NTNU
Innovation and Creativity

Optimal Reservoir control using nonlinear MPC and ECLIPSE

Patrick Meum

Master of Science in Engineering Cybernetics
Submission date: June 2007
Supervisor: Ole Morten Aamo, ITK
Co-supervisor: John-Morten Godhavn, Statoil Forskningsenter, Rotvoll

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The production of oil and gas from a reservoir with a number of production wells and in some cases injection wells for pressure support is typically based on simulations and optimisation with a reservoir model (e.g. ECLIPSE). Long and short term plans are made and production engineers follow up these plans by daily or weekly changing well choke positions, considering situations, such as topside processing capacity constraints, equipment failures, critical reservoir pressure, max sand free rate, etc. The idea of this exercise is to see if this model based optimization can be solved by model predictive control (MPC). Linear model representations were investigated in an MSc project, and it was concluded that the responses were highly nonlinear. A potential for nonlinear MPC was found. The idea is to use ECLIPSE for predictions.

Tasks:
1. Perform a literature review on MPC with a particular focus on nonlinear MPC.
2. Implement a nonlinear MPC in the in-house Statoil MPC tool, SEPTIC
3. Develop an interface between SEPTIC and ECLIPSE for simulations and model predictions
4. Tune and test the NMPC for two different reservoir models, assuming no model errors.
5. Discuss the NMPC algorithm performance and the potential of optimal reservoir control.
6. Write a paper/abstract to a selected conference.

Assignment given: 08. January 2007
Supervisor: Ole Morten Aamo, ITK

Summary

Recent years advances within well deployment and instrumentation technology offers huge potentials for increased oil recovery from reservoir production. Wells can now be equipped with controllable valves at reservoir depth, which may possibly alter the production profitability of the field completely, if the devices are used in an intelligent manner. This thesis investigates this potential by using model predictive control to maximize reservoir production performance and total oil production.

The report describes an algorithm for nonlinear model predictive control, using a single shooting, multistep, quasi-Newton method, and implements it on an existing industrial MPC platform - Statoil's in-house MPC tool SEPTIC. The method is an iterative method, solving a series of quadratic problems analogous to sequential quadratic programming, to find the optimal control settings. An interface between SEPTIC and a commercial reservoir simulator, ECLIPSE, is developed for process modelling and predictions. ECLIPSE provides highly realistic and detailed reservoir behaviour and is used by SEPTIC to obtain numerical gradients for optimization.

The method is applied to two reservoir examples, Case 1 and Case 2, and develops optimal control strategies for each of these. The two examples have conceptually different model structures. Case 1 is a simple introduction model. Case 2 is a benchmark model previously used in Yeten, Durlofsky and Aziz (2002) and models a North Sea type channelized reservoir. It is described by a set of different realizations, to capture a notion of model uncertainty. The report addresses each of the available realizations and shows how the value of an optimal production strategy can vary for equally probable realizations.

Improvements in reservoir production performance using the model predictive control method are shown for all cases, compared to basic controlled references cases. For the benchmark example improvements range up to as much as 68% increase in one realization, and 30% on average for all realizations. This is an increase from the results previously published for the benchmark, with a 3% average. However, the level of improvement shows significant variation, and is only marginal for example Case 1. A thorough field analysis should therefore be performed before deciding to take the extra cost of well equipment and optimal control.

Preface

This report is written as a Master's thesis, and concludes my research work done in the final project for my Master's Degree at the Department of Engineering Cybernetics, Faculty of Information Technology, Mathematics and Electrical Engineering at the Norwegian University of Science and Technology (NTNU), spring semester 2007. The work has been carried out under the supervision of Professor Ole Morten Aamo and co-supervised by Staff Engineer Dr. Ing. John-Morten Godhavn and Senior Engineer Dr. Ing. Petter Tøndel at Statoil Research Center.

Statoil Research Center has been involved in the project formulation, and provided data and models used in the report, along with a work space at their office location with complete access to Statoil's software utilites.

This work could not have been completed without the help and support from a number of people. First of all I would like to thank Professor Ole Morten Aamo for giving me this assignment and for valuable help and support. Thanks to my co-supervisors Dr. Ing. John-Morten Godhavn and Dr. Ing Petter Tøndel for their daily help and inspiration.

I would like to thank Department Manager Dr. Ing. Gunleiv Skofteland for granting me the necessary access to Statoil locations and everyone else at the Process Control unit. Special thanks are due to MPC Specialist Dr. Ing. Stig Strand and Senior Engineer Morten Fredriksen for their invaluable help to get the SEPTIC application up and running.

The main results in this report would not have been possible to accomplish had it not been for the models provided to me by Dr. Burak Yeten. My thanks go to him for taking the time to respond to my enquiries.

Finally I would like to thank my co-students John Petter Jensen for all help and exchange of views during this semester and to Håvard Torpe for our productive collaboration in developing the SSMQN method and the writing of Chapters 2 and 3 in this report.

Patrick Meum
Trondheim, June 26, 2007

Contents

Introduction

1

"(...), nothing arises in the universe in which one cannot see the sense of some maximum or minimum."
- Leonhard Euler

In a time experiencing increasing energy demands, and estimated depletion of the already mapped out oil reserves within the next 50 years, there is a obvious need for enhancing the recovery factor of existing as well as for future reservoirs. Reservoir management has traditionally been performed on the basis of long and short term plans made by production engineers in a manual, *ad hoc* fashion. Reservoir models were generally viewed as to large and computer resources to scarce, to apply full scale optimization of the production chain. Meanwhile, on the downstream end of the production line and in process industry in general, advanced control techniques have been gradually developing and implemented with prosperous results.

Recent technological advances have opened for new possibilities within reservoir production. Mapping techniques like 4D seismic offers better quality reservoir model estimates for predictions, smart wells are equipped with valves downhole and multiphase flow meters to enhance control and computer speed has reached an acceptable level. This has made the oil industry spawn research programs to address the issue of closed-loop reservoir management. The goal is ultimately to apply real-time, optimal control to reservoir production. Inspired by the rest of the process community, one solution may be to use a model based optimization scheme. Such methods offer robust control to the type of constrained problems reservoir control represents.

This report seeks to investigate two main questions related to this topic. The first question treats the use of model predictive control (MPC) for reservoir optimization. Other authors have previously addressed reservoir production using optimal control theory, but further investigation using the concept of MPC was found necessary. The second question concerns the use of a reservoir simulator, ECLIPSE, for modelling purposes. The author has previously investigated reservoir optimization using system identification theory to derive linear model representations from ECLIPSE in Meum (2006). It was concluded that the reservoir model responses where highly nonlinear, suggesting the use of a nonlinear MPC (NMPC) approach instead.

An NMPC algorithm is developed and implemented in Statoil's MPC platform, SEPTIC. The algorithm is customized to handle black-box models and an interface between SEPTIC and ECLIPSE is established. The application is then

tested with two reservoir example models. Under the assumption of a perfect model simulations are carried out using the same models both for predictions and process simulations.

1.1 Structural contents

This report is structured in three main parts. The first part concerns the subject on MPC. A literature review on the topic is presented in Chapter 2, outlining a general picture of the MPC concept. Some basic connections to optimal control theory is described, along with a short historical survey. The chapter ends with a quick look on principles of NMPC and important controller properties. Chapter 3 develops an NMPC algorithm to be implemented in SEPTIC. The chapter describes the algorithm down to a pseudo code detail level, before applying it to a nonlinear test case model to present the algorithm performance.

The second part focuses on the reservoir production problem and reservoir simulation. Chapter 4 gives a description of the problem overview along with a presentation of previous work on the subject. Some of the major contributions are mentioned, with a short explanation of their scope and results. A quick introduction to reservoir simulation is given in Chapter 5, along with a description of basic ECLIPSE properties.

The third part describes the details of the reservoir examples investigated in this report and the simulation results, in Chapters 6 and 7. Finally conclusions and reflections on further work given in Chapter 9.

Model Predictive Control: A Short Survey

2

MPC is one of today's most commonly used techniques within advanced control. Although MPC is the widely used term and by now the conventional name of this technique, one can also find the term receding horizon control (RHC) used on occasions in older literature. For simplicity this report will mostly refer to the former term, though RHC will be used when found appropriate. Mayne, Rawlings, Rao and Scokaert (2000) describes MPC as:

> a form of control in which the current control action is obtained by solving *on-line*, at *each* sampling instant, a finite horizon open-loop optimal control problem.

By repeatedly solving the open-loop control sequence at each time step, the controller has an inherently closed-loop effect. As the new control sequence is calculated from present state of the system[1], a stabilizing feedback control can be obtained. This is where this technique has its main difference from other pre-computed, optimal control laws. The control law is calculated for a given horizon, T_c, and the dynamic behaviour of the system is predicted for a horizon T_p, where $T_c \leq T_p$. As the controller moves forward in time, so does the horizon (hence the mentioned term RHC). The basic idea of this is illustrated in Fig. 2.1. A system is sought to be controlled at a set point, given by $r(t)$. The controller calculates an optimal input sequence, parameterized as a piecewise constant function of time, for the control horizon. As time progresses all horizons are moved ahead as well, so they slide along by one sampling interval at each step.

The MPC is now described in a more formal, mathematical formulation, following notation from Allgöwer, Findeisen and Nagy (2004). Consider a general class of continuous time systems described by some differential equation

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0 \tag{2.1}$$

which is subject to input and state constraints of the form:

$$u(t) \in U, \, \forall \, t \geq t_0, \tag{2.2}$$

$$x(t) \in X, \, \forall \, t \geq t_0. \tag{2.3}$$

[1] In case of the system state not being fully measured, but estimated in an observer, the best estimate available is used as the basis for the calculation

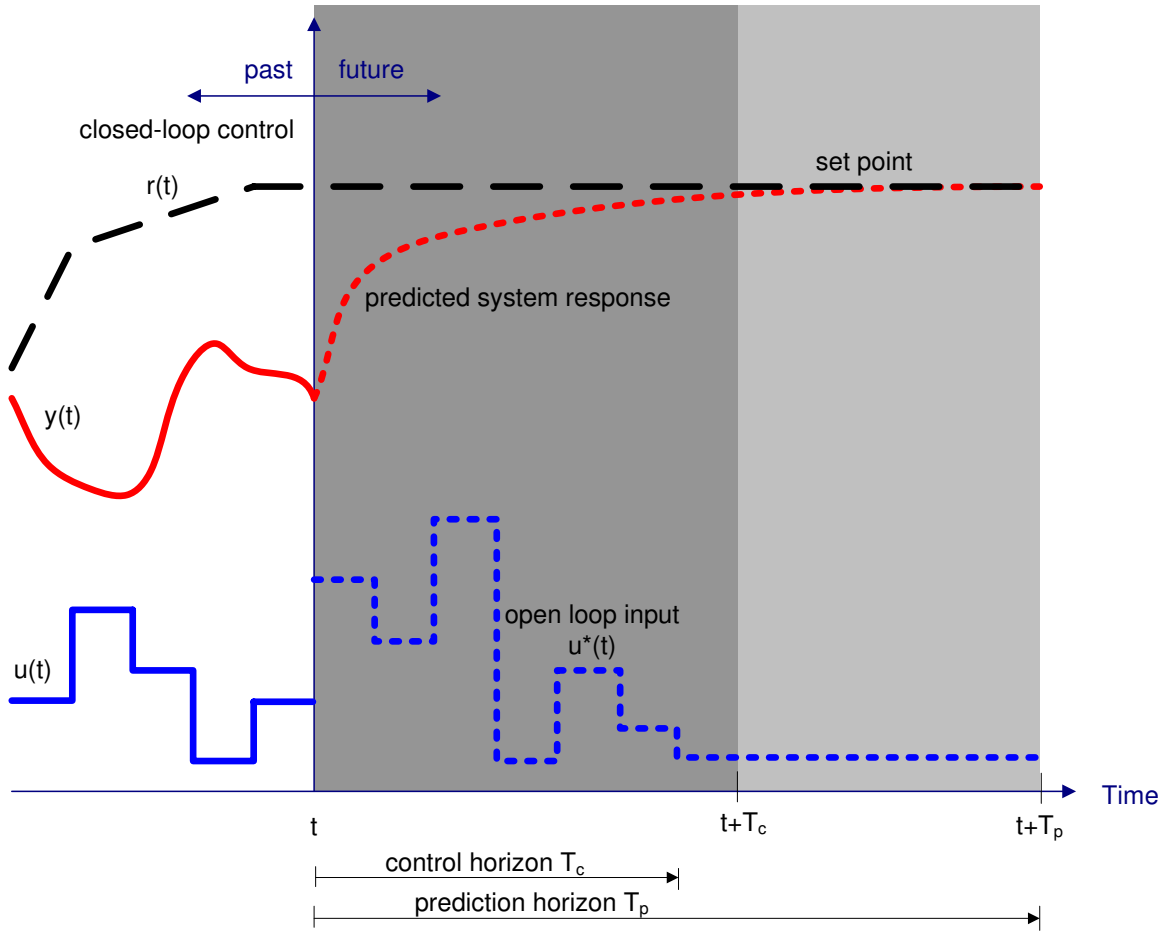


Figure 2.1: Principle of MPC

The inputs are given in the vector $u(t) \in \mathbf{R}^m$ and $x(t) \in \mathbf{R}^n$ denote the state vector, and the sets U and X are assumed to satisfy necessary topological properties. The optimal open-loop control is given by solving problem at every time instant:

$$\min_{\bar{u}(\cdot)} J(x(t), \bar{u}(\cdot)) = \int_t^{t+T_p} F(\bar{x}(\tau), \bar{u}(\tau)) \mathrm{d}\tau \tag{2.4a}$$

subject to

$$\dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)), \ \bar{x}(t) = x(t) \tag{2.4b}$$
$$\bar{u}(\tau) \in U, \ \forall \ \tau \in [t, t + T_c] \tag{2.4c}$$
$$\bar{u}(\tau) = \bar{u}(t + T_c), \ \forall \ \tau \in [t + T_c, t + T_p] \tag{2.4d}$$
$$\bar{x}(\tau) \in X, \ \forall \ \tau \in [t, t + T_p], \tag{2.4e}$$

where T_p and T_c refers to the control horizons already introduced above. The $\bar{u}$ denotes internal controller variables, and $\bar{x}$ refers to the system response to the

input vector $\bar{u}$, i.e. the solutions to (2.4b). The cost functional J is a sum of performance costs, $F(\cdot)$, at each time step. The cost function can in principle take any shape or form, but often arises from some economical consideration on the systems operational point (x_{op}, u_{op}) through a quadratic form:

$$F(x,u) = (x - x_{op})Q(x - x_{op}) + (u - u_{op})R(u - u_{op}). \tag{2.5}$$

Thus, the cost is given as a result of deviations from the operational set point, specified by positive definite *weighting matrices* Q and R. Often one can also find additional terms in F which penalize movements of the inputs where that is appropriate.

We define the solution to (2.4) as $\bar{u}^*(t; x(t))$, where the first set of inputs are applied to the system:

$$u(t) = \bar{u}^*(t_0; x_0) = \bar{u}_0^*. \tag{2.6}$$

The optimal cost yielded by $\bar{u}^*$ is then a function of the state $x(t)$ alone. This optimal cost is often referred to as the value function

$$V(x) = J(x(t), \bar{u}^*(t; x(t)). \tag{2.7}$$

This chapter gives a short review of selected portions of the existing theory available today on the topic of MPC. It does by no means set out to include every aspect of the subject, as that would be far out of scope for this report. This review relies heavily on the reviews made by Mayne et al. (2000), Qin and Badgwell (2003), Morari and Lee (1999), Allgöwer et al. (2004) and the book by Maciejowski (2002). The reader should assume to find everything mentioned here in these excellent papers. Citations are made in the largest extent possible, while simultaneously trying to maintain some degree of readability. The interested reader should in any case seek to investigate these surveys for a more comprehensive picture. Also recommended is the book by Allgöwer and Zheng (2000). Now, we will first focus on the underlying fundamentals which make up MPC by outlining the historical development from linear theory. Then we expand our horizon and move the attention the work made on nonlinear MPC (NMPC), arising new issues on i.e. solution time, algorithm design and feasibility considerations to name a few. The chapter ends with some discussion on the topics of stability and robustness for MPC in general and NMPC in particular.

2.1 The History and Evolution of MPC

Predictive control is the one general class of advanced control methodologies to have a significant impact on the practice in industrial control engineering,

Maciejowski (2002) states. Much of this class is covered by varieties of MPC, though there other subclasses can be found which share common properties with MPC (e.g. general predictive control). We will here only consider MPC, without much loss of generality. MPC started out as an industrial success only, because the advantages of this control were first recognized by the industrial engineering community alone. It was within the industry that much of the early work was done on the topic, covering only the analysis necessary to perform satisfactory performance. Only in the past 20-25 years have MPC gotten the attention it deserves from academia, which has made important contributions to clarifying the properties of stability and robustness for predictive control.

In the following of this subsection we consider a discrete, linear time-invariant (LTI) model

$$x_{k+1} = Ax_k + Bu_k \tag{2.8a}$$
$$y_k = Cx_k. \tag{2.8b}$$

2.1.1 Origin of MPC

Though MPC in the strong sense evolved *within* communities of industrial engineering, it did in fact evolve *from* important academic work done in the 1960s on the topic of optimal control. Mayne et al. (2000) states that MPC links

> Hamilton-Jacobi-Bellman theory (Dynamic Programming), which provides sufficient conditions for optimality and a constructive procedure for determining an optimal *feedback* controller $u = \kappa(x)$, and the maximum principle, which provides necessary conditions of optimality and motivates computational algorithms for the determination of the optimal open-loop control $\bar{u}^*(\cdot; x)$ for a given initial state x. The link is
>
> $$\kappa(x) = \bar{u}_0^* \tag{2.9}$$

This represent the ideal case of optimal control. The feedback control is given by the solution to the open-loop control problem, as in (2.4), for every x. Kalman (1960a) made an important complementary observation to this when showing that optimal control does not imply stability in the general finite horizon case. He found that stability can be shown, with some assumptions on the system conditions, for an infinite horizon optimal controller, known as the Linear Quadratic Regulator (LQR) (Kalman, 1960a,b). The LQR generates the optimal control sequence from a static state feedback law where the feedback gain is found via the solution of an Algebraic Riccati Equation (ARE), because the Hamilton-Jacobi-Bellman equations simplifies to an ordinary differential

equation for a LTI system, the *Riccati* equation. Stability can be guaranteed by ensuring that the problem objective function is positive definite through the choices of weighting matrices (Morari and Lee, 1999).

However, the stability properties alone were not enough to make the process industry embrace the LQR Qin and Badgwell (2003) reports, listing the following subjects that the theory failed to address:

- *constraints* - typically the most economically profitable operating point of a process lies in an intersection of several input or output constraints;
- *process nonlinearities* - LQR are based on LTI models;
- *model uncertainties* - LQR stability guarantees are made on assumptions of a perfect model;
- *performance criterion flexibility* - many process units may require to combine several objectives of different nature.

To include constraints in the infinite horizon controller, ones again needs to solve (2.9) as the solution based on ARE was no longer valid. However, computing the solution to the Hamilton-Jacobi-Bellman equations in the infinite case is a difficult task. From an on-line perspective this is usually not even close to practical to obtain. In fact, it often can not be found analytically at all, even in the simplest form, the unconstrained case (Allgöwer et al., 2004).

The solution to the infinite horizon problem was to redefine it as a receding horizon optimal control problem. Mayne et al. (2000) mentions the work of Kleinman (1970); Thomas (1975); Kwon and Pearsons (1977) and Kwon, Bruckstein and Kaliath (1983) as important here, as they all proposed different extensions to a stabilizing receding horizon alternative. With some variations the all shared a common approach, by introducing various types of terminal constraints. Although, as we will see later on in Section 2.2.3, the concept of adding a terminal constraint to the problem would show to have large impact on later research, the stability results of Kleinman, Thomas and Kwon et al. was limited to hold for unconstrained linear systems only. Hence, they also lacked many of the same properties as the LQR.

2.1.2 Industrial applications

The above listed weaknesses of the LQR were indeed reasons for the lack of its industrial support, but in addition, it has been claimed that the main reason was a cultural difference represented by LQR and the process industry. Control engineers either had no exposure to LQR concepts or regarded them as impractical (Qin and Badgwell, 2003). So the industry developed its own

methodology, by including properties such as input/output constraints and explicit process models from which could be estimated from test data.

In the following the first industrial MPC applications are presented, as presented in Qin and Badgwell (2003). These represent pioneering work which has influenced later MPCs in one way or the other.

IDCOM The first publication which explained MPC in the broader sense is often credited to Lee and Markus (1967). However, Richalet, Rault, Testud and Papon (1976) are the ones credited for describing the first MPC control application, with their IDCOM (Identification and Command). IDCOM was described by the authors as a model predictive *heuristic* control, because a transfer function of the control law was not available. This is due to the fact that MPC is not a linear controller, since it behaves nonlinear in terms of dealing with constraints. Even so, the IDCOM is what today is known as a linear MPC, because of its linear model representation. This was an impulse response model, known as a *finite impulse response* (FIR) model. The model had inputs called manipulated variables (MVs), if adjustable by the controller, and disturbance variables (DVs), if not available for control. The outputs were termed controlled variables (CVs). The FIR was identified from plant test data using a parameter estimation algorithm to minimize plant and model outputs. To calculate the control problem IDCOM used the same algorithm, by noting that control is the mathematical dual of identification (Qin and Badgwell, 2003). Most importantly, the IDCOM included what the LQR lacked, an explicit formulation of input and output constraints. These were included in the control calculations by checking for feasibility in all algorithm iterations.

The contributions by Richalet et al. are important because they proposed an application that satisfied particular demands of process control. Also, they pointed out the importance of embedding dynamic control in a control hierarchy to be effective. The significant economical benefits lies not in the low level dynamic control by reducing output variations, but in the above level by dynamically allocation variable set point as close to the operational constraints as possible. This has since been one of the crucial arguments in favour of MPC to be chosen as the control application.

DMC Dynamic Matrix Control (DMC) was presented by Cutler and Ramaker (1979, 1980) and was the other algorithm to form the first generation of MPC, together with IDCOM. DMC computed the optimal inputs as a solution to a least-squares problem, and did not include constraint handling in the original version. This was taken care of by Prett and Gillette (1980) which contributed with a modification to the DMC in which constraint handling was included for absolute input constraints.

The DMC used a linear step response model where the output is described by a weighted sum of past input changes, i.e. an integral of the impulse response. For multiple outputs DMC imposed the superposition principle. By utilizing this step response model the predicted future output changes can be written as a linear combination of future input moves, given an initial state. This way one can parameterize the computation of the optimal inputs as relative movement from the initial input, a fact which is still appreciated in MPC today. Considering the system from (2.8), and the fact that every future output can be written as a sequence of changes from the current value

$$\begin{aligned}
u_k =& \Delta u_k + u_{k-1} \\
u_{k+1} =& \Delta u_{k+1} + u_k \\
=& \Delta u_{k+1} + \Delta u_k + u_{k-1} \\
u_{k+2} =& \Delta u_{k+2} + u_{k+1} \\
=& \Delta u_{k+2} + \Delta u_{k+1} + \Delta u_k + u_{k-1} \\
& \vdots \\
u_{k+H_u-1} =& \Delta u_{k+H_u-1} + u_{k+H_u-2} \\
=& \Delta u_{k+H_u-1} + \Delta u_{k+H_u-2} \\
& \cdots + \Delta u_{k+1} + \Delta u_k + u_{k-1},
\end{aligned}$$

the future outputs can be written over the prediction horizon H_p as

$$\begin{aligned}
x_{k+1} &= Ax_k + B(u_{k-1} + \Delta u_k) \\
x_{k+2} &= Ax_{k+1} + B(u_k + \Delta u_{k+1}) \\
&= A^2 x_k + (AB + B)u_{k-1} + (AB + B)\Delta u_k + B\Delta u_{k+1} \\
&\vdots \\
x_{k+H_p} &= A^{H_p} x_k + (A^{H_p-1}B + \ldots + AB + B)\Delta u_k \\
&\quad \ldots + (A^{H_p-1}B + \ldots + AB + B)\Delta u_{k+1} \\
&\quad \ldots + (A^{H_p-H_u}B + \ldots + AB + B)\Delta u_{k+H_u-1} \\
&\quad + (A^{H_p-1}B + \ldots + AB + B)u_{k+1}.
\end{aligned}$$

Now we can stack the states and input changes over the prediction horizon in two vectors $\mathcal{X}(k)$ and $\Delta\mathcal{U}(k)$ respectively, and write it more compactly as

$$\mathcal{X}(k) = \psi x_k + \upsilon u_{k-1} + \theta\Delta\mathcal{U}(k),$$

where

$$\psi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_p} \end{bmatrix}, \; \upsilon = \begin{bmatrix} B \\ AB \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i B \end{bmatrix},$$

$$\theta = \begin{bmatrix} B & \cdots & 0 \\ AB + B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_p-1} A^i B & \cdots & \sum_{i=0}^{H_p-H_u} A^i B \end{bmatrix}.$$

The linear combination ties future inputs and outputs together through a so-called *Dynamic Matrix*. The dynamic matrix is a sensitivity matrix, $\mathcal{S}$, as will used later on in this report, which expresses the influence from every input to every output in the discrete parameterization over the prediction horizon. This matrix founds the basis for the least-squares problem definition, as the solution is given by (Maciejowski, 2002)

$$\Delta\mathcal{U} = \mathcal{S} \backslash \left[\mathcal{X}_{ref} - \mathcal{X}\right],$$

where $\mathcal{X}_{ref}$ depicts the control target.

Morari and Lee (1999) states that DMC had a tremendous impact on industry, estimating that probably every major oil company in world has a DMC inspired application installed in most new installations or revamps. But even if DMC, as well as IDCOM, did get the recognition LQR failed to receive in process industry, there still was weaknesses in their handling of constraints.

QDMC The breakthrough in constraint handling came when García and Morshedi (1986) showed how the DMC objective could be written as a standard quadratic program (QP), by introducing a quadratic cost function similar to (2.5). The QP provides efficient constraint handling in the control algorithm through the performance objective, both for inputs and outputs. As in standard DMC the dynamic matrix plays a important role, as the process constraints can be related directly to the input moves, re-written from u_k to Δu_k.

The beauty behind QDMC lays in the fact that a QP is a simple optimization problem to solve in the convex case. Since the Hessian of the QP is positive definite for linear plants, QDMC could easily calculate optimal control inputs using standard commercial optimization codes. This allowed the QDMC scheme to grow in model size and complexity, since not much time was spent on calculations.

Though several other industrial MPC applications have been proposed after the QDMC, this report will end this part of the review here. The interested reader is referred to Qin and Badgwells excellent survey on the topic to learn more about the third and the fourth generation of MPC. The next subsection will take a look at MPC from a nonlinear perspective, as this is where the interest currently lays. The QDMC seems like the natural place to stop, as it has proven to be the basis of several attempts to extend the MPC technology to incorporate nonlinear process models.

2.2 NMPC

This section addresses various elements on the topic of NMPC, from a control theoretical point of view. First possible model representations are presented, then different system discretizations and solution methods. Stability and robustness will also be given some attention, before ending with some notes on feasibility.

2.2.1 Model

The model is the heart of MPC. Qin and Badgwell (2003) states that, in principle, the model can take any required mathematical form. Unsurprisingly, a wealth of model formulations is used, although some are more common than others. In Qin and Badgwell (2003) some of the model forms used in commercial products are listed; finite impulse response (FIR), velocity FIR, Laplace transfer function, linear state-space, auto-regressive with exogenous input, Box-Jenkins and multi-model, input-output, first-principle, nonlinear state-space, nonlinear neural net and static nonlinear polynomial. These labels are not mutually exclusive, nor do they represent an exhaustive list of models that can be used. This section will look at some different categories in which most models can be sorted. The classification is based on Meadows and Rawlings (1997) and Rawlings (2000).

Perhaps the most important classification is the divide between *linear* and *nonlinear* models. For linear models, the superposition principle holds. That is, any linear combination of solutions for the linear system is in itself also a solution. Powerful tools for the analysis and control of such systems are available. Nonlinear models will in general have no special characteristics, and are perhaps only characterized by not being linear. An introduction to nonlinear models and nonlinear model identification, is presented in Pearson (1997).

Similarly, models can be classified as either *first principles* or *experimental*. First principles models are based on physical knowledge of the system and

are also referred to as physical models. In general a first principles model will be given as differential equations, either as ordinary differential equations (ODE) or as differential algebraic equations (DAE). Experimental models, also called black-box models, are used to obtain models that fit the process data sets. An experimental model will only contain the characteristics that were exhibited by the system during the identification process. Hence, the data used for model identification should be carefully selected, as the predictive ability of an experimental model is small outside the range of data that was used to identify it. Common experimental model formulations include step response and auto regressive external input models. The use of neural networks as models in MPC is an active area of research. The interested reader can consult Su and McAvoy (1997) with references. As a first principles model is based on physical insight into the system, it can be expected to describe system dynamics more completely than empirical models (Pearson, 1997). It should be mentioned that there exist model representations that tries to combine both experimental and first principles, so-called *hybrid* models. These can be thought of as experimental models based on fundamental physical laws of the underlying process, which is tuned with respect to model parameters to be in accordance to plant data.

Other classifications are possible. *Continuous time* or *discrete time*, *distributed parameters* or *lumped parameters*, *deterministic* or *stochastic*, *input-output* or *state-space* and *frequency domain* or *time domain*. Although these classifications can be used to further categorize models, they are less important than the first two already discussed.

2.2.2 Solution methods

There are three main strategies for solving the NMPC optimization problem (Allgöwer et al., 2004; Strand, 1991; Tenny, Wright and Rawlings, 2004; Barclay, Gill and Rosen, 1997). These can be categorized as either *sequential* or *simultaneous* approaches (Biegler, 2000).

If the behaviour of a system is completely determined by its initial values it is called an initial value problem (IVP). In order to determine the states in a time interval, $[t_0, t_1]$ only one simulation is necessary (single shot). This approach is also described as single shooting, sequential approach or feasible path approach and are more thoroughly described in Oliveira and Biegler (1995); Silva and Oliveira (2002); Tenny et al. (2004). A fuller description of single shooting is given in Section 3.

Multiple shooting differs in that the time interval is divided into multiple intervals. In each of these subintervals an IVP is solved independently. These IVPs must be solved iteratively with updated initial values that converge to the

end value in the preceding interval. Multiple shooting is therefore also coined as a sequential approach. Diehl, Bock, Schlöder, Findeisen, Nagy and Allgöwer (2002) lists several advantages with multiple shooting:

- As a simultaneous strategy, it allows to exploit solution information in controls, states and derivatives in subsequent optimization problems by suitable embedding techniques.
- Efficient state-of-the-art DAE solvers are employed to calculate the function values and derivatives quickly and accurately.
- Since the integrations are decoupled on different multiple shooting intervals, the method is well suited for parallel computation.
- The approach allows a natural treatment of control and path constraints as well as boundary conditions.

The third option is to solve the differential equations and the optimization problem simultaneously, thus the term simultaneous approach. The differential equations are discretized and enter the optimization problem as equality constraints. The manner in which the differential equations are discretized is important. Silva and Oliveira (2002) lists weighted residuals, orthogonal collocation and finite differences schemes as possible techniques for discretization/parameterization. Of these, *collocation* is the technique most often referenced. Barclay et al. (1997) explains collocation as *"a form of multiple shooting in which an appropriate implicit Runge-Kutta (IRK) formula is used to solve the initial-value problem"*.

2.2.3 Stability

Before theory on stability and robustness is presented, it is appropriate to define what the terms refer to. Skogestad and Postlethwaite (2005) mentions two types of stability — nominal stability (NS) and robust stability (RS). If the system is stable for the nominal plant, it is said to be nominally stable. Similarly, if the system remains stable for all plants in the uncertainty set, it is said to be robustly stable. In this report, stability is taken to mean nominal stability, while robustness is taken to mean robust stability.

Stability of NMPC is becoming a mature area of research, if not as mature as the stability of linear MPC. A good starting point to NMPC stability literature is provided in Mayne et al. (2000) and somewhat simpler stated in Allgöwer et al. (2004), the two of which this discussion is based. The literature is focused on the stabilization of a steady state.

Due to constraints, even linear MPC will result in a nonlinear control law, requiring nonlinear tools for the study of stability. For almost all stability

analysis, Lyapunov theory using the value function (recalling (2.7)) is therefore used. Mayne et al. (2000) starts the stability analysis by listing several modifications to the MPC scheme such that stability can be guaranteed.

Since MPC solves a open loop problem over a horizon, the easiest method of stabilization to grasp is to extend the horizon infinitely. Due to Bellmanns principle of optimality, the optimal trajectory in the next sample will be the remaining trajectory of the previous sample. Since an infinite horizon input trajectory in general will be impossible to compute in finite time, several approximations exists that ensures closed loop stability with finite horizon. One can, without loss of generality, assume that the system is to be stabilized at the origin. *Terminal equality constraint* ensures stability by requiring that all states should be set to zero at the end of the prediction horizon and the control inputs used to maintain the system at the origin should also be zero,

$$\bar{x}(t + T_p) = 0. \tag{2.10}$$

The obvious disadvantage of a terminal constraint is that it will force the system to a selected state in finite time, heavily decreasing the solution space of the problem. Also, there is in addition the extra computational burden of finding an exact satisfaction to the imposed equality, as this transforms the problem to a boundary-value problem (BVP). *Terminal cost function* uses no exact terminal constraint. Instead, a terminal cost is used to ensure the stability. Such a terminal cost can unfortunately only be obtained generally for linear unconstrained or linear constrained stable systems. The terminal costs can in these cases be computed by solving the Lyapunov equation, see for example Maciejowski (2002). A method related to terminal equality constraints is the *terminal region constraint* where the states are required to lie within a terminal set at the end of the prediction horizon,

$$\bar{x}(t + T_p) \in \Omega. \tag{2.11}$$

This terminal region, Ω, are defined by calculating a controller that drives all states within the terminal set exponentially fast to the origin. Finally, it is possible to combine terminal set with terminal cost. A cost E is added to (2.4a) for the terminal state, giving

$$J(x(t), \bar{u}(\cdot)) = \int_t^{t+T_p} F(\bar{x}(\tau), \bar{u}(\tau))\mathrm{d}\tau + E(\bar{x}(t + T_p)). \tag{2.12}$$

If the terminal cost equals the residual of weights for all $t > T_p$, the solution of (2.4) extended with (2.11) and (2.12) will, in effect, become the infinite horizon solution. The closer the terminal cost is to the infinite horizon weight, the more of the benefits gained by infinite horizon are realized. A deduction of an infinite

horizon scheme is given in Chen and Allgöwer (1998a) and Chen and Allgöwer (1998b).

Allgöwer et al. (2004) gives the following theorem ensuring stability of the NMPC, after modifying the original problem (2.4) with (2.11) and (2.12):

Theorem 2.1 *Assume that:*

1. $U \subset \mathbf{R}^m$ *is compact,* $X \subseteq \mathbf{R}^n$ *is connected and the origin is contained in the interior of* $U \times X$*.*
2. *The vector field* f*:* $\mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^n$ *is continuous in* u *and locally Lipschitz in* x *and satisfies* $f(0,0) = 0$*.*
3. F*:* $\mathbf{R}^n \times U \to \mathbf{R}$ *is continuous in all arguments with* $F(0,0) = 0$ *and* $F(x,u) > 0 \forall (x,u) \in \mathbf{R}^n \times U \backslash 0,0$*.*
4. *The terminal penalty* $E : \Omega \to \mathbf{R}$ *is continuous with* $E(0) = 0$ *and that the terminal region* Ω *is given by* $\Omega := x \in X | E(x) \leq e_1$ *for some* $e_1 > 0$ *such that* $\Omega \subset X$*.*
5. *There exists a continuous local control law* $u = k(x)$ *such that* $k(x) \in U$ *for all* $x \in \Omega$ *and* $\frac{\mathrm{d}E}{\mathrm{d}x} f(x, k(x)) + F(x, k(x)) \leq 0, \forall x \in \Omega$*.*
6. *The NMPC open-loop optimal control problem has a feasible solution for* $t = 0$*. Then for any sampling time* $0 < \delta \leq T_p$ *the nominal closed-loop system given by the problem* (2.4) *extended with* (2.11) *and* (2.12)*, and the input* (2.6)*, is asymptotically stable and the region of attraction* $\mathbf{R}$ *is given by the set of states for which the open-loop optimal control problem has a feasible solution.*

Mayne et al. (2000) concludes the discussion about stability with the conclusion that a combination of terminal set and terminal cost seems to be the best way to ensure both stability and good performance.

2.2.4 Robustness

When NMPC is applied in practice, it is unrealistic to assume that the model used for prediction will accurately match the process and similarly that no un-modelled disturbances enters the process. Obviously a successful implementation of NMPC relies on being able to cope with these issues. The MPC formulation has an inherent robustness, due to its similarity with optimal control (Allgöwer et al., 2004). According to Qin and Badgwell (2003) most commercial MPC products relies on this robustness and brute force evaluation of model mismatch to ensure robustness. Qin and Badgwell (2003) calls for MPC formulations that ensure robust stability. A survey of different schemes

to ensure robust stability can be found in Allgöwer et al. (2004) and Mayne et al. (2000) which is also the basis for this discussion.

Three approaches to achieve robust stability are outlined. The first of these is to *solve an open-loop min-max problem*, which minimizes the maximum objective function value for a set of uncertainties. The input sequence must be feasible for every realization in the set of uncertainties, possibly giving feasibility problems or conservative solutions. *H_∞ MPC* is another approach, using H_∞ control to achieve robustness. Large computational cost and the need for a global optimum are drawbacks for this approach. The third approach is to use the parameters of a *feedback controller* as optimization variables instead of using inputs directly. Because a feedback controller is used, disturbances are rejected also between sampling times, reducing the need for conservative control.

Mayne et al. (2000) concludes the discussion of robustness by stating that current robustness schemes must be regarded as conceptual rather than practical.

2.2.5 Feasibility

Model based predictive control is based on finding a solution to the optimization problem (2.4). Feasibility problems arise when no solution to this problem can be found. Normally, feasibility problems are solved by relaxing process constraints. Process constraints are often output constraints that define safety constraints, process equipment limitation or product specifications. Such constrains are by nature different from input constraints that are physical constrains imposed by the actuator. Such constrains can be maximum power, maximum opening etc. Thus, input constraints can normally not be violated.

Different ways of imposing soft constraints are possible. Oliveira and Biegler (1994) advocates for the use of exact penalty functions. Hence, process constraints are only violated if no other feasible solution exists. It is also possible to penalize constraints relaxation using a normal quadratic penalty. A quadratic penalty implies that, if the optimal operation is on the constraint, a small violation of the constraint will be allowed. This violation will be reduced as the penalty is increased. However, as long as the penalty is finite, the violation will also be finite.

When discussing feasibility of MPC, there is one additional type of constraints that needs to be considered, namely model constraints. Naturally, these constraints cannot be relaxed. For a linear system, the model constraints can be solved exactly and simultaneously with the optimization problem. In general, this will not be possible for nonlinear models. A discussion of how nonlinear equalities enters the optimization can be found in Section 2.2.2. A

short discussion of the different solution methods impact on the feasibility of the model follows.

Single shooting enjoys the advantage that the model equations are satisfied along the whole prediction horizon. Thus, each iterate will be a feasible albeit suboptimal solution of the optimization problem. Multiple shooting and collocation allows iterations, inconsistent with the system dynamics. (Tenny et al., 2004) Thus, the solution cannot be guaranteed feasible until convergence is achieved.

This concludes the chapter on MPC theory. From a focus on methodological overview and important properties such as stability and robustness, the following chapter will present a dedicated NMPC application, derived from a SQP principle.

A multistep quasi-Newton method

3

This chapter will focus on the NMPC method developed in this report, a single shooting, multistep, quasi-Newton method (SSMQN). The first section contains the motivation for using NMPC along with some background on the chosen method. A detailed description of the SSMQN method follows in the second section, while the implementation used for this report is described in the third section. Finally, an example used extensively in the development of the SSMQN — NMPC used on a CSTR system, is presented.

3.1 Background and motivation

When the model used for predictions in an MPC application is changed from a linear model to a nonlinear model, the optimization problem to be solved online changes from a QP (or LP if a linear objective function is used) to a NLP[2]. While QP's are solved fast and reliably with standard QP solvers, algorithms for solving NLP's are much more an area of research. In this report, the NLP will be solved using a single shooting multistep quasi-Newton method. The method was described by Li and Biegler (1989) extending the previous single-step method by Li, Biegler, Economou and Morari (1990). An algorithm very similar to the one used in this report can be found in Oliveira and Biegler (1995) where more general objective functions can be accommodated. A linearized input output model is obtained from a black box nonlinear model around the nominal input sequence found at the previous sample. A search direction is found by formulating a QP sub problem using the linearized model. In order to avoid algorithm divergence, a linesearch is performed along the calculated search direction, ensuring descent in the objective function. The solution found is checked against a convergence criterion, and, if not satisfactory, set as nominal input trajectory and the procedure repeated. Summarized, the algorithm is a SQP algorithm especially adapted to the NLP's generated by a NMPC.

Single shooting has some known stability and robustness issues (Barclay et al., 1997; Biegler, 2000). If the system is locally unstable, the states can diverge, giving immediate failure. It can also be difficult to find input trajectories with bounded outputs except very close to the optimal trajectory making single

[2]NLP is a commonly used acronym for a *nonlinear programming*

shooting unsuitable as an optimization strategy. Despite these problems, single shooting has been reported to successfully control some nonlinear systems, e.g. batch processes in Silva and Oliveira (2002).

3.2 Algorithm description

3.2.1 Outline of algorithm

The following outline of the algorithm is based on the algorithm described in Oliveira and Biegler (1995)

1. Set the QP sub problem iteration counter to zero.
2. Compute input sensitivities for the nominal input trajectory.
3. Solve the QP sub problem with the linearized model to find a search direction
4. Employ a line search algorithm to determine a suitable step size along the search direction.
5. If the solution found satisfies a defined convergence criteria, the first input of the computed input trajectory given to the controlled system, and the new input trajectory shifted one sample and set as the nominal input trajectory.
6. If the solution fails to satisfy the convergence criteria, the QP sub problem iteration counter is checked, and if less than the iteration limit, it is incremented and the algorithm jumps to Step 2. If the maximum iteration limit is reached, the algorithm is stopped and the best input trajectory found returned.

The following subsections will describe some of these steps in greater detail. It can be practical to return to this algorithm outline to keep the right perspective.

3.2.2 Input sensitivities

Step 2 of the algorithm is to linearize the model by computing sensitivities. Depending on the level of model control, there are several different possibilities as to which sensitivities should be calculated. Silva and Oliveira (2002) shows figures of two possible methods of calculating sensitivities. The first method is to calculate the sensitivities from inputs to state, state to state and state to output. The second is to only calculate input to state and state to output sensitivities. The drawback of this approach is that the input sensitivity for one

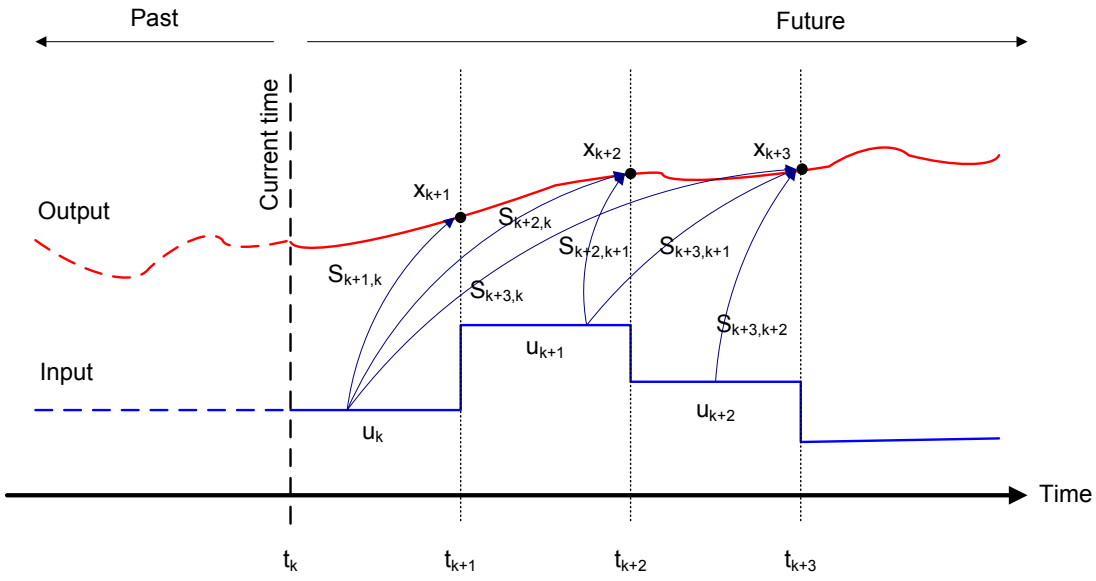


Figure 3.1: Calculation of sensitivities

input must be calculated for all subsequent states as opposed to calculating only the sensitivity to the states in the next sample, and then to use state to state sensitivities. However, it can be seen as an advantage that it is not necessary to perturbate the states, as this is problematic for black box models.

The scheme for calculating sensitivities used in this report, can be seen as a variant of the second approach. Sensitivities are calculated from input to outputs directly, analogous to defining the outputs as the state vector x and using a unity measurement sensitivity. Inspired from Silva and Oliveira (2002) the sensitivities are given in a matrix with elements given by

$$\mathcal{S}_{j+1,i} = \frac{\partial x_{j+1}}{\partial u_i}, \quad j \geq i, \tag{3.1}$$

$$\mathcal{S}_{j+1,i} = 0, \quad \text{else}, \tag{3.2}$$

$$j \in [k, T_p - 1],\ i \in [k, T_c]. \tag{3.3}$$

The sensitivities calculated are shown in Fig. 3.1. As can be seen, the sensitivity from each input parameter to each subsequent coincidence point is calculated. Notice that in Fig. 3.1 only perturbations in the inputs are necessary.

The sensitivity matrix calculated gives a linearized input-output model of the

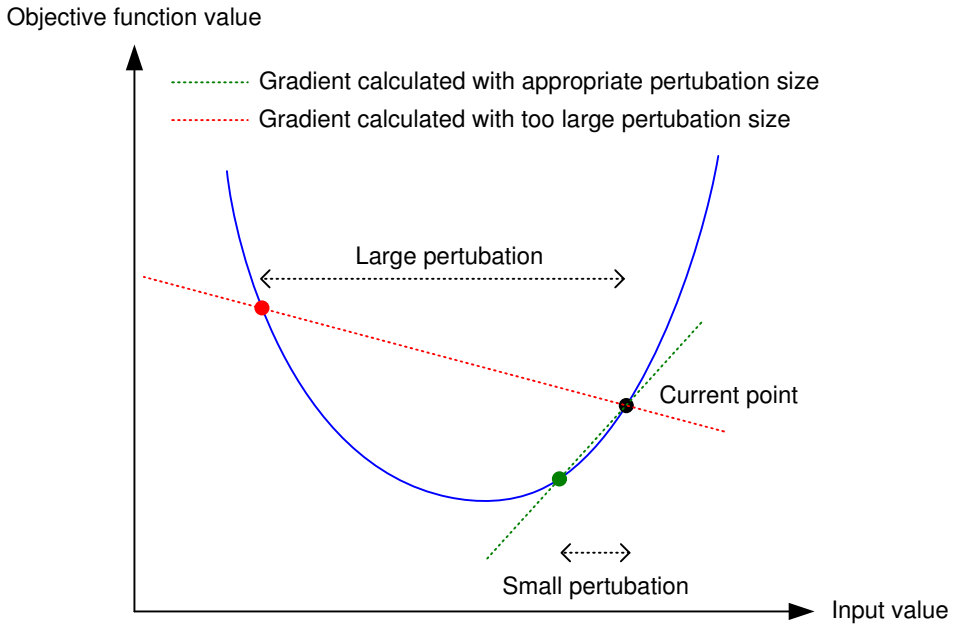


Figure 3.2: Effect of too large perturbation on numeric gradients

form:

$$\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+T_p} \end{bmatrix} = \begin{bmatrix} \mathcal{S}_{k+1,k} & 0 & \dots & 0 \\ \mathcal{S}_{k+2,k} & \mathcal{S}_{k+2,k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}_{k+T_p,k} & \mathcal{S}_{k+T_p,k+1} & \dots & \mathcal{S}_{k+T_p,k+T_c} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+T_c} \end{bmatrix} + \mathbf{b}, \quad (3.4)$$

where x_{k+i} is the output at time $k+i$, Δu_{k+i} is the change in input at time $k+i$ and $\mathbf{b}$ is a vector containing the constant terms of the linearization.

The sensitivity matrix is obtained by perturbating the inputs by small, but finite, values, giving numerical gradients. The size of the perturbations influences the accuracy of the sensitivity, and hence also the convergence rate of the SQP algorithm. Too small perturbations give sensitivities dominated by numerical noise, while too large perturbations will give inaccurate sensitivities and possibly problems with convergence. It is therefore desirable to select a perturbation size as small as possible without encountering numerical difficulties. The lower limit on perturbations is determined by the accuracy of the solver used to simulate the nonlinear model. An example showing the effect of too large perturbation is shown in Fig. 3.2. The large perturbation will wrongly indicate an increase in the objective function value moving from right to left, where as the small perturbation correctly shows a descent direction towards the minimum.

3.2.3 The QP sub problem

This section will present the QP sub problem generated in SQP algorithms. Solving this QP sub problem is done in *Step 3* in order to find a search direction. The introduction and the notation will follow Nocedal and Wright (1999).

Assuming a nonlinear problem of the form:

$$\min f(x) \tag{3.5a}$$
$$s.t.\ c_i(x) = 0, i \in E \tag{3.5b}$$
$$c_i(x) \geq 0, i \in I, \tag{3.5c}$$

where $f(x)$ is the function to be minimized, subject to state constraints $c_i(x), i \in E \cup I$. E is the set which contains the indices of equality constraints, as I is the set of indices of inequality constraints.

The QP subproblem (3.6) to be solved in order to find the search direction, p, are obtained by linearizing both equality and inequality constraints

$$\min \frac{1}{2} p^T W_k p + \nabla f(x_k)^T p \tag{3.6a}$$
$$s.t.\ \nabla c_i(x_k)^T p + c_i(x_k) = 0, i \in E \tag{3.6b}$$
$$\nabla c_i(x_k)^T p + c_i(x_k) \geq 0, i \in I \tag{3.6c}$$

Where $W_k = W(x_k, \lambda_k) = \nabla^2_{xx} L(x_k, \lambda_k)$ denotes the Hessian of the Lagrangian of (3.5), with the Lagrangian defined as $L(x, \lambda) = f(x) - \lambda^T c(x)$.

For general SQP algorithms it is important to maintain W_k positive definite to avoid generating non-descending search directions, p. This is necessary in order to guarantee convergence on non-convex problems and from remote starting points. There are several possibilities as to how this can be accomplished. Nocedal and Wright mentions using full quasi-Newton approximations, such as the BFGS formula, using the Hessian of an augmented Lagrangian functions or using reduces-Hessian approximations. However, none of these techniques are required for NMPC if using a special choice of Hessian, as proven in the following section.

In MPC, the objective function $f(x) = \frac{1}{2} x^T Q x$ are quadratic in x. Thus, the Hessian becomes $\nabla_{xx} f(x) = Q$ and will be positive definite as long as Q is chosen positive definite. The only possibility for W_k not to become positive definite is if $-\nabla^2_{xx} \lambda^T c(x)$ is negative definite. By linearizing the constraints $c(x)$ before the QP sub problem is formulated, the contribution to W_k from the Hessian of the constraints will always be zero. This is equivalent to the "Gauss-Newton" choice of Hessian where the effects of the nonlinear model equations on the Hessian is neglected. The effects of neglecting the contribution from the nonlinear model

is described (among other choices for Hessians) in Tenny et al. (2004). Here it is stated that this choice in not asymptotically equivalent to the true Hessian of the Lagrangian unless the model is linear or the Langrange multipliers for the model is zero. It is also pointed out that when x in (3.5) is close to its optimal value, the state equations will be only weakly active, and their Lagrange multipliers close to zero. The Gauss-Newton choice of Hessian will therefore be close to the true Hessian. In fact, according to Biegler (1998), a NMPC has a structure that, for a solution interior to the constraints, will have $W_k \approx \nabla^2 L$ when using the "Gauss-Newton" choice of Hessian. The approximated Hessian of the Lagrange function will be asymptotically equivalent to the actual Hessian at the solution, resulting in a Q-quadratic convergence rate.

3.2.4 Linesearch

The QP sub problems are based on a quadratic approximation of the original NLP. If the approximation is poor, the solutions found may in fact cause the algorithm to diverge, if implemented unbounded. In order to ensure that the step taken is a descending step, a linesearch algorithm is employed in *Step 4*. This algorithm must decide the fraction of the step to be taken. Newton methods have a natural step length of one. The fraction, α, of the step taken should therefore be in the interval $(0, 1]$. One can potentially solve an optimization problem, finding the step length, $\alpha \in (0, 1]$, minimizing the objective function. However, it will in general be a compromise between doing a thorough linesearch with fewer QP solutions, or a approximate linesearch with more QP solutions. Nocedal and Wright (1999, chap. 3) describes some approaches to linesearch based on the Wolfe conditions — the sufficient decrease condition and the curvature condition. The sufficient decrease condition ensures that the step is descending while the curvature condition makes certain that the step length is not unacceptably short steps.

3.2.5 Convergence criterion

The optimization termination criteria in *Step 5* can be specified as in generic SQP algorithms. However, since the single-shooting quasi-Newton methods search along a feasible path, optimization can be terminated before exact convergence is achieved, still yielding feasible solutions although not optimal. Especially since NMPC generates a series of NLP's, where the solution of the previous NLP can be used as a starting point for the next NLP, and where the solution often converges over time, the convergence criterion can be relaxed. Since the accuracy of the sensitivities are bounded by the perturbation size (as discussed earlier in section on sensitivities (Section 3.2.2)), and thus also the solver accuracy, it can be convenient to use a relaxed convergence criterion.

3.3 Implemented algorithm

The previous section described some issues in the single shooting multistep quasi-Newton algorithm used in this report. In this section, the solutions used are described, as well as a pseudo code of the implementation.

3.3.1 Description of SEPTIC

The NMPC application was implemented in STATOIL's in-house MPC tool, SEPTIC (Statoil Estimation and Prediction Tool for Identification and Control). SEPTIC was first used in 1997 Strand and Sagli (2003), and has since then been implemented on 70 installations. Most of these applications have been implemented using linear and experimental models, as these have proved sufficiently accurate for the respective processes. There is however possible to use first principles nonlinear models, either as models programmed in SEPTIC, or through a interface against an external simulator. Either way, SEPTIC treats the model as a black-box model. None of the previous applications have used nonlinear simulator models, and as such SEPTIC did not include a robust algorithm for an iterative NMPC scheme. Hence, there was a need to develop the algorithm presented here.

3.3.2 Computation of sensitivity

Because of the simple model interface in SEPTIC, each column in the sensitivity matrix in (3.4) is calculated by simulating the system from t_k to t_{k+T_p}, which might seem unnecessary as all columns except the one to the extreme left contains one or more zero elements in the top. Two facts, however, implies that this implementation is not as wasteful as it might seem at first sight. Firstly, input blocking are normally implemented with increasing block length. Consequently the number of zero elements in the sensitivity matrix will be much less than half the total number of elements. Secondly, if the sensitivity matrix is found from parallel processing where each input parameter sensitivity is simultaneously calculated on separate CPUs, all processes calculating sensitivities for inputs $u_i, i \in [k+1, k+T_c]$ must wait for the process calculating the sensitivity for u_k to end. Simulating shorter horizons for $u_i, i \in [k+1, k+T_c]$ will therefore have no effect on the algorithm run time.

3.3.3 QP solver

The QP solver used in SEPTIC is a C compilation of a Fortran implementation developed by Schittkowski (2005) This implementation is based on the dual

method developed by Goldfarb and Idnani (1983) The motivation for the development of this algorithm was the need for fast and robust solutions to the QP sub problems generated by a SQP algorithm. The dual QP solver implemented requires the QP to be positive definite. Earlier in this chapter, in section 3.2.3, it was shown that this will be true for the Gauss-Newton choice of Hessian approximation used in SEPTIC.

Compared to a primal algorithm, the advantage presented by a dual method is that there is no need for an expensive search for a feasible starting point (Goldfarb and Idnani, 1983). The unconstrained solution of the primal problem is a feasible solution for the dual problem. A feasible solution for the primal problem is obtained by solving series of sub problems where violated constraints are added until all constraints are satisfied. This procedure is in Goldfarb and Idnani (1983) considered superior to primal algorithms when no feasible starting point is immediately available.

In conclusion the QP sub problem solved in SEPTIC are stated in (3.7) where the Hessian of the objective function will be positive definite as long as the Q used is positive definite.

$$\min \frac{1}{2} p^T Q p + \nabla f(x_k)^T p \tag{3.7a}$$

$$s.t.\ \nabla c_i(x_k)^T p + c_i(x_k) = 0, i \in E \tag{3.7b}$$

$$\nabla c_i(x_k)^T p + c_i(x_k) \geq 0, i \in I \tag{3.7c}$$

3.3.4 Linesearch

Backtracking is the implemented method of linesearch. The algorithm is a modified version of the backtracking linesearch found in Nocedal and Wright (1999, procedure 3.1, pages 41-42). First the full step is evaluated, then smaller and smaller fractions are evaluated until an acceptable step length is found. The difference between subsequent steps is a constant factor, ξ. For a step ξ^i to be accepted, two criteria needs to be fulfilled. First, the objective value with step length ξ^i must be less than the objective value with step length ξ^{i+1}. Second, the objective value with step length ξ^i must be less than the objective value with step length zero. Fig. 3.3 shows a case designed to illustrate the linesearch convergence criteria. Here, the algorithm will choose a step length $\alpha = \xi^3$, since this is assumed to be a local minima with greater objective function values on both sides.

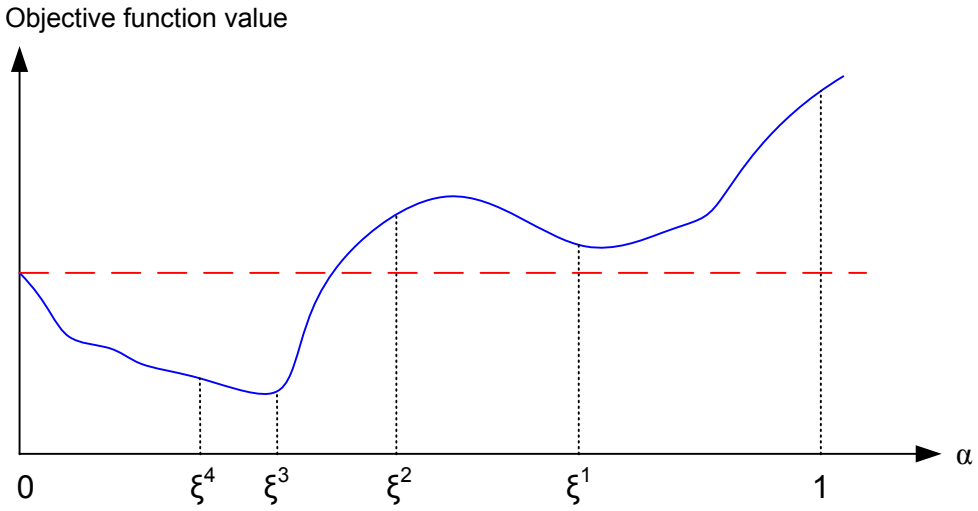


Figure 3.3: Backtracking linesearch

3.3.5 Convergence criterion

Different algorithm convergence criteria considered in the algorithm developed in this report were linearization error, change in objective function and change in input parameters. The original SEPTIC algorithm used the linearization error criterion — after a step, the difference in the response of the controlled variables using the nonlinear and the linearized model is computed. If the linearization error is below a limit, the linear model is assumed to be an accurate description of the nonlinear model along the solution. However, consistent performance was difficult to obtain with this criterion.

Instead, the norm of the input change was used. Li and Biegler (1989) proposed this in their algorithm, as $\sum_k \left\|\bar{u}_k - \bar{u}_{k-1}\right\|^2 \leq \epsilon$, but noting "*that it is not really necessary to solve the QP problem (...) repeatedly (i.e. ϵ can be large)*". This was found not to be true using models of such a degree of nonlinearity as in this report. Thus, a small value should be chosen for ϵ, so that a sufficient number of QP's are solved at each iteration.

$$\sum_k ||\Delta u_k||^2 < MV_{norm}. \tag{3.8}$$

The explicit criterion used is given in (3.8) where MV_{norm} is the convergence limit. This criterion gave consistent performance while the resulting algorithm was significantly simpler. Alternatively, the change in the objective function could have been used. The obvious drawback is that if a step gives a reduction in the objective function value below the limit, the step may still be of significant

length and the sensitivity around the new solution may be quite different from the previous. Using only the descent in objective function will then stop the optimization even if the new solution can result in a significant improvement in objective function value. Possibly, a criterion combining change in objective function with the norm of the input change would be better. However, the algorithm in its current form performs well.

3.3.6 Algorithm pseudo code

The SQP algorithm implemented in SEPTIC is given as pseudo code in Algorithm 1. The linesearch used is also given in pseudo code in Algorithm 2. Pseudo code is used in order to be able to state the algorithms more clearly, as the actual SEPTIC code makes extensive use of specialized methods and data structures which would have made the presentation of the algorithm unduly involved. Some previously presented and some new parameters will be used. MV_{norm} and ξ was described in Sections 3.3.5 and 3.3.4 respectively. In addition will $N_{QP,max}$ be used to represent the maximum number of QP iterations allowed in one NMPC step, $N_{ls,max}$ the maximum number of linesearch iterations and U_{valid} is the feasible input values. In addition will some of the variables used in Algorithm 1 also be used in Algorithm 2. The used variables are declared at the top of Algorithm 2.

3.4 Test system: a continuously stirred tank reactor

This section shows NMPC tested on a small continuously stirred tank reactor (CSTR) model. Even though the intension behind developing the SSMQN algorithm was to apply it to a computational expensive simulator model, developing it on such models hardly proved practical considering the time spent on retrieving results after algorithm modifications or tuning. To develop and verify the NMPC application there was a need for quick and simple model, which could simulate a number of runs in only a few seconds time, but still exhibit severe nonlinearities in order to challenge the algorithm robustness.

3.4.1 CSTR model description

The test model used is a CSTR with multiple steady states, previously investigated by both Oliveira and Biegler (1995) and Martinsen, Biegler and Foss (2004). The model equations are given in (3.9). The equations are similar to the system given in Martinsen et al. (2004), with the exception that the effect of inputs u_1 and u_2 on $\frac{\mathrm{d}x_1}{\mathrm{d}t}$ is quadratic. This modification was done in order to make the system more nonlinear, and does not reflect any physical

Algorithm 1 SEPTIC SQP algorithm

Require: $u_{nom} \in U_{valid}$, MV_{norm} and $N_{QP,max}$
  $Y_{best} \leftarrow$ calcObjectiveValue(u_{nom})
  $S \leftarrow$ calcSensitivity(u_{nom})
  $\Delta u \leftarrow$ solveQP, $n_{QP} \leftarrow 1$
  $Y_{new} \leftarrow$ calcObjectiveValue($u_{nom} + \Delta u$)
  $runSQP \leftarrow$ true
  while $runSQP$ **do**
    $\alpha_{best} \leftarrow$ doLineSearch
    $u_{nom} \leftarrow u_{nom} + \alpha_{best} * \Delta u$
    $Y_{best} \leftarrow$ calcObjectiveValue(u_{nom})
    if $||\alpha_{best} * \Delta u|| < MV_{norm}$ **then**
      $runSQP \leftarrow$ false
    else if $n_{QP} == N_{QP,max}$ **then**
      $runSQP \leftarrow$ false
      $warning \leftarrow$ maximum number of iterations without convergence
    else
      $S \leftarrow$ calcSensitivity(u_{nom})
      $\Delta u \leftarrow$ solveQP(u_{nom}), $n_{QP} \leftarrow n_{QP} + 1$
      $Y_{best} \leftarrow$ calcObjectiveValue(u_{nom})
    end if
  end while

interpretation.

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = u_1^2 + u_2^2 - k_1\sqrt{x_1} \tag{3.9a}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = (C_{B_1} - x_2)\frac{u_1}{x_1} + (C_{B_2} - x_2)\frac{u_2}{x_1} - \frac{k_2 x_2}{(1+x_2)^2} \tag{3.9b}$$

The initial parameters, following Martinsen et al. (2004), are set to: $k_1 = 0.2$, $k_2 = 1$, $C_{B_1} = 24.9$ and $C_{B_2} = 0.1$. With inputs $(u_1, u_2) = (1, 1)$ the system has three equilibrium points. These can be found at $x_1 = 100$ and $x_2 \in (0.633, 2.78, 7.07)$, the middle one being unstable. Figure 3.4 shows computed trajectories from different starting points with the initial parameter values. The plot was made using pplane7 and the Dormand-Prince solver, and shows the systems two stabile equilibrium points, and the unstable one in $(x_1, x_2) = (100, 2.78)$.

Martinsen et al. (2004) uses a nonlinear combination of variables as the

Algorithm 2 doLineSearch

Require: Y_{new}, Y_{best}, u_{nom}, Δu, ξ and $N_{ls,max}$
  $n_{ls} \leftarrow 0$, $\alpha \leftarrow 1$
  $runLS \leftarrow$ true
  while $runLS$ **do**
    $n_{ls} \leftarrow n_{ls} + 1$
    $\alpha_{new} \leftarrow \xi * \alpha$
    $Y_{old} \leftarrow Y_{new}$
    $Y_{new} \leftarrow$ calcObjectiveValue$(u_{nom} + \alpha_{new} * \Delta u)$
    if $Y_{old} <= Y_{best}$ and $Y_{old} <= Y_{new}$ **then**
      $runLS \leftarrow$ false
      $\alpha_{best} \leftarrow \alpha$
    else if $n_{ls} == N_{ls,max}$ **then**
      $runLS \leftarrow$ false
      $warning \leftarrow$ no good step length found
      if $Y_{new} <= Y_{best}$ **then**
        $\alpha_{best} \leftarrow \alpha_{new}$
      else
        $\alpha_{best} \leftarrow 0$
      end if
    else
      $\alpha \leftarrow \alpha_{new}$
    end if
  end while
  return α_{best}

controlled variable to be held at a constant reference value,

$$y = k_1\sqrt{x_1}x_2, \tag{3.10}$$

giving the extended CV vector

$$x = \begin{bmatrix} x_1 & x_2 & y \end{bmatrix}^T. \tag{3.11}$$

The MVs are gathered in the vector

$$u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T. \tag{3.12}$$

To impose additional nonlinearities to the system, y^{ref} is introduced as an arbitrary reference trajectory for y to follow, transforming the problem into a trajectory tracking problem.

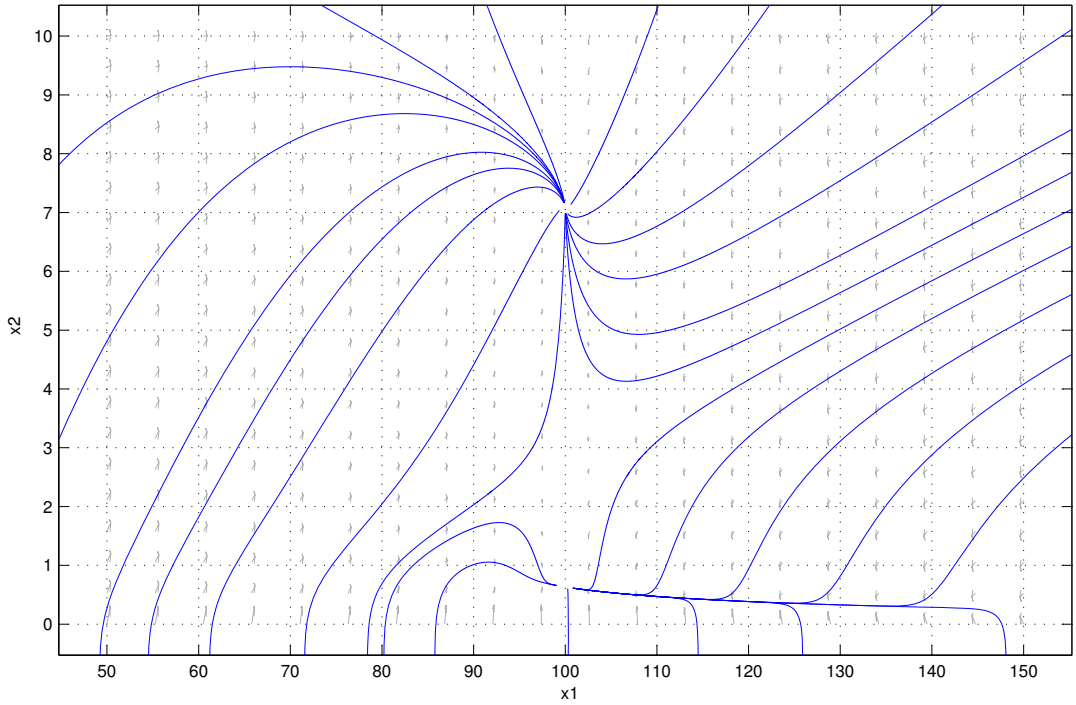


Figure 3.4: Solution trajectories for CSTR system

3.4.2 Algorithm Performance

This section presents the performance of the SSMQN algorithm applied to the CSTR example. The sampling time T_s was set to one second, and simulations where carried out for 20 seconds. T_c and T_p was chosen to 30 steps, parameterizing each input with 10 blocks, yielding a total of 20 optimization parameters considering both u_1 and u_2. Weighting matrices was chosen as

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 4e10^4 \end{bmatrix}, \quad R = P = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}, \tag{3.13}$$

where P is a penalty term added to the cost functional on movements in Δu giving

$$\begin{aligned} F(x,u) =& (x - x_{op})^T Q (x - x_{op}) + (u - u_{op})^T R (u - u_{op}) \\ &+ \Delta u^T P \Delta u. \end{aligned} \tag{3.14}$$

The system is attempted driven to the unstable equilibrium point $\begin{pmatrix} x_{1,op} & x_{2,op} \end{pmatrix} = \begin{pmatrix} 100 & 2.78 \end{pmatrix}$, $u_{op} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ and $y = y^{ref}$ for two different reference trajectories.

Simulations for both reference trajectories were done with different values for the linesearch factor ξ and the MV_{norm} convergence criterion, as described in Section

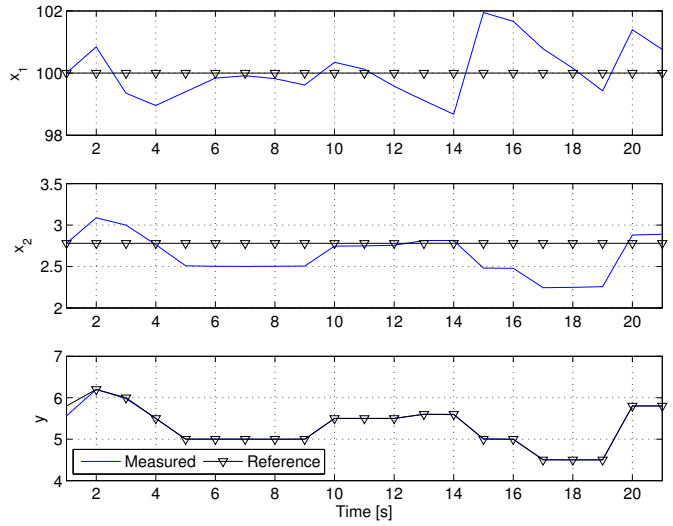


Figure 3.5: CSTR state plot for y_1^{ref}

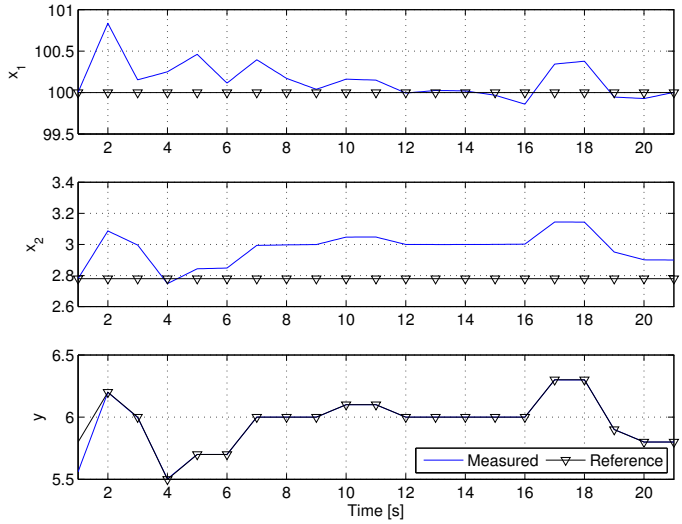


Figure 3.6: CSTR state plot for y_2^{ref}

3.3, to investigate the effects of these parameters on algorithm performance. Simulation results are shown in Fig. 3.5 for y_1^{ref} and in Fig. 3.6 for y_2^{ref}. We only show one simulation for each reference vector, since the state plots differ only slightly from one parameter setting to another. As we can se from both figures, the algorithm controls the system in a satisfying manner. The reference trajectory is closely followed by y, and only some slight deviations for x_1 and x_2, as a result of the tuning in Q. An additional remark on the plot for x_2 must be made in Fig. 3.6. The deviation from set point here is caused by the values

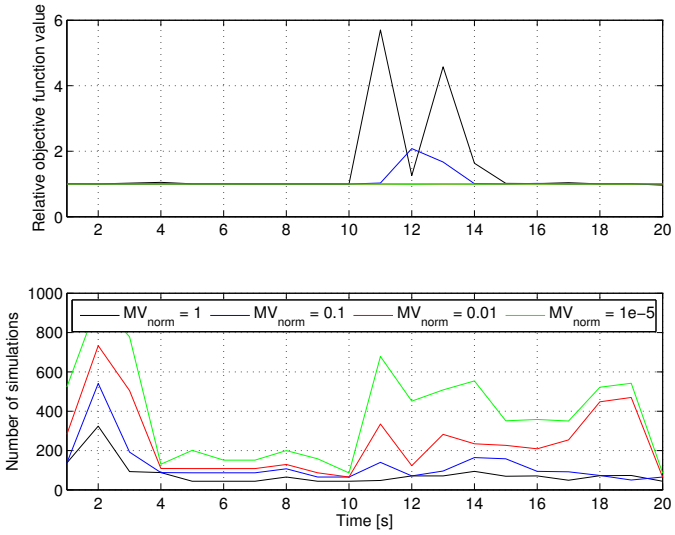


Figure 3.7: Algorithm performance for different MV_{norm} using y_1^{ref}

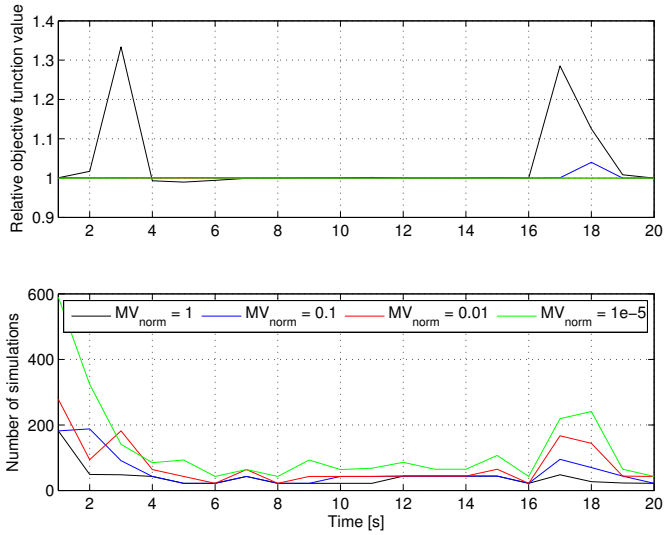


Figure 3.8: Algorithm performance for different MV_{norm} using y_2^{ref}

in y_2^{ref} which are found further away from the "nonlinear equilibrium"

$$y_{op} = k_1\sqrt{x_{1,op}}x_{2,op} \tag{3.15}$$

$$= 0.2 \cdot \sqrt{100} \cdot 2.78 = 5.56. \tag{3.16}$$

Fig. 3.7 and Fig. 3.8 shows the difference in the algorithm performance for different values of MV_{norm} for both reference trajectories. Both figures shows the performance for the different MV_{norm} relative to $MV_{norm} = 0.01$, along with the total number of simulations done at each sample interval for each

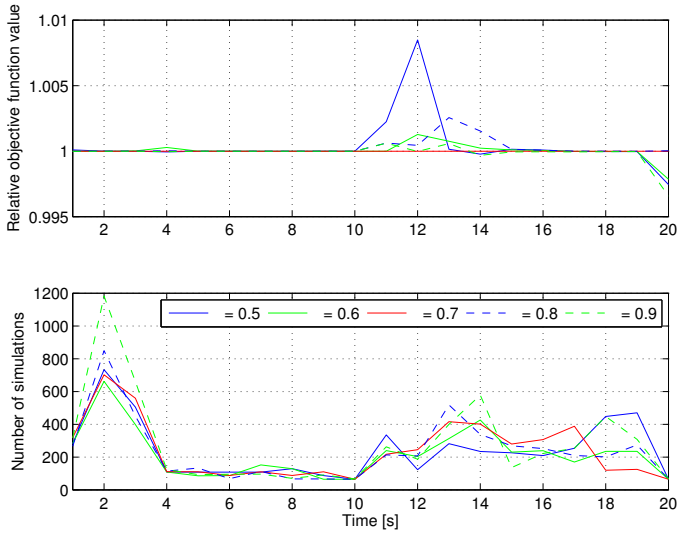


Figure 3.9: Algorithm performance for different ξ using y_1^{ref}

value of MV convergence criterion tested. As one might expect, the more relaxed MV_{norm} values perform more poorly then for the more rigid (lower) convergence criteria, but only in a few isolated samples, e.g. sample 11-14 in Fig. 3.7 and samples 2-3 and 17-19 in Fig. 3.8. The cost of a rigid MV_{norm} value is a high number of extra simulations at each timestep, and may in many cases not be justifiable. The figures show that performing an extensive number of extra simulations does in many cases not yield a significant objective value improvement. When using large commercial simulator models for prediction the simulation time overhead can be of considerable size if the algorithm performs additional simulations without producing objective improvement.

Fig. 3.9 and Fig. 3.10 shows the difference in the algorithm performance for different values of ξ for both reference trajectories. Both figures shows the performance for different ξ values relative to $\xi = 0.7$, along with the total number of simulations done at each sample interval for each value of backstepping factor tested. Here it is not so obvious what is the effect of the choice of ξ, as there are no clear trends shown like in the case of the varying MV_{norm}. The performance of the different values of ξ gives no indication of any value being superior to the others, and ought to be chosen from knowledge of the underlying process to be controlled. It should be noted that in these simulations $N_{ls,max}$ were chosen large enough to avoid any linesearch from being terminated before finding a satisfying step length, α_{best}. If the control application is facing tight real-time deadlines, leaving only a limited number of simulations available for the algorithm to converge and obtain a solution, the $N_{ls,max}$ must be chosen low. If this is the case, the algorithm performance will benefit from a lower ξ, as the linesearch will cover a greater part of the α interval $(0, 1]$ in fewer iterations.

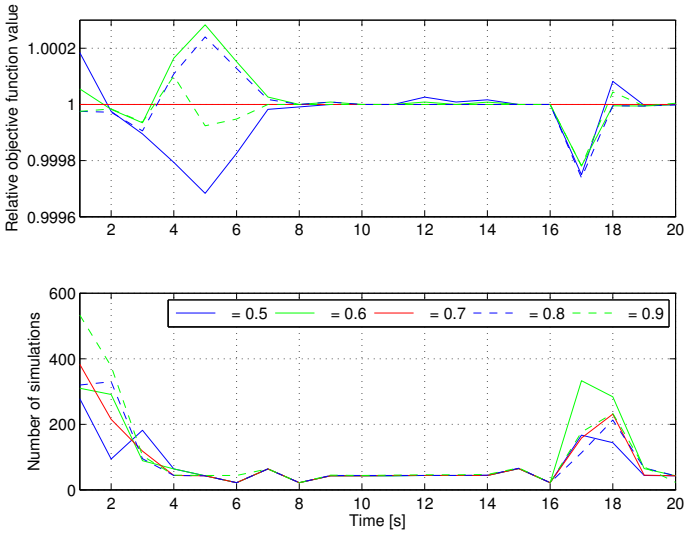


Figure 3.10: Algorithm performance for different ξ using y_2^{ref}

Fig. 3.11 and 3.12 shows bar plots of the total number of model simulations for each reference vector and for the different choices of MV_{norm} and ξ respectively. These figures further confirm the trends discussed earlier, as we can see a clear effect on the number of simulations from the value of MV_{norm}, the number of simulations increasing as the convergence criterion is tightened. The effect of the value of ξ is, however, not as evident. The problem seems to need a certain number of simulations for convergence, regardless of the value of ξ.

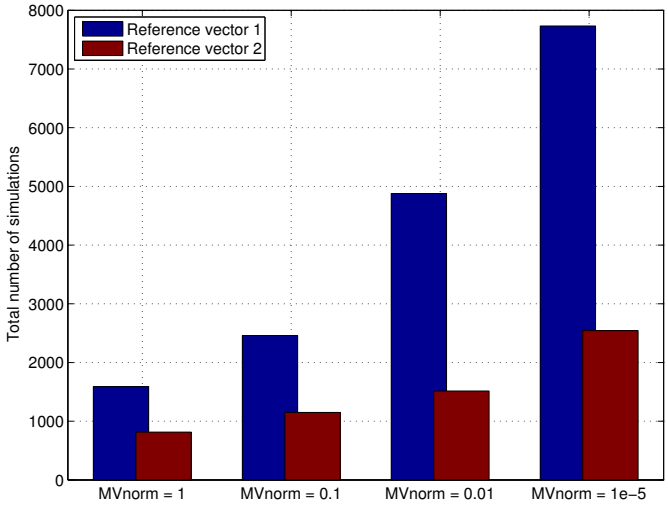


Figure 3.11: Total number of simulations for different values of MV_{norm}

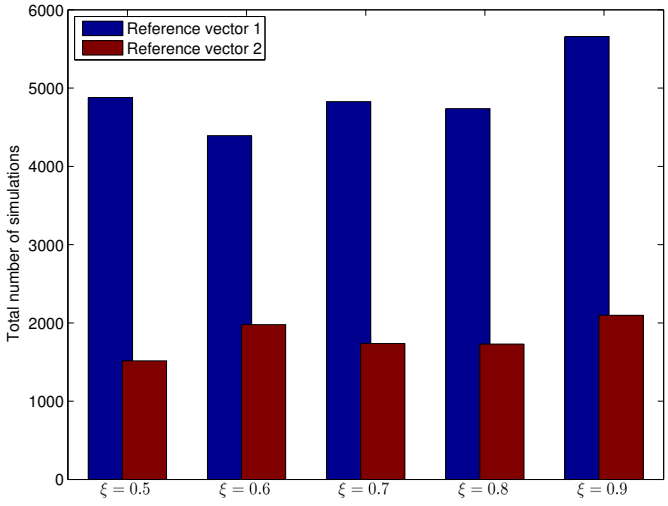


Figure 3.12: Total number of simulations for different values of ξ

The Reservoir Production Problem

4

The reservoir production problem is a complex problem, for many reasons. First of all, reservoir models tend to grow very large if set out to reflect some sense of realism compared to real life reservoirs. That is, realism in scale, behaviour or, preferably, both. Such models can consist of dynamic variables in the range of 10^3-10^6 (Heijn, Markovinović and Jansen, 2004), and certainly earn the right to classify as high-order models. In addition to the state variables, there are static variables in the same number of range, which possibly need to be estimated for model correction. Secondly, reservoir models are strongly nonlinear in their nature, as they involve multiphase flow in fields of variable geology properties (permeability and porosity). The reservoir model must be able capture the duality of the state dynamics involved, from the rapidly changing pressures to the more slowly varying saturation ratios of every fluid involved. This causes the state vector to show a clearly twofold character, and must be taken into consideration when optimizing production since reservoir control objectives normally are subject to a number of constraints. These constraints add another dimension to the problem complexity, as they often appear as nonlinear control-state path inequality constraints[3].

Facts like these where the reason why not much research was done on optimal reservoir production until some few years ago. Calculations were considered to be hard and too slow to acquire, and there were a lack of degrees of freedom in the available production inputs. This changed rapidly in the 1990's. First of all computational costs went down to an acceptable level. Even more important was the introduction of non conventional wells with downhole instrumentation. While conventional wells are vertically drilled, or only slightly deviated from the vertical line, nonconventional wells can either be horizontal, highly deviated or forked into multiple laterals (Yeten, 2003). A well of any of these sorts can in the greater sense be made with some completion instrumentation, and are then called smart (or intelligent) wells. Often this instrumentation is referred to as smart completion, which means wells are equipped with potentially everything from sensors, valves and other inflow control devices. However, the term smart well basically indicate a nonconventional well and is seldom found used on conventional wells with smart completion. This is because a conventional well

[3]E.g. maximum water injection rate constraints, maximum production water-cut constraints, gas/oil ratio constraints or similar

with smart completion can be considered as a multi branch well with separately controlled branches. Thanks to these technological improvements the needed flexibility for well operation was available, yielding a potential production increase if utilized in a proper manner.

4.1 Problem overview

Reservoir production strategy is often referred to as *reservoir flooding*. The target is obviously to maximize the production of the field, most often from a superior economical view. This economical target implies maximizing the total production in most cases, i.e. retrieving the highest recovery factor for the reservoir. To achieve the wanted production it is necessary to replace the produced fluids with another fluid to maintain pressure support in the reservoir. In most cases this is done by injection of some other fluid, preferably an undesired fluid present in the production rate (e.g. non-profitable gas) or perhaps water from the sea above. As production advances and the reservoir contents is gradually replaced, the contact front of product and replacement fluid is pushed towards the producing end. Hereof comes the term *flooding*, as one can imagine the contact front moving in a wave-like manner. This phenomenon is illustrated nicely in Fig. 4.1 found in Brouwer and Jansen (2004) and Sarma, Chen, Durlofsky and Aziz (2006). Without going too much into the details of the figure, one can clearly see the blue fluid (water) flooding the reservoir from left to right forcing the red product (oil) to the producing well.

Here we are at the core of the problem. In order to yield maximum recovery it is crucial to flood the reservoir in a way that ensures a contact front which is as smooth as possible. This way we can achieve close to a simultaneous breakthrough of the injected fluid at all the producing wells, and not the undesired breakthrough, or *coning*, at an early stage at only a few wells. The early breakthrough deteriorates the productivity of the reservoir since the injected fluid will more easily flow following the lines of the breakthrough channels, and pushing less on the oil or gas. In a sense one can see this as the injected fluid following a shortest path. It is therefore often desired to avoid or delay coning to the greatest extent possible, by selecting the appropriate production policy of the field.

The general reservoir optimization problem belongs to a class of optimal control problems subjected to what is known as *control-state path* inequality constraints. This basically means constraints that need to be satisfied in every time step. These constraints can possibly be nonlinear with respect to the chosen controls, i.e. from a controllable valve or bottom hole pressure (BHP) to a maximum flow constraint. It is acknowledged that such path constraints are particularly difficult to handle (Sarma et al., 2006). Hence, to maximize the production

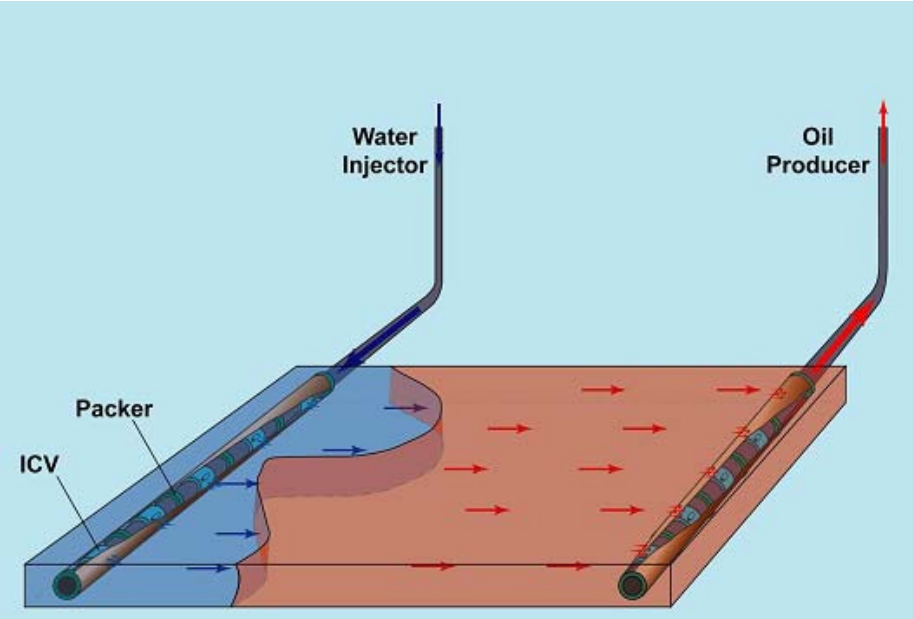


Figure 4.1: Reservoir flooding (Brouwer and Jansen, 2004; Sarma et al., 2006)

potential of a reservoir it is necessary to solve a constrained NLP. As the recent years have generated significant interest for closed-loop, model-based reservoir management, this report will continue with presenting some of the major contributions to the field.

4.2 Optimal Reservoir Control

Optimal dynamic multivariable control has already been used to a great extent in process industry in general, and also in the downstream segment of the oil industry. However, in the upstream end of the production line, advanced control systems have not yet gotten foothold, in fact many reservoirs are still run manually by operators and reservoir engineers. Considering the high energy demands and the rapidly increasing oil price of today, the oil companies have a huge potential for extra profits if they can increase their production totals. A small percentage increase in recovery could mean millions in additional revenues. Therefore, this is an exciting field for new control research, and has already been adressed previously by a number of authors.

Although some early attempt was made by Asheim (1988) and Virnovsky (1991), it is common practice to acknowledge the work of Sudaryanto and Yortsos (2000) as the first systematic approach to solve the fundamental problem. Considering a 2D geometry model with multiple injectors, this work investigates the optimal allocation of the available injection rate for the injectors

to optimize displacement efficiency. The displacement is viewed as a measure for the flooding efficiency, where the rate policy obtained from optimal theory is compared to a "conventional approach (...) to design symmetric well patterns, and allocate injection rates equally to the wells" (Sudaryanto and Yortsos, 2000). Displacement efficiency was selected for the time of break through of the injected fluid, as it was recognized that delaying this event is desirable for "many recovery processes". The work concludes with a "bang-bang" injection strategy to be optimal, for both the homogeneous and heterogeneous reservoir models investigated.

Brouwer, Jansen, van der Starre, van Kruijsdijk and Berentsen (2001) did an early study from a more practical point of view, focusing on the potential available through smart well technology. They delayed water breakthrough by controlling the injection and production rates. The rate strategy was found from a heuristic algorithm, which performed a *static* optimization. Static refers to finding the optimal input settings if these are to be kept constant over the time horizon. They concluded to show significant improvements for some simple reservoir models. However, they did recognize that even more improvement could be expected by *dynamic* optimization. As a contrast to the static case, dynamic methods allow the input to take different values over the time horizon. This was investigated in Dolle, Brouwer and Jansen (2002). Using gradient-based optimization they implement an iterative scheme, calculating the adjoint equation backwards in time to obtain the gradients with respect to the injection and production rates. The algorithm was tested on fields of different heterogeneity types, and confirming the expected improvement over the results from the static approach[4].

Common to all the above mentioned efforts are that they all consider cases operated at constant injection and production rates. This will hardly be feasible in practice for any real reservoir, as such constant rate policies in many cases would cause violation of production constraints, i.e. give unrealistically low BHPs. Brouwer and Jansen (2004) addressed this by applying dynamic optimal control to cases subject to two extreme well-operation conditions. The first case was subject to pure rate constraints, and the other was under pressure constraints only. Both cases showed reduction in both produced as well as injected water rates. In addition, for the rate constrained case the optimal strategy showed both production acceleration and an increase in the oil production totals.

In their work, Yeten et al. (2002) further investigated production rate optimization by utilizing features available in a commercial reservoir simulator,

[4]They managed to show improvements in most cases, though there were to cases where the static solution gave better result. This was blamed on numerical instability problems with the adjoint equation.

ECLIPSE (Schlumberger, 2006). Although costly in computation time, using ECLIPSE for model prediction should improve the reliability for most reservoir cases over the simpler models used in the other publications. Particularly, using a highly developed simulator tool like the ECLIPSE will most certainly make it easier to incorporate models of more complex geological structures and representation of different smart well segments such as inflow control valves (ICVs). Hence, the sense of realism can be increased in the problems addressed. Yeten et al. (2002) describes a optimization procedure for control of ICVs for smart wells, using a conjugate gradient (CG) algorithm to solve the NLP. At a given time step of the simulation, they calculate the optimal static ICV setting for the rest of the prediction horizon. These settings are applied for one time step, and then a new set of optimal static ICV settings are found. As an analogy to the MPC methodology presented in the previous chapters, this can be viewed as a fixed horizon MPC with only one blocking for the inputs, i.e. one optimization parameter for each input m. The gradients are found numerically using a forward finite difference approximation. To test their proposed algorithm they considered a number of fairly large models of 3 dimensions with 3 phases present (oil, gas and water). The results showed significant improvement in oil production totals for the optimized case here as well. However, by representing geological uncertainty by different model realizations, optimization results also showed large variations for a type of fluvial reservoir models. This was an important recognition, as it was found that not all reservoirs may justify the extra cost of smart well completion.

Sarma et al. (2006) presented a gradient based optimization scheme with improvements in computational efficiency compared to similar methods previously proposed. They use an approximate feasible direction NLP algorithm, which only uses two evaluations of the adjoint function in all iterations, one for the objective function gradient and one for a gradient made from combining the active constraints. Besides the benefits from an increase in computational efficiency, they argue for guaranteed feasibility for all iterates by explicitly "solving the constraints during the forward model evaluation". Implementing test cases on a in-house simulator facility at Stanford University providing the adjoint models directly from the simulator, they managed to show similar results compared to a case also used in Yeten (2003). However, requiring only a fraction of the total simulations Yeten (2003) needed to generate numerical gradients, these results are acquired with far less computational effort.

Although Yeten et al. (2002) introduced an aspect of model uncertainty by considering algorithm performance for a number of slightly different models of the same characteristics and size, the above mentioned studies all assume to have an exact model with all properties known a-priori. In reservoir engineering this is seldom assumed to be the case, since all reservoir modelling is based on

data from geological measurements with a high degree of uncertainty. Thus, a reservoir model can only be expected to be a rough approximation of the actual field at best, and therefore hold limited predictive value. In petroleum industry, there is therefore a large subject area devoted to reservoir model updating based on actual production data (e.g. BHPs and multiphase flows), known as *history matching*. The idea is to fit the model to the available measurements by tuning the geological parameters. Traditionally this problem has been solved in a least squares sense, finding a set of model parameters with the best fit between model output and production history. For closed-loop reservoir control with model mismatch it will be necessary to include a model updating scheme to achieve satisfying performance. This is especially important with respect to violation of constraints. This was recognized in Brouwer, Nævdal, Jansen, Vefring and van Kruijsdijk (2004) as they proposed way to combine optimal reservoir control with online model updating from measured data. Here, they extended their previous work from Brouwer and Jansen (2004) by incorporating an estimator methodology known as the *ensemble Kalman filter* (EnKF) from Nævdal, Johnsen, Aanonsen and Vefring (2005). They found that when starting of with an erroneous model for predictions, they were able to produce results close to those from the optimization with a perfect model, as the EnKF proved to capture the reservoirs permeability field after only a few simulation steps.

As the order of the reservoir grows larger, e.g. increasing scope form near-well models to complete fields, optimization will become more costly. To more efficiently handle such large models Heijn et al. (2004) investigated methods for model reduction for a two-phase heterogeneous reservoir. Discussing both mathematical reduction of first principle models and identification techniques for low-order black-box models, they found promising results especially for two methods, *proper orthogonal decomposition* (POD) and *subspace identification*. In practice, most reservoir models are given in a commercial simulator description and not by explicit model equations. In these cases mathematical reduction methods, like the POD, is unsuitable, and pure identification methods, like the subspace identification method, should be chosen of the two. Subspace identification was used in a previous report by the author (Meum, 2006) to derive low-order model representations of a simulator model to use for predictions in a MPC. The identified models were able to show predictive qualities for a limited time window, but deteriorated as nonlinearities became more influential as the simulation time elapsed. The report remarks that even though one could potentially develop a set of models to represent different part of the reservoir life time, it would possibly be a better solution to introduce some adaptive scheme for continuous model updating. E.g. use the subspace identification method on-line, using simulator predictions and production data, to update the identified model in each step. Saputelli, Nikolaou and Economides (2006) presented an analogous approach to this, with a *Self-learning Reservoir*

model. They use a hybrid model for predictions, based on basic fluid mechanics, which is updated on-line through parameter estimation by standard linear regression. By dividing the optimization problem into different levels depending on their dominant time constants, long term optimization was done on an upper level, while short term optimal control was found from an MPC application utilizing the adaptive hybrid model. The upper level provides set points for the MPC, which performs as close to these as possible. Saputelli et al. (2006) were able to show a considerable increase in production efficiency for a multi-well reservoir model.

This concludes the overview of the reservoir production problem. The interested writer should seek to get familiar with some of the papers mentioned here for a more detailed picture on each of them respectively. For a nice survey on offshore real time production, Bieker, Slupphaug and Johansen (2006) is recommended. They outline every aspect of upstream production, from well to reservoir, for a bigger picture. Also highly recommended is Yeten (2003) which addresses optimal reservoir production from the optimal placement of wells, to optimal flooding strategies and considerations on geological uncertainty, in a great detail.

Reservoir Simulation

5

This chapter gives a brief introduction to reservoir modelling and simulation. First an overview of the model description is given, along with the basic model equations. Then the commercial simulator, ECLIPSE, is presented with some basic properties.

5.1 Reservoir Model Description

A reservoir model is a geometric model which contains a detailed description of geological properties and the dynamics of complex multiphase flow in a porous media. Fig. 5.1 shows an example model for illustration. The colouring scheme can be considered to represent a physical property (e.g. permeability) in the different regions of the geological structure. The reservoir model will

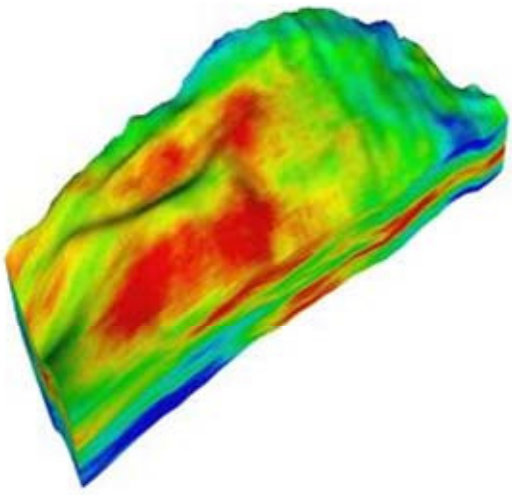


Figure 5.1: Illustration of a reservoir model

try to describe the different static properties in all regions, along with the dynamic variables, i.e. saturations and pressures. Preferably one could wish to describe these model parameters continuously in space, to best capture the true reservoir dynamics. For reservoirs containing discontinuities, such as barriers, a piecewise continuous model would be desirable. However, the model needs to

be discretized to use in simulations, and a continuous description can become excessively complex. The common approach is therefore to model the reservoir with spatial blocks, as a discretization in space. The blocks are known as grid blocks. Appropriate static and dynamic parameters are assigned to each grid block, found from geological measurements, e.g. collected during drilling or from seismic data. As the parameters are associated with a degree of uncertainty, the model will often need to be updated by history matching several times through the reservoir lifetime.

The other important aspect of reservoir modelling concerns the description of multiphase flows. The reservoir may contain water, liquid hydrocarbon substances of different types or gas, in any combination. During production fluids will be removed and replaced from the reservoir, as described in Section 4.1. Hence, it is crucial to have a good model of the flow behaviour to select the optimal production strategy. The next section describes the basics for a particular mathematical multiphase flow model widely used in reservoir modelling.

5.2 Model Equations

There exist various types of different mathematical models for the flow of multiphase fluids in porous media. One of these types is the "Black-Oil" model, where it is assumed that, in addition to water, there are two phases of hydrocarbon substance (oil and gas) present in the reservoir. The Black-Oil equations can also apply to a two-phase representation if assuming no gas in the reservoir. The model can be expressed as a set of partial differential equations, describing the saturation and pressure in the fluids. Basically these equations arise from conservation of mass and an assumption on the relation between the flow velocity and the pressure known as Darcy's law. Dolle et al. (2002) gives the equations on the form

$$\nabla \cdot \left(h\rho_l \frac{k \cdot k_{rl}}{\mu_l} \nabla p_l \right) + hq_l\rho_l = h\frac{\partial \left(\rho_l S_l \phi\right)}{\partial t}, \quad \forall\, l \in \{o, w, g\}\,, \tag{5.1}$$

where h is grid block height, ρ_l is the density of the liquid, k_{rl} is the relative permeability of the liquid, k is the absolute permeability, μ_l is the viscosity of the liquid, ϕ is the porosity, S_l is the liquid saturation, p_l is the liquid pressure and o, w, g refers to oil, water and gas respectively. Some of these parameters are functions of the reservoir coordinates, such as the permeability and porosity, if one is not making the naive assumption of a completely homogeneous geology (e.g. $\phi = \phi(x, y, z)$). We recognize the nonlinearity of the system for the dynamic states p_l and S_l.

In order to include (5.1) in a numerical simulator, and to fit the grid divided reservoir description, the equations need to be discretized, with respect to time and space respectively, as mentioned in Section 5.1. This is done according to the Simultaneous Solution method (Peaceman, 1977; Aziz and Settari, 1986). The spatial discretization can be done either for corner-point space grid, or block-centred grid. Variable grid size can also be applied to decrease the number of variables and parameters.

A state vector can be defined as

$$x(t) = \begin{bmatrix} \mathbf{p}_l(t) & \mathbf{S}_l(t) \end{bmatrix}^T, \tag{5.2}$$

and an input vector as

$$u(t) = \mathbf{q}_l(t), \tag{5.3}$$

where the bold face of $\mathbf{p}_l(t)$, $\mathbf{S}_l(t)$ and $\mathbf{q}_l(t)$ denotes they contain the state variables for every grid block, stacked as vectors.

Now, one could make a complete system description by introducing the proper matrices containing grid permeabilities, porosities and mass as described in Heijn et al. (2004) on the form

$$\begin{aligned} \dot{x}(t) &= \Psi^{-1}V^{-1}Tx(t) + \Psi^{-1}u(t) \\ &= f(x(t), u(t)), \end{aligned}$$

where Ψ is a matrix containing pore fluid compressibility and porosity, V is a matrix of grid cell masses and T is a transmissibility matrix.

Such an approach will not be further pursued here, as this report relies on a commercial simulator for modelling. Suffice to say that implementing such a model for simulations would be time consuming and complex, especially for the 3D, three-phase reservoir. The next section continues with a description of the ECLIPSE reservoir simulator.

5.3 ECLIPSE 100 - A commercial reservoir simulator

ECLIPSE (Schlumberger, 2006) is an oil and gas reservoir simulator developed and distributed by Schlumberger. Being used by several oil and gas companies world wide, ECLIPSE can be considered one of the most acknowledged reservoir simulators in the industry.

The ECLIPSE is a fully-implicit, strongly coupled, three-phase, three-dimensional, general purpose Black-Oil simulator[5]. This means ECLIPSE solves the Black-Oil equations in (5.1) using a IMPES (IMplicit Pressure Explicit Saturation)

[5]In addition, ECLIPSE also have an option available to model gas condensate

method. The program is written in FORTRAN77 and can be used on any computer with an appropriate compiler and sufficient memory available. Although ECLIPSE is a three-phase model, it allows for two-phase options, solving only a two component system to save computational cost (i.e. memory and time).

Input data for simulations can be prepared in a free format using a keyword system. One might be tempted to call it intricate considering the number of keywords available, which is a few thousand. Even so, the system is simple in the way that any standard editor can be used to prepare the input file. All that is required is that the input file must have the postfix `.DATA`, and that it includes the following sections in correct order:

- `RUNSPEC` – General model description, such as title, dimensions, phases present, etc.
- `GRID` – Here the model geometry must be specified, i.e. the grid partitioning and and geological properties.
- `EDIT` – (Optional) Modifications to calculated pore volumes, grid block centre depths and transmissibilities.
- `PROPS` – Tables of properties of reservoir rock and fluids as functions of fluid pressures, saturations and compositions.
- `REGIONS` – (Optional) To define different regions which should differ in properties, e.g. initial conditions.
- `SOLUTION` – Is used to specify the initial conditions in the reservoir.
- `SUMMARY` – (Optional) Specifies the required form of the system outputs, and which values the simulator should output.
- `SCHEDULE` – This section gives the production specifications, such as production rates, BHPs etc. This section offer several different options for built in well control.

As an alternative the ECLIPSE Office package may be used to prepare data interactively through a graphical interface. This package also offers tools for visualization of model and simulation results.

Each of the different sections must be specified by a number of other keywords. These will not be explained here, instead the report refers the reader to the User's Manual in Schlumberger (2006). Though, a comment should be made regarding the `SCHEDULE` section offering well control. The user must specify the wanted type of well control, and operational bounds on the production rates and well pressures. ECLIPSE then operates the wells at the wanted set point, either at a wanted BHP or a given production rate (e.g. the total liquid rate

produced (LRAT)), while satisfying the specified constraints. To illustrate; a well specified for BHP control will at all times produce the rate corresponding to the set point pressure. If this corresponding rate violates a specified rate constraint, ECLIPSE will switch to rate control, until the rate constraint allows it to again produce at the set point pressure. Also, ECLIPSE includes an option to model a well as multiple segments, through a strongly coupled model for wellbore flow. Yeten et al. (2002) explains that the "multi-segment well option uses the drift flux model for the representation of multiphase flow in the wellbore, which enables the phases to flow with different velocities in the well". The different segments can then be viewed as ICVs, as the segments can adjust the area open for flow, i.e. can be incrementally opened or incrementally closed. This report will in the next chapter present a case study where well segments are modelled as ICVs. See Section 6.2 for a more detailed explanation.

5.3.1 Interfacing with ECLIPSE

For the control engineer, it will be important to be familiar with the basics of ECLIPSE and to have knowledge of important different keywords offering functional means to use it for modelling purposes in an optimal control application. Most of these are options are available trough the `SCHEDULE` section, providing a number of different control alternatives. The reader is referred to the ECLIPSE Reference Manual (Schlumberger, 2006) and advised to study the parts on the `SCHEDULE` section in some detail before setting off to design an control application to wrap around an ECLIPSE model.

Independent of what keyword facility found suitable for the control problem at hand, an application must either way establish the correct interface to the simulator for efficient use of the model. The obvious choice for such an interface is through file I/O, since ECLIPSE reads all simulation data from text files. By writing different text files with the appropriate keywords for production data for each simulation step, and make sure to include them in the `.DATA` file, the application can extract the wanted information from the model. ECLIPSE writes all data to files with a user specified format, so the application can obtain simulator output using file I/O for this as well.

Another important ECLIPSE feature, which should be used for efficient use in optimization, is the `RESTART` file option. As the optimization progresses forward in time, it would be extremely inefficient to re-run the `.DATA` file from the start time at each sample. This should be avoided by restarting the model from the end of the previous sample. `RESTART` files can be requested from ECLIPSE by including the appropriate keywords in the data file (see Schlumberger, 2006, pp. 2075–2080). This saves all time-invariant data, e.g. geological properties, to a separate file, and all time-dependent data to subsequent restart files. ECLIPSE

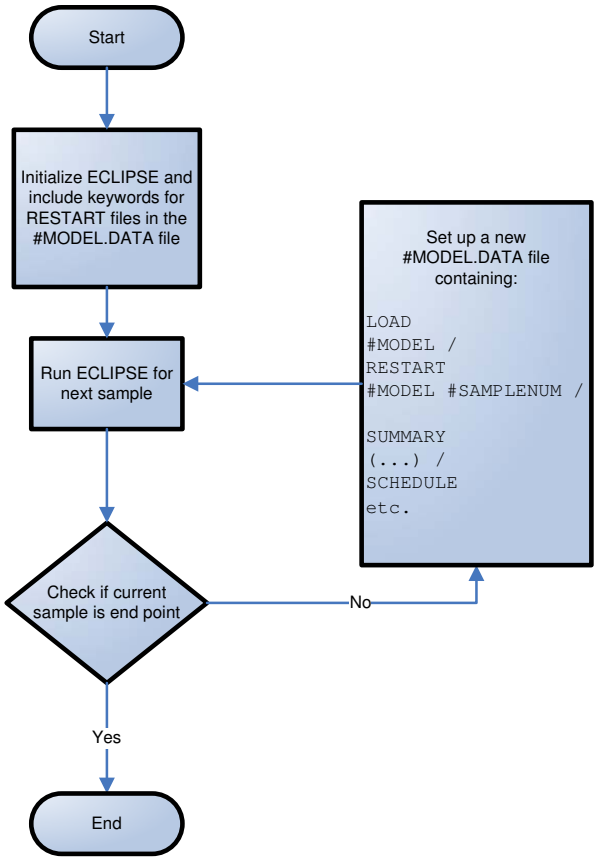


Figure 5.2: ECLIPSE restart procedure

can then be restarted from previous time steps by specifying the sample number. The procedure is illustrated as a flow chart in Fig. 5.2.

Applications

6

This chapter contains detailed descriptions of the two reservoir models applied to optimization in this report. The first model is a simple, cubic model, designed as a multi-purpose model to use in student projects. As the model is fairly small, it functions well as an introduction to reservoir modelling and control for the non-petroleum engineer. The second model is a strongly channelized reservoir, previously investigated in the work of Yeten et al. (2002) and Yeten (2003). This model is larger in dimensions, and also represents a more realistic reservoir. Hence, the model is more interesting from a control point of view, since the solution to the optimal control problem is not as trivial as it is for the simpler case. The section also includes some description of special implementation adjustments and particular important properties for each model.

6.1 Case 1: The Shoe Box model

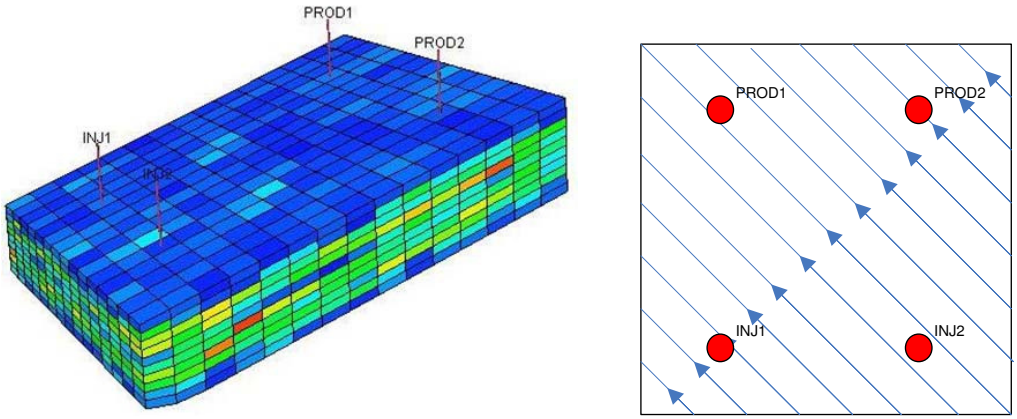
The first model implemented for optimization is a small, spatial reservoir with heterogeneous porosity and permeability fields, named the Shoe Box model. The model has previously been used in Meum (2006) and in an concurrent Master's thesis on EnKF by Jensen (2007)[6]. The reservoir is shown in Fig. 6.1(a), also displaying the placement of wells in the reservoir. With two injectors, INJ1 and INJ2, and two producing wells PROD1 and PROD2, available for control, this represents a multi-variable control problem.

The model is a two-phase model, containing no gas, only oil and water. Structurally the field consists of two high permeable layers separated by a layer of lower permeability, and flow lines tend to go diagonally from the lower right to the upper left of the reservoir, see Fig. 6.1(b). This gives a varying water flooding time for the different vertical layers in the model. A more detailed plot of the geological properties are given in Appendix A, Fig. A.1, showing grid permeabilities and porosities in each layer.

Some of the reservoir properties are presented in Table 6.1. We notice the number of grid cells

$$\bar{m} \times \bar{n} \times \bar{k} = 15 \times 15 \times 10 = 2250,$$

[6]The models conceptually equal, though have some small differences in properties.



(a) Model overview (b) Flow line structure

Figure 6.1: The Shoe Box reservoir

and the total size of the reservoir

$$(\bar{m} \cdot \Delta x) \times (\bar{n} \cdot \Delta y) \times (\bar{k} \cdot \Delta z) = (15 \cdot 94) \times (15 \cdot 80) \times (10 \cdot 5.6)\text{m}^3$$
$$= 0.095 \text{ km}^3.$$

Variable	Description	Value	Units
$\bar{m}$, $\bar{n}$	grid blocks in x and y direction	15	-
$\bar{k}$	grid blocks in z direction	10	-
Δx	grid block width	94	m
Δy	grid block width	80	m
Δz	grid block width	5.6	m
h	reservoir depth	2600	m
$P_{o,init}$	initial pressure	260	bar

Table 6.1: Reservoir properties

Even though this is a multivariable control problem, the model is very symmetric. To make it more asymmetric, the injectors are assigned to operate in different layers of the reservoir. This is done by having the injectors perforated in different segments. Fig. 6.2 shows how the wells are perforated, INJ1 in layers 5-7 and INJ2 in layers 2-4 respectively.

6.1.1 Base case definition

The reservoir base case is defined by the following operational conditions. The producers are to produce a constant LRAT of 2000 SM^3/DAY each, while maintaining a minimum BHP of 200 bar and a minimum tubing head pressure

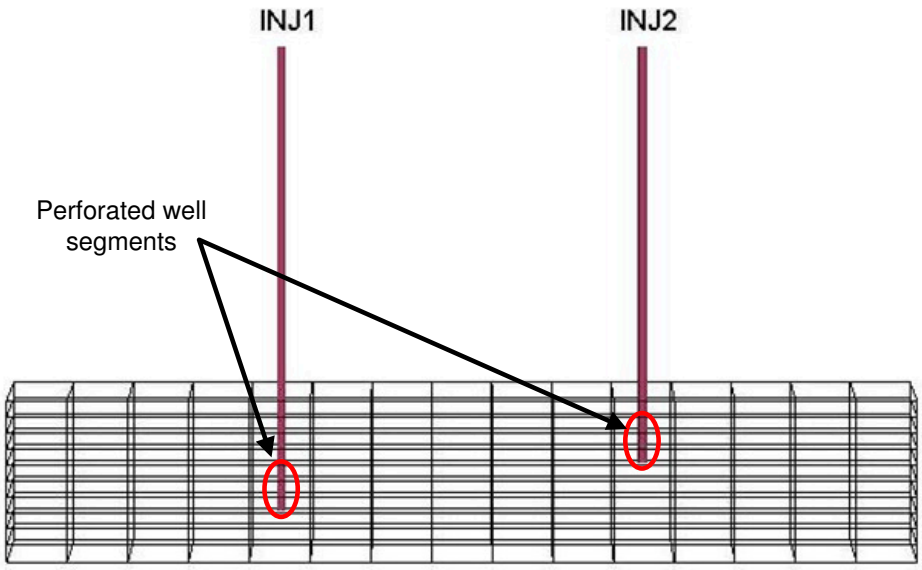


Figure 6.2: Perforation of Shoe Box injectors

(THP) of 10 bar for lift considerations. The THP is given from a frictionless model

$$p^t = p^b - (\rho_w * q_w + \rho_o q_o) \cdot \frac{gh}{q_l}, \tag{6.1}$$

where p^t is the well THP, p^b is the well BHP, ρ_w is water density, ρ_o is density of oil, q_w is the water production rate, q_o is the oil production rate, q_l is the liquid production rate, g is the gravity constant and h is the reservoir depth. The multiphase rates are given from ECLIPSE.

The maximum production rate water cut (WCT) is 0.9. To maintain reservoir pressure, INJ1 and INJ2 are injection a water rate (WIR) of 2500 SM^3/DAY while maintaining a maximum BHP of 400 bar. Reservoir is simulated for a total time of 6 years, i.e. 2190 days.

6.1.2 Implementation

The SEPTIC control problem is formulated to maximize the field oil production rate from the two producers over the prediction horizon 720 days. With a 30 day sample time, this gives a prediction horizon , $T_c = T_p = 24$ steps.

MVs are the injection and production rates at the four wells, $q_k^{\mathcal{W}}, \mathcal{W} \in \{$INJ1, INJ2, PROD1, PROD2$\}$. The SEPTIC application optimizes the distribution of the available production and injection rates on the different wells. The available rates are determined from the production rates of the base case. E.g., the

injectors have a total rate of 5000 SM^3 available per day, and the producers a total of 4000 SM^3/DAY available. Also, all wells have a high rate constraint at 3000 SM^3/DAY.

The same pressure constraint as in the base case was imposed, as well as the WCT constraint. This gives the following problem formulation

$$\min_{q^{\mathcal{W}}} \sum_{k=t}^{t+T_p} q_k^{\text{FOPR}} - q_k^* \tag{6.2a}$$

subject to

$$q_k^{\mathcal{W}} \leq 3000 \text{ SM}^3/\text{DAY} \tag{6.2b}$$
$$q_k^{\text{INJ1}} + q_k^{\text{INJ2}} \leq 5000 \text{ SM}^3/\text{DAY} \tag{6.2c}$$
$$q_k^{\text{PROD1}} + q_k^{\text{PROD2}} \leq 4000 \text{ SM}^3/\text{DAY} \tag{6.2d}$$
$$200 \text{ bar} \leq p_k^{b,\mathcal{W}} \leq 400 \text{ bar} \tag{6.2e}$$
$$p_k^{t,\mathcal{P}} \geq 10 \text{ bar} \tag{6.2f}$$
$$r_k^{\text{WCT}} \leq 0.9 \tag{6.2g}$$
$$q_k^{\mathcal{W}} \geq 0 \text{ SM}^3/\text{DAY} \tag{6.2h}$$

where q_k^{FOPR} is the field oil production rate at time k, q_k^* is an unattainable set point, $q_k^{\mathcal{W}}$ is the flow rate in the set of wells $\mathcal{W} \in \{\text{INJ1}, \text{INJ2}, \text{PROD1}, \text{PROD2}\}$, $p_k^{b,\mathcal{W}}$ is the well BHP, $p_k^{t,\mathcal{P}}$ is the well THP for the set of producers $\mathcal{P} \in \{\text{PROD1}, \text{PROD2}\}$ and r_k^{WCT} is the production WCT. All four MVs are parameterized as five piecewise continuous blocks over the control horizon, giving SEPTIC 20 parameters to optimize in each sample.

Using the file I/O interfaces between SEPTIC and ECLIPSE, the production rates was given directly to ECLIPSE through use of the `WCONPROD` keyword.

6.2 Case 2: A fluvial reservoir model

The second case model is a model provided by Dr. Burak Yeten, holding a PhD from Stanford University. Previously used for publications (Yeten et al., 2002; Yeten, 2003) this model has already proved potential for production optimization, and is as such an excellent opportunity to validate the quality of the algorithm developed in this report, through comparable results.

Rendering the reservoir description given in Yeten et al. (2002), this model represents a virtual, North Sea type *fluvial* reservoir, made using a software called `fluvsim`. Fluvial means that the reservoir is channelized, as it contains impermeable zones through the layers. This is illustrated in Fig. 6.3, where the

dark blue colour shows the areas of high permeability. Model dimensions are $5000 \times 5000 \times 100\,\text{ft}^3$, and the model contains three phases. A detailed list on the other model parameters can be found in Appendix B, Table B.1. Gas is present at the top of the field, while water is represented through an aquifer[7]. The aquifer provides such a strong connection to the reservoir, that no injections are needed for additional pressure support.

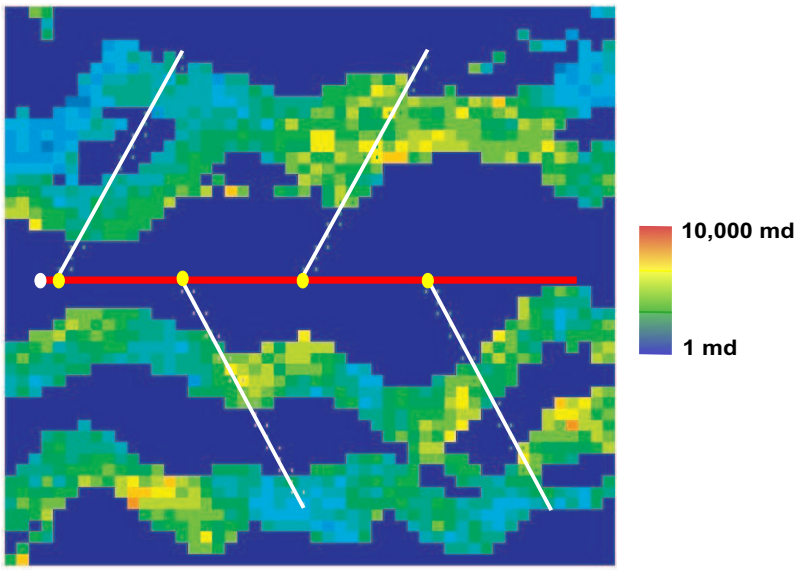


Figure 6.3: Fluvial reservoir model (Yeten et al., 2002)

The reservoir contains a multilateral horizontal well, also shown in Fig. 6.3. The red line shows the unperforated mainbore, while the white lines connected to the main bore, are four fully perforated laterals. As one can see, the laterals intersect with both permeable and impermeable zones. The length of the laterals are approximately 2150 ft long, and the well is placed 15 ft above the water-oil contact, giving rapid water-breakthrough after production start. To control each lateral they are all connected to the main bore with a valve (ICV) at the lateral heel, indicated in Fig. 6.3 by the yellow circles. In addition there is a valve at the start of the main bore, giving control over the total flow rate in the well (white circle).

ICVs are introduced in the ECLIPSE model as well segments with adjustable cross-sectional area (Schlumberger, 2006). The pressure drop in the segment is then given as the sum of the frictional pressure loss and the pressure loss due to the valve opening, and is modelled as a sub-critical flow in a pipe with a

[7]An aquifer can be seen as a water reservoir in connection with the hydrocarbon reservoir, providing pressure support by continuously replacing the produced volumes through water influx.

constriction. Following the definitions given in Yeten et al. (2002), then:

$$\Delta p_t = \Delta p_c + \Delta p_f, \tag{6.3}$$

where Δp_t is the total pressure drop, Δp_c is the pressure drop resulting of flow through the constriction and Δp_f is frictional pressure loss. The terms in (6.3) are calculated from the two equations

$$\Delta p_c = C_u \frac{\rho_m v_c^2}{2C_v^2}, \tag{6.4}$$

$$\Delta p_f = 2C_u f \frac{L}{D} \rho_m v_p^2, \tag{6.5}$$

respectively, where ρ_m is the density of the multiphase flow in the segment, C_u is a unit conversion factor[8], C_v is a dimensionless valve coefficient, v_c and v_p a flow velocities through the choke (denoted by subscripted c) and pipe (subscripted p). The Fanning factor enters the equations by f, and L and D are, respectively, the length and the diameter of the well segment. By defining the open cross-sectional area of the choke, A_c, and introducing this as an input in (6.4) through the connection

$$v_c = \frac{q_c}{A_c}, \tag{6.6}$$

the pressure drop over the constriction can be controlled for the current flow rate q_c.

The valve setting is specified to the ECLIPSE model for every time step using the `WSEGLABY` keyword in the `SCHEDULE` section, by explicitly declaring the flow area A_c. The flow area available is dependent on the diameter of the well segment, D, and can take any value in the set $(0, A_{max}]$, where

$$A_{max} = \frac{1}{4}\pi D^2 = 0.1363 \text{ ft}^2, \tag{6.7}$$

for $D = 0.4166667$ ft. Notice the open end bracket, as A_c is not allowed the value of zero. Referring to (6.6), this would cause obvious problems in the simulation. In the control application the valve settings are considered to lie in the interval $(0, 1]$, which is acquired by a simple transformation.

Reservoir models used in practice are made from geological and seismic data. These are often associated with different degrees of uncertainty. To reflect this uncertainty, this model is therefore reproduced as five unconditional realizations from a geostatistical distribution. In Fig. 6.4 a histogram with the global statistics of the permeability for all five models, along with some basic statistical properties shown in Table B.2 found in Appendix B. Well properties, like

[8] All units used in Case 2 are field units

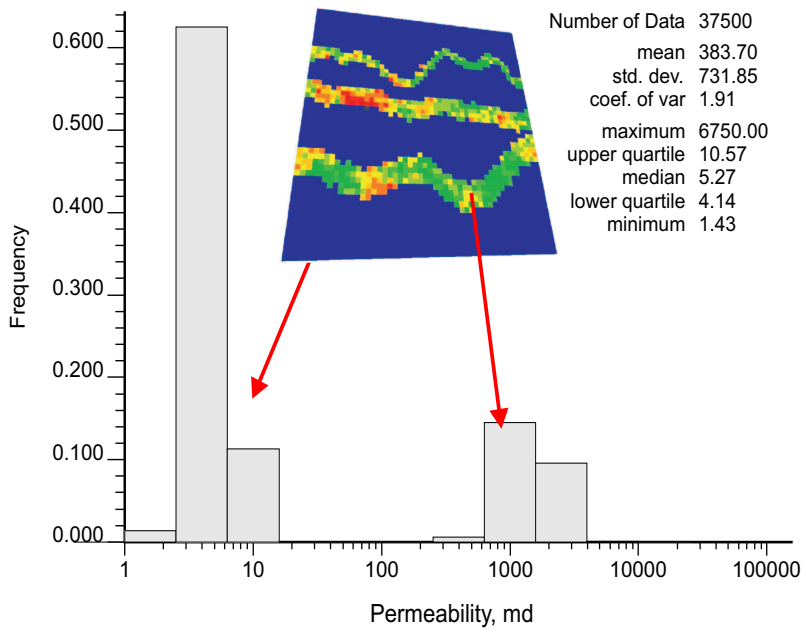


Figure 6.4: Permeability distribution for the 5 model realizations

location, architecture and instrumentation are the same in all five models, so all that differs are the surrounding permeability fields. An illustration of the different models is shown in Fig. 6.5. The idea is to investigate production increase potentials for equally probable models, to see if instrumentation and optimization applications are justifiable given model uncertainty.

6.2.1 Base Case Definition

To compare the value of the optimized production results, a definition of a base case is needed. This section describes a set of simple production conditions, which functions as the base case. The conditions was chosen to be exactly the same as found in Yeten et al. (2002). LRAT was specified to 10.0 MSTB/DAY, with a constraint on gas production not to exceed an gas/oil ratio (GOR) over 5.0 MSCF/STB. If the constraint on GOR was violated, the LRAT was cut by 10% every time the constraint was reached. Hence, the base case is not an uncontrolled case, as it includes some simple constraint handling offered by ECLIPSE. This control was necessary to ensure some stability in the simulator model. There was however no control on the lateral valves. In fact, these was not modelled in the base case, but replaced as well segments of ordinary tubing. There was also specified a constraint on the WCT of the production rate at 0.8%, to pose as an economical constraint. When WCT reached this value, the well was specified to close down, as it was considered no longer to be

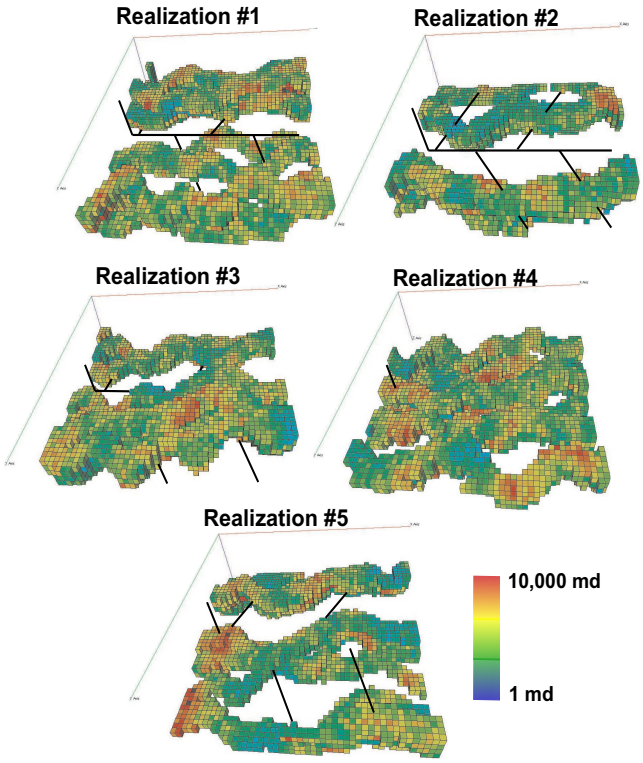


Figure 6.5: The five geostatistical realizations

profitable. The pressure at the well was not to go below 1500 psi, given from lift considerations. However, this constraint never got active in any of the cases, as the aquifer supplies sufficient pressure support.

These models has been investigated in Yeten et al. (2002) and Yeten (2003). Hence, their results are also available for comparison. A short description of their optimizations have already been given in Section 4.2, and will not be repeated here, but some major differences will be pointed out when appropriate. The rest of this report will draw comparisons from the results produced here with both the base case and the results presented in Yeten et al. (2002). The latter will from here on out be referred to as SPE 79031.

6.2.2 Implementation

This section presents some of the most important implementation aspects and adjustments in interfacing the Case 2 models with the SEPTIC application and SSMQN algorithm.

The MVs are chosen is the opening of each of the four valves in connection with the laterals. This is similar to the inputs used in SPE 79031. Also, we define the

LRAT as a MV, which is necessary to meet the production constraints on GOR in some of the model realizations. This gives the application an advantage over the base case and the SPE 79031 case, as both of these seeks to fulfil the GOR constraint by the ad hoc solution of cutting back production by 10% whenever the constraint becomes active. Such a solution is obviously only suboptimal at best, and the better approach will be to seek optimal production rate as well. The five MVs are then given in the vector

$$u = \begin{bmatrix} v_{\mathrm{L1}} & v_{\mathrm{L2}} & v_{\mathrm{L3}} & v_{\mathrm{L4}} & q_{\mathrm{LRAT}}, \end{bmatrix}^T \tag{6.8}$$

where $v_{\mathrm{L}m}$ is the valve setting for the valve $m \in \{1-4\}$ and q_{LRAT} is the well total production rate. The CVs available are pressures and flow rates in the well, and in each lateral respectively.

The NMPC is adjusted to include the MVs by using the `WSEGLABY` keyword to set the valve openings. The production rate is controlled through the `WCONPROD` as in Case 1. The valve settings in the controller takes dimensionless values from $(0, 1]$ referring to the degree of opening, where 1 represents fully open. ECLIPSE demands the specified value to give the cross sectional area open for flow, so the value is linearly transformed between the sets

$$(0, 1] \Leftrightarrow (0, A_{max}] \overset{(6.7)}{=} (0, 0.1363],$$

when exchanging valve settings from SEPTIC to ECLIPSE. A valve setting equal to zero is prevented using a value close to zero at the minimum value for ECLIPSE to receive.

The step length is chosen to 30 days, as an estimate of how often it is realistic to change control settings in a real reservoir. The prediction horizon as well as the control horizon is fixed at 900 days. Though some of these models would possibly have a life time beyond 900 with a receding horizon controller, this was done to maintain the possibility for comparison with SPE 79031, where only the first 900 days of production was optimized. This application then becomes a *batch optimization* NMPC, rather then a "standard" NMPC. The NMPC framework is therefore modified to not call ECLIPSE for simulations when running a time step exceeding 900 days. To ensure zero sensitivity to the NMPC all outputs are after 900 days are equal to the ECLIPSE outputs at 900 days. The step length of 30 days is shorter than the 180 days used in SPE 79031. This increase in sampling frequency will cause a much larger simulation time as the number of ECLIPSE simulations needed is heavily increased. Though, it can be argued that a sample time shorter than 180 days is fairly reasonable.

The control problem is formulated to maximize the value q_{FOPT} after 900 days while satisfying the constraints on GOR and WCT, similar to the formulation in Case 1.

Results

7

Having presented the control method, the reservoir production problem and the case models at hand, this chapter will present the simulation results from the implementations and the algorithmic performance of the SSMQN. Following the structure from the previous chapter, each case model is addressed in an orderly fashion, starting of with Case 1. The chapter will show the SSMQN method to give production increases of variable degree for all models, and even more importantly will it show the quality of the SSMQN method compared to previously published results.

7.1 Case 1

The Shoe Box model was simulated for 6 year period, once with the base case conditions and once using optimal control with the SSMQN controller. Fig. 7.1 show the total oil production of the reservoir (OPR) for the two cases. One can observe an increase for the optimal control case, although only a marginal one – approximately 1.2%. Before making appropriate comments on this result, a few more figures are presented to analyse the efforts put in to achieve this gain.

Fig. 7.2(a) and (b) This shows the base case injection strategy versus the SSMQN strategy. The main difference between the two is how the optimal strategy uses more injection from INJ2 than INJ1 for the first 400 hundred days, before shifting other way around. The reason why both the SSMQN case and the base case decreases the total injection rate after 600 days is because the BHPs hits the 400 bar constraint for both the injecting wells. From there on out, both cases lies very close to this constraint, injecting at the maximum available rate. The figures also indicate a larger water injection total for the SSMQN than for the base case. No cost on injected water was included in the optimization, so the SSMQN uses the available injection rate as it sees fit. Such a cost could possibly and relatively easily have been included.

The production rate strategies for the two cases are given in Fig. 7.3(a) and (b). These are also very similar for the two cases. One exception is clearly visible in Fig. 7.3(a). From 600 days and out the SSMQN rate is much higher than for the base case. This is due to the SSMQN assigning the available

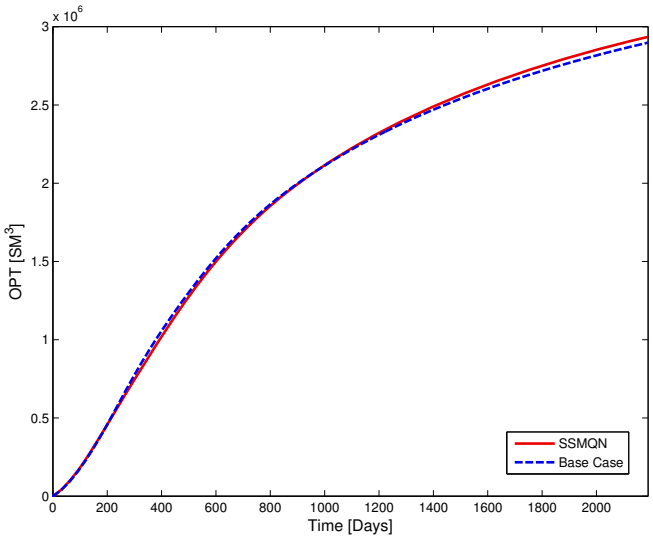
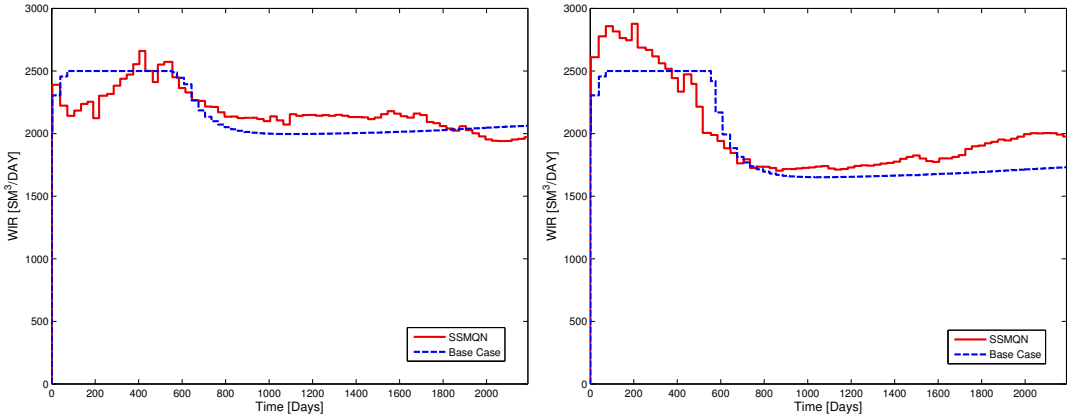


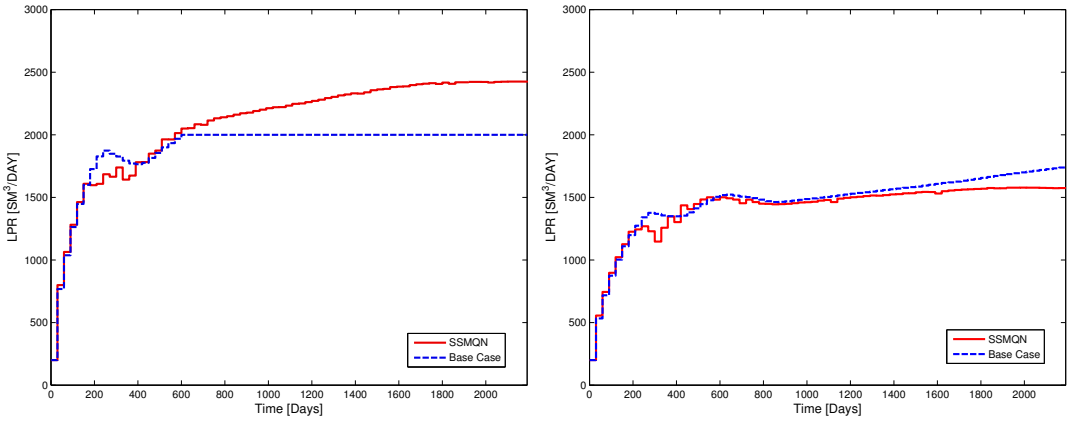
Figure 7.1: Case 1 – Oil production total

production rate, left unused by PROD2 because of an active THP constraint, to PROD1. The base case does not recognize this opportunity, and settles at the pre-selected production. The magnitude of the difference is also caused by the injection strategy building up more pressure at the reservoir side containing INJ1 and PROD1 by injecting more water into this area, allowing a larger rate to be produced in PROD1. Recalling the models flow pattern illustrated in Fig. 6.1(b), this behaviour is can be further understood. Since INJ2 has better connection to both producers from the pattern than INJ1, it is logical to use more force through INJ1 to increase recovery.



(a) INJ1 (b) INJ2

Figure 7.2: Case 1 – Water injection rates

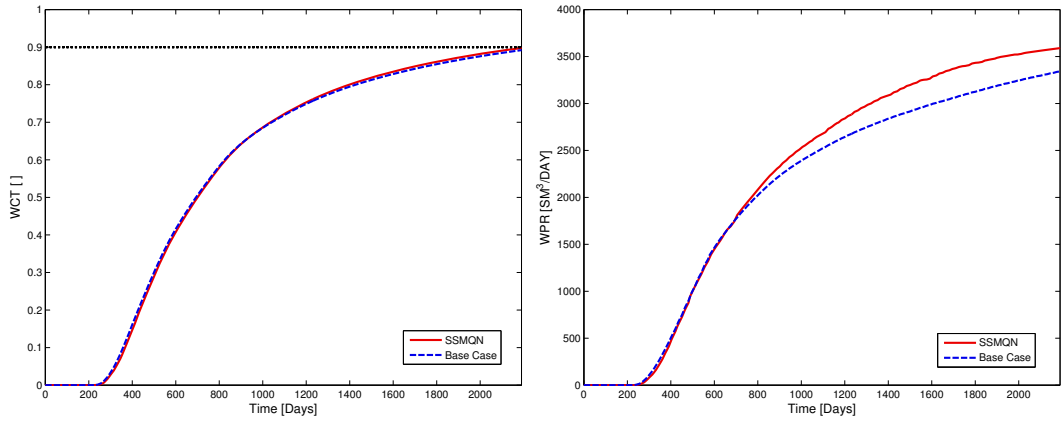


(a) PROD1 (b) PROD2

Figure 7.3: Case 1 – Liquid production rates

The optimal strategy increase of total injected water has already been commented. Fig. 7.4 show the influences of water at the producers. Fig. 7.4(a) shows a plot of the WCT versus time for the two cases. The behaviours are very similar in the two cases, but a short delay in water breakthrough is noticeable in favour of the optimal case. This is in accordance with the problem review in Chapter 4. Fig. 7.4(b) shows the water production rates (WPR) for the two cases. The SSMQN produces much more water than the base case. This is a result of the increased production shown in Fig. 7.3. Keeping the similar WCTs in mind, it is fairly obvious that a higher production rate will give higher water rate for equal WCTs. So the increase in water production is the cost of the increase in oil production. Again, there was not formulated any explicit cost on produced water in SEPTIC, and therefore it is not taken into consideration when computing the optimal strategy.

The SSMQN managed to show a minor increase in the OPT for Case 1. An increase in water injection totals and production was also shown, however, not more severe than staying inside operational constraints. There are several reasons that can explain for the low efficiency increase for this case model. The most obvious reason is the model simplicity. The model has been modified to inherit some additional complexity by inserting multiple injectors with different perforation profiles. Even so, the model produces fairly good results with the naive, equal distribution base case strategy. Another reason may be that the defined operational conditions are to constraining for the optimal case to perform at its full potential. Because of lack of pressure support in the reservoir, both the base case and the optimal case operate very close to the minimum THP level for the producers and maximum BHP for the injectors. Potentially these issues could have been investigated further, e.g. slacking the operational



(a) Production water cut (b) Water production rate

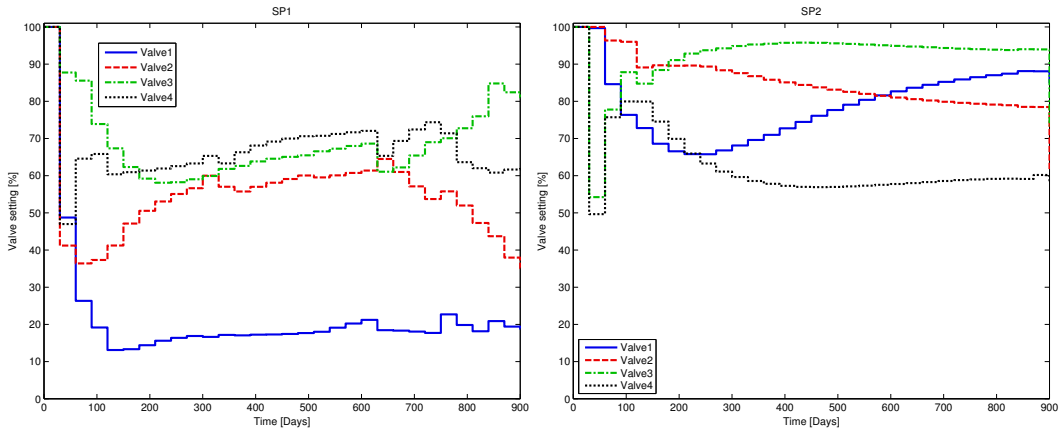
Figure 7.4: Case 1 – Water production

conditions, rearrange placement of well or changing the whole model structure. This was found not to be of enough interest, when considering the limited model potential. Instead, this report moves on to the second case example.

7.2 Case 2

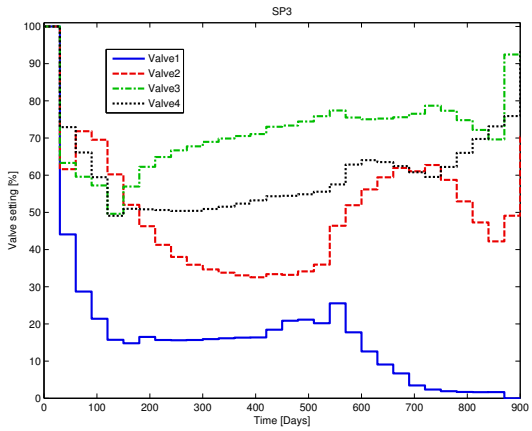
The five models representing Case 2 was each simulated with control for a total 900 days. All simulations were performed with no modelling errors, i.e. the same model was used as both model and process plant. Fig. 7.5(a)–(f) shows the optimal controls found from the SSMQN method, for all models. SP1 denotes one model realization, SP2 a second realization, SP3 a third and so forth.

Fig. 7.5(a)–(e) shows the optimal valve settings over the time horizon for each of the five cases respectively. Even though the different models all have fairly different optimal valve settings, there are some common characteristics in the plots. In all cases except for SP2 the valve at the branch closes to the well heel (Valve1) is choked down the most. In three of the cases the second most chocked down valve is Valve2, at the second closest branch. This is explained by the fact that the pressure in the reservoir causes much more flow in the branches closest to the heel then the outer ones when the valves are fully opened. The difference in flow gives a more rapid breakthrough of water and gas and the closest branches, deteriorating the production performance. The solution is to distribute more equal rates on all the branches. Fig. 7.6 shows that this is exactly the solution found by the SSMQN algorithm for SP1. The optimized case, Fig. 7.6(b), produces at more equal branch rates than the base case in Fig. 7.6(a), and therefore manages to be productive for a longer time. The

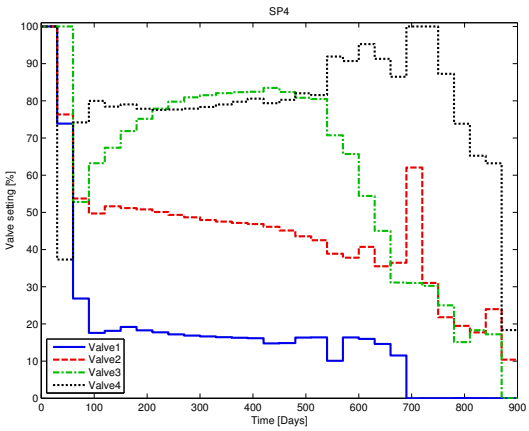


(a) Model SP1

(b) Model SP2



(c) Model SP3

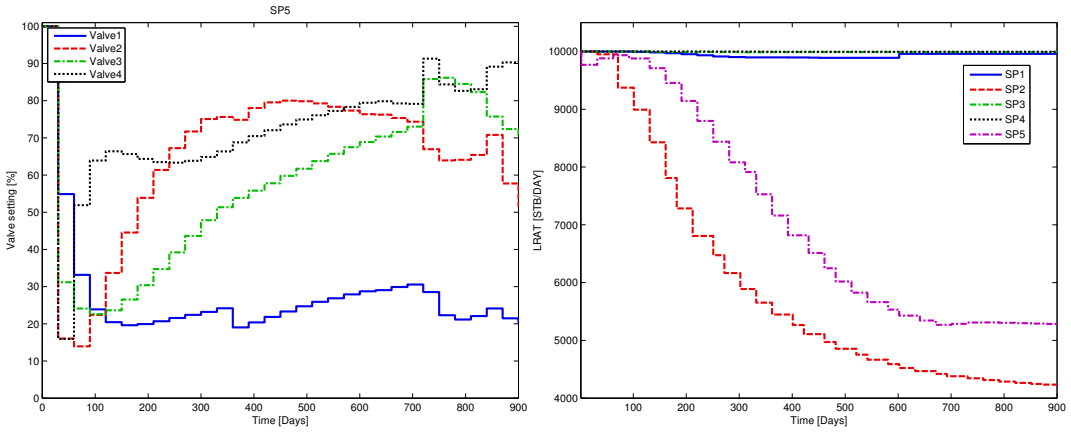


(d) Model SP4

Figure 7.5: Case 2 – Optimal controls from SSMQN algorithm

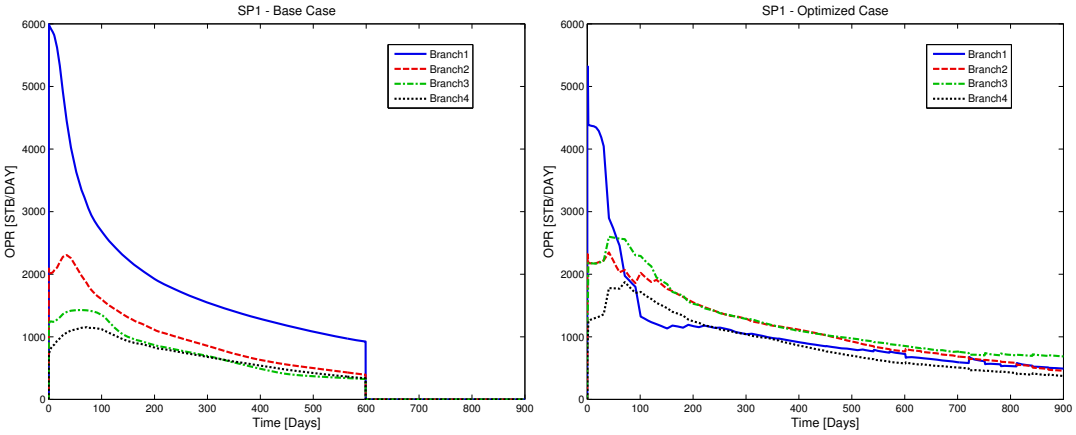
other model simulations show the same tendencies.

Fig. 7.5(f) shows the optimal LRATs versus time for the models. Recalling the base case specified production rate at 10000 STB/DAY, the figure shows that for models SP1, SP3 and SP4 this is close to the computed optimal LRATs. However, in models SP2 and SP5 the controller uses the MV to a great extent. A natural question arises on how such severe decrease in liquid production is optimal to maximize oil recovery. Fig. 7.7(a)–(e) showing the well GOR production rates gives the answer. Paying extra attention to the GOR for SP2 and SP5, in Fig. 7.7(b) and Fig. 7.7(e) respectively, these figures shows how the LRAT decrease is used to prevent the GOR to hit the constraint at 5 MSCF/STB. The SSMQN case performs much better, with the lower GOR, than the base case 10% cut-back strategy. The three other cases also reduce the GOR to a substantial degree, using only the choke settings. This indicates that in cases



(e) Model SP5 (f)

Figure 7.5: Case 2 – Optimal controls from SSMQN algorithm



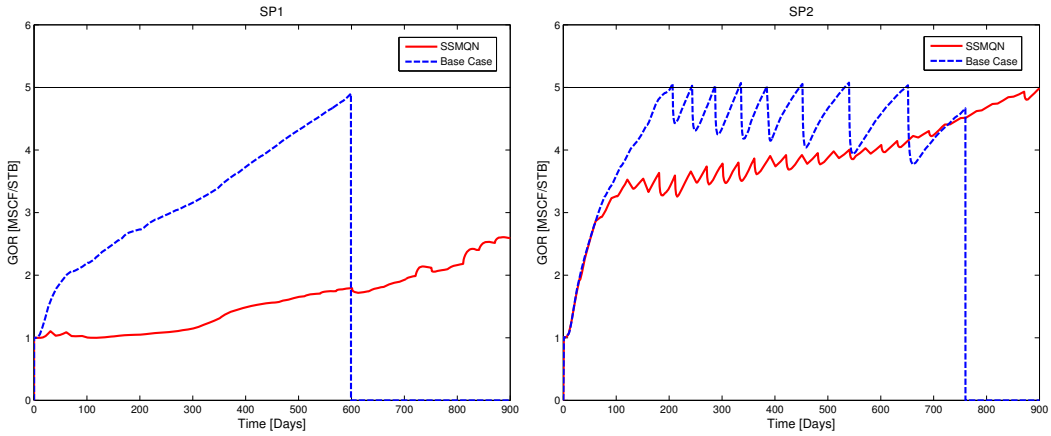
(a) Base Case (b) Optimized Case

Figure 7.6: Case 2 – SP1 Lateral oil production rates

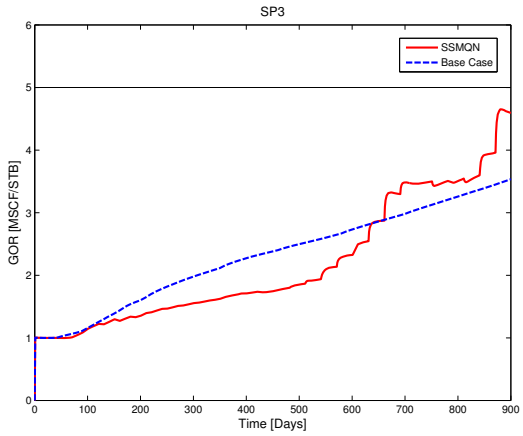
where the presence of gas is not so extreme, valve control can be sufficient, but in the extreme case LRAT control will add an important contribution to increase production.

The gas production constraint is important to handle efficiently, since minimizing the GOR will yield a higher OPR. Also, to high GOR can cause well instability. Equally important in this case is that the WCT constraint is satisfied to the longest extent possible, because of the specification to close down production when the constraint is reached. Fig. 7.8(a)–(e) show time plot of the WCT behaviour for the different models.

It is clear to see that the optimized case outperforms the base case on WCT



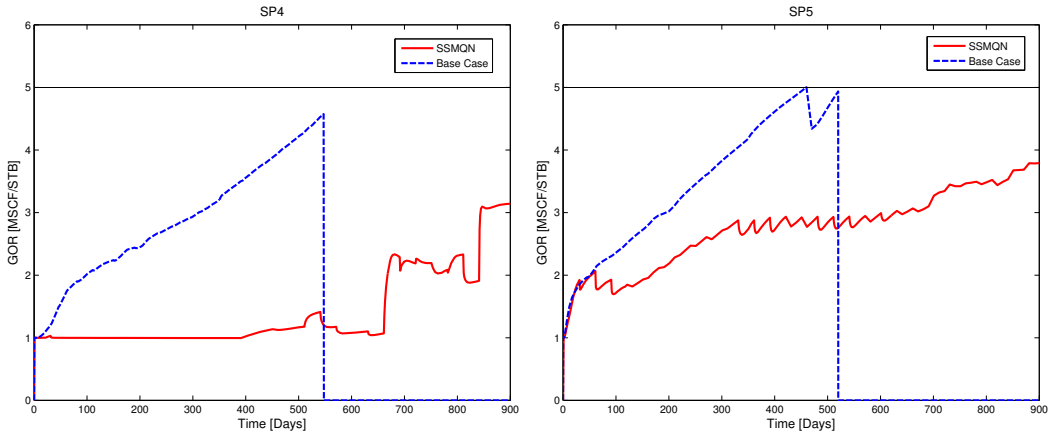
(a) Model SP1 (b) Model SP2



(c) Model SP3

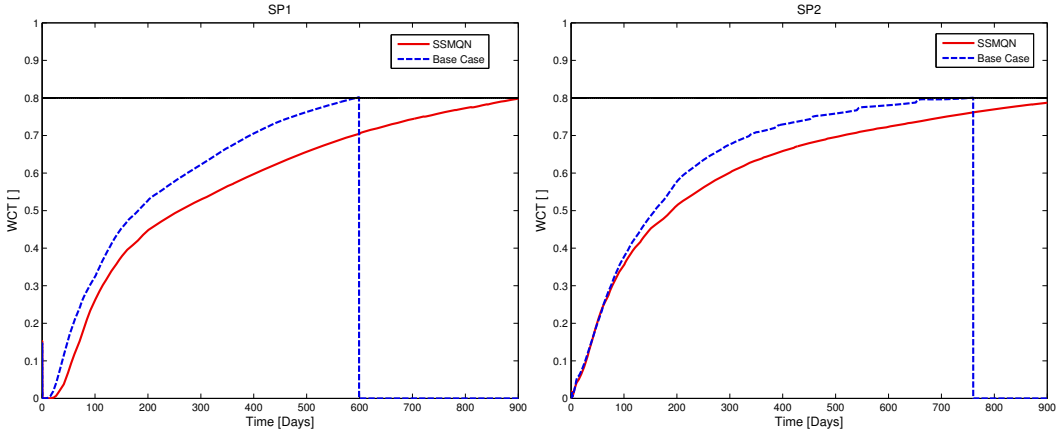
Figure 7.7: Case 2 – Gas production comparison

from these figures. The water breakthrough is either delayed or the time of WCT reaching the constraint of 0.8 is postponed, or as in the case for model SP4 (Fig. 7.8(d)) – both. Also, we recognize how the WCT meets the constraint at approximately 900 days for SP1 (Fig. 7.8(a)), SP2 (Fig. 7.8(b)) and SP5 (Fig. 7.8(e)). This is a direct result of the fixed prediction horizon, as the NMPC will not see any profit in holding the WCT back any longer beyond the horizon end. These cases could therefore possibly gain from a using standard receding horizon NMPC instead. The last thing to notice is how the WCT constraint is not active after 900 days of simulation in case of SP3 and SP4, Fig. 7.8(c) and Fig. 7.8(d). This means these cases would increase their production totals if run beyond 900 days. Alternatively, they could have benefited from LRATs exceeding 10000 STB/DAY. However, this was not investigated, as it would impose operational conditions making the results not comparable to the results



(d) Model SP4 (e) Model SP5

Figure 7.7: Case 2 – Gas production comparison



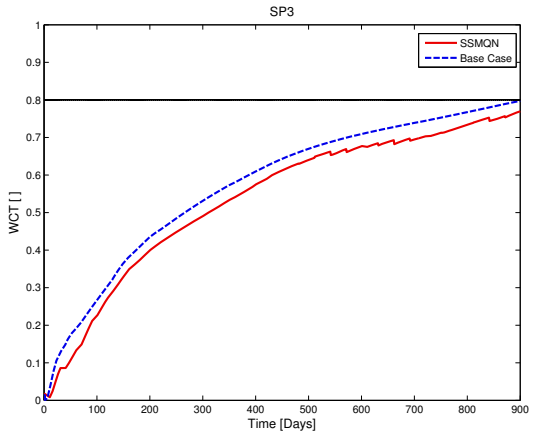
(a) Model SP1 (b) Model SP2

Figure 7.8: Case 2 – Production water cut comparison

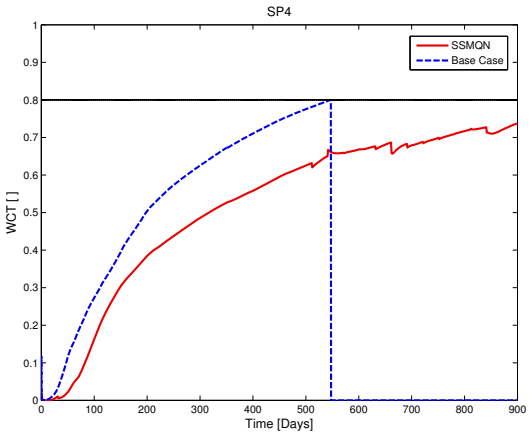
published in SPE 79031.

Before making that final comparison, the time plots of the OPRs in the optimized case versus the base case are presented. These are displayed in Fig. 7.9(a)–(e), showing that the optimized case exceeds the production for the base case in all model realizations. What is equally noticeable, is the large differences in the production increase for the different model realizations, with the two extremes of marginal improvement for SP2 in Fig. 7.9(b) and the huge improvements for SP1 and SP4 in Fig. 7.9(a) and Fig. 7.9(d).

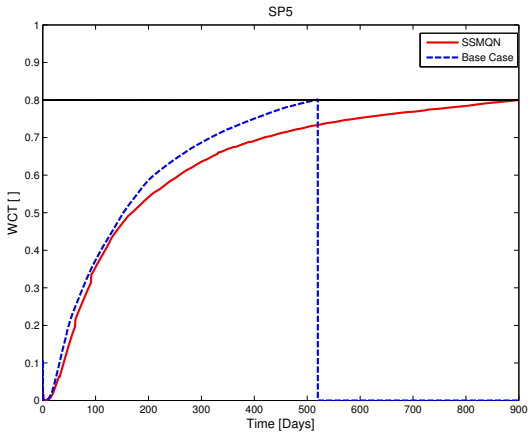
The optimization problem was formulated to maximize the oil production total



(c) Model SP3



(d) Model SP4

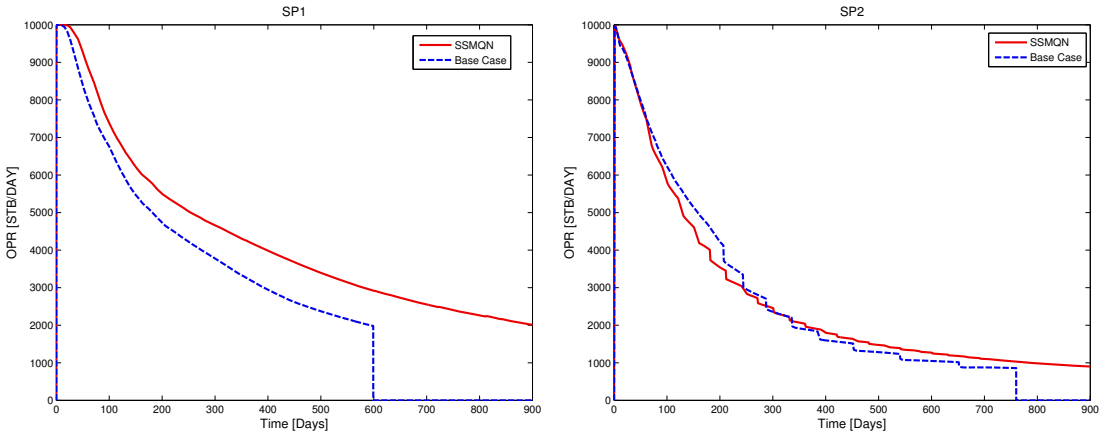


(e) Model SP5

Figure 7.8: Case 2 – Production water cut comparison

(OPT) for the models. As Fig. 7.9(a)–(e) has shown, the NMPC application gives increase in all models, varying from substantial to minor increases. What is left is to quantify these production gains. The numbers are given in Table 7.2. The table lists the production for every model realization on separate lines. The columns consist of the total oil production after in 900 days for three cases, the base case, the optimized case and also the results found in SPE 79031. The gain columns shows production increase found for the two optimization methods relative to the base case. The results from the SSMQN show large improvements over the base case production. The relative gain spans from 5.4% to 67.6%, with an average of 29.9%.

A similar table was given in SPE 79031, without the SSMQN results, where Yeten et al. (2002) commented on the large variations in production gains for



(a) Model SP1 (b) Model SP2

Figure 7.9: Case 2 – Oil production rate comparison

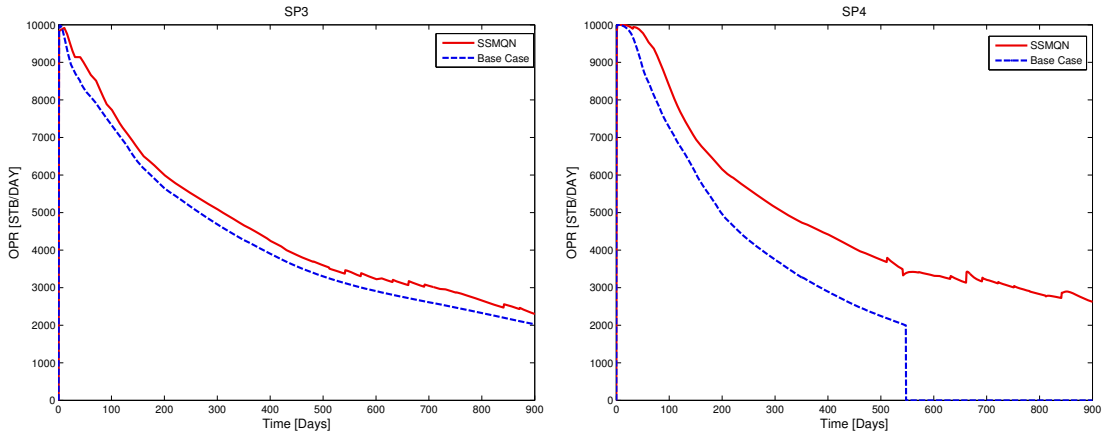
the five cases. Although able to show significant increase in some cases, SP2 only produced an extra 1.8% when using optimization and instrumentation. On this basis Yeten et al. (2002) suggested that with the geological uncertainty represented by the different models, optimizations did not give a consistent answer on whether or not instrumentation is economically justifiable. In some models, the gain found was fairly small, on the other hand, they stated that "significant resources might be lost by not deploying the control devices".

Model realization	Base case (MMSTB)	SPE 79031 (MMSTB)	Gain %	SSMQN (MMSTB)	Gain %
SP1	2.61	3.83	46.7	3.86	47.9
SP2	2.22	2.26	1.8	2.34	5.4
SP3	3.80	4.13	8.7	4.14	8.9
SP4	2.59	4.27	64.9	4.34	67.6
SP5	2.18	2.48	13.8	2.61	19.7
Average	2.68	3.40	27.2	3.46	29.9
Std. Dev.	0.66	0.95	27.2	0.92	26.9

Table 7.1: Comparison of Oil Production Totals

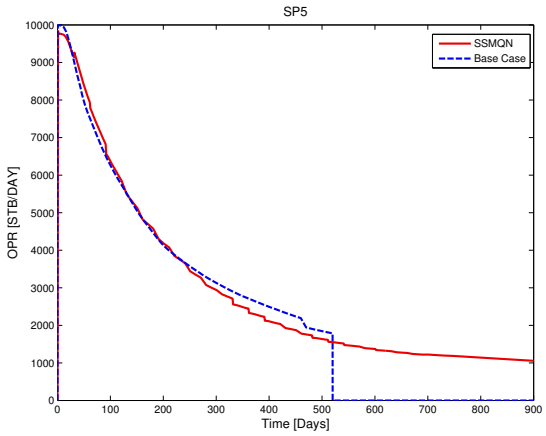
Comparing the results from the SSMQN to the results from SPE 79031, the former comes out on top for all the model realizations. On average, the SSMQN gives an additional 3% increase, relative to the base case, over the SPE 79031. The three main explanations behind this are

1. **Decreased sample time** – the NMPC used time steps of 30 days, whereas the SPE 79031 optimized with 180 day intervals.



(c) Model SP3

(d) Model SP4



(e) Model SP5

Figure 7.9: Case 2 – Oil production rate comparison

2. **Dynamic optimization** – in SPE 79031 the optimization problem parameters was kept to a minimum as each step found the optimal set of static inputs for the rest of the reservoir lifetime. The SSMQN opened for computation of dynamically changing inputs by increasing the number optimization parameters. Where SPE 79031 solved for one parameter per input, the SSMQN solved for 4, giving a total of 20.

3. **Optimizing LRAT** – the SSMQN included the LRAT as an optimization parameter. This way the constraint handling could be taken care of by the NMPC method, instead of by the heuristic method offered by ECLIPSE. Since the models are carrying information about the constrained rates, the GORs and WCTs, the natural thing is to include all the available MVs for controlling these in the optimization problem.

The listed reasons imply a higher computational cost for the SSMQN than for the SPE 79031. A larger number of optimization problems are solved, the problems are harder to solve and the application requires more ECLIPSE simulations. Still the SSMQN performs well within real-time demands, as the 900 days optimal control simulation took about 3 days each on a single computer. Also, the results suggest that the extra computations are compensated for, as they give additional increase in all cases. An interesting thing is to notice the SSMQN performance compared to the SPE79031 for SP2 and SP5. These where the to cases where the LRAT MV was used the most, recalling Fig. 7.5(f). The table shows that these are the two cases providing the most increase over SPE 79031, indicating the NMPC being most valuable for models with the toughest operational conditions. Evident improvements are also shown for SP2 and SP4, representing the most profitable cases in both this report and in SPE 79031. The least increase over SPE 79031 are found for SP3. This is the model with the highest base case OPT. Fig. 7.8(c) shows the WCT for this case. Notice how the WCT constraint is not active after 900 days in the optimized case. Hence, this model would have shown a better result if run for a longer time period as its natural lifetime is longer than 900 days for a maximum LRAT of 10000 STB/DAY. This would potentially have given better performance compared to the SPE 79031 as well.

To summarize, the results from the SSMQN simulations confirms and further strengthens the results from SPE 79031 – valuable resources will potentially be lost if a reservoir like this is not equipped with smart wells. In addition to increasing the most promising results in SPE 79031, the SSMQN increases the "worst case" scenario as well, by recovering 5.4% more oil in SP2, compared with SPE 79031s 1.8%. Proper decision making methods should still be considered before investing the extra costs of smart well optimization for a reservoir. However, these results strengthen the argument of applying reservoir optimization for complicated reservoir structures.

Discussion

8

This chapter gives a brief discussion of the main work and results presented in this thesis, before proposing some thoughts and propositions on further work and ending the report with the final conclusions.

In Chapter 3 the details of a NMPC algorithm, the SSMQN, was presented. Basing on previous work using a similar approach, the SSMQN was designed to control severely nonlinear processes by using a black-box simulator model for predictions. Applying the method to a benchmark CSTR model for testing showed satisfying algorithmic performance, for different variations of algorithm parameters. Depending on the control application to be implemented, the parameters should be chosen wisely to yield optimal performance. The section makes a special comment on giving the convergence parameters extra thought if the time between samples for calculations is limited. The method is developed considering only stable system, and is as such limited to perform satisfactory only for stable systems. To be implemented on a unstable process additional measures must be taken to guarantee stability, e.g. pre-stabilize the unstable process or impose terminal constraints or regions. Care must also be taken to avoid the unstable simulator model returning infinite predictions.

Chapter 4 and 5 gives a review of the reservoir problem and reservoir simulation basics. The ECLIPSE reservoir simulator is described along with a few properties. As ECLIPSE is a huge simulation tool, the author has only gathered limited knowledge to the features available. The features used are only mentioned or described on a higher level. Reservations are taken considering the possibility that ECLIPSE offers features that could have been used to simplify any part of the presented work.

Case model descriptions was given in Chapter 6 and the simulation results was presented in Chapter 7. Again it must be stated that all simulations was performed without any modelling error. Considering the highly different behaviour of the five model realizations in Case 2, it would have been an interesting exercise to investigate optimal production potential using erroneous models for predictions. This was however out of scope for this report. Since all results are gained from perfect reservoir model knowledge they can only be seen as best case scenarios of a real-life reservoir application.

8.1 Considerations for further work

There are several issues which needs to be address to continue the progression on this topic. Before drawing the final conclusions, a few will be listed here.

- **Model errors** – The effects of model errors need to be investigated. This means both NMPCs ability to prevent violation of operational constraints through feedback, and also the effects of applying an "optimal production strategy" to a reservoir calculated from an erroneous model. The question is whether or not the model error will cause a loss of production compared to the base case. The five model realizations in Case 2 would make nice candidates for such a task. E.g. show how much of SP4s production potential will be lost if SP2 is used for predictions, or how the GOR would behave for the reversed case.
- **Model updating** – When model errors are present there is an obvious need for model updating. As production data becomes available one would wish to use the deviation from the predicted values to update the model to make better predictions. The EnKF is a strong candidate for this purpose, but other methods are also available.
- **Model reduction** – To be able to run faster simulations for predictions, model reduction methods should be further investigated. E.g. create ECLIPSE models with fewer grid cells and/or variable grid sizes, so called proxy models. Full scale reservoir models may still take hours to run, making them inefficient to use in a NMPC application.
- **Parallel processing** – Another possibility is to speed up calculations by assigning all prediction simulations to different CPUs for parallel processing. This was not used in this report, as all simulations were run sequentially. Considering every optimization taking about 3 days, parallel processing could have easily decreased this time with at least 90%.

Also, a paper presenting the results made on the benchmark reservoir Case 2 is under preparation. An abstract is to be submitted to the SPE Intelligent Energy Conference and Exhibition 2008. The paper abstract and introduction can be found as an Appendix on page 89. The full paper will also be submitted to the SPE Journal when finished.

Conclusions

9

In this report the author has developed and implemented a NMPC application in Statoil's MPC tool SEPTIC, using a single shooting, multistep, quasi-Newton approach. The application has been successfully tested against different nonlinear models, showing satisfactory control performance. The NMPC was interfaced to ECLIPSE, a commercial reservoir simulator, to use for simulations and predictions to solve the optimal reservoir control problem. The simulation results show that reservoir productions can be increased by considerable amounts by using a NMPC controller compared to reference cases produced using simple constant rate strategies. For one case in example Case 2, an increase with up to 68% was found.

The results found also confirm that not all reservoir models justify the extra cost of smart completions. Example Case 1 showed very little improvement when control was applied. As a general conclusion it can be stated that a reservoir should either include some structural model complexity or an unbalanced production profile across the wells to yield a sufficiently potential gain from optimal control.

The developed algorithm was applied to a benchmark reservoir model to validate the quality of the NMPC results. The application managed to show an increase in production over the previously published results on the model. Hence, the quality of the method was proven.

References

Allgöwer, F. and Zheng, A. (2000), (Eds.). *Nonlinear model predictive control, progress in systems and control theory*, Vol. **26**. Birkhauser Verlag.

Allgöwer, F., Findeisen, R. and Nagy, Z. K. (2004). "Nonlinear model predicticve control: From theory to application", *J. Chin. Inst. Chem. Engrs.*, Vol. **35**, no. 3, pp. 299–315.

Asheim, H. (1988). "Maximization of water sweep efficiency by controlling production and injection rates" in paper **SPE 18365** presented at the *SPE European Petroleum Conference*, London, UK, October 16-19.

Aziz, K. and Settari, A. (1986). *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers.

Barclay, A., Gill, P. and Rosen, J. (1997), "SQP methods and their application to numerical optimal control". `citeseer.ist.psu.edu/barclay97sqp.html`. Numerical Analysis Report 97–3, Department of Mathematics, University of California, San Diego, CA.

Biegler, L. T. (1998). "Advances in nonlinear programming concepts for process control", *Journal of Process Control*, Vol. **8**, no. 5/6, pp. 301–311.

Biegler, L. T. (2000). In *Nonlinear Model Predictive Control, F. Allgöwer & A. Zheng (Eds).*, Chapter "Efficient Solution of Dynamic Optimization and NMPC Problems", pages 219–244. Birkhäuser. ISBN 3-7643-6297-9.

Bieker, H., Slupphaug, O. and Johansen, T. A. (2006). "Real time production of offshore oil and gas production systems: A technology survey" in paper **SPE 99446** presented at the *SPE Intelligent Energy Conference and Exhibition*, Amsterdam, the Netherlands, April 11-13.

Brouwer, D. R. and Jansen, J.-D. (2004). "Dynamic optimization of waterflooding with smart wells using optimal control theory", *SPE Journal*, Vol. **9**, no. 4, pp. 391–402.

Brouwer, D. R., Jansen, J.-D., van der Starre, S., van Kruijsdijk, C. P. J. W. and Berentsen, C. W. J. (2001). "Recovery increase through waterflooding using smart well technology" in paper **SPE 68979** presented at the *SPE European Formation Damage Conference*, The Hague, the Netherlands, May 21-22.

Brouwer, D. R., Nævdal, G., Jansen, J.-D., Vefring, E. H. and van Kruijsdijk, C. P. J. W. (2004). "Improved reservoir management through optimal control and continuous model updating" in paper **SPE 90149** presented at the *SPE*

Annual Technical Conference and Exhibition, Houston, Texas, September 26-29.

Chen, H. and Allgöwer, F. (1998a). "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability", *Automatica*, Vol. **34**, no. 10, pp. 1205–1217.

Chen, H. and Allgöwer, F. (1998b). "A computationally attractive nonlinear predictive control scheme with guaranteed stability for stable systems", *Journal of Process Control*, Vol. **8**, no. 5/6, pp. 475–485.

Cutler, C. R. and Ramaker, B. L. (1979). "Dynamic matrix control - a computer control algorithm". In Proc. *AIChE 86th National Meeting*, Houston, TX.

Cutler, C. R. and Ramaker, B. L. (1980). "Dynamic matrix control - a computer control algorithm". In Proc. *The Joint Automatic Control Conference*, San Francisco, CA.

Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z. and Allgöwer, F. (2002). "Real-time optimization and nonlinear model predictive control of processes governed by differntial-algebraic equations", *Journal of Process Control*, Vol. **12**, pp. 577–585.

Dolle, N., Brouwer, D. R. and Jansen, J.-D. (2002). "Dynamic optimization of waterflooding with multiple injectors and producers using optimal control theory". In Proc. *XIV International Conference on Computational Methods in Water Resources*, Delft, June 23-28.

García, C. E. and Morshedi, A. M. (1986). "Quadratic programming solution of dynamic matrix control (QDMC)", *Chemical Engineering Communucations*, Vol. **46**, pp. 73–87.

Goldfarb, D. and Idnani, A. (1983). "A numerically stable dual method for solving strictly convex quadratic programs", *Mathematical Programming*, Vol. **27**, pp. 1–33.

Heijn, T., Markovinović, R. and Jansen, J.-D. (2004). "Generation of low-order reservoir models using system-theoretical concepts", *SPE Journal*, Vol. **9**, no. 2, pp. 202 – 218. ISSN 1086-055X.

Jensen, J. P. (2007). "Ensemble Kalman Filter for state and parameter estimation on reservoir model". *Master's thesis*, Norwegian University of Science and Technology, Trondheim, Norway. Submitted June, 2007.

Kalman, R. E. (1960a). "Contributions to the theory of optimal control", *Bulletin de la Societe Mathematique de Mexicana*, Vol. **5**, pp. 102–119.

Kalman, R. E. (1960b). "A new approach to linear filtering and prediction problems", *Transactions of ASME, Journal of Basic Engineering*, Vol. **87**, pp. 35–45.

Kleinman, B. L. (1970). "An easy way to stabilize a linear constant system", *IEEE Transactions on Automatic Control*, Vol. **15**(6), pp. 692.

Kwon, W. H. and Pearsons, A. E. (1977). "A modified quadratic cost problem and feedback stabilization of a linear system", *IEEE Transactions on Automatic Control*, Vol. **22**(5), pp. 838–842.

Kwon, W. H., Bruckstein, A. M. and Kaliath, T. (1983). "Stabilizing state-feedback design via the moving horizon method", *International Journal of Control*, Vol. **37**(3), pp. 631–643.

Lee, E. B. and Markus, L. (1967). *Foundations of optimal control theory*. Wiley.

Li, W. C. and Biegler, L. T. (1989). "Multistep, Newton-type control strategies for constrained, nonlinear processes", *Chem. Eng. Res. Des.*, Vol. **67**, pp. 526 – 577.

Li, W. C., Biegler, L. T., Economou, C. G. and Morari, M. (1990). "A constrained pseudo-Newton control strategy for nonlinear systems", *Computers and Chemical Engingeering*, Vol. **14**, no. 4/5, pp. 451 – 468.

Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice-Hall.

Martinsen, F., Biegler, L. T. and Foss, B. A. (2004). "A new optimization algorithm with application to nonlinear MPC", *Journal of Process Control*, Vol. **14**, no. 8, pp. 853–865.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scokaert, P. O. M. (2000). "Constrained model predictive control: Stability and optimality", *Automatica*, Vol. **36**(6), pp. 789 – 814.

Meadows, E. S. and Rawlings, J. B. (1997). In *Nonlinear process control, M. A. Henson & D. E. Seborg (Eds).*, Chapter 5: "Model predictive control". Prentice Hall PTR. ISBN 0-13-625179-X.

Meum, P. (2006). "Optimal reservoir management through model predictive control". *Project Assignment Report*, Norwegian University of Science and Technology, Trondheim, Norway.

Morari, M. and Lee, J. H. (1999). "Model predictive control: Past, present and future", *Computers and Chemical Engingeering*, Vol. **23**, pp. 667–682.

Nævdal, G., Johnsen, L. M., Aanonsen, S. I. and Vefring, E. H. (2005). "Reservoir monitoring and continuous model updating using ensemble Kalman filter", *SPE Journal*, Vol. **10**, no. 1, pp. 66–74.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer-Verlag New York. ISBN 0-387-98793-2.

Oliveira, N. M. C. and Biegler, L. T. (1995). "An extention of Newton-type algorithms for nonlinear process control", *Automatica*, Vol. **31**, no. 2, pp. 281–286. ISSN 0005-1098.

Oliveira, N. M. C. and Biegler, L. T. (1994). "Constraint handling and stability properties of model-predictive control", *American Institute of Chemical Engineers Journal*, Vol. **40**, no. 7, pp. 1138–1155.

Peaceman, D. W. (1977). *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Publishing Company.

Pearson, R. K. (1997). In *Nonlinear process control, M. A. Henson & D. E. Seborg (Eds).*, Chapter 2: "Nonlinear Process Identification". Prentice Hall PTR. ISBN 0-13-625179-X.

Prett, D. M. and Gillette, R. D. (1980). "Optimization and constrained multivariable control of a catalytic cracking unit". In Proc. *The Joint Automatic Control Conference*, San Francisco, CA.

Qin, S. J. and Badgwell, T. A. (2003). "A survey of industrial model predictive control technology", *Control Engineering Practice*, Vol. **11**, no. 7, pp. 733–764.

Rawlings, J. B. (2000). "Tutorial overview of model predictive control", *IEEE Control Systems Magazine*, Vol. **20**, no. 3, pp. 38–52.

Richalet, J., Rault, A., Testud, J. and Papon, J. (1976). "Algorithmic control of industrial processes". In Proc. *The 4th IFAC symposium on identification and system parameter estimation*, pages 1119–1167.

Saputelli, L., Nikolaou, M. and Economides, M. J. (2006). "Real-time reservoir management: A multiscale adaptive optimization and control approach", *Computational Geosciences*, Vol. **10**, no. 1, pp. 61 – 96. ISSN 1420-0597.

Sarma, P., Chen, W. H., Durlofsky, L. J. and Aziz, K. (2006). "Production optimization with adjoint models under nonlinear control-state path inequality constraints" in paper **SPE 99959** presented at the *SPE Intelligent Energy Conference and Exhibition*, Amsterdam, the Netherlands, April 11-13.

Schittkowski, K. (2005). "QL: A Fortran code for convex quadratic programming - User's guide, Version 2.11". Technical report, Department of Mathematics, University of Bayreuth.

Schlumberger (2006), "ECLIPSE Reference Manual 2006.2".

Silva, D. C. M. and Oliveira, N. M. C. (2002). "Optimization and nonlinear model predictive control of batch polymerization systems", *Computers and Chemical Engineering*, Vol. **26**, pp. 649–658.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons. ISBN 0470011688.

Strand, S. and Sagli, J. R. (2003). "MPC in STATOIL - advantages with in-house technology". In Proc. *ADCHEM 2003*, pages 97–103.

Strand, S. T. (1991). *Dynamic optimization in state-space predictive control schemes*. PhD thesis, The University of Trondheim, Norwegian Institute of Technology.

Su, H. T. and McAvoy, T. J. (1997). In *Nonlinear process control, M. A. Henson & D. E. Seborg (Eds).*, Chapter 7: "Artificial Neural Networks for Nonlinear Process Identification and Control". Prentice Hall PTR. ISBN 0-13-625179-X.

Sudaryanto, B. and Yortsos, Y. C. (2000). "Optimization of fluid front dynamics in porous media using rate control. I. Equal mobility fluids", *Physics of Fluids*, Vol. **12**, no. 7.

Tenny, M. J., Wright, S. J. and Rawlings, J. B. (2004). "Nonlinear model predictive control via feasibility-pertubed sequential quadratic programming", *Computational Optimization and Applications*, Vol. **28**, pp. 87–121.

Thomas, Y. A. (1975). "Linear quadratic optimal estimation and control with receding horizon", *Electronics Letters*, Vol. **11**(1), pp. 19–21.

Virnovsky, G. A. (1991). "Waterflooding strategy design using optimal control theory". In Proc. *the 6th European IOR Symposium*, Stavanger, Norway.

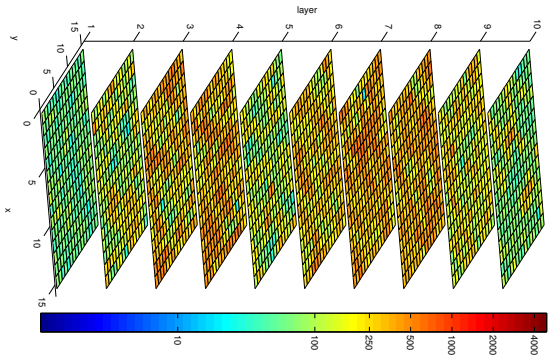
Yeten, B., Durlofsky, L. J. and Aziz, K. (2002). "Optimization of smart well control" in paper **SPE 79031** presented at the *SPE International Thermal Operations and Heavy Oil Symposium and International Horizontal Well Technology Conference*, Calgary, Alberta, Canada, November 4-7.

Yeten, B. (2003). *Optimum Deployment of Nonconventional Wells*. PhD thesis, Stanford University.

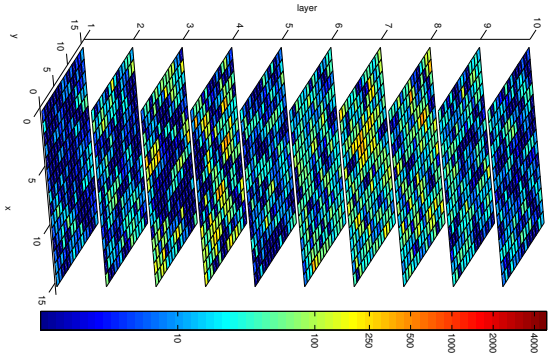
Appendices

Case 1: Model Properties

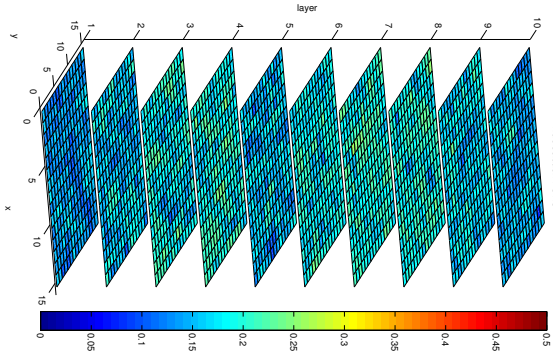
A



(a) Horizontal permeability



(b) Vertical permeability



(c) Porosity

Figure A.1: The Shoe Box reservoir properties

Case 2: Model Properties

B

drainage area		5000×5000 ft^2
oil thickness		50 ft
gas cap thickness		50 ft
ϕ		0.20
gas cap PV		0.625 MMft^3
R_s		1.0 MSCF/STB
c	at p_{bub}	3.0×10^{-6} psi^{-1}
k_{ro}		$\begin{cases} 0.8 \text{ at } S_{wc} = 0.20 \\ 0.8 \text{ at } S_{gr} = 0.05 \end{cases}$
k_{rw}		0.4 at $S_{or} = 0.30$
k_{rg}		0.9 at $S_{wc} = 0.20$
γ	at 14.7 psi	
oil		0.85
water		1.0
gas		0.71
μ, cp	at p_{bub}	
oil		0.42
water		0.30
gas		0.02
B, V/V	at p_{bub}	
oil		1.55
water		1.02
gas		0.71

Table B.1: Case 2 model properties (Yeten et al., 2002)

Facies	Average (md)	Standard Deviation (md)	Coefficient of Variation
Channel Sand	1534	635	0.4
Mudstone	4.9	1.5	0.3

Table B.2: Case 2 permeability statistics (Yeten et al., 2002)

Paper draft

Optimization of Smart Well Production using Nonlinear Model Predictive Control

P. Meum, O. M. Aamo, P. Tøndel and J.-M. Godhavn

Abstract submitted to *SPE Intelligent Energy Conference and Exhibition*
Amsterdam, The Netherlands, February 25–27, 2008

SPE Paper

Optimization of Smart Well Production through Nonlinear Model Predictive Control

Patrick Meum[1], Ole Morten Aamo,[1] Petter Tøndel,[2] John-Morten Godhavn,[2]
[1]Norwegian University of Science and Technology, Trondheim, [2]Statoil ASA T&P

Abstract

In this paper, we present an algorithm for optimizing reservoir production using smart well technology. The term smart well is used to indicate a nonconventional well equipped with downhole inflow control devices (ICVs) and instrumentation. This additional instrumentation extends the degree of freedom in the field production planning, since production can be efficiently distributed on the different well segments available. By proper utilization of the ICVs through optimal production planning, an increase in the oil recovery factor for the reservoir can be expected. We propose a method for optimal closed-loop production known from control theory as *model predictive control* (MPC). A commercial reservoir simulator, ECLIPSE, is used for modelling and predictions. MPC is chosen for its ability to provide an optimal solution for the constrained multivariable control problem. To compute the optimal ICV settings, we propose using a nonlinear MPC (NMPC) application, which can handle the severe nonlinearities found in reservoir models. The NMPC uses a single shooting, multistep, quasi-Newton (SSMQN) method to solve the optimization problem. As the term multistep suggests, this is an iterative method, which solves a sequence of quadratic problems (QPs) in each timestep. We apply our method to a benchmark reservoir model with multiple geostatistical realizations. This model has already proven potential for increased oil recovery by using optimizarion techniques. We show an even additional increase over the former approach in production totals, using the SSMQN method, with as much as 68% increase in one case, and 30% on average compared to the uncontrolled reference case.

1 Introduction

A smart well is a nonconventional well equipped downhole with ICVs. Such wells offer control of the total flow through individual segments and branches, as well as multiphase flow and pressure measurements. The potential benefits from proper use of ICVs in a real-time control application are quite substantial. Especially because continuous redistribution of the production rate on the available branches can delay or avoid coning of gas and/or water for as long as possible.

Reservoir optimization is currently an exiting field of research, and has already been investigated by a number of authors since the turn of the millennium. Although, some early attempts were made by Asheim (1988) and Virnovsky (1991), most acknowledge Sudaryanto and Yortsos (2000) to be the first to systematically address the flooding problem. They used optimal control theory to maximize sweep efficiency for a multiple source (injector), single sink (producer) system. By optimally allocating the injection rate for each injector they showed a “bang-bang” strategy to maximize the displacement efficiency, as this caused a simultaneous breakthrough from both sources at the producing end. Brouwer et al. (2001) made a study from a less theoretical point of view, focusing on production potential available through smart well control. Using a heuristic algorithm for static optimization, the developed a production rate strategy for simple reservoir models. This work was extended by Dolle et al. (2002) which developed an algorithm for dynamic optimization, using gradient-based optimization. In addition to improving the results from the static approach, they also addressed reservoirs with heterogeneous permeability fields.

The above mentioned work was made with an assumption of constant production rates. Brouwer and Jansen (2004) recognized that this was hardly common in pratice, and therefore investigated the problem further, as they compared the constant production rate case with the constant bottom hole pressure (BHP) case. These cases were argued to illustrate the two extremes of well-operating conditions, as practical production planning need to take them both into consideration. Yeten et al. (2002) combined optimization, using a conjugate gradient method, with features available in a commercial reservoir simulator. The simulator is used for numerical gradients, as well as for efficient modelling of ICVs. Though costly in terms of

computation time, their algorithm produced promising results. Sarma et al. (2006) used an approximate feasible direction algorithm, in combination with a general purpose simulator. Exchanging exact gradient information directly with the simulator, and from the efficiency of the approximate feasible direction algorithm, they proved to match results by Yeten (2003) using only a fraction of the CPU time.

In this paper we will describe an NMPC algorithm, and apply it to a set of reservoir models, previously used in Yeten et al. (2002) and Yeten (2003). The models are geostatistical realizations of a fluvial reservoir with a horizontal multilateral well. All models are applied to the optimization routine, and results are compared both to base case numbers, and the results from previously published optimizations. We will show that our NMPC algorithm, by dynamic optimization and reducing the sample time, further increases production potentials over the previous used methods.