NTNU – Trondheim
Norwegian University of
Science and Technology

Mixed Integer Model Predictive Control of Multiple Shale Gas Wells

Espen T Nordsveen

Master of Science in Engineering Cybernetics

Submission date:	May 2012
Supervisor:	Bjarne Anton Foss, ITK
Co-supervisor:	PhD-student Brage Rugstad Knudsen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

Horizontal wells with multistage hydraulic fracturing are today the most important drilling technology for shale gas extraction. Considered unprofitable before, the production has now become economically profitable due to advances in technology. Shales main characteristics is its low permeability, making the gas challenging and expensive to extract. Hydraulic fracturing stimulates the wells by creating additional conductivity, making the gas flows from storage pores to the well. This flow only possible in a short time scale, and states the need for multistage fracturing. Shale gas flow therefore exhibits a high initial peak, followed by a rapid decline in production rates. The use of shut-ins of shale gas wells allows for pressure build-up and may prevent liquid loading, as a means of boosting production.

Shut-ins are used as on/off control variables in short-term model-based optimization of multiple shale gas wells with the objective of tracking a reference rate, while at the same time avoiding liquid loading. Previous work have focused on open-loop optimization. Here, an open-loop formulation is compared to a closed-loop formulation, in the form of mixed integer model predictive control. Both formulations are implemented in IBM ILOG CPLEX, with and without disturbances.

Optimal production settings are solved in the presence of global constraints on production rates and minimal shut-in time. This allows for shut-ins with variable periods. The implementation is sensitive to initial conditions, horizons and weighting factors. The closed-loop formulation shows the best ability to reduce the effects of disturbances.

Sammendrag

Horisontale brønner med multiple hydrauliske oppsprekninger er per i dag den viktigste teknologien ved drilling av nye brønner for skifergass-utvinnelse. På grunn av store teknologiske fremskritt de siste årene, har denne prosessen, som før ble betraktet som ulønnsom, nå blitt meget lønnsom. Hovedkarakteristikken til skifergass er lav permeabilitet, hvilket gjør gassen svært vanskelig og dyr å utvinne. Hydraulisk oppsprekning stimulerer reservoaret ved å danne porer med høyere permeabilitet hvor gassen kan strømme fra. Siden gasstrømningen kun er gyldig i fåminutter, får skifergass-strømning en karakteristisk høy initiell produksjon, hvorfra strømningen raskt avtar. Nedstegning av brønnene brukes for å bygge opp trykket i brønnen og unngå væskeopphopning, hvilket medfører høyere produksjon.

Av/på- ventiler kan brukes som binær-variable i kortsiktig modell-basert optimering av flere skifergass-brønner, med mål å følge en gitt referanse og unngå væske-opphopninger. Tidligere arbeid har fokusert på åpen sløyfe-optimering. I denne rapporten sammenliknes en åpen sløyfe-formulering med en lukket sløyfe-formulering. Lukket sløyfe-formuleringen implementeres som en mikset heltall modellbasert prediktiv regulator. Begge formuleringene implementeres i IBM ILOG CPLEX, både med og uten forstyrrelser.

Optimale produksjonssettinger løses basert på globale beskrankninger på produksjonsrate og minimal nedstegningstid. Implementeringen viser seg å være sensitiv i forhold til initial-betingelser, horisonter og vektinger. Lukket sløyfe-formuleringen viser best evne til å redusere effektene av forstyrrelser.

Preface

This master thesis is written as part of my last semester in the Master of Science program in Technical Cybernetics at the Norwegian University of Science and Technology. The thesis is motivated by the increasing interest of unconventional gas exploration all over the world. The work has been very interesting and challenging. The development of a dynamic wellhead model, and the implementation of the model predictive controller in CPLEX via OPL have especially been a time consuming task.

During both the final year project and the master thesis, I have been lucky working to with PhD student Brage Rugstad Knudsen. I am deeply grateful for his determined guidance and encouraging discussions throughout the year. I would also express my gratitude to my supervisor Professor Bjarne A. Foss, for his good ideas and constructive feedback. Finally, I would like to thank my fellow students for an encouraging working environment, and my girlfriend Marit, for her support.

Espen Taraldsvik Nordsveen
May 2012

Contents

Chapter 1

Introduction

This chapter gives a brief background of shale gas properties and recovery techniques as it appears today. In addition, a short presentation of production optimization is included.

This chapter is to a large extent based on the introductory chapter in Nordsveen (2011).

1.1 Background

Shale gas has become one of the most rapidly growing forms of natural gas, and the potential of shale gas has been well known for a long time. The success started in the Barnett field, the most developed shale basin in the U.S. The U.S has serveral reservoirs, spread over large fields, see figure 1.1. Economist (2011) reports estimated recoverable reserves to be 862 trillion cubic feet, based on information from America's Energy Information Administration. There are also a considerable amount of shale gas reserves even in Europe, estimated to be 639 trillion cubic feet. Europe has traditionally not been a significant exporter of shale gas, mainly because of geologically challenges, and thus also higher costs of extraction. In 2008, 100 rigs were operating in Europe, in contrast to 1600 in America (Economist, 2011).

The extraction of the gas has been limited by the technology. The major challenges with shale are its low permeability, that is, its ability to transmit fluid. Shale is characterized as tight reservoir, and the extraction of the gas is therefore complex and cost expensive.

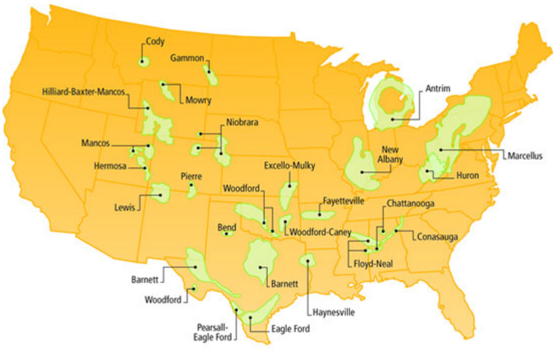


Figure 1.1: U.S shale gas basins. Source: US Department of Energy

1.2 Shale gas characteristics

Shales are special in the sense that it can store large amounts of gas within it. Three different storage mechanisms exists

- Within natural micro-fractures
- Within minute rock pores
- As adsorbed gas

The main characteristics of shale gas is its low permeability, ranging from $10^{-3} - 10^{-6}$ mD. This makes transmition of fluid difficult.

The tightness of shale formations makes it possible to produce gas only when extensive network of fractures exists. The gas will then flow into these fractures, and into the producing well. However, this flow is only valid in the range of minutes (Carlson and Mercer, 1991), causing additional complexity in gas production, espescially when there is insufficient fractures in the formations. Additional conductivity is thus needed. This is done by stimulating the reservoir by *hydraulic fracturing*, creating large and complex networks of fractures.

Shale gas has traditionally been modelled as *dual-porosity* models, a concept first introduced by Warren and Root (1963). The concept builds on an idealized model of matrix blocks which supplies interconnected set of fractures with fluid. Figure 1.2, taken from Carlson and Mercer (1991), shows two examples of idealized fractured reservoirs, compared to an actual reservoir. The upper left shows vertical fractures, the upper right is with cubical elements. The matrices can store large quantities of gas, but cannot transort the gas for long distances. The fractures can transport the gas, but has limited storage

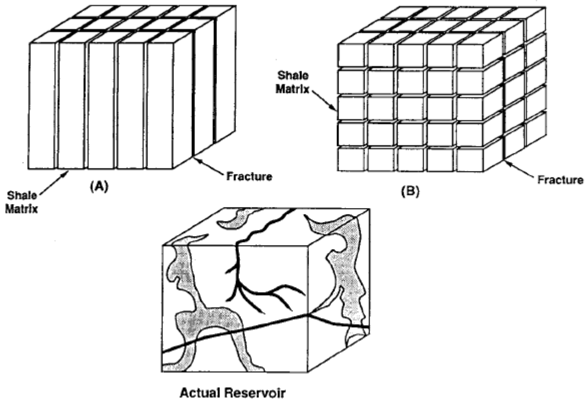


Figure 1.2: Ideal dual-porosity-models. Source: Carlson and Mercer (1991)

capabilities.

The chemical composition of shale gas is mainly methane, and thus emits less carbon dioxide than traditional fossil fuel like oil and coal. Emission of nitrogen oxides (NO_x) and sulphur dioxide (SO_2). Shale gas is thus considered to be central in reduction of greenhouse gases.

1.3 Recovery techniques

Traditionally, shale gas wells has been vertically drilled, with one well site per reservoir. In the past few year, much effort has been put into recovery techniques, bringing along improved downhole technology and reduced costs. This has made horizontal well drilling the prefered drilling technology. The use of hydraulic fracturing to stimulate the reservoirs are also beeing deployed extensively.

1.3.1 Horizontal drilling

Due to improved downhole technology and reduced costs, horizontal drilling has now become the the preferred method of choice when drilling new wells. In the Barnett shale basin, more than 90% of all new wells now exert horizontal drilling (Jenkins and Boyer, 2008). The advantages are obvious; the wellbore is exposed to a significantly larger area of the reservoir than traditional vertical drilling. However, this comes at a price. Drilling of horizontal wells can be as much as 300 percent more costly than vertical drilling on the same target reservoir (Helms, 2008). Horizontal drilling is hence only applied when the economical benefits are ensured. Karcher et al. (1986) reported that the productivity could increase by as much as 10 times or more, due to the increased areal sweep efficiency, and thus can be more economically beneficial. In conventional reservoirs with high permeabilities, a vertical well could produce nearly the same amount of gas as a horizontal well, only at a lower cost. In unconventional reservoirs with low permeabilities, a horizontal well has the ability to produce as much as 2.5 to 7 times more gas than vertical wells (Helms, 2008).

Horizontal wells are drilled with a hydraulic motor mounted directly above the drill bit, which can be controlled from the surface. Without rotating the drill pipe, the drill bit can be rotated by the motor. This ”steerable” motor is responsible for drilling in the desirable direction, and can both be steered from vertical to horizontal direction, and to left and right. The curved section of the drill string has typically a radius of 300-500 ft (Helms, 2008). The drill string contains various sensors that transmit sensor reading of among others pressure and temperature at the bottom hole, and azimuth[1] and inclination[2] of the drill string.

1.3.2 Hydraulic fracturing

Because of the low permeability in shale formations, additional permeability needs to be created by stimulating the existing networks of natrual fractures in the shale (Ridley, 2011). This is done by *hydraulic fracturing* (”fracking”). Fracking is done on-site by the use of, among others, truck-mounted pumps and fracturing fluid (Guo et al., 2007). Figure 1.3 shows an example of equipment layout on-site when drilling unconventional wells.

When a new well is drilled, it is cased and cemented, whereas the casing is perforated with explosive charges. Water is then mixed with sand, and pumped with high pressure through the perforations. The mixing of sand prevents the fractures from closing. The process usually takes 3-10 days, and creates channels in which the gas can flow into the wellbore.

[1] The horizontal angle measured clockwise from north

[2] The angle relative to the vertical

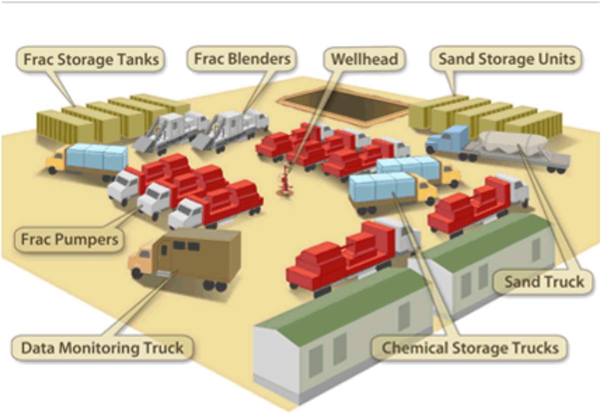


Figure 1.3: An example of equipment layout in hydraulic fracturing treatment. Source: http://fracfocus.org/

1.4 Production optimization

Production optimization is the complex task of maximizing the production over a the long term, whilst minimizing the production costs. Usually this is based on the net present value (NPV), and imposes great challenges, as several engineering disciplines are involved, and the interaction between several operational parameters. To achieve and maintain a profitable production, evaluation and monitoring of the whole production system is essential; production equipment, production history, reservoir, fluids, wellbore etc (O'How and Kubat, 1996). This is dependent of data collection and expertise in fields ranging from engineering, politics, economics and psychology (Saputelli et al., 2003).

Figure 1.4 shows a typical layout for the hierarchy in the oil and gas industry (Saputelli et al., 2003). The complexity of an automation process is divided into different layers, making the problem tasks more manageable. In the upper levels, the whole project is managed through capacity and operational planning, while the well control and manipulation of critical variables are handled in the lower levels. Decisions in higher levels will eventually affect the lower levels. The main question is how to operate the reservoir in the best fashion, based on control elements in the production facilities.

Traditionally, mathematical models and field data are used in the open-loop optimization. In the recent years, more intelligent models have come to life, including real-time model-based field management with feedback (Giorgio et al., 2012; Saputelli et al., 2003).

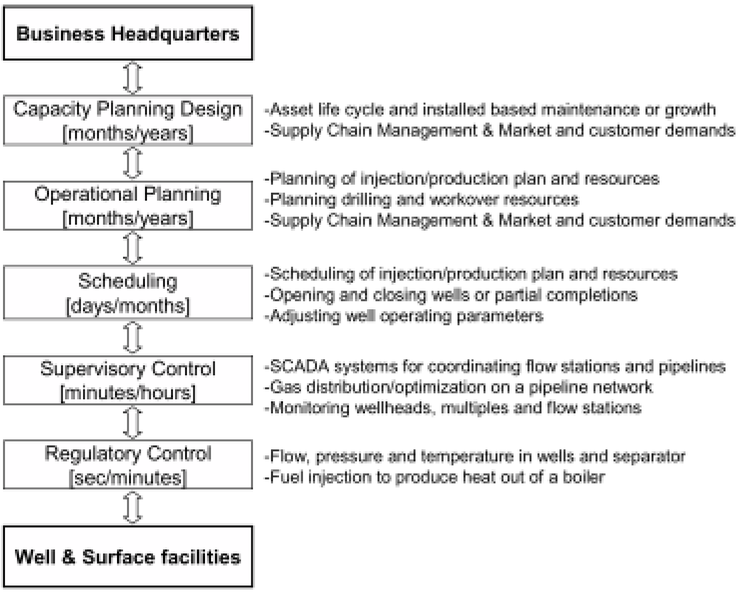

Business Headquarters
Capacity Planning Design [months/years]
-Asset life cycle and installed based maintenance or growth
-Supply Chain Management & Market and customer demands
Operational Planning [months/years]
-Planning of injection/production plan and resources
-Planning drilling and workover resources
-Supply Chain Management & Market and customer demands
Scheduling [days/months]
-Scheduling of injection/production plan and resources
-Opening and closing wells or partial completions
-Adjusting well operating parameters
Supervisory Control [minutes/hours]
-SCADA systems for coordinating flow stations and pipelines
-Gas distribution/optimization on a pipeline network
-Monitoring wellheads, multiples and flow stations
Regulatory Control [sec/minutes]
-Flow, pressure and temperature in wells and separator
-Fuel injection to produce heat out of a boiler
Well & Surface facilities

Figure 1.4: The hierarchy in a field operation. From Saputelli et al. (2003)

1.5 Scope

The main objective of this thesis is to study the introduction of model predictive control into shale gas production. The intention is to compare and analyse the potential of MPC over open-loop control. The main control variabel are the on/off valve used to switch a well, making it a mixed integer problem. Short-term tracking performance of a reference rate is used as a means of comparing both strategies.

1.6 Report outline

- Chapter 2 gives an introduction to mixed integer linear programming and model predictive control, two important subjects throughout this thesis.
- Chapter 3 presents the reservoir model developed in Nordsveen (2011), some further modelling results of the model and a litterature review of possible gas metering.
- In chapter 4, the open-loop and closed-loop production optimization

problems are described

- Chapter 5 discuss implementation strategies for solving both the open-loop and closed-loop problems in CPLEX
- Chapter 6 presents results from the simulations, with discussion in chapter 7

Chapter 2

Theory

This chapter is ment to give the reader an introduction to important concepts in *mixed integer linear programming* (MILP) and *model-based predictive control*. First, MILPs and solution methods are presented, along with available solution software. Second, a brief presentation of the general *receding horizon* idea in MPC is given.

2.1 Mixed integer linear programming

Linear programming is a class of mathematical programs with certain properties; they have linear objective functions and linear constraints (both equalities and inequalities). The development of linear programming and the *simplex method* (the solution method), made way for formulating and analyzing large models in efficient and systematic ways. However, linear programs imposes restrictions on the variables, namely the assumption of *divisibility*. The divisibility assumption allows linear programs to have *any* values, including noninteger values. In certain problems, only integer values makes sense for the decision variables. Imposing this restriction to the decision variables makes the problem an *integer linear program*. Further, if only some decision variables are allowed to have integer values, the problem is referred to as a *mixed integer linear program*. Integer linear programs leads to more demanding solution procedures.

Following the notation in Pochet and Wolsey (2006), any *mixed integer linear program* (MILP) may be written as

$$Z(X) = min\left\{cx + fy : (x, y) \in X\right\} \tag{2.1}$$

where *min* means that this is a minimization problem, and x and y denotes the continuous and integer variables, respectively. X denotes the set of feasible solutions. It is described by a number of nonnegative linear constraints on x and y. Mathematically, this is written as

$$X = \left\{(x, y) \in \mathbb{R}^n_+ \times \mathbb{Z}^p_+ : Ax + By \geq b\right\} \tag{2.2}$$

where p and n denotes the dimensions. Often, the integers y is in the binary domain. In this case, $\mathbb{Z} \rightarrow \{0, 1\}$. Note that if $(x, y) \in \mathbb{Z}^p$ only, the problem is reduced to a *pure integer problem.* If the set X is empty, then $Z(X) = +\infty$ by convention.

2.1.1 Linear relaxations

Generations of relaxation is a key issue in MILPs. Or, more correctly, it plays an important role in solving MILPs in the sense of optimization algorithms. Several methods for relaxation is possible (Floudas, 1995);

- Omitting one or several constraints in P
- Setting coefficients of binary variables in the objective function equals to zero
- Replacing integrality conditions on the binary variables

The relaxation most often used is the last one; replacing the integrality condition, that is, $y \in \{0, 1\} \rightarrow y \in [0, 1]$. This form of relaxation leads to a *linear program* relaxation.

The principle is as follows; Say for instance that P_X is an extended feasible set, or a *formulation*, of (2.2), that is

$$P_X = \{(x, y) \in \mathbb{R} \times [0, 1] : Ax + By \geq b\} \tag{2.3}$$

and $X \subseteq P_X$. Then, X includes all the points in P_X where y is an integer. This is written as $X = P_X \cap (\mathbb{R}^n \times \mathbb{Z}^p)$. A linear relaxation of (2.1) will thus be

$$Z(P_X) = min \{cx + fy : (x, y) \in P_X\} \tag{2.4}$$

As X is a subset of P_X, and (2.4) is a minimization problem, (2.4) defines a lower bound on the optimal objective value; namely $Z(P_X) \leq Z(X)$. This holds in general. An upper bound $Z(X) \leq Z$ is achieved by any feasible solution.

2.1.2 Solving MILP models

Although MILPs falls under the *linear program* category, integer program problems are NP-hard, and no efficient algorithm exists which solves every problem. However, by the use of relaxations, most MILPs today are successfully solved by incorporating linear program algorithms, such as the simplex method into the solution steps. These solution methods falls into one of the following categories (Gunnerud, 2011)

- Cutting plane methods
- Enumerative methods

Pure cutting plane methods will not be paid much attention here. Instead, an introduction to *branch-and-bound* and *branch-and-cut* algorithms are given. Branch-and-cut is a hybrid solution method, consisting of both cutting planes and branch-and-cut methods. Information about cutting plane methods can be found in Wolsey (1998), Avriel (2003) and Cornuejols (2008).

Branch-and-bound solution algorithm

The branch-and-bound is the most used algorithm for solving integer programs. The algorithm is characterized as a *divide-and-conquer* algorithm, based on multi-branched recursion; it recursively breaks down a problem into sub-problems, until the sub-problems are easy enough to solve. The solutions are then combined to give the solution of the entire problem.

The algorithm is initialized by solving a linear relaxation of (2.1). This step creates a first upper bound on the solution. If $y^* \neq \mathbb{Z}$, the solution is not feasible, as it contains noninteger variables. Thus, two new constraints arises: $y \leq \lfloor y^* \rfloor$ and $y \geq \lceil y^* \rceil$. The solutions between these two constraints must be removed.

The removal of the solutions is done by *branching* the variable, based on some branching rule. The set P_X is then replaced by two disjoint set. The linear relaxation will thus always give a lower upper bound solution compared to the step before. The two disjoint sets are then analyzed separately.

Each new subproblem is solved by the simplex method, which obtain new bounds on the solution. Which set to be chosen first is a matter of the chosen search strategy; either *depth-first-, breadth-first-, best-bound search*, or a combination of these. More information about each type is found in Pochet and Wolsey (2006).

Several cases may arise when solving each new subproblem. The relaxation may have no feasible solution, $Z \leq Z^*$, $Z \geq Z^*$, where Z^* is the best solution so far, or the solution is *optimal*, that is, the solution has integer values for the integer-restricted variables. The algorithm terminates when there is no new subproblems.

Branch-and-cut solution algorithm

As mentioned, the branch-and-cut-algorithm is a hybrid solution strategy available for MILPs, as it combines a cutting plane with a branch-and-bound algorithm. It is initialized in the same matter as the branch-and-bound algorithm. If the initial solution contains fractional solutions for one or more of the integer variables, a cutting plane algorithm is called. This algorithm is calling a separation algorithm to repeatedly generate valid inequalities that is needed; that is, it generates contraints which are satisfied by all feasible integer points, but violated by the current (fractional) solution. It will eventually start branching the solution space into subspaces by the branch-and-bound algorithm.

2.1.3 Solution software

A number of commercial solver packages are available for MILPs. The most prominent and most used solvers are CPLEX[1] and Xpress[2]. They are both state-of-the-art and highly efficient. Alternative solvers are Gurobi and MOSEK. They all have the advantage of having support for different programming languages, including Matlab through the toolbox YALMIP[3].

[1] from IBM®

[2] from FICO™

[3] See Löfberg (2004)

2.2 Model predictive control

Throughout the history, the ideas behind *Model Predictive Control* has been know under several names; *Model predictive heuristic control, dynamic matric control, adaptive predictive control*, to name a few (see Richalet et al. (1978), Cutler and Ramaker (1980) and Martin-Sanchez (1976)). They all shared the essential features of MPC; an internal model, computations of control signal by optimization based on constraints, and the *receding horizon* idea (Maciejowski, 2002). A more general description of a model predictive controller is one which

- uses a internal model to predict future behavior
- optimizes future behavior
- handles constraints on input and output signals

It is now one of the most used controller in the process industry, and it is also gaining acceptance in other application areas.

The presence of constraints makes MPC differ from the traditionally *linear quadratic* (LQ) controller, where a feedback control law is computed based on minimization of a given *cost function* without constraints. A model predictive controller is therefore an integration of a LQ controller.

2.2.1 Basic formulation

In the literature, many different notations in model predictive control exists. However, they all explains the same. Here, the notation in Imsland (2007) is used, and presents the *general* idea behind the receding horizon.

Cost function

The cost function is essential in MPC theory. This is the function which is to be minimized or maximized, depending on the problem, based on some constraints. In the discrete, general trajectory tracking SISO case, the cost function penalizes deviations from the desired controlled outputs $\hat{y}_{k+i+1}$ from a desired reference trajectory $y(k+i+1)^d$. The subscript $(k+i+1)$ indicates that the signal depends on conditions at time k, with $i = 0, 1, \dots, N$. The notation $\hat{y}$ indicates that only an estimate of the state variables is known. That is, the state cannot be measured. The penalization is done with some tuning matrices $q(i)$ and $r(i)$, which reflects the relative importance of the signals. Another tuning parameter is the *prediction horizon* N. This is expressed mathematically as

$$J(k) = \sum_{i=0}^{N} q(y_{k+i+1} - y^d_{k+i+1})^2 + r(u_{k+i} - u^d_{k+i})^2 \tag{2.5}$$

Note that the cost function is quadratic, which is the case in general. In addition, if the constraints are in the form of linear inequalities, the problem is a quadratic programming (QP) problem. Solution methods are known to solve such a problem reliably and quickly (Maciejowski, 2002).

Constraints

Constraints are what makes MPC differs from the LQ controller. The constraints are most often on the input u_k, input rate Δu_k, and output z_k. Constraints on the state x_k are also common. Physically, constraints on the input can be interpreted as some saturation characteristics; for instance, finite range in valves, maximum flow due to pipe diameter, and so on. If, for instance, actuators have limited slew rate, constraints on input rate is also present.

2.2.2 MPC principle

Processes do, in many cases, have multiple inputs and multiple outputs (MIMO). Model predictive controls are able to handle MIMO systems in a nice way. A MIMO system may be described by a discrete state space model

$$x_{k+1} = Ax_k + Bu_k \tag{2.6}$$

$$y_k = Cx_k \tag{2.7}$$

which may be obtained from linearization of a nonlinear process. For a constant setpoint, the performance to be optimized is based on minimization of the cost function (Imsland, 2007)

$$J = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \tag{2.8}$$

If, however, the objective is to track some given, possibly changing, set points, the objective function would be

$$J = \sum_{k=0}^{\infty} (y_k - y_k^d)^T Q (y_k - y_k^d) + (u_k - u_k^d)^T R (u_k - u_k^d) \tag{2.9}$$

where y_k^d is the reference. It may be constant along the prediction horizon. Note the similarities with the SISO case in the cost function (2.5).

The essence of MPC is prediction of future variables y_k, x_k and u_k. For the open-loop problem, the predictions are done straightforward based on (2.7):

$$\begin{aligned} x_1 &= Ax_0 + Bu_0 \\ x_2 &= Ax_1 + Bu_1 = A^2x_0 + ABu_0 + Bu_1 \\ &\vdots \\ x_N &= Ax_N + Bu_N = A^Nx_0 + A^{N-1}Bu_0 + \ldots + Bu_{N-1} \end{aligned}$$

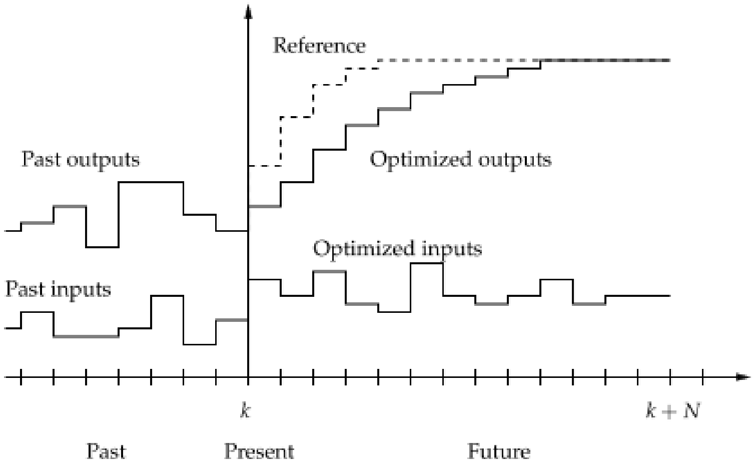


Figure 2.1: The principle of a model predictive controller. Source: Imsland (2007)

2.1 shows how the model predictive controller works. At each timestep, the controller optimizes future inputs and outputs based on a given reference, by only using the first of the calculated inputs as input to the process. That is, at each timestep, a QP problem is solved over the entire prediction horizon, a new input sequence $(u_k, u_{k+1}, \ldots, u_{k+N})$ is obtained, and the first input in the sequence is applied to the process. In the next time instant $(k + 1)$, the process is repeated. This is the receding horizon idea.

Chapter 3

Reservoir modeling and shale gas metering

This chapter starts with a summary of the reservoir model and inflow model developed in Nordsveen (2011). The reader is referred to this report for further details on the modelling. A study of liquid loading are adressed, and a simulation of applying constant shut-in times is included. A first order wellhead choke model are developed and studied. At the end, a litterature review of possible measurements in gas wells is carried out.

3.1 The reservoir model

The radial shale gas model developed and analyzed in Knudsen (2010b) yielded a radial one-dimensional model. This model was further developed into two dimensions in cartesian coordinates in Nordsveen (2011). The model is as follows

$$\phi\mu c\frac{\partial m}{\partial t} = \frac{\partial}{\partial x}(k(x,y)\frac{\partial m}{\partial x}) + \frac{\partial}{\partial y}(k(x,y)\frac{\partial m}{\partial y}) \tag{3.1}$$

where ϕ is the porosity, μ is the viscosity, and c is the compressibility of the gas. The model is based on the following assumptions

- The well is horizontal, and gravity effects are neglected
- The permeability is uniform along the half length
- The gas is single phase

- The reservoir consists of only one layer

3.1.1 State space formulation

Due to symmetry reasons and no-flow boundaries, only half a fracture and half a reservoir was modeled (referred to as half geometry). For one fracture, the simulation results of the model needs thus to be multiplied by 2. Likewise, for n number of fractures, the results needs to be multiplied by $2n$.

By applying a finite difference scheme on 3.1, a set of ODEs are obtained. This was done in Nordsveen (2011). The state space formulation of the model in (3.1) was expressed as

$$\mathbf{E}\dot{\mathbf{m}}(t) = \mathbf{A}\mathbf{m}(t) + \mathbf{B}\mathrm{q}(t) \tag{3.2}$$

or, in standard form

$$\begin{aligned}
\dot{\mathbf{m}}(t) &= \mathbf{E^{-1}}\mathbf{A}\mathbf{m}(t) + \mathbf{E^{-1}}\mathbf{B}\mathrm{q}(t) && (3.3)\\
\mathbf{m}(0) &= \mathbf{m_{init}} && (3.4)
\end{aligned}$$

where $\mathbf{m} = [m_{1,1}, m_{2,1}, m_{3,1} \dots m_{N_x,N_y}]$. With $\mathbf{E}$ expressing the volume element $V_{i,j} = \Delta x_i \Delta y_j h$, that is

$$\mathbf{E} = \begin{bmatrix} V_{1,1}\phi\mu c & 0 & \dots & 0 \\ 0 & V_{2,1}\phi\mu c & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & V_{N_x,N_y}\phi\mu c \end{bmatrix} \tag{3.5}$$

then $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix}
-\Lambda_{1+\frac{1}{2},1+\frac{1}{2}} & \alpha_{1+\frac{1}{2},1} & \dots & \beta_{1,1+\frac{1}{2}} & 0 & 0 & \dots & 0 \\
\alpha_{2-\frac{1}{2},1} & -(\Lambda_{2+\frac{1}{2},1+\frac{1}{2}} + \alpha_{2-\frac{1}{2},1}) & \alpha_{1+\frac{1}{2},1} & \dots & \beta_{2,1+\frac{1}{2}} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \ddots & \dots & \ddots & \dots & 0 \\
\beta_{1,2-\frac{1}{2}} & \dots & \ddots & \ddots & \ddots & \dots & \ddots & 0 \\
0 & \ddots & \dots & \ddots & \ddots & \ddots & \dots & \beta_{N_x,N_y-1+\frac{1}{2}} \\
0 & \dots & \ddots & \dots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & \ddots & \ddots & \ddots & \ddots & \alpha_{N_x-1+\frac{1}{2},N_y} \\
0 & \dots & 0 & \dots & \beta_{N_x,N_y-\frac{1}{2}} & \dots & \alpha_{N_x-\frac{1}{2},N_y} & -\Lambda_{N_x-\frac{1}{2},N_y-\frac{1}{2}}
\end{bmatrix} \tag{3.6}$$

$\alpha_{i+\frac{1}{2},j} = h\Delta y_j(\frac{k_{i+\frac{1}{2},j}}{x_{i+1}-x_i}), i = 1, 2, \ldots, N_x - 1$

$\alpha_{i-\frac{1}{2},j} = h\Delta y_j(\frac{k_{i-\frac{1}{2},j}}{x_i-x_{i-1}}), i = 2, 3, \ldots, N_x$

$\beta_{i,j+\frac{1}{2}} = h\Delta x_i(\frac{k_{i,j+\frac{1}{2}}}{y_{j+1}-y_j}), j = 1, 2, \ldots, N_y - 1$

$\beta_{i,j-\frac{1}{2}} = h\Delta x_i(\frac{k_{i,j-\frac{1}{2}}}{y_j-y_{j-1}}), j = 2, 3, \ldots, N_y$

$\Lambda_{i\pm\frac{1}{2},j\pm\frac{1}{2}} = \alpha_{i\pm\frac{1}{2},j} + \beta_{i,j\pm\frac{1}{2}}$

where $\Delta x_{i\pm\frac{1}{2}}$ and $\Delta y_{j\pm\frac{1}{2}}$ is defined as

$\Delta x_{i\pm\frac{1}{2}} = \pm(x_{i\pm1} - x_i)$

$\Delta y_{j\pm\frac{1}{2}} = \pm(y_{j\pm1} - y_j)$

Note the pentadiagonal form of the matrix **A**, as the model is in 2D. **B** is defined as

$$\mathbf{B} = \begin{bmatrix} -\frac{2Tp_{sc}}{T_{sc}} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.7}$$

3.1.2 Well inflow model

In Nordsveen (2011), two different well inflow models were developed and discussed; one two dimensional areal flow model, and one radial inflow model. It was shown that the radial inflow model, given as

$$q_{sc} = \frac{hkT_{sc}}{2Tp_{sc}(ln\frac{R_{wellbore}}{R_{tubing}} + s)}(m_{1,1} - m_{wf}) \tag{3.8}$$

gave the most reasonable response in terms of stability and convergence. s is the skin factor. Equation (3.8) can also be represented as

$$q(t) = \alpha w \left[m_1 - m_{wf}\right] \tag{3.9}$$

where

$$w = \frac{hkT_{sc}}{2Tp_{sc}(ln\frac{R_{wellbore}}{R_{tubing}} + s)} \tag{3.10}$$

and $\alpha \in \{0, 1\}$ is the valve setting. α can be controlled directly, and represents the control variable in applying switchings.

3.2 Liquid loading

All producing gas wells needs to adress the problem of liquid loading. This phenomenon arises when the gas rate falls below a critical rate, known as the *Turner rate* or *minimum rate to lift* (Turner et al., 1969). This causes the gas to lose its transportation energy, and is hence uncapable of transporting co-producing liquids to the surface. The liquids may be present in the gas as condensate, water or oil (Whitson et al., 2012). If the liquids are not transported to the surface, they will accumulate in the well. This may cause severe slugging, or kill the well completely.

Turner et al. (1969) calculated the minimum gas flow velocity based on an analysis of flow conditions necessary to remove the largest drops of liquid that can exist in the well, leading to the following expressions

$$v_t = 20.4\frac{\sqrt[4]{\sigma\Delta\rho}}{\sqrt{\rho_g}} \quad \text{(field units)} \tag{3.11a}$$

$$v_t = 6.2\frac{\sqrt[4]{\sigma\Delta\rho}}{\sqrt{\rho_g}} \quad \text{(SI units)} \tag{3.11b}$$

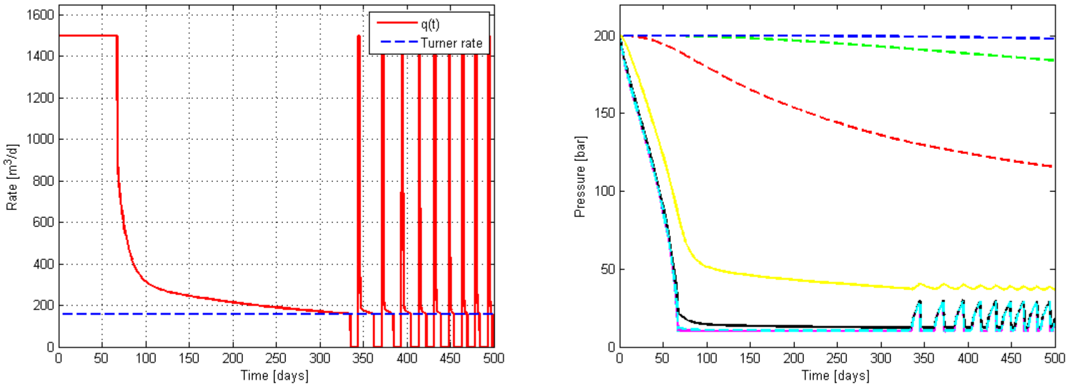
The expressions in (3.11) can be used to estimate the minimum rate to lift, q_{gc}:

$$q_{gc} = 3.06\frac{A}{B_g}v_t \tag{3.12}$$

A is the cross-sectional tubing area and B_g is the volume formation factor. The rate is given in (MMscf/day). Converting the expression to SI units (m^3/day), (3.12) can be written as

$$q_{gc} = 86.649 \cdot 10^3\frac{ApT_{sc}}{p_{sc}TZ}v_t \tag{3.13}$$

Figure 3.1 shows an illustrative example of what happens when the flow hits the critical rate, with a constant shut-in time of 10 days; the well is switched



(a) Flow rate profile

(b) Pressure build-up

Figure 3.1: The effect on flow rate and pressure when a well is shut down

off, which results in a zero production rate. This causes the pressure to build up. When the well is swtiched on after being shut-in, the build-up of pressure causes more gas to be extracted from the well.

3.3 Wellhead choke control

Wellhead chokes are often used to control and monitor flow rates from the well it is installed on, and to protect the reservoir and surface equipment from pressure fluctuations and slugging. There are mainly two types of wellhead chokes (Abou-Kassem et al., 2006)

- Fixed chokes
- Adjustable chokes

The bottomhole pressure can be determined for any given wellhead pressure. Thus, controlling the wellhead pressure is the same as controlling the bottomhole pressure.

The reservoir model in (3.2) is only a simplified model. Because of the assumption of stationary flow, the dynamics when a well is started up is lost. The result is a high initial production when a well is opened. In Knudsen (2010b) and Nordsveen (2011), this was bypassed by imposing a maximum flow.

$$q(t) \leq q_{max} \tag{3.14}$$

However, in a optimization scheme, this constraint causes a non-linear problem formulation, in the form of the minimum function

$$q(t) = \min(q_{max}, \alpha w(m_{1,1} - m_{wf}) \tag{3.15}$$

This may be rewritten as linear constraints instead, leading to a mixed integer linear program. However, both additional variables and new constraints are introduced. This is obviously a drawback when it comes to optimization; more variables causes a higher run-time.

3.3.1 Dynamic model

Because of the extra variables and constraints introduced when rewriting the nonlinearity, it is in interest to avoid this, and instead introduce a dynamic model of the wellhead pressure.

By assuming that the bottomhole pressure immediately after the well is opened equals the pressure in gridblock 1, and then drops to m_{wf} constant value, a dynamic model of the bottomhole pressure can be written as

$$\dot{m}_{wf} = -\frac{1}{T}m_{wf} + \tilde{m} + \alpha c \tag{3.16}$$

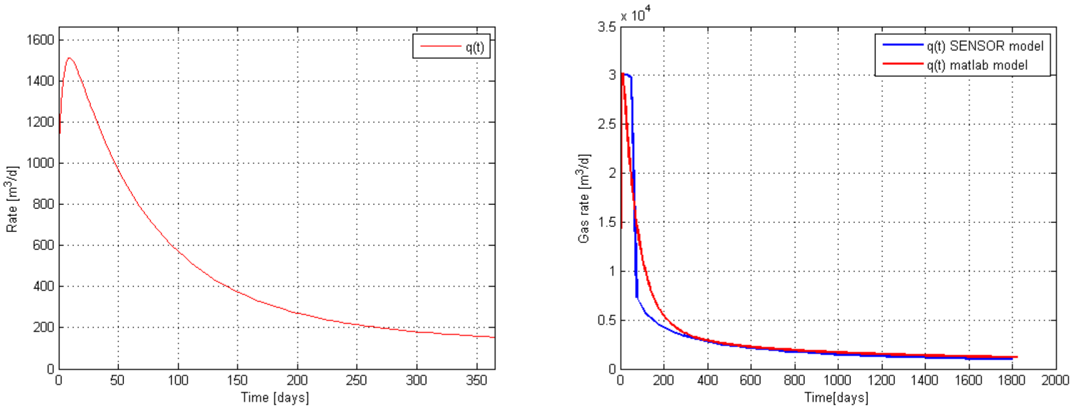
where T, $\tilde{m}$ and c is tuning parameters. α is the control variable, which means that the parameter c is only applied when the well is open.

3.3.2 Flow rate profiles

Without switchings

Figure 3.2b shows the flow profile of the dynamic model in (3.16) plotted against a SENSOR[1] simulation. The profile is more realistic in the start-up region; the production starts at zero, not at q_{max}. Some accuracy is lost in the transition area compared to the model with constant m_{wf}. However, the initial plateu, caused by the saturation on the flow, is avoided.

[1] Coats Engineering, Inc., see http://www.coatsengineering.com



(a) 360 days

(b) 1800 days

Figure 3.2: Flow rate with stepwise change in bottomhole pressure - comparison between SENSOR model and matlab model

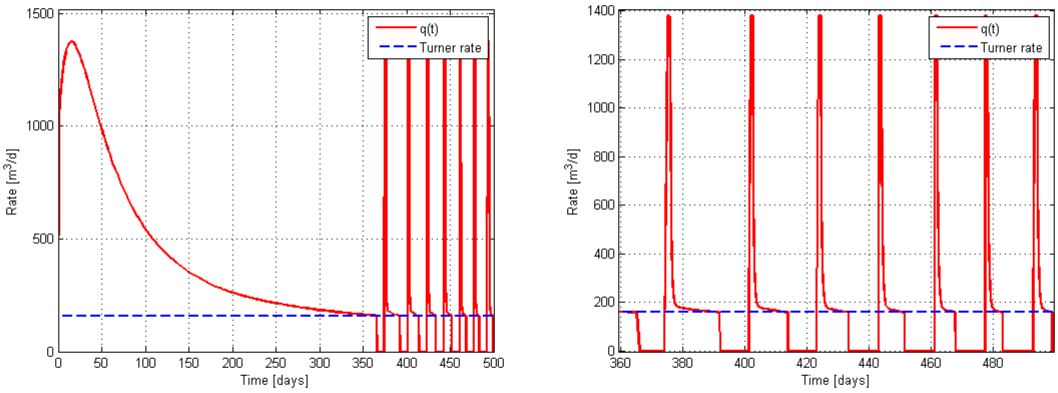
With switchings

Although the model in (3.16) works well when a well is opened, the unmodelled dynamics in (3.1) is not handled well in (3.16). The high initial production when a well is re-opened after being closed is still present. This can be seen in figure 3.3. A maximum flow restriction $q_{max} = 1600$ m^3/s had to be included, or else the magnitude of the flow when the well is reopened is in the magnitude of 10^8. Different values of the tuning paramteres were tried out, without luck. In addition, an PI controller were tried out, to make the bottomhole pressure follow the pressure buildup in gridblock 1. The result did not improve much.

3.4 Shale gas metering

Although the model in (3.1) is assumed to be single phase, the hydraulic fracturing process involves injection of large amounts of water. The gas would thus in reality act as multiphase (two-phase) flow. Although it is beyond the scope of this thesis to take two-phase flow into account, it is of importance in gas metering in practice.

The general definition of multiphase flow is (Falcone et al., 2010): "Multiphase flows consists of the simultaneous passage in a system of a stream composed of two or more phases." Much focus has been put into multiphase



(a) 500 days

(b) 140 days

Figure 3.3: Flow rate with stepwise change in bottomhole pressure and switchings

flow metering (MFM) the last 30 years. MFM means measurement of every single phase in a multiphase flow. Today, MFM includes also wet gas metering and metering of heavy oils. The benefits of MFM in the oil and gas industry includes reduced costs, better reservoir management and improved layout of production facilities (Falcone et al., 2002).

3.4.1 Wet gas metering

When a shale field experiences late life production near the critical flow rate, condense may build up in the reservoir. The immediate effect of this is wetting of the dry gas - small amounts of liquid in the gas. This causes flow metering to be difficult. Wet gas metering is a relatively new technology, and the industry has struggled to meter individual well rates with adequate accuracy (Izgec et al., 2008; Steven, 2002).

In a cluster of producing fields of small or remote gas fields, wet gas flow measurement is required before mixing the gas (Lide et al., 2007). Two ways are normally employed;

- Multiphase flow metering
- Dry gas metering with corrections based presence of liquid in the gas stream

The last method requires a priori knowledge of the liquid flow, and this has to be obtained in another way. However, the most favoured wet gas metering method is using MFM. No single instrument exists that will measure individual flow rates directly. One approach is by using the *venturi effect*, and is hence referred to as *venturi flow meters*. The venturi effect is the pressure drop experienced when the fluid flows through a narrowed section of a pipe. This type of metering seems to obtain relatively good results (Busaidi and Bhaskaran, 2003). The advantage with such a flow meter is that no separation of phases is required. To extract the gas flow rate using venturitype flow meters, a correlation needs to be applied. In lack of other alternatives, the correlation has traditionally been carried out by using Orifice plate meter correlations (Steven, 2002).

3.4.2 Distributed temperature sensing

The flow rate metering needs to be complemented with flow rate estimation, both because of validation of measurement quality and to fill in missing information in infrequent measurements. This is done by using temperature readings from sensors in the wellbore; Distributed temperature sensing (DTS) have gained increased popularity the last few years, and successful results have been reported from gas wells ranging from hydraulic fractured low-permeability tight reservoirs to high permeability reservoirs (Johnson et al., 2006). Izgec et al. (2010) presents two different methods for flow rate estimation based on temperature data; *Entire wellbore method* and *single point method*. The first method requires both the wellhead pressure and the wellhead temperature, while the second method uses only transient temperature formulations at a single point in the wellbore.

The *entire wellbore method* depends on a relationship between the production rate and the temperature of the flowing fluid. Because of temperature differences between the wellbore fluid and the surroundings, a heat exchange is taking place.

In the *single point method*, a unique temperature profile can be extracted from the total amount of fluid passing through a given point in the wellbore. The difference between the measured and the calculated temperature is beeing minimized by continuously iterating on the mass flow rate by a Newton-Raphson method for mass flux. This method works off in any point in the wellbore from temperature measured in that point.

More info can be found in Izgec et al. (2008), Izgec et al. (2010) and Kabir et al. (2008).

Chapter 4

Optimization of shale gas production

This chapter presents the open-loop and closed-loop optimization formulations for use on multiple shale gas wells in a joint production setting. An implicit time discretization of the state space model is also carried out. At the end, a discussion of possible disturbances in the system is given.

4.1 Production facilities

The production optimization considered in this report are performed on multiple wells. Figure 4.1 illustrates the simplified layout of the production facilities for multiple wells in a joint production setting. The surface facilities includes both separation units and compressors to compress the gas and increase the pressure enough to make the gas flow from the production fields to the market via pipelines. The separation units and compressors will, however, not be included in the problem formulation. The chokes seen in the figure are modelled as on/off valves.

4.2 Time discretization

For the state space model to be applicable in a optimization scheme, it has to be time discretized. As (3.2) is a *stiff* system, it is difficult to solve with explicit methods (Egeland and Gravdahl (2003)).

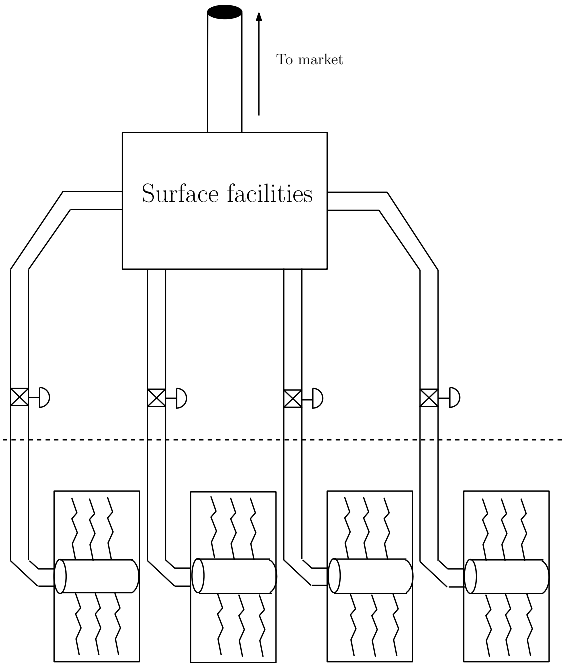


Figure 4.1: Production facilities in shale gas optimization

The discretization is done using the backward Euler method. This method is an implicit method, but only of first order. Methods like Crank-Nicolson, a second order method, would be more accurate. However, such a method might be more demanding when applied into a optimization scheme, as it requires smaller timesteps, and oscillations are introduced. This is further discussed in section 7.1.4.

Consider the implicit Euler method in general form

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \mathbf{f}(\mathbf{y}_{n+1}, t_{n+1}) \tag{4.1}$$

where h is the time step; a solution computed for $(t_0, t_1, \ldots, t_N)$ with $n = 1, 2, \ldots, N$, gives $h = t_{n+1} - t_n$. Reformulating (4.1) gives

$$\frac{\mathbf{m_{n+1}} - \mathbf{m_n}}{h} = \mathbf{f}(\mathbf{m_{n+1}}, \mathrm{q_{n+1}}) \tag{4.2}$$

Applying this to the state space model in (3.3) gives

$$\begin{aligned} \frac{\mathbf{m_{n+1}} - \mathbf{m_n}}{h} &= \mathbf{E^{-1}Am_{n+1}} + \mathbf{E^{-1}B}\mathrm{q_{n+1}} && (4.3)\\ (\mathbf{I} - \mathbf{E^{-1}A}h)\mathbf{m_{n+1}} &= \mathbf{m_n} + \mathbf{E^{-1}B}h\mathrm{q_{n+1}} && (4.4) \end{aligned}$$

Reformulating the terms $(\mathbf{I} - \mathbf{E^{-1}A}h)$ and $\mathbf{E^{-1}B}h$ as

$$(\mathbf{I} - \mathbf{E^{-1}A}h) = \mathbf{A_d} \tag{4.5}$$

and

$$\mathbf{E^{-1}B}h = \mathbf{B_d} \tag{4.6}$$

the full discrete state space system can be written as

$$\begin{aligned} \mathbf{A_d}^k \mathbf{m_{n+1}}^k &= \mathbf{m_n}^k + \mathbf{B_d}^k \mathrm{q}^{\mathrm{k}}_{\mathrm{n+1}} && (4.7)\\ \mathbf{m^k_0} &= \mathbf{m^k_{init}} && (4.8)\\ q^k_n &= \alpha^k w^k [m_{1,n} - m_{wf}] && (4.9) \end{aligned}$$

where $n = 0, 1, 2, \ldots, N - 1$ is the time step, and $k = 1, 2, \ldots, N_w$ is the number of wells.

Any choice of the time step h is possible, as the implicit Euler's method is A-stable (Egeland and Gravdahl (2003)).

4.3 Open-loop formulation

In this section, an open-loop problem is formulated and analyzed. This section is based on Knudsen (2010b).

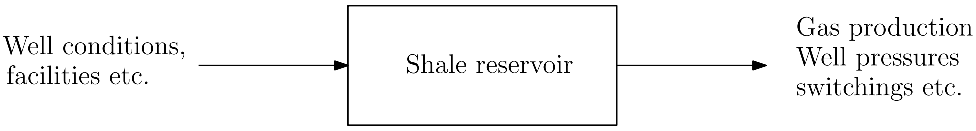


Figure 4.2: Illustration of open-loop optimization in shale gas

Figure 4.2 shows graphically how the process of the open-loop problem works. Well conditions, type of surface facilities etc. determines how much gas is produced, based on some given cost function and constraints, and without feedback.

4.3.1 The objective function

Production optimization in the oil and gas industry is typically driven by the means of maximizing the *net present value* (NPV). This often includes operational expenditures (OPEX) and captial expenditures (CAPEX). The optimization here will, however, be formulated as a short-term scheduling problem, where the objective is to follow a given reference rate, possible time-varying, while avoiding liquid loading in the well. The objective function can be formulated as in Knudsen et al. (2012) as

$$\max \sum_{n=0}^{N-1} \left[\sum_{k=1}^{N_w} c_g q_n^k - W_1 q_{upper} - W_2 q_{lower} \right] h \tag{4.10}$$

where c_g is the gas price per unit volume, denoted in US dollars , q is the gas, and q_{upper} and q_{lower} are auxilliary variables. W_1 and W_2 are used as penalty factors, by punishing either too much or too less, respectively. These parameters are given in US dollars, and may be selected based on gas sales contract specifications (Knudsen et al., 2012). Table 4.1 displays the notation used in the objective function.

Table 4.1: Indicies, sets and data used in the objective function

(a) Indicies		(b) Sets	
k	well index	N	Number of time steps
n	time step	N_m	Number of grid blocks
		N_w	Number of wells

OPEX and CAPEX

The objective function in (4.10) does not include operational expenditures (OPEX) or captial expenditures (CAPEX), as the scheduling problem is defined to track a given reference. Maximizing the net present value, however, will require an analysis of both OPEX and CAPEX into the objective function. A small presentation of these expressions therefore follows

OPEX

OPEX will include both on-site costs of operating the well, and downstream processing of the gas. These costs are to a large extend related to the cost of compressing the gas. By assuming a reciprocating compressor, which also leads to the assumption of an isentropic process, the following expression yields the power consumption in the compressor (Guo et al. (2007))

$$P_s = \frac{k_c}{k_c - 1} P_1 Q_1 \left[\left(\frac{P_2}{P_1} \right)^{\frac{k_c-1}{k_c}} - 1 \right] \tag{4.11}$$

where k_c is the gas specific ratio, P_1 is the inlet pressure to the compressor, P_2 is a constant downhole pressure, and Q_1 is the volumetric flow rate into the compressor. Q_1 equals the gas flow rate at each timestep.

CAPEX

Wright (2008) estimated the capital costs (CAPEX) to be $3,400,000 for horizontal well drill and completion, and $1,000,000 for horizontal well refrac, based on an approximation of current conditions in the Barnett field. These numbers are only mean, and additional costs related to restimulation of the shale at late-life production is not included. Introducing CAPEX into the model would only be valid for long-term optimization, and only short-term optimization is done here.

4.3.2 Constraints

Flow rate

Turner rate and maximum constraint due to saturation. As mentioned in Nordsveen (2011), the maximum constraint is needed in order to ensure realistic gas rates when the well is opened.

$$\begin{aligned} q_n^k &\leq q_{max}^k \quad (4.12) \\ q_n^k &\geq q_{gc} \quad (4.13) \end{aligned}$$

The maximum flow rate constraint (4.12) will cause infeasibility. This could be overcome by a dynamical model for the bottomhole pressure m_{wf}, which was discussed in chapter 3. The disadvantage with such a solution is the addition of an extra control variable. The advantage is the reduction in number of variables which is introduced by the reformulation in (4.14). However, no solution for this was found.

Instead, the infeasibility is avoided by reformulating the constraint as a nonlinear, continuous function, where the flow rate is determined by a minimum value imposed by either the maximum saturation value, or the well inflow model in equation (3.8). This is formulated mathematically as (Knudsen (2010b))

$$q_n^k = min\left\{q_{max}^k, \alpha_n^k w(m_{1,1,n}^k - m_{wf}^k)\right\} \tag{4.14}$$

Equation (4.14) is a *nonlinear* function, and included in a optimization scheme, (4.14) would lead to a *mixed integer nonlinear program* (MINLP). To ensure a efficient solution strategy, this expression thus needs to be reformulated as a linear constraint. This will be done in a later section.

As (4.12) will cause an infeasibility in the solution, so will (4.13) when shut-ins are applied. As α is defined as a binary variable, (4.13) can be rewritten as (Knudsen (2010b))

$$\alpha_n^k q_{gc} \leq q_n^k \tag{4.15}$$

Summarized, the constraints on the flow rate for well number k in discrete time is

$$\begin{aligned} q_n^k &= min\left\{q_{max}^k, \alpha_n^k w(m_{1,n}^k - m_{wf}^k)\right\} && (4.16)\\ \alpha_n^k q_{gc}^k &\leq q_n^k && (4.17)\end{aligned}$$

Reference trajectory

In a real production setting, there is often a requested daily production to be fullfilled. This desired production may be given as a time varying reference $q_{ref} = q_{ref}(t)$, which reflects the operators choice of changing the reference in relation to varying demands. This reference is included in the problem formulation as a constraint, that is

$$\sum_{k=1}^{n_w} q_n^k - q_{upper} + q_{lower} = q_{ref,n} \tag{4.18}$$

where q_{lower} and q_{upper} balances the total rate.

Switchings

The focus on shut-ins in this thesis is on the optimal shut-in time during production, and to compare the results from the open- and closed-loop performance.

To be able to set the minimum shutin time allowed, the number of switchings during the optimization needs to be counted. A new variable η_n^k is thus introduced. It is defined as

$$\left|\alpha_n^k - \alpha_{n-1}^k\right| \leq \eta_n^j \tag{4.19}$$

The minimum shutin period can thus be formulated in the additional constraint

$$\sum_{u=n+1}^{n+L} \eta_u^k \leq 1 - \eta_n^k \tag{4.20}$$

where L represents the minimum shutin period.

4.3.3 Problem formulation

Based on the discussion of objective function and constraints, the problem formulation can therefore be summarized as the following mixed integer non-linear program

$$\max \sum_{n=0}^{N-1} \left[\sum_{k=1}^{N_w} c_g q_n^k - W1 q_{upper} - W2 q_{lower} \right] h \tag{4.21}$$

subject to

$$\begin{aligned}
\sum_{k=1}^{N_w} q_n^k - q_{upper} + q_{lower} &= q_{ref,n} && (4.22a)\\
\mathbf{A_d}^k \mathbf{m_{n+1}}^k &= \mathbf{m_n}^k + \mathbf{B_d}^k \mathrm{q}_{\mathrm{n+1}}^{\mathrm{k}} && (4.22b)\\
q_n^k &= min\left\{q_{max}^k, \alpha_n^k w(m_{1,1,n}^k - m_{wf}^k)\right\} && (4.22c)\\
\alpha_n^k q_{gc}^k &\leq q_n^k && (4.22d)\\
\left|\alpha_n^k - \alpha_{n-1}^k\right| &\leq \eta_n^j && (4.22e)\\
\sum_{u=n+1}^{n+L} \eta_u^k &\leq 1 - \eta_n^k && (4.22f)\\
\alpha_n^k &\in \{0, 1\} && (4.22g)
\end{aligned}$$

4.4 Model predictive control

In the last section, optimal shutin times were calculated open-loop, without any feedback. This is unlikely when an optimization scheme is deployed in practice. In the sense of fracking, reservoirs are often diverse, and model uncertainty do occure. Because of the nature of shales, reservoir parameters are likely to change during a wells lifetime. In addition, certain wells may experience sudden shut-ins, to avoid liquid loading, or because surface facilities fails. Model updates are therefore required.

The main idea behind a model predictive controller, as described in chapter 2, is to use an internal model to predict future response of the system. These predictions are optimized in the sense of maximizing tracking performance, and handling the constraints. Only the first sample of the optimal prediction is applied to the plant at time t. In the next time step, a new sequence of optimal predictions are obtained, which are replacing the old ones. This provides the desired closed-loop control feature. The benefits with feedback may include enhanced reservoir understanding and economic performance (Bemporad and Morari, 1999; Saputelli et al., 2003).

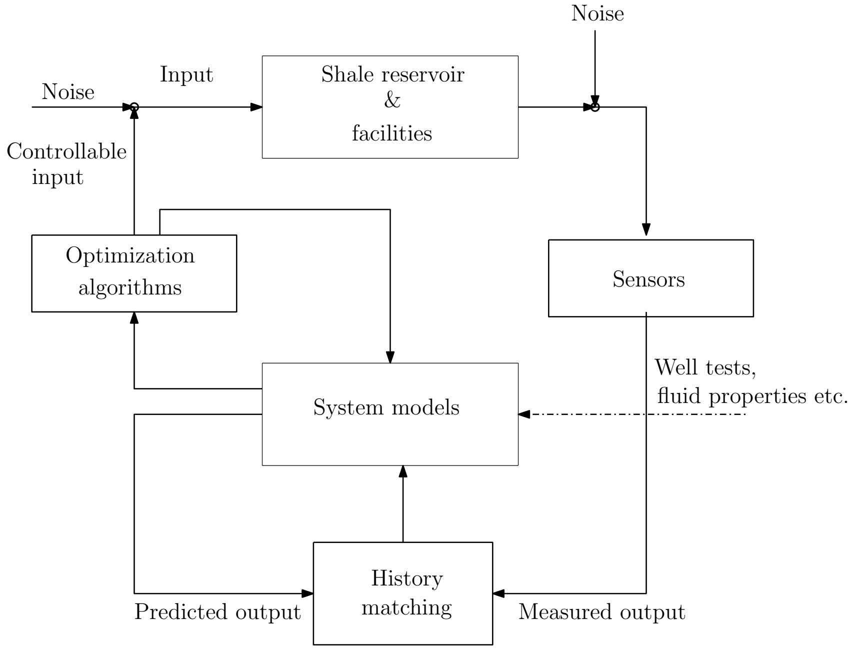


Figure 4.3: Closed-loop control process. Based on Jansen et al. (2009)

Figure 4.3 displays the essential elements in a model predictive control setting. The system models may typically include the well inflow model, geological models and wellbore models. Sensors are measuring the processes (i.e. wellhead pressure, gas flow rate, temperature). The system models are updated in accordance to the measured output through history matching, such that the uncertainties in the models are minimized. Based on this, the switching rates are optimized, and fed back to the system. This is done for each time step over a prediction horizon.

4.4.1 Problem formulation

In the open-loop cost function (4.21), the objective is to maximize the shale gas production in terms of a given reference rate. Auxiliary variables are introduced to balance the total produced rate for each time step, and the penalty parameters W_1 and W_2 are used to punish either too much, or too less gas compared to the reference rate.

Using the same objective function in the closed-loop case, this can be seen as a simple form of MPC tracking, with the objective function maximized. However, it is linear, and not a quadratic function, as in a standard MPC tracking problem. There are several advantages using this linear formulation, as stated by Knudsen et al. (2012); MILP solvers are more efficient in reusing excisting solutions and basis for consecutive nodes in branch-and-bound and branch-and-cut algorithms, and the objective function preserves the economical interpretation. The worst case solution time of an mixed integer quadratic program (MIQP) is in fact exponentially dependant on the number of integer variables in the problem (Bemporad and Morari, 1999).

The use of the same objective function with feedback from the flow rate will of course require that the flow rate is measureable (c.f section 3.4). In addition, it requires feedback from the states and α. The states are here assumed measureable. As they in practice are not, they need to be estimated. This is discussed in section 7.4.2.

The problem formulation is similar to (4.21) and (4.22), and can be formulated as

$$\max \quad J(n) = \sum_{i=0}^{N_p-1} \left[\sum_{k=1}^{n_w} c_g q_{n+i|n}^k - W_1 q_{upper} - W_2 q_{lower} \right] h \tag{4.23}$$

subject to

$$\begin{aligned}
\sum_{k=1}^{n_w} q^k_{n+i|n} - q_{upper|n} + q_{lower|n} &= q_{ref,n|n} && \text{(4.24a)}\\
\mathbf{A_d}^k \mathbf{m_{n+1|n}}^k &= \mathbf{m_{n|n}}^k + \mathbf{B_d}^k \mathrm{q}^{\mathrm{k}}_{\mathrm{n+1|n}} && \text{(4.24b)}\\
q^k_{n+i|n} &= min\left\{q^k_{max}, \alpha^k_{n+i|n} w(m^k_{1,n+i|n} - m^k_{wf})\right\} && \text{(4.24c)}\\
\alpha^k_{n+i|n} q^k_{gc} &\leq q^k_{n+i|n} && \text{(4.24d)}\\
\left|\alpha^k_{n+i|n} - \alpha^k_{n+i-1|n}\right| &\leq \eta^j_{n+i|n} && \text{(4.24e)}\\
\sum_{u=n+i+1|n}^{n+i+L|n} \eta^k_u &\leq 1 - \eta^k_{n+i|n} && \text{(4.24f)}\\
\alpha^k_{n+i|n} &\in \{0,1\} && \text{(4.24g)}
\end{aligned}$$

where the notation $(n+i|n)$ means that the variable depends on the conditions at time n.

4.4.2 Disturbances

Processes never operate under optimal conditions. Disturbances are always a part of a real plant process. Introduction of disturbances often give rise to great economic losses, and it is therefore important to include disturbances in the modelling of the production.

Disturbances can be divided into two categories (Lindholm et al., 2011)

- Local disturbances
- Plant-wide disturbances

Local disturbances represents disturbances at a certain place in a process, and will not necessary cause economic losses over the entire site. Plant-wide disturbances will, however, affect the economic performance of the production. An example of this is pressure drop in the wellhead.

Further, the causes of disturbances can be divided into the following

- Personnel
- Material
- Equipment

Material can be divided into raw materials (i.e. materials that is part of a reaction process), and utilities (i.e. not part of the final product, but important for operation). Personnel disturbances includes actions taken by operators, equipment includes wear and tear of production facilities (Lindholm et al., 2011).

The main disturbances which will be included in the closed-loop production setting can be divided into the following

- Planned shut-ins made by personnel
- Sudden shut-ins caused by failure in production facilities

Planned shut-ins are actions taken if maintainance are to be performed on the wellsite. In this report, the disturbances will be static (i.e. occuring at a preset time). This as a means of comparing performance of the open-loop and closed-loop problem.

Model errors are also a form of disturbance, especially uncertainty in reservoir parameters. This will be discussed further in section 7.4.3.

Chapter 5

Implementation

This chapter presents some implementation strategies for the problems in the previous chapter. Firstly, a presentation of the solution software and scaling is given. Reformulation of the nonlinearities in (4.22) is performed, to make a transition from MINLP to MILP. At the end, some aspects concering solution parameters in CPLEX are presented.

5.1 Solution software & hardware

The MILP problem is implemented in ILOG CPLEX Optimization studio v.12.3[1]. CPLEX is a state-of-the-art tool for solving linear and convex optimization problems. This includes LP, QP, QCP, MIP and MIQP problems. A big benefit with the CPLEX software is the possibility of using parallel optimizers on hardware platforms with parallel processors (IBM, 2009). On large models, this may be beneficial with regards to the CPU/solution time.

CPLEX builds on a robust branc & cut algorithm, mentioned in section (2.1.2). The solution might generate several types of cuts. See IBM (2009) for more information.

Lower and upper bounds of the objective values are essential in MILP problems, and are gradually improving as a new solution is found. CPLEX logs this *gap*, and is defined as

$$\frac{(\text{best integer} - \text{best node}) * \text{objective}}{\mid \text{best integer} \mid + 10^{-9}} \tag{5.1}$$

[1] IBM®, see www.ibm.com

The software is run on a Dell Optiplex 32-bit Windows 7 machine with 3 GB RAM and Intel Core2 Duo 3.00GHz CPU.

5.2 Scaling

The initial pseudopressures, in the range of 10^{19}, will eventually cause numerical problems in the optimization. The pseudopressures are therefore diagonally scaled as

$$\mathbf{m} = \mathbf{M}_s \tilde{\mathbf{m}} \tag{5.2}$$

where $\mathbf{M}_s$ is the diagonally scaling matrix

$$\mathbf{M}_s = \begin{bmatrix} m_s & 0 & \dots & 0 \\ 0 & m_s & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & m_s \end{bmatrix} \tag{5.3}$$

The discrete state space formulation in (4.7) is thus updated as follows to include (5.2)

$$\mathbf{A_d}^k \mathbf{M}_s \tilde{\mathbf{m}}_{n+1}^k = \mathbf{M}_s \tilde{\mathbf{m}}_n^k + \mathbf{B_d}^k \mathrm{q}_{\mathrm{n+1}}^{\mathrm{k}} \tag{5.4}$$

5.3 Reformulation of non-linearities

The problem with the constraints from (4.22) is that the min-function

$$q_n^k = min\{q_n^{max}, \alpha_n^k w(m_{1,n}^k - m_{wf})\} \tag{5.5}$$

is in fact a non-continuous, non-linear function. The non-linearities lies in both the min-function, and the product $\alpha_n^k m_{1,n}^k$. Knudsen (2010b) implemented this reformulation *exact*, based on FICO (2009), as

$$\begin{aligned}
\alpha_n wm_{wf} &\leq \tilde{q}_{nl,n} \leq \alpha_n wm_{max} && (5.6a)\\
wm_{wf}(1-\alpha_n) &\leq wm_{1,1,n} - \tilde{q}_{nl,n} \leq wm_{max}(1-\alpha_n) && (5.6b)\\
\tilde{q}_n &= \tilde{q}_{nl,n} - \alpha_n wm_{wf} && (5.6c)\\
q_n &\leq q_{max} && (5.6d)\\
q_n &\leq \tilde{q}_n && (5.6e)\\
q_n &\geq q_{max} d_n && (5.6f)\\
q_n &\geq \tilde{q}_n - \tilde{q}_n^{\infty} d_n && (5.6g)
\end{aligned}$$

where the non-linear part $\tilde{q}_{nl,n} = \alpha_n wm_{1,1,n}$, m_{max} is the maximum pressure value in gridblock 1, and d_n is a binary value, whose value is 1 if q_{max} is the maximum value, 0 if not. Note however, that rewriting the min-function as (5.6), nine inequality constraints and one binary variable are introduced for *each* well at each timestep.

In addition, the absolute value

$$\left|\alpha_n^k - \alpha_{n-1}^k\right| \leq \eta_n^k \tag{5.7}$$

is simply formulated as

$$\begin{aligned}
\alpha_n^k - \alpha_{n-1}^k &\leq \eta_n^k && (5.8a)\\
\alpha_{n-1}^k - \alpha_n^k &\leq \eta_n^k && (5.8b)
\end{aligned}$$

5.4 Shutin period

In the open-loop case, one need not worry about the minimum shutin period. The value is set at the start of the optimization, and is static over the time horizon. In the closed-loop case, however, the minimum shutin period will be dynamic, and the remaining shutin time needs to be saved for each iteration.

This will affect the minimum shutin constraint

$$\sum_{u=n+1}^{n+L} \eta_u^k \leq 1 - \eta_n^k \tag{5.9}$$

which is implemented as

$$\sum_{u=n+1}^{n+L-r} \eta_u^k \leq 1 - \eta_n^k, \quad n = r+1, L+r+2, \ldots, r(L+1)+N \tag{5.10}$$

with $r = 0, 1, 2, \ldots, L$.

Note that for both the open-loop and closed-loop optimizations, the shut-in times are variable, in contrast to the simulations in section 3.2. One of the main objectives are in fact to find optimal shut-in times for each well in the optimization, to maximize production.

5.5 Branch-and-bound solution parameters

In CPLEX, the user are allowed to change a number of parameters to control factors like solution methods, scaling, tolerances, limits, MIP strategies. Some of these are manually changed. The reader is referred to IBM (2009) for further details on parameters in CPLEX.

5.5.1 Gap tolerance

The duality gap in CPLEX, as defined in (5.1), makes it possible to view the solution progress in real time. This shows the relative deviation from optimality of the currently best integer solution. The default relative gap tolerance in CPLEX is 1e-04. When the gap (5.1) falls below this value, the optimization stops.

The default value may, however, cause a long runtime when applied in both open- and closed-loop optimization. In general, for a big problem, the decrease of the upper bound is slow when the number of nodes are high. A larger gap may be acceptable, especially for difficult models with uncertainty in the input data. Thus, the MIP gap tolerance is set to terminate at 20%[2].

5.5.2 Solution limit

Another termination parameter in MILP is the number of integer solutions found before stopping. As default, this limit is set to 2100000000. To reduce

[2] with the parameter *epgap*

the runtime of the optimization, the maximum number of integer solution is therefore set to 10[3].

5.5.3 Integrality tolerance

The default integrality tolerance 1e-05. This value will, especially for the closed-loop system, give unrealistic flow rates and switchings. This is thus reduced to a value between 1e-06 and 5e-06 [4]. Note that by lowering this value, the solution time increases.

5.5.4 MIP cuts

As stated in section 5.1, CPLEX builds on a branch & cut algorithm, and alot of differ CPLEX have a large number of cuts which may me added to the model to cut away noninteger solutions that would otherwise be a solution of the continuous relaxation, and are automatically handled by the optimizer. These are so-called cutting plane methods. Adding cuts to the model may reduce the number of branching steps needed to solve the problem, and thus reducing the runtime.

However, cuts may also, at occasions, increase the runtime, or even give infeasible solutions. The user are able to turn off cuts selectively. This is, however, not done. More information about the different cut limit parameters can be found in IBM (2009).

[3] with the parameter *intsollim*

[4] with the parameter *epint*

Chapter 6

Results

This chapter contains results and analysis of both the open-loop and closed-loop optimization on multiple wells presented in chapter 4. First, a base case is presented in which start points are extracted from. Then the open-loop and closed-loop multiwell optimization are simulated both with and without disturbances, and compared to each other.

6.1 Base case

The choice of total number of gridblocks in the model affects the solution time when applied in a optimization scheme. More gridblocks gives mores accurate results, to few makes the model lose accuracy. The number of gridblocks is based on a trade-off between accuracy and solution time.

Each well is divided into $N_x = 9$ and $N_y = 3$, in total $N_m = 27$ gridblocks. This is the smallest number of gridblocks in which the accuracy is still good.

6.2 Well and gas specifications

Table 6.3 shows the initial gas price used in the optimization, retrieved 26.04.2012. The results from the simulations are naturally affected by the present gas price. This will be further discussed in chapter 7.

Table 6.1: Reservoir geometry

Parameter	Units	SI units
$\mathrm{x}_{\mathrm{fracture}}$	0.5 ft	0.15 m
$\mathrm{x}_{\mathrm{reservoir}}$	100 ft	30.48 m
$\mathrm{y}_{\mathrm{halflength}}$	100 ft	30.48 m
h	200 ft	60.96 m
$\mathrm{R}_{\mathrm{tubing}}$	0.2 ft	0.06 m
$\mathrm{R}_{\mathrm{wellbore}}$	0.33 ft	0.1 m

Table 6.2: Reservoir properties

Parameter	Units	SI units
μ	$2.02 \cdot 10^{-5}$ Pa.s	$2.02 \cdot 10^{-3}$ cp
ϕ	5 %	-
c	$8.46 \cdot 10^{-8}$ Pa^{-1}	$5.83 \cdot 10^{-4}$ psi^{-1}
k_o	100 mD	-
k_i	0.000075 mD	-
T	200 ft	60.96 m
p_{init}	200 bar	2901 psi
$p_{bottomhole}$	10 bar	145 psi

Table 6.3: Initial gas price

Label	Field units	SI units
c_g	\$2.12 / mcf	\$0.075 / m^3

The optimization is done on the half-fracture model in (3.2). Due to symmetry, the production from one fracture is found by multiplying the results with 2. It is assumed that one well contains 10 fractures, such that the results needs to be multiplied by 20.

The prediction time is set to 10 days, with a time step of 12 hours. All wells operate under constant bottomhole pressure (10 bar), and they have the same minimum rate $q_{gc} = 233.28$ m^3/day. Note that the minimum rate is for the half-fracture. The rate for *one* well is $q_{gc} = 4665.6$ m^3/day.

To get a meaningful short-term optimization problem, the base case from Nordsveen (2011) is simulated for 900 days prior to the optimization. A number of shut-ins are also included in the simulation, and grid pseudopressures are collected at random points in the simulation. These pseudopressures are used as initial grid pressures in the optimization.

Table 6.4: States of the wells

Well number	1	2	3	4	5	6
q_{max} (m^3/day)	1477.44	1503.36	1607.04	1365.12	1512	1356.48
State	C	C	C	P	C	P

Table 6.4 shows q_{max} and the state of each well. P means that the well is production, C means that the well is closed. The q_{max} values are randomly varying ±10 % around the base case value $q_{max} = 1500$ m^3/day.

6.3 Multiwell open-loop scheduling

This section presents the results from the open-loop problem presented in section 4.3.3, with the reformulation of the non-linearities from section 5.3. It is thus solved as a MILP formulation using CPLEX. Only short-term production planning for six wells are considered.

6.3.1 Optimization parameters

Table 6.7 displays the optimization parameters used. The weights are chosen to achieve as close tracking of the target rate as possible. L is the minimum shut-in time for the well to build up enough pressure to produce at a rate higher than the critical rate.

Table 6.5: Initial optimization parameters

Label	Value
W_1	$2.5c_g$
W_2	$2.5c_g$
L	1 day
q_{ref}	49019 (m^3/day)

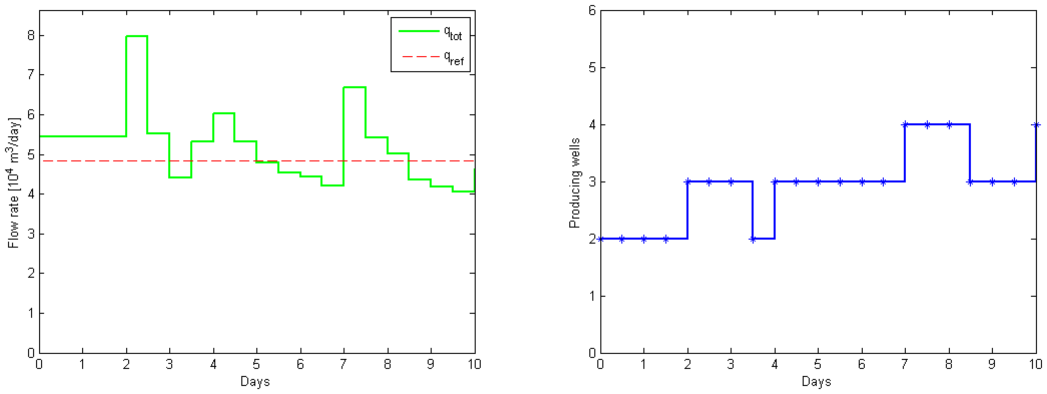
6.3.2 Simulation results

Without disturbances

The optimizer for the open-loop problem without disturbances terminates after 52 minutes, with a duality gap of 26%. Figure 6.1 displays the total flow rate along with the total switchings. The production is close to constant the first two days, as the producing wells have a production at the maximum allowed rate. At day 2, the flow rate shoots of, with a high excessive production from the reference rate. This high peak in production is caused by the three wells producing at its maximum rate. However, for two wells producing at maximum rate, the peak would be closer to the rate. This unexpected behavior of having 3 wells open may be caused by some tolerance errors. The reason could also be the weighting factors. However, it was experienced that increasing W_1 and W_2, the optimizer had problems finding solutions for some time steps.

Looking at the flow rate profiles in figure A.1, well number 6 seems to not follow the decline profile that should be expected from figure 3.1. This discrepancy is caused by the integrality tolerance. Setting this value lower is helping on the flow rate profile, but the high peak is still present.

The shut-ins builds the pressure up, but the wells are only able to produce at rates higher than the minimum rate for some days (cf figure 3.1). The flow rates for each well are displayed in figure A.1 in appendix A.



(a) Total flow rate

(b) Total switchings

Figure 6.1: Open-loop optimization

The number of wells controlled are limited to 6, and α is the only control variable. The result is a relative large deviation from the reference trajectory. More wells would possibly make the total flow rate follow the reference better. Note that 6 wells is chosen based on getting an efficient optimization scheme without too long run time.

As q_{gc} and L are only *minimum* values, the wells might be shut-in at rates higher than q_{gc}, and have shut-in periods longer than L.

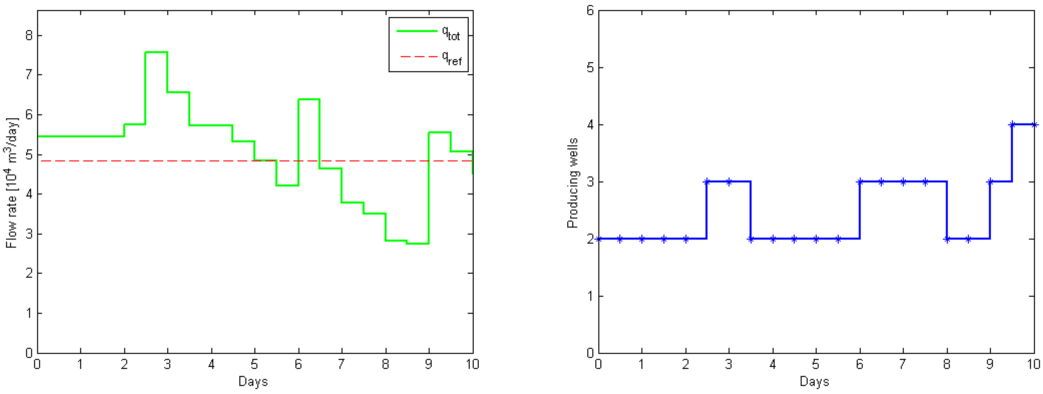
With disturbances

Table 6.6: Disturbances

Well number	Closing time	Closed for
2	Day 7	2 days
3	Day 4	2 days

Table 6.6 shows the disturbances used in the optimization. Well number 2 and 3 are closed down at day 7 and 4, respectively. Although these disturbances in reality would occur at random times, they are set static to be able to compare the results with the closed-loop optimization. The disturbances could therefore also be considered as planned shut-ins.

By introducing disturbances into the system, the flow rate profile changes. Figure 6.2a displays the total flow rate. The day before the first dirsturbance kicks in, more wells are activated to prevent loss in production. The result is extra surplus gas. In fact, at most almost 60% extra gas from the reference rate is produced. The flow rate then drops until day 6, when there is no disturbances in the system, and the same trend is seen as for the first disturbance; a large deviation from the reference rate, possibly to prevent too much loss in production. When the second disturbance is introduced at day 7, the flow rate drops again, and experiences a major production loss at the end of the disturbance period. At the end of the time horizon, the wells are able to produce at a rate with small deviatons from the reference. The individual flow rates for each well is shown in figure A.2 in appendix A. Note that there are no switchings in well 6. This is somewhat unexpected, considering the deviation in production before the first disturbance, but may have connection with the weighting factors. As mentioned, setting the factors higher resulted in no solution for a couple of timesteps. In addition, limited wells to control (with only one control variable) makes the deviation significant.



(a) Total flow rate

(b) Total switchings

Figure 6.2: Open-loop optimization with disturbances

6.4 Closed-loop reservoir management

This section presents results based on an implemented MILP model predictive controller developed in section 4.4.1. The controller is implemented in ILOG CPLEX by using *flow control*. Flow control in CPLEX includes a *main* function to orchestrate model, model data and solving. The main file used in the optimizations are attached in appendix B

6.4.1 Optimization parameters

To be able to see the potential improvements of MPC in shale gas production, the same well and gas specifications as presented in section 6.3 are used. This also includes the weighting factors and minimum shutin period.

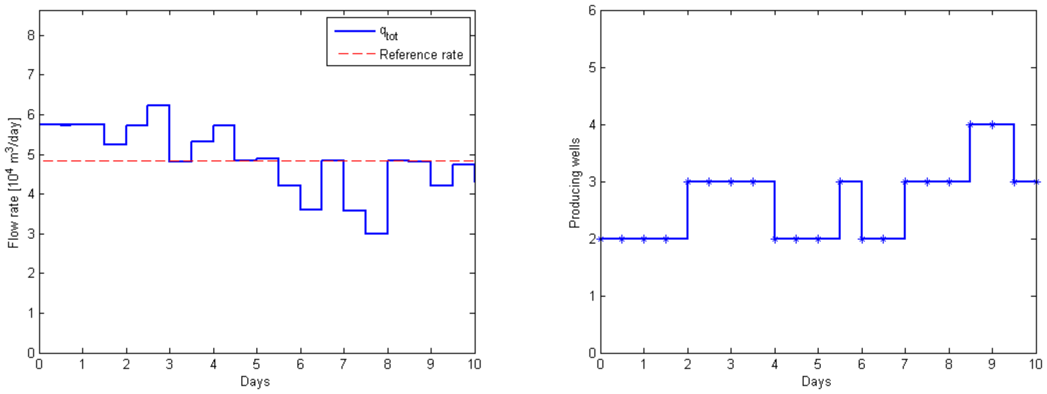
Table 6.7: Initial optimization parameters

Label	Value
W_1	$2.5c_g$
W_2	$2.5c_g$
L	1 day
q_{ref}	49019 (m^3/day)
Prediction horizon	10 days
Control horizon	10 days

6.4.2 Simulation results

Without disturbances

The total flow rate for the closed-loop case is shown in figure 6.3a. Comparing the figure with 6.1a, it is observed that the high peak between day 2 and 3, and at day 6 is reduced. The discrepancy between the highest peak and reference value in figure 6.1a is close to 60%, in figure 6.3a the discrepancy is close to 40%. This confirms an improvement when it comes to tracking the reference. However, the rate experiences a drop of the total rate at the end of the prediction horizon, so that the tracking is not as tight as expected.



(a) Total flow rate

(b) Total switchings

Figure 6.3: Closed-loop optimization

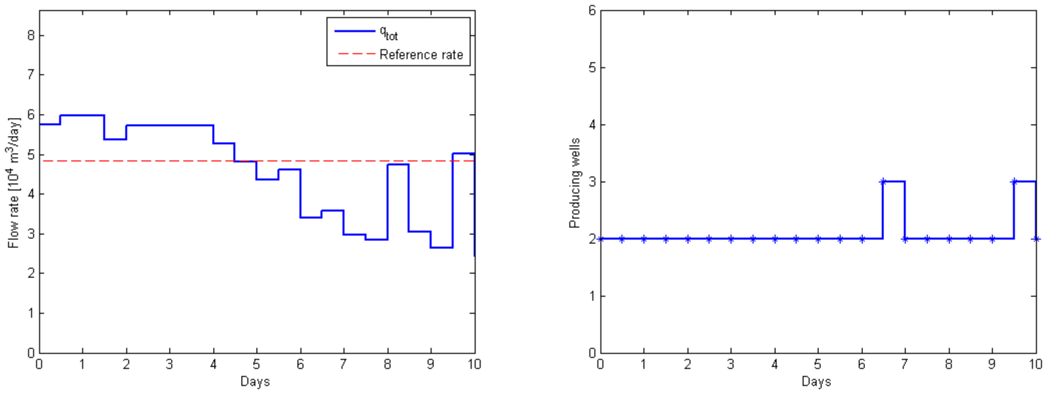
With disturbances

The most interesting comparison between the open-loop and the closed-loop formulation is by the introduction of disturbances into the system. In section 6.3.2, disturbances in the open-loop formulations revealed the somehow suprising high peaks in production a day before the wells were shut off. This is surprising, as the open-loop problem has no feedback, and are not able to predict forward in time. Optimizing the current timestep is only based on the initial information at time $t = 0$.

The receding horizon idea in MPC makes it possible to predict forward in time, a mechanism that has the potential of foresee sudden or planned shut-ins of wells in the production network, and thus suppesing them.

Figure 6.4 displays total flow rate and the number of producing wells during the control horizon. Again, the simulations confirms that closing the loop helps reducing the peaks in production. Comparing figure 6.3a and 6.4a, it is seen that with disturbances in the system, the rate stays close to constant the first 3 and a half days, after which it drops when well 3 is closed down. This suggest that the MPC controller is able to forsee shutdowns of some wells, and adapt the rate to it. However, the drop in production at the end of the horizon, observed in both open- and closed-loop simulations with and without disturbances, is not improved by much, and may suggest that the optimizer struggles to keep the reference rate, and that it may be too high.

Comparing figures 6.2a and 6.4a, it is observed that the closed-loop formulation actually performs better than open-loop, in the sense of reduced peaks and tighter tracking of the reference rate. It was expected that the tracking of the reference should be even tighter at the end of the horizon. Increasing the prediction horizon should yield better results. However, it is believed that the hardware used in the simulation is not sufficient, as CPLEX runs out of memory for prediction horizons longer than the control horizon. An upgrade of the hardware is highly recommended.



(a) Total flow rate

(b) Total switchings

Figure 6.4: Closed-loop optimization with disturbances

Chapter 7

Discussion

This chapter further discuss the results in the previous chapters. Optimization aspects are discussed, and drawbacks and advantages of open-loop against a MPC formulation are reviewed. State estimation and model updates are also reviewed.

7.1 Model applicability

The model (3.1), developed in Nordsveen (2011), is a two-dimensional model, building on work done in Knudsen (2010a,b). This model was analyzed and discussed in Nordsveen (2011), and no thorough discussion on the model itself will be done here. However, it is interesting and important to discuss some aspects around model size and model definition.

7.1.1 Model size

Discretizing (3.1) yields the state space formulation in (3.2). The model is valid for two dimensional flow, and the number of gridblocks has the potential of growing very large; expanding the number of gridpoints N_y with 1 in y-direction gives N_x[1] extra gridblocks.

In total, 27 gridblocks were chosen in the optimization ($N_x = 9$ and $N_y = 3$). A smaller gridblock size would affect the accuracy of the model, a bigger would make it more accurate. A small number was chosen to make the

[1] N_x is the number of gridpoints in x-direction

optimization as effective as possible, but one should consider increase this number in practice.

7.1.2 Dynamic wellhead model

As mentioned in section 3.3, the maximum flow rate imposed to the flow rate yields a non-linear *min*-function in the optimization scheme, shown in (3.15). Reformulating this to exact linear constraints, which yields a MILP formulation of the problem, introduces 9 inequality constraints plus 1 binary variable for each well at each timestep. Obviously, this increases the problem size dramatically, making the optimization less effective.

A dynamic wellhead model of first order (3.16) was tried out in chapter 3. The response of the flow did appear to be good at the startup region when no shutins were applied, shown in figure 3.2a; the flow builds up from zero production, and avoids beeing saturated due to the tuned model. This is more realistic than having the flow start at the maximum flow rate q_{max}. When applying shutins, the startup region the first 320 days is still good. However, after a shutin, the flow goes off, an still gets saturated at q_{max}.

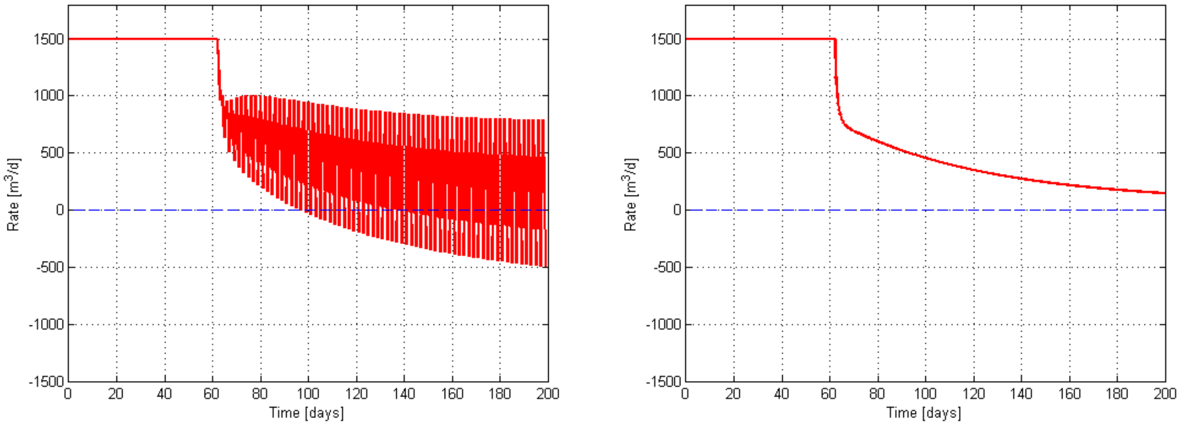
The unrealistic high flow rate after a shutin is due to the unmodelled dynamic in the model (3.1). (3.16) is only of first order. Developing a model with higher order may provide some extra dynamics, and thus overcome this problem to a certain extent. This is left out as further work.

7.1.3 Liquid loading

By use of the Turner rate in section 3.2, it is possible to determine when a well must be shut in to avoid accumulation in the well. This is formulated in the constraint

$$\alpha_n^k q_{gc}^k \leq q_n^k \tag{7.1}$$

for both the open- and closed-loop problem. The advantage of using this lower bound on the flow rate is that there is no need to include water saturation in the reservoir model (3.1) (Knudsen et al., 2012). Note that the use of shutins as a means of building up the pressure is not intended to replace the well stimulation by hydraulic fracturing, but comes in addition. In this way, the OPEX may be reduced, as fracturing is an expensive process.



(a) Timestep 24 hours

(b) Timestep 0.1 hours

Figure 7.1: Oscillations are introduced by implementing the Crank-Nicolson method as discretization scheme, but are reduced with smaller timesteps

7.1.4 Numerical errors

Both for time and spatial discretization, numerical errors are introduced, which limits the achievable accuracy of the optimization. There are tradeoffs between model size and sample size in optimization, and the solution time for the optimizer.

For the time discretization, a backward (implicit) Euler method was chosen. It is A-stable, which means that any time step can be chosen. However, it is only of order one, and is thus the least accurate discretization method. A second order method, such as the *Crank-Nicolson*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}(\mathbf{f}(\mathbf{y}_n, t_n) + \mathbf{f}(\mathbf{y}_{n+1}, t_{n+1}) \tag{7.2}$$

give more accurate results. It is a stable method (Quarteroni et al., 2010). However, for large time steps, the approximate solution might contain oscillations. Two simulations, both over 200 days, is shown in figure 7.1; figure 7.1a has timestep 24 hours, figure 7.1b has timestep 0.1 hours (6 minutes). The trend is clear; reducing timestep removes the oscillations. However, simulating over 200 days with 6 minutes timestep will substantially increase the solution time, and Crank-Nicolson is therefore not used here.

7.2 Optimization concerns

The emphasis of the implementation strategies in chapter 5 has been to provide good solution quality and reduced solution time. However, there are trade-offs between these two criteria.

7.2.1 Sensitivity

The initial inputs to the model have a big impact on the optimizer. The initial values are grid pseudopressures, maximum possible flow, state of the well, permeability and maximum pseudopressure in gridblock 1. These are the inputs that are most sensitive, both in the form of solution and run time of the optimizer. Much time was spent trying to "tune" the model, i.e. trying to make a realistic case.

For some parameters, CPLEX reported "infeasible or unbounded" solution, caused by conflicts in the model. A conflict is a set of constraints which makes the model infeasible, and removing these makes the problem feasible (IBM, 2009). For instance, a conflict happens when x is defined as both bigger than y, and lower than z, with $y \neq z$. For the open-loop case, the optimizer is able to solve this automatically. For flow control (i.e. the closed-loop case), the optimizer terminates with the message "No solution". The parameters was chosen to avoid the conflicts, and make the optimizer terminate with an optimal solution.

Time horizons

Ideally, future behavior of the process should be optimized over an infinite prediction horizon. In practice this is hard to achieve. The process is often under hard time constraints, and with limiting software/hardware solutions. With large prediction horizons, it was observed that the optimizer had problems improving solutions, and the branch-and-cut tree grew very big. This resulted in out-of-memory error, and the optimizer terminated. A prediction horizon equal the control horizon appeared to show the best effect.

Initial grid pseudopressures

The initial grid pseudopressures are chosen before the optimization, from a simulation of the base case presented in section 6.1. For some initial pressures, especially for low values, the problem became infeasible due to conflicts.

Integrality tolerance

As mentioned in section 5.5.3, CPLEX uses an default integrality tolerance of 1e-05. It was observed, especially in the closed-loop case, that some wells did not follow the decline profile as they should with this value. Some wells appeared to produce at a constant rate. The tolerance was reduced to between 1e-06 and 5e-06 in both the open-loop, and closed-loop case, at the cost of increased run time. Also, for too low tolerance, and especially for high weighting factors, the optimizer was unable to find a solution in the closed-loop case.

7.2.2 Validation

To validate the results from the optimizations, the tracking should be simulated in SENSOR, with optimal shut-in times obtained from CPLEX. Comparing both the open-loop and closed-loop simulations with a simulation based on constant shut-in times, one should see the true potential of tracking performance.

7.3 Model predictive control vs. open-loop optimization

In the open-loop case, the flow rates and the switchings in each wells are defined by the initial information (i.e. initial grid pseudopressures) at time $t = 0$ only. In the closed-loop case, the system response are defined by the feedback from the grid pressures and flow rates at the latest time step. The closed-loop response thus has the potential to change the transient repsonse of the system, which is not possible in the open-loop case (Foss and Jensen, 2011). Because of the receding horizon idea in model predictive control, closed-loop approach is more robust than an open-loop approach (Bemporad and Morari, 1999).

Model predictive control is in fact a mix of both open-loop and closed-loop strategies. The receding horizon idea, with reoptimization at each time step over a certain prediction horizon, provides the feedback to the system. The reoptimization at each time step therefore requires alot of computational work. For systems with short time constants, this may be a problem (Foss and Jensen, 2011). However, for the system considered here, the time constant is long (weeks). The prediction horizon will obviously affect the runtime, and the prediction horizon may be large. A too short horizon may not capture disturbances introduced to the system, and the closed-loop response will in that case not be able to adjust the parameters to reject it.

Introduction of model predictive control would only be of practical interest when there are clear benefits. Such benefits may include (Going et al., 2006)

- Reduced costs and enhanced profits
- Enhanced efficiencies in resources, workflows and equipment
- Improved safety and reliability

The use of closed-loop control control has the potential of reducing the number of personell required on remote sites. Monitoring systems provide data, which is sent to an off-site operational centre. The system can take its own decisions under observation of operators, often based on prequalified alarm conditions and user feedback notifications. This is cost effective and efficient, while the safety is improved for personell.

Closing the loop also makes it possible to predict future production. This is important, as a means of setting up a production plan based on the market, and to ensure that it is in fact possible to meet future demands. This is especially crucial on behalf of governments and investor groups (Going et al., 2006).

The MPC formulation used in this thesis is only "semi-predictive". The objective function only punishes the auxiliary q_{lower} and q_{upper}, and not the actual error ($\sum_{k=0}^{N_w} q_{n+i|n}^k - q_{ref}$). Note that this formulation makes it possible to give independent weights on over- and underproduction. This is similar to the *zone* objective principle, in which the output remains inside specified boundaries (Maciejowski, 2002), except that the zone is very thin. The output is penalized when it leaves the "zone".

The traditional MPC have a quadratic cost function. The MILP MPC in chapter 4 is only linear. A linear cost function was chosen to exploit the efficient MILP solution methods in CPLEX. Actually, for a MIQP problem, the worst case solution time is exponentially dependant on the number of integer variables (Bemporad and Morari, 1999). When this number gets high, so does the solution time.

The MPC controller was implemented in CPLEX, with *optimization programming language* (OPL). OPL have some inconvenient ways of implementing feedback, and a transition to Yalmip[2] in matlab should be further investigated. Yalmip supports CPLEX as solver. Through yalmip, one has better control over each variable. In addition, implementing state estimation and model updates are inconvenient in CPLEX. Note that Yalmip was tried out, but the solver only reported an infeasible problem, and terminated. This should be investigated further.

7.4 Practical optimization aspects

The results from the open- and closed-loop optimization is not valid if it can not be put into practical perspective. Theoretical gas optimization is only an "ideal" case, and real life operations will bring along challenges not included in the formulation.

7.4.1 Gas price

The optimization results will inevitably vary with the gas price. However, it can be difficult to estimate. It is strongly dependant of future demand, and this might give a clue to the future development. 7.2 shows a short time projection of the gas price into the beginning of 2013. The price is not expected to fluctuate much from the current gas price.

7.4.2 State estimation

The model predictive controller considered in this thesis requires that the full state vector of the pseudopressures in the gridblocks is available. These are, however, not measurable states, and the states needs to be estimated, typically with a Kalman filter.

The Kalman filter estimates the states from a process that is containing noise and measurement errors, based on a sequence of measurements. It requires separate knowledge of the disturbances that enters the state estimation and the measurements. The state observer equations is based on the discrete

[2] See Löfberg (2004)

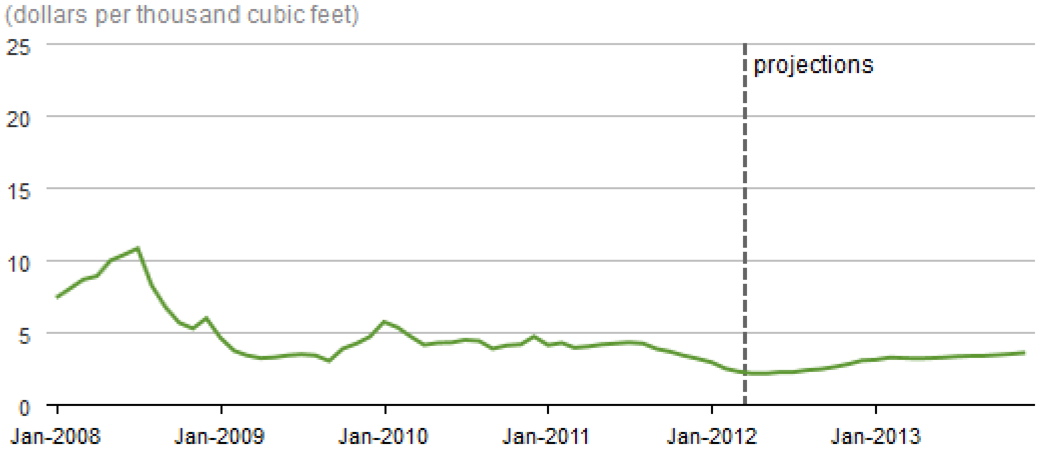


Figure 7.2: U.S. Natural gas prices with short-term projections retrieved from U.S. Energy Information Administration

system in (4.7), with a *blending factor*, Kalman gain, $\mathbf{K}_n$ chosen such that the mean square error of the state estimation is as small as possible, and may be expressed as (Brown and Hwang, 1997; Yan and Bitmead, 2005) (adapted to the model in 4.7)

$$\begin{aligned}
\hat{\mathbf{m}}^k_{n|n} &= \hat{\mathbf{m}}^k_{n|n-1} + \mathbf{K_{n|n}}^k(\mathrm{q}^\mathrm{k}_\mathrm{n|n} - \mathbf{C}^\mathrm{k}\hat{\mathbf{m}}_\mathrm{n|n-1}{}^k) && (7.3)\\
\mathbf{A_d}^k\hat{\mathbf{m}}^k_{n+1|n} &= \hat{\mathbf{m}}^k_{n|n} + \mathbf{B_d}^k\mathrm{q}^\mathrm{k}_\mathrm{n+1|n} && (7.4)\\
\mathbf{K_{n|n}}^k &= \mathbf{\Sigma}^k_{n|n-1}\mathbf{C}^T(\mathbf{C}\mathbf{\Sigma}_{n|n-1}\mathbf{C}^T + \mathbf{\Lambda}_n)^{-1} && (7.5)
\end{aligned}$$

$\mathbf{\Sigma}_n$ is the error covariance matrix, so that $\mathbf{K_n}$ minimizes the terms along the diagonal of $\mathbf{\Sigma}_n$, and is thus optimal in some sense. Note that $\mathrm{m}_{1,1}$ may be found from q through the inflow model in (3.8).

State estimation is an iterative process, and would indeed take up some processing time and further increase the runtime of the closed-loop optimization. However, in this thesis, state estimation is not included into the problem formulation in (4.24). It is left out as further work.

7.4.3 Parameter identification

In the closed-loop formulation in 4.24, it is assumed that the internal model is a perfect match of the process. This is a very unrealistic assumption, as parameters change continuously over time. Model errors are therefore prominent, and model updates are essentially. Referring to figure 4.3, the model updating can be done on the basis of predicted and measured outputs, in addition to well tests, fluid properties and so on, to determine the permeability.

The parameters in state matrix **A** is highly uncertain. The most prominent uncertainty is the permeability in the reservoir, which have only been guessed. This uncertainty can have a large influence on the prediction of future production. It is unlikely that an optimization scheme will be deployed in real life without any form of parameter estimation.

It is in fact common in reservoir engineering to do a so-called *history matching*, or data assimilation, to reduce the uncertainty of the reservoir parameters. History matching means minimizing the difference between a model and the history of the reservoir. This has traditionally been done by hand, by varying different reseroir parameters until a satisfactory match has been found. This is very time consuming and error prone. Gradient-based optimization techniques have therefore been adopted from the oil industry.

The optimization techniques involves automatic variation of the reservoir parameters until it reaches a satisfactory stopping critera (good histoy matching); a cost function is defined and minimized over all possible parameter values (Tavassoli et al., 2004). $\boldsymbol{\theta}$ is defined as a vector with the unknown parameters (i.e. the permeability values), and it is assumed that measurements are available, generated by the system (Zandvliet, 2008)

$$\begin{aligned} \mathbf{m_{n+1}} &= \mathbf{A}(\theta)\mathbf{m}_n + \mathbf{B}(\theta)q_n && (7.6a)\\ \mathbf{m_0} &= \mathbf{m}_{\text{init}} && (7.6b)\\ y_n &= q_n && (7.6c)\end{aligned}$$

As θ includes the permeability in all gridblocks, virtually all parameters non-identifiable, and the problem is ill-posed. The usual way of reducing the effect of this problem is to include the difference between the estimated $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{init}$ in the cost function (Jansen et al., 2009; Zandvliet, 2008). The problem may be expressed as

$$\begin{aligned} min\;\; V(\boldsymbol{\theta}) &= \sum_{n=1}^{N}(y_n - y_n(\boldsymbol{\theta}))\mathbf{P}_y^{-1}(y_n - y_n(\boldsymbol{\theta})) && (7.7a)\\ &+ (\boldsymbol{\theta}_n - \boldsymbol{\theta}_{init})\mathbf{P}_{\theta_n}^{-1}(\boldsymbol{\theta}_n - \boldsymbol{\theta}_{init}) && (7.7b)\\ \mathbf{m_{n+1}} &= \mathbf{A}(\theta)\mathbf{m}_n + \mathbf{B}(\theta)q_n && (7.7c)\\ \mathbf{m_0} &= \mathbf{m}_{\text{init}} && (7.7d)\\ y_n &= q_n && (7.7e)\end{aligned}$$

where $\mathbf{P}_y^{-1}$ and $\mathbf{P}_{\theta_n}^{-1}$ are the inverse of the covariance matrix of the measurements. The cost function is often expressed as a quadratic function such that gradient-based optimization techniques can be used.

These approaches are only a suggestion on how the identification can be done in practice, and have not been tried out in this thesis. This is left out as further work.

Chapter 8

Conclusion

In this thesis, the use of mixed integer model predictive control of shale gas production has been studied. This introduces other challenges than conventional MPC, because of the binary variables used to control the switchings of the wells. The same objective function are used in both the open-loop and closed-loop formulation, with disturbances introduced in the system. The objective is to track a reference rate. Both run time and tracking performance differs, especially for disturbances in the system.

The open-loop and closed-loop formulations shows how short-term production planning with the objective of following a reference rate yields different results in the sense of optimal shut-in times. Closed-loop control yields better tracking performance. Its predictive characteristics reduces deviations from the reference rate. This is especially observed when disturbances are introduced into the system. However, for both the open-loop and closed-loop problems, a decrease in gas flow rate is observed along the time horizons. This drop is hard to improve by the current models.

The optimizations in this thesis assumes a perfect internal model. As reservoir and system parameters reduces over time, model updates are required in order to apply the optimization setting in practice. The model are in reality non-linear. Reformulations yields extra constraints and variables in the problem. Transition to a non-linear model, possible including startup dynamics, might improve the results, but will result in a more demanding model.

Chapter 9

Further work

In this thesis, a model predictve controller has been developed. The controller assumed that the state vector was directly available. This is obviously a false assumption. State estimation was not included into the model, but to give it a practical applicability, this should, in the form of a Kalman filter.

In addition, a perfect internal model was assumed. This model, however, changes over the reservoirs lifetime. System identification and history matching is therefore important to include to update the model, and should be investigated further.

Also, an effort to try to make a dynamic wellhead model, to avoid the nonlinear min-function was a time consuming task. This worked without shut-ins of the wells. The model was only of first order. A model of higher order could lead to better results. The advantages of such a model is reduction in both constraints and decision variables in the MILP model.

List of symbols

Nomenclature

ϕ	Reservoir porosity	[-]
μ	Viscosity	[Pa.s]
ρ	density	[kg/m^3]
A	Cross-sectional area	[m^2]
c	Gas compressibility	[1/Pa]
c_g	Gas price	[\$]
g	Gravity	[m/s^2]
h	Reservoir height	[m]
k	Permeability	[mD]
k_i	Permeability in the inner region	[mD]
k_o	Permeability in the outer region	[mD]
m	Pseudopressure	[Pa/s]
m_{init}	Pseudopressure	[Pa/s]
m_{wf}	Bottomhole pseudopressure	[Pa/s]
N	Time horizon	[-]
N_m	Total number of gridpoints	[-]
N_w	Number of wells	[-]
p	Reservoir grid pressure	[bar]
R_{tubing}	Radius of tubing	[m]
$R_{wellbore}$	Radius of wellbore	[m]
T	Reservoir temperature	[K]
q	Gas flow rate	[m^2/s]
v_t	Minimum gas flow velocity	[m/s]
$x_{fracture}$	Radius of fracture	[m]
$x_{reservoir}$	Reservoir length	[m]
$y_{half-length}$	Fracture half-length	[m]
Z	Gas compressibility factor	[-]

Subscripts

d	Discrete
gc	Gas critical rate
n	Time instant
sc	Standard conditions

Superscripts

k	Well number

Bibliography

Abou-Kassem, J. H., S. M. F. Ali, and M. R. Islam (2006). *Petroleum Reservoir Simulation: A Basic Approach*. Gulf Publishing Company.

Avriel, M. (2003). *Nonlinear programming: Analysis and Methods*. Dover Publications.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica 35*, 407–427.

Brown, R. G. and P. Y. C. Hwang (1997). *Introduction to random signals and applied Kalman filtering*. John Wiley & sons.

Busaidi, K. and H. Bhaskaran (2003). Multiphase flow meters: Experience and assessment in pdo. In *SPE Annual Technical Conference and Exhibition, Colorado*.

Carlson, E. S. and J. C. Mercer (1991). Devonian shale gas production: Mechanisms and simple models. *Journal of Petroleum Technology 1*, 476–482.

Cornuejols, G. (2008). Valid inequalities for mixed integer linear programs. *Mathematical programming 112*, 3–44.

Cutler, C. and B. Ramaker (1980). Duynamic matrix control - a computer control algorithm. In *Proceedings, Joint American Control Conference, San Fransisco*.

Economist (2011, November). Fracking here, fracking there. http://www.economist.com/node/21540256.

Egeland, O. and J. T. Gravdahl (2003). *Modeling and Simulation for Automatic control*. Tapir Trykkeri.

Falcone, G., C. Alimonti, G. F. Hewitt, and B. Harrison (2002). Multiphase flow metering:current trends and future developments. *Journal of Petroleum Technology April*, 77–84.

Falcone, G., G. F. Hewitt, and C. Alimonti (2010). *Multiphase flow metering: principles and applications.* Elsevier.

FICO (2009). *MIP formulations and linearization. Xpress Optimization Suite Documentation.*

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization.* Oxford University Press.

Foss, B. and J. P. Jensen (2011). Performance analysis for closed-loop reservoir management. *SPE Journal 16*(1), 183–190.

Giorgio, V., A. Danilo, D. Marco, and S. Almatasem (2012). Integrated production optimization and surface facilities management through advanced optimization techniques. In *SPE International Production and Operations Conference and Exhibition, Qatar.*

Going, W. S., B. L. Thigpen, P. M. Chok, A. B. Anderson, and G. P. Vachon (2006). Intelligent-well technology: Are we ready for closed-loop control? In *SPE Intelligent Energy Conference and Exhibition, Amsterdam.*

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

Guo, B., W. C. Lyons, and A. Ghalambor (2007). *Petroleum production engineering : a computer-assisted approach.* Gulf Professional Publishing.

Helms, L. (2008). Horizontal drilling. *North Dakota Department of Mineral Resources (DMR) Newsletter 35*, 1–3.

IBM (2009). *IBM ILOG CPLEX V12.1. User's Manual for CPLEX.* IBM.

Imsland, L. (2007). Introduction to model predictive control. Course notes in TTK4135 Optimization and control.

Izgec, B., A. R. Hasan, D. Lin, and C. Kabir (2010). Flow-rate estimation from wellhead-pressure and -temperature data. *SPE Production & Operations 25*, 31–39.

Izgec, B., A. R. Hasan, D. Lin, and C. S. Kabir (2008). Flow-rate estimation from wellhead-pressure and -temperature data. In *SPE Annual Technical Conference and Exhibition, Denver.*

Jansen, J., D. Douma, D. Brouwer, P. V. den Hof, O. Bosgra, and A. Heemink (2009). Closed-loop reservoir management. In *SPE Reservoir simulation symposium, Texas.*

Jenkins, C. D. and C. M. Boyer (2008). Coalbed- and shale-gas reservoirs. *Journal of Petroleum Technology 60(2)*, 92–99.

Johnson, D., J. Sierra, and J. Kaura (2006). Successful flow profiling of gas wells using distributed temperature sensing data. In *SPE Annual Technical Conference and Exhibition, Colorado.*

Kabir, C., B. Izgec, A. Hasan, X. Wang, and J. Lee (2008). Real-time estimation of total flow rate and flow profiling in dts-instrumented wells. In *International Petroleum Technology Conference, Malaysia.*

Karcher, B., E. Aquitaine, F. Giger, and J. Combe (1986). Some practical formulas to predict horizontal well behavoir. In *61st Annual Technical Conference and Exhibiton of the SPE, New Orleans, US.*

Knudsen, B., B. Foss, C. H. Whitson, and A. R. Conn (2012). Target-rate tracking for shale-gas multi-well pads by scheduled shut-ins. In *IFAC Proceedings of the International Symposium on Advanced Control of Chemical Processes*, Singapore.

Knudsen, B. R. (2010a). Modeling and simulation of shale gas reservoirs. Technical report, Norwegian University of Science and Technology. Project work.

Knudsen, B. R. (2010b). Production optimization in shale gas reservoirs. Master's thesis, Norwegian University of Science and Technology.

Lide, F., Z. Tao, and J. Ningde (2007). A comparison of correlations used for venturi wet gas metering in oil and gas industry. *Journal of Petroleum Science and Engineering 57*, 247–256.

Lindholm, A., H. Carlsson, and C. Johnsson (2011). A general method for handling disturbances on utilities in the process industry. In *18th IFAC World Congress, Milano.*

Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan.*

Maciejowski, J. M. (2002). *Predictive control with constraints.* Pearson Education Limited.

Martin-Sanchez, J. M. (1976).

Nordsveen, E. T. (2011). Modelling and simulation of shale gas horizontal wells with hydraulic fractures. Technical report, Norwegian University of Science and Technology.

O'How, D. and G. Kubat (1996). Advances in production optimization and production. In *Annual technical meeting, Calgary.*

Pochet, Y. and L. A. Wolsey (2006). *Production Planning by Mixed Integer Programming.* Springer.

Quarteroni, A., F. Saler, and P. Gervasio (2010). *Scientific computing with matlab and octave.* Springer.

Richalet, J., A. Rault, J. L. Testud, and J. Papon (1978). Model predictive heuristic control: applications to industrial processes. *Automatica 14*, 413:428.

Ridley, M. (2011). The shale gas shock. Technical report, The Global Warming Policy Foundation.

Saputelli, L., M. Nikolaou, and M. J. Economides (2003). Self-learning reservoir management. In *SPE Annual Technical Conference and Exhibition, Denver, USA.*

Steven, R. N. (2002). Wet gas metering with a horizontally mounted venturi meter. *Flow Measurement and Instrumentation 12*, 361–372.

Tavassoli, Z., J. N. Carter, and P. R. King (2004). Errors in history matching. *SPE Journal 9(3)*, 352–361.

Turner, R. G., M. G. Hubbard, and A. E. Dukler (1969). Analysis and prediction of minimum flow rate for the continuous removal of liquids from gas wells. *Journal of Petroleum Technology 21(11)*, 1475–1482.

Warren, J. E. and P. J. Root (1963). The behavior of naturally fractured reservoirs. *SPEJ SPE 426*, 245–255.

Whitson, C. H., S. D. Rahmawati, and A. Juell (2012). Cyclic shut-in eliminates liquid-loading in gas wells. In *SPE/EAGE European Unconventional Resources Conference and Exhibition, Austria.*

Wolsey, L. A. (1998). *Integer programming.* John Wiley & Sons, New York.

Wright, J. D. (2008). Economic evaluation of shale gas reservoirs. In *SPE Shale Gas Production Conference, Texas, U.S.*

Yan, J. and R. R. Bitmead (2005). Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica 41*, 595–604.

Zandvliet, M. J. (2008). *Model-based lifecycle optimization of well lications and production settings in petroleum reservoirs.* Ph. D. thesis, Delft University of Technology.

Appendix A

Additional plots

A.1 Open-loop optimization

Figure A.1 shows the individual flow rates for each well from section 6.3.2 without disturbances in the system. Figure A.2 shows the flow rates when disturbances are present in the system.

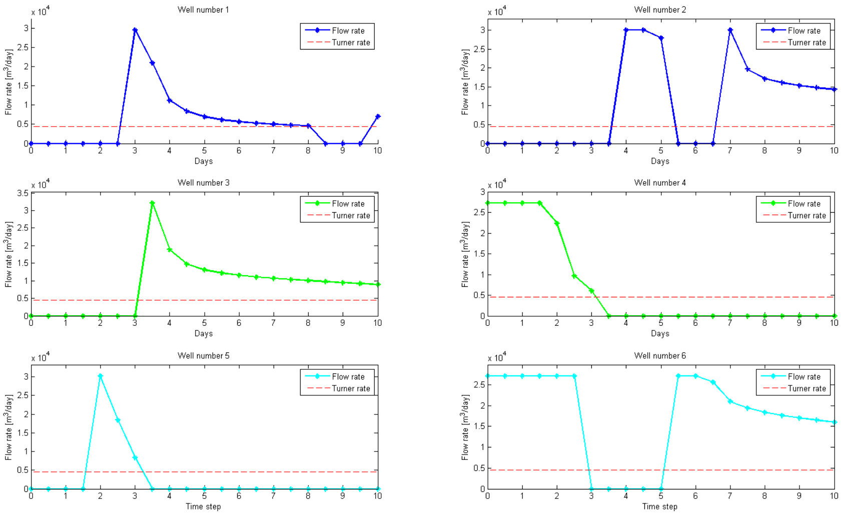


Figure A.1: Flow rate for each well in open-loop optimization

The switchings of each well without disturbances is shown in figure A.3, A.4 shows the switchings with disturbances.

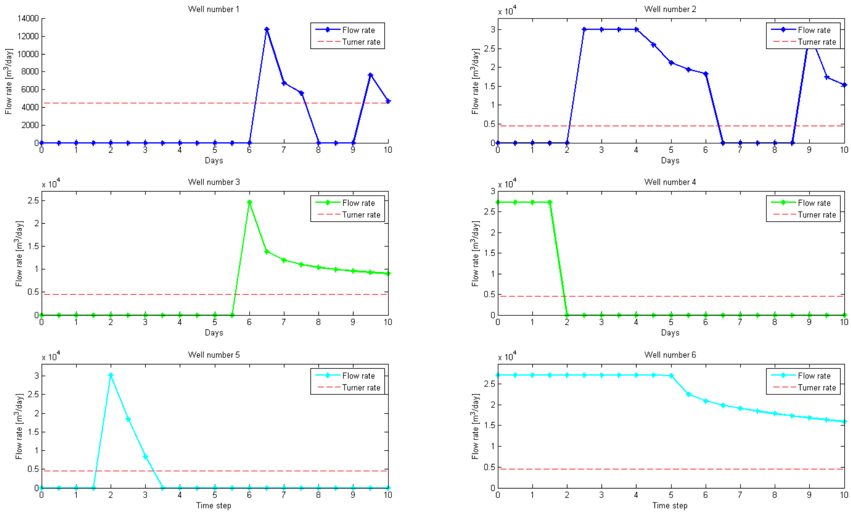


Figure A.2: Flow rate for each well in open-loop optimization with disturbances

A.2 Closed-loop control

Figure A.1 shows the individual flow rates for each well from section 6.3.2 without disturbances in the system.

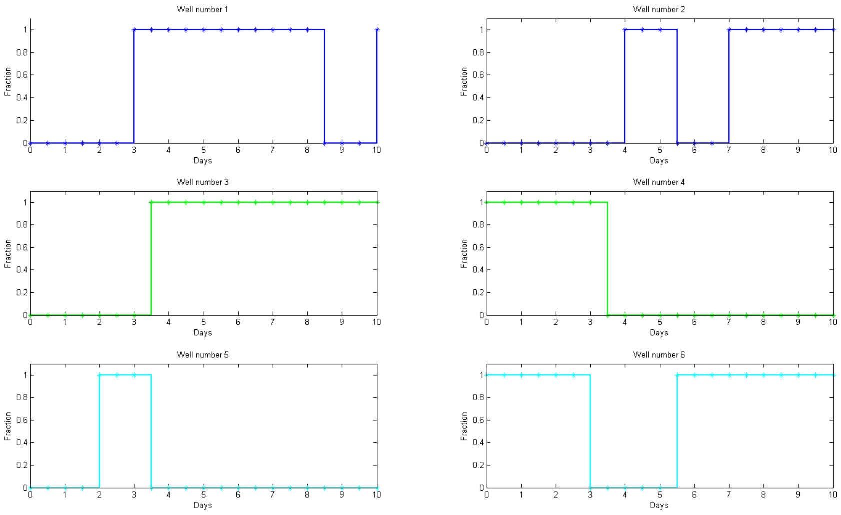


Figure A.3: Switching pattern in each well in open-loop optimization

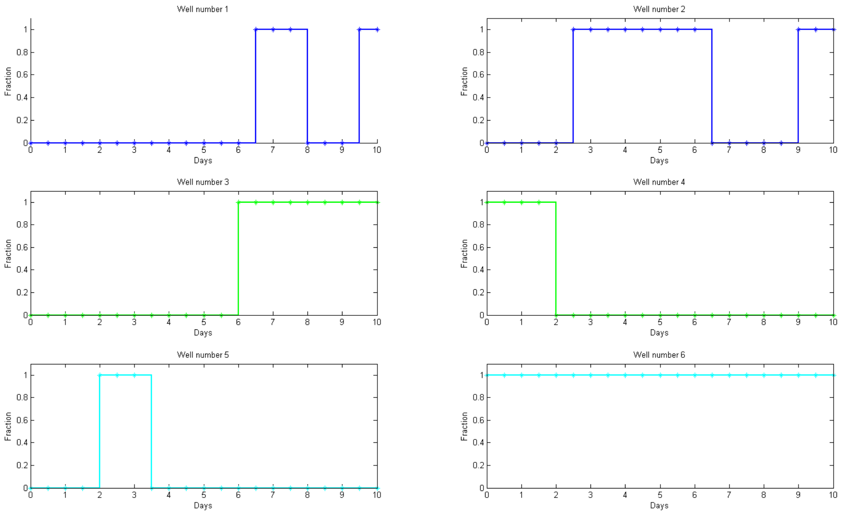


Figure A.4: Switching pattern in each well in open-loop optimization with disturbances

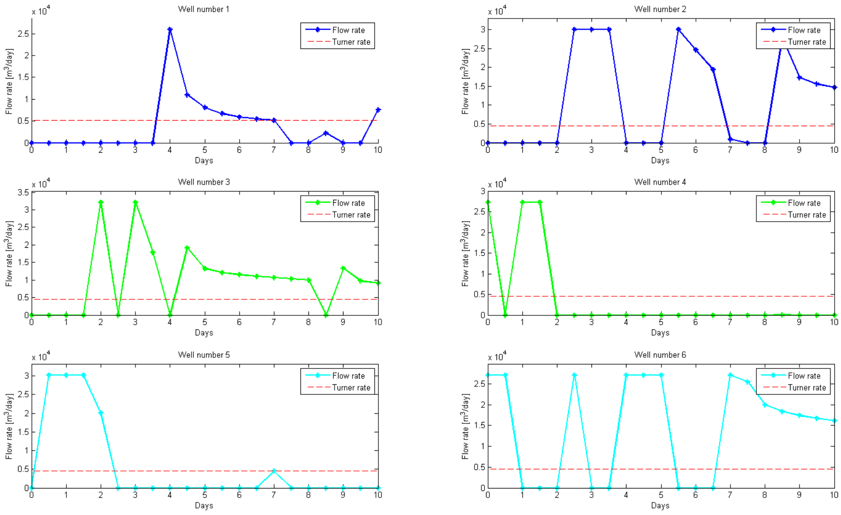


Figure A.5: Flow rate for each well in closed-loop optimization

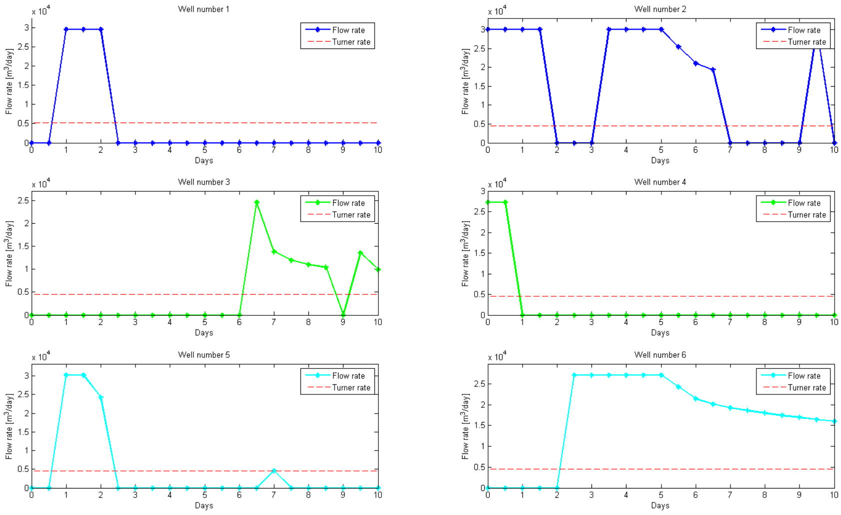


Figure A.6: Flow rates for closed-loop optimization with disturbances

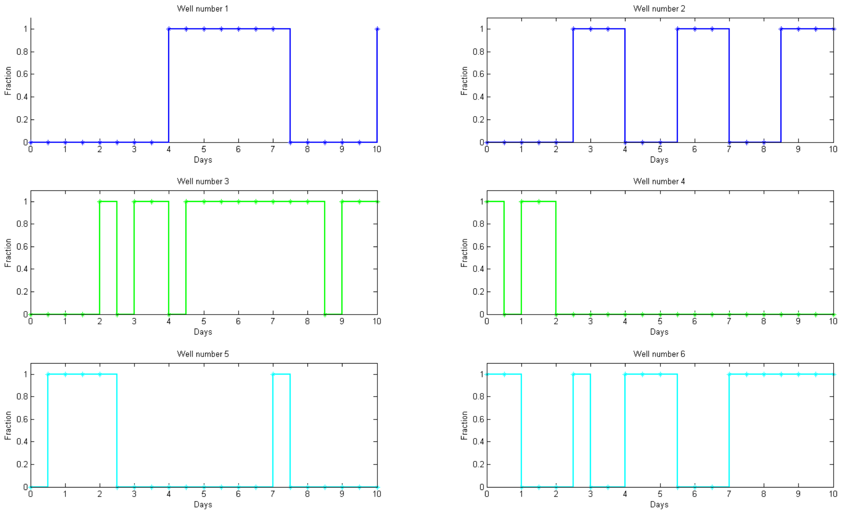


Figure A.7: Switching pattern for closed-loop optimization

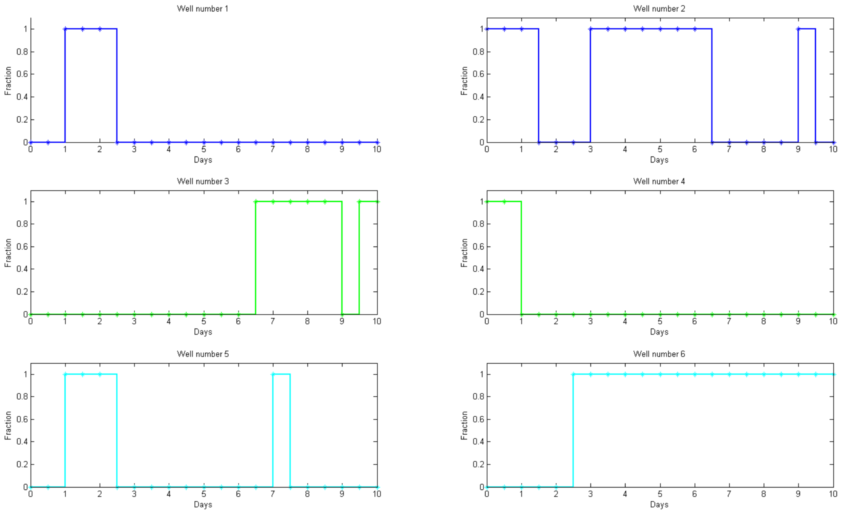


Figure A.8: Switching pattern for closed-loop optimization

Appendix B

Program script

B.1 CPLEX main file

Below is the main file from the flow control used in CPLEX to implement the model predictive controller.

```
main
{

var status = 0;
thisOplModel.generate();
var CurrSol = thisOplModel;
var shutinTime = CurrSol.l;

// Initial pseudopressures
var m1 = CurrSol.m_cl1;
var m2 = CurrSol.m_cl2;
var m3 = CurrSol.m_cl3;
var m4 = CurrSol.m_cl4;
var m5 = CurrSol.m_cl5;
var m6 = CurrSol.m_cl6;

var feedback = 2;
var constSI = 2;  //Predefined shutin time = 1 days
var SI = 0;

//Control horizon
var Nu = 23;
```

```
// Define counting variables
var States = CurrSol.Nm;
var best;
var curr = Infinity;
var tot_q = 0;
var tot_alpha = 0;
var iter = 1;

// Define output files
var ofile2 = new IloOplOutputFile(filename);
ofile2.writeln("Timestep = ", Nu-1);
ofile2.writeln("Referance = ", CurrSol.tot_rate);
var alphafile = new IloOplOutputFile(filename);
alphafile.writeln("Timestep = ", CurrSol.N1);

var Well1rate = new IloOplOutputFile(filename);
var Well2rate = new IloOplOutputFile(filename);
var Well3rate = new IloOplOutputFile(filename);
var Well4rate = new IloOplOutputFile(filename);
var Well5rate = new IloOplOutputFile(filename);
var Well6rate = new IloOplOutputFile(filename);

var ofileAlpha1 = new IloOplOutputFile(filename);
var ofileAlpha2 = new IloOplOutputFile(filename);
var ofileAlpha3 = new IloOplOutputFile(filename);
var ofileAlpha4 = new IloOplOutputFile(filename);
var ofileAlpha5 = new IloOplOutputFile(filename);
var ofileAlpha6 = new IloOplOutputFile(filename);

var vekt2 = 0;

  while( iter != Nu )
  {
   if(iter < 2)
{
 vekt2 = 1;
  }
  else
  {
    vekt2 = 0;
  }
    best = curr;
```

```
// Output to script log
   writeln("Solve for iteration ", iter);
   writeln("Shut-in time left: ", CurrSol.l);
if ( cplex.solve() ) {
   curr = cplex.getObjValue();
   writeln();
   writeln("OBJECTIVE: ",curr);
   writeln();
 writeln("alpha = ", CurrSol.alpha);
  }

  else
  {
   writeln("No solution!");
   break;
  }

// Define feedback variables
var q1 = CurrSol.q[1][feedback-vekt2].solutionValue;
var q2 = CurrSol.q[2][feedback-vekt2].solutionValue;
var q3 = CurrSol.q[3][feedback-vekt2].solutionValue;
var q4 = CurrSol.q[4][feedback-vekt2].solutionValue;
var q5 = CurrSol.q[5][feedback-vekt2].solutionValue;
var q6 = CurrSol.q[6][feedback-vekt2].solutionValue;

var alpha1 = CurrSol.alpha[1][feedback-vekt2].solutionValue;
var alpha2 = CurrSol.alpha[2][feedback-vekt2].solutionValue;
var alpha3 = CurrSol.alpha[3][feedback-vekt2].solutionValue;
var alpha4 = CurrSol.alpha[4][feedback-vekt2].solutionValue;
var alpha5 = CurrSol.alpha[5][feedback-vekt2].solutionValue;
var alpha6 = CurrSol.alpha[6][feedback-vekt2].solutionValue;

var eta1 = CurrSol.eta[1][feedback-vekt2].solutionValue;
var eta2 = CurrSol.eta[2][feedback-vekt2].solutionValue;
var eta3 = CurrSol.eta[3][feedback-vekt2].solutionValue;
var eta4 = CurrSol.eta[4][feedback-vekt2].solutionValue;
var eta5 = CurrSol.eta[5][feedback-vekt2].solutionValue;
var eta6 = CurrSol.eta[6][feedback-vekt2].solutionValue;

for(var stateiter in States)
{
m1[stateiter] = CurrSol.m[1][stateiter][feedback-vekt2].solutionValue;
m2[stateiter] = CurrSol.m[2][stateiter][feedback-vekt2].solutionValue;
m3[stateiter] = CurrSol.m[3][stateiter][feedback-vekt2].solutionValue;
m4[stateiter] = CurrSol.m[4][stateiter][feedback-vekt2].solutionValue;
m5[stateiter] = CurrSol.m[5][stateiter][feedback-vekt2].solutionValue;
m6[stateiter] = CurrSol.m[6][stateiter][feedback-vekt2].solutionValue;
}

// Output rate and alpha for each well
Well1rate.writeln(q1);
Well2rate.writeln(q2);
Well3rate.writeln(q3);
Well4rate.writeln(q4);
Well5rate.writeln(q5);
Well6rate.writeln(q6);

ofileAlpha1.writeln(alpha1);
ofileAlpha2.writeln(alpha2);
ofileAlpha3.writeln(alpha3);
ofileAlpha4.writeln(alpha4);
ofileAlpha5.writeln(alpha5);
ofileAlpha6.writeln(alpha6);


// Output total flow rate
tot_q = q1 + q2 + q3 + q4 + q5 + q6;
ofile2.writeln(tot_q);

// Output total alpha
tot_alpha = alpha1 + alpha2 + alpha3 + alpha4 + alpha5 + alpha6;
alphafile.writeln(tot_alpha);

// Prepare next iteration
var def = CurrSol.modelDefinition;
var data = CurrSol.dataElements;

if ( CurrSol != thisOplModel )
{
 CurrSol.end();
}

CurrSol = new IloOplModel(def,cplex);
```

```
// Remaining shutin time
if( shutinTime >= 1 )
{
 shutinTime = shutinTime - 1;
 data.l = shutinTime;
}
else
{
 shutinTime = constSI;
 data.l = shutinTime;
}

// Update data source with new q from feedback
data.q_init[1] = q1;
data.q_init[2] = q2;
data.q_init[3] = q3;
data.q_init[4] = q4;
data.q_init[5] = q5;
data.q_init[6] = q6;

 // Iterate predefined shutin periode
 if(alpha1 == 0)
  SI1 = constSI+iter;
 else
  SI1 = 0;
 if(iter < SI1)
  data.AlphaMin1[1] = 0;


 if(alpha2 == 0)
  SI2 = constSI+iter;
 else
  SI2 = 0;
 if(iter < SI2)
  data.AlphaMin1[2] = 0;


 if(alpha3 == 0)
  SI3 = constSI+iter;
 else
  SI3 = 0;
 if(iter < SI3)
  data.AlphaMin1[3] = 0;
 if(alpha4 == 0)
  SI4 = constSI+iter;
 else
  SI4 = 0;
 if(iter < SI4)
  data.AlphaMin1[4] = 0;


 if(alpha5 == 0)
  SI5 = constSI+iter;
 else
  SI5 = 0;
 if(iter < SI5)
  data.AlphaMin1[5] = 0;


 if(alpha6 == 0)
  SI6 = constSI+iter;
 else
  SI6 = 0;

 if(iter < SI6)
  data.AlphaMin1[6] = 0;

 // Update disturbance time
 if(data.disturbanceStart == 1)
  data.disturbanceStart = 1;
 else
  data.disturbanceStart = data.disturbanceStart-1;

 if(data.disturbanceEnd == 1)
  data.disturbanceEnd = 1;
 else
  data.disturbanceEnd = data.disturbanceEnd-1;

 if(data.disturbanceStart2 == 1)
  data.disturbanceStart2 = 1;
 else
data.disturbanceStart2 = data.disturbanceStart2-1;

 if(data.disturbanceEnd2 == 1)
```

```
  data.disturbanceEnd2 = 1;
 else
  data.disturbanceEnd2 = data.disturbanceEnd2-1;

 // Update data source with new pseudopressures
 for(var i in States)
 {
  data.m_init_tilde[1][i] = m1[i];
  data.m_init_tilde[2][i] = m2[i];
  data.m_init_tilde[3][i] = m3[i];
  data.m_init_tilde[4][i] = m4[i];
  data.m_init_tilde[5][i] = m5[i];
data.m_init_tilde[6][i] = m6[i];
 }

 // Add updated data source to the problem
 CurrSol.addDataSource(data);
 CurrSol.generate();


 iter++;
}


// Close output files
Well1rate.close();
Well2rate.close();
Well3rate.close();
Well4rate.close();
Well5rate.close();
Well6rate.close();

ofileAlpha1.close();
ofileAlpha2.close();
ofileAlpha3.close();
ofileAlpha4.close();
ofileAlpha5.close();
ofileAlpha6.close();

  ofile2.close();
  alphafile.close();
}
```