr*M_g*g*L_w/2/R/T_t;
p_bh = ((1+r_glr+a)*p_ti + rho_l*g*L_w)/(1+r_glr-a);
p_ai = (R*T_a/V_a/M_g + g/2/A_a)*m_ga;

% Densities
```

```
rho_p = rho_l*M_g*p_p*m_t/(rho_l*R*T_t*m_lt + M_g*p_p*m_gt);
rho_gi = M_g/R/T_a*p_ai;

% Mass flows
w_p = C_pc*sqrt(rho_p*max(0,p_p-p_m))*u_pc;
w_lp = m_lt/m_t*w_p;
w_gp = w_p - w_lp;
w_lr = rho_l*Q_max*(1 - (1-C)*(p_bh/p_r) - C*(p_bh/p_r)^2);
w_gr = r_glr*w_lr;
w_gi = C_iv*sqrt(rho_gi*max(0,p_ai-p_ti));
w_gl = u_gl;

% Mass differentials
mdot_ga = w_gl - w_gi;
mdot_gt = w_gr + w_gi - w_gp;
mdot_lt = w_lr - w_lp;

% Return differentials and 'measurements'
mdot = [mdot_ga mdot_gt mdot_lt]';
y = [p_p p_ti p_bh p_ai w_lp w_gp w_lr w_gr w_gi rho_p rho_gi]';
```

D.2 Static model implementation

The MATLAB implementation of the static model is shown in the box below:

```
function m = steady_state(u,well)
    global general_parameters;
    tol = 1e-5;

    % Solving steady-state tubing mass
    mdot_t = @(m)well_model_stable_tubing(m,u,well,general_parameters);
    % The well_model_stable_tubing function is the same as the well model,
    %  but only considering tubing mass, and with w_gi = u_gl
    options = optimset('TolFun',tol,'TypicalX',m0_t);
    m_t = fsolve(mdot_t,m0_t,options);

    % Solving steady-state annulus mass using the bisection method
    mdot_a = @(m_a)well_model_stable_annulus(...
        m_a,m_t,u,well,general_parameters);
    % The well_model_stable_annulus function is the same as the well model,
    %  but only considering mass in the annulus, and with mass in the
    %  tubing known

    % Initialize search region
    % a0 is the low estimate, a2 is the high estimate, a1 is the average
    a0 = 0;
    a2 = m0_a;
    % Ensuring the high estimate has a decreasing derivative:
    while mdot_a(a2) > 0
        a0 = a2;
        a2 = 2*a2;
    end
    a1 = (a0 + a2)/2;

    % Bisection method
    ma = mdot_a(a1);
    while abs(ma) > tol
        if ma > 0
            a0 = a1;
            a1 = (a0 + a2)/2;
        else
            a2 = a1;
            a1 = (a0 + a2)/2;
        end
        ma = mdot_a(a1);
    end
    m_a = a1;

    % Return steady-state masses
    m = [m_a; m_t];
end

function mdot_t = well_model_stable_tubing(m,u,well,general_parameters)
```

```
    % The well_model_stable_tubing function is the same as the well model,
    %  but only considering tubing mass, and with w_gi = u_gl
% ...
mdot_t = [mdot_gt mdot_lt]';
end

function mdot_a = well_model_stable_annulus(...
    m_a,m_t,u,well,general_parameters)
    % The well_model_stable_annulus function is the same as the well model,
    %  but only considering mass in the annulus, and with the mass in the
    %  tubing known
% ...
mdot_a = mdot_ga;
end
```

D.3 Stable regions

The script to find estimates of stable regions for the wells is shown below.

find_stable_regions.m:

```
% Function parameters
tol = 0.05;     % Tolerance in estimation of casing heading
pc_step = 0.1;  % Step size for production choke
max_pc = 1;
well_numbers = 1:7; % Considered wells [1..7]
number_of_wells = length(well_numbers);

% Simulation parameters
dt = 0.5;  % Time steps, seconds
hours = 4; % Total simulation time
N = 3600*hours/dt; % Number of time steps

% Initialization
global well_parameters;
init_parameters;
data = [];

for well_number = well_numbers

    output = [0 0];
    well = well_parameters(:,well_number);

    u_gl = 0.05;
    index = 0;
    % Try all different production choke openings
    for u_pc = pc_step:pc_step:max_pc
        progress = ((well_number-1)*100 + (u_pc-pc_step)*100/max_pc)...
            /number_of_wells;
        disp(['Progress: ' int2str(progress) ' percent']);

        h = u_gl + 0.2;
        l = max(0,u_gl-0.05);
        u_gl = (l+h)/2;

        % Find gas-lift rate at stability limit
        while (h-l) > tol
            index = index + 1;
            u = [u_pc u_gl]';

            % Find steady state
            m0 = steady_state(u,well);
            % Simulate well while 'measuring' liquid production
            m = m0;
            w_lp = zeros(1,N);
            for i = 1:N
                [mdot measurements] = well_model(m,u,well);
```

```
                w_lp(i) = measurements(5);
                m = m + mdot*dt; %Forward Euler integration
            end

            % Cut first part of the simulations
            lp = w_lp(floor(end/100):end);

            % Test: change in variations over time
            lp1 = lp(1:floor(end/2));        % First half
            lp2 = lp(floor(end/2)+1:end);    % Second half
            max1 = max(lp1);
            min1 = min(lp1);
            max2 = max(lp2);
            min2 = min(lp2);

            % Test here
            % Stable requirement: variations in the second half should be
            % less than 90% of the variations in the first half of the
            % simulations
            if (max2-min2) > 0.9*(max1-min1)
                disp('Not (sufficiently) decreasing variations')
                % Unstable operation detected, increase gas-lift
                l = u_gl;
            else
                disp('Stable')
                % Stable operation detected, reduce gas-lift
                h = u_gl;
            end
            u_gl = (h+l)/2;
        end
        output = [output; u_pc u_gl];
    end
    data = [data output];
end
```

D.4 Optimization free strategy

This section provides the MATLAB functions used to find the routing and input settings without numerical optimization.

Routing:

```
function routing = base_case_routing(well_numbers)
% Function to find the best routing based on potential production
%   from each well, calculated based on the well parameters r_wc, r_gor and
%   Q_max



%well parameters
n = length(well_numbers);
global well_parameters
init_parameters
parameters = well_parameters(4:6,well_numbers);
% r_wc  = [  0.4     0.7     0.2     0.5     0.6     0.8     0.3  ];
% r_gor = [  0.08    0.07    0.07    0.09    0.06    0.08    0.10 ];
% Q_max = [  0.025   0.050   0.035   0.100   0.020   0.015   0.005];

% Brute force minimization search
min = Inf;
rout = gen_rout_mat(n);
for i  = 1:size(rout,1)
    routing = rout(i,:);
    prod = zeros(4,2);
    for j = 1:n
        wc = parameters(1,j);
        gor = parameters(2,j);
        qmax = parameters(3,j);
        % Oil
        o = qmax*(1-wc);
        prod(1,routing(j)+1) = prod(1,routing(j)+1) + o;
        % Water
        w = qmax*wc;
        prod(2,routing(j)+1) = prod(2,routing(j)+1) + w;
        % Liquid
        prod(3,routing(j)+1) = prod(3,routing(j)+1) + qmax;
        % Gas
        prod(4,routing(j)+1) = prod(4,routing(j)+1) + 10*gor*o;
    end
    val = sum(abs(prod(:,1)-prod(:,2)));
    if val < min
        min = val;
        bestr = routing;
        bestprod = prod;
    end
end
routing = bestr;
```

Input settings:

```
function [z,fval] = opt_free_solution(well_numbers,routing)
% Function to decide routing and inputs without using optimization

global constraints

n = length(well_numbers);
n1 = sum(routing);
n0 = n - n1;

% Get constraints
G = constraints(1);
L = constraints(2);
W = constraints(3);
GL = constraints(4)*0.8247/86400;
% Divide the liftgas equally between the two flowlines
GL0 = GL/2;
GL1 = GL0;

% Set stability region limits:
aa = [0.19  0.00  0.00  0.00  0.36  0.63  0.20]';
aa = aa(well_numbers);

% Distribute the liftgas equally between the wells in each flowline
z = ones(2*n,1);
for i = 1:n
    if routing(i)
        z(2*i) = GL/2/n1;
    else
        z(2*i) = GL/2/n0;
    end
end

% Get production constraints values
nonlcon = ss_nonl_con(z,well_numbers,routing);
nlcon0 = nonlcon(1:3);
nlcon1 = nonlcon(4:6);

% Stability check (casing heading)
stable = true;
for i = 1:n
    if z(2*i) < aa(i) && z(2*i) > 0
        stable = false;
    end
end

nogl = zeros(n,1);
n_nogl0 = 0;
n_nogl1 = 0;
stable = false;
while  ~stable || sum(nlcon0<=0) < 3 || sum(nlcon1<=0) < 3
```

```
% 1. Find stable inputs

while ~stable
    max_ind0 = 0;
    max_aa0 = 0;
    max_ind1 = 0;
    max_aa1 = 0;
    for i = 1:n
        if aa(i) > max_aa0 && nogl(i) < 1 ...
                && ~routing(i) && z(2*i) < aa(i)
            max_ind0 = i;
            max_aa0 = aa(i);
        elseif aa(i) > max_aa1 && nogl(i) < 1 ...
                && routing(i) && z(2*i) < aa(i)
            max_ind1 = i;
            max_aa1 = aa(i);
        end
    end
    i0 = max_ind0;
    i1 = max_ind1;
    if i0 > 0
        i = i0;
        nogl(i) = 1;
        z(2*i) = 0;
        aa(i) = 0;
        n_nogl0 = n_nogl0 + 1;
    end
    if i1 > 0
        i = i1;
        nogl(i) = 1;
        z(2*i) = 0;
        aa(i) = 0;
        n_nogl1 = n_nogl1 + 1;
    end

    % Redistribute liftgas
    gl0 = GL0/(n0-n_nogl0);
    gl1 = GL1/(n1-n_nogl1);
    stable = true;
    for i = 1:n
        if ~nogl(i)
            if routing(i)
                z(2*i) = gl1;
            else
                z(2*i) = gl0;
            end
        end
        % Stability check
        if z(2*i) < aa(i) && z(2*i) > 0
            stable = false;
        end
    end
end
```

```
    % 2. Check production limits for each flowline, reduce if necessary
    nonlcon = ss_nonl_con(z,well_numbers,routing);
    nlcon0 = nonlcon(1:3);
    nlcon1 = nonlcon(4:6);
    if sum(nlcon0<=0) < 3
        GL0 = GL0 - 0.01*n0;
    end
    if sum(nlcon1<=0) < 3
        GL1 = GL1 - 0.01*n1;
    end

end

fval = -ss_obj_fun(z,well_numbers);
```

D.5 Static optimization

The MATLAB code for the static optimization solution approach implemented is given below.

Test script:

```
% Test script to find optimal routing and inputs

clc
tic_start = tic;
init_parameters;
init_ss_data;

% Set considered wells (index vector, e.g. [1 3 4 6])
well_numbers = 1:7;
n = length(well_numbers);

% Set constraints
constraints = [500000 30000 15000 500000]'*n/7; %[gas liquid water liftgas]
available_liftgas = constraints(4)*0.8247/86400; % Converting Sm3/h to kg/s

%Routing
bestf = 0;
bestr = [];
bin = gen_rout_mat(n);
% (gen_rout_mat returns a binary matrix containing all possible routing
% combinations)
fvals = zeros(size(bin,1),2);
fvals(:,1) = 1:size(bin,1);
act_con = zeros(size(bin,1),6);

% Starting point for fmincon
z0 = zeros(2*n,1);
z1 = z0;
z2 = z0;
z0(1:2:2*n) = 0.1;
z1(1:2:2*n) = 0.5;
z2(1:2:2*n) = 1;
z2(2:2:2*n) = available_liftgas/n;

fval_bb = 0;
% Try all different routings
for i = 27:28;%1:size(bin,1)
    tic
    routing = bin(i,:);

    % Solve optimization problem using branch and bound
    % Starting point 1 of 3
    [z_bb0,f0] = ...
        ss_optimization_bb(well_numbers,constraints,z0,routing,-fval_bb);
    disp(['fval: ' num2str(f0)])
```

```
    if f0 > fval_bb
        fval_bb = f0;
        z_bb = z_bb0;
        bestr = routing;
    end
    % Starting point 2 of 3
    [z_bb1,f1] = ...
        ss_optimization_bb(well_numbers,constraints,z1,routing,-fval_bb);
    disp(['fval: ' num2str(f1)])
    if f1 > fval_bb
        fval_bb = f1;
        z_bb = z_bb1;
        bestr = routing;
    end
    % Starting point 3 of 3
    [z_bb2,f2] = ...
        ss_optimization_bb(well_numbers,constraints,z2,routing,-fval_bb);
    disp(['fval: ' num2str(f2)])
    if f2 > fval_bb
        fval_bb = f2;
        z_bb = z_bb2;
        bestr = routing;
    end

    fvals(i,2) = max([f0 f1 f2]);
    toc
end
routing = bestr;
```

ss_optimization_bb.m:

```
function [z,fval] = ss_optimization_bb(indices,constraints,z0,routing,UB)
% Steady-state optimization to find optimal inputs with stability regions

% Gas-lift constraint converted from Sm3/d to kg/s
available_liftgas = constraints(1)*0.8247/86400;

% Get well data
n = length(indices); % Number of wells

% Construct bounds on inputs
lb = zeros(2*n,1);
ub = zeros(2*n,1);
ub(1:2:2*n) = 1;
ub(2:2:2*n) = available_liftgas;

%   Set sum gaslift <= available gaslift, Ax<=B
A = zeros(1,2*n);
A(2:2:2*n) = 1;
B = available_liftgas;
```

```
% Setting up branch conditions:
%   a*u_pc - u_gl <= -b (A1,B1)
%       OR
%   u_gl = 0 (A2,B2)
b =  [0.00  0.00  0.00  0.00  0.00  0.33  0.15]';
aa = [0.19  0.00  0.00  0.00  0.36  0.63  0.20]';
a = aa-b;
a = a(indices);
b = b(indices);
A1 = zeros(n,2*n);
A2 = zeros(n,2*n);
for i = 1:n
    A1(i,2*i-1) = a(i);
    A1(i,2*i) = -1;
    A2(i,2*i) = 1;
end
B1 = -b;
B2 = zeros(n,1);

% Optimization

% Create objective function handle
obj_fun = @(z)ss_obj_fun(z,indices);

% Set nonlinear constraints (water/gas production capacities)
nlcon = @(z)ss_nonl_con(z,indices,constraints,routing);

% Set options for branch and bound algorithm
opt = optimset('Display','notify','MaxFunEvals',15000,...
   'Algorithm','sqp','TolFun',1e-5,'TolCon',1e-5);

% Initialize iteration count
iteration = 0;
% Run branch and bound
[z,fval] = bbound(obj_fun,z0,A,B,A1,B1,A2,B2,lb,ub,nlcon,opt,UB);

fval = -fval;

end
```

bbound.m:

```
function [z,fval] = ...
    bbound(obj_fun,z0,A,B,A1,B1,A2,B2,lb,ub,nonlcon,options,UB)

%Recursive function to solve the optimization problem:
% min obj_fun(x)
% s.t.
%
% lb <= x <= ub
```

```
% A*x <= B
% a1*x <= b1 or a2*x <= b2
% (where a1 and a2 are rows of A1 and A2,
%    and b1 and b2 are according elements in B1 and B2.)
% c(x) <= 0
% ceq(x) = 0

% Branch and bound
    % Solve opt problem
    % if not all constraints satisfied
        % Find greatest deviance
        % Branch on this:
            % Branch 1: Solve with A1(i)x <= B1(i)
                % If not all constraints satisfied
                    % RECURSIVE
                % Else
                    % Return solution
            % Branch 2: Solve with A2(i)x <= B2(i)
                % If not all constraints satisfied
                    % RECURSIVE
                % Else
                    % Return solution
            % Compare solutions, return best feasible
    % else
        % Return solution

    % Iteration counter
    global iteration;
    display = false;
    iteration = iteration +1;
    this_iteration = iteration;
    disp(['Node: ' int2str(iteration)]);

    % Solve optimization problem
    [z,fval,exitflag] = ...
        fmincon(obj_fun,z0,A,B,[],[],lb,ub,nonlcon,options);
    if display
        disp(['fval in this node: ' num2str(fval)]);
    end

    if exitflag > 0 % A feasible solution was found
        % Validate solution
        satisfied = true;
        nsat = [];
        n = length(B1);
        for i = 1:n
            if A1(i,:)*z > B1(i) && A2(i,:)*z > B2(i)
                satisfied = false;
                nsat = [nsat; i];
            end
        end

        if ~satisfied
            if display
```

```
            disp('Solution at ')
            z
            fval
            disp(['violates constraint(s) ' int2str(nsat')]);
            disp('A1: ');disp(A1);
            disp('B1: ');disp(B1);
        end

        if fval < UB
            % Find greatest deviance
            maxdev = 0;
            for i = 1:length(nsat)
                j = nsat(i);
                dev = min(abs(A1(j,:)*z-B1(j)),abs(A2(j,:)*z-B2(j)));
                if dev > maxdev
                    maxdev = dev;
                    maxind = j;
                end
            end
            if display
                disp(['Branching on constraint ' int2str(maxind)]);
            end

            % Branch on constraint with index maxind
            i = maxind;
            a1 = A1(i,:);
            b1 = B1(i);
            a2 = A2(i,:);
            b2 = B2(i);
            all_exc_maxind = [1:(i-1),(i+1):n];
            A1new = A1(all_exc_maxind,:);
            B1new = B1(all_exc_maxind);
            A2new = A2(all_exc_maxind,:);
            B2new = B2(all_exc_maxind);
            Anew1 = [A;a1];
            Bnew1 = [B;b1];
            Anew2 = [A;a2];
            Bnew2 = [B;b2];
            % Branching here (recursive)
            if abs(A1(i,:)*z-B1(i))>abs(A2(i,:)*z-B2(i))
                [z2 fval2] = bbound(obj_fun,z0,...
                    Anew2,Bnew2,A1new,B1new,A2new,B2new,...
                    lb,ub,nonlcon,options,UB);
                [z1 fval1] = bbound(obj_fun,z0,...
                    Anew1,Bnew1,A1new,B1new,A2new,B2new,...
                    lb,ub,nonlcon,options,fval2);
            else
                [z1 fval1] = bbound(obj_fun,z0,...
                    Anew1,Bnew1,A1new,B1new,A2new,B2new,...
                    lb,ub,nonlcon,options,UB);
                [z2 fval2] = bbound(obj_fun,z0,...
                    Anew2,Bnew2,A1new,B1new,A2new,B2new,...
                    lb,ub,nonlcon,options,fval1);
            end
```

```
                % Comparing solutions, returning the best
                if display
                    disp(['Branching in node '...
                        int2str(this_iteration) ' gave:'])
                    fval1
                    fval2
                end
                if fval1 < fval2
                    fval = fval1;
                    z = z1;
                else
                    fval = fval2;
                    z = z2;
                end
            else
                if display
                    disp('fval > UB');
                end
                % Not a feasible solution
                fval = 0;
            end
        else
            if display
                disp('Feasible solution at: ');
                z
                fval
            end
        end
    else
        if display
            disp(['No feasible solution found. Exitflag: '...
                int2str(exitflag)]);
        end
        fval = 0;
    end

end
```

ss_obj_fun.m:

```
function obj = ss_obj_fun(z,well_numbers)
    global well_parameters ss_data;
    wells = well_parameters(:,well_numbers);

    % Initialize return value
    obj = 0;

    % Calculate total oil production
    for i = 1:length(z)/2
        wn = well_numbers(i);
```

```
        u = z(2*i-[1 0]);
        well = wells(:,i);
        ssd.u_pc = ss_data.u_pc;
        ssd.u_gl = ss_data.u_gl;
        ssd.m_ga = ss_data.m_ga(:,:,wn);
        ssd.m_gt = ss_data.m_gt(:,:,wn);
        ssd.m_lt = ss_data.m_lt(:,:,wn);
        m_ss = ss_fun(u,ssd); % Using precalculated data
%         m_ss = steady_state(u,well); % Using static model
        [~, y] = well_model(m_ss,u,well);
        w_lp = y(5);
        r_wc = well(4);
        % Update return value
        obj = obj + (1-r_wc)*w_lp;
    end
    obj = -obj;
end
```

ss_nonl_con.m:

```
function [c ceq] = ss_nonl_con(z,well_numbers,constraints,routing)

    % Production constraints must be given in bpd and Sm3/day

    global general_parameters well_parameters ss_data
    rho_o = general_parameters(7);
    rho_w = general_parameters(8);
    wells = well_parameters(:,well_numbers);

    if length(routing) ~= length(well_numbers);
        routing = zeros(length(well_numbers),1);
    end

    % Initialize return values
    gas0 = 0;
    gas1 = 0;
    liquid0 = 0;
    liquid1 = 0;
    water0 = 0;
    water1 = 0;

    % Calculate total production
    for i = 1:length(z)/2
        wn = well_numbers(i);
        u = z(2*i-[1 0]);
        well = wells(:,i);
        ssd.u_pc = ss_data.u_pc;
        ssd.u_gl = ss_data.u_gl;
        ssd.m_ga = ss_data.m_ga(:,:,wn);
        ssd.m_gt = ss_data.m_gt(:,:,wn);
```

```
        ssd.m_lt = ss_data.m_lt(:,:,wn);
        m_ss = ss_fun(u,ssd); % Using precalculated data
%         m_ss = steady_state(u,well); % Using static model
        [~, y] = well_model(m_ss,u,well);
        w_lp = y(5);
        w_gp = y(6);
        r_wc = well(4);

        % Liquid in bpd
        liq = (r_wc*w_lp/rho_w + (1-r_wc)*w_lp/rho_o)/0.158987295*86400;
        % Gas in Sm3/day
        gas = w_gp/0.8247*86400;
        % Water in bpd
        wat = r_wc*w_lp/rho_w/0.158987295*86400;
        % Update sums
        if routing(i)
            gas1 = gas1 + gas;
            liquid1 = liquid1 + liq;
            water1 = water1 + wat;
        else
            water0 = water0 + wat;
            liquid0 = liquid0 + liq;
            gas0 = gas0 + gas;
        end
    end

    ceq = [];
    g0 = gas0 - constraints(1);
    l0 = liquid0 - constraints(2);
    w0 = water0 - constraints(3);
    g1 = gas1 - constraints(1);
    l1 = liquid1 - constraints(2);
    w1 = water1 - constraints(3);

    c = [g0 l0 w0 g1 l1 w1]';

end
```

D.6 Dynamic optimization

This section provides the MATLAB code for the dynamic optimization.

Test script:

```
% Test script used to calculate well inputs using the different dynamic
%  optimization methods

clc
global well_parameters general_parameters constraints
init_parameters;

% Load precalculated scenario data
scenario = 'sce2';
sce_ss = [scenario '_ss'];
sce_dyn  = [scenario '_dyn1'];
s = load(sce_ss);
m0 = s.m0;
zs = s.z;
rs = s.routing;
tis = s.ti;

% Create data for objective and constraint functions
wn = [];
z_end = [];
for i = 1:7
    if zs(2*i-1) > 0
        wn = [wn i];
        z_end = [z_end; zs(2*i-1:2*i)];
    end
end
nw = length(wn);
ti = [0 5 10 15 30 45 60 75 90 105 120];
ni = length(ti)-1;
% Initial value for optimization algorithm
z0 = [];
for i = 1:ni
    z0 = [z0; z_end];
end

constr = constraints;
routing = rs;

% Find steady-state masses
m_ss = [];
for i = 1:nw
    u = z0(2*i-1:2*i);
    well = well_parameters(:,wn(i));
    m_ss = [m_ss steady_state(u,well)];
end
```

```
% Set bounds on inputs
available_liftgas = constraints(4)*0.8247/86400; % Converting to kg/s
lb = zeros(2*nw*ni,1);
ub = ones(2*nw*ni,1);
ub(2:2:end) = available_liftgas;

% Set sum gaslift <= available gaslift, Ax<=B
A1 = zeros(ni,2*nw*ni);
for i = 1:ni
    A1(i,2*(i-1)*nw+2:2:2*i*nw) = 1;
end
B1 = ones(ni,1)*available_liftgas;
% Set u_gl <= 10*u_pc (Using a lot of gas-lift )
A2 = zeros(ni*nw,2*nw*ni);
for i = 1:ni*nw
    A2(i,2*i-1:2*i) = [-10 1];
end
B2 = zeros(ni*nw,1);
A = [A1; A2];
B = [B1; B2];


% Create function handles for objective function and nonlinear constraints
% obj_fun = @(z)dyn_obj_fun(z,wn,ti,m0,well_parameters);
obj_fun = @(z)dyn_obj_fun2(z,wn,ti,m0,m_ss,well_parameters);
% Method 1:
nonl_con = @(z)dyn_nonl_con(z,wn,routing,ti,m0,m_ss,...
    general_parameters,well_parameters,constr);

% Set options
options = optimset(...
    'Display','iter',...
    'Algorithm','sqp',...
    'TolFun',1e-3,...
    'TolCon',1e-4,...
    'MaxFunEvals',20000);

% Run fmincon
tic
disp('Running fmincon...');
% Method 1:
[z_dyn,fval] = fmincon(obj_fun,z0,A,B,[],[],lb,ub,nonl_con,options);
% Method 2 and 3:
[z_dyn,fval] = fmincon(obj_fun,z0,A,B,[],[],lb,ub,[],options);

toc

% Restructure before saving
hrs = ti(end)/60;
z = zeros(14,length(ti)-1);
for i = 1:nw
    z(2*wn(i)-1,:) = z_dyn(2*i-1:2*nw:end)';
    z(2*wn(i),:) = z_dyn(2*i:2*nw:end)';
end
```

```
z = [z zs];

r = zeros(7,length(ti));
for i = 1:7
    r(i,:) = routing(i);
end
routing = r;

save(sce_dyn,'m0','z','routing','ti','hrs')
```

Objective function:

```
function fval = dyn_obj_fun(z,wn,ti,m0,m_ss,well_parameters)
% Objective function based on simulation

% Input variables:
% z  - decision variables, inputs for considered wells for each time
%       interval
%       z = [z1; z2; z3; ... zn;]
% wn - well numbers in consideration, e.g. [1 2 4 5 7]
% ti - time intervals for inputs in minutes, last element is endtime
%       e.g. [0 60 120 180]
%       minimum start and end time
% m0 - initial states for simulation, [3x7] matrix
% m_ss - Steady-state masses for the system with static optimization
%       optimal inputs
% well_parameters - from init_parameters (to avoid global variable)


par = well_parameters(:,wn);
dt = 20; % Simulation time steps, seconds

% Simulate
fval1 = 0;
fval2 = 0;
fval3 = 0;
iter = 1;
t = 0;
t_end = 60*ti(end);
in = 1;
next_int = 60*ti(in+1);
u = z(2*(in-1)*length(wn)+1:2*in*length(wn));
m = m0(:,wn);
while t < t_end
    iter = iter+1;
    t = t + dt;

    % Simulate all wells one time step
    for w = 1:length(wn);
        well = par(:,w);
```

```
        mw = m(:,w);
        uw = u(2*w-1:2*w);
        [mdot, meas] = well_model(mw,uw,well);
        op = (1-par(4,w))*meas(5);
        m(:,w) = max(0,m(:,w) + mdot*dt);

        % Update objective function value:

        % Method 1:
        fval1 = fval1 - op;
        % Method 2:
        fval2 = fval2 + sum(((m(:,w)-m_ss(:,w))./m_ss(:,w)).^2);
        % Method 3:
        fval3 = fval3...
            + exp(2*t/t_end-2)*sum(((m(:,w)-m_ss(:,w))./m_ss(:,w)).^2)...
            - 0.0003*exp(-10*t/t_end)*op;
    end

    % Check time interval, change inputs
    if t >= next_int
        in = in + 1;
        if in < length(ti)
            next_int = 60*ti(in+1);
            u = z(2*(in-1)*length(wn)+1:2*in*length(wn));
        end
    end
end

% Select method
fval = fval1;
```

Nonlinear constraints function:

```
function [c ceq] = dyn_nonl_con(z,wn,routing,ti,m0,m_ss,...
    general_parameters,well_parameters,constr)
% Nonlinear constraint function based on simulation

% Input variables:
% z - decision variables, inputs for considered wells for each time
%       interval
%       z = [z1; z2; z3; ... zn;]
% wn - well numbers in consideration, e.g. [1 2 4 5 7]
% routing - routing of the wells to flowline 0 or 1
%       e.g. [0 1 1 0 0]
% ti - time intervals for inputs in minutes, last element is endtime
%       e.g. [0 60 120 180]
%       minimum start and end time
% m0 - initial states for simulation, [3x7] matrix
% m_ss - steady-state solution at end, [3x(length(nw))] matrix
% constr - system constraints [gas liquid water gaslift]
```

```
rho_o = general_parameters(7);
rho_w = general_parameters(8);

par = well_parameters(:,wn);
dt = 20; % Simulation time steps, seconds
nw = length(wn);

% Simulate
iter = 1;
niter = 60*ti(end)/dt+1;
t = 0;
in = 1;
next_int = 60*ti(in+1);
u = z(2*(in-1)*nw+1:2*in*nw);
mm = zeros(3,nw,niter);
mm(:,:,iter) = m0(:,wn);
yy = zeros(11,nw,niter);
while t < 60*ti(end)
    iter = iter+1;

    % Simulation, one time step
    for w = 1:nw;
        well = par(:,w);
        mw = mm(:,w,iter-1);
        uw = u(2*w-1:2*w);
        [mdot meas] = well_model(mw,uw,well);
        yy(:,w,iter) = meas;
        mm(:,w,iter) = mm(:,w,iter-1) + mdot*dt;
    end
    t = t + dt;

    % Update time interval and inputs
    if t >= next_int
        in = in + 1;
        if in < length(ti)
            next_int = 60*ti(in+1);
            u = z(2*(in-1)*nw+1:2*in*nw);
        end
    end
end


% Overall production constraints

% Get measurements and wc parameter
w_gp = []; w_gp(:,:) = yy(6,:,:);
w_lp = []; w_lp(:,:) = yy(5,:,:);
r_wc = par(4,:);
% Initialize
gp0 = zeros(1,niter);
lp0 = gp0;
wp0 = gp0;
gp1 = gp0;
```

```
lp1 = gp0;
wp1 = gp0;
% Calculate production from each well
for w = 1:nw
    % Gas in Sm3/day
    gpw = w_gp(w,:)/0.8247*86400;
    % Liquid in bpd
    lpw = (r_wc(w)*w_lp(w,:)/rho_w + (1-r_wc(w))*w_lp(w,:)/rho_o)...
        /0.158987295*86400;
    % Water in bpd
    wpw = r_wc(w)*w_lp(w,:)/rho_w/0.158987295*86400;
    % Summarize for each flowline
    if routing(w)
        gp1 = gp1 + gpw;
        lp1 = lp1 + lpw;
        wp1 = wp1 + wpw;
    else
        gp0 = gp0 + gpw;
        lp0 = lp0 + lpw;
        wp0 = wp0 + wpw;
    end
end
% Set constraints for max and mean production
tol = 10; % [Percent] tolerance for max production
cp = [...
    max(gp0) - (1+tol/100)*constr(1);
    max(lp0) - (1+tol/100)*constr(2);
    max(wp0) - (1+tol/100)*constr(3);
    max(gp1) - (1+tol/100)*constr(1);
    max(lp1) - (1+tol/100)*constr(2);
    max(wp1) - (1+tol/100)*constr(3);
    mean(gp0) - constr(1);
    mean(lp0) - constr(2);
    mean(wp0) - constr(3);
    mean(gp1) - constr(1);
    mean(lp1) - constr(2);
    mean(wp1) - constr(3);
    ];
c = 1e-9*cp;

% c = [];


% End constraint, m_end = m_ss
ceq = [];
for i = 1:nw
    ss = m_ss(3*i-2:3*i)';
    m_end = [];
    m_end(:,:) = mm(:,i,niter);
    ceq = [ceq; 1e-3*(m_end - ss)./ss];
end
```

Appendix E

Simulations of the scenarios

This appendix presents simulations of the test scenarios using the different optimization methods.

E.1 Scenario 0

Simulations of scenario 0 using two different dynamic production optimization methods:

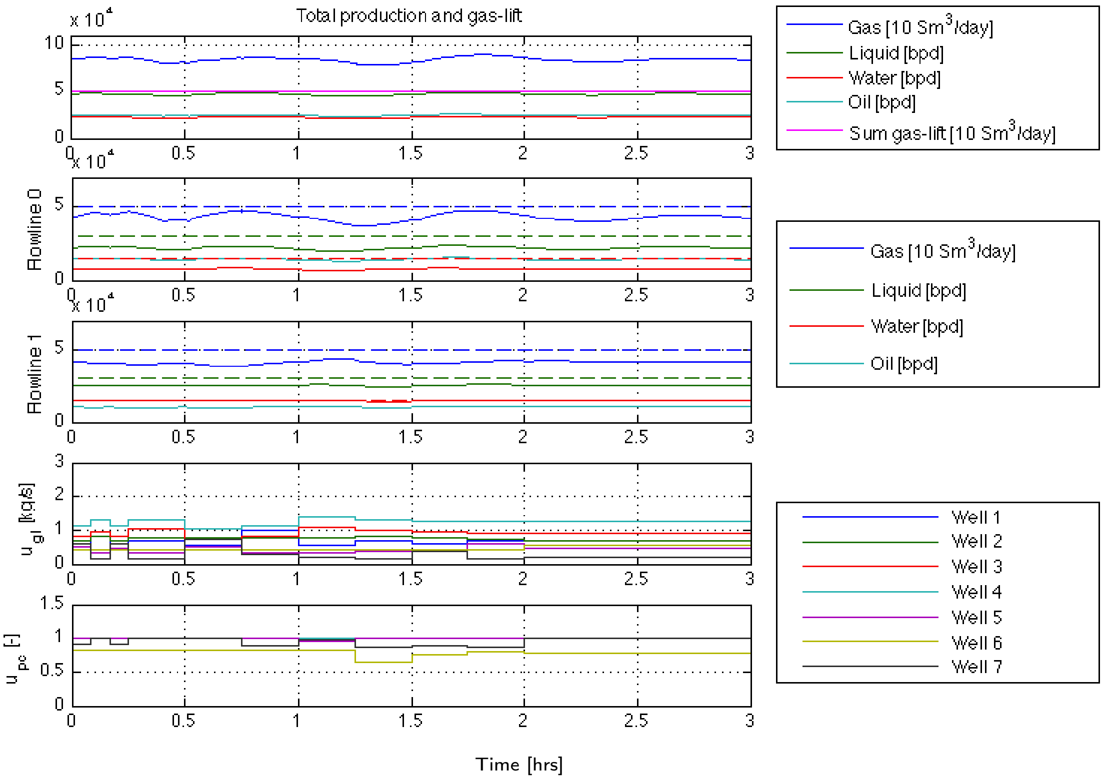


Figure E.1: Simulation of scenario 0 using dynamic otimization method 1

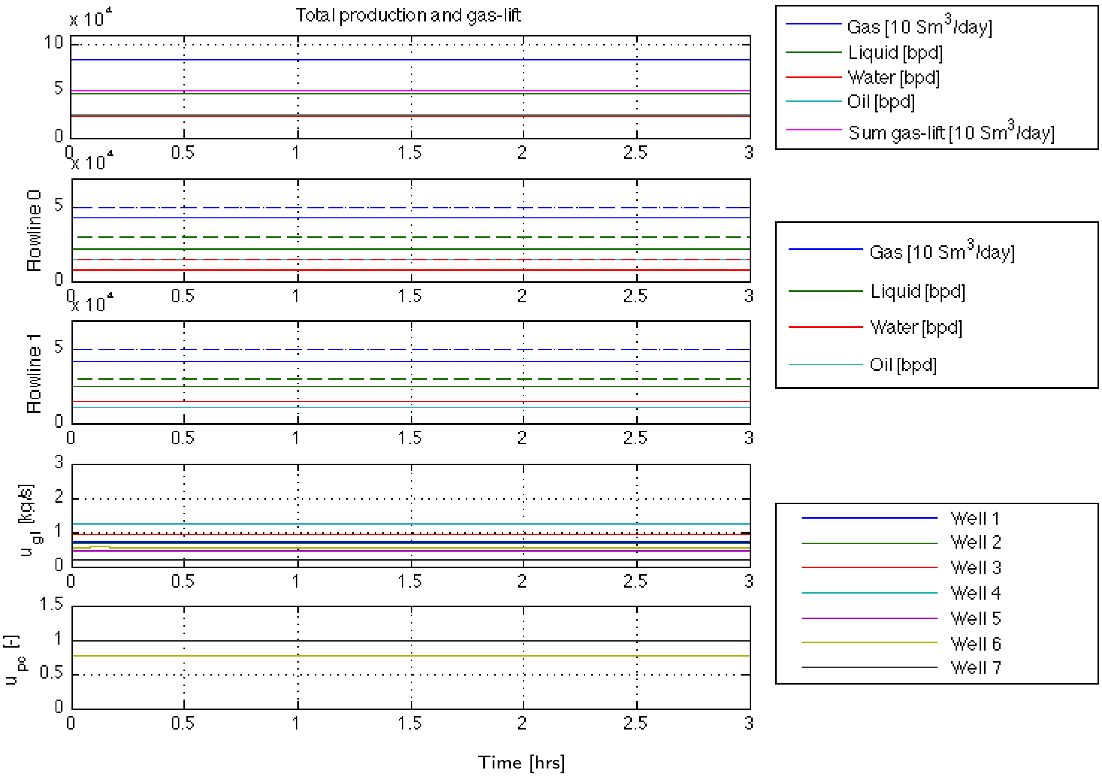


Figure E.2: Simulation of scenario 0 using dynamic otimization method 3

E.2 Scenario 1

Simulations of scenario 1 using the different production optimization methods:

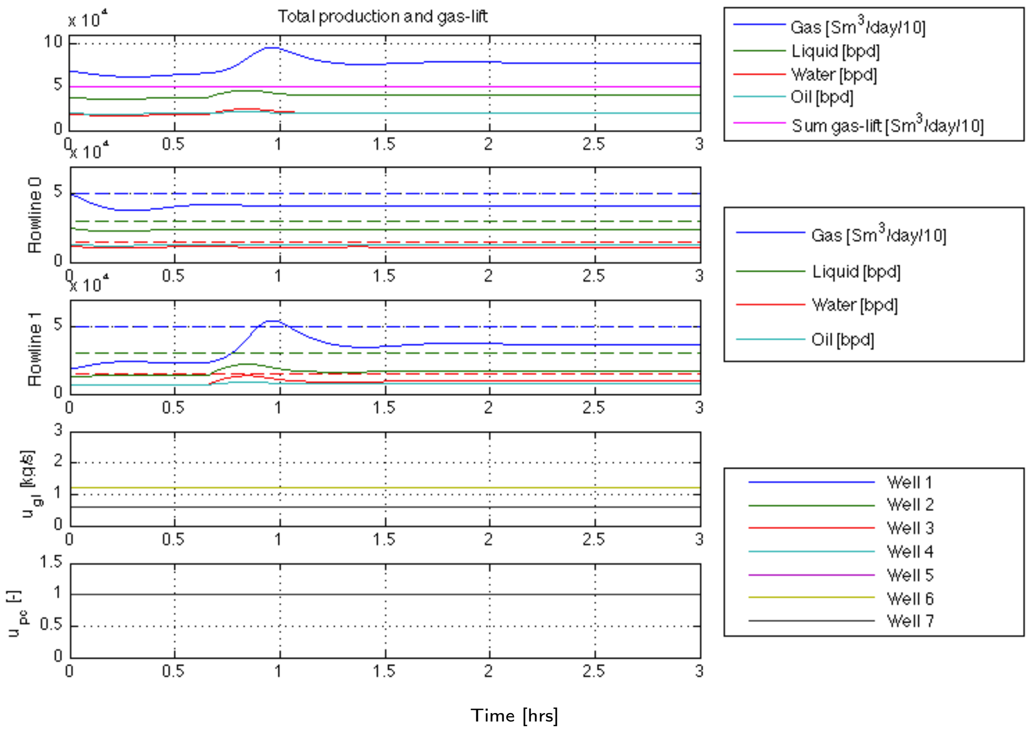


Figure E.3: Simulation of scenario 1 using optimization free method

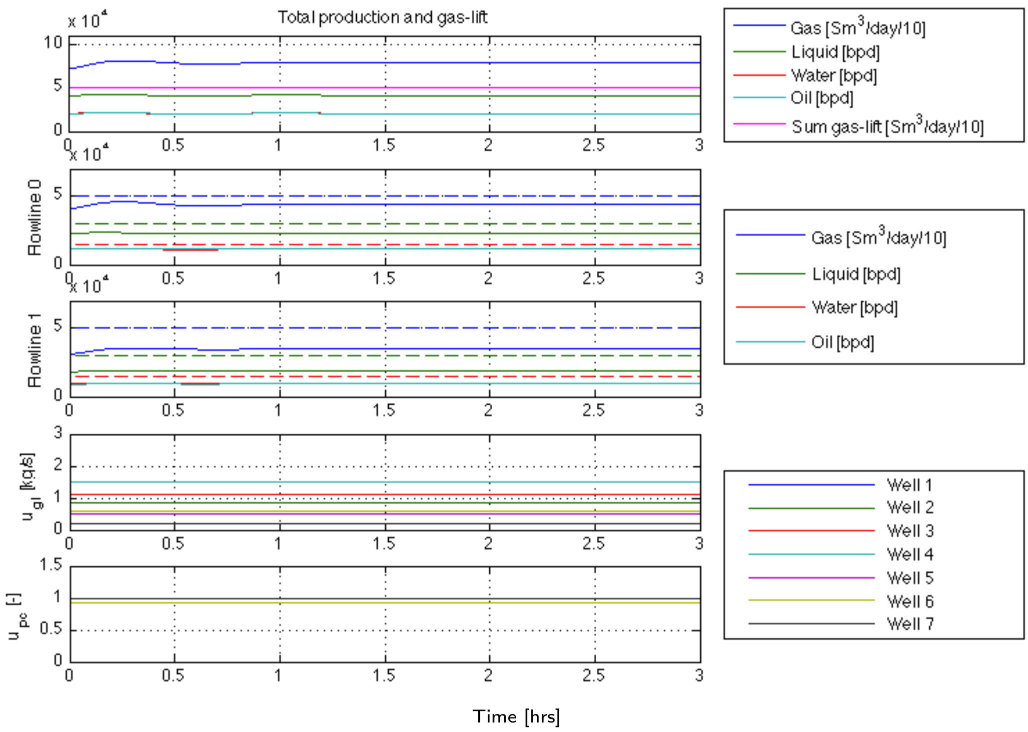


Figure E.4: Simulation of scenario 1 using static optimization method

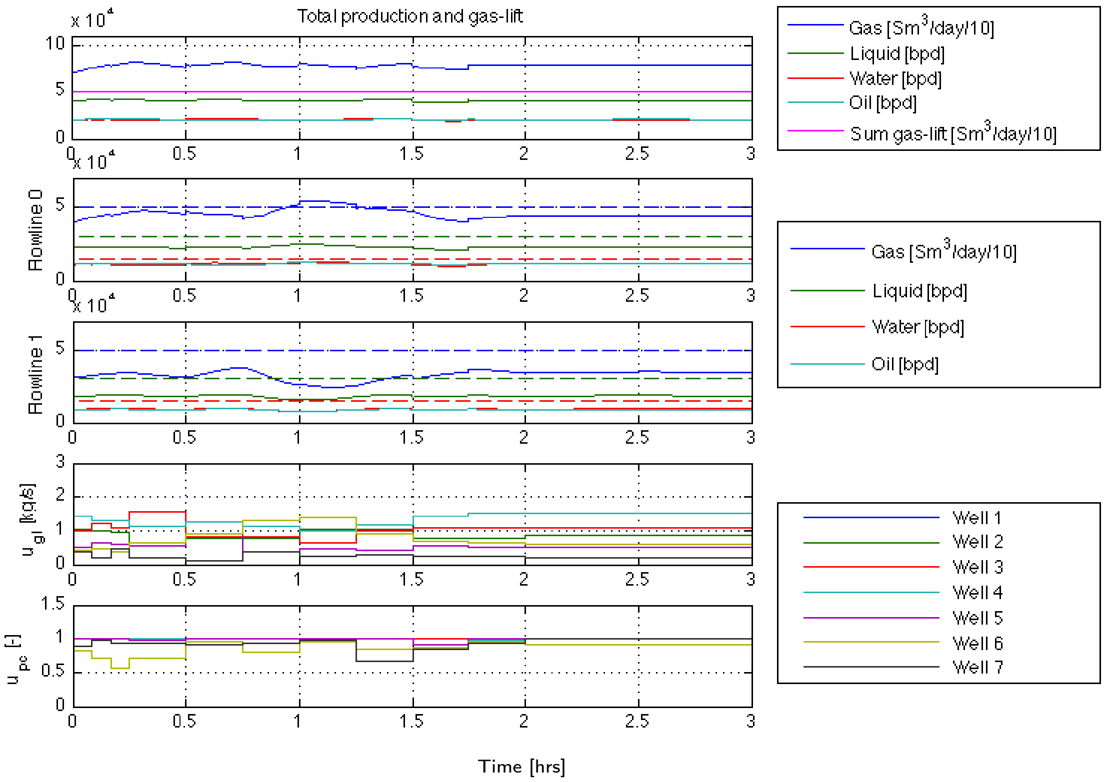


Figure E.5: Simulation of scenario 1 using dynamic optimization method 1

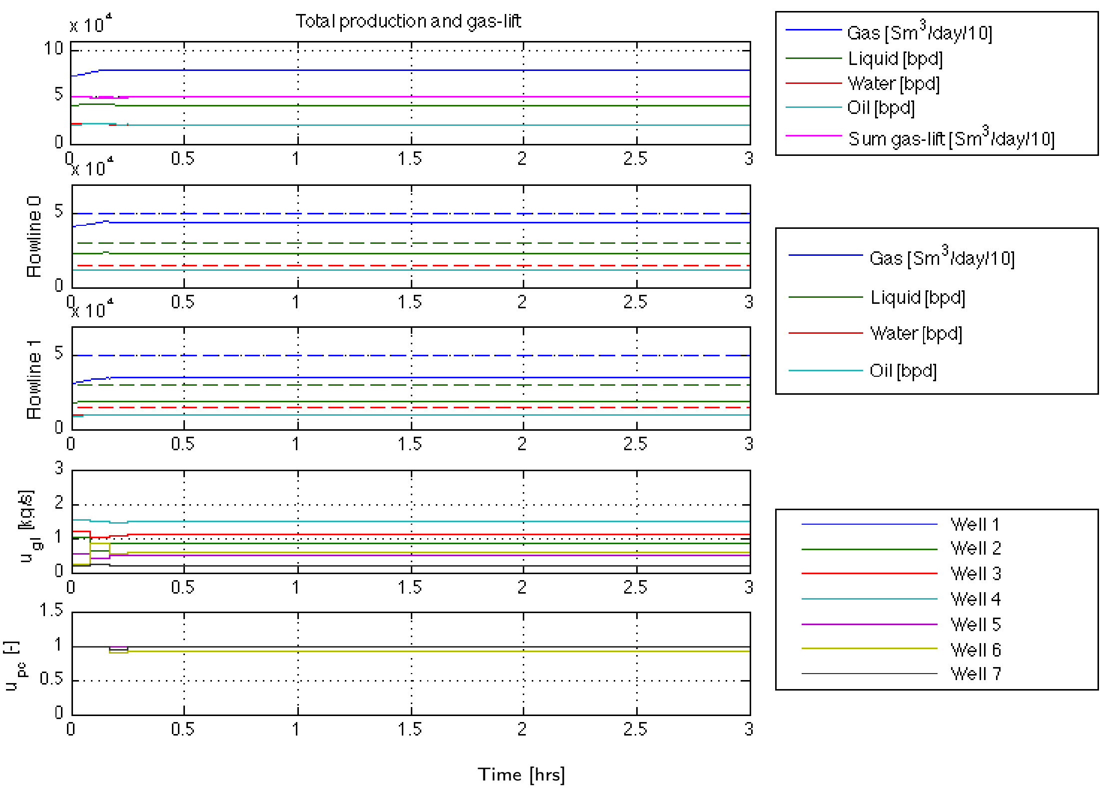


Figure E.6: Simulation of scenario 1 using dynamic optimization method 2

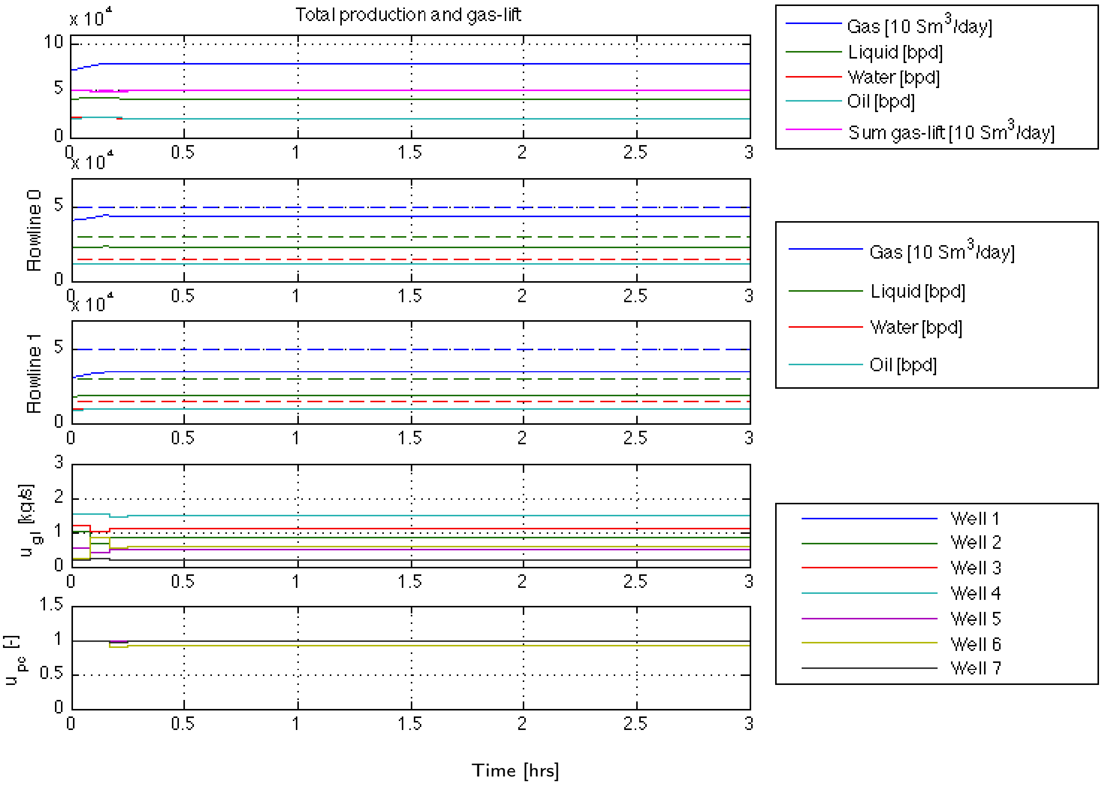


Figure E.7: Simulation of scenario 1 using dynamic optimization method 3

E.3 Scenario 2

Simulations of scenario 2 using the different production optimization methods:

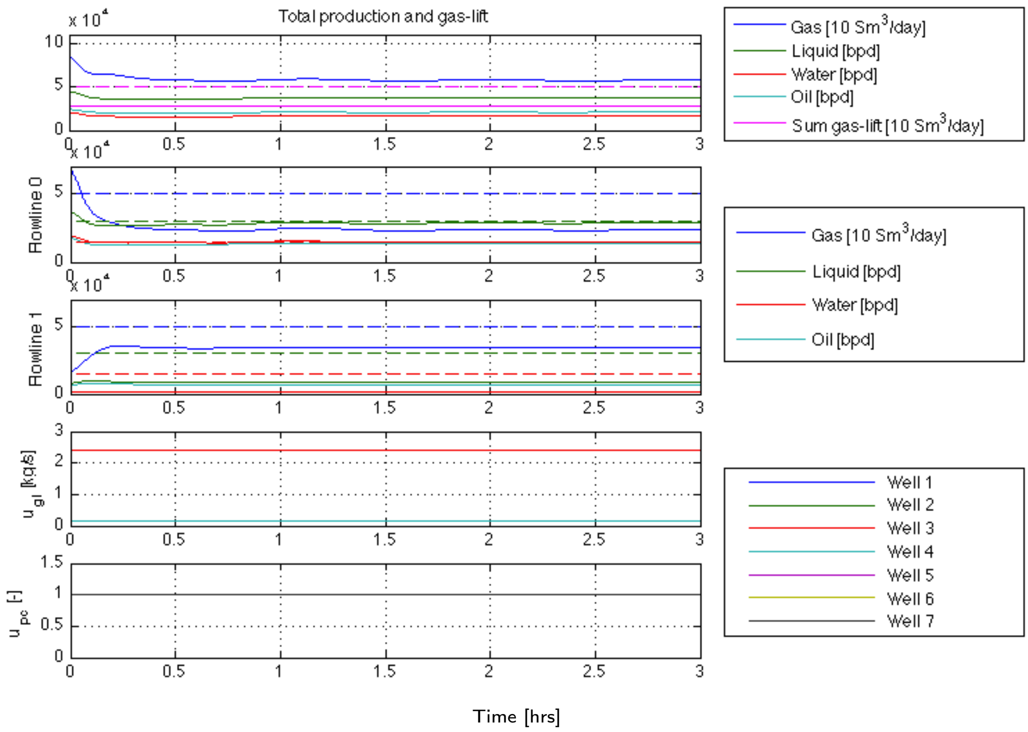


Figure E.8: Simulation of scenario 2 using optimization free method

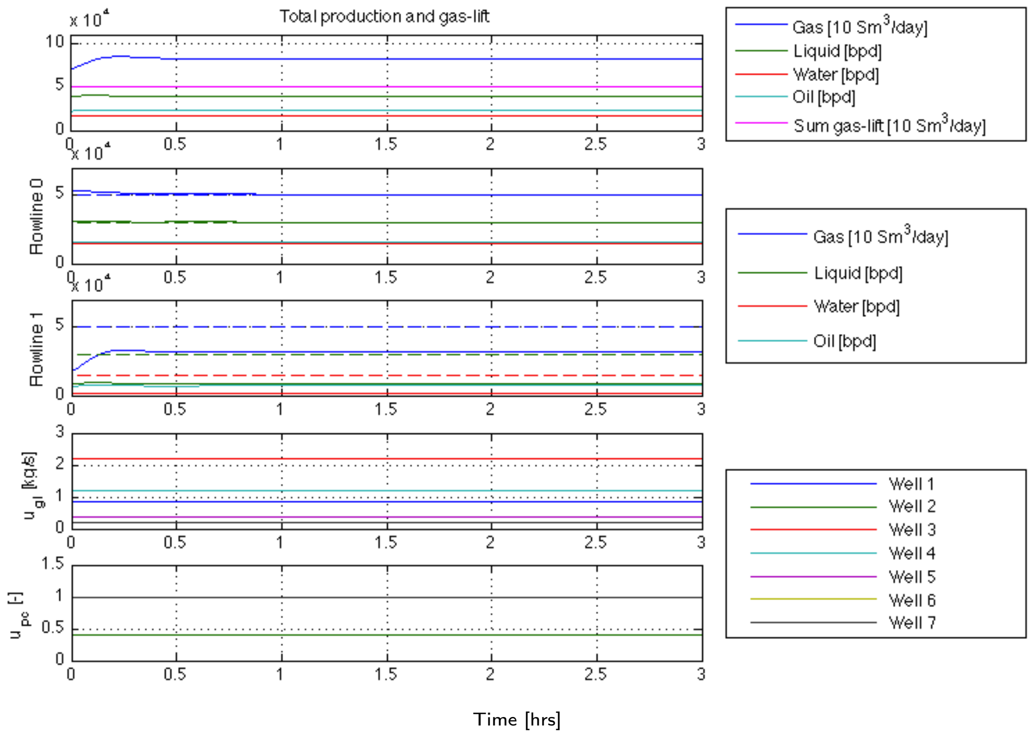


Figure E.9: Simulation of scenario 2 using static optimization method

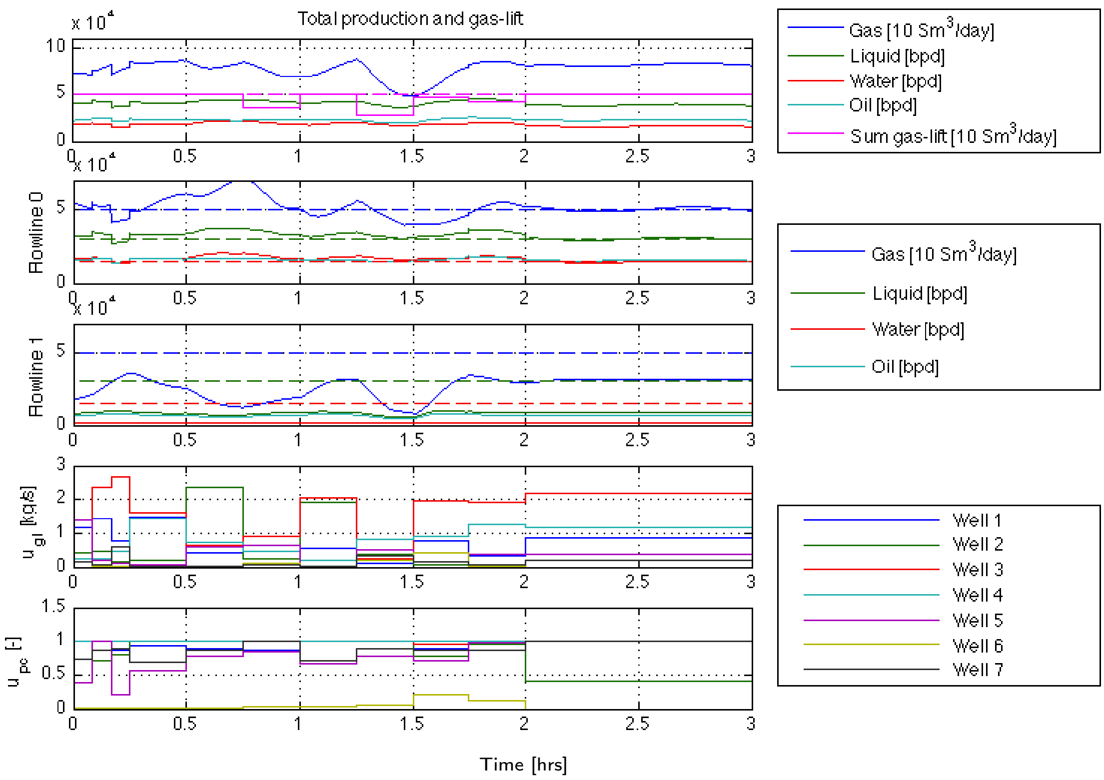


Figure E.10: Simulation of scenario 2 using dynamic optimization method 1

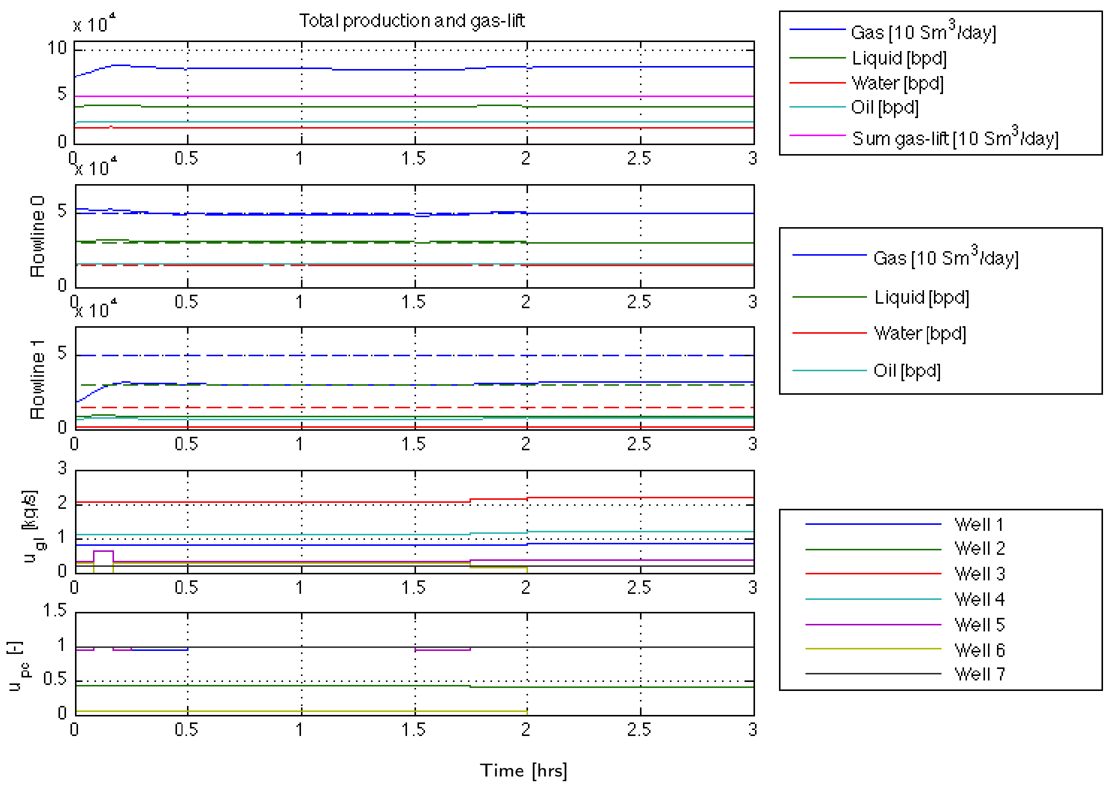


Figure E.11: Simulation of scenario 2 using dynamic optimization method 2

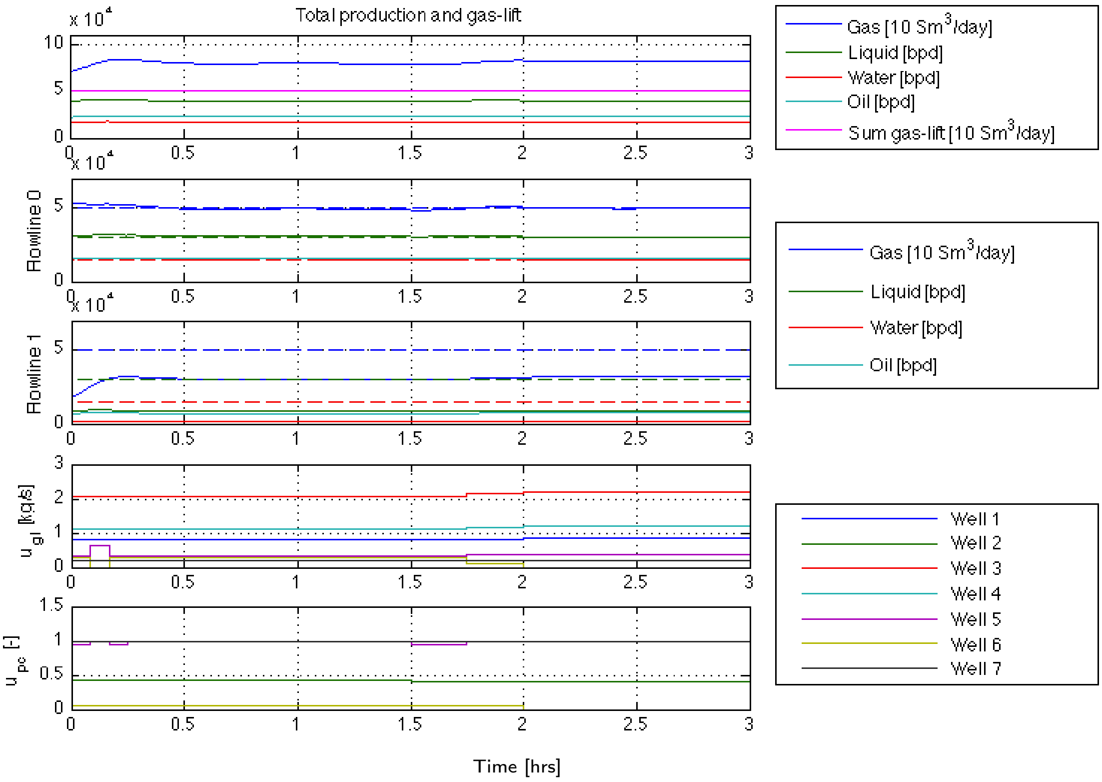


Figure E.12: Simulation of scenario 2 using dynamic optimization method 3

Appendix F

Parameters and nomenclature

In this appendix, the parameters used in this project and a complete nomenclature are presented. The well parameters used in this report are inspired by some of the wells producing to the FPSO 'P-35' in the Marlim field (see section 1.2). To keep the real parameters confidential, they have been changed, but the parameters used should still be quite realistic.

General parameters and parameters considered equal in all seven wells are presented in table F.1, and the well parameters are presented in table F.2. The system constraints are given in table F.3, and some relevant conversion factors used are given in table F.4. The nomenclature is found at the end of this appendix.

Parameter	Value	Unit
R	8.3145	J/(K·mol)
g	9.81	m/s^2
T	350	K
p_m	$5 \cdot 10^6$	Pa
p_r	$25 \cdot 10^6$	Pa
M_g	0.0195	kg/mol
ρ_o	930	kg/m^3
ρ_w	1030	kg/m^3
A_a	0.02	m^2
A_t	0.012	m^2
C	0.8	-
C_{pc}	0.0016	m^2
C_{iv}	0.00016	m^2

Table F.1: General parameters

Parameter	Well 1	Well 2	Well 3	Well 4	Well 5	Well 6	Well 7	Unit
V_a	30	30	25	30	25	30	30	m^3
V_t	18	18	15	18	15	18	18	m^3
L_w	400	400	800	600	500	600	800	m
r_{wc}	0.4	0.7	0.2	0.5	0.6	0.8	0.3	-
r_{gor}	0.08	0.07	0.07	0.09	0.06	0.08	0.10	-
Q_{max}	0.025	0.05	0.035	0.10	0.02	0.015	0.005	m^3/s

Table F.2: Well parameters

Constraint	Value
G_j	500 000 $[Sm^3/day]$
L_j	30 000 $[bpd]$
W_j	15 000 $[bpd]$
GL	500 000 $[Sm^3/day]$

Table F.3: System constraints

Unit	SI-unit
1 Sm^3/day (Standard cubic meters of gas per day)	$9.545\,139 \cdot 10^{-6}\ kg/s$
1 bpd (oil) (Barrels per day)	$1.711\,322\ \cdot 10^{-3} kg/s$
1 bpd (water) (Barrels per day)	$1.895\,335\ \cdot 10^{-3} kg/s$

Table F.4: Some conversion factors

Nomenclature

A_{max}	Maximum orifice area of a valve $[m^2]$
A_o	Area of the orifice in a valve $[m^2]$
A_t	Cross sectional area of the tubing $[m^2]$
C	Constant parameter in Vogel's equation $[-]$
C_d	Constant discharge coefficient of a valve $[-]$
C_{gl}	Valve specific constant for the gas-lift choke valve $[m^2]$
C_{iv}	Valve specific constant for the injection valve $[m^2]$
C_{pc}	Valve specific constant for the production choke valve $[m^2]$
C_v	Valve specific constant $[m^2]$
Δp	Pressure difference $[Pa]$
g	Acceleration of gravity, $g \approx 9.81$ $[m/s^2]$
L_a	Vertical height of the annulus $[m]$
L_w	Vertical distance from the reservoir to the injection valve $[m]$
M	Molecular weight $[kg/mol]$
m	Mass $[kg]$
M_g	Molecular weight of the gas $[kg/mol]$
m_{ga}	Mass of gas in the annulus, $[kg]$
m_{gt}	Mass of gas in the tubing $[kg]$
m_{lt}	Mass of liquid in the tubing $[kg]$
m_t	Total mass in the tubing $[kg]$
n	Number of moles $[mol]$
ν	Specific volume, $\nu = 1/\rho$ $[m^3/kg]$
ν_l	Specific volume of liquid in the tubing $[m^3/kg]$
p	Pressure $[Pa]$
p_a	Pressure in the annulus at the gas-lift choke valve $[Pa]$
p_{ai}	Pressure in the annulus at the injection valve $[Pa]$
p_{bh}	Bottom hole pressure (BHP) in a well $[Pa]$

p_{gl}	Pressure of the lift gas at the manifold $[Pa]$
p_m	Pressure in the flowlines at the manifold $[Pa]$
p_p	Pressure in the tubing at the production choke valve $[Pa]$
p_r	Reservoir pressure $[Pa]$
p_{ti}	Pressure in the tubing at the injection valve $[Pa]$
q	Volume flow $[m^3/s]$
Q_{max}	Theoretical absolute open flow potential of a well $[m^3/s]$
R	Universal gas constant, $R = 8.3145$ $[J/(K \cdot mol)]$
r_{glr}	Gas-to-liquid ratio in a well $[-]$
r_{gor}	Gas-to-oil ratio in a well $[-]$
r_{wc}	Water cut (WC) in a well $[-]$
ρ	Density $[kg/m^3]$
ρ_{gi}	Density of gas in the annulus at the injection valve $[kg/m^3]$
ρ_{gl}	Density of the lift gas at the gas-lift choke valve $[kg/m^3]$
ρ_l	Density of liquid flowing from the reservoir $[kg/m^3]$
ρ_o	Density of produced oil $[kg/m^3]$
ρ_p	Density of fluid in the tubing at the production choke valve $[kg/m^3]$
ρ_w	Density of produced water $[kg/m^3]$
T	Temperature $[K]$
T_a	Temperature in the annulus $[K]$
T_t	Temperature in the tubing $[K]$
u_{gl}	Mass flow of lift gas into the annulus (input variable) $[kg/s]$
u_{glc}	Gas-lift choke valve setting (Dimensionless variable in the range [0,1]) $[-]$
u_{pc}	Production choke valve setting (Dimensionless variable in the range [0,1]) $[-]$
V	Volume $[m^3]$
V_a	Volume of the annulus above the injection valve $[m^3]$
V_{lt}	Volume of liquid in the tubing $[m^3]$
V_t	Volume of the tubing above the injection valve $[m^3]$
w	Mass flow $[kg/s]$
w_{gi}	Mass flow of gas through the injection valve $[kg/s]$
w_{gl}	Mass flow of gas through the gas lift choke valve $[kg/s]$
w_{gp}	Mass flow of gas through the production choke valve $[kg/s]$
w_{gr}	Mass flow of gas from the reservoir into the tubing $[kg/s]$
w_{lp}	Mass flow of liquid through the production choke valve $[kg/s]$

w_{lr} Mass flow of liquid from the reservoir into the tubing $[kg/s]$

w_{op} Mass flow of oil through the production choke valve $[kg/s]$

w_{or} Mass flow of oil from the reservoir $[kg/s]$

w_p Total mass flow through the production choke valve $[kg/s]$

w_{wp} Mass flow of water through the production choke valve $[kg/s]$

w_{wr} Mass flow of water from the reservoir $[kg/s]$

Bibliography

[Ahmed 2006] Tarek Ahmed: "Reservoir Engineering: Handbook, Third Edition," Elsevier Inc, 2006, (ISBN 9780750679725)

[Arubi 2011] I.M.T. Arubi and H. Yeung: "Gamma Radiation Methods in Characterizing Horizontal and Vertical Multiphase Flow," Offshore Technology Conference, 2011 (OTC 22414)

[Balchen 2003] Jens G. Balchen, Trond Andresen and Bjarne A. Foss: "Reguleringsteknikk," Institutt for teknisk Kybernetikk, NTNU, 2003 (ISBN 9788247151471)

[Bampi 2010] Dirceu Bampi and Odair José Costa: "Marlim Field: An Optimization Study on a Mature Field," Society of Petroleum Engineers, 2010 (SPE 139376)

[Binder 2011a] Benjamin J. T. Binder: "Production Optimization," Report from internship, Petrobras, August 2011

[Binder 2011b] Benjamin J. T. Binder: "Dynamic production optimization of subsea well clusters," Project report, NTNU, December 2011

[Brown 1997] Robert Grover Brown and Patrick Y. C. Hwang: "Introduction to Random Signals and Applied Kalman Filtering," John Wiley & Sons Inc, 1997 (ISBN 9780471128397)

[Chen 1999] Chi-Tsong Chen: "Linear System Theory and Design," Oxford University Press Inc, 1999 (ISBN 9780195117776)

[CIPEG web] CIPEG website:
http://www.petroleo.rj.gov.br/site/index.php/component/content/article/161-recorde-producao
visited 07.02.2012

[Dzubur 2011] Lamija Dzubur and Andrea Langvik: "Optimization of Oil Production," Project report, NTNU, 2011

[Egeland 2002] Olav Egeland and Jan Tommy Gravdahl: "Modeling and Simulation for Automatic control," Marine Cybernetics AS, 2002 (ISBN 9788292356012)

[Eikrem 2008] Gisle Otto Eikrem, Ole Morten Aamo and Bjarne A. Foss: "On Instability in Gas Lift Wells and Schemes for Stabilization by Automatic Control," Society of Petroleum Engineers, 2008 (SPE 101502)

[Gunnerud 2011] Vidar Gunnerud: "On decomposition and piecewise linearization in petroleum production optimization," NTNU, 2011

[Hillier 2010] Frederick S. Hillier and Gerald J. Lieberman: "Introduction to Operations Research, International Edition 2010" McGraw-Hill, 2010 (ISBN 9780071267670)

[Hu 2004] Bin Hu: "Characterizing gas-lift instabilities," NTNU, 2004

[Imsland 2007] Lars Imsland: “Introduction to Model Predictive Control,” NTNU, 2007

[Ioannou 1996] Petros Ioannou and Jing Sun: “Robust Adaptive Control,” Prentice Hall Inc, 1996 (Out of print in 2003, electronic copy at: http://www-rcf.usc.edu/~ioannou/Robust_Adaptive_Control.htm)

[Johann 2011] Paulo Johann, Carlos Eduardo Abreu, Marcos Grochau and Edgar Thedy: “Advanced Seismic Imaging Impacting Brazilian Offshore Fields Development,” Offshore Technology Conference, 2011 (OTC 21934)

[Jones 2011] Cleveland M. Jones and Hernani Aquini F. Chaves: “Assessment of Yet-To-Find Oil in the Brazilian Pre-Salt Region,” Society of Petroleum Engineers, 2011 (SPE 143911)

[Lorenzatto 2004] R. A. Lorenzatto, R. Juiniti, J.A.T. Gomes, J.A. Martins: “The Marlim Field Development: Strategies and Challenges,” Offshore Technology Conference, 2004 (OTC 16574)

[de Luca 2003] João Carlos de Luca: “Brazilian Petroleum Exploration and Production: From Monopoly to Free Market,” Offshore Technology Conference, 2003 (OTC 15051)

[Maciejowski 2002] J. M. Maciejowski: “Predictive Control with Constraints,” Pearson Education Ltd, 2002 (ISBN 9780201398236)

[Nocedal 2006] Jorge Nocedal and Stephen J. Wright: “Numerical optimization,” Springer Science+Business Media LLC, 2006 (ISBN 9780387303031)

[NOIA web] National Ocean Industries Association website:
http://www.noia.org/website/article.asp?id=123
visited 05.02.2012

[Oliveira 1989] C.A.P. de Oliveira and R.P. Coelho: “Marlim Field Development,” Offshore Technology Conference, 1989 (OTC 5971)

[Pinto 2003] A.C. Capeleiro Pinto, C.C.M. Branco, J.S. de Matos, P.M. Vieira, S. da Silva Guedes, C. Pedroso Jr., A.C. Decnop Coelho, M.M. Ceciliano: “Offshore Heavy Oil in Campos Basin: The Petrobras Experience,” Offshore Technology Conference, 2003 (OTC 15283)

[PSC web] PSC website:
http://gasliftnd.com/GasLift.aspx
visited 02.02.2012

[Ribeiro 2005] N. Ribeiro, D. Steagall, R. Oliveira, L. Formiga, P. Kerber, and M. Jaeger: ”Challenges of the 4D Seismic in the Reservoir Management of the Marlim Field”, Society of Petroleum Engineers, 2005 (SPE 94905)

[Rigzone web a] Rigzone website:
http://www.rigzone.com/data/utilization_trends.asp
visited 05.02.2012

[Rigzone web b] Rigzone website:
http://www.rigzone.com/news/image_detail.asp?img_id=4082&a_id=52788
visited 07.02.12

[Sangesland 2007] Sigbjørn Sangesland: “Subsea Production Systems - Introduction to Subsea Production Technology,” NTNU, 2007

[Shell web] Shell website:
http://www.shell.us/home/content/usa/aboutshell/media_center/
news_and_press_releases/2011/11172011_perdido.html
visited 07.02.2012

[Skogestad 2005] Sigurd Skogestad and Ian Postlethwaite: "Multivariable Feedback Control - Analysis and Design," John Wiley & Sons Ltd, 2005 (ISBN 9780470011683)

[Skogestad 2009] Sigurd Skogestad: "Prosessteknikk," 3. utgave, Tapir Akademisk Forlag, 2009, (ISBN 9788251924573)

[Storvold 2011] Vegard Storvold: "Optimization of long run production performance in aging offshore petroleum fields," Project report, NTNU, 2011

[Valle 1998] Arne Valle: "Multiphase Pipeline Flows in Hydrocarbon Recovery," Multiphase Science Technology vol. 10, 1998

[Vogel 1968] J.V. Vogel: "Inflow Performance Relationships for Solution-Gas Drive Wells," Society of Petroleum Engineers, 1968, (SPE 1476)

[White 2008] Frank M. White: "Fluid Mechanics, Sixth Edition," McGraw-Hill, 2008 (ISBN 9780071286459)

[Xu 1989] Z.G. Xu and M. Golan: "Criteria for Operation Stability of Gas-Lift Wells," Society of Petroleum Engineers, 1989 (SPE 19362)