Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Adshayan Karunananthan

Real-time optimization of oil production using modifier adaptation with machine learning

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Struen Imsland
June 2019

NTNU
Norwegian University of Science and Technology

Adshayan Karunananthan

Real-time optimization of oil production using modifier adaptation with machine learning

Master's thesis in Cybernetics and Robotics
Supervisor: Lars Struen Imsland
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

PROJECT DESCRIPTION SHEET

Name of the candidate: Adshayan Karunananthan

Thesis title (English): Real-time optimization of oil production using modifier adaptation with machine learning

Background

Mature oil fields that have passed "plateau" production often has a complex bottleneck structure, which implies that choosing which wells to produce from, and how much, is not obvious. On top of this, uncertainty of reservoir characteristics (and equipment capacities) makes model-based optimization challenging.

The topic of this project is to investigate whether the "modifier adaptation" approach to real-time optimization can be a possible solution to the challenges mentioned above, by using measurements to update gradients in the optimization problem. To achieve better learning from data, the project should extend the method using machine learning in the form of Gaussian Processes (GP), from the paper

> Ferreira et al. (2017), "Real-Time Optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes", European Conference on Control, 2017

The methods ("standard" modifier adaptation and the extension with GP) should be tested and compared on a simple example from oil production.

Work description

1. Give a brief overview over oil and gas production systems and the need for production optimization
2. Describe the modifier adaptation approach to real-time optimization (MA-RTO), and its extension using GPs.
3. Set up a static optimization problem for production optimization in a simple two-well system with gas capacity constraints
4. Illustrate the use of MA methods to handle unknown/uncertain reservoir and/or equipment parameters

Start date: January 20, 2019 **Due date:** June XX, 2019

Supervisor: Lars Imsland
Co-advisor(s):

Trondheim, __January 20th, 2019_____

__ __

Lars Imsland
Supervisor

Address Sem Sælandsvei 5 NO-7491 Trondheim
Org.no. 974 767 880 E-mail: postmottak@itk.ntnu.no http://www.itk.ntnu.no
Location O.S. Bragstads plass 2D NO-7034 Trondheim
Phone + 47 73 59 43 76
Fax + 47 73 59 45 99
Phone: + 47 47 23 19 49

Abstract

The purpose of this study was to investigate modifier adaptation schemes to handle plant-model mismatch in real-time optimization of uncertain processes. Building upon the works of (Marchetti et al. 2016) and (Ferreira et al. 2018), two modifier adaptation algorithms are presented. The first one is a Modifier Adaptation scheme that uses first-order corrections to update the gradients in the optimization problem. However, first-order corrections rely on accurate estimations of plant-gradients, which can be both impossible and costly. A second variant is a machine learning based Modifier Adaptation scheme, which replaces the first order corrections with Gaussian process regression to represent the plant-model mismatch and estimation of plant gradients. Both algorithms are demonstrated in a two-well oil production system. The simulations show that the latter algorithm outperforms the former in terms of noise mitigation. However, the latter algorithm did not satisfy feasible-side convergence. Therefore, further investigation is needed to determine if the Aodifier Adaptation scheme with Gaussian process regression is able to handle plant-model mismatch in real-time optimization of uncertain processes.

Sammendrag

Hensikten med denne studien var å undersøke om optimalisering med "Modifier Adapation"-rammverket er i stand til å takle modelleringsfeil i "Real-time optimization" av usikre prosesser. To ulike algortimer, basert på arbeidet i (Marchetti et al. 2016) og (Ferreira et al. 2018), er presentert i oppgaven. Den første metoden bruker førsteordens korreksjoner til å oppdatere gradientene i optimaliseringsproblemet. For tilstrekkelig utførelse er denne metoden sterkt avhengig av presise estimater av gradientene, noe som kan være både umulig og kreve mye regnekraft. Den andre varianten er en "Modifier Adaptation"-metode som tar i bruk maskinlæring, ved å erstatte førsteordens korreksjoner med "gaussian process" regresjon, til å prediktere modellfeil og gradientene i optimaliseringsproblemet. Begge algoritmene er anvendt på et oljeproduksjonssytem med to oljebrønner. Basert på simuleringene kan man se at den sistnevnte algoritmen utkonkurrerer den første, med tanke på håndtering av målestøy. Selv om maskinlæringsalgoritmen klarer å konvergere til det optimale punktet, klarte den dog ikke å holde seg innenfor mulighetsområde under optimaliseringsprosessen. Derfor trengs det videre undersøkelser for å bestemme om "Modifier Adaptation" med "gaussian process"-regresjon er i stand til å løse modellfeil-problemet i Real-time optimization" av usikre prosesser.

Preface

0.1 Previous work

This master thesis is based on a project thesis carried out, by me, in fall 2018. The project thesis involved Real-time optimization of oil production using an algorithm called Modifier Adaptation. The models presented in this thesis are an extension of the models developed in the project thesis. Moreover, some of the content from the project thesis is reused here. However, this master thesis is a standalone work and can be read independently from previous work.

0.2 This thesis

This thesis is the culmination of my work at the Norwegian University of Science and technology, under the supervision of Professor Lars Struen Imsland. The thesis is written at the Department of cybernetics, during the spring semester of 2019.

The assumed background of the reader should be an education within control engineering and mathematical optimization. Familiarity with machine learning, and in particular function approximators such as Gaussian Process regression, is beneficial. However, the theory needed to understand the specifics of optimization

and Gaussian Process regression will be presented.

Mature oil fields that have passed “plateau” production often have a complex bottleneck structure. This makes it hard to decide which well to produce from, and how much to produce at a given time. Furthermore, the uncertainty of reservoir characteristics and equipment capacities makes model-based optimization challenging. The motivation of this thesis is to investigate whether the “Modifier Adaptation” approach to real-time optimization can be a solution to the mentioned challenges, by using measurements of the plant to update the gradients in the optimization problem.

Acknowledgments

I would like to thank my supervisor Professor Lars Struen Imsland for his guidance and valuable insights during this work. I would also like to thank Postdoc Leif Erik Andersson for taking the time to answer my questions and for sharing some of his expertise on the topics of Modifier Adaptation and Gaussian processes. Thanks go out to my fellow graduate students for inspiring and educational discussions. Lastly, my kindest thanks to my family and friends for the love and for encouraging me throughout this journey.

Adshayan Karunananthan

Trondheim, 22.06.2019

Contents

List of Tables

List of Figures

Abbreviations

AI	=	Artificial Intelligence
FDA	=	Finite-difference approximation
GOR	=	Gas-oil ratio
GP	=	Gaussian Processes
LP	=	Linear programming
MA	=	Modifier Adaptation
MA-RTO	=	Real-time optimization using Modifier Adaptation
ML	=	Machine Learning
QP	=	Quadratic programming
RTO	=	Real-time optimization
SQP	=	Sequential quadratic programming

Chapter 1

Introduction

1.1 Background and motivation

As oil and gas reserves are getting more expensive and hard to explore it is important that we ensure the best utilization of the resources. These projects are quite complex, thus there are many decisions that have to be taken carefully. The development of these fields involves multibillion investments with huge expecting returns. Therefore, the industry has been forced to come up with innovative strategies to obtain the optimal operation of production. To optimize there are many factors that have to be taken into consideration such as increased water depths, environmental conditions, reserve structure and the ratio between gas and oil.

Since the oil is the most valuable product in the reserves the optimal production in oil fields involves maximizing oil production. When producing oil there will also be produced gas and water, but the amount of the different fluids varies from well to well. The production facilities have gas handling capacities, which is the

maximum amount of gas the field can handle. Therefore, in oil production the ratio between gas and oil in each well, gas-oil ratio (GOR), is an important factor when deciding how to produce the oil. In other words it is essential that the oil production is from the wells with the lowest GOR at all times, for optimal production with respect to the gas handling capacity constraint.

At the moment there exist simulators that can predict the GOR for different wells. Combining these simulators with model-based optimization, as for example Real-time Optimization (RTO), is a powerful approach to optimize large scale production problems. However, if there are disturbances, sudden changes in GOR or reservoir pressures between the RTO iterations the GOR simulators will not be accurate enough. Hence the optimal operation calculated by the RTO will not be the actual optimal operation point for the plant. This problem, called plant-model mismatch, is researched a lot by the process engineering community and several RTO variants have arisen[1]. In this thesis I will investigate if this plant-model mismatch problem can be solved by a RTO method called Modifier Adaptation (MA).

1.2 Goal and method

The main goal of this thesis is to investigate whether the Modifier Adaptation approach to real-time optimization can be a solution to the plant-model mismatch problem, introduced above. First of all the standard “modifier adaptation” approach proposed in [1], using measurements to update gradients in the optimization problem, will be investigated. Then, to achieve better learning from data, the “modifier adaptation” approach with an extension using machine learning in the form of Gaussian processes (GP)[2] will also be investigated. Both the standard modifier adaptation and with extension using GP will be tested and

compared on a simple experiment from oil production.

1.3 Outline of report

Chapter 2 is committed to the literature review. This chapter covers the fundamental basics of mathematical optimization. Moreover, an introduction to machine learning will be given, with emphasis on Gaussian process regression. Lastly, the used software is introduced.

Chapter 3 gives an overview of the mathematical model of the oil-well. This chapter is based on chapter 3 in the project thesis.

Chapter 4 provides a formulation of a static optimization problem in the model developed in chapter 3. Furthermore, simulations on the model are presented. Here some important concepts and notation used further will be explained.

Chapter 5 presents the implementation of Real-time optimization using Modifier Adaptation (MA-RTO) to cope with mismatches between the model, modeled in chapter 4, and the plant. Moreover, simulation results are presented and a discussion of these. Lastly, we draw a conclusion based on the results and discussions and propose further work.

Chapter 2

Background and theory

2.1 Introduction to Optimization

Mathematical optimization is an important tool in numerous fields [3][4][5]. It is widely used in the fields of science, engineering, and economics. These include big industries such as the offshore industry. Developing oil fields involve multi-billion investments, hence well calculated and optimized decisions can save investors a lot of money[6]. Therefore, developing innovative optimization strategies for the process industry is highly prioritized in the research community.

First of all an optimization problem consist of an objective function and decision variables[3]. To use optimization techniques the objective function must be identified, a scalar measure to study the performance of the solution. The objective is described as a function of the decision variables. Hence the goal is to decide the value of the decision variables that optimizes the objective. In other words, if the objective is the profit of an oil production facility, the goal is to find the optimal variables that maximize the profit. Moreover, many problems are

constrained optimization problems. This means that the problems include a third component, constraints, that have to be respected when solving the problem. Finally, an optimization problem can be described as the following:

$$\min_{x} \quad f(\mathbf{x}) \tag{2.1}$$

subject to

$$c_i(\mathbf{x}) = 0 \qquad i \in \epsilon \tag{2.2}$$

$$c_i(\mathbf{x}) \leq 0 \qquad i \in \mathcal{I} \tag{2.3}$$

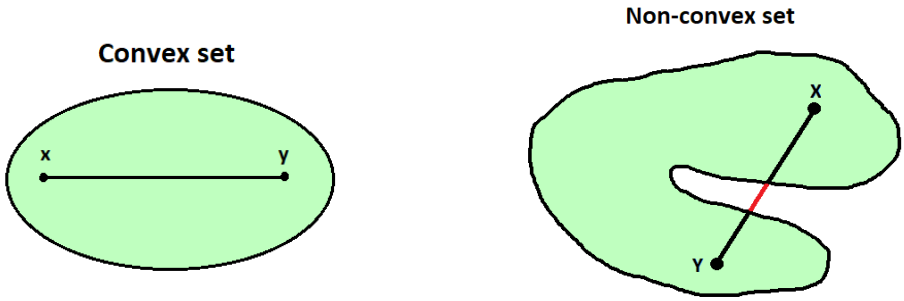
where x consists of the decision variables, f is the objective function, ϵ and $\mathcal{I}$ are the set of equality and inequality constraints, respectively. To point out, any maximization problem can be solved as a minimizing problem by minimizing the negative objective of the maximizing problem. This means that the optimal point of equation 2.1 is the same as for: $\max \ -f(\mathbf{x})$

The solution of equation 2.1 is at point $\mathbf{x}^* \in \Omega$ if:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad for\ all\ \mathbf{x} \in \Omega$$

Such a point is a global minimum. A local maximum $\mathbf{x}^*$ provides a maximum objective value within some neighborhood instead of the whole feasible region and is defined as follows:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad for\ all\ \mathbf{x} \in \|\mathbf{x}^* - \mathbf{x}\| \leq \epsilon$$



(a) Convex set (b) Non-convex set

Figure 2.1: Illustration of a convex set and a non-convex set

2.1.1 Convexity

In optimization, the problems can be divided into convex and non-convex problems. The concept of convexity is fundamental in optimization. Problems which possess this property are generally easier to solve both in theory and practice.

Convexity can be used to describe both sets and functions. A convex set is a set $S \in \mathbb{R}^n$, where the straight line segment connecting any two points in S lies entirely inside S. Figure 2.1a illustrates a convex set and figure 2.1b shows a non-convex set. Furthermore, a function is convex if its domain S is a convex set and if for any two points x and y in S, the following conditions holds:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \quad for\ all\ \alpha \in [0,1] \tag{2.4}$$

If both the objective function and the feasible region for optimization is convex, the problem holds the key property that a local optimal point is also a global optimal point.

Convex programming is used to describe constrained optimization problems where [3]:

1. the objective function is convex,
2. the equality constraint functions are linear, and
3. the inequality constraint functions are concave.

2.2 Linear Programming

If both the objective function and the constraints in equations 2.1, 2.2 and 2.3 are linear, we have the simplest form of optimization problems, called linear programs. Linear programs can generally be written as following:

$$\min_{x} \quad \mathbf{c}^T\mathbf{x} \tag{2.5}$$

subject to

$$\mathbf{A}_e\mathbf{x} - \mathbf{B}_e = 0 \tag{2.6}$$

$$\mathbf{A}_i\mathbf{x} - \mathbf{B}_i \leq 0 \tag{2.7}$$

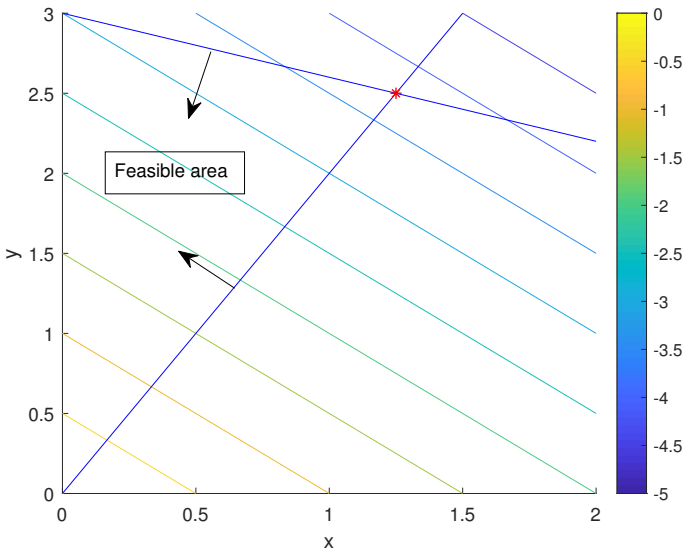


Figure 2.2: LP problem

Figure 2.2 illustrates a simple Linear Program problem with two decision variables, where the objective is:

$$\min \quad -(x + y) \tag{2.8}$$

The problem is constrained by two inequality constraints, illustrated with the two blue lines. As can be observed the contour lines shows how the objective function changes linearly. The red star is a global optimal point where the objective is maximized with respect to the constraints. Since LP problems are convex the global optimal point is easily found as the KKT conditions are both necessary and sufficient to meet a global solution. For further discussion on this the reader can read chapter 12 in [3].

2.3 Quadratic Programming

When the objective function in equation 2.1 is a quadratic function and all constraints are linear the problem is a quadratic programming problem, known as a QP problem. QP problems are some of the most important nonlinear programming problems. These arise as subproblems in methods for general constrained optimization as in sequential quadratic programming (SQP) which will be discussed in section 2.4.1. QP problems are defined as:

$$\min_{x} \quad \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{2.9}$$

subject to

$$\mathbf{A}_e\mathbf{x} - \mathbf{B}_e = 0 \tag{2.10}$$

$$\mathbf{A}_i\mathbf{x} - \mathbf{B}_i \leq 0 \tag{2.11}$$

As all constraints are linear the feasible set is convex. QP problems are convex if the symmetric matrix Q is positive semidefinite.

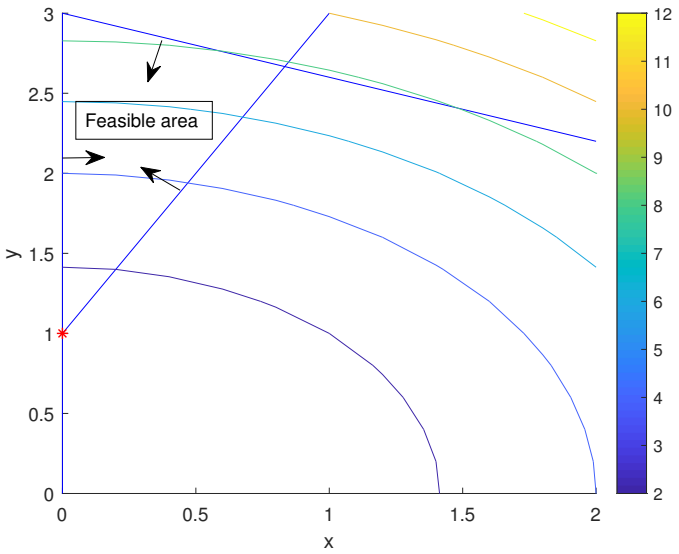


Figure 2.3: QP problem

Figure 2.3 illustrates a simple QP problem with two decision variables, where the objective is:

$$\min \quad \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.12}$$

As illustrated by the three blue lines in figure 2.3 the problem is constrained by three inequality constraints. Different from the LP problem in section 2.2 the contour lines here are quadratic, as the objective is quadratic. The red star is a global optimal point where the objective is minimized with respect to the constraints.

A key point about QP problems is that they always can be solved, or shown

to be infeasible, in a finite amount of computation[3]. First thing to remember is that the required computational effort is highly dependent on the objective function and the number of inequality constraints. In fact, a convex objective function often makes the difficulty level of the problem similar to an LP problem. Not surprisingly a non-convex QP is more challenging, regarding that the problem can have several stationary points and local optimums.

2.4 Nonlinear programming

Nonlinear optimization problems include nonlinear objective functions or/and nonlinear constraints. These problems are in general non-convex which makes it challenging to find global optimums [3][7]. QP problems, introduced in section 2.3, are also by definition nonlinear problems, but are considered separately due to their intrinsic importance, since their characteristics can be exploited by efficient algorithms [3]. Figure 2.4 shows a non-convex function with one parameter. As can be observed both the red and black stars are local maximums. In fact, the black star is also a global maximum. The challenge in non-convex optimization problems, as illustrated here, is to converge to the black star instead of the red star. Luckily it has been developed powerful optimization methods to cope with this, which will be discussed later in the chapter.

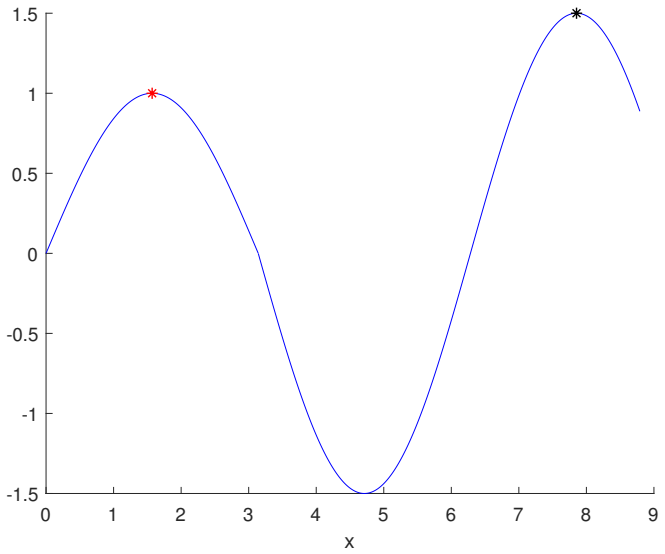


Figure 2.4: Non-convex objevtive

2.4.1 Sequential quadratic programming

Regarding that most of the optimization problems in the process field are highly nonlinear and non-convex, solving these are not straightforward. As mentioned in section 2.3, both convex and non-convex QP problems can be solved in a finite amount of computation. Sequential quadratic programming (SQP) algorithms take advantage of this key property to solve other nonlinear problems. More precisely SQP approximates a nonlinear problem to a QP problem and solves it for each iteration. At point $\mathbf{x}_k$, in iteration k, SQP algorithms approximate equation 2.1 - 2.3 to:

$$\max_{p} \; f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)\mathbf{p} + \frac{1}{2}\mathbf{p}^T \nabla^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}_k, \lambda_k)\mathbf{p} \tag{2.13}$$

subject to

$$\nabla c_i(\mathbf{x_k})^T \mathbf{p} + c_i(\mathbf{x_k}) = 0 \qquad i \in \epsilon \tag{2.14}$$

$$\nabla c_i(\mathbf{x_k})^T \mathbf{p} + c_i(\mathbf{x_k}) \leq 0 \qquad i \in \mathcal{I} \tag{2.15}$$

where

$$\mathcal{L}(\mathbf{x}_k, \lambda_k) = f(\mathbf{x}_k) + \sum_{i \in \epsilon \cup \mathcal{I}} \lambda_i c_i(\mathbf{x}_k) \tag{2.16}$$

and λ_i are the lagrangian multipliers. For detailed information on this, the reader is referred to chapter 18 in [3]. After solving this problem the SQP algortihm solve the same problem for $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}$, where $\alpha \in [0, 1]$ denotes the step. This is done sequentially until the convergence properties of the algorithm is satisfied.

2.5 Applications

2.5.1 Real-time optimization

Real-time optimization, known as RTO, is a model-based optimization approach. RTO methods are widely used in the industry because they are powerful when optimizing large scale problems[8]. These methods are typically structured in three main steps [1]. First of all the process optimization contain a model of the process, also known as process modeling, followed by numerical optimization of the obtained process model. The numerical optimization is done by some optimization algorithms, in example SQP from section 2.4.1. Then an application of the optimal inputs, obtained from the numerical optimization, are applied on the plant. In theory this seems quite simple and straightforward, however, in reality, the models are usually not perfect representations of the plants. Hence solving these problems in the real world are more challenging.

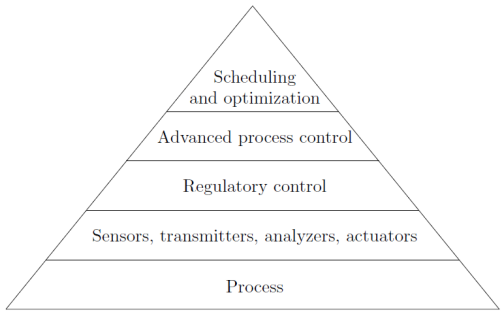


Figure 2.5: Typical RTO control hierarchy [7]

The model-based optimal inputs, calculated by the RTO algorithms, are indeed optimal for the model, but as long as the model is not a perfect representation the inputs are often sub-optimal for the plant. However, RTO has shown its power to converge to optimal inputs, even when there exist mismatches between the plant and the model. This problem, called plant-model mismatch, is researched a lot by the process engineering community and several RTO variants have arisen. In this thesis, a specific RTO method called Modifier Adaptation(MA) is being investigated. Figure 2.5 illustrates a typical RTO control hierarchy.

2.5.2 Modifier Adaptation approach

In section 2.5.1 the challenge of making an accurate model of the plant was introduced. These inaccuracies, called plant-model mismatches, are mainly caused by one of the following reasons or a combination of multiple of them[1]:

- Parametric uncertainty - when the model parameters do not correspond to the real process.

- Structural plant-model mismatch - when there is a mismatch in the structure of the model, for instance due to simplified/neglected dynamics or unknown characteristics of the process.
- Process disturbances.

When choosing an RTO-method, some important properties have to be considered. Guaranteed plant optimality upon convergence is one of the desired properties. In addition, fast convergence and feasible-side convergence are two ideal properties too. In fact, the latter two properties often oppose each other. As an illustration fast convergence often require large steps, while feasible-side convergence often calls for small steps. Hence to satisfy these two requirements there must be a compromise between large and small step sizes. The key characteristic of modifier adaptation is that it guarantees convergence to plant optimum even if there exist structural plant-model mismatches.

Modifier Adaptation is an RTO method that uses the process measurements to improve cost and constraint functions. In general, it uses correction terms for the cost and constraint function to update the plant model, instead of estimating the plant parameters, which is a more common strategy. More precisely it estimates the plant gradients from the measurements, which are used as gradient correction terms in the model to modify both cost and constraint functions in the optimization problem. The use of gradients is justified by the necessary conditions of optimality that include constraints with sensitivity conditions. By enforcing the plant's and model's necessary conditions of optimality to match, the modified model will be a likely candidate to solve the plant optimization problem.

As mentioned above the Modifier Adaptation method is implemented by adding some correction terms to the optimization model. Hence the optimization problem

for equation 2.1 becomes:

$$x^*_{k+1} = \max_{\boldsymbol{x}} \quad f(\mathbf{x}) + (\boldsymbol{\lambda}^f_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) \tag{2.17}$$

$s.t.$

$$c_{i,k}(\boldsymbol{x}) = c_i(\boldsymbol{x}) + \epsilon^{c_i} + (\boldsymbol{\lambda}^{c_i}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) = 0 \qquad i \in \epsilon \tag{2.18}$$

$$c_{i,k}(\boldsymbol{x}) = c_i(\boldsymbol{x}) + \epsilon^{c_i} + (\boldsymbol{\lambda}^{c_i}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) \leq 0 \qquad i \in \mathcal{I} \tag{2.19}$$

where the modifiers are defined as follows:

$$\begin{aligned}
\epsilon^{c_i} &= c_{p,i}(\boldsymbol{x}_k) - c_i(\boldsymbol{x}_k) \\
\boldsymbol{\lambda}^f_k &= (1 - K^f)\boldsymbol{\lambda}^f_{k-1} + K^f(\widehat{\nabla \boldsymbol{f}}_{p,k} - \widehat{\nabla \boldsymbol{f}}_k) \\
\boldsymbol{\lambda}^{c_i}_k &= (1 - K^{c_i})\boldsymbol{\lambda}^{c_i}_{k-1} + K^{c_i}(\widehat{\nabla \boldsymbol{c}}_{p,i,k} - \widehat{\nabla \boldsymbol{c}}_{i,k})
\end{aligned}$$

The variables with subscript p, $c_{p,i}$, $\widehat{\nabla \boldsymbol{f}}_{p,k}$ and $\widehat{\nabla \boldsymbol{c}}_{p,i,k}$ to be specific, are based on measurements directly from the plant. Similarly the same variable without subscript p are the model values for the same variables. Note that the modifiers becomes 0 when there is no mismatch between the model and the plant. In this case the model is a perfect representation of the plant and the Modifier Adaptation terms are not needed. Moreover the gradients, $\widehat{\nabla \boldsymbol{f}}_{p,k}$, $\widehat{\nabla \boldsymbol{f}}_k$, $\widehat{\nabla \boldsymbol{c}}_{p,i,k}$ and $\widehat{\nabla \boldsymbol{c}}_{i,k}$, are estimated as it is very challenging to measure them directly [1]. Estimation methods of the gradients are discussed in more detail in section 2.5.3.

2.5.3 Gradient estimation

Regarding that the plant gradients cannot be measured directly, implementing MA in real situations can be difficult. Especially obtaining reliable estimates of the gradients from noisy measurements can be quite challenging. Estimating

gradients can be done by dynamic perturbation methods, that use transient data, or steady-state perturbation methods, which is simpler since they only use stationary data.

In this thesis, two different gradient approximation methods will be applied and the performances of each will be compared. Multiple methods are presented in [1]. One of the proposed methods is finite-difference approximation using past RTO points, which will be used in this project. In addition to that a Machine learning approach called Gaussian processes will be investigated[2]. In the next sections these two methods will be explained in detail.

2.5.3.1 Finite difference approximation using past RTO points

Finite difference approximation is the simplest steady-state method to estimate the gradients. First, each input is perturbed around the operating point at the current step. Secondly, the corresponding gradient elements get measured when the process reaches steady state.

Gradients with finite-difference approximation can be obtained using past RTO points. Initialization of this technique requires a number of operating points equal to the number of inputs in the problem, n_x, plus one in order to estimate the gradients. For instance, in a problem with two input variables, the method requires 3 operating points. One can obtain these points by perturbating each input around the current operating point two times. Thus the following matrices can be constructed

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{x}_k - \boldsymbol{x}_{k-1} & \boldsymbol{x}_k - \boldsymbol{x}_{k-2} & \dots & \boldsymbol{x}_k - \boldsymbol{x}_{k-n_x} \end{bmatrix}$$

$$\delta \widetilde{f}_{p,k}{}^T = \begin{bmatrix} \widetilde{f}_{p,k} - \widetilde{f}_{p,k-1} & \widetilde{f}_{p,k} - \widetilde{f}_{p,k-2} \end{bmatrix}$$

$$\delta\widetilde{c}_{p,i,k}{}^{T} = \begin{bmatrix} \widetilde{c}_{p,i,k} - \widetilde{c}_{p,i,k-1} & \widetilde{c}_{p,i,k} - \widetilde{c}_{p,i,k-2} \end{bmatrix}$$

Finally the plant gradients become

$$\widehat{\nabla \boldsymbol{f}}_{p,k} = \delta\widetilde{f}_{p,k}{}^{T}(U_k)^{-1}$$
$$\widehat{\nabla \boldsymbol{c}}_{p,i,k} = \delta\widetilde{c}_{p,i,k}{}^{T}(U_k)^{-1}$$

Finite difference approximation is sufficient for processes without noise and with few inputs. However, most realistic processes have noise and the method can lead to constraint violation when it operates near a constraint. A more robust alternative to finite difference approximation is a quadratic approximation[1]. The quadratic model obtains a local quadratic approximation of the cost and constraint functions, using the current and past operating points. Further, the model calculates the plant gradient. Since higher order of approximation captures more precise information of the process it consequently decreases the influence of the noise. Hence this is more accurate but is indeed more complex.

2.5.3.2 Gaussian processes

Gaussian processes, known as GP, is a regression method to estimate unknown functions. The method has been very popular in the machine learning community for a long time, and is now also becoming popular in the field of optimization and control due to its simplicity and effectiveness[2][9][10]. By simplicity and effectiveness, it is referred to GPs ability to capture complex unknown functions using just a few parameters. The technique is probabilistic and non-parametric

and can be seen as a multivariate normal distribution with up to infinitely many random variables. Different from parametric methods that only use the training data in the learning phase to identify the parameters, GP uses kernel methods that use all the available data to estimate the mapping functions between input and output data.

With GP the objective is to approximate the unknown function

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \tag{2.20}$$

using observations

$$y_i = f(\boldsymbol{x}_i) + v \tag{2.21}$$

It is assumed that the observations y_i differ from the actual values $f(\boldsymbol{x}_i)$ by an additive noise v. The noise is assumed to follow an independent, identically distributed Gaussian distribution with zero mean and variance σ_v^2.

$$v \sim \mathcal{N}(0, \sigma_v^2) \tag{2.22}$$

Furthermore n_p available input-output pairs

$$\bar{X} = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots\ \boldsymbol{x}_{np}] \in \mathbb{R}^{x_{dim} \times np}, \quad \bar{\boldsymbol{y}} = [y_1,\ y_2,\ \ldots\ y_{np}]^T \in \mathbb{R}^{np \times 1} \tag{2.23}$$

are used to establish a relationship between the inputs and the outputs. From this a corresponding conditional distribution of the output y for a new input $\boldsymbol{x}$ can be obtained:

$$f(\boldsymbol{x}) \mid (\bar{X}, \bar{\boldsymbol{y}}) \sim \mathcal{N}(\mu_f, \sigma_f^2) \tag{2.24}$$

$$\mu_f = \mu_f(\bar{X}, \bar{\boldsymbol{y}}) = \boldsymbol{c}^T(\bar{C} + \sigma_v^2 I_{np})^{-1}\bar{\boldsymbol{y}} \tag{2.25}$$

$$\sigma_f = \sigma_f(\boldsymbol{x}, \bar{X}, \bar{\boldsymbol{y}}) = c(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{c}^T(\bar{C} + \sigma_v^2 I_{np})^{-1}\boldsymbol{c} \tag{2.26}$$

where I_{np} is the identity matrix with dimension $np \times np$ and $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the kernel. [2] discuss multiple different kernels, but in this project the auto relevance determination square exponential covariance function is used, defined as follows:

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_c^2 \exp\left(-\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \Lambda (\boldsymbol{x}_i - \boldsymbol{x}_j)}{2}\right), \quad \Lambda = diag(\lambda_1, ..., \lambda_{x_{np}}) \tag{2.27}$$

Accordingly the GP regression need to learn the hyperparameters in $\boldsymbol{\theta}$, from $\bar{X}$ and $\bar{\boldsymbol{y}}$:

$$\theta = [\sigma_{\mathbf{c}}^2, \lambda_{\mathbf{1:x_{np}}}] \in \mathbb{R}^{\mathbf{x_{dim}+1}} \tag{2.28}$$

Further $\bar{C} \in \mathbb{R}^{np \times np}$ is the covariance matrix with entries:

$$\bar{C}_{i,j} = c(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.29}$$

and

$$\boldsymbol{c}^T = [c(\boldsymbol{x}, \boldsymbol{x}_1), c(\boldsymbol{x}, \boldsymbol{x}_2), ..., c(\boldsymbol{x}_i, \boldsymbol{x}_{np})] \in \mathbb{R}^{np \times 1} \tag{2.30}$$

The parameters are learned by maximizing the log-marginal likelihood:

$$\mathcal{L}(\boldsymbol{\theta}, \bar{X}, \bar{\boldsymbol{y}}) = -\frac{1}{2}\bar{\boldsymbol{y}}^T \boldsymbol{M}(\boldsymbol{\theta}, \bar{X})^{-1}\bar{\boldsymbol{y}} - \frac{1}{2}\log|\boldsymbol{M}(\boldsymbol{\theta}, \bar{X})| - \frac{n}{2}\log(2\pi) \tag{2.31}$$

where

$$\boldsymbol{M}(\boldsymbol{\theta}, \bar{X}) = \bar{C} + \sigma_v^2 I_{np} = \bar{C}(\boldsymbol{\theta}, \bar{X}) + \sigma_v^2 I_{np} \tag{2.32}$$

As can be observed the log-marginal likelihood consists of three terms. First of all the data-fit part $-\frac{1}{2}\bar{\boldsymbol{y}}^T \boldsymbol{M}(\boldsymbol{\theta}, \bar{X})^{-1}\bar{\boldsymbol{y}}$, which is the only term involving the observed targets. Then there is a complexity penalty, $\frac{1}{2}\log|\boldsymbol{M}(\boldsymbol{\theta}, \bar{X})|$, which depends on

the covariance matrix and the noise. The last part $\frac{n}{2}\log(2\pi)$ is a normalization constant. For further details on the model selection, the reader is referred to chapter 5.4 in [2].

To find the optimal hyperparameters:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{argmax}\, \mathcal{L}(\boldsymbol{\theta}, \bar{X}, \bar{\boldsymbol{y}}) \tag{2.33}$$

one can use both deterministic and stochastic methods. The maximization problem is both nonlinear and non-convex. For instance one can use an optimization algorithm, which uses the partial derivatives of the marginal likelihood with respect to the hyperparameters, to find the optimal hyperparameters. Using the following properties:

$$\frac{\partial}{\partial\theta}K^{-1} = -K^{-1}\frac{\partial K}{\partial\theta}K^{-1}, \quad \frac{\partial}{\partial\theta}\log(K) = tr\left(K^{-1}\frac{\partial K}{\partial\theta}\right) \tag{2.34}$$

the partial derivatives can easily be computed as

$$\frac{\partial}{\partial\theta_i}\mathcal{L}(\boldsymbol{\theta}, \bar{X}, \bar{\boldsymbol{y}}) = \frac{1}{2}\bar{\boldsymbol{y}}^T\boldsymbol{M}(\boldsymbol{\theta}, \bar{X})^{-1}\frac{\partial \boldsymbol{M}(\boldsymbol{\theta}, \bar{X})}{\partial\theta_i}\boldsymbol{M}(\boldsymbol{\theta}, \bar{X})^{-1}\bar{\boldsymbol{y}} - \frac{1}{2}tr\left(\boldsymbol{M}(\boldsymbol{\theta}, \bar{X})^{-1}\frac{\partial \boldsymbol{M}(\boldsymbol{\theta}, \bar{X})}{\partial\theta}\right) \tag{2.35}$$

GP can also compute the output distribution with more weight on the nearest input. Hence the predicted output is more influenced by the nearby input-output pairs from the training set. This flexibility is one of the main advantages of GP compared to fixed input-output structures based on parametric methods[9]. Consequently, GP is able to capture complex nonlinear relationships with the use of only a few parameters. Another key-point is GPs ability to handle measurements

with noise. This property comes in handy, regarding that most measurements in the industry consist of noise [11].

In the context of RTO, the goal of using GP is to estimate the mismatch between the true plant and the available model. As mentioned in section 2.5.3 the biggest challenge with MA is to accurately estimate the gradients in the modifiers. In GP the modifiers from the general MA framework are replaced by higher order regression functions.

For the sake of compact notation, GP estimation of some unknown function f will from now on be written as:

$$\boldsymbol{y} = (\mathcal{GP})^{f}(\boldsymbol{x}, \bar{X}, \bar{\boldsymbol{y}}) \tag{2.36}$$

with input-output pairs $\bar{X}$ and $\bar{\boldsymbol{y}}$ and new query input $\boldsymbol{x}$.

Finally the optimization problem in equations 2.17 - 2.19 with GP estimation becomes:

$$x_{k+1}^{*} = \max_{\boldsymbol{x}} \quad f(\mathbf{x}) + (\mathcal{GP})^{(f_k^p - f_k)} \tag{2.37}$$

s.t.

$$c_{i,k}(\boldsymbol{x}) = c_i(\boldsymbol{x}) + \epsilon^{c_i} = 0 \qquad i \in \epsilon \tag{2.38}$$

$$c_{i,k}(\boldsymbol{x}) = c_i(\boldsymbol{x}) + \epsilon^{c_i} \leq 0 \qquad i \in \mathcal{I} \tag{2.39}$$

where the modifiers are defined as following:

$$\epsilon^{c_i} = (\mathcal{GP})^{(c_{i,k}^p - c_{i,k})}$$

2.5.3.3 Gaussian process regression example

In this section, Gaussian process regression will be illustrated with a simple example. As discussed in section 2.5.3.2, the objective is to predict some unknown function based on some observations. In the following example the observed function is $f(x) = x \cdot \sin(x)$, but the measurements are influenced by an independent, identically distributed Gaussian distribution with zero mean. The disturbance is implemented using the communication toolbox in MATLAB. In order to predict the true response, GP regression is applied to 50 observations in the domain $x \in [0, 10]$. To demonstrate GP's properties, the regression method is set to also predict responses relatively far away from the observations. More precisely it is set to predict response for the interval $x \in [-10, 25]$.

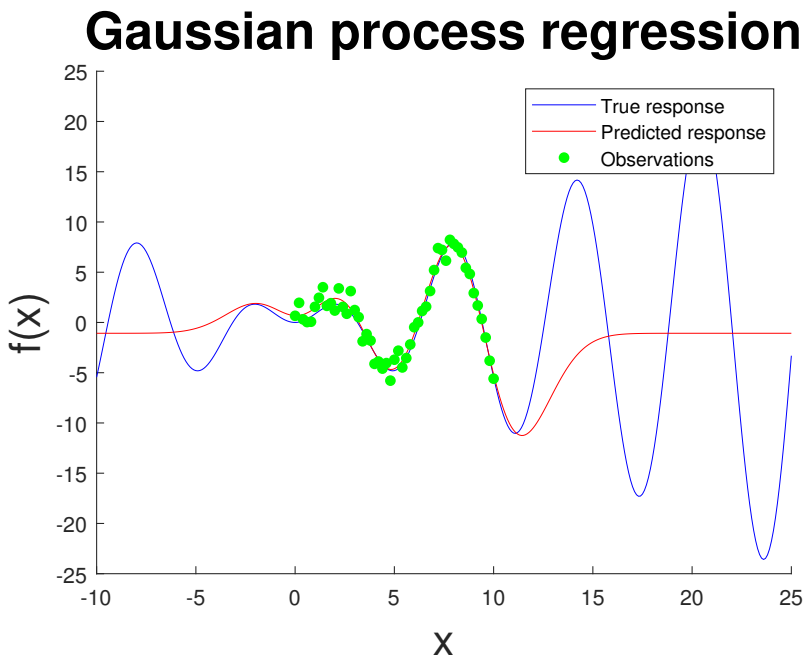


Figure 2.6: GP regression example

The results are illustrated in figure 2.6. The GP regression is implemented with the statistic and machine learning toolbox in MATLAB. As can be seen, the observed points are influenced by noise. However, the GP regression predicts the response quite well for responses close to the observations. This illustrates GPs ability to handle noise. On the other hand, it is far off to predict the response far away from the measurements. This is not a surprise, since there is no information about these points [2]. It can also be seen from the kernel, equation 2.27, that correlation between two points decays with the distance between the points. Hence closer points are expected to behave more similar.

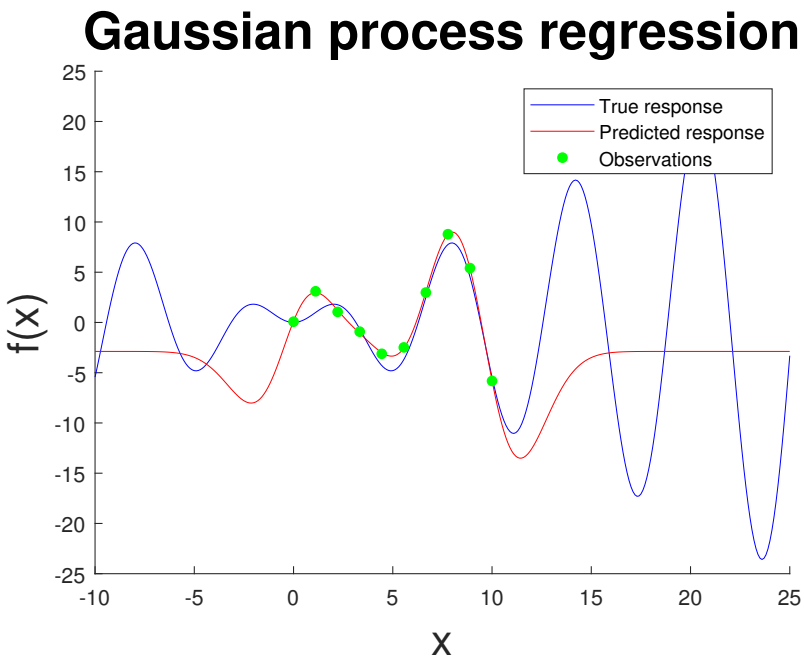


Figure 2.7: GP regression example with only 8 samples

Figure 2.7 illustrates the same GP regression example as before, but with only 8 observation points. The prediction seems to fit the training data very well, which is no surprise regarding that there are little training data available. This kind of overfitting can be a problem for noise with high variance. On the other hand, the regression is still quite good even with few samples, which is a very desirable property GP holds. This is an interesting trade-off between exploration and exploitation, which has to be taken into consideration when implementing GP[2].

2.6 Software

2.6.1 Fmincon

All the optimization problems presented in this project have been solved using a nonlinear programming solver, from the optimization toolbox in MATLAB. The particular solver used is called fmincon. To give a brief introduction on how fmincon works, a general nonlinear optimization problem will be defined:

$$\min_{x} \quad f(x)$$

$$s.t.$$

$$\begin{aligned} \boldsymbol{A}x &\leq \boldsymbol{b} \\ \boldsymbol{A}_{eq}x &= \boldsymbol{b}_{eq} \\ \boldsymbol{c}(x) &\leq 0 \\ \boldsymbol{c}_{eq}(x) &= 0 \\ lb \leq & x \leq ub \end{aligned}$$

where $f(x)$ can be a nonlinear objective function, and both $\boldsymbol{c}(x)$ and $\boldsymbol{c}_{eq}(x)$ are nonlinear constraints. The powerful solver, fmincon, will solve this problem, or show it to be infeasible, in finite amount of computation.

Chapter 3

Model of a two-well system

As mentioned in chapter 1 the objective in this thesis is to investigate an optimization problem for oil production. To simplify the analysis and understanding the optimization problem is investigated on an oil-well system with two wells, but the same principles yields for facilities with more wells. First the two-well system has to be modeled. The following sections will give an overview model. The model presented in this chapter is based on the work I did in the project thesis in fall 2018.

3.1 Two-well system

As mentioned earlier a two-well case, well a and well b, with a separator is considered. The two-well system is based on the system proposed by Grimholt and Skogestad in [12]. For simplicity, only the well valves, z_a and z_b, can be controlled and the top valve is fixed to fully open. Hence the system has two degrees of freedom.

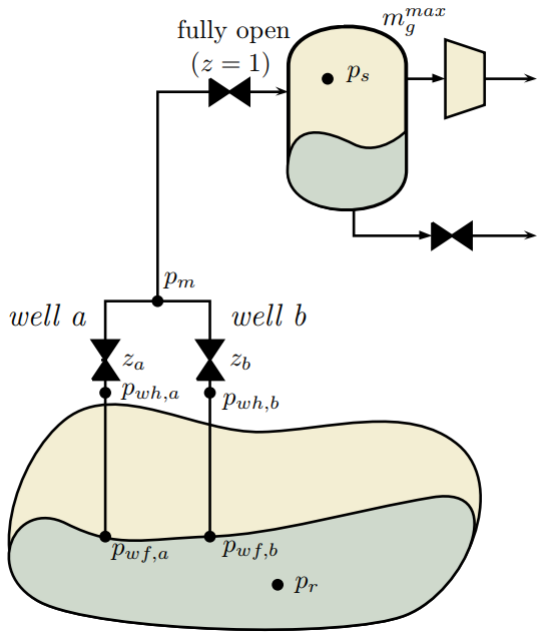


Figure 3.1: Sketch of the two-well system, based on [12]

As can be verified from figure 3.1 the system consists of three submodels: One

for reservoir inflow, a model for pressure drop through a vertical pipe and a final one for the flow across a valve. The three submodels will be described in detail and modeled in the following sections.

3.2 Reservoir inflow

The reservoir inflow is assumed to follow Fetkowichs quadratic deliverability equation[13]:

$$\dot{m}_o = k_o(p_r^2 - p_{wf}^2) \tag{3.1}$$

$$\dot{m}_w = k_w(p_r^2 - p_{wf}^2) \tag{3.2}$$

where $\dot{m}_o$ denotes the mass flow of the oil production and $\dot{m}_w$ is the mass flow of water from production. k_o and k_w are the flow coefficients for oil and water, respectively. p_r is the reservoir pressure and p_{wf} denotes the pressure in the pipe orifice in the reservoir. In addition to water and oil, the reservoir fluid consists of gas [14]. Combining the oil mass flow with GOR, gas oil ratio, the mass flow for gas production, $\dot{m}_g$, is obtained as follows:

$$\dot{m}_g = GOR \cdot \dot{m}_o \tag{3.3}$$

3.3 Modeling one phase pseudo fluid

Before modeling pressure drops, the three-phase fluid has to be modeled. The fluid from the reservoir consists of three parts, specifically oil, water, and gas. To simplify, the multiphase fluid will be approximated as a one-phase pseudo-fluid. Accordingly, no mixing volumes are assumed. Furthermore, oil and water are incompressible and the gas is assumed to follow the ideal gas law, described by

the following equation:

$$\rho_g^{ig} = \frac{pM_g}{RT} \tag{3.4}$$

where ρ_g^{ig} is the density, p is the pressure, M_g molar weight of the gas, R is the ideal gas constant and T is the Temperature. Further, the one phased pseudofluid is approximated by its volumetric average:

$$\rho_{mix}(p) = \frac{\dot{m}_o + \dot{m}_g + \dot{m}_w}{\frac{\dot{m}_o}{\rho_o} + \frac{\dot{m}_w}{\rho_w} + \frac{\dot{m}_g}{\rho_g^{ig}(p)}} \tag{3.5}$$

where ρ_o and ρ_w are the densities of the oil and water, respectively.

3.4 Pressure drop across valve

The mass flow across a valve is given by the following valve equation:

$$\dot{m}_o + \dot{m}_w + \dot{m}_g = f(z)C_d A\sqrt{\rho_{avg}(p_2 - p_1)} \tag{3.6}$$

where f(z) is describing the valve characteristics, with z between 0, when fully closed, and 1, when fully open. C_d is the valve constant, A is the cross section area of the pipe and the pressure on each side is denoted by p_1 and p_2, as illustrated in figure 3.2. Hence the pressure drop across the valve is given by $\Delta p = p_2 - p_1$. From equation 3.5 it can be observed that the density for the one-phase fluid is dependent on the pressure. Therefore, ρ_{avg} in equation 3.6 is approximated by the average of the density on each side of the valve, given by the following equation:

$$\rho_{avg} = \frac{1}{2}(\rho_{mix}(p_1) + \rho_{mix}(p_2)) \tag{3.7}$$

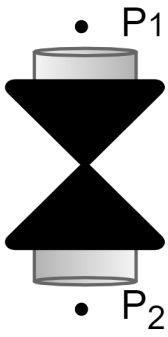


Figure 3.2: The pressure drop across the valve

Regarding that the manifold pressure, p_m, is set by the designer the pressure on the upper side, p_1 of the valve can easily be calculated. Further the pressure on the downside of the valve, p_2, will be an equation with respect to the mass flows and f(z). In this model the valve characteristics are assumed to be linear as follows:

$$f(z) = z, \quad z \in [0, 1] \tag{3.8}$$

By combining equations 3.6, 3.7 and 3.8 with some manipulations, the following second order equation for p_2 is obtained:

$$\begin{aligned} p_2^2(\rho_{mix,1}\beta + \dot{m}_{tot}) + p_2(\rho_{mix,1}\alpha - p_1(\rho_{mix,1}\beta + \dot{m}_{tot}) - \frac{2}{\gamma}\dot{m}_{tot}^2\beta) \\ -(p_1\rho_{mix,1}\alpha + \frac{2}{\gamma}\dot{m}_{tot}^2\alpha) = 0 \end{aligned} \tag{3.9}$$

$$\alpha = \frac{RT\dot{m}_g}{M_g} \tag{3.10}$$

$$\beta = \frac{\dot{m}_o}{\rho_o} + \frac{\dot{m}_w}{\rho_w} \tag{3.11}$$

$$\dot{m}_{tot} = \dot{m}_o + \dot{m}_w + \dot{m}_g \tag{3.12}$$

$$\rho_{mix,1} = \frac{\dot{m}_{tot}}{\frac{\alpha}{p_1} + \beta} \tag{3.13}$$

$$\gamma = (f(z)C_dA)^2 \tag{3.14}$$

As can be seen, the valve pressure drop is obtained by solving equation 3.9.

3.5 Pressure drop through vertical pipe

To estimate the pressure drop through a vertical pipe the stationary mechanical energy balance is used. No slip between phases and no friction is assumed. In addition to that work and kinetic energy are neglected. Hence the mechanical energy balance is as follows:

$$dp = \rho_{mix} g dh \tag{3.15}$$

where g is the gravitation acceleration. Combining 3.5, 3.10, 3.11 and 3.12 the following equation is obtained:

$$dp = \frac{\dot{m}_{tot}}{\frac{\alpha}{p} + \beta} g dh \tag{3.16}$$

Integrating 3.16 from (p_1, h_1) to (p_2, h_2) the relation becomes

$$\beta(p_2 - p_1) + \alpha ln(\frac{p_2}{p_1}) = \dot{m}_{tot} g(h_2 - h_1) \tag{3.17}$$

As can be observed equation 3.17 cannot be solved exactly for p_2, due to the logarithm. For this reason, a serial expansion of the natural logarithm is used:

$$ln(\frac{p_2}{p_1}) = ln(1 + \frac{p_2 - p_1}{p_1}) \approx \frac{p_2 - p_1}{p_1} \tag{3.18}$$

Combining 3.17 and 3.18, and using $\Delta h = h_2 - h_1$, the pressure p_2 can be expressed as

$$p_2 = p_1 + \frac{\dot{m}_{tot} g p_1 \Delta h}{\alpha + \beta p_1} \tag{3.19}$$

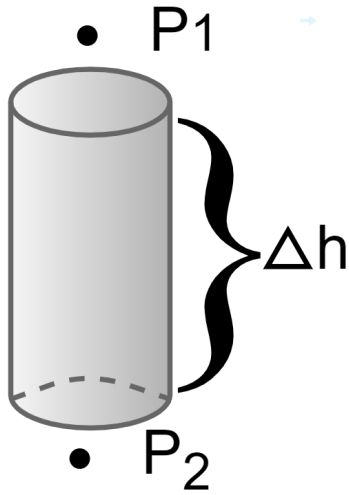


Figure 3.3: Pressure drop across vertical pipe

Chapter 4

Static optimization in a two-well system

As the dynamics of the system are modeled, in chapter 3, the optimization problem can be formulated. From section 2.1 we know that the decision variables have to be determined. Further, an objective function and constraints have to be formulated.

4.1 Formulating the optimization problem

As mentioned in section 3.2 the reservoir fluid consists of oil, water, and gas. Regarding that oil is the most valuable part, the objective is to maximize oil production. In addition to that, an oil production facility usually have capacity constraints on how much water and gas it can handle. In this thesis, it is assumed that the facility can handle an infinite amount of water, but only a certain amount

of gas. Therefore, the mass flows of oil and gas, $\dot{m}_o$ and $\dot{m}_g$, respectively have to be described by the decision variables. From equation 3.6 it can be observed that the mass flows are dependent of the valve opening function $f(z) = z$. When z is fully open there will be maximum mass flow. Contrary when z is fully closed, $z = 0$, there will be no mass flow. Hence, the problem has two degrees of freedom, one for the valve opening in each well. With this in mind the decision variables can be chosen as:

$$x^T = [x_1 \quad x_2] = [z_1 \quad z_2]$$

where the subscripts denote which well the variables belong to.

In view of that the objective is to maximize oil production the objective function will be as simple as:

$$\max = \dot{m_{o,1}} + \dot{m_{o,2}} \tag{4.1}$$

Before formulating the problem, some restrictions have to be taken into consideration. Certainly, the physics behind the reservoir mass flow, described by equation 3.1, have to be respected. Furthermore, the gas capacity constraint, from here denoted as $\dot{m}_{g,max}$, have to be considered. This constraint sets an upper limit on how much gas that can be produced.

Before formulating the problem, the mass flows and other variables will be pre-

sented with respect to the decision variables.

$$\dot{m}_{o,i} = K_{o,i}({P_r}^2 - \mathbf{P^2_{wf_i}}(x_i)) \qquad i = [1,2] \tag{4.2}$$

$$\dot{m}_{w,i} = K_{w,i}({P_r}^2 - \mathbf{P^2_{wf_i}}(x_i))) \qquad i = [1,2] \tag{4.3}$$

$$\dot{m}_{g,i} = GOR_i K_{o,i}({P_r}^2 - \mathbf{P^2_{wf_i}}(x_i))) \qquad i = [1,2] \tag{4.4}$$

$$\dot{m}_{tot,i} = \dot{m}_{o,i} + \dot{m}_{w,i} + \dot{m}_{g,i} \qquad i = [1,2] \tag{4.5}$$

$$\alpha_i = \frac{RT\dot{m}_{g,i}}{M_g} \qquad i = [1,2] \tag{4.6}$$

$$\beta_i = \frac{\dot{m}_{o,i}}{\rho_o} + \frac{\dot{m}_{w,i}}{\rho_w} \qquad i = [1,2] \tag{4.7}$$

$$\rho_{avg,i} = \frac{1}{2}\left(\frac{\dot{m}_{tot,i}}{\frac{\alpha_i}{p_m} + \beta_i} + \frac{\dot{m}_{tot,i}}{\frac{\alpha_i}{P_{wh,i}} + \beta_i}\right) \qquad i = [1,2] \tag{4.8}$$

Where $\mathbf{P^2_{wf_i}}(x_i)$ denotes the inflow pressure as a function of the valve opening, x_i. Finally, using equiation 3.6 and 3.19 , the optimization problem can be formulated:

$$\max \quad \dot{m}_{o,1} + \dot{m}_{o,2} \tag{4.9}$$

$$s.t.$$

$$\mathbf{P_{wh_1}} = p_m + \frac{\dot{m}^2_{tot,1}}{\rho_{\mathbf{avg,1}} \cdot (x_1 C_d A)^2} \tag{4.10}$$

$$\mathbf{P_{wf_1}} = \mathbf{P_{wh_1}} + \frac{\dot{m}_{tot,2} \cdot gh \cdot \mathbf{P_{wh_1}}}{\alpha_1 + \beta_1 \cdot \mathbf{P_{wh_1}}} \tag{4.11}$$

$$\mathbf{P_{wh_2}} = p_m + \frac{\dot{m}^2_{tot,2}}{\rho_{\mathbf{avg,2}} \cdot (x_2 C_d A)^2} \tag{4.12}$$

$$\mathbf{P_{wf,_2}} = \mathbf{P_{wh_2}} + \frac{\dot{m}_{tot,2} \cdot gh \cdot \mathbf{P_{wh_2}}}{\alpha_2 + \beta_2 \cdot \mathbf{P_{wh_2}}} \tag{4.13}$$

$$\dot{m}_{g,1} + \dot{m}_{g,2} \leq \dot{m}_{g,max} \tag{4.14}$$

4.2 Solving the optimization problem

4.2.1 Problem set-up

It is assumed that all observations are perfectly measured without disturbance. The presented static optimization problem of the two-well system is solved for different GOR levels in the wells. The change in GOR settings is done to illustrate the relation between GOR and the amount of oil production. The numerical values and units of all the well constants in the optimization problem can be found in section A.1, which are based on the facility specifications from the case study in [12]. The problems were solved using the optimization toolbox in MATLAB as describes in section 2.6.1.

4.2.2 Results and comments

The surface plot in figure 4.1 presents the solutions for the optimization problem described in equation 4.9. The problem was solved 625 times, where the GORs in equation 4.5 were changed each time. Consequently the gas production constraint in equation 4.14 were modified. The produced oil on the z-axis is the objective from equation 4.9 with the optimal valve openings solved by the SQP-algorithm, described in section 2.4.1.

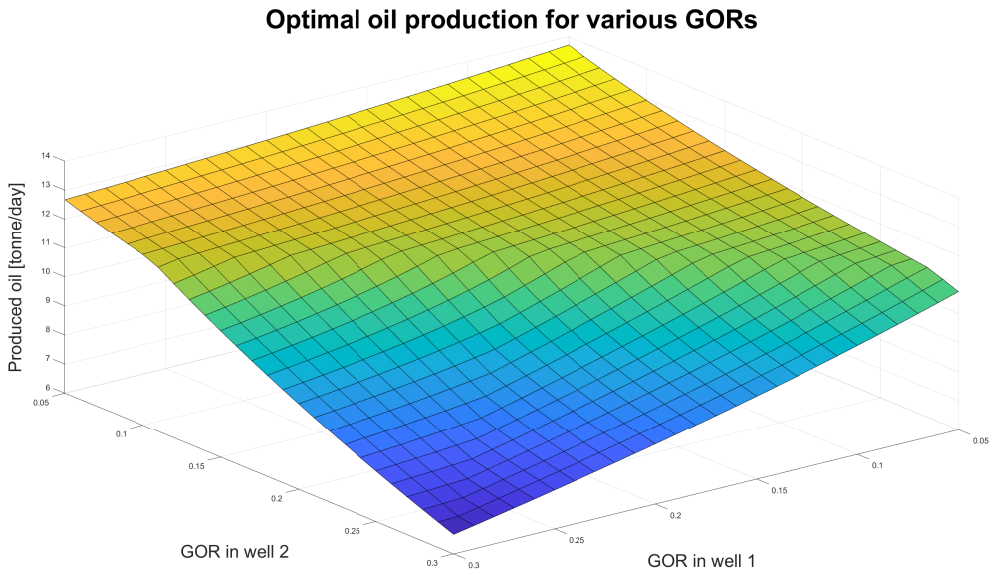


Figure 4.1: Optimal oil production for different Gas-oil ratios

As we can see from figure 4.1 the oil production is low for relatively high GORs. High GOR leads to a relatively high production of gas. This will reduce oil production since the production facility can only handle a certain amount of gas. This relation is very interesting, regarding that the GORs have an important impact on how much we want to tune the valves for the different wells.

As discussed in section 4.1, the two-well static optimization problem has two degrees of freedom, one for each of the two valve openings. These valve openings decide the lower flow pressures, P_{wf_1} and P_{wf_2}, which are the only variables in the objective function. For this reason, in most of the further analysis, the graphs will be visualized with respect to the lower flow pressures. Figure 4.2 shows the

contour plot of the produced oil with respect to the lower flow pressures. For our analysis, the area of interest is the feasible area, marked in the upper right corner. The feasible area is determined by the set of constraints, and will vary with variations in GOR. Figure 4.3 illustrates how the feasible area changes for different GOR.

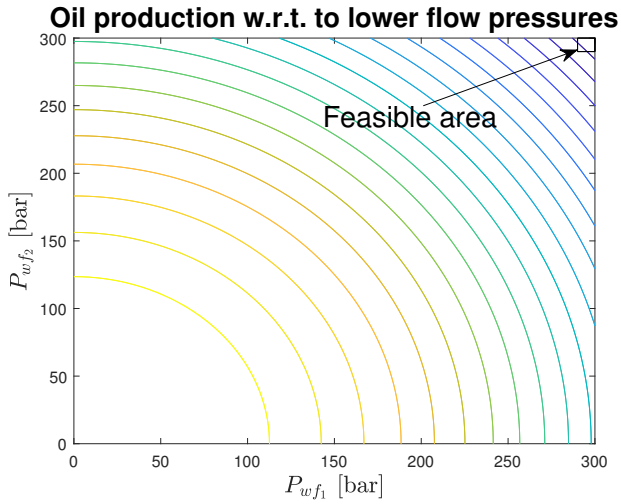
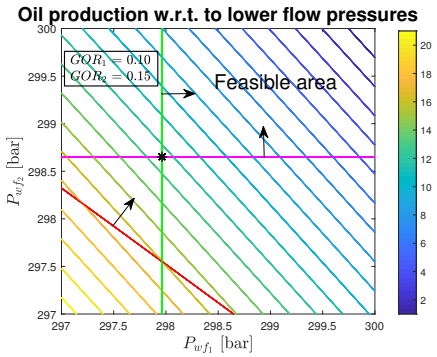


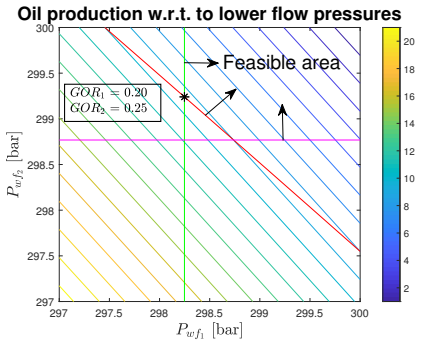
Figure 4.2: Contour plot of produced oil w.r.t. lower flow pressures.

As expected the contour lines in figure 4.2 show that the production in oil increases with decrease in the lower flow pressures, P_{wf_1} and P_{wf_2}. This can also be verified from the objective function in equation 4.9.

As mentioned above, the area of interest is the feasible area. Figure 4.3 shows contour plots scaled for the feasible region.



(a) Contour plot of oil production, and constraints for $GOR_1 = 0.10$ and $GOR_2 = 0.15$.



(b) Contour plot of oil production, and constraints for $GOR_1 = 0.20$ and $GOR_2 = 0.25$.

Figure 4.3: Contour plots of oil production w.r.t. lower flow pressures

As can be observed the gas-capacity constraints differs in both plots, which is due to the difference in gas-oil-ratios in the two problems. If one takes a closer look, both the vertical and the horizontal constraints differ in the two problems. This is to illustrate the importance of the gas-oil- ratios in the optimization problem, which will be very important for the analysis in the next chapter.

As mentioned above, and observed from the plots and equations, the objective increases with a decrease in lower flow pressures. Hence, all the inequality constraints will push the lower flow pressures up in order to constraint the maximization problem. The gas-capacity constraints are illustrated by the red sloped lines. For higher gas capacity the red gas capacity constraint will parallel shift down, to allow lower P_{wf}. This makes perfect sense with our intuition, that the more gas the facility can handle the more oil can be produced. The steepness of the constraint is determined by the ratio between the GORs in both wells, which can be verified from equation 4.14.

In reality, it can be very difficult to model and predict the GOR in the different wells. Therefore, there will be mismatches between the model and plant. In this section, only the model was simulated, but when implementing the system it will be ideal to include process observations and regulate it to cope with model errors. In the next chapter, we will use the model developed in this chapter in a MA-RTO framework, to investigate if it can cope with plant-model mismatches. The analysis in the next chapter will include plots like in figure 4.3, where the inequality constraints determine the feasible area in the same way as here.

Chapter 5

MA-RTO of a two-well system with uncertain parameters

As discussed in section 4.2.2, GOR is a very important parameter when solving the optimization problem. GOR can be modeled as a function of reservoir pressures and lower flow pressure. Thus GOR uncertainty can be a result of disturbances in measurements of the pressures. Disturbances in the process, sudden changes in GOR or reservoir pressures will also cause the wrong GOR in the model, which can result in a suboptimal optimum for the plant when solving the optimization problem. As described earlier this uncertainty is called plant-model mismatch. In this section, we will try to solve this problem using the Modifier Adaptation approach, which is described in section 2.5.2.

The Modifier Adaptation approach which has been used in this thesis is described in detail in [1]. First of all two models have to be developed in order to study the plant-model mismatch. One model for the plant is needed, which will be updated by the observations, referred to as plant. Further one model for

the optimization layer, using the modifier adaptation framework, referred to as the model. Moreover, by applying high-order correction terms the model-based optimization is supposed to reach the plant optimum. The correction terms are referred to as the MA modifiers.

As introduced in section 2.5.2, we will apply two variants of the Modifier Adaptation approach. First, the MA method with gradient estimation using past RTO points will be applied. In this case, the zeroth order correction modifier is added. The zeroth order modifiers correspond to the difference between the plant and model values in each iteration. In other words, the difference between the most recent plant measurement and model values are used in the optimization. Similarly, the differences between the plant gradients and the gradients of the model are applied as the first order modifiers. Secondly, the MA-RTO method using Gaussian processes will be applied. In this variant, the zeroth and first order modifiers from the previous method are replaced by high-order regression functions [2].

5.1 Formulating the MA optimization problem

In this thesis, the focus is to study the plant-model mismatch. Therefore, in the MA scheme, it is assumed that both cost functions and constraint functions are measured without noise. Hence the plant-model mismatch only comes from the parametric uncertainty of the gas-oil ratios. However, we will also simulate cases where there is measurement noise to see how the MA approach handles this.

Applying the MA framework from section 2.5.2 to the static optimization problem

in equation 4.9 to 4.14, the modifier adaptation optimization problem becomes:

$$x^*_{k+1} = \max_{\boldsymbol{x}} \quad K_{o_1}({P_r}^2 - \mathbf{P}^2_{\mathbf{wf_1}}(x_1)) + K_{o_2}({P_r}^2 - \mathbf{P}^2_{\mathbf{wf_2}}(x_2)) + \epsilon^J_k + \lambda^J_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \tag{5.1}$$

$$s.t.$$

$$P_{wh_1} = p_m + \frac{\dot{m}^2_{tot,1}}{\rho_{\mathbf{avg}}(P_{wh_1}) \cdot (x_1 C_d A)^2} + \epsilon^{C_1}_k + \lambda^{C_1}_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \tag{5.2}$$

$$P_{wf_1} = P_{wh_1} + \frac{\dot{m}_{tot,2} \cdot gh \cdot P_{wh_1}}{\alpha_1 + \beta_1 \cdot P_{wh_1}} + \epsilon^{C_2}_k + \lambda^{C_2}_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \tag{5.3}$$

$$P_{wh_2} = p_m + \frac{\dot{m}^2_{tot,2}}{\rho_{\mathbf{avg}}(P_{wh_2} \cdot (x_2 C_d A)^2} + \epsilon^{C_3}_k + \lambda^{C_3}_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \tag{5.4}$$

$$P_{wf_2} = P_{wh_2} + \frac{\dot{m}_{tot,2} \cdot gh \cdot P_{wh_2}}{\alpha_2 + \beta_2 \cdot P_{wh_2}} + \epsilon^{C_4}_k + \lambda^{C_4}_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \tag{5.5}$$

$$\dot{m}_{g,1} + \dot{m}_{g,2} + \epsilon^{C_5}_k + \lambda^{C_5}_k \begin{bmatrix} x_1 & - & x^k_1 \\ x_2 & - & x^k_2 \end{bmatrix} \leq \dot{m}_{g,max} \tag{5.6}$$

where the MA modifiers are λ^J_k, $\epsilon^{C_i}_k$ and $\lambda^{C_i}_k$, for $i = [1, 2, 3, 4, 5]$. The implementation of the modifiers will differ from estimation method to estimation method. As described in section 2.5.2, $\epsilon^{C_i}_k$ are the zero order modifiers, while λ^J_k and $\lambda^{C_i}_k$

are the first order modifiers. In general the modifiers are defined as:

$$\epsilon_k^J = \widetilde{J}_{p,k} - \widetilde{J}_k \tag{5.7}$$

$$\lambda_k^J = (1 - K^J)\lambda_{k-1}^J + K^J(\widehat{\nabla \mathbf{J}}_{\mathbf{p,k}} - \widehat{\nabla \mathbf{J}}_{\mathbf{k}}) \tag{5.8}$$

$$\epsilon_k^{C_i} = \widetilde{C}_{p,i,k} - \widetilde{C}_{i,k} \qquad i = [1, 2, 3, 4, 5] \tag{5.9}$$

$$\lambda_k^{C_i} = (1 - K^{C_i})\lambda_{k-1}^{C_i} + K^{C_i}(\widehat{\nabla \mathbf{C}}_{\mathbf{p,i,k}} - \widehat{\nabla \mathbf{C}}_{\mathbf{i,k}}) \qquad i = [1, 2, 3, 4, 5] \tag{5.10}$$

$\widetilde{C}_{p,i,k}$ is the measurement of constraint i for the plant model in iteration k. Thus the plant constraints $\widetilde{C}_{p,i,k}$ for i=1 to i=5 are the plant measurements of P_{wh_1}, P_{wf_1}, P_{wh_2}, P_{wf_2} and $\dot{m}_{g,1} + \dot{m}_{g,2}$, respectively. Similarly $\widetilde{C}_{i,k}$ are the model values for the same constraints. Moreover, by using these values the derivatives of the model constraints, $\widehat{\nabla \mathbf{C}}_{\mathbf{i,k}}$, and plant constraint gradients, $\widehat{\nabla \mathbf{C}}_{\mathbf{p,i,k}}$, are estimated. In addition, the plant cost gradient, $\widehat{\nabla \mathbf{J}}_{\mathbf{p,i,k}}$, is also estimated from the measurements. The biggest challenge when estimating these are noise, due to the fact that the gradients cannot be measured directly. The implementations to estimate these using FDA and using Gaussian Process regression are described in section 5.2 and section 5.3, respectively.

5.2 Gradient estimation using FDA with past RTO points

In [1], multiple methods are presented. One of these is the finite-difference approximation using past RTO points. Initialization of this technique requires the number of operating points equal to the number of inputs in the problem, plus one to estimate the gradients. Hence it will require 3 operating points in this case, regarding that this problem has 2 inputs. One can obtain these points by perturbating each input around the current operating point two times. Thus the following matrices can be constructed

$$U_k = \begin{bmatrix} x_k^1 - x_{k-1}^1 & x_k^1 - x_{k-2}^1 \\ x_k^2 - x_{k-1}^2 & x_k^2 - x_{k-2}^2 \end{bmatrix}$$

$$\delta\widetilde{J}_{p,k}{}^T = \begin{bmatrix} \widetilde{J}_{p,k} - \widetilde{J}_{p,k-1} & \widetilde{J}_{p,k} - \widetilde{J}_{p,k-2} \end{bmatrix}$$

$$\delta\widetilde{C}_{p,i,k}{}^T = \begin{bmatrix} \widetilde{C}_{p,i} - \widetilde{C}_{p,i,k-1} & \widetilde{C}_{p,i,k} - \widetilde{C}_{p,i,k-2} \end{bmatrix}$$

Finally the plant gradients become

$$\widehat{\nabla \mathbf{J}}_{\mathbf{p,k}} = \delta\widetilde{J}_{p,k}{}^T (U_k)^{-1}$$

$$\widehat{\nabla \mathbf{C}}_{\mathbf{p,i,k}} = \delta\widetilde{C}_{p,i,k}{}^T (U_k)^{-1} \qquad i = [1, 2, 3, 4, 5]$$

where k indicates which iteration the values belong to.

5.2.1 The algorithm

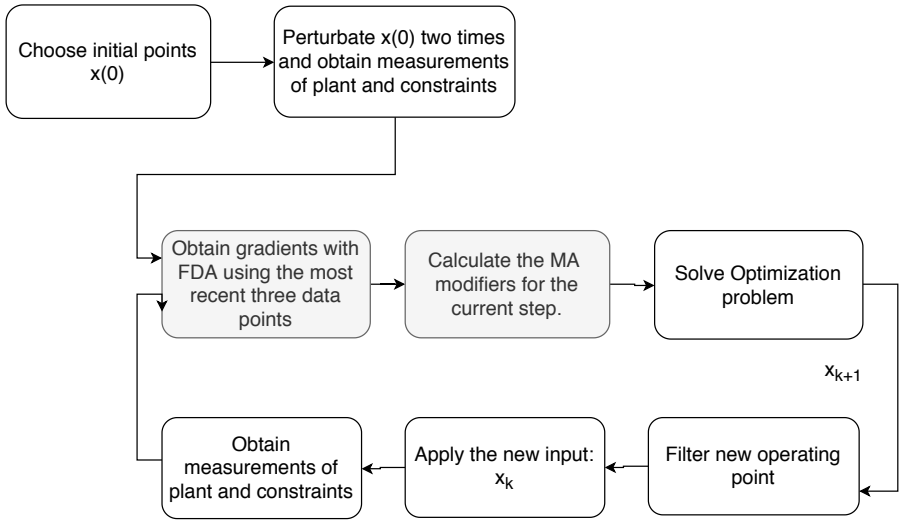


Figure 5.1: Flowchart of MA-RTO with FDA gradient estimation

1. Choose an initial point $[x_0^1 \quad x_0^2]$
2. Perturbate x(0) two times to get enough data points for initialization.
3. Obtain the gradients with FDA using the recent three data points.
4. Calculate the current Modifiers: λ_k^J, $\lambda_k^{C_i}$
5. Solve the static optimization problem in equation 5.1 and find the next optimal input, x_{k+1}.
6. Apply the new input , x_{k+1}, and measure the outputs.
7. Return to point 3.

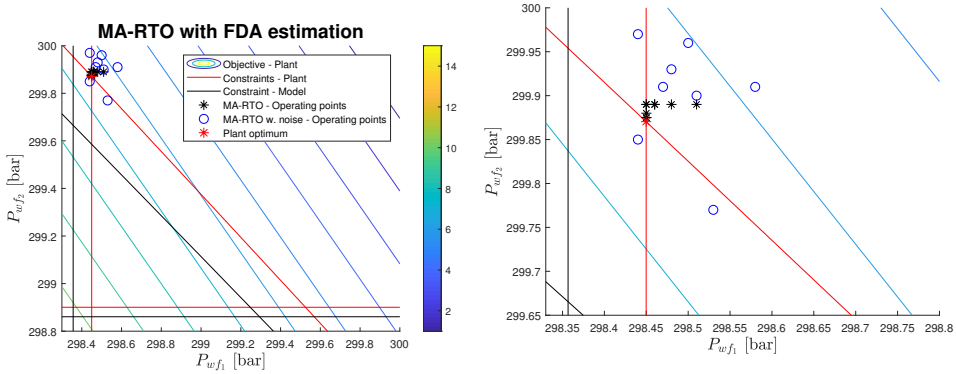
From the algorithm, one can observe that the perturbation is only used in the finite-difference approximation to estimate the initial gradient. Further, it is solving a static optimization problem in every iteration. Accordingly, it is applying the new valve input after each iteration and solving the problem with the most recent data points. Hence the modifiers are also updated after each input, using the new gradient estimations from the finite-difference approximation.

5.2.2 Simulations and results

To investigate how the MA-RTO with FDA gradient estimation handles plant-model mismatch, the optimization has been applied to four different cases, where the degree of plant-model mismatch varies from case to case. In addition to that, all cases have been simulated two times each, one time without measurement noise and one with measurement noise. The measurement noise has been implemented as an identically distributed Gaussian distribution with zero mean and unit variance.

The modeled-GOR and plant-GOR in the following cases are mentioned in the headlines on the form $GOR_{plant/model} = [GOR_{well_1} \ GOR_{well_2}]$.

5.2.2.1 Case 1: $GOR_{plant} = [0.30 \ \ 0.40]$ - $GOR_{model} = [0.25 \ \ 0.35]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

(b) A closer view of the operating points for the simulations with and without measurement noise.

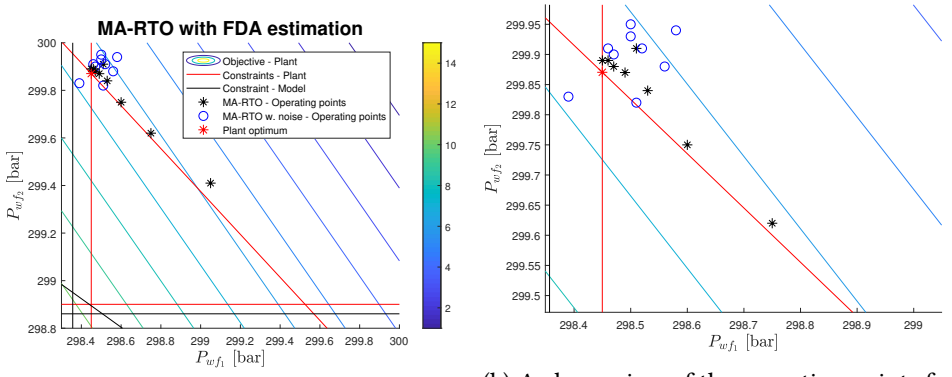
Figure 5.2: MA-RTO simulations using FDA gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \ \ 0.40]$ and $GOR_{model} = [0.25 \ \ 0.35]$. MA-FDA corrected iterations are plotted for both simulations, with and without measurement noise.

In this case, the model underestimates the amount of gas in the plant. This can be observed by comparing the plant- and model-constraints. To clarify the feasible area for the plant is on the right side of the plant-constraints. Similarly, the feasible area for the model is on the right side of the model-constraints. The model-constraints are left-shifted compared to the plant-constraint the model, indicating that it allows lower P_{wf}, which means lower GOR, which can be verified by the gas capacity constraint in equation 4.14. However, it can be observed that the RTO performs well to cope with the mismatches. Especially the case without measurement noise have great performance, regarding its convergence to the plant optimum in a few iterations. Moreover, the operating points are

on the feasible side. On the other hand, the RTO with measurement noise has weaker performance. In fact, it is slow and oscillatory. Furthermore, it has three operating points outside the feasible area of the plant.

All in all the MA-RTO with FDA estimation without measurement noise performs very well to cope with the GOR mismatches in this case, including both fast and feasible-side convergence. As discussed in section 2.5.2, MA-RTO has its weaknesses when noise is present, thus as expected the algorithm is not sufficient in the noisy case.

5.2.2.2 Case 2: $GOR_{plant} = [0.30 \ \ 0.40]$ - $GOR_{model} = [0.15 \ \ 0.30]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

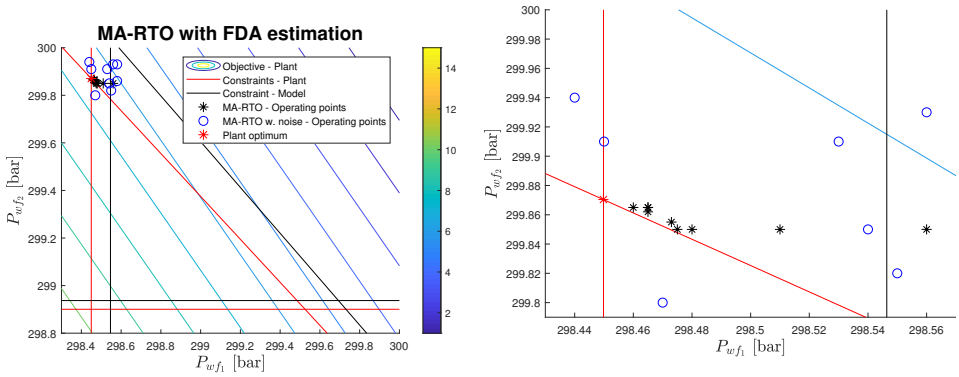
(b) A closer view of the operating points for the simulations with and without measurement noise.

Figure 5.3: MA-RTO simulations using FDA gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \ \ 0.40]$ and $GOR_{model} = [0.15 \ \ 0.30]$. MA-FDA corrected iterations are plotted for both simulations, with and without measurement noise.

As in case 1, the model in case 2 also underestimates the amount of gas in the plant, but this time the plant-model mismatch is greater in both wells. However, in the case without measurement noise, we can observe that the RTO performs quite well to cope with the mismatches. Even though it is not exactly on the plant optimum, it is practically in the optimum, regarding that the objective is just slightly better in the plant optimum compared to the convergence point of the MA-RTO, which can be observed from the contour lines. Then again, the case with measurement noise performance weaker. At the same time, some of the operating points are quite close to the plant-optimum.

After all, the MA-RTO with FDA estimation without measurement noise performs adequately to cope with the GOR mismatches in this case, including both fast and feasible-side convergence. Not surprisingly, the case with measurement noise was not sufficient.

5.2.2.3 Case 3: $GOR_{plant} = [0.30 \ \ 0.40]$ - $GOR_{model} = [0.36 \ \ 0.45]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

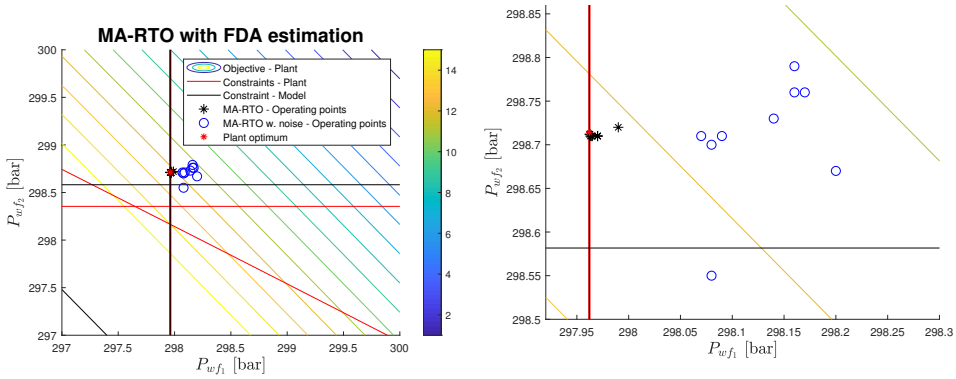
(b) A closer view of the operating points for the simulations with and without measurement noise.

Figure 5.4: MA-RTO simulations using FDA gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \ \ 0.40]$ and $GOR_{model} = [0.36 \ \ 0.45]$. MA-FDA corrected iterations are plotted for both simulations, with and without measurement noise.

Different from the previous two cases, this model overestimates the amount of gas in the plant. In other words, we can produce more oil from the plant-wells, with respect to the gas-capacity constraint, than the model proposes. Consequently, the MA-RTO operating points crosses the model-constraint to cope with the mismatches. The MA-RTO without measurement noise performs quite good and converges very close to the plant optimum. As in case 2, the algorithm practically converges to the optimum. Then again, the algorithm struggles in the case with measurement noise. As a matter of fact, it is even more oscillatory than in the previous cases.

All in all the MA-RTO without noisy measurements performs satisfactory, with both fast and feasible-side convergence. On the other hand, the performance in the case with measurement noise was not sufficient.

5.2.2.4 Case 4: $GOR_{plant} = [0.10 \;\; 0.20]$ - $GOR_{model} = [0.10 \;\; 0.10]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

(b) A closer view of the operating points for the simulations with and without measurement noise.

Figure 5.5: MA-RTO simulations using FDA gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.10 \;\; 0.20]$ and $GOR_{model} = [0.10 \;\; 0.10]$. MA-FDA corrected iterations are plotted for both simulations, with and without measurement noise.

In this case, the GOR in the plant is significantly lower than in the previous three cases. The optimal valve opening for the plant in this case is "fully open" for both wells ($x_1 = x_2 = 1$). In other words, the production facility, in this case, will have no problem handling the gas production, since the oil-wells contains a small amount of gas. Another key point, the modeled GOR in $well_1$ coincides with the plant-GOR in the same well. This can be verified by observing that the vertical

model-constraint and the vertical plant-constraint correspond to each other.

As can be observed from figure 5.5b, the MA-RTO algorithm converges accurately and fast in the case without measurement noise. The algorithm performs excellently, including both fast and feasible-side convergence. On the contrary, the algorithm still struggles when there is measurement noise present.

5.3 Gradient estimation using Gaussian processes

Contrary to FDA using past RTO points, GP does not calculate the gradients at each RTO iteration. In fact, these gradients are replaced by GP regression functions that describe the plant-model mismatch. More accurately, the modifiers from equations 5.8 and 5.10 will not be used in the MA with GP estimation. Moreover, the zeroth- and first-order modifiers in equation 5.9 will be replaced by higher order GP regression functions, which estimates the plant-model mismatches. By adding these GP regression terms, the cost and constraint functions of the modified optimization problem locally match those of the plant. Finally, the optimization problem becomes:

$$x^*_{k+1} = \max_{\boldsymbol{x}} \quad K_{o_1}({P_r}^2 - \mathbf{P^2_{wf_1}}(x_1)) + K_{o_2}({P_r}^2 - \mathbf{P^2_{wf_2}}(x_2)) + (\mathcal{GP})^{(J_p - J)}(\boldsymbol{x}, \bar{X}) \quad (5.11)$$

$$s.t.$$

$$P_{wh_1} = p_m + \frac{\dot{m}_{tot,1}^2}{\rho_{\text{avg}}(P_{wh_1}) \cdot (x_1 C_d A)^2} + (\mathcal{GP})^{(c_{p,1}-c_1)}(\boldsymbol{x}, \bar{X}) \tag{5.12}$$

$$P_{wf_1} = P_{wh_1} + \frac{\dot{m}_{tot,2} \cdot gh \cdot P_{wh_1}}{\alpha_1 + \beta_1 \cdot P_{wh_1}} + (\mathcal{GP})^{(c_{p,2}-c_2)}(\boldsymbol{x}, \bar{X}) \tag{5.13}$$

$$P_{wh_2} = p_m + \frac{\dot{m}_{tot,2}^2}{\rho_{\text{avg}}(P_{wh_2} \cdot (x_2 C_d A)^2} + (\mathcal{GP})^{(c_{p,3}-c_3)}(\boldsymbol{x}, \bar{X}) \tag{5.14}$$

$$P_{wf_2} = P_{wh_2} + \frac{\dot{m}_{tot,2} \cdot gh \cdot P_{wh_2}}{\alpha_2 + \beta_2 \cdot P_{wh_2}} + (\mathcal{GP})^{(c_{p,4}-c_4)}(\boldsymbol{x}, \bar{X}) \tag{5.15}$$

$$\dot{m}_{g,1} + \dot{m}_{g,2} + \epsilon_k^{C_5} + (\mathcal{GP})^{(c_{p,5}-c_5)}(\boldsymbol{x}, \bar{X}) \leq \dot{m}_{g,max} \tag{5.16}$$

Hence, we will construct the following 6 GP regression functions:

$$(\mathcal{GP})^f(\boldsymbol{x}, \bar{X}, \boldsymbol{y}) \qquad f \in \left\{J_p - J, c_{p,1} - c_1, ..., c_{p,5} - c_5\right\} \tag{5.17}$$

where $\bar{X}$ and $\boldsymbol{y}$ are all the input-output pairs for the functions we want to predict.

5.3.1 The algorithm

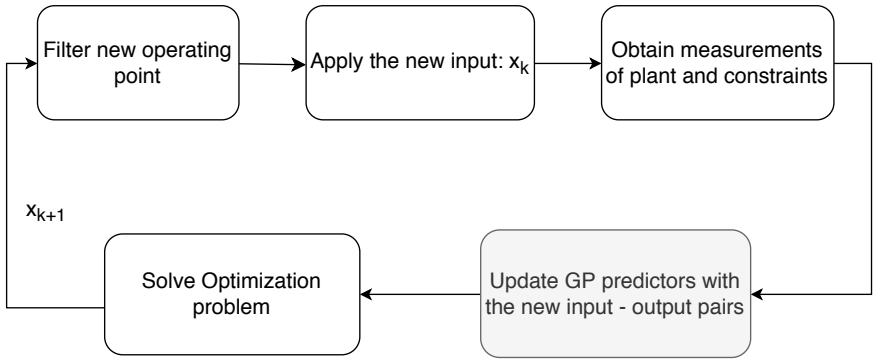


Figure 5.6: Flowchart of MA-RTO with GP estimation

1. Train GP predictors $(\mathcal{GP})^f$ in equation 5.17 using initial data sets $(\bar{X}^0, \boldsymbol{y}^0)$
2. Solve the optimization problem in equation 5.8.
3. Filter new operating point $\boldsymbol{x_{k+1}}$
4. Apply new input, $\boldsymbol{x_{k+1}}$
5. Obtain measurements of the cost and constraint functions and update the data sets $(\bar{X}, \boldsymbol{y})$.
6. Return to point 2.

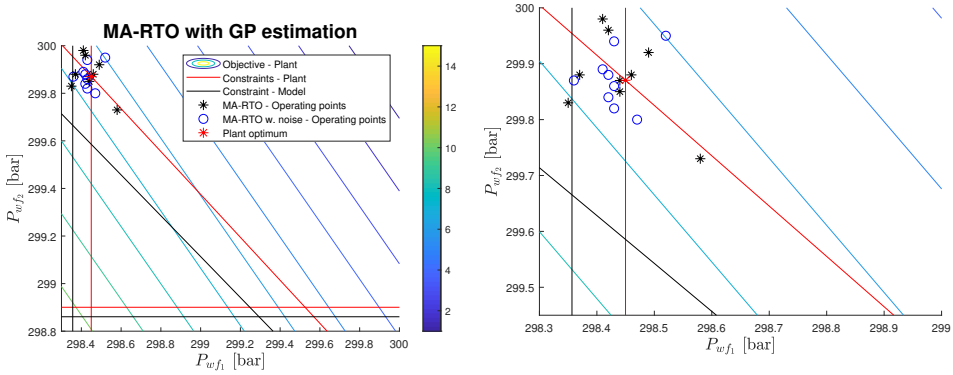
5.3.2 Simulations and results

To investigate how the MA-RTO with GP regression estimation handles plant-model mismatch, the optimization has been applied to four different cases, where the degree of plant-model mismatch varies from case to case. In addition to that,

all cases have been simulated two times each, one time without measurement noise and one with measurement noise. The measurement noise has been implemented as an identically distributed Gaussian distribution with zero mean and unit variance.

The simulations are done in the same cases as for the FDA estimation experiment in section 5.2.2. The modeled-GOR and plant-GOR, which indicates the mismatch, in the following cases are mentioned in the headlines on the form $GOR_{plant/model} = [GOR_{well_1} \ GOR_{well_2}]$. Every case has used the same GP regression model, meaning that the same training data has been used to construct the machine learning model for all the cases.

5.3.2.1 Case 1: $GOR_{plant} = [0.30 \ \ 0.40]$ - $GOR_{model} = [0.25 \ \ 0.35]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

(b) A closer view of the operating points for the simulations with and without measurement noise.

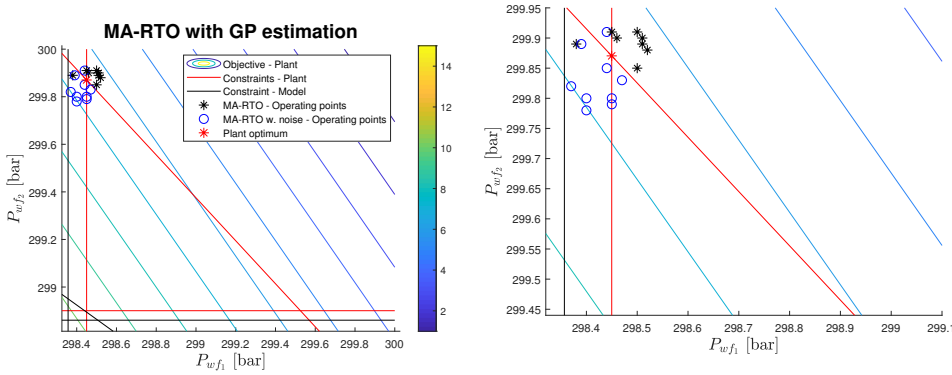
Figure 5.7: MA-RTO simulations using GP gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \ \ 0.40]$ and $GOR_{model} = [0.25 \ \ 0.35]$. MA-GP corrected iterations are plotted for both simulations, with and without measurement noise.

In this experiment, the model underestimates the amount of gas in the wells, which can be observed from the constraint in figure 5.7a. A big difference from the FDA cases is that several operating points are outside of the plant feasibility area. As a matter of fact, only two of the operating points for the case without noise and one for the simulation with noise are in the plant feasible area. However, the operating points converge to an area very close to the plant optimum, even in the case with measurement noise.

Overall the MA-RTO with GP estimation performs well to cope with the GOR mismatches, even with noise present. In fact, it operates very close to the optimum

and converges practically to the optimum. The algorithm handles measurement noise quite good too. On the other hand, it does not have feasible-side convergence, which will be discussed in section 5.4.

5.3.2.2 Case 2: $GOR_{plant} = [0.30 \;\; 0.40]$ - $GOR_{model} = [0.15 \;\; 0.30]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

(b) A closer view of the operating points for the simulations with and without measurement noise.

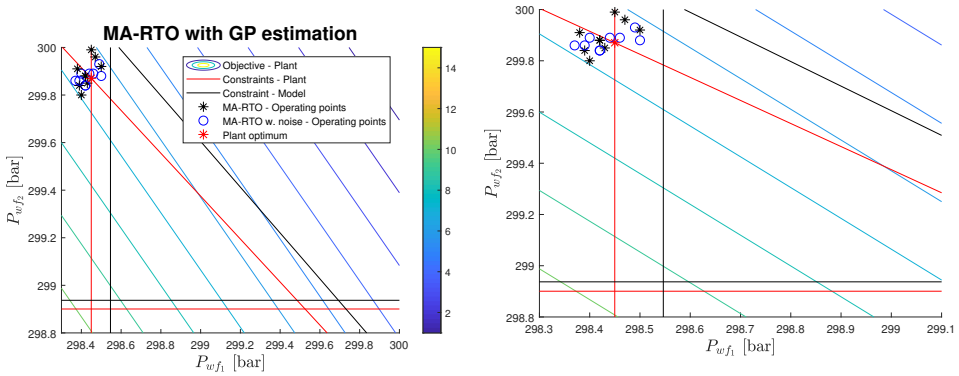
Figure 5.8: MA-RTO simulations using GP gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \;\; 0.40]$ and $GOR_{model} = [0.15 \;\; 0.30]$. MA-GP corrected iterations are plotted for both simulations, with and without measurement noise.

As in case 1, the model in case 2 also underestimates the amount of gas in the plant. Although the plant-model mismatch is greater in both wells, in this case, the algorithm performs at least as good as in the previous case. In fact, for the experiment without noise, only one of the operating points is outside the feasible region of the plant. Furthermore, it practically converges to plant optimum, meaning that the difference in objective between the MA-optimum and plant-optimum

is negligibly small.

All in all the MA-RTO with GP estimation performs well to cope with the GOR mismatches, with fast convergence to the optimum. However, it does not hold the property of feasible-side convergence.

5.3.2.3 Case 3: $GOR_{plant} = [0.30 \ \ 0.40]$ - $GOR_{model} = [0.36 \ \ 0.45]$



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

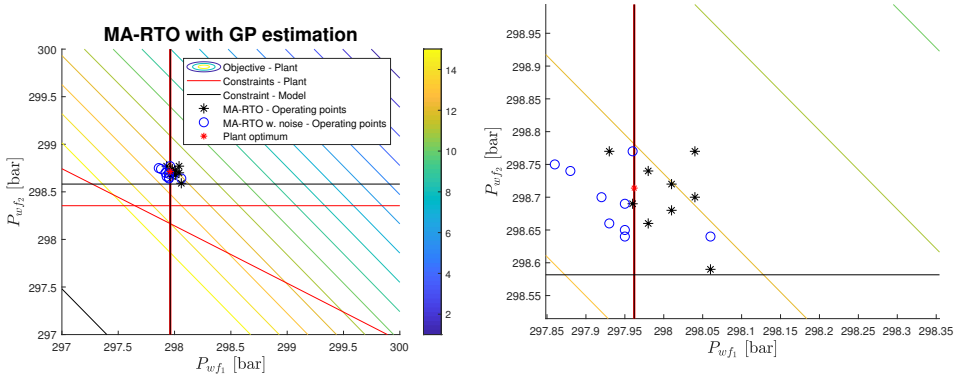
(b) A closer view of the operating points for the simulations with and without measurement noise.

Figure 5.9: MA-RTO simulations using GP gradient estimation to cope with plant-model mismatch. $GOR_{plant} = [0.30 \ \ 0.40]$ and $GOR_{model} = [0.36 \ \ 0.45]$. MA-GP corrected iterations are plotted for both simulations, with and without measurement noise.

Unlike the previous two cases, this model overestimates the amount of gas in the plant. As in the previous two cases, the algorithm is quite accurate with fast convergence towards the plant optimum. However, several operating points are on the infeasible side of the plant. The MA-RTO with GP estimation performs more

or less equally good for both experiments with and without noise measurement.

5.3.2.4 Case 4: GOR_{plant} = [0.10 0.20] - GOR_{model} = [0.10 0.10]



(a) Contour plot of the objective function of the plant with plant- and model-constraints.

(b) A closer view of the operating points for the simulations with and without measurement noise.

Figure 5.10: MA-RTO simulations using GP gradient estimation to cope with plant-model mismatch. GOR_{plant} = [0.10 0.20] and GOR_{model} = [0.10 0.10]. MA-GP corrected iterations are plotted for both simulations, with and without measurement noise.

In this case, the GOR in the plant is significantly lower than earlier. The plant-optimal valve opening is “fully open” for this case. Note that the gas-capacity limit will not be a problem when producing from this plant, which can be verified by the gas-capacity constraint, the sloped model-constraint, in figure 5.10a. To put it differently, the plant feasible area will not change even if we remove the gas-capacity constraint.

As can be seen, the MA-RTO with GP estimation converges fast and accurately,

but lacks feasible-side convergence.

5.4 Discussion

In this section, both MA-RTO methods, formulated in this chapter, have been applied to 4 different cases with plant-model mismatches. To investigate the properties of the algorithms, they have been tested for both environments, with and without measurement noise, for each case.

The MA-RTO method with gradient estimation using past RTO points performed very differently with and without measurement noise. As can be seen from the contour plots in figures 5.2 - 5.5 the algorithm performs very well to handle plant-model mismatches for the simulations without measurement noise. Overall the algorithm reaches plant-optimality within a few numbers of RTO iterations. Moreover, it is robust to starting points far from the plant-optimum and satisfies plant feasibility throughout the optimization process.

From the contour plots in figures 5.2 - 5.5, it can be observed that the MA-RTO with gradient estimation using past RTO points struggles to cope with plant-model mismatches when there is noise in the plant observations. However, the operating points are generally not too far from the plant-optimum. In the comparison to the simulations without noise, it performs poorly.

Another key point here is that the algorithm was very sensitive to the starting point. In fact, for simulations with starting points far from the plant-optimum, it was not even able to solve the optimization problem. This is a huge limitation because this means that we will need a lot of information about the plant to select a sufficient starting point. However, this can be solved by trying different initial

points and choosing the best one. This could be fine for a two-well system, but would be a real challenge for more complex systems.

The algorithm also struggled to satisfy plant feasibility throughout the optimization process, when there was measurement noise involved. From the contour plots, one can verify that several operating points are outside of the feasible region for the plant. A possible solution for this problem could be smaller steps with a stronger filter than the one I implemented.

On the other hand, the MA-RTO method with GP estimation was very robust against measurement noise. In fact, the algorithm performed equally good with and without measurement noise. Although GPs ability to handle noise was known, I did not expect it to perform that well. Even though they were generally equally close to the plant optimum, the simulation with measurement noise violated the constraints several times more. This can be verified by the plots in case 2 and case 4 in figure 5.8 and figure 5.10, respectively.

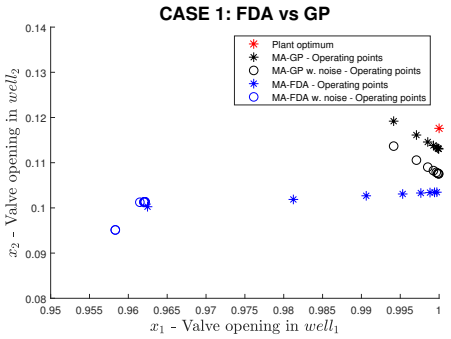
Another key point is GPs robustness against the initial point. It usually converged to the plant optimum regardless of the starting point of the algorithm. Different from the FDA-method, which only uses the past 3 RTO points, GP uses kernel methods that use all the available data to estimate the plant-model mismatch. For this reason, the GP regression can achieve better learning from data. However, in reality, it can be very costly or even infeasible to get enough data, from an oil production facility in order to construct a good GP model.

A big challenge with the MA-RTO with GP estimation was that many of the operating points were outside of the feasible region. This problem occurred for experiments both with and without measurement noise. This could have been

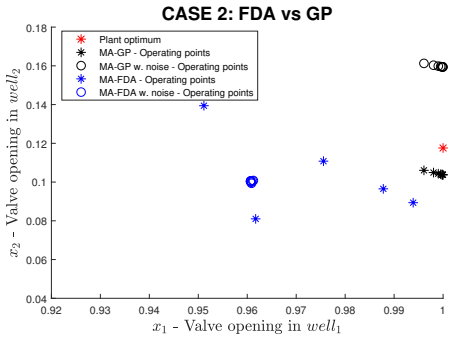
omitted with a stronger filter on the inputs. The implemented filter was 0.5, but increasing it could solve the constraint violation problem. This would lead to slower convergence, due to smaller steps, but could be required to push the operating points to the feasible region. Instead, a trust region could have been imposed on the inputs, which would require some knowledge about the plant. However, an adaptive trust-region, which develops through the RTO-iterations could be a simpler solution.

5.4.1 Comparison of Finite-difference approximation and Gaussian processes

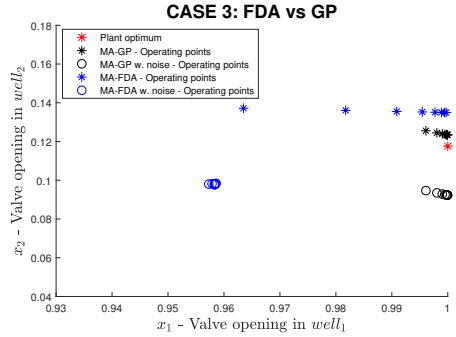
As discussed above both MA-RTO methods have their limitations and benefits. Figure 5.11 compare both approaches in each case. To get a clear understanding, the methods are compared based on the input values.



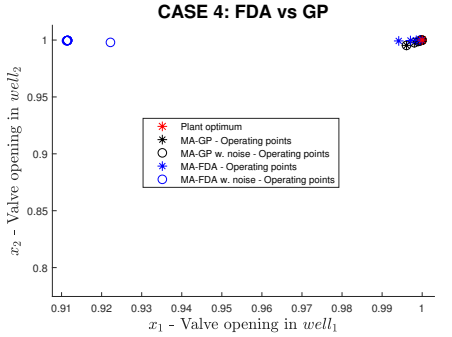
(a) $GOR_{plant} = [0.30 \ \ 0.40]$, $GOR_{model} = [0.25 \ \ 0.35]$



(b) $GOR_{plant} = [0.30 \ \ 0.40]$, $GOR_{model} = [0.15 \ \ 0.30]$



(c) $GOR_{plant} = [0.30 \ \ 0.40]$, $GOR_{model} = [0.36 \ \ 0.45]$



(d) $GOR_{plant} = [0.10 \ \ 0.20]$, $GOR_{model} = [0.10 \ \ 0.10]$

Figure 5.11: Comparison of MA-RTO with FDA and GP, for all cases

Obviously, the MA-RTO with gradient estimation using past RTO points performs very well when there is no noise involved but unfortunately it is insufficient when measurement noise is present. On the other hand, the MA-RTO with Gaussian process estimation converges towards plant optimum, regardless of noise present in the measurement. In fact, in case 1 both MA-GP simulations are closer to the

optimum than the MA-FDA method without noise.

Although the Euclidean distance between the MA-GP operating points is closer to the optimum, we have to remember that this method violated the plant-constraint numerous times before convergence. However, after some iterations, the algorithm operates in the feasible region. This can be explained by the fact that the algorithm has more data to train the model after some iterations. Consequently, it will be able to make a better machine learning model. Therefore, one has to consider the limitations to construct a sufficient machine learning model, in terms of having enough training data. In fact, few data points increase the chance of the Gaussian process model to make poor generalizations from the training data to unseen data, also known as an overfitted model [2]. This is particularly important when we are working with noisy data, regarding that our objective is to construct a machine learning algorithm that will separate the signal from the noise. If we overfit the model, it can end up learning the noise instead of the actual signal. One possibility is to start by applying the standard MA-FDA method until enough data is collected to make a good Gaussian process model.

Anyways, the MA-FDA method alone has shown its weaknesses when noise is present. In fact, this was no surprise, regarding that obtaining reliable estimates of plant gradients is the biggest drawback of the MA-method [1]. Hence, including noise in the plant measurements will obviously make these estimates more challenging to obtain. The use of quadratic approximation to estimate gradients instead of finite-difference approximation is also an interesting approach, which most likely would give more accurate estimations of the gradient. On the other hand higher order estimations as a quadratic approximation will require more computational power. Especially if we want to apply this method to a bigger system with more than two wells, we will have to take this into account.

In reality, an appropriate solution could be to use the MA-FDA method initially before applying the MA-GP when we have enough data to construct a good Gaussian process model. To increase the accuracy, the MA-FDA method can be replaced by the MA with quadratic approximation. This would indeed require more power, but will only be needed to learn more about the model before a simple and powerful GP model can be applied. However, the perfect solution would be an MA-GP model, as implemented in this project, but without constraint violations. This can be achieved by trust-region methods or a stronger filter, but will be an exercise for future work.

5.5 Conclusion

This thesis has proposed two Real-time optimization schemes, one that uses the standard Modifier Adaptation approach with gradient estimation using finite-difference approximation and a second one that combines Modifier Adaptation and machine learning through Gaussian Process regression. The schemes, which are applied to cope with plant-model mismatches have been illustrated, employing the RTO on an oil production system with two wells.

Simulations have shown that the proposed approaches perform well to find plant-optimum, despite plant-model mismatches. A comparison between standard MA and GP-based MA indicates that the latter outperforms the former in terms of noise mitigation. The MA-GP scheme converged relatively fast to the plant optimum, despite the fair amount of noise added to the measurements. However, the simulations show that it does not satisfy feasible-side convergence. On the other hand, the MA-FDA approach could only converge to the plant optimum in the absence of process noise.

After all, the standard Modifier Adaptation approach has shown its power to solve plant-model mismatches when it can obtain reliable measurements. The GP-based MA has surely illustrated its robustness to solve plant-model mismatches in the presence of measurement noise, and has great potentials if the feasible-side convergence problem can be solved. As a final conclusion of this work, it can be stated that including machine learning in the RTO leads to simplicity, robustness and better performance, in terms of noise attenuation, and has great potential to solve the plant-model mismatch problem in RTO.

5.5.1 Future work

From the simulations, we see that we achieve quite good results to cope with plan-model mismatches in RTO. However, some interesting challenges have to be investigated. In future work, it would be interesting to investigate the GP-based MA-scheme and attempt to solve the feasible-side convergence problem. If this problem could be solved, I am confident that the GP-based MA-approach will be valuable. As part of future work, it can also be interesting to apply an adaptive trust region method on the input. Moreover, a more conservative filter on the input can be applied. In addition to that, I will suggest applying both MA-schemes to a larger scale RTO problem.

Appendix A

A.1 Units

Table A.1: Units and numerical values used in the experiments

	Unit	**Value**
p	bar	
h	m	1000
C_d	$\sqrt{\frac{kg}{mbarday^2}}$	84600
M_g	$\frac{kg}{mol}$	16.04
R	$\frac{J}{kmol \cdot Kelvin}$	8314
T	Kelvin	373
k_{o1}	$\frac{Tonne}{bar^2}$	$6.576 \cdot 10^{-3}$
k_{g1}	$\frac{Tonne}{bar^4}$	$8.239 \cdot 10^{-7}$
k_{w1}	$\frac{Tonne}{bar^2}$	$3.344 \cdot 10^{-3}$
k_{o2}	$\frac{Tonne}{bar^2}$	$5.462 \cdot 10^{-3}$
k_{g2}	$\frac{Tonne}{bar^4}$	$5.373 \cdot 10^{-7}$
k_{w2}	$\frac{Tonne}{bar^2}$	$1.031 \cdot 10^{-2}$
ρ_o	$\frac{kg}{m^3}$	800
ρ_w	$\frac{kg}{m^3}$	1000
g	$\frac{m}{s^2}$	9.81
K^J		0.5
$K^{C_i} \quad i = [1, 2, 3, 4, 5]$		0.5

References

[1] A. Marchetti, G. François, T. Faulwasser, and D. Bonvin, "Modifier Adaptation for Real-Time Optimization—Methods and Applications," *Processes*, 2016.

[2] C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Regression," *Gaussian Processes for Machine Learning*, 2006.

[3] J. Nocedal and S. J. Wright, *Numerical optimization 2nd edition*. 2000.

[4] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Traces and Emergence of Nonlinear Programming*, 2014.

[5] L. Komzsik, "Mathematical optimization," in *Approximation Techniques for Engineers*, 2018.

[6] M. A. El-Reedy, *Offshore Structures*. 2012.

[7] B. Foss and T. A. N. Heirung, *Merging Optimization and Control*. 2016.

[8] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson, *RTO: An overview and assessment of current practice*, 2011.

[9] T. D. A. Ferreira, H. A. Shukla, T. Faulwasser, C. N. Jones, and D. Bonvin, "Real-Time optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes," in *2018 European Control Conference, ECC 2018*, 2018.

[10] E. A. Rio, C. J. E. Alves, E. Politecnica, and D. De, "Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization," *12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems*, 2018.

[11] J. E. Edwards and D. W. Otterson, "Tech talk: (4) pressure measurement basics," *Measurement and Control (United Kingdom)*, 2014.

[12] C. Grimholt and S. Skogestad, "Optimization of oil field production under gas coning conditions using the optimal closed-loop estimator," in *IFAC-PapersOnLine*, 2015.

[13] M. Fetkovich, "The Isochronal Testing of Oil Wells," *Fall Meeting of the Society of Petroleum Engineers of AIME*, no. American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc. 1973.

[14] F. Jahn, M. Cook, and M. Graham, *Hydrocarbon Exploration & Production*. 2008.

NTNU
Norwegian University of
Science and Technology