NTNU
Norwegian University of
Science and Technology

Boundary Control of the Gas Coning Problem

Jonathan Ronen

Master of Science in Engineering Cybernetics
Submission date: September 2010
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Svein Ivar Sagatun, ITK
Agus Hasan, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Gas coning is a challenge in some oil fields located on the Norwegian Continental Shelf. This phenomenon occurs in reservoirs with a thick gas cap on top of a thin oil rim. Once the gas oil contact (GOC) reaches the well perforation, free gas may dominate the total production rate to the extent that further operation of the well becomes uneconomical and the well must be shut-in. Therefore, there is an incentive to produce such wells in their subcritical phase as long as possible, i.e. before free gas reaches the well perforation. The reason is that gas handling capacity often is a constraint.
Oil rim production frequently applies horizontal well technology. In this case gas breakthrough usually occurs at the well heel.

The figure above presents a generic framework for gas coning control. The control law can be constructed using various formal design methods; Backstepping, Lyapunov design, Formal power series parameterization, etc. Further, the observer may be constructed using the Backstepping method or some nonlinear Kalman filter.

The project includes the following tasks:
1. Perform a literature survey on gas coning in the oil industry.
2. Propose alternative control laws based on formal design methods like Backstepping, Lyapunov design and Formal power series parameterization; both for linear and nonlinear systems. Assess and compare performance and user friendliness.
3. Design observer(s), applying Backstepping and possibly a nonlinear Kalman-filter, for the gas coning problem for alternative sensor and actuator configurations. Study its properties; both formally and through numerical simulations.
4. Study properties and assess performance of a combined observer-control concept.
5. Perform an overall assessment and make a recommendation which emphasizes theoretical properties, implementation issues and user friendliness.

The numerical assessment should be performed with simple simulators like GORM as well as more realistic models.

The candidate should use its contacts in industry in order to quality control the thesis and its disposition. The thesis report may include a paper to a selected conference with the main results of this work.

Assignment given: 12. March 2010
Supervisor: Bjarne Anton Foss, ITK

Boundary control for the gas-coning problem

Jonathan Ronen

September 11, 2010

Abstract

This thesis was set to tackle the gas coning problem in oil-rim reservoirs with horizontal wells. The focus was short term production planning in the sub-critical phase only. Different controllers were developed and assessed based on the objective of net-present-value (NPV) of oil produced in the sub-critical phase.

The reservoir model is a 1-D partial differential equation describing the dynamics of the gas-oil contact (GOC), for a homogeneous reservoir. Gas coning is considered to be the deformation of the GOC towards the well.

Several controllers were developed and assessed alongside control laws from previous research: (i) the Backstepping method was used to develop a state-feedback controller, along with an observer. Coupled they make the Backstepping output-feedback controller. (ii) an output-feedback controller based on the structure proposed by previous research. (iii) linear-quadratic optimal control.

An extended Kalman filter was also considered as a state observer, alongside the Backstepping observer.

The backstepping controller did not deliver an increase of sub-critical payout which warrants the complicated structure of an observer-controller pair. It was even outperformed by simpler, output-feedback control laws. The optimal linear-quadratic controller achieved the best NPV of sub-critical production by far. This makes it the most attractive control strategy presented, even when considering that a real-world implementation will need to be paired with a state observer.

Contents

Nomenclature

ODE Ordinary Differential Equation

PDE Partial Differential Equation

GOR/WOR Gas-Oil Ratio / Water-Oil Ratio

GORM Gas-Oil Ratio Model

GOC/OWC Gas-Oil Contact / Oil-Water Contact

NPV Net Present Value

QP Quadratic Programming

EKF Extended Kalman Filter

Neumann Boundary Condition Boundary conditions of the form $h_x(0) = a$

Dirichlet Boundary Condition Boundary conditions of the form $h(0) = a$

Robin Boundary Condition Mixed boundary condition

$q(t)$ Oil flux into the well (out of the reservoir)

$q_d(t)$ Oil flux in the reservoir at the point $x = d$

k,ϵ Tuning parameters in controllers

I Interest Rate [%]

K Permeability Tensor, Darcy

$h(t, x)$ Height of oil column in reservoir

h_t,h_x,h_{xx} $\frac{\partial h}{\partial t}$,$\frac{\partial h}{\partial x}$,$\frac{\partial^x h}{\partial x^2}$

p Pressure, Pascal

μ_0 Viscosity

φ Porosity

Chapter 1

Introduction

As oil reserves get increasingly harder to exploit, the industry has shifted to more and more advanced technologies in order to produce hard to get oil and gas. These efforts are collectively referred to as *smart* or *intelligent* wells, Integrated Operations, as well as a plethora of other names. This thesis addresses the gas-coning problem, which is only one area where advanced technology can help exploit otherwise difficult to get oil resources.

A Control Engineering approach is taken throughout this thesis. A mathematical model is used to increase the understanding of the problem. Based on that model, observers and controllers are designed to address the problem. A general framework is given in Figure 1.1.

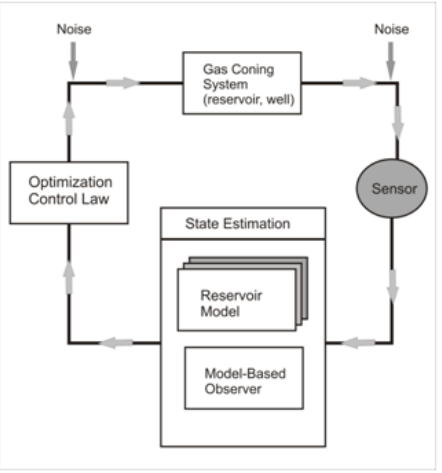


Figure 1.1: Generic framework for gas-coning control.

1.1 Gas Coning

In 1935, Muskat [1] described a phenomenon known as *water coning*. It was observed that, when producing oil from a reservoir situated on top of an aquifer, the pressure-drop due to the oil being removed caused the layer of water to *cone* up towards the well, see Figure 1.2.

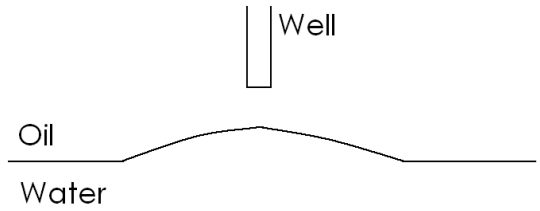


Figure 1.2: A water cone deforms the Oil-Water contact, and draws towards the producing well.

A similar phenomenon known as *gas coning* occurs when there is a gas cap above the oil layer in the reservoir. The gas will then gradually form a cone towards the well. Konieczek [2] describes a *critical production rate* that, if exceeded, results in water/gas breakthrough – the cone breaks and gas/water is produced in addition to oil. This is sometimes referred to as *viscous fingering instability*. The production before breakthrough is known as the *sub-critical* production phase, and after breakthrough the well enters the *super-critical* phase. In the super-critical phase, gas will replace some of the oil being produced, and the Gas-Oil ratio will steadily increase [1],[3]. It is therefore advantageous to produce as much oil as possible in the sub-critical phase.

The critical rate depends, among other factors, on the vertical permeability of the reservoir [1],[3] and the distance between the well and the fluid contact surfaces – the *stand-off* [4],[5]. Muskat [1] describes one experiment where the stand-off was increased, to better than expected results. The results were attributed to the low vertical-permeability zone the well was pulled back to, which provided more resistance to vertical water movement. Since the critical rate depends on the stand-off, which gets smaller with time as the cone grows, the critical rate decreases with time. Other factors that affect the critical rate include porosity, anisotropy, reservoir size, (horizontal) well perforation interval and the density difference between oil and the coning fluid [6].

After gas-breakthrough, it is not possible to simply throttle back the production, as residual gas in the well will cause the pressure to be lower and thus lower the critical rate [7].

It may not be economical to produce below the critical rate, and limitations on water or gas production may make it inadvisable to produce at the super-critical phase. Horizontal wells allow fluid draining over a long distance, thus reducing the pressure-drop and increasing the critical rate [3]. Thus horizontal wells may produce at econom-

ical rates from reservoirs that are otherwise difficult to produce. One such reservoir is the Troll field [8].

Other techniques to prevent water/gas coning include injecting a block of cement below the well, in order to block the water cone [9]. Others have suggested using polymers or gels injected around the well to make a gas-blocking zone [6]. In reservoirs surrounded by both an aquifer and a gas gap, it has been suggested [10] that the well be placed in the position where the gas and water breakthrough would occur simultaneously. Otherwise, if only gas-coning is considered, the well should be placed as far from the Gas-Oil contact as possible. Another related technique suggested in [11] is to sink water below the OWC, thus eliminating water-coning. The total production of water remains unchanged.

A different strategy, described in [12] suggests *reverse-coning* – tapping oil through the water zone as an oil-cone. This would reduce gas-coning as gas would now have to pass through two layers of fluid in order to reach the well. This may work well in Troll, because gas handling capabilities are limited while water is easier to dispose of [8],[13]. This aspect of the problem makes Shirman [11]'s approach of sinking water below the OWC impractical for gas-coning at Troll.

Some research has also gone to controlling the Gas-Oil ratio post-breakthrough. Notably, [8] introduces a Gas-Oil-Ratio-Model to predict the GOR and optimize production. While [8] focuses on short and medium term production from a single well, [13] notes that the wells in the reservoir are interconnected, and the main constraint is the *total* gas handling capacity. The authors present a way to optimize production from multiple wells at the same time, optimally distributing gas production between them.

Another approach to post-breakthrough control of gas fraction is [14]. The authors use a dynamic coupled-well model, which they claim is more accurate than stand-alone well models such as that in [8]. They use PID controllers to control the gas rates, once the optimal GOR has been calculated. An interesting result of that research is that gas-coning was used to *increase* end oil-production. This is because the gas flow helps to lift the oil in wells below 2000 meters.

The focus of this thesis will be on sub-critical production planning only. The original mention of the critical rate [2] suggested producing at below the critical rate and as close to it as technologically feasible. Since it may not be economical to produce below the critical rate as the reservoir is depleted, Sagatun [7] proposes maximizing the *net present value* of the production in the sub-critical phase, and then passing the well management to the other strategies described above, for the super-critical phase. In [7] the model from [8] is used but only in the sub-critical phase. It is the first work to utilize active control in order to maximize NPV of the sub-critical production.

The control strategy of [7] is designed using a linearization of the model from [8], and achieves a substantial increase in NPV of sub-critical payout compared to any constant-rate strategies that have been used traditionally.

In a term project based on [7], I developed other strategies of active control and

was able to improve upon Sagatun's results. These strategies, as well as others, will be described in detail in the coming chapters.

The rest of the report is organized as follows: Chapter 2 presents the well-reservoir model – a 1D pre-breakthrough GORM. Chapter 3 presents active control strategies, and Chapter 4 presents two observer designs. Then, simulation results are presented in Chapter 5 and discussed in Chapter 6. Finally, a recommendation is made towards a control strategy that should be tried in practice.

Chapter 2

The Well-Reservoir model

As this thesis is based on my previous work [15], the well-reservoir model will be taken directly from that paper. The model was originally presented in [7], and is presented here in its unmodified form. The dynamic model is based on work presented in [2]. It is a simple one-dimensional GORM, for the sub-critical production phase.

2.1 The model

The model relies on several assumptions:

1. Only one transversal section of the reservoir will be considered–the section where gas breakthrough will first occur.

2. The longitudinal flow (parallel to the well) in the reservoir is neglected. This is justified for long reservoirs where the transversal width is insignificant compared to the longitudinal distance.

3. The vertical flow component is neglected. This is justified since the reservoir thickness h is much smaller than the reservoir width L and the slope $\frac{\partial h}{\partial x} \ll 1$. The flow close to the the well bore is neglected.

4. It is assumed that the reservoir section is rectangular, isotropic and homogeneous with respect to porosity φ and (horizontal) permeability k_h. We also assume that the well is located in the middle of the reservoir at the Oil-water contact, thus it is only necessary to consider half of the reservoir.

5. The reservoir has a gas cap with constant pressure in time and space.

6. The reservoir bottom is assumed to be a no flow boundary, thus water coning is neglected.

7. The reservoir's outer boundary is a no flow boundary.
8. The capillary forces are neglected and a segregated flow is assumed, thus there is a well-defined GOC (Gas-oil contact). Consequently, the effect of connate water saturation and residual oil is neglected.
9. The oil and gas PVT (Pressure-Volume-Temperature) properties are assumed constant in the reservoir. Thus the oil is treated as incompressible. Investigation of the PVT properties reveals that this assumption is safely assumed for most homogeneous horizontal reservoirs.
10. Horizontal reservoir pressure gradients are neglected.
11. Only Darcy flow due to gravity-based pressure gradient is considered.

Assumptions 3 and 10 make up the so called Dupuit assumptions [7].

The starting point for the model derivation is the equation of continuity, which may be written as

$$-\nabla q + s = \frac{\partial(V\varphi)}{\partial t} \tag{2.1}$$

where q is the volumetric flux, s denotes a source, and V is a volume which is projected onto the (x, y) plane to a unit area (Figure 2.1).

Assumptions 6 and 7 translate to $s = 0$. Assumptions 10 and 11 lead to the pressure gradient in an oil column during depletion being

$$\nabla p(x, y, z) = -\Delta\rho g \nabla h \tag{2.2}$$

where ρ_G denotes density of gas and ρ_O denotes oil density and $\Delta\rho = (\rho_G - \rho_O)$. g denotes the acceleration of gravity and h the oil column height. The velocity vector $\mathbf{u}$ is given by Darcy's Law:

$$\mathbf{u} = -\frac{1}{\mu_0} K \nabla p(x, y, z) \tag{2.3}$$

where μ_0 denotes viscosity. K is the permeability tensor and is given, assuming the permeability is aligned with the well and that the horizontal permeabilities are equal ($k_{xx} = k_{yy}$), by

$$K = \begin{pmatrix} k_h & 0 & 0 \\ 0 & k_h & 0 \\ 0 & 0 & k_v \end{pmatrix} \tag{2.4}$$

where k_v and k_h denote vertical and horizontal permeability, respectively. Neglecting flow in longitudinal and vertical directions (assumptions 2 and 3), K is reduced to $K = k_h$. Plugging this and equation (2.2) into equation (2.3), we get

$$u = \frac{k_h}{\mu_0} \Delta\rho g \frac{\partial h}{\partial x}. \tag{2.5}$$

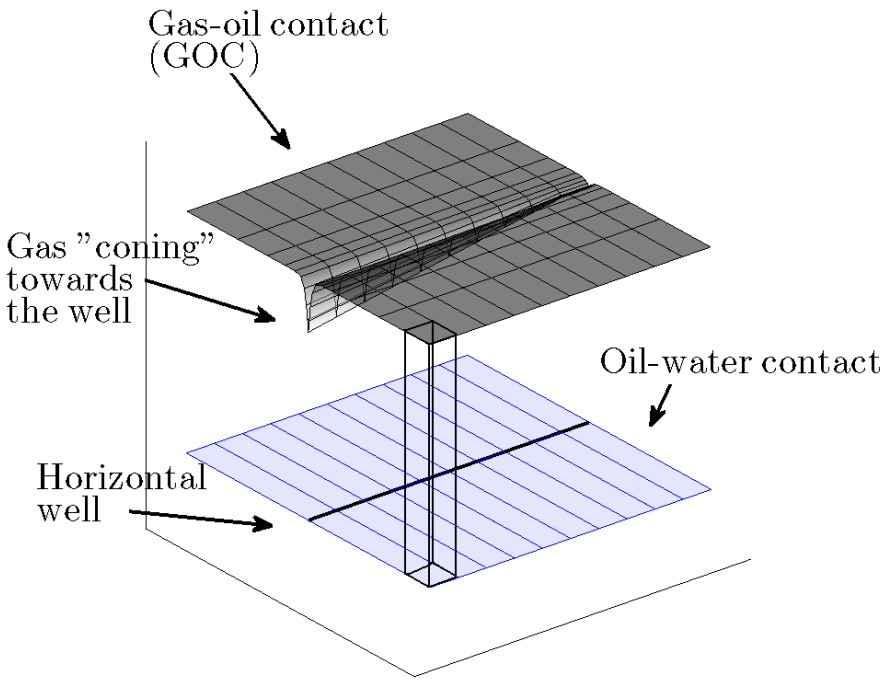


Figure 2.1: Horizontal Well Model. The figure shows the Gas-oil contact and the oil-water contact under depletion. The well is drilled along the reservoir at the oil-water contact. The gas "cones" down towards the well. The box illustrates the volume V, which is projected to a unit area on the (x, y) plane.

Note that the last term has $\frac{\partial h}{\partial x}$ instead of ∇h as we only consider the x direction.

Integrating u over a unit length of the well from 0 to $h(x,t)$, we get the volumetric flow rate into the well:

$$q(x,y,z) = \int_0^h u\, dz$$

$$q(x,y,z) = \Delta\rho g \frac{k_h}{\mu_0} \frac{\partial h}{\partial x} z|_{z=0}^{h}$$

$$q(x,y,z) = \Delta\rho g \frac{k_h}{\mu_0} h \frac{\partial h}{\partial x}. \tag{2.6}$$

x denotes transversal distance from the well towards the edge of the reservoir. $q(x,t)$ is one half of the production per unit length of well, because of assumption 4 (symmetry).

The volume of oil in a section of one unit area is then $h\varphi(1-S_w)$, where S_w is the water saturation. Combining equations (2.1) and (2.6), while noting that V, the unit volume in equation (2.1) is the same as h, we get

$$\frac{\partial}{\partial t}(h\varphi) + \nabla q = 0$$

or

$$\frac{\partial h}{\partial t} = \frac{k_h g \Delta\rho}{\mu_0 \varphi} \frac{\partial}{\partial x}\left(h \frac{\partial h}{\partial x}\right). \tag{2.7}$$

Equation (2.7) is known as Boussineq's equation for unsteady flow [7]. It can be made dimensionless by introducing the following variables:

$$\begin{aligned}
\bar{h} &= \frac{h}{h_0} \\
\bar{x} &= \frac{x}{h_0}, \bar{x} = [0, L] \\
\bar{t} &= t\frac{k_h g \Delta\rho}{\mu_0 \varphi h_0} \\
\bar{q} &= q\frac{\mu_0}{k_h g \Delta\rho h_0}.
\end{aligned}$$

Equation (2.7) becomes

$$\frac{\partial \bar{h}}{\partial \bar{t}} = \frac{1}{2}\nabla^2 \bar{h}^2. \tag{2.8}$$

For the rest of this report, the bars denoting dimensionless variables will be omitted. Written out, equation (2.8) becomes

$$\begin{aligned}
\frac{\partial h}{\partial t} &= \frac{1}{2}\frac{\partial^2}{\partial x^2}(h^2) \\
&= \frac{1}{2}\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}h^2\right) \\
&= \frac{1}{2}\frac{\partial}{\partial x}\left(2h\frac{\partial h}{\partial x}\right) \\
&= h\frac{\partial^2 h}{\partial x^2} + \left(\frac{\partial h}{\partial x}\right)^2.
\end{aligned}$$

$$h_t = hh_{xx} + h_x^2, \tag{2.9}$$

and the boundary and initial conditions:

$$h(0, x) = 1 \tag{2.10}$$

$$h_x(t, L) = 0 \tag{2.11}$$

$$h_x(t, 0)h(t, 0) = -\frac{1}{2}q(t). \tag{2.12}$$

The mixed boundary condition (2.12) comes from (2.6).

In [7], the model is then linearized as follows:

$$h_t = h_c h_{xx}. \tag{2.13}$$

This concludes the reproduction of the well-reservoir model from [7].

Neumann Control

In order to formulate the model with Neumann actuation, we may simply use the transformation

$$q(t) = -2h(t,0)U(t), \tag{2.14}$$

and thus write boundary condition (2.12) as

$$h_x(t,0) = U(t). \tag{2.15}$$

Alternatively, by placing more pressure sensors in the well, it is possible to measure $h_x(t,0)$ [16]. Then, using the transformation

$$q(t) = -2h_x(t,0)U(t) \tag{2.16}$$

we may arrive at the Dirichlet boundary condition

$$h(t,0) = U(t). \tag{2.17}$$

Using the Neumann boundary condition, the linearized system can then be written as:

$$h_t(t,x) = h_c h_{xx}(t,x) \tag{2.18}$$
$$h(t,x) = 1 \tag{2.19}$$
$$h_x(t,1) = 0 \tag{2.20}$$
$$h_x(t,0) = U(t) \tag{2.21}$$

2.2 Spacial discretization

Some of the techniques for analysis and design of both controllers and observers in the following chapters rely on a discretized model, where the system is governed by a set of ODEs rather than a PDE. This chapter is about a discretization in space, not in time. Instead of representing the oil column height as a continuous function $h(t,x)$, we will now consider a discrete set of $h_i(t) = h(t, i\Delta x)$, where $i = 0..N-1$, N is the number of mesh points and $\Delta x = \frac{1}{N}$.

The linear PDE model is given by

$$h_t(t,x) = h_c h_{xx}(t,x) \tag{2.22}$$
$$h_x(t,0) = 0 \tag{2.23}$$
$$h_x(t,1) = u(t). \tag{2.24}$$

An order-two accurate, two-sided approximation of the second derivative is [17]

$$\frac{\partial^2 h_i}{\partial x^2} \approx \frac{h_{i+1} - 2h_i + h_{i+1}}{\Delta x^2}. \tag{2.25}$$

For all $1 \leq i \leq N-2$, we may thus write

$$\dot{h_i} = h_c \frac{h_{i-1} - 2h_i + h_{i+1}}{\Delta x^2}, \tag{2.26}$$

whereas for $i = 0$ and $i = N-1$, we must enforce the boundary conditions (2.23) and (2.24). Since the boundary conditions are of Neumann type, we must approximate the derivative there too. A one-sided, order-two accurate approximation is [17]

$$\frac{\partial h_i}{\partial x} \approx \frac{-3h_i + 4h_{i+1} - h_{i+2}}{2\Delta x}. \tag{2.27}$$

We may use this to enforce $h_x(t,0) = 0$ by demanding that

$$\frac{\partial h_0}{\partial x} \approx \frac{-3h_0 + 4h_1 - h_2}{2\Delta x} = 0. \tag{2.28}$$

Then, the boundary condition (2.23) translates to

$$h_0 = -\frac{-4h_1 + h_2}{3}. \tag{2.29}$$

Using the same approximation, the boundary condition (2.24) becomes

$$h_{N-1} = \frac{2\Delta x u(t) + 4h_{N-2} - h_{N-3}}{3}. \tag{2.30}$$

By plugging (2.29) and (2.30) into (2.26) for $i = 1$ and $i = N-2$ respectively, we may write the system in matrix form, with the boundary conditions enforced, on the state vector $\mathbf{h} = (h_1, h_2, \cdots, h_{N-2})^T$ as

$$\dot{\mathbf{h}} = \mathbf{A}\mathbf{h} + \mathbf{b}u, \tag{2.31}$$
$$y = h_{N-1} = \mathbf{C}\mathbf{h} + Du, \tag{2.32}$$

with

$$\mathbf{A} = \begin{pmatrix} -\frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & & & \vdots \\ \vdots & & & & 0 & 1 & -2 & 1 \\ 0 & & \cdots & & & 0 & \frac{2}{3} & -\frac{2}{3} \end{pmatrix} \frac{h_c}{\Delta x^2}, \tag{2.33}$$

$$\mathbf{b} = \begin{pmatrix} 0 & \cdots & 0 & \frac{2h_c}{3\Delta x} \end{pmatrix}^T, \tag{2.34}$$

$$\mathbf{C} = \begin{pmatrix} 0 & \cdots & 0 & -\frac{1}{3} & \frac{4}{3} \end{pmatrix} \tag{2.35}$$

$$D = \frac{2\Delta x}{3}. \tag{2.36}$$

Discretizing the nonlinear model

In this section, the nonlinear model (2.9)–(2.12) will be discretized in space. The equation is given by

$$h_t(t,x) = \frac{1}{\alpha}\left(hh_{xx} + h_x^2\right), \tag{2.37}$$

with the boundary conditions (2.23),(2.24). Using the same finite-differences scheme as for the linear case, we use the approximation (2.25) for the second derivative. An approximation of the first derivative is [17]

$$\frac{\partial h_i}{\partial x} \approx \frac{h_{i+1} - h_{i-1}}{2\Delta x}. \tag{2.38}$$

Thus we may write

$$\dot{h_i} = \frac{1}{\alpha}\left(h_i \frac{h_{i-1} - 2h_i + h_{i+1}}{\Delta x^2} + \left(\frac{h_{i+1} - h_{i-1}}{2\Delta x}\right)^2\right) \tag{2.39}$$

for $i = 1..N-2$, and plug in (2.29) and (2.30) to enforce the boundary conditions.

Chapter 3

Active Control

In this chapter several control strategies will be presented. The control objective will throughout this thesis be to maximize earnings in the sub-critical production phase. That is, to maximize *net present value* of the oil production prior to gas breakthrough. This will be represented using the functional

$$J[q(t)] = \int_{t_0}^{t_f} q(t) \underbrace{(1+I)^{\frac{t}{c}}}_{npv(t)} dt \tag{3.1}$$

where $q(t)$ is the volumetric flow of oil and I is the interest rate. t_f is the time of gas breakthrough – the time when the well shifts from the sub-critical to the super-critical phase, and c is a constant (the number of seconds in a year).

The nonlinear dynamics (2.9)–(2.12) govern the behavior of the well-reservoir system. The actuation enters through the nonlinear boundary condition (2.12), and thus we say the system is *boundary actuated*. The nonlinear Robin-type boundary condition may be transformed to Neumann or Dirichlet-type boundary conditions, as was shown in Section 2.1. Boundary actuated plants with that kind of boundary conditions have been analyzed in the past (for instance [18],[19],[20]) and stabilizing boundary controllers have been synthesized.

In Section 2.2, the PDE model was discretized in space to produce a system of connected ODEs. One may also call these boundary actuated, since the control input affects only the boundary state. Some of the control strategies discussed below will be designed directly for the PDE model, while others will be designed for the ODE system.

The next section will discuss the closed-loop stability of the full, nonlinear PDE model. Then, control laws from previous work as well as new ones will be presented.

3.1 Stability

This section will examine the stability of the the well-reservoir system described in section 2.1. The model (2.9)–(2.12) can be written

$$h_t(t,x) = h(t,x)h_{xx}(t,x) + h_x^2(t,x) \tag{3.2}$$

$$h(t,1)h_x(t,1) = -\frac{1}{2}q(t) \tag{3.3}$$

$$h_x(t,0) = 0. \tag{3.4}$$

It is possible to flip the reservoir – the well now at $x = 1$ instead of $x = 0$. Notice that the dynamics are still the same. Then, choosing the Lyapunov functional

$$V\left[h(t,x)\right] = \frac{1}{2}\int_0^1 h^2(t,x)dx, \tag{3.5}$$

and noting that $V > 0$ for $h(t,x) \neq 0$ and $V \equiv 0$ for $h(t,x) = 0$ (V is positive definite), we may compute its time derivative:

$$\begin{aligned}
\dot{V} &= \int_0^1 h(t,x)\Big(h(t,x)h_{xx}(t,x) + h_x^2(t,x)\Big)dx \\
&= \int_0^1 h^2(t,x)h_{xx}(t,x)dx + \int_0^1 h(t,x)h_x^2(t,x)dx \\
&= \Big[h^2(t,x)h_x(t,x)\Big]_0^1 - 2\int_0^1 h(t,x)h_x^2(t,x)dx + \int_0^1 h(t,x)h_x^2(t,x)dx \\
&= -\int_0^1 h(t,x)h_x^2(t,x)dx - \frac{1}{2}h(t,1)q(t).
\end{aligned} \tag{3.6}$$

Since $h(t,x) \geq 0$, both terms are non-positive and $\dot{V} \leq 0$, with equality if and only if $h(t,x) = 0$. Together with positive definiteness of V, we have that every positive choice of $q(t)$ guarantees global asymptotic stability of (3.2) – (3.4). It follows that for each of the control strategies developed in the following sections, the only criteria for stability is $q(t) > 0\ \forall t$. Stabilization of (3.2) – (3.4) should be interpreted as $h \to 0$, meaning all oil is drained from the reservoir.

3.2 Sagatun's Controller

In [7], the controller

$$q(t) = -kh^2(t,1), \tag{3.7}$$

where the well is located at $x = 1$, is shown to stabilize the linear plant (2.13). The plant is then solved analytically, and the solution is plugged into the objective function (3.1). The objective function can thus be maximized analytically, for any set of reservoir parameters.

Since the previous section has shown that the nonlinear plant is stable for any $q(t) > 0$, Sagatun's controller may also be applied directly to the nonlinear model. However, a solution is not possible to compute analytically for the nonlinear PDE, and thus the objective function will have to be optimized numerically.

3.3 Backstepping

This section is taken in its entirety from [15].

The backstepping method comes from the field of nonlinear ODEs. Consider the system

$$\begin{aligned}\dot{x_1} &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x_2} &= f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \dot{x_3} &= f_3(x_1, x_2, x_3) + u,\end{aligned}$$

where u is the control input. Notice the triangular dependency structure [21]. The idea in backstepping is to treat x_2 as the input to the $\dot{x_1}$ equation, and design a stabilizing control law. Then, x_3 is used to design a control law for the $\dot{x_2}$ equation which makes it act as the desired control for $\dot{x_1}$. Finally, u is used to make x_3 behave as desired. Thus, we have stepped back through the system of equations, exploiting the triangular structure, and come up with a stabilizing control law. The generalization of the backstepping method to PDEs in [18], for linear PDEs, and in [22],[23] for nonlinear PDEs, is nontrivial. The lower-triangular structure comes from a change of variables which involves a Volterra integral operator with a lower-triangular structure. It is a method of boundary control because the control input will be at the boundary of the PDE domain, as in equation (2.18). This is comparable to the ODE case above, where the control u enters through x_3, which may be thought of as the boundary of the state space. Backstepping is best illustrated through an example.

Example 1 (Backstepping for unstable ODE system)**.** *Consider the system*

$$\dot{x_1} = x_1^2 + x_2 \tag{3.8}$$

$$\dot{x_2} = x_3 \tag{3.9}$$

$$\dot{x_3} = u \tag{3.10}$$

where u is the control input. This system suffers from a finite escape-time instability (Figure 3.1). Treating x_2 as a virtual control *for the $\dot{x_1}$ system, we may design a stabilizing input such as $\phi_1(x_1) = -x_1 - x_1^2$. Doing so would reduce the $\dot{x_1}$ system to*

$$\dot{x_1} = -x_1 \tag{3.11}$$

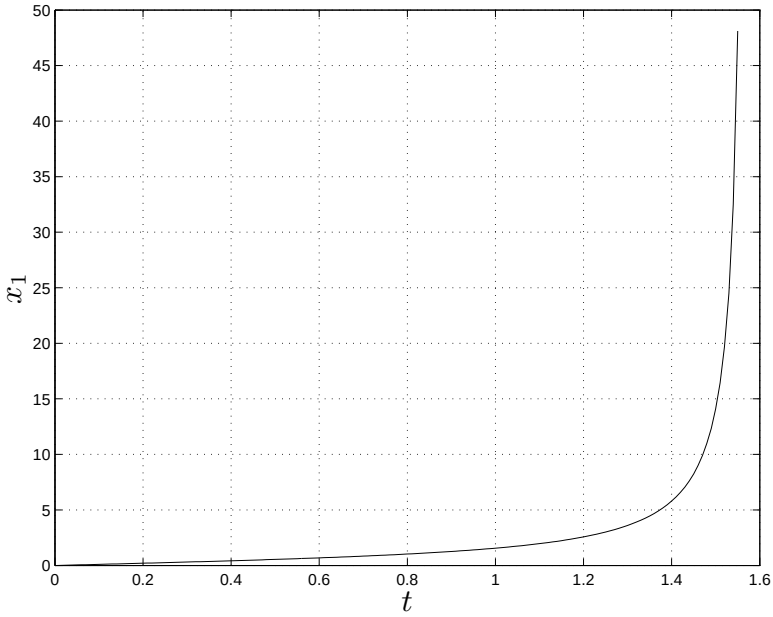


Figure 3.1: Simulation of system with finite escape-time. Initial conditions: $x_1(0) = x_2(0) = 0$, $x_3(0) = 1$. $u(t) = 0$.

which can be proven to be globally asymptotically stable using the Lyapunov-function

$$V_1(x_1) = \frac{1}{2}x_1^2 \tag{3.12}$$

Adding and subtracting ϕ_1 on both sides of the $\dot{x_1}$ equation, we may write

$$\begin{aligned}\dot{x}_1 &= x_1^2 + \phi_1(x_1) + (x_2 - \phi_1(x_1)) \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= u.\end{aligned}$$

Now, using the transformation $z_2 = x_2 - \phi_1(x_1)$ and, noting that $\dot{\phi}_1(x_1) = \frac{\partial \phi_1}{\partial x_1}\dot{x_1}$ we arrive at

$$\begin{aligned}\dot{x}_1 &= -x_1 + z_2 \\ \dot{z}_2 &= x_3 - \dot{\phi} = x_3 - (-2x_1 - 1)(-x_1 + z_2) \\ &= x_3 + 2x_1^2 - 2x_1 z_2 + x_1 - z_2 \\ \dot{x}_3 &= u.\end{aligned}$$

The process may be repeated again, treating x_3 as a virtual control for the rest of the plant. Introducing the change of variables

$$z_3 = x_3 - x_1 + z_2 - 2x_1^2 + 2x_1 z_2 + z_2 \tag{3.13}$$

and the control

$$u = -x_1 - 2x_1^2 + 6x_1 z_2 - 3z_3 + 2x_1 z_3, \tag{3.14}$$

the system (3.10) *is transformed to the globally asymptotically stable*

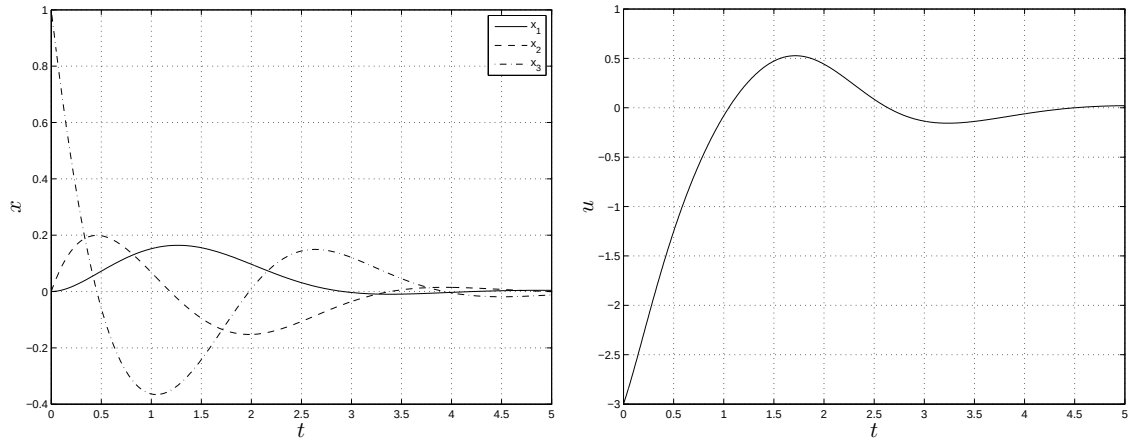
$$\dot{z_1} = -z_1 + z_2 \tag{3.15}$$

$$\dot{z_2} = -z_2 + z_3 \tag{3.16}$$

$$\dot{z_3} = -z_3. \tag{3.17}$$

Note that $z_1 = x_1$.

The nonlinear instability in x_1 *has thus been* backstepped *through two integrators, and the system* (3.8)–(3.10) *has been stabilized (Figure 3.2(a)).*



(a) (b)

Figure 3.2: The ODE plant (3.8)–(3.10) was stabilized using the backtepping method. (a) shows the response $x(t)$ and (b) is the control input u when using the controller (3.18).

Written in x *coordinates, the control law is*

$$u = -4x_1 - 3x_3 - 6x_2 + 2x_1 x_3 + 4x_1 x_2 + 4x_1^2 x_2 + 4x_1^3 + 4x_1^4. \tag{3.18}$$

This example has demonstrated how backstepping may be used to transform an unstable ODE plant into a stable one, and showed the level of sophistication of the control law.

3.3.1 Backstepping for PDEs

Similarly to a system of ODEs, the goal of backstepping for a PDE plants is to stabilize the system. Also, the transformation is achieved by a change of variables and a control law. The following example, reproduced from [18], illustrates the method for an unstable parabolic diffusion-reaction plant.

Example 2 (Backstepping for parabolic PDE)**.** *Consider the plant*

$$u_t(t,x) = u_{xx}(t,x) + \lambda u(t,x) \tag{3.19}$$
$$u_x(t,0) = 0 \tag{3.20}$$
$$u(t,1) = U(t) \tag{3.21}$$

where $U(t)$ is the control input. It can be shown that the plant is unstable for $\lambda > threshold$, see Appendix A.2. The plant will be transformed into the stable system

$$w_t(t,x) = w_{xx}(t,x) \tag{3.22}$$

using the transformation of variables

$$w(t,x) = u(t,x) - \int_0^x k(x,y)u(t,y)dy \tag{3.23}$$

and the control law

$$U(t) = \int_0^1 k(x,1)u(t,y)dy. \tag{3.24}$$

The boundary conditions of the target system are found by substituting (3.19)-(3.21) *and* (3.24) *into* (3.23)*:*

$$w_x(t,0) = u_x(t,0) = 0 \tag{3.25}$$
$$w(t,1) = u(t,1) - \int_0^1 k(x,y)u(t,y)dy = 0. \tag{3.26}$$

This is known as the Volterra integral transformation. The kernel *$k(x,y)$ is given by a PDE which is found by taking the time and space derivatives of* (3.23) *and substituting them into* (3.22)*. In order to do that we must use the Leibniz rule of differentiation under the integral:*

$$\frac{d}{dx}\int_0^x f(x,y)dy = f(x,x) + \int_0^x f_x(x,y)dy. \tag{3.27}$$

Differentiating (3.23) *with respect to x, we get (omitting time dependence)*

$$w_x(x) = u_x(x) - k(x,x)u(x) - \int_0^x k_x(x,y)u(y)dy \tag{3.28}$$

and

$$\begin{aligned} w_{xx}(x) = u_{xx} &- \frac{d}{dx}(k(x,x))u(x) - k(x,x)u_x(x) - k_x(x,x)u(x) \\ &- \int_0^x k_{xx}(x,y)u(y)dy, \end{aligned} \tag{3.29}$$

where $\frac{d}{dx}k(x,x) = k_x(x,x) + k_y(x,x)$. *Differentiating with respect to time, we have*

$$\begin{aligned} w_t(x) &= u_t(x) - \int_0^x k(x,y)u_t(y)dy \\ &= u_{xx}(x) + \lambda u(x) - \int_0^x k(x,y)[u_{xx}(y) + \lambda u(y)]dy \\ &= u_{xx}(x) + \lambda u(x) - k(x,x)u_x(x) + k(x,0)u_x(0) \\ &+ \int_0^x k_y(x,y)u_y(y)dy - \int_0^x \lambda k(x,y)u(y)dy \\ &= u_{xx}(x) + \lambda u(x) - k(x,x)u_x(x) + k_y(x,x)u(x) \\ &- k_y(x,0)u(0) - \int_0^x k_{yy}(x,y)u(y)dy - \int_0^x \lambda k(x,y)u(y)dy, \end{aligned} \tag{3.30}$$

where we use integration by parts in the last two steps. In order to have (3.22)*, we require that* (3.29) *equals* (3.30)*. Subtracting* (3.29) *from* (3.30)*, we have*

$$\begin{aligned} w_t(x) - w_{xx} = &\left[\lambda + 2\frac{d}{dx}k(x,x)\right]u(x) - k_y(x,0)u(0) \\ &+ \int_0^x \left(k_{xx}(x,y) - k_{yy}(x,y) - \lambda k(x,y)\right)u(y)dy. \end{aligned} \tag{3.31}$$

In order to satisfy (3.22)*, all three terms on the right hand side must equal zero. Thus, we get the following equations for the gain kernel* $k(x,y)$*:*

$$k_{xx}(x,y) - k_{yy}(x,y) = \lambda k(x,y) \tag{3.32}$$

$$k_y(x,0) = 0 \tag{3.33}$$

$$k(x,x) = -\frac{\lambda}{2}x. \tag{3.34}$$

Equation (3.34) *comes from integrating the first term on the right hand side of* (3.31)*. This wave-equation-like hyperbolic PDE is well-posed and the closed-form solution has been computed in [18]. The solution is given by*

$$k(x,y) = -\lambda x \frac{I_1\left(\sqrt{\lambda(x^2-y^2)}\right)}{\sqrt{\lambda(x^2-y^2)}}. \tag{3.35}$$

The function $I_1(x)$ *in* (3.35) *is the solution to a modified Bessel function (see Appendix A.2 in [18]). This example showed how to find equations for the kernel in the Volterra integral transformation which stabilizes an unstable diffusion-reaction PDE.*

In the following section, a Backstepping controller for the horizontal well will be designed.

3.3.2 Backstepping for the horizontal oil well

The well-reservoir model (2.9)–(2.12) is nonlinear in both its terms. Only one result for a general class of PDEs, given by

$$u_t(t,x) = u_{xx}(t,x) + \lambda(x)u(t,x) + F[u](t,x) + uH[u](t,x), \tag{3.36}$$

has been published [22],[23]. In their papers, a method is presented for finding Backstepping transformations for the class of systems given by (3.36), where $F[u]$ and $H[u]$ are Volterra series nonlinearities. The notation $F[u](t,x)$ means that $F[u]$ is a functional of $u(t,x)$. The Volterra series is defined in [22]. Other results, stabilizing nonlinear PDEs were found for the viscous Burgers equation [20],[24], and the Navier-Stokes equation [25].

However, none of these include treatment of the nonlinear diffusion term, hh_{xx} and the nonlinear advection term h_x^2. Therefore, the Backstepping design will be based on the linear model (2.13). Sagatun's design in [7] achieves promising results when the controller design is based on the linearized model. The same approach may then yield good results for the Backstepping method. The Backstepping method is to transform the system

$$h_t(t,x) = h_{xx} \tag{3.37}$$
$$h_x(t,0) = 0 \tag{3.38}$$
$$h_x(t,1) = u(t) \tag{3.39}$$

where $u(t)$ is the control signal, into the target system

$$w_t(t,x) = w_{xx} \tag{3.40}$$
$$w_x(t,0) = 0 \tag{3.41}$$
$$w_x(t,1) = 0, \tag{3.42}$$

which is globally exponentially stable [18]. Note that

$$u(t) = -\frac{q(t)}{2h(t,1)}. \tag{3.43}$$

The transformation is performed using the Volterra integral transformation

$$w(t,x) = h(t,x) - \int_0^x k(x,y)h(t,y)dy \tag{3.44}$$

and the control law

$$u(t) = k(1,1)h(t,1) + \int_0^1 k_x(1,y)h(y,t)dy, \tag{3.45}$$

as in Example 2. However, since the control enters as a Neumann condition, unlike the Dirichlet control from Example 2, the transforming control will now be $u(t) = h(1)k(1,1) + \int_0^1 k_x(1,y)h(y)dy$. If, however, we follow the same procedure as in Example 2, we arrive at $k(x,y) \equiv 0$; it is indeed easy to see that $u(t) = 0$ transforms the plant (3.37)–(3.39) to the target system (3.40)–(3.42). Let us instead consider a perturbation of the model (2.13), given by

$$u_t(t,x) = u_{xx} \tag{3.46}$$
$$u_x(t,0) = -\epsilon u(0,t) \tag{3.47}$$
$$u_x(t,1) = U(t), \tag{3.48}$$

where $U(t)$ is the control signal. ϵ in (3.47) is a small parameter, ($0 < \epsilon \ll 1$), ensuring the plant (3.46)–(3.48) is close to the nominal system (2.18),(2.20),(2.21).

We may also look at this perturbation as modeling only the dynamics up to a distance d from the well, rather than the whole reservoir. Then, (3.37)–(3.39) become

$$h_t(t,x) = h_{xx} \tag{3.49}$$
$$h_x(t,1) = u(t) \tag{3.50}$$
$$h(t,d)h_x(t,d) = -\frac{1}{2}q_d(t), \tag{3.51}$$

where $0 < d < 1$. The boundary condition (3.51) is the same as (2.12), and the physical interpretation is the same – q_d is the oil flux across the reservoir at $x = d$. Figure 3.3 illustrates the concept.

Without loss of generality, we write $d = 0$, shifting the variable space x. The system can thus be written as

$$h_t(t,x) = h_{xx} \tag{3.52}$$
$$h(t,0)h_x(t,0) = -\frac{1}{2}q_d(t) \tag{3.53}$$
$$h_x(t,1) = u(t). \tag{3.54}$$

Now we follow the procedure of Example 2. First, note that the equation for w_{xx} does not depend on the particular plant, and is the same as (3.29). Next, we differentiate (3.23) with respect to time, again omitting time-dependence:

$$\begin{aligned} w_t(x) &= h_t(x) - \int_0^x k(x,y)h_t(y)dy \\ &= h_{xx}(x) - \int_0^x k(x,y)h_{yy}(y)dy \\ &= h_{xx}(x) - \left(k(x,x)h_x(x) - k(x,0)h_x(0) - \int_0^x k_y(x,y)h_y(y)dy\right) \\ &= h_{xx}(x) - k(x,x)h_x(x) + k(x,0)h_x(0) + k_y(x,x)h(x) \\ &\quad - k_y(x,0)h(0) - \int_0^x k_{yy}(x,y)h(y)dy. \end{aligned} \tag{3.55}$$

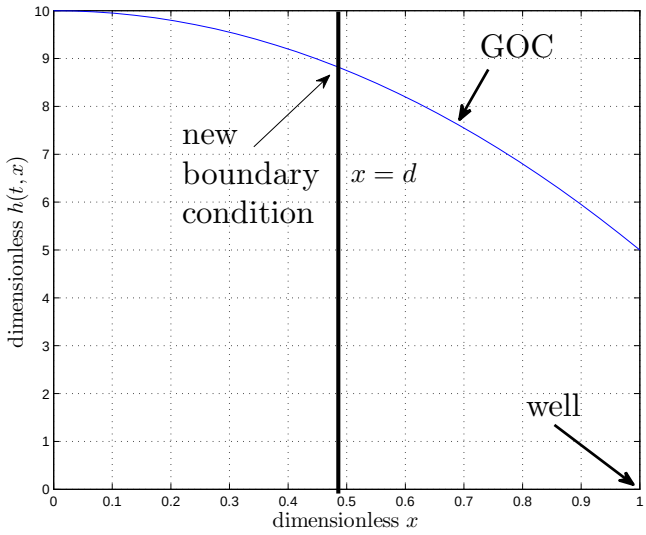


Figure 3.3: Near-well modeling. We may consider the boundary at $x = d$, a distance from the well, instead of $x = 0$, which is the edge of the reservoir.

Finally, we subtract (3.29) from (3.55):

$$\begin{aligned} w_t - w_{xx} = 2\frac{d}{dx}k(x,x)h(x) - k_y(x,0)h(0) + k(x,0)h_x(0) \\ + \int_0^x \left(k_{xx}(x,y) - k_{yy}(x,y)\right) h(y)dy. \end{aligned} \tag{3.56}$$

Still using (3.40)–(3.42) as the target system, the right hand side of (3.56) needs to be identically zero. This leaves us with three conditions:

$$2\frac{d}{dx}k(x,x) = 0 \tag{3.57}$$

$$k_y(x,0)h(0) = k(x,0)h_x(0) \tag{3.58}$$

$$k_{xx}(x,y) = k_{yy}(x,y). \tag{3.59}$$

Integrating condition (3.57), and using (3.28), we get

$$k(x,x) = k(0,0) \tag{3.60}$$

$$w_x(x) = h_x(x) - k(x,x)h(x) - \int_0^x k_y(x,y)h_y(y)dy. \tag{3.61}$$

Setting $x = 0$ in (3.61) and using the boundary conditions (3.41) and (3.53), we get

$$w_x(0) = h_x(0) - k(0,0)h(0) \tag{3.62}$$

$$0 = -\frac{1}{2}q_d(t) - k(0,0)h^2(0) \tag{3.63}$$

$$k(x,x) = -\frac{q_d(t)}{2h^2(t,0)}. \tag{3.64}$$

Similarly, inserting (3.53) into condition (3.58), we get

$$k_y(x,0) = -\frac{q_d(t)}{2h^2(0)}k(x,0). \tag{3.65}$$

The gain kernel PDE is then given by equations (3.59),(3.64) and (3.65).

The quantities $q_d(t)$ and $h(0)$ are not directly measurable. However, noting that $q_d \ll h^2(0)$, we may simplify the equations, and get the gain kernel PDE

$$k_{xx}(x,y) - k_{yy}(x,y) = 0 \tag{3.66}$$

$$k_y(x,0) = -\epsilon k(x,0) \tag{3.67}$$

$$k(x,x) = -\epsilon, \tag{3.68}$$

with $\epsilon \ll 1$. This ϵ may be used as a tuning parameter for the emerging controller. The PDE (3.66) – (3.68) has the solution (see Appendix A.1)

$$k(x,y) = -\epsilon e^{\epsilon(x-y)}. \tag{3.69}$$

Thus, the backstepping design results in the controller

$$u(t) = -\epsilon h(t,1) - \epsilon^2 \int_0^1 e^{\epsilon(1-y)}h(t,y)dy, \tag{3.70}$$

which, using (3.43) leads to the flow controller

$$\boxed{q(t) = 2\epsilon h^2(t,1) + 2\epsilon^2 h(t,1)\int_0^1 e^{\epsilon(1-y)}h(t,y)dy.} \tag{3.71}$$

3.4 Generalizing Sagatun

In [15], I proposed a generalization of Sagatun's controller, based on the finding that any positive q stabilizes the nonlinear model. It was suggested that the controller

$$q(t) = kh^a(t,1) \tag{3.72}$$

may provide an improvement in performance. Like Sagatun's controller, it is an output-feedback control law, unlike the Backstepping controller which requires state-feedback. The generalized Sagatun's controller has two parameters, which may be used to maximize the objective function (3.1).

3.5 Max-Min Control

Simulations show that when using the generalized Sagatun controller, the production profile takes the form shown in Figure 3.4.

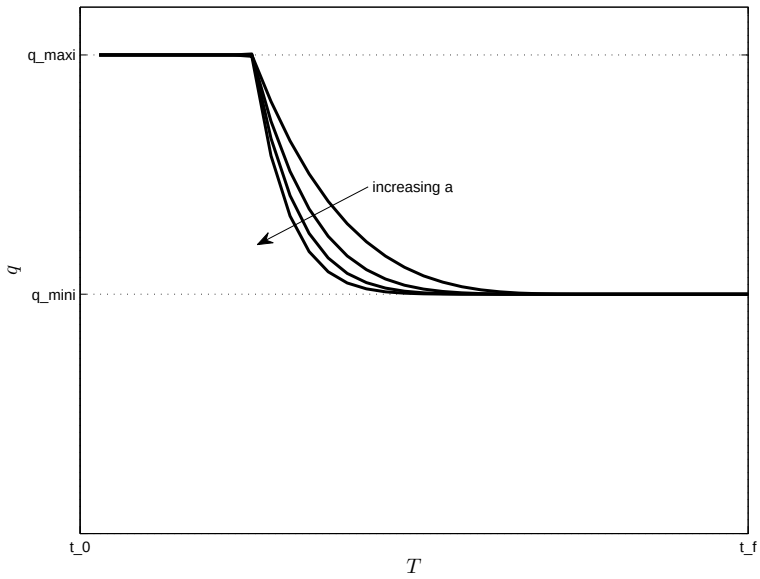


Figure 3.4: Production profile when using the controller $q(t) = -kh^a(t, 1)$. Increased a results in a sharper transition from maximum to minimum capacity.

Simulations also show that an increased a brings an increase in *npv* of sub-critical production. It is not possible to let $a \to \infty$ when simulating due to numeric instability issues. It is clear though, that the limit as a approaches ∞ is a step from q_{max} to q_{min}. An optimization problem would then be to find the optimal time-instance t_{step} at which this transition should take place.

This idea is credited to Vidar Gunnerud at the Department of Engineering Cybernetics, Norwegian University of Science and Technology.

3.6 Optimal Control

In the previous sections of this chapter, classical controllers have been developed with at least one control parameter. That parameter may be varied, and the resulting system behavior assessed according the the control objective – the objective function (3.1). Controllers are generally designed to be primarily stabilizing, and only secondarily to optimize an objective function. For the well-reservoir system in this project, any

controller is stabilizing, and we have therefore been free to explore different controller configurations in an attempt to maximize the objective function.

A different strategy is that of *optimal control*. Here, we attempt to find a sequence $q(t)$ that directly optimizes the control objective. The problem statement is

$$\min_{q(t)} J[q(t)] \text{ subject to} \tag{3.73}$$

the constraints

$$h_t(t,x) = f(h, h_x, h_{xx}) \tag{3.74}$$
$$h_x(t,0) = 0 \tag{3.75}$$
$$h_x(t,1)h(t,1) = -\frac{1}{2}q(t) \tag{3.76}$$
$$h(0,x) = h_0 \tag{3.77}$$
$$h(t,1) \geq 1 \tag{3.78}$$
$$q_{min} \leq q(t) \leq q_{max}, \tag{3.79}$$

where $J[q(t)]$ is given by (3.1), f in (3.74) is a linear or nonlinear function given in Chapter 2, and (3.78) is meant to keep the well in the sub-critical production phase. Conditions (3.74)–(3.77) are equality constraints, and (3.78),(3.79) are known as inequality constraints.

Since this problem is nonlinear even in the case of the linear $f(h, h_x, h_{xx}) = h_c h_{xx}$, because of the nonlinear boundary condition (3.76), the problem may be non-convex, and is therefore difficult to solve.

At this point we are also faced with the dilemma of discretization, both in space and time. Depending on the chosen approach to solve (3.73)–(3.79), the system may have to be discretized in either time, space or both.

Numerical Optimization

Powerful numerical methods have been developed for both linear and nonlinear optimization problems. These algorithms are iterative in nature, and the way these iterations are performed differs from algorithm to algorithm [26].

Active-Set methods assume a subset of the inequality constraints are *active*, and treats them as inequality constraints. It then solves the Karush-Kuhn-Tucker (KKT) equations [26]. Then, if the solution is outside of the feasible region (violates the constraints), a line-search is employed to find a feasible solution, a new active set of constraints is found, and a new iteration begins.

In *Interior-Point* methods, a series of approximated, equality-constrained problems are solved. At each step, the KKT equations are solved [26].

Both of these algorithms are available for nonlinear optimization in MATLAB's `fmincon` – nonlinear constrained optimization.

3.6.1 Quadratic Programming

The well-reservoir PDE is a large, stiff system. Nonlinear numeric optimization is heavy. 20 spacial variables and 100 time-points result in a state vector $(20+1)\cdot 100 = 2100$ states. In order to enforce the stiff system dynamics, a stable solver is necessary. The solver will need to be run n_t times at each evaluation of the constraints, which proves to be slow as well as memory-consuming.

Powerful quadratic programming algorithms exist for quadratic optimization problems of the kind

$$\begin{aligned}\min_z \frac{1}{2} z^T H z + f^T z \text{ subject to}\\ A_e z = b_e\\ Az \leq b\\ lb \leq z \leq ub.\end{aligned}$$

Quadratic programming algorithms such as *active-set* methods are more efficient than their general, nonlinear counterparts. It would therefore be advantageous if the optimization problem (3.73)–(3.79) can be approached by a quadratic programming problem. Therefore, the linear model $h_t = h_c h_{xx}$ will be used in this section.

In order to enforce the system dynamics using linear constraints of the form $A_e z = b_e$, the system needs to be discretized in time, as well as space. Spacial discretization was already done in Section 2.2. Since the system is linear, its solution is well known, and no approximations need to be made in order to discretize it in time. Note that this was not possible to do for spacial discretization, as the solution depends on the nonlinear boundary condition, whereas we here only deal with the simplified $\dot{\mathbf{h}} = A\mathbf{h} + Bu$. The following procedure from [27] is known as *exact discretization*.

Continuing in the notation of Section 2.2, we write $h_i(t) = h(t, i\Delta x)$ and $\mathbf{h} = (h_1, .., h_{N-2})^T$. Now let

$$\mathbf{h}_k = \mathbf{h}(k\Delta t) \text{ for } k = 0, 1, 2, ... \tag{3.80}$$

The solution $\mathbf{h}(t)$ is given by

$$\mathbf{h}_k = \mathbf{h}(k\Delta t) = e^{Ak\Delta t}\mathbf{h}(0) + \int_0^{k\Delta t} e^{A(k\Delta t - \tau)} Bu(\tau)\,\mathrm{d}\tau. \tag{3.81}$$

Then we may write

$$\begin{aligned}\mathbf{h}_{k+1} = e^{A\Delta t}\Big(e^{Ak\Delta t}\mathbf{h}(0) + \int_0^{k\Delta t} e^{A(k\Delta t-\tau)} Bu(\tau)\,\mathrm{d}\tau\Big)\\ + \int_{k\Delta t}^{(k+1)\Delta t} e^{A(k\Delta t + \Delta t - \tau)} Bu(\tau)\,\mathrm{d}\tau.\end{aligned} \tag{3.82}$$

Noting that $u(t)$ is piecewise constant and writing $u(t) = u_k$ for $k\Delta t \leq t < (k+1)\Delta t$, (3.82) may be written as

$$\mathbf{h}_{k+1} = e^{A\Delta t}\mathbf{h}_k + \Big(\int_0^{\Delta t} e^{A\alpha}\,\mathrm{d}\alpha\Big)Bu_k, \tag{3.83}$$

where the substitution of variables $\alpha = k\Delta t + \Delta t - \tau$ was used. We may now write the discrete-time model as

$$\mathbf{h}_{k+1} = \Phi\mathbf{h}_k + \Psi u_k, \tag{3.84}$$

where $\Phi = e^{A\Delta t}$ and $\Psi = \Big(\int_0^{\Delta t} e^{A\alpha}\,\mathrm{d}\alpha\Big)B$.

Now, defining the state vector z from the optimization problem statement as

$$z = (\mathbf{h}_0, u_0, \mathbf{h}_1, u_1, \ldots, \mathbf{h}_{M-1}, u_{M-1})^T, \tag{3.85}$$

the state space equation (3.84) may be enforced through the constraint $A_e z = b_e$, where

$$A_e = \begin{pmatrix} I_{k\times k} & 0 & & & & & & & \\ \Phi_{k\times k} & \Psi_{1\times 1} & -I_{k\times k} & 0 & & & & & \\ 0 & 0 & \Phi_{k\times k} & \Psi_{1\times 1} & -I_{k\times k} & 0 & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & \Phi_{k\times k} & \Psi_{1\times 1} & -I_{k\times k} & 0 \end{pmatrix}, \tag{3.86}$$

with $\mathbf{h} \in \mathbb{R}^k$ and

$$b_e = (\mathbf{h}(0), 0, \ldots, 0)^T. \tag{3.87}$$

The dimensions of the vectors and matrices are

$$\begin{aligned} z &\in \mathbb{R}^{(k+1)M}, \\ A_e &\in \mathbb{R}^{kM\times(k+1)M}, \\ b_e &\in \mathbb{R}^{kM}. \end{aligned}$$

Inequality Constraints

Recall that the original system is (nonlinearly) boundary actuated in the form

$$h(0)h_x(0) = -\frac{1}{2}q(t), \tag{3.88}$$

and that $q(t)$ is constrained to lie within $q_{min} \leq q(t) \leq q_{max}$. A reversible[1] transformation was used to write $u(t) = h_x(0) = \frac{1}{2h(0)}q(t)$. Thus, based on the $u(t)$ signal,

[1]For $h(0) \neq 0$, and we are only interested in $1 \leq h(0) \leq 10$, where 10 and 0 correspond to initial and breakthrough column heights.

given by, for example, the backstepping controller, the value $q(t)$ was computed. One of the main strengths of optimization used for control purposes is the ability to explicitly handle constraints. However, the linear constraints on $q(t)$ translate to nonlinear constraints on $u(t)$.

A simplification resulting in conservative constraints may be used here, in the form

$$u_{min} = -\frac{q_{max}}{2h(0)_{min}} \leq u(t) \leq -\frac{q_{min}}{2h(0)_{max}} = u_{max}. \tag{3.89}$$

The values for $h(0)_{min}$ and $h(0)_{max}$ are chosen to the breakthrough column height and the initial column height, respectively.

Another constraint is the sub-critical phase constraint – we wish to keep the gas-cone small enough in size so it doesn't break through. This may be enforced by keeping the gas cone above a threshold. For simplicity, this will be the same threshold used to detect breakthrough in the other simulations – the oil-column height at the well $h(t,1) \geq 1$. The height of the oil column at the well may be found, in the linear system dynamics described in Chapter 2, as $C\mathbf{h} + Du$. The constraint may be enforced through the matrix inequality $A_i z \leq b_i$, where

$$A_i = \begin{pmatrix} 0_{1\times k} & 0_{1\times 1} & -C_{1\times k} & -D_{1\times 1} & 0 & & & \\ 0_{1\times k} & 0_{1\times 1} & 0_{1\times k} & 0_{1\times 1} & -C_{1\times k} & -D_{1\times 1}0 & & \\ & & & & \ddots & & & \\ & & & & & & 0_{1\times k} & 0_{1\times 1} & -C_{1\times k} & -D_{1\times 1} \end{pmatrix}, \tag{3.90}$$

and

$$b_i = \begin{pmatrix} -1 & \dots & -1 \end{pmatrix}^T. \tag{3.91}$$

The Quadratic Objective Function

The objective function $J = \int_{t_0}^{t_f} q(t) npv(t)\,\mathrm{d}t$ needs to be stated in the quadratic form $z^T H z + f^T z$. Since the state vector z is time-discrete, the integral may be calculated by its Riemann-sum,

$$J = \sum_{k=1}^{M} q_k \cdot npv_k \cdot \Delta t_k. \tag{3.92}$$

Recalling that $q(t) = -2\alpha\varphi h(t, L)u(t)$, we may rewrite (3.92) as

$$J = \sum_{k=1}^{M} \left(-2\alpha\varphi \left(\mathbf{Ch}_k + Du_k\right) u_k\right) \cdot npv_k \cdot \Delta t_k \tag{3.93}$$

$$J = \sum_{k=1}^{M} \left(-2\alpha\varphi \left(\mathbf{Ch}_k u_k + Du_k^2\right)\right) \cdot npv_k \cdot \Delta t_k \tag{3.94}$$

$$J = \sum_{k=1}^{M} \frac{1}{2} \begin{pmatrix} \mathbf{h}_k \\ u_k \end{pmatrix}^T H_k \begin{pmatrix} \mathbf{h}_k \\ u_k \end{pmatrix}, \tag{3.95}$$

where

$$H_k = -2\alpha\varphi \begin{pmatrix} 0 & \cdots & & npv_k\Delta t_k C_1 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 0 & npv_k\Delta t_k C_{N-2} \\ npv_k\Delta t_k C_1 & \cdots & npv_k\Delta t_k C_{N-2} & 2npv_k\Delta t_k D \end{pmatrix}. \tag{3.96}$$

The objective function may thus be written

$$J = \frac{1}{2} z^T H z, \tag{3.97}$$

where $H = diag(H_1, \ldots, H_M)$.

Convexity

As the objective function (3.97) is *not positive-definite*, the optimization problem is non-convex[2]. As a result, any solution to it may at best be classified as a local minimum, and say nothing about how good the solution is globally. In order to achieve a *somewhat global* minimum, it is possible to start the optimization problem with many different initial guesses – however, the best of a finite group of local minima is not necessarily a global minimum.

Running the optimization with an initial guess that is the trajectory of a known *good solution*, like the trajectory of the backstepping controller, or that of Sagatun's controller, will likely result in an optimized, similar trajectory. However, in order to explore new control strategies which may be better suited than the ones studied earlier, it would be necessary to explore the state space further, using random initial guesses.

Gas Breakthrough

Unlike the other control strategies, this optimal control scheme explicitly prohibits the gas cone to get too large. While other control strategies inevitably lead to gas breakthrough, that would constitute an infeasible optimization problem. In order to work

[2] H_k is indefinite, and as a result, so is H.

around this, we are faced with two options; we can increase the simulation time until no feasible solution exists, or we could fix the simulation time, and see how large a sub-critical payout is possible to get given that end-time.

For simplicity, the latter option will be chosen – a fixed simulation time, which will coincide with breakthrough times of other control strategies.

Chapter 4

Observer Design

State-feedback controllers require measurement of the state at each point in the domain (the reservoir). These measurements are, however, not available – measurements may at most be available some distance into the reservoir, but are chiefly only available at the well. The Backstepping controller is an example of such a controller, as well as any receding horizon implementation of the optimal control problem posed in the previous chapter.

In order to use such controllers in a real life scenario, when full state measurements are unavailable, state observers are necessary to design and deploy. In this chapter two observers will be designed. First, the Backstepping observer, a novel approach that works well for systems where a backstepping-style transformation may be found. It will therefore only be described for a linearized system. Secondly, an extended Kalman filter will be developed. The EKF is the state-of-the-art of nonlinear state estimation in the presence of model disturbances and measurement noise.

4.1 Backstepping Observer

The backstepping observer uses the same ideas as the backstepping controller. It will be designed for the linearized model (2.13)

$$h_t(t,x) = h_c h_{xx}(t,x), \tag{4.1}$$
$$h_x(t,0) = 0, \tag{4.2}$$
$$h_x(t,1) = u(t). \tag{4.3}$$

The observer takes the form of "copy the original system plus output injection" [18],

$$\hat{h}_t(t,x) = h_c\hat{h}_{xx}(t,x) + p_1(x)\left[h(t,1) - \hat{h}(t,1)\right], \tag{4.4}$$
$$\hat{h}_x(t,0) = 0, \tag{4.5}$$
$$\hat{h}_x(t,1) = u(t) - p_{10}\left[h(t,1) - \hat{h}(t,1)\right]. \tag{4.6}$$

In order to analyze and design it, we consider the estimation error $e(t,x) = h(t,x) - \hat{h}(t,x)$. The error signal $e(t,x)$ is governed by the PDE

$$e_t(t,x) = h_c e_{xx}(t,x) - p1(x)e(t,1), \tag{4.7}$$
$$e_x(t,0) = 0, \tag{4.8}$$
$$e_x(t,1) = p_{10}e(t,1). \tag{4.9}$$

Since the goal of an observer is to mimic the state of the system, i.e. $\hat{h} \to h$, stability and fast convergence of $e(t,x)$ are desirable. This can be achieved by designing $p_1(x)$ and p_{10} such that (4.7)–(4.9) is exponentially stable.

This will be achieved by using the transformation

$$e(t,x) = w(t,x) - \int_x^1 p(x,y)w(t,y)\,\mathrm{d}y \tag{4.10}$$

to transform the error system (4.7)–(4.9) into the *exponentially stable* (for large c)

$$w_t(t,x) = h_c w_{xx}(t,x) - cw(t,x), \tag{4.11}$$
$$w_x(t,0) = 0, \tag{4.12}$$
$$w_x(t,1) = 0. \tag{4.13}$$

In order to find the $p(x,y)$ that does the transformation, we need to match (4.10)

with (4.7)–(4.9). First, differentiating (4.10) with respect to time yields

$$\begin{aligned}
e_t(t,x) &= w_t(t,x) - \int_x^1 p(x,y) w_t(t,y)\,\mathrm{d}y \\
&= h_c w_{xx}(t,x) - cw(t,x) - \int_x^1 p(x,y)\Big(h_c w_{yy}(t,y) - cw(t,y)\Big)\,\mathrm{d}y \\
&= h_c w_{xx}(t,x) - cw(t,x) - h_c p(x,y) w_y(t,y)\Big|_{y=x}^1 + h_c \int_x^1 p_y(x,y) w_y(t,y)\,\mathrm{d}y \\
&+ \int_x^1 cp(x,y) w(t,y)\,\mathrm{d}y \\
&= h_c w_{xx}(t,x) - cw(t,x) - h_c p(x,y) w_x(t,x) + h_c \int_x^1 p_y(x,y) w_y(t,y)\,\mathrm{d}y \\
&+ \int_x^1 cp(x,y) w(t,y)\,\mathrm{d}y \\
&= h_c w_{xx}(t,x) - cw(t,x) - h_c p(x,y) w_x(t,x) + h_x p_y(x,y) w(t,y)\Big|_{y=x}^1 \\
&- h_c \int_x^1 p_{yy}(x,y) w(t,y)\,\mathrm{d}y + \int_x^1 cp(x,y) w(t,y)\,\mathrm{d}y \\
&= h_c w_{xx}(t,x) - cw(t,x) - h_c p(x,y) w_x(t,x) + h_c p_y(x,y) w(t,1) \\
&- \int_x^1 w(t,y)\Big(h_c p_{yy}(x,y) - cp(x,y)\Big)\,\mathrm{d}y.
\end{aligned} \tag{4.14}$$

In order to differentiate (4.10) with respect to x, we need to use the Leibniz integral rule, which takes the form

$$\frac{\mathrm{d}}{\mathrm{d}x}\int_x^1 f(x,y)\,\mathrm{d}y = -f(x,x) + \int_x^1 \frac{\partial}{\partial x} f(x,y)\,\mathrm{d}y. \tag{4.15}$$

Then we get

$$e_x(t,x) = w_x(x) + p(x,x)w(t,x) - \int_x^1 p_x(x,y) w(t,y)\,\mathrm{d}y \tag{4.16}$$

and

$$\begin{aligned}
e_{xx}(t,x) &= w_{xx}(t,x) + w(x)\frac{\mathrm{d}}{\mathrm{d}x}p(x,x) + p(x,x)w_x(t,x) + p_x(x,x)w(t,x) \\
&- \int_x^1 p_{xx}(x,y) w(t,y)\,\mathrm{d}y,
\end{aligned} \tag{4.17}$$

where $\frac{\mathrm{d}}{\mathrm{d}x}p(x,x) = p_x(x,x) + p_y(x,x)$.

Next, substituting (4.14) and (4.17) into (4.7), and noting that $e(t,1) = w(e,1)$, we get

$$\begin{aligned}
-p_1(x)w(t,1) &= e_t(t,x) - h_c e_{xx}(t,x) \\
&= h_c w_{xx}(t,x) - cw(t,x) + h_c p(x,x)w_x(t,x) + h_c p_y(x,1) - h_c p_y(x,x)w(t,x) \\
&\quad - \int_x^1 w(t,y)\Big(h_c p_{yy}(x,y) - cp(x,y)\Big)\,\mathrm{d}y \\
&\quad - h_c\Bigg[w_{xx}(t,x) + w(t,x)\frac{\mathrm{d}}{\mathrm{d}x}p(x,x) + w_x(t,x)p(x,x) + p_x(x,x)w(t,x) \\
&\quad - \int_x^1 p_{xx}(x,y)w(t,y)\,\mathrm{d}y\Bigg] \\
&= w(t,x)\Big(-c - h_c p_y(x,y) - h_c p_x(x,y) - h_c\frac{\mathrm{d}}{\mathrm{d}x}p(x,x)\Big) \\
&\quad - \int_x^1 w(t,y)\Big(h_c p_{yy}(x,y) - h_c p_{xx}(x,y) - cp(x,y)\Big)\,\mathrm{d}y + p_y(x,y)w(t,1).
\end{aligned} \tag{4.18}$$

Matching the left and right sides and setting $p_1(x) = p_y(x,1)$, we note that the other two terms need to be exactly zero. We arrive at the hyperbolic PDE

$$h_c p_{yy}(x,y) - h_c p_{xx}(x,y) = cp(x,y) \tag{4.19}$$

and the condition

$$2h_c\frac{\mathrm{d}}{\mathrm{d}x}p(x,x) = -c. \tag{4.20}$$

Integrating (4.20), we get

$$p(x,x) = \int_0^x -\frac{c}{2h_c}\,\mathrm{d}\xi = -\frac{c}{2h_c}x. \tag{4.21}$$

Next, by matching (4.16) with the boundary condition (4.8) we get

$$e_x(t,0) = w_x(t,0) + p(0,0)w(t,0) - \int_0^1 p_x(0,y)w(t,y)\,\mathrm{d}y \tag{4.22}$$

which, since $w_x(t,0) = 0$, $e_x(t,0) = 0$ and $p(0,0) = 0$, gives us the boundary condition

$$p_x(0,y) = 0. \tag{4.23}$$

Further, matching (4.16) with the boundary condition (4.9) gives us

$$p_{10}e(t,1) = p(1,1)e(t,1), \tag{4.24}$$

which means

$$p_{10} = p(1,1). \tag{4.25}$$

Now, all that needs to be done in order to find the observer is to find $p(x,y)$ that satisfies (4.19),(4.21) and (4.23). Using the change of variables $\tilde{x} = y$, $\tilde{y} = x$ and $\tilde{p}(\tilde{x},\tilde{y}) = p(x,y)$, the observer kernel PDE is transformed to

$$h_c\tilde{p}_{\tilde{x}\tilde{x}}(\tilde{x},\tilde{y}) - h_c\tilde{p}_{\tilde{y}\tilde{y}}(\tilde{x},\tilde{y}) = c\tilde{p}_(\tilde{x},\tilde{y}) \tag{4.26}$$

$$\tilde{p}_{\tilde{y}}(\tilde{x},0) = 0 \tag{4.27}$$

$$\tilde{p}(\tilde{x},\tilde{x}) = \frac{c}{2h_c}\tilde{x}. \tag{4.28}$$

The solution to (4.26)–(4.28) has been found in [18], and is given by

$$\tilde{p}(\tilde{x},\tilde{y}) = -\frac{c}{2h_c}\tilde{x}\frac{I_1\left(\sqrt{\frac{c}{2h_c}(\tilde{x}^2-\tilde{y}^2)}\right)}{\sqrt{\frac{c}{2h_c}(\tilde{x}^2-\tilde{y}^2)}}, \tag{4.29}$$

where $I_1(x)$ is a *modified Bessel function* [18, Appendix A.2].

Now what remains is to find the observer kernel

$$p_1(x) = \tilde{p}_{\tilde{x}}(1,\tilde{y}) \tag{4.30}$$

and the constant

$$p_{10} = \tilde{p}(1,1). \tag{4.31}$$

However, p_{10} is not directly computable from (4.31), as (4.29) is not defined at $(x,y) = (1,1)$. Instead, the limit value as $(x,y) \to (1,1)$ will be used. Since the kernel $p(x,y)$ is defined at that point, and (4.29) is the solution, the limit value should be the same regardless of the approach direction. Therefore it will now be computed from only one such direction. We begin by defining

$$\tilde{p}(1,\tilde{y}) = -\lambda\tilde{y}\frac{I_1\Big(f(\tilde{y})\Big)}{f(\tilde{y})}, \tag{4.32}$$

with $\lambda = \frac{c}{h_c}$ and $f(\tilde{y}) = \sqrt{\lambda(1-\tilde{y}^2)}$. Then, the limit may be taken as

$$\begin{aligned}
\lim_{\tilde{y}\to 1}\tilde{p}(1,\tilde{y}) &= -\lambda\lim_{\tilde{y}\to 1}\frac{\tilde{y}I_1\Big(f(\tilde{y})\Big)}{f(\tilde{y})} \\
&\overset{l'H}{=} -\lambda\lim_{\tilde{y}\to 1}\frac{I_1\Big(f(\tilde{y})\Big) + \frac{\tilde{y}}{2}\left(I_0\Big(f(\tilde{y})\Big) - I_2\Big(f(\tilde{y})\Big)\right)f'(\tilde{y})}{f'(\tilde{y})} \\
&= -\frac{\lambda}{2}I_0\Big(f(1)\Big) \\
&= -\frac{\lambda}{2},
\end{aligned} \tag{4.33}$$

where *l'Hôpital's rule* is used as well as some properties of the Bessel functions given in [18, Appendix A.2].

The observer kernel may be found from (4.30) by differentiating $\tilde{p}(\tilde{x},\tilde{y})$ with respect to $\tilde{x}$:

$$\tilde{p}_{\tilde{x}}(\tilde{x},\tilde{y}) = -\lambda\left(\frac{I_1\left(\sqrt{\lambda(\tilde{x}^2-\tilde{y}^2)}\right)}{\sqrt{\lambda(\tilde{x}^2-\tilde{y}^2)}} + x^2\frac{I_2\left(\sqrt{\lambda(\tilde{x}^2-\tilde{y}^2)}\right)}{\tilde{x}^2-\tilde{y}^2}\right). \tag{4.34}$$

Now, from (4.30) we have

$$p_1(x) = -\lambda\left(\frac{I_1\left(\sqrt{\lambda(1-x^2)}\right)}{\sqrt{\lambda(1-x^2)}} + \frac{I_2\left(\sqrt{\lambda(1-x^2)}\right)}{1-x^2}\right). \tag{4.35}$$

Now the backstepping observer (4.4)–(4.6) is derived, with p_{10} and $p_1(x)$ given by (4.33) and (4.35), with a free parameter, c, which may be used to control the convergence rate.

4.2 Kalman Filter

In this section a Kalman Filter for the linear model presented in Section 2.2 will be discussed. The material is taken from [28]. Let us re-state the process and measurement model. In the Kalman Filter formulation, the system model is given by

$$\dot{\mathbf{h}} = A\mathbf{h} + Bu + G\mathbf{w} \tag{4.36}$$

and the measurement by

$$y = C\mathbf{h} + Du + v, \tag{4.37}$$

where $\mathbf{w}$ is a white noise vector representing process disturbances and model uncertainties, and v is a white noise scalar representing measurement noise. The power of $\mathbf{w}$ and that of v is given by $E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{Q}\delta(t-\tau)$ and $E[v(t)v(\tau)] = R\delta(t-\tau)$, respectively.

The Kalman Filter observer will be given by

$$\dot{\hat{h}} = A\hat{\mathbf{h}} + Bu + K\left(y - C\hat{\mathbf{h}} - Du\right), \tag{4.38}$$

where K, the *Kalman Gain*, is given by

$$K = \frac{PC^T}{R}, \tag{4.39}$$

and P is given by the matrix riccati equation

$$\dot{P} = AP + PA^T - \frac{PC^TCP}{R} + G\mathbf{Q}G. \tag{4.40}$$

For simplicity, the *stationary kalman filter* will be used in this work. The stationary kalman filter is achieved by calculating $P(t \to \infty)$, which is possible to find by setting $\dot{P} = 0$ in (4.40).

Since $\mathbf{Q}$ and R represent model and measurement uncertainty, they may be used to tune the Kalman Filter.

Extended Kalman Filter

As the process equations are really nonlinear, the Kalman filter described above will not be able to predict the real state of the system. The extended Kalman filter approach solves this by linearizing at every time-step and thus a linear filter will be used around an accurate linearization point at each step.

The nonlinear ODE model derived in Section 2.2 may be written, in Kalman filter notation, as

$$\dot{\mathbf{h}} = f(\mathbf{h}, u, \mathbf{w}), \tag{4.41}$$

$$y = C\mathbf{h} + Du + v. \tag{4.42}$$

At each time-step, the Jacobian of f will be calculated at the estimated state. Then, the matrices A and b in the state space representation will be replace by

$$A(t) = \left.\frac{\partial f(\mathbf{h}, u)}{\partial \mathbf{h}}\right|_{\hat{\mathbf{h}}(t), u(t)} \tag{4.43}$$

$$b(t) = \left.\frac{\partial f(\mathbf{h}, u)}{\partial u}\right|_{\hat{\mathbf{h}}(t), u(t)}. \tag{4.44}$$

The covariance matrix P and the Kalman gain K are computed as before, however, the state observer will take the nonlinear form

$$\dot{\hat{h}} = f(\hat{\mathbf{h}}, u, \mathbf{w}) + K\left(y - C\hat{\mathbf{h}} - Du\right). \tag{4.45}$$

Since we have no knowledge of the process disturbances or model uncertainties, we may continue to assume a constant G. This is the continuous-time extended Kalman filter. More details may be found in [29].

In Section 2.2, it was shown that

$$f_i(\mathbf{h}, u) = h_i \frac{h_{i-1} - 2h_i + h_{i+1}}{\Delta x^2} + \frac{h_{i11}^2 - 2h_{i-1}h_{i+1} + h_{i+1}^2}{4\Delta x^2}, \text{ for } i = 2..N-3. \tag{4.46}$$

Written out, it may be simplified to

$$f_i(\mathbf{h}, u) = \frac{1}{\Delta x^2}\left(h_i h_{i-1} - 2h_i^2 + h_i h_{i+1} + \frac{h_{i-1}^2 - 2h_{i-1}h_{i+1} + h_{i+1}^2}{4}\right), \tag{4.47}$$

and thus the derivative is

$$\left.\frac{\partial f_i(\Delta\mathbf{h},\Delta u)}{\partial\mathbf{h}}\right|_{\hat{\mathbf{h}}} = \frac{1}{\Delta x^2}\Bigg((\hat{h}_i - \frac{1}{2}\hat{h}_{i+1} + \frac{1}{2}\hat{h}_{i-1})\Delta h_{i-1} + (\hat{h}_{i-1} - 4\hat{h}_i + \hat{h}_{i+1})\Delta h_i \\ +(\hat{h}_i + \frac{1}{2}\hat{h}_{i+1} - \frac{1}{2}\hat{h}_{i-1})\Delta h_{i+1}\Bigg), \tag{4.48}$$

for $i = 2..N-3$.

As for the boundary conditions, they may be enforced by substituting (2.29) and (2.30) into f_1 and f_{N-2}. The equations become

$$f_1(\mathbf{h},u) = \frac{1}{9\Delta x^2}\left(4h_2^2 - 2h_1h_2 - 2h_1^2\right) \tag{4.49}$$

for the left boundary, and

$$f_{N-2}(\mathbf{h},u) = \frac{u^2}{9} + \frac{1}{9\Delta x^2}\left(-2h_{N-2}^2 + 4h_{N-3}^2 + 10h_{N-2}u - 4h_{N-3}u - 2h_{N-2}h_{N-3}\right) \tag{4.50}$$

for the right boundary.

The derivatives are

$$\left.\frac{\partial f_1(\Delta\mathbf{h},\Delta u)}{\partial\mathbf{h}}\right|_{\hat{\mathbf{h}}} = \frac{1}{9\Delta x^2}\Big((-4\hat{h}_1 - 2\hat{h}_2)\Delta h_1 + (8\hat{h}_2 - 2\hat{h}_1)\Delta h_2\Big), \tag{4.51}$$

and

$$\left.\frac{\partial f_{N-2}(\Delta\mathbf{h},\Delta u)}{\partial\mathbf{h}}\right|_{\hat{\mathbf{h}},u^*} = \frac{1}{9\Delta x^2}\Big((-4\hat{h}_{N-2} + 10u^* - 2\hat{h}_{N-3})\Delta h_{N-2} \\ +(8\hat{h}_{N-3} - 4u^* - 2\hat{h}_{N-2})\Delta h_{N-3}\Big). \tag{4.52}$$

The derivative with respect to u is

$$\left.\frac{\partial f_{N-2}(\Delta\mathbf{h},\Delta u)}{\partial u}\right|_{\hat{\mathbf{h}},u^*} = \Big(\frac{2}{9}u^* + \frac{10}{9\Delta x^2}\hat{h}_{N-2} - \frac{4}{9\Delta x^2}\hat{h}_{N-3}\Big)\Delta u, \tag{4.53}$$

and $\frac{\partial f_i}{\partial u} = 0$ for all $i \neq N-2$.

The extended Kalman filter will, then, at each time-step, use the system matrices (4.43), (4.44), which have been computed above, in order to calculate the Kalman gain.

$$A(t) = \frac{1}{\Delta x^2}\begin{pmatrix} -\frac{4\,\hat{h}_1}{9} - \frac{2\,\hat{h}_2}{9} & \frac{8\,\hat{h}_2}{9} - \frac{2\,\hat{h}_1}{9} & 0 & 0 & 0 \\ \frac{\hat{h}_1}{2} + \hat{h}_2 - \frac{\hat{h}_3}{2} & \hat{h}_1 - 4\,\hat{h}_2 + \hat{h}_3 & \hat{h}_2 - \frac{\hat{h}_1}{2} + \frac{\hat{h}_3}{2} & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \frac{\hat{h}_{N-4}}{2} + \hat{h}_{N-3} - \frac{\hat{h}_{N-2}}{2} & \hat{h}_{N-4} - 4\,\hat{h}_{N-3} + \hat{h}_{N-2} & \hat{h}_{N-3} - \frac{\hat{h}_{N-4}}{2} + \frac{\hat{h}_{N-2}}{2} \\ 0 & 0 & 0 & \hat{h}_{N-3} - \frac{2\,\hat{h}_{N-2}}{9} - \frac{4\,u}{9} & \frac{10\,u}{9} - \frac{4\,\hat{h}_{N-2}}{9} - \frac{2\,\hat{h}_{N-3}}{9} \end{pmatrix} \tag{4.54}$$

$$b(t) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{2}{9}u^* + \frac{1}{9\Delta x^2}\left(10\hat{h}_{N-2} - 4\hat{h}_{N-3}\right) \end{pmatrix} \tag{4.55}$$

Chapter 5

Results

In this chapter simulation results will be presented. Unless stated otherwise, simulations are run on the full, nonlinear PDE model (2.9). First, performance of the controllers will be presented. Then the observers' performance will be shown and lastly, for the state-feedback class controllers, their performance when used in combination with an observer will be presented.

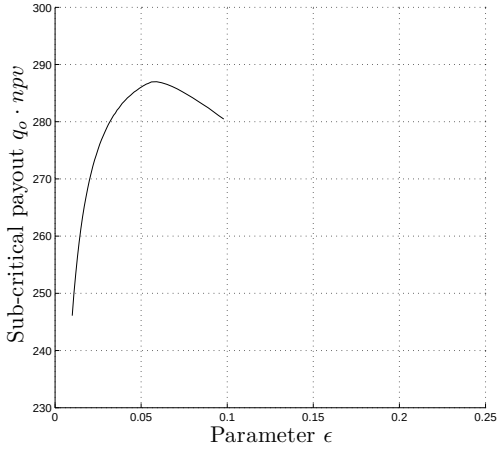
In the controller simulations, the simulation will stop once gas-breakthrough occurs, or 10 years have passed.

5.1 Controller performance

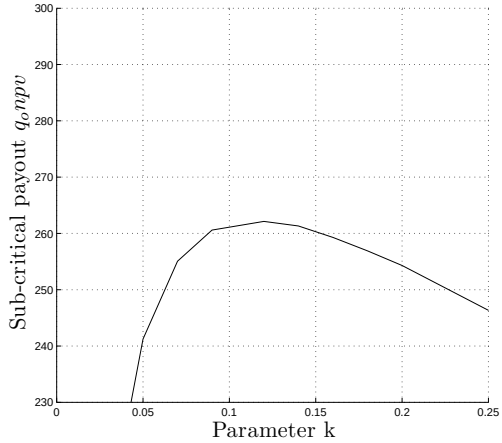
Generalized Sagatun $\&$ Backstepping controllers

The system (3.2)–(3.4) was simulated using the Backstepping controller as well as controller (3.72). When simulating with the latter, the parameter a was varied from $a = 1$ to $a = 10$. The case $a = 2$ corresponds to the controller studied in [7]. Each controller has a free parameter – k for (3.72) and ϵ for the backstepping controller. This parameter was varied in search of an optimal *sub-critical payout*, given by (3.1).

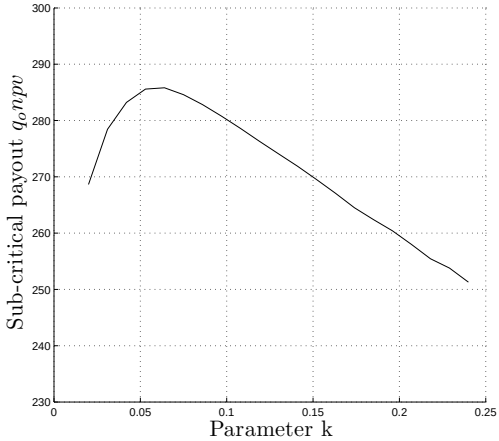
Figure 5.1 shows the *sub-critical payout* as a function of the controller parameter ϵ for the backstepping controller, and the parameter k for controller (3.72). In the latter, three cases are shown for parameter a: $a = 1$, $a = 2$, and $a = 3$. Table 5.1 summarizes the results, showing the maximum attained sub-critical payout, using the optimal k or ϵ value. The values in Table 5.1 correspond to the peaks in Figure 5.1. Figure 5.2 shows the optimal sub-critical payout for controller (3.72) over a range of the parameter a. In each case, the parameter k has been varied and the peak has been found.



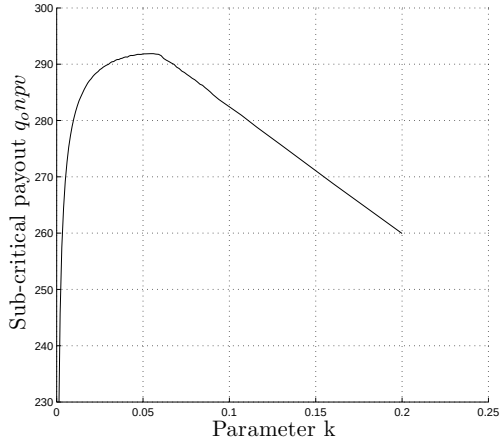
(a) Backstepping Controller



(b) Controller (3.72), $a = 1$



(c) Controller (3.72), $a = 2$



(d) Controller (3.72), $a = 3$

Figure 5.1: Net present value of sub-critical production, $q(t) \cdot npv(t)$ as a function of the controller parameter. The parameter is k for controller (3.72) and ϵ for the backstepping controller.

Controller	*Maximum payout* $q(t) \cdot npv(t)$
Backstepping Controller	287.0
Controller (3.72), $a = 1.0$	262.1
Controller (3.72), $a = 2.0$. Controller from [7].	285.8
Controller (3.72), $a = 3.0$	291.9
Controller (3.72), $a = 4.0$	293.4
Controller (3.72), $a = 5.0$	294.1
Controller (3.72), $a = 6.0$	294.6
Controller (3.72), $a = 7.0$	294.9
Controller (3.72), $a = 8.0$	295.1
Controller (3.72), $a = 9.0$	295.2
Controller (3.72), $a = 10.0$	295.3

Table 5.1: Summary of Results for backstepping and generalized Sagatun controllers.

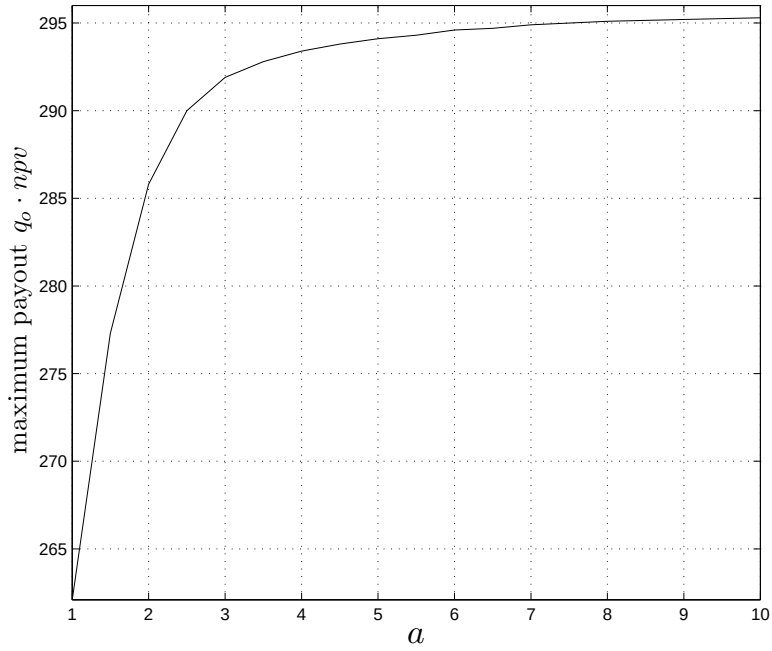


Figure 5.2: Maximum sub-critical payout $q_o \cdot npv$ when using the controller $q(t) = 2kh^a(t, 1)$. Maximum sub-critical payout seems to grow logarithmically with a, flatting out at approximately $a = 6$.

Figure 5.4 shows the trajectory of one half of the reservoir under depletion, using the Backstepping controller. The trajectory is similar when using the other controller. Figure 5.4(b) shows the profile for increasing t in a 2D plot. Noting that the profile is symmetric about $x = 0$, the gas cone is evident from that figure. Figure 5.3 is the production trajectory when using the Backstepping controller. Note the saturation at the top in Figure 5.3 due to constraints imposed on the oil flux. Again, the production profile is similar when using the other controller.

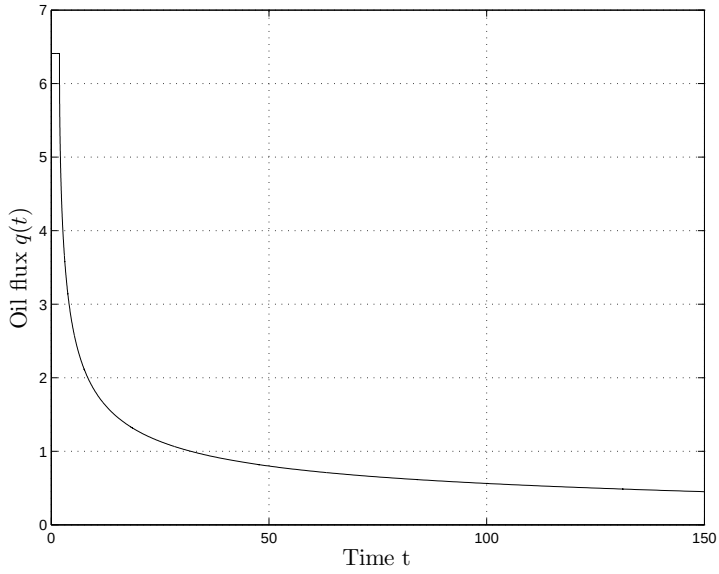
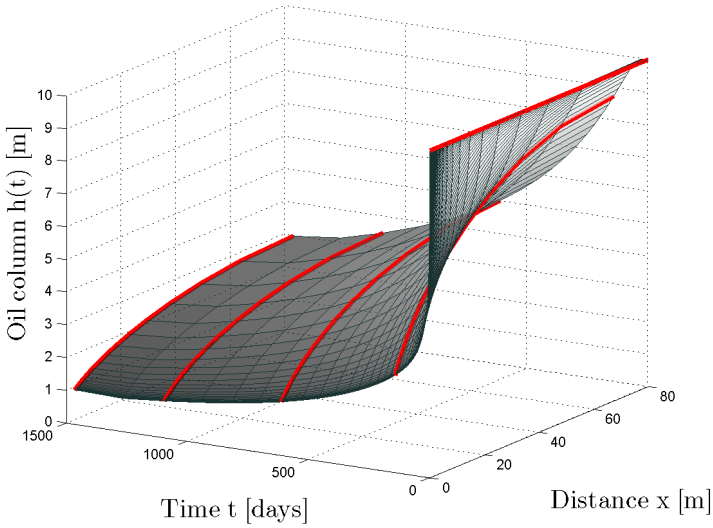


Figure 5.3: Dimensionless oil flux $q(t)$, first 150 days of production, using the backstepping controller.

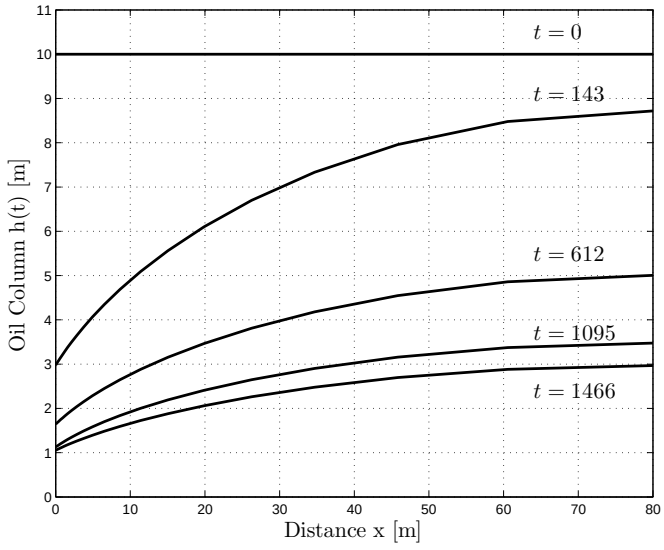
Mini-Maxi Controller

Section 3.5 outlined a control strategy where the production rate would start off at a maximum, to be turned down to the minimum allowable in one step at a predetermined time. This predetermined time-instance would be an optimization parameter for the controller. When simulating that strategy it became evident that running the well at maximum capacity, gas-breakthrough would occur at a very early stage (less than one day).

Instead of starting off at maximum capacity, then, the well was started off at a nominal rate suggested by [7], and then switched to minimum capacity at $t = T_{switch}$. Figure 5.5 shows sub-critical payout plotted against T_{switch}, the time at which the production is switched from the *nominal* value to the minimal rate.



(a)



(b)

Figure 5.4: Gas-Oil contact profile, for one half of the reservoir. The well is drilled along $x = 0$. In (a), the first axis (x) goes from the well at $x = 0$ and towards the edge of the reservoir. The second axis shows time in days, and the third axis shows the oil column height. The profile is symmetric about the well. The red lines on the surface are constant x values, illustrating the curve down towards $x = 0$ at every point in time. The GOC drops most rapidly at $x = 0$, which is the well, and then the rest of the reservoir follows. This is the gas cone. (b) shows the GOC profiles that are the red curves in (a).

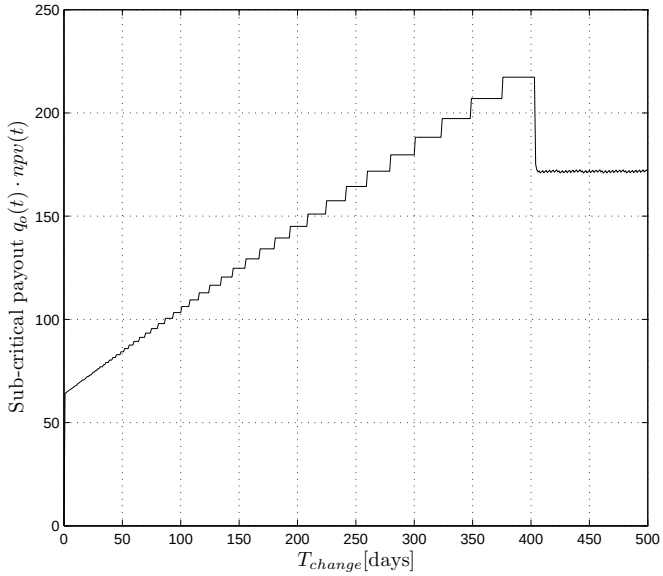


Figure 5.5: Sub-critical payout vs. time of switching from high production rate to a low one. Note the sharp drop at $T \approx 400$. At this point, the switching never occurs, because gas breakthrough happens before the switch was scheduled.

5.2 Optimal Control

The open-loop optimization problem described in Section 3.6.1 was solved using MATLAB's `quadprog`. The initial guess was a system trajectory generated by a constant oil flux, however the solver converged to the same solution when allowed to pick its own starting point. The trajectory generated by the optimization solver is shown in Figure 5.6, with the optimal oil flux q and the linear system input u shown in Figures 5.7 and 5.8, respectively. **This trajectory achieves a NPV of sub-critical production of 339**, in 700 days – much higher than the 'classical' control laws presented above, who also take a longer time to reach their optimum payout – upward of 1000 days. On my 2GHz core duo laptop, this optimization takes approximately 2 minutes.

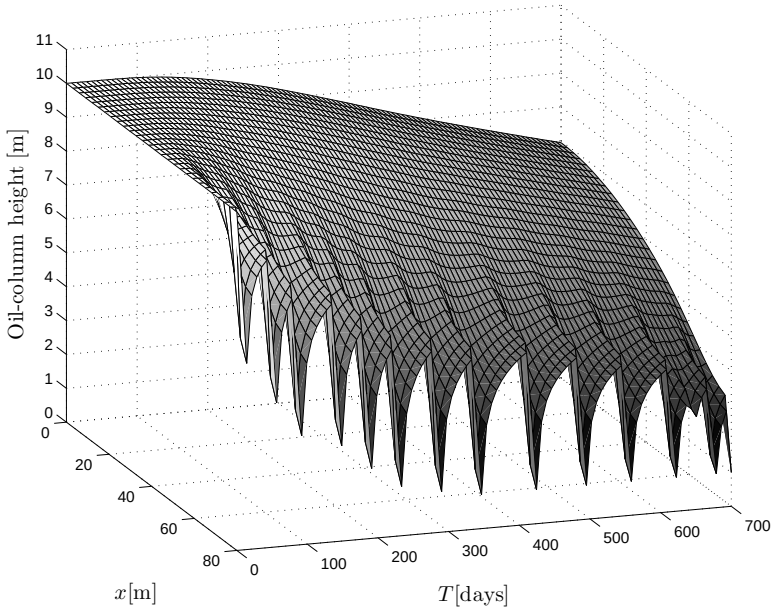


Figure 5.6: Optimal trajectory generated by quadratic programming optimization. The well is drilled along $x = 80$. The first axis (x) goes from the well at $x = 80$ and towards the edge of the reservoir. The second axis shows time in days, and the third axis shows the oil column height. The profile is symmetric about the well.

Notice how close to the constraints the input operates. This demonstrates the main advantage of optimal control – where the optimal trajectory lies near the boundary of the feasible region. It also makes it clear that, using a nonlinear model and less conservative bounds on u, better results are attainable.

The same oil flux was used as an input to the full, nonlinear PDE model. The resulting system trajectory is shown in Figure 5.9. Notice how the well flats out at around day 400 – this is due to gas breakthrough. While the linearized model exhibited no gas

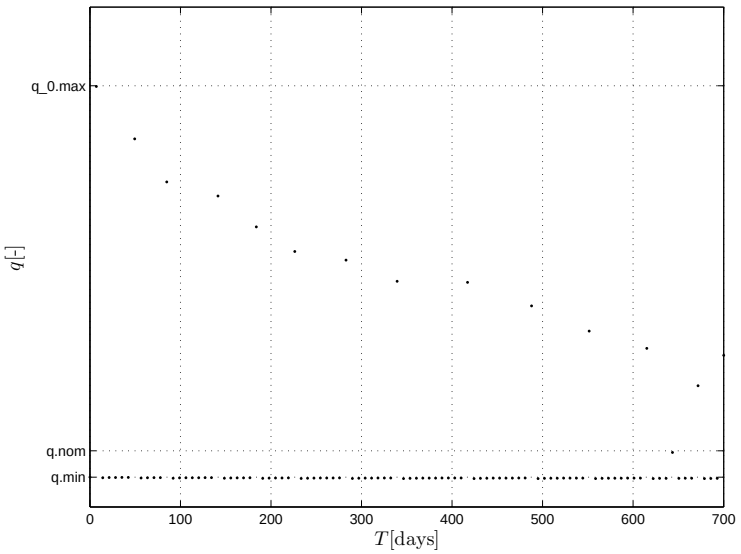


Figure 5.7: Optimal oil flux q generated by quadratic programming optimization. In the second axis, q0.min and q0.max correspond to minimum stable production rate and the maximum production rate due to friction.

breakthrough using this 'optimal' oil flux, when using the more accurate, nonlinear model, the well goes super-critical.

The model uncertainty presented by linearization was great enough to cause the well to go super-critical when simulating on a more accurate model. Rather than optimizing on a nonlinear model, it is possible to account for the model uncertainties by using more conservative bounds for the system states. As it is evident that in the nonlinear model the oil-column will drop faster, we may apply stricter bounds for the linear model, thus hoping that the nonlinear model, when simulated, will at least adhere to less strict bounds.

Followingly, a heuristic lower boundary on the oil-column height at the well, $h(t, 1) \geq 3$ was imposed when simulating the linear model[1]. The optimal trajectory generated by the solver is shown in Figure 5.10, with the corresponding q and u in Figures 5.11 and 5.12, respectively. The response of the nonlinear model to this optimal trajectory is shown in Figure 5.13.

This trajectory, with the more conservative lower bounds, **achieves a net present value of sub-critical production of 326,** slightly less than using the non-conservative constraints, but still considerably higher than both the backstepping and Sagatun's controllers.

[1]The original, non-conservative boundary was $h(t, 1) \geq 1$.

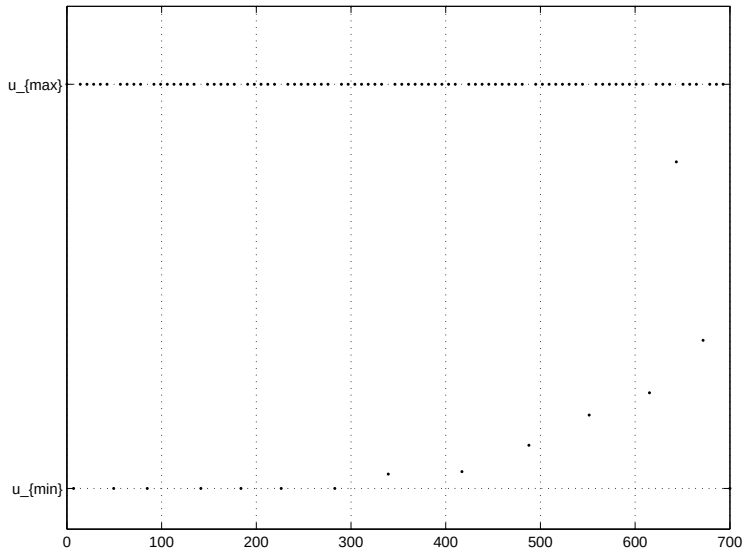


Figure 5.8: Optimal control variable u generated by quadratic programming optimization.

The cumulative payout is shown in Figure 5.14. The graph suggests that it is in the brief bursts of high-rate production that most of the value is accumulated, while almost no value is gained during the low production rate periods, which allow the cone to settle.

5.2.1 Summary

Control Strategy	*Sub-critical payout* $q(t) \cdot npv(t)$	*Improvement* [%]
Sagatun's Controller	285.8	00.00%
Backstepping Control	287.0	00.04%
Generalized Sagatun	295.3	03.32%
Quadratic Optimal Control	326.0	14.07%

Table 5.2: Summary of controller performance evaluation.

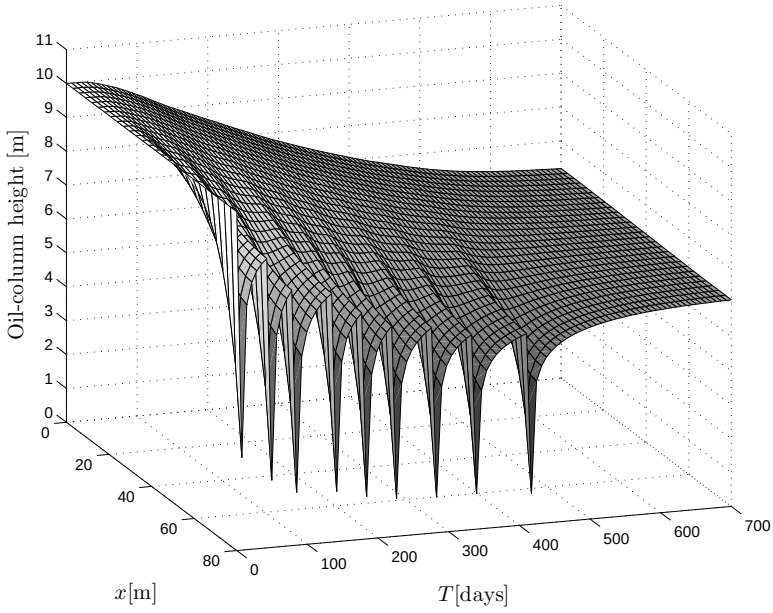


Figure 5.9: Trajectory of nonlinear system responding to optimal q, shown in Figure 5.7. The well is drilled along $x = 80$. The first axis (x) goes from the well at $x = 80$ and towards the edge of the reservoir. The second axis shows time in days, and the third axis shows the oil column height. The profile is symmetric about the well.

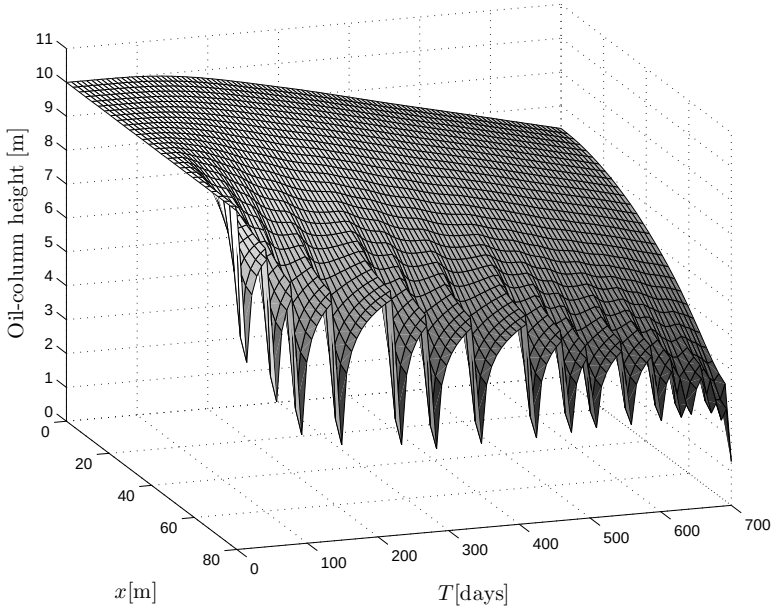


Figure 5.10: Optimal trajectory generated by quadratic programming optimization with conservative lower constraints. The well is drilled along $x = 80$. The first axis (x) goes from the well at $x = 80$ and towards the edge of the reservoir. The second axis shows time in days, and the third axis shows the oil column height. The profile is symmetric about the well.

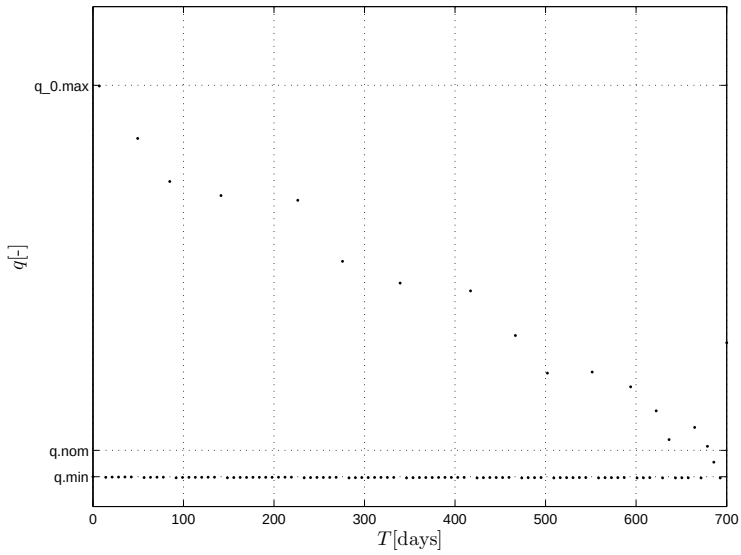


Figure 5.11: Optimal oil flux q generated by quadratic programming optimization with conservative lower constraints. In the second axis, q0.min and q0.max correspond to minimum stable production rate and the maximum production rate due to friction.

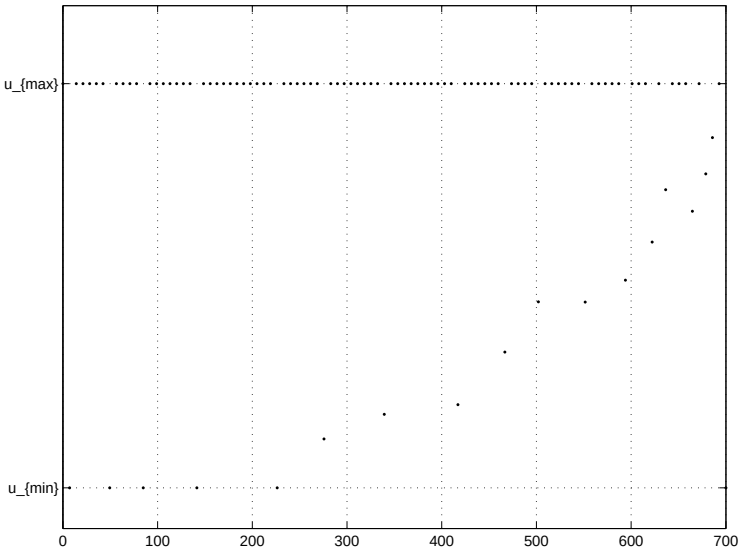


Figure 5.12: Optimal control variable u generated by quadratic programming optimization with conservative lower constraints.

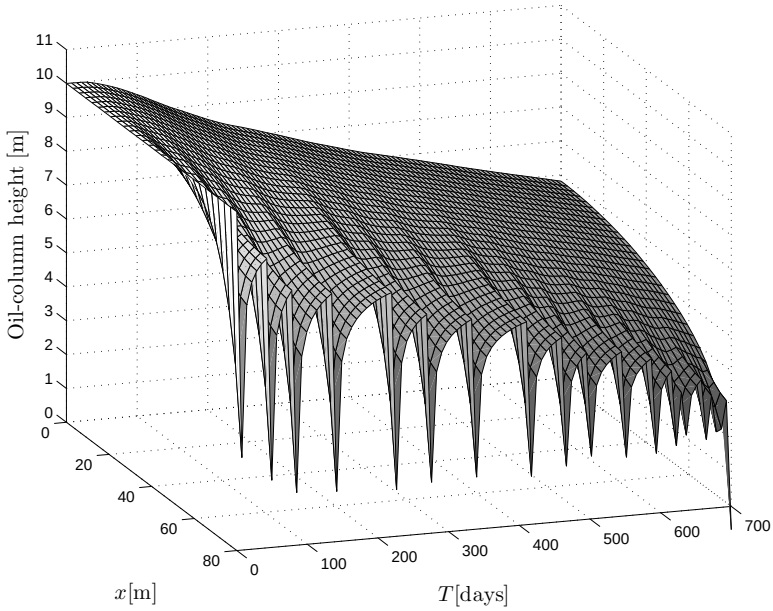


Figure 5.13: Trajectory of nonlinear system responding to optimal q, shown in Figure 5.11. The well is drilled along $x = 80$. The first axis (x) goes from the well at $x = 80$ and towards the edge of the reservoir. The second axis shows time in days, and the third axis shows the oil column height. The profile is symmetric about the well.

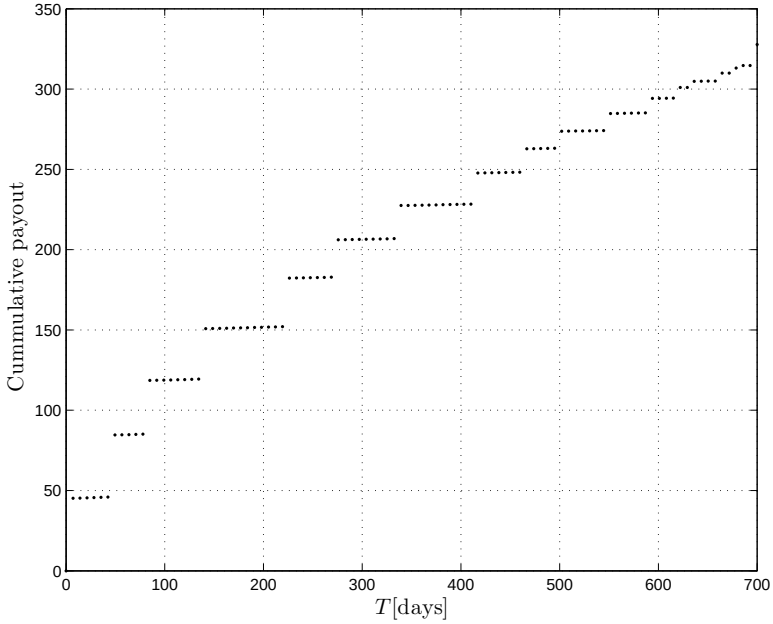


Figure 5.14: Cumulative NPV of payout in the sub-critical phase, generated by solving quadratic programming problem with conservative constraints. The corresponding oil flux $q(t)$ is shown in Figure 5.11

5.3 Observer Performance

Backstepping Observer

The backstepping observer described in Section 4.1 was simulated alongside the linear PDE. The observer kernel $p_1(x)$ takes the form shown in Figure 5.15.

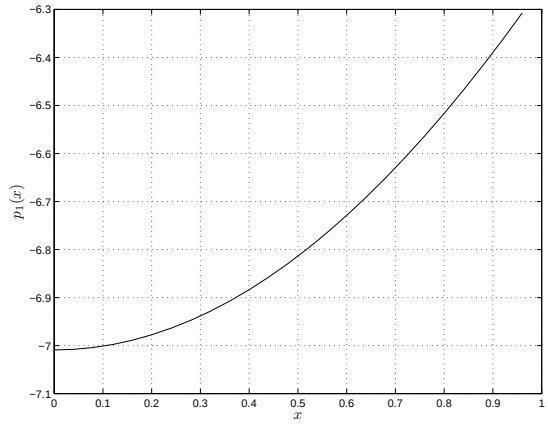


Figure 5.15: Backstepping Observer kernel function $p_1(x)$.

In the simulations, the state estimates were initiated as a random noise around the real state, as well as a constant error from the real state, in two different scenarios. Figure 5.16 shows the state as well as the estimate for the first case, while Figure 5.17 shows the same for the constant initial error case. Figures 5.18 and 5.19 show the corresponding squared error $e^2(t, x)$.

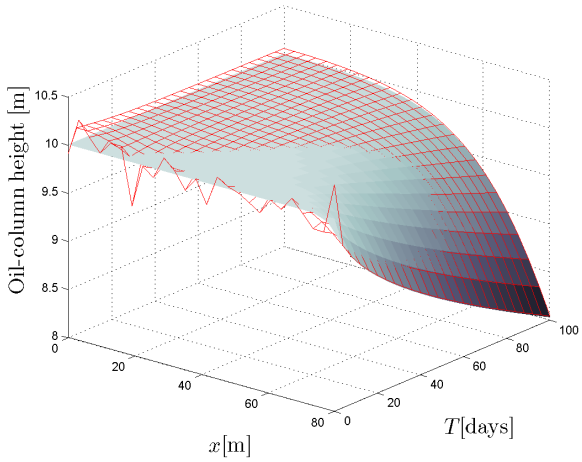


Figure 5.16: State and estimate when using Backstepping observer. Initial estimate is real state plus noise. The red mesh is the estimated state, while the blue surface is the real state.

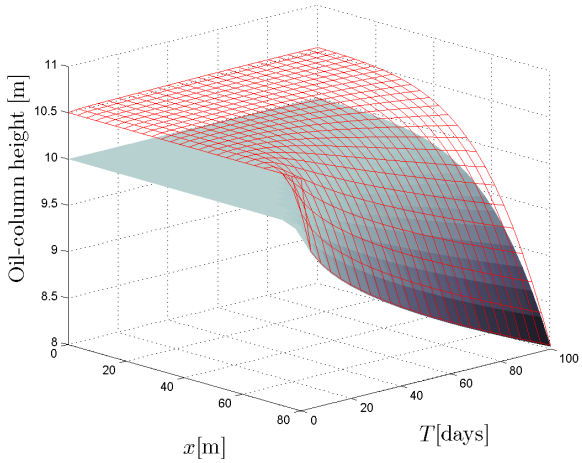


Figure 5.17: State and estimate when using Backstepping observer. Initial estimate is real state plus constant error. The red mesh is the estimated state, while the blue surface is the real state.

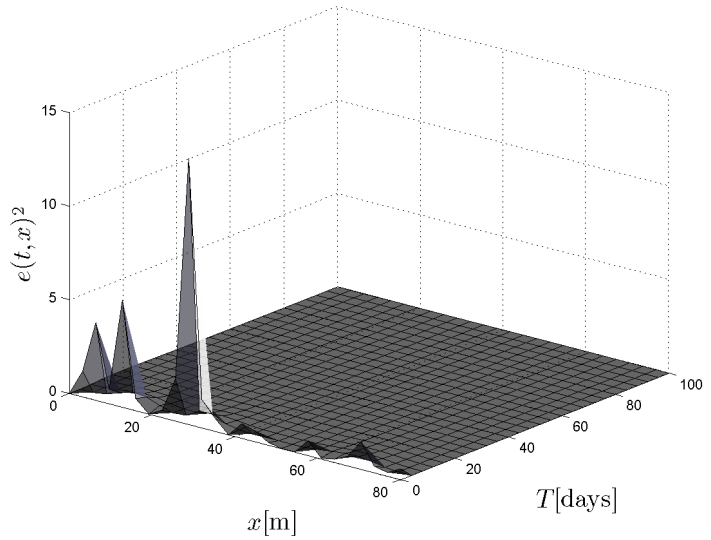


Figure 5.18: Squared estimation error when using backstepping observer. Initial error is noise.

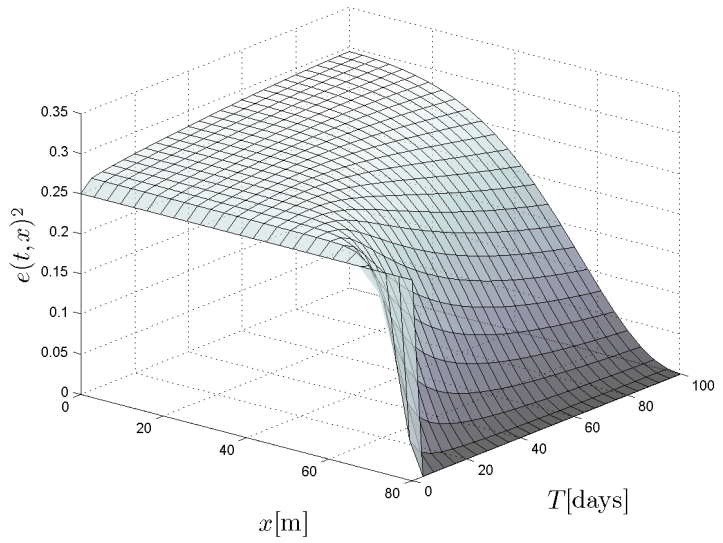


Figure 5.19: Squared estimation error when using backstepping observer. Initial error is a constant.

Extended Kalman Filter

The EKF observer was simulated together with the nonlinear ODE model. While the well-reservoir plant was discretized at $N = 26$ spacial points, the EKF was only of order $N = 8$, because the pair (A, C) becomes almost unobservable, because of numerical errors, for higher orders.

In the simulations, the state estimates were initiated as a random noise around the real state, as well as a constant error from the real state, in two different scenarios. Figure 5.20 shows the state as well as the estimate for the first case, while Figure 5.21 shows the same for the constant initial error case. Figures 5.22 and 5.23 show the corresponding squared error $e^2(t, x)$.

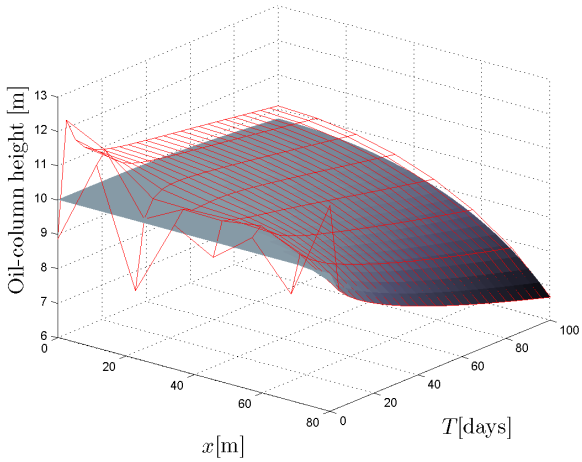


Figure 5.20: State and estimate when using EKF observer. Initial estimate is real state plus noise. The red mesh is the estimated state, while the blue surface is the real state.

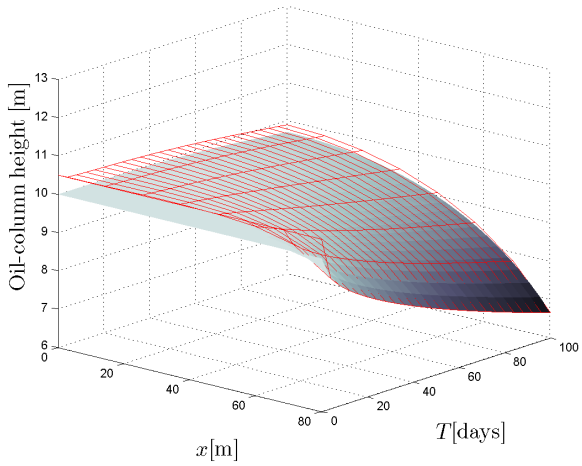


Figure 5.21: State and estimate when using EKF observer. Initial estimate is real state plus constant error. The red mesh is the estimated state, while the blue surface is the real state.

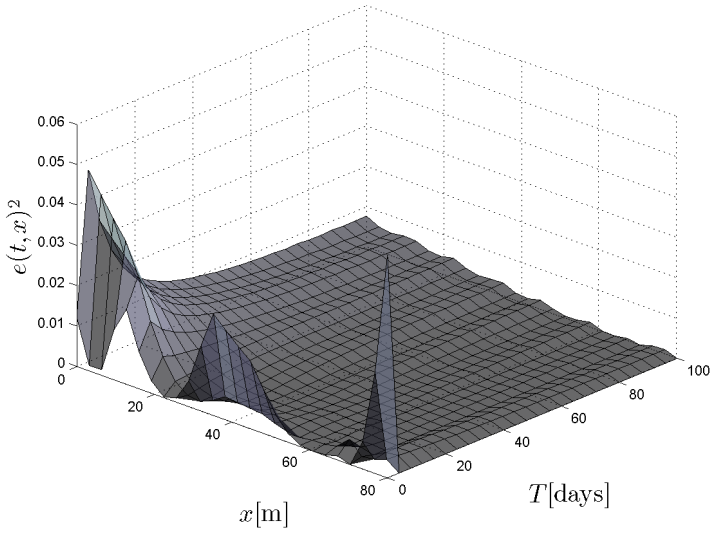


Figure 5.22: Squared estimation error when using EKF observer. Initial error is noise. Estimate vector $\hat{h}(t,x)$ interpolated to match order of full system.

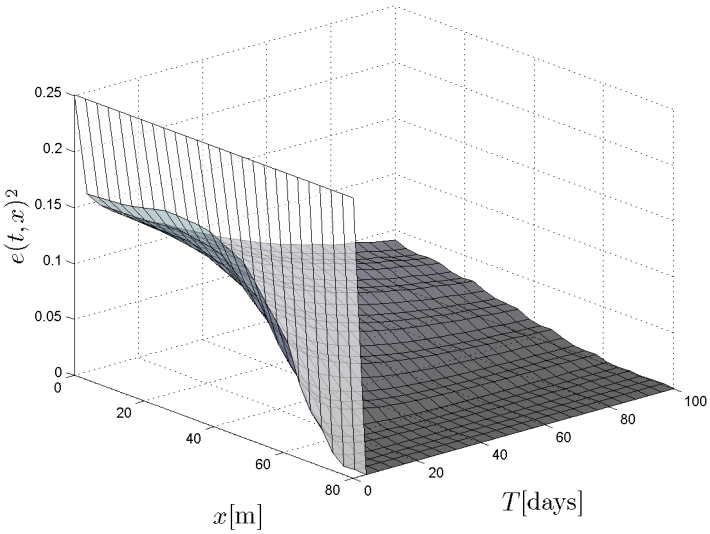


Figure 5.23: Squared estimation error when using EKF observer. Initial error is a constant.Estimate vector $\hat{h}(t,x)$ interpolated to match order of full system.

5.4 Combined Controller-Observer

As the system is boundary-sensed and the Backstepping control strategy is a state-feedback controller, its real-world implementation would require it to be used in combination with an observer. Paired with either the backstepping observer or the extended Kalman filter, the backstepping controller may be used in an *output-feedback* control strategy.

The Backstepping output-feedback controller is the combination of a backstepping controller and a backstepping observer. As the observer is stable, and the well in conjunction with the backstepping controller is also stable, the separation principle says the combination of the two will also be stable. However, stability of the well is not an issue as it is stable for any positive oil flux. The performance is instead measured by maximization of an objective function.

An issue when simulating observer-controller combinations is how the observer should be initialized. The initial value of the state estimate will undoubtedly affect the controller output trajectory. As a result, the performance measurement, in terms of sub-critical payout, will vary based on the observer initialization. Such simulations should not be taken into account when comparing controller performance.

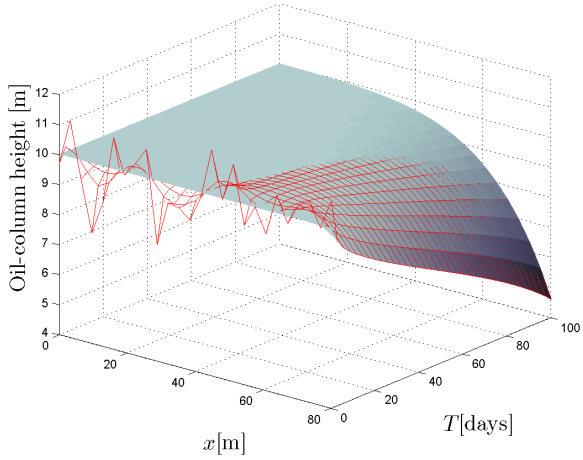


Figure 5.24: State and estimate when using the Backstepping output-feedback controller. Initial estimate is real state plus noise. The red mesh is the estimated state, while the blue surface is the real state.

The real trajectory when simulating using the Backstepping observer-controller pair is shown in Figure 5.24, together with the estimate from the observer. The results are similar to running the backstepping controller on the full state. Figure 5.25 shows squared estimation error.

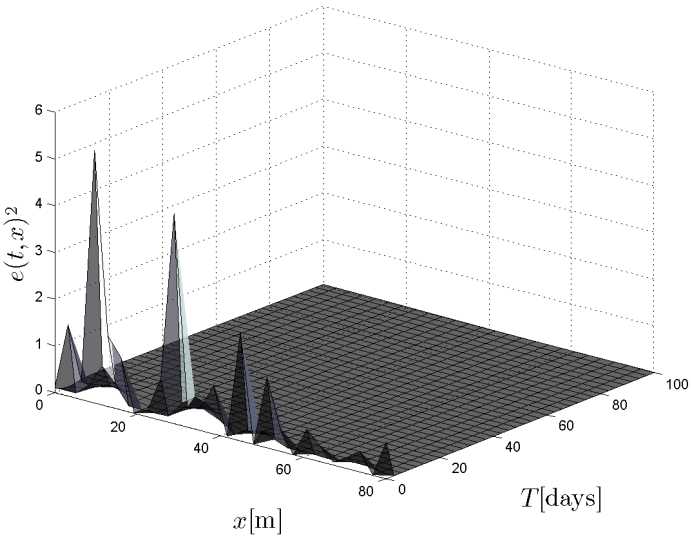


Figure 5.25: Squared estimation error when using Backstepping output-feedback controller. Initial error is white noise.

Chapter 6

Discussion and Conclusion

In this chapter the results from chapter 5 will be examined. The conclusion will take the form of a recommendation for a system to deal with gas-coning that I feel would be the best to use in actual, real-world reservoirs.

The importance of the stability result of Section 3.1 should be re-iterated at this point; it is shown that any control strategy will keep the well-reservoir stable, given than the oil flux q is kept positive. As a result, formal design methods such as backstepping may be useful for coming up with clever control strategies, but ultimately their merit is measured by how well they perform with respect to maximizing the objective function – profit from sub-critical production.

Controller performance

By the measure of sub-critical phase profits, the backstepping design outperforms Sagatun's controller from [7], as is shown in Table 5.1. The performance gain is, however, marginal. It is unclear if the small increase (0.4%) justifies the added complexity – Sagatun's controller is an output-feedback control law, whereas the backstepping method is a full state feedback control law. In addition, the computations themselvs are more complex. The added complexity in controller structure and needed measurements make the Backstepping controller uneconomical, when the meager performance gain is considered. The results are consistant with results presented in [30].

The same table also shows that for the generalized Sagatun controller, a more substantial performance increase may be achieved, upwards of 3%. Note that this may not be directly translated to well-lifetime earnings, for the super-critical production is beyond the scope of this thesis. The generalized Sagatun controller is preferable to the backstepping controller both in terms of performence, and in terms of complexity – it keeps the simpler, output feedback structure, while outperforming both other control laws in terms of sub-critical payout.

The generalized Sagatun controller achieves a much greater increase in earnings

than the Backstepping controller, while preserving the output-feedback structure of Sagatun. This makes it a viable candidate for production planning.

The one step controller suggested in Section 3.5 failed to live up to expectations. It was outlined as an ideal limit to the generalized Sagatun controller, representing the terminal value of Figure 5.2. However, as is clearly visible from Figure 5.5, the Maxi-Mini controller in its current form is no contender, in terms of performance. Possibly, starting rates need to be varied as well as switch times, in order to get this simple controller to yield better results.

The nonlinear optimal control alternative, while very tempting, is infeasible. The system is too large to be solved on a present-day consumer PC. While further research may yield better results on this interesting path, it is my view that simpler control strategies are more viable at this point.

While a nonlinear optimal control problem may be infeasible at this point, a simplified QP formulation achieves a production schedule which far out-performs any of the other control laws, in terms of sub-critical payout. Examining the optimal oil flux in Figure 5.11 and the corresponding reservoir trajectory in Figure 5.10, it is apparent the optimal control strategy may be worded as follows; produce at or close to the critical rate for short bursts, while producing at a minimal production rate between those bursts. This way, the next burst will be allowed to produce at a higher critical rate, since the cone was allowed to settle, and the stand-off is greater [4],[5].

The optimal controller's tendency to produce at the critical rate supports the suggestion of [2] to produce as close to the critical rate as possible.

It is also possible that shutting the well completely during the minimum production rate periods would yield even higher payouts.

Observer performance

The Backstepping observer works satisfactorily, as is evident in Figures 5.16 and 5.18, at least when the initial estimate error is centered about the real state. Convergence is quick and the estimated state follows the real state closely. However, when the initial estimation error is a constant, the backstepping controller fails to quickly converge, as is seen in Figures 5.17 and 5.19.

The good results attained by the backstepping observer make a *Backstepping output-feedback controller* a viable choice, as may be seen in Figure 5.24. Other state-feedback methods may also be paired with this observer to create output-feedback designs. However, the backstepping observer is designed for a linearized system and as such reflects a simplification of system dynamics.

The extended Kalman filter works very well as a nonlinear observer. This is evident from Figures 5.20–5.23. The backstepping observer seems to be able to deal with noise initial errors better than the EKF, but note that the EKF also converges rather quickly – and for the nonlinear system. The backstepping observer was only designed for a

linearized model. The main weakness of the EKF design is the lower order. While the linear backstepping observer is defined in a continuous range, the EKF requires a discretization of the state vector, and may only be run on relatively low-order discretizations. When the initial estimation error is a constant, the EKF converges quicker than the backstepping observer.

6.1 Conclusion

Based on the discussion above, I propose a short-term production planning method for the sub-critical phase based on quadratic optimal control. The performance gain this control scheme provides is so great, that it makes up for the added complexity. A 14% increase in earnings in a multi-million dollar business justifies the need for complex computations as well as the implementation of a state observer.

Such an implementation should rely on the receding horizon principle. In that scheme, the optimal trajectory should be calculated for the entire sub-critical phase of the well, but only the first part of the corresponding production plan should be used. Then, using new information, which will differ from the predicted one, the same optimization should be run again. It is highly neccesary to do this here, as the optimal controller is based on a linear model, and the model should be re-linearized at each iteration, much like the extended Kalman filter.

Since measurements are not available throughout the domain, the optimal controller will need to be paired with a state observer. The extended Kalman filter was proven to be effective at predicting the nonlinear dynamics of gas coning. As the industrial standard, it should also be used in this application.

Bibliography

[1] M. Muskat and R. D. Wycokoff, "An approximate theory of water coning in oil production," *Petroleum Transactions, AIME, Volume 114, 1934, pages 144-163.*, 1934.

[2] J. Konieczek, "The Concept of Critical Rate in Gas Coning and its Use in Production Forecasting," *SPE Annual Technical Conference and Exhibition, 23-26 September 1990, New Orleans, Louisiana*, 1990.

[3] A. Singhal, "Water and gas coning/cresting: A technology overview," *The Journal of Canadian Petroleum Technology, 96-04-06*, 1996.

[4] D. Sobocinski and A.J.Cornelius, "A correlation for predicting water coning time," *JPT, May 1965, pp. 594-600*, 1954.

[5] C. Bournazel and B. Jeanson, "Fast water coning evaluation," *SPE 3628*, 1971.

[6] A. Benamara and F. Tiab, "Gas coning in vertical and horizontal wells, a numerical approach," *SPE 71026*, 2001.

[7] S. I. Sagatun, "Boundary conctol of a horizontal oil reservoir," *SPE J. SPE-135534-PA*, 2010.

[8] S. S. Are Mjaavatten, Robert Aasheim and O. Gronning, "Model of gas coning and rate-dependent gas/oil radio in an oil-rim reservoir," *SPE 102390*, 2008.

[9] G. Howard and C. Fast, "Squeeze cementing operations," *AIME 189, pp. 53-64*, 1950.

[10] T. Wagenhofer and D. Hatzignatiou, "Optimization of horizontal well placement," *SPE 35714*, 1996.

[11] E. Shirman and A. Wojtanowicz, "More oil using downhold water-sink technology: A feasibility study," *SPE 66532*, 2000.

[12] W. F. B.T. Haug and T. Kydland, "Horizontal wells in the water zone: the most effective way of tapping oil from thin oil zones?" *SPE 22929*, 1991.

[13] V. Gunnerud and E. Langvik, "Production planning optimization for the troll-c field," Master's thesis, Norwegian University of Science and Technology, 2007.

[14] A. Leemhuis, "Gas coning for smart wells using a dynamic coupled well-reservoir simulator," *SPE 112234*, 2008.

[15] J. Ronen, "Backstepping boundary control for a horizontal oil well," Fall 2009, term project, Norwegian University of Science and Technology.

[16] S. I. Sagatun, personal correspondence, 2009.

[17] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations, 2nd Edition*. SIAM, 2004.

[18] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs: a course on backstepping designs*. SIAM, 2008.

[19] W. Liu, "Boundary Feedback Stabilization of an Unstable Heat Equation," *SIAM J. CONTROL OPTIM., vol 42, No. 3, pp. 1033-1043*, 2003.

[20] M. Krstic, "On global stabilization of Burgers' equation by boundary control," *Systems and Control Letters, 37, 123-142*, 1999.

[21] H. K. Khalil, *Nonlinear systems, 3rd ed.* Prentice Hall, 2002.

[22] R. Vazquez and M. Krstic, "Control of 1-D parabolic PDEs with Volterra nonlinearities, Part I: Design," *Automatica 44, 2778-2790*, 2008.

[23] ——, "Control of 1-D parabolic PDEs with Volterra nonlinearities, Part II: Analysis," *Automatica 44, 2791-2803*, 2008.

[24] W. Liu and M. Krstic, "Backstepping boundary control of burgers' equation with actuator dynamics," *Systems and Control Letters, 41, 291-303*, 2000.

[25] A. Balogh, O. Aamo, and M. Krstic, "Optimal mixing enhancement in 3d pipe flow," *IEEE Transactions on Control Systems Technology, 13, pp. 27–41*, 2005.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization, 2nd ed.* Springer, 2000.

[27] C.-T. Chen, *Linear System Theory and Design, 3rd. Ed.* Oxford University Press, 1999.

[28] R. Brown and P. Hwang, *Introductions To Random Signals And Applied Kalman Filtering, Third Edition*. Wiley, 1997.

[29] D. Simon, *Optimal State Estimation: Kalman, H-infinity and nonlinear approaches*. Wiley-Interscience, 2006.

[30] S. I. S. Agus Hasan and B. Foss, "Well rate control design for gas coning problem," *49th IEEE Conference on Decision and Control*, 2010.

[31] J. Balchen, T. Andresen, and B. Foss, *Reguleringsteknikk, 5th ed.* NTNU-Trykk, 2003.

Appendix A

A.1 Solution of gain kernel PDE

This appendix solves the gain kernel PDE from Section 3.3.2.

The gain kernel PDE is given by

$$k_{xx}(x,y) = k_{yy}(x,y) \tag{A.1}$$
$$k_y(x,0) = -\epsilon k(x,0) \tag{A.2}$$
$$k(x,x) = -\epsilon. \tag{A.3}$$

The general solution of the PDE has the form $k(x,y) = \phi(x-y) + \psi(x+y)$ where ϕ and ψ are arbitrary functions. From (A.3), we get

$$k(x,x) = \phi(0) + \psi(2x) = -\epsilon. \tag{A.4}$$

This implies that $\psi(x)$ is constant, and may therefore be absorbed into ϕ; hence $\psi \equiv 0$. Using $k(x,y) = \phi(x-y)$ and (A.2) we have

$$-\phi'(x) = -\epsilon\phi(x) \tag{A.5}$$
$$\int \frac{d\phi}{\phi} dx = \epsilon \int dx \tag{A.6}$$
$$\ln \phi(x) = \epsilon x + C \tag{A.7}$$
$$\phi(x) = e^C e^{\epsilon x}. \tag{A.8}$$

Using (A.3) yields $e^C = -\epsilon$, and thus we have

$$k(x,y) = -\epsilon e^{\epsilon(x-y)}. \quad \square \tag{A.9}$$

A.2 Proof of instability for $\lambda \leq$ threshold

This appendix shows the proof from Example 2.

The PDE is given by

$$u_t(t,x) = u_{xx}(t,x) + \lambda u(t,x) \tag{A.10}$$
$$u_x(t,0) = 0 \tag{A.11}$$
$$u(t,1) = 0. \tag{A.12}$$

One method for obtaining the solution is separation of variables and the Laplace transform. We assume the solution $u(t,x)$ is separable and may be written as a product of a function of space and a function of time,

$$u(t,x) = T(t)X(x). \tag{A.13}$$

Substituting (A.13) into the equation (A.10) and using $u_t = \dot{T}X$ and $u_{xx} = TX''$ yields

$$X(x)\dot{T}(t) = X''(x)T(t) + \lambda X(x)T(t). \tag{A.14}$$

Gathering the terms, we may write

$$\frac{\dot{T}(t)}{T(t)} = \frac{X''(x) + \lambda X(x)}{X(x)} = \sigma, \tag{A.15}$$

which is a constant, since the two sides of the equality are independent. This results in an ODE for $T(t)$ and an ODE for $X(x)$:

$$\dot{T} = \sigma T \tag{A.16}$$
$$X'' + (\lambda - \sigma)X = 0. \tag{A.17}$$

Equation (A.16) has the initial condition $T(0) = T_0$, and (A.17) has the boundary conditions $X'(0) = 0$ and $X(1) = 0$. These follow from the boundary conditions (A.11)–(A.12). The solution to (A.16) is given by

$$T(t) = T_0 e^{\sigma t}. \tag{A.18}$$

Equation (A.17) may be solved using the Laplace transform:

$$\mathscr{L}\{X''(x) + (\lambda - \sigma)X(x)\} = s^2\mathcal{X}(s) + sX(0) + (\lambda - \sigma)\mathcal{X} = 0 \tag{A.19}$$
$$\mathcal{X}(s) = X(0)\frac{s}{s^2 + (\lambda - \sigma)} \tag{A.20}$$
$$X(x) = \mathscr{L}^{-1}\left\{X(0)\frac{s}{s^2 + (\lambda - \sigma)}\right\} \tag{A.21}$$
$$X(x) = X(0)\cos\left(\sqrt{\lambda - \sigma}x\right). \tag{A.22}$$

The last step can be found in the inverse Laplace transform table on p. 550 of [31]. From the boundary condition $X(1) = 0$ we have

$$X(1) = X(0) \cos \sqrt{\lambda - \sigma} = 0, \tag{A.23}$$

which, in addition to the trivial solution $X(0) = 0$, which is not interesting, can only hold if $\sqrt{\lambda - \sigma} = \frac{\pi}{2} + \pi n$ for $n \in \mathbb{Z}$, or

$$\sigma = \lambda - \pi^2 \left(n + \frac{1}{2} \right)^2. \tag{A.24}$$

Next, plugging (A.18) and (A.22) into (A.13), we have

$$u_n(t, x) = T_0 X_n(0) e^{\left(\lambda - \pi^2 \left(n + \frac{1}{2}\right)^2\right) t} \cos \left(\pi \left(n + \frac{1}{2} \right) x \right). \tag{A.25}$$

This is a particular solution, and since (A.10) is a linear PDE, the sum of particular solutions is also a solution. The general solution is therefore given by

$$u(t, x) = \sum_{n=1}^{\infty} C_n e^{\left(\lambda - \pi^2 \left(n + \frac{1}{2}\right)^2\right) t} \cos \left(\pi \left(n + \frac{1}{2} \right) x \right). \tag{A.26}$$

We see that the terms $e^{\left(\lambda - \pi^2 \left(n + \frac{1}{2}\right)^2\right) t}$ determine the rate of growth of the solution. Therefore, the exponential factor $\left(\lambda - \pi^2 \left(n + \frac{1}{2} \right)^2 \right)$ all need to be negative for the plant to be stable. It follows that the plant is stable if and only if the largest eigenvalue, when $n = 0$ is negative. Hence, the plant is *unstable* for

$$\lambda > \frac{\pi^2}{4}. \quad \square \tag{A.27}$$