NTNU

Norwegian University of
Science and Technology

Real Time Optimization and Nonlinear Model Predictive Control of a Gas Lifted Oil Network

Sarry Haj Yahia

Chemical Engineering and Biotechnology
Submission date: March 2018
Supervisor: Johannes Jäschke, IKP
Co-supervisor: Tamal Das, IKP
Eka Suwartadi, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Summary

As oil reserves become scarcer, oil production and recovery must be enhanced. The gas lift is one of the techniques used to enhance oil production and recovery. It is usually used in reservoirs that suffer from insufficient production rates because of inadequate reservoir pressures. The gas lift injects gas, from external sources, into the fluid mixture flowing out of the reservoir. This in turn will reduce the density of the fluid which enables the reservoir pressure to lift the mixture to the top. Thus, the gas injection increases the production rates of the oil. However the maximum gas injection does not necessarily lead to maximum production of fluid or oil. Thus, gas injection can be both advantageous and disadvantageous.

This thesis investigates a gas lifted network consisting of 3 wells and a riser. Nonlinear dynamic equations were derived for the mass of liquid and gas using the outflow rates from different component of the network. The mathematical model used to describe the system takes into consideration a three phase system consisting of oil, water and gas. In addition, the model uses a simplified equation to represent the friction in the tubing. For simulation purposes, the mathematical model was formulated as a system of *differential algebraic equations* (DAEs) and simulated in Matlab

In this thesis, a *real time optimization* (RTO) problem is built to find the optimal steady state solutions for a given objective. The objective considered is the maximization of liquid and minimization of gas injection. In addition, a *nonlinear model predictive control* (NMPC) is applied to the plant to operate and track the solutions produced by the RTO. Both the RTO and NMPC optimization problems built are solved using CasADi software.

The simulation studies have been carried out to test the open loop response of the plant and the RTO steady states optima. Further, the NMPC sensitivity to measurement noise is investigated. Finally, NMPC robustness with respect to parameter changes in the closed loop structure is studied and analyzed. The analysis and simulations show that the simplified gas lift model represents the system dynamics as expected. In addition, the NMPC built in this system is sensitive for highly noisy measurements while it is accurate and stable for slightly noisy measurements. Finally, the control structure of the entire gas lifted network is able to adapt to new optimal operations when subject to disturbances. This is beneficial for future work when considering optimizing the quality of production.

Preface

Concluding this long and challenging journey of study which started in 2012 would not be sufficient without sharing this research accomplishment with numerous people who nurtured, encouraged, facilitated, orchestrated and enabled me in materializing a well-done research document. I owe them a great debt, I will nonetheless attempt to thank here a few of them.

Sincere thanks to my Supervisor, Associate Professor Johannes Jaschke, whom I met in Boston MIT, then he accepted me to be his student. Thereby, I do appreciate his consistent direction, emotional support, goodwill and useful comments. His adherence to high standard enabled me to create and craft a qualitative final research document, that I am proud of being the author of it.

A considerable amount of gratitude to my Co-Supervisor, Eka Suwartadi-Post Doctoral Fellow, for his technical support and programming here in Norway and while he is away in the USA. Alongside, his wisdom and insight have a tremendous impact on me by installing enthusiasm, perseverance, and instigation which motivated me towards more and further deep understanding of my research topic.

While writing this final research document based on many days and nights. I am profoundly indebted to numerous people who surrounded me with support and encouragement, my Co-Supervisor Tamal Das- PHD Candidate, who assisted me greatly with the modeling and understanding the concepts of my work alongside his advice. I wish to acknowledge his assistance, facilitation, which helped me fashion an exciting, and constructive research.

I would like also to extend deep appreciation to professor Sigurd Skogstad-Professor for always being available, helpful and supportive. Special thanks especially to Hege Johannessen– Student Advisor for her loving solicitude and substantive and essential assistance throughout 5 years long of study.

Finally, to my parents and friends in Norway and the world, I wish to thank them from the bottom of my heart for their genuine concern and caring. They have inspired me and assisted me in internalizing the core values of hard work, perseverance, persistence and happiness.

Declaration of Compliance

I hereby declare that this thesis is an independent work according to the exam regulations of the Norwegian University of Science and Technology (NTNU)

Trondheim, Norway

March 12, 2018 Sarry Haj Yahia

Table of Contents

List of Figures

List of Tables

Abbreviations

CV	Controlled Variable
MV	Manipulated Variable
PID	Proportional Integral Derivative
CasADI	Computer algebra system for Automatic Differentiation
DAE	Differential Algebraic Equations
ODE	Ordinary Differential Equations
DMS	Direct Multiple Shooting
DSS	Direct Single Shooting
LP	Linear Programming
QP	Quadratic Programming
NLP	Nonlinear Programming
SQP	Sequential Quadratic Programming
IP	Interior Point
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
OP	Optimization Problem
OCP	Optimal Control Problem
RTO	Real Time Optimization
IPOPT	Interior Point OPTimizer
RHC	Receding Horizon Control
EKF	Extended Kalman Filter

List of Symbols

Modeling Symbols

Symbol	Description	Unit
R	Universal gas constant	$J/K/mol$
μ	Viscosity	$Pa \cdot s$
ρ_L	Liquid density	kg/m^3
ρ_{oil}	Oil density	kg/m^3
ρ_{water}	Water density	kg/m^3
ρ_{mix}	Three phase mixture density	kg/m^3
ρ_G	Gas density	kg/m^3
$\bar{\rho}$	Average density	kg/m^3
g	Gravity	m/s^2
GOR	Mass gas oil ratio	$-$
PI	Productivity index	$kg/s/Pa$
WC	Water cut	$-$
M_G	Gas molecular weight	kg/mol
T	Temperature	K
L_a	Annulus length	m
L_{bh}	Bottom hole length	m
L_t	Tubing length	m
L_r	Riser length	m
V_t	Tubing volume	m^3
V_r	Riser volume	m^3
V_{bh}	Bottom hole volume	m^3
S_{bh}	Bottom hole cross section	m^2
P_{bh}	Bottom hole pressure	bar
P_{fr}	Pressure drop due to friction	bar
P_{res}	Reservoir pressure	bar
K_{gs}	Gas lift choke coefficient	-
K_{pr}	Production choke coefficient	-
ϵ	Wall roughness	m

Chapter 1

Introduction

The following work is inspired by the importance of optimizing reservoir production and enhancing oil recovery to utilize energy resources to the fullest, which can reduce environmental effects.

1.1 Motivation

As human population is growing rapidly, so is energy consumption and demand. Energy consumption exerts demands on energy resources making them scarcer. As energy resources become harder to extract, oil recovery becomes exceedingly important and complex in various environments. Hence, it is of great importance to optimize reservoir production and enhance oil recovery, to meet the rush for energy, while increasing profitability, efficiency and productivity of well oil production.

Industry experts are convinced that much of the demands can be met with artificial lift technologies, that can increase long term potential production capabilities. About 90% of current oil wells worldwide operate with artificial lifts for enhancing oil production [12]. More than 30,000 wells use the gas lift technique, primarily because of its economic viability [12]. Generally, the reservoir pressure in these wells is considered inadequate to assure oil flow to the surface. Therefore, artificial lift techniques are essential to extend life time of wells and boost oil production rates, profitability and reservoir utilization. There are several major forms of artificial lifts which are used

widely in the oil industry, these include: Sucker-rod (beam) pumping, electrical submersible pumping (ESP), reciprocating and jet hydraulic pumping systems and finally gas lifts, [3]. The emphasis of this master thesis is the artificial gas lift.

Despite the many advantages of the gas lift such as increasing production, and increasing utilization of resources in reservoirs, the gas lift is a highly coupled and very nonlinear system. Under different conditions of gas injection and pressures the system becomes highly unstable and can lead to reduced oil production and damage. In the oil and gas industry, a major objective is to maximize production and minimize costs, at operating points which optimizes production. The focus of this thesis is to apply a simplified gas lift model and study different optimal points of operations under different circumstances, using optimization and control techniques.

In this thesis a three well system and a riser sharing the same manifold is considered. The production of oil is based on maximizing the total oil outflow which is the sum of oil produced from each of the wells, as well as, minimizing the injection of gas into the wells' tubing, while operating under stable conditions. A proper automated control scheme allows to optimize and maximize the total production of oil, and minimize the total gas injection rate while operating under stable conditions.

For processes with multiple variables and constraints, such as the gas lifted oil network, predictive control has been found to be a very good controller design scheme. Many of artificial gas lifts wells with high production rates operate with manual driven gas injection and production [12]. The introduction of such control schemes is slow mainly due to the prohibitive cost of well intervention to install new sensors and actuators [17], that have to cope with very harsh conditions, such as, high pressure, temperatures and vibrations [17]. There are several important articles related to the optimization, control and modeling of gas lifts such as [7] on stabilization, and [8] stabilization based on state estimation, also [17] and [23] for the modelling and control of the gas lifts using NMPC.

1.2 Scope and emphasis

This master thesis is not a direct continuation of the project assignment. The scope of this thesis is to modify the gas lift model by introducing a simplified pressure loss due to friction equation and to introduce water cuts

to the system. The reason for introducing a simplified friction term is to create a more stable system to study the behavior when subject to disturbances. Further the water cut is introduced to study its effects on the total production. With the simplified model, the scope is further to develop a network consisting of 3 wells and a riser used for the application of a control structure consisting of a real time optimization (RTO) and a tracking (NMPC). Both of which are developed during the work of this thesis. The model inside the NMPC was integrated using the direct orthogonal collocation. The objective is to study the functionality and ability of the RTO to produce optimal steady states under different circumstances as well as to study the ability of the NMPC to produce suitable control trajectories to track these points. Hence the emphasis is to study the gas lifted oil network, and the development and application of RTO and tracking NMPC on such system using CasADi software.

1.3 Outline of thesis

This thesis is outlined in the following way: Chapter 2 presents the gas lift techniques and explains the main concept while also introducing the necessity of optimization. In addition, the chapter motivates the concept of optimization, as well as it introduces how optimal control problems are formulated and solved. Finally, the hierarchy of control structure is presented to show the interconnection between different control layers in the system. Chapter 3 presents the gas lifted oil network that was studied, the three wells and a riser system. Moreover, it shows how the water cuts were introduced to the system and how the pressures in different parts of the plant were represented mathematically. Towards the end, the chapter shows how the gas lifted oil network was represented as a system of differential algebraic equations (DAEs). Further, Chapter 4 introduces the steady state optimization and the dynamic optimization problems formulations, and their solution methods. Hereafter, Chapter 4 presents the entire control structure built on top of the plant using steady state and nonlinear dynamic optimization of the gas lifted oil network, as well as the software used in the simulations. In Chapter 5 simulation results and analysis are presented for different case studies and tests on the promising control structure developed in the thesis. The conclusions are shown in Chapter 6 where the recommended control structure and results are concluded, and future work is noted. The appendices contain the constants used in the models and the programming codes coded during the work of this thesis.

Chapter 2

Background

This chapter begins by presenting the gas lift technique in offshore oil production. In addition, it justifies the need and benefits of optimization and control in gas lifted systems. The reader of this chapter can also find brief background and introduction to the formulation of optimization and control and the solution methods that are widely used to solve these optimal control problems (OCPs). Finally, the structural decisions involved in the control system design of a chemical plant are presented in the end.

2.1 The gas lift technique

In offshore oil production the pressure of the reservoir plays an important role. The higher the pressure, the higher the oil production is, hence reservoir pressure is the main driving force in the offshore oil production. In order to raise the oil production through the pipelines, the driving force must be high, thus high reservoir pressure is desired. As the oil well matures during its lifetime, the reservoir pressure decreases due to large oil extraction. Subsequently, when the reservoir pressure declines to a certain point after several years of production, certain changes pertaining to fluids occur in porous media, affecting reservoir performance significantly. The latter could have significant economic effects, and thus gas-lift techniques can be applied to solve this oil production issue [10]. The gas-lift is used merely to inject gas near the bottom of the pipeline to reduce the bottom fluid density, which in return reduces the weight of the fluids in order to

successfully push the oil up.

To understand the process, one must take into consideration the changes in the reservoir that occur during the pressure decline. These changes include: (1) changes in reservoir fluid volume and density. (2) changes in fluid compressibility, viscosity and mobility. (3) changes in fluid composition due to evaporation of lighter hydrocarbons from the liquid or condensation of liquid from the gas. (4) changes in the gas to oil ratio (GOR) and water cut (WC). For further reading on reservoir theory see [20]. Later in this thesis, a comprehensive analysis on the effects of changes to water cut and gas to oil ratios is conducted. The latter affects the well production rates and ultimate recovery from a reservoir and design of surface facilities. The reservoir pressure is a result of natural forces that trap the fluids within the pores of the reservoir, porosity and permeability. When a hole is drilled into a reservoir the fluids are allowed to escape, due to the pressure difference between the bottom of the reservoir and the top end near the hole. The well can be considered as that hole. When introducing the well to the reservoir, the fluids are provided with a new flow path and start flowing through the bottom of the pipeline to the top facilities, usually a gravity separator. In order for the mixture fluid to choose the path through the well, the reservoir pressure must be larger than the column pressure, which is larger than facility surface pressure. Otherwise, the oil will not be able to flow up through the pipelines, and the system will fail to produce oil. Therefore, when the bottom hole pressure is nearing the reservoir, employing an artificial gas-lift will be beneficial [7].

A simple gas-lift model is depicted in Figure 2.1. The gas-lift introduces extra gas source through the annulus around the tubing. The gas is usually injected near the bottom of the tubing through an injection valve. When gas is injected into the tubing and enters the fluid, it will consequently reduce the average density of the mixture at that point. As the density of the fluid mixture is decreased it will subsequently decrease the weight of the fluid, which in turn cause a decline in the pressure drop in the tubing. As the pressure declines to a certain point, and the pressure difference between the reservoir and gas-lift is sufficient, oil will start flowing through the well to the top. Gas-lift is an excellent technique to enhance production rates in reservoirs that suffer from low production rate due to insufficient natural reservoir pressure.

Gas-lifts are not always optimal, in some cases gas-lift techniques may harm oil production rates. In some cases constraints imposed on the system may cause instabilities. Instabilities in flow regimes cause severe oscillations that

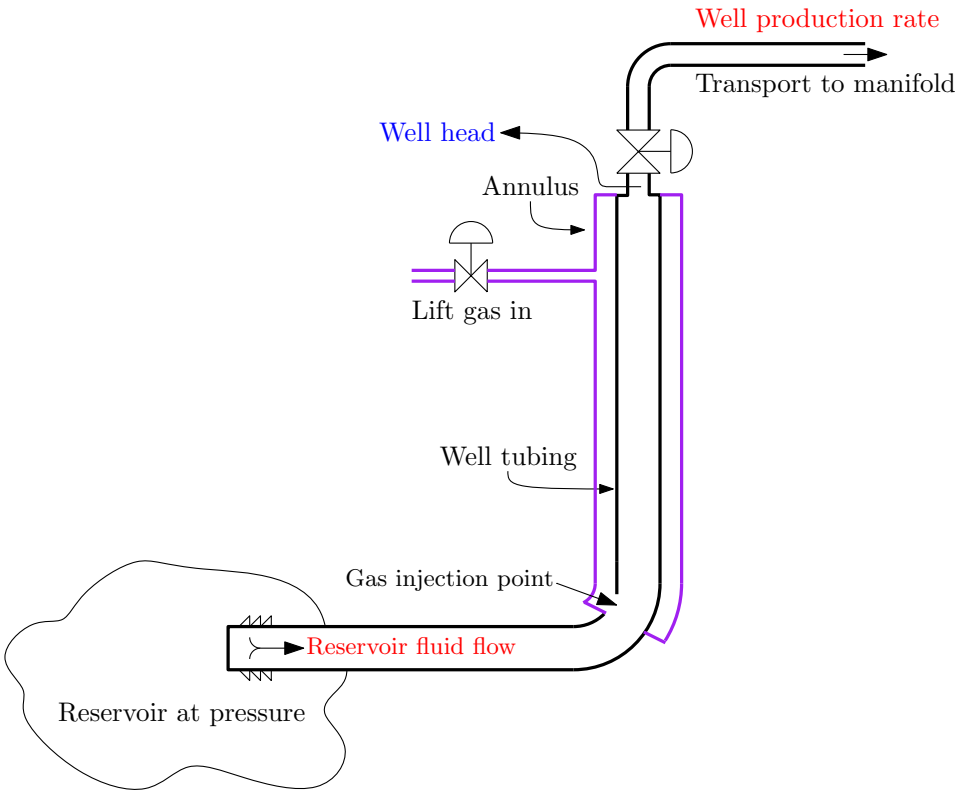


Figure 2.1: Representation of the gas lift design.

reduces average production and damage equipment. The most common case is the casing heading instability, however, it is not in the scope of this master thesis. Finally, when considering a system of wells sharing the same manifold, the properties of the reservoirs, such as GOR and WC, must be taken into consideration so that oil production from each well operates efficiently and profitably.

2.1.1 Gas lift optimization

The need for optimization and control in the gas lift can be justified by taking into consideration the relationship between production rates and gas injection rates. To investigate the possible benefits of control, a comprehensive simulation study was performed using the gas lift setup that was developed in this thesis. The gas lift model will be further discussed in details in chapter 3, for which the constants used in the setup are presented

in Appendix A.

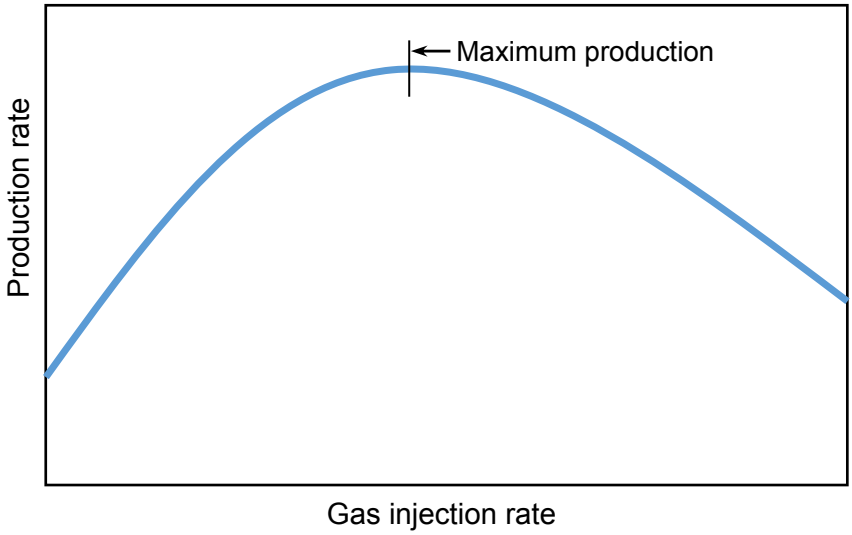


Figure 2.2: Production rate subject to gas injection rate. Maximum oil production is also displayed.

Figure 2.2 shows the relationship curve between the production rate of oil and the gas injection rate of a hypothetical well. Figure 2.2 indicates that the production rate of oil increases rapidly to a certain point before it starts dropping again. This implies that the injection rate of gas can be not only advantageous but also disadvantageous. This relation or curve is justified by hydrostatic pressure drop that cannot compensate the increased pressure drop due to friction which is originating from increased gas flow inside the tubing. This master thesis focuses on factors such as reservoir parameters, gas injection rates, noises in measurements and multi-well system connected with the same manifold, as the factors that might have an effect on optimization.

The oil production is determined by the gas lift optimization procedure. The common understanding about gas lift optimization is to obtain the maximum production by manipulating the operating conditions. However, the maximum production of oil does not necessary mean the maximum profit if oil production is dependent on injection [11]. Therefore, in this thesis different cases and scenarios of optimization are studied to show how maximum oil production through liquid production optimization can be obtained when subject to uncontrollable changes in parameters.

2.2 Introduction to numerical optimization

To begin a study of a system, an objective must be identified. This objective is usually determined by the maximum profit a company can reach meanwhile costs are at their minimum. The costs are primarily from energy consumption and raw materials necessary to operate the chemical plant. Certain unknown variables in the chemical plant affect the objectives, and thus the aim of optimization is to find a good combination of these unknown variables to reach the maximum or minimum goals. Furthermore, the optimization must occur within the bounds and constraints imposed on those variables.

Prior to formulating an optimization problem, a model has to be developed to decide suitable values and objectives for the different components of an optimization problem [6]. There are three main components, an *objective function, decision variables and constraints.* The objective function is a scalar function which describes a property. Optimization is the maximization or minimization of an objective function subject to constraints and bounds. The decision variables may be real variables, integers or binary variables. Constraints are normally divided into two types, *equality* constraints and *inequality* constraints. Hence, the optimization problem can be formulated as follows:

$$\min_x \ f(x) \tag{2.1a}$$

subject to

$$g_i(x) = 0, \qquad i = 1,..,m \tag{2.1b}$$

$$h_j(x) \leq 0, \qquad j = 1,...,n \tag{2.1c}$$

$$x_{min} \leq x \leq x_{max} \tag{2.1d}$$

Here $f(x)$ is a scalar objective function that is subject to a set of equality constraints $g_i(x)$ and inequality constraints $h_j(x)$ with i and j are disjunct index sets. All these functions are dependent on x which is a set of decision variables constrained by lower and upper bounds, $x_{min} \leq x \leq x_{max}$. Equation 2.1 represents a general mathematical formulation of optimization problems. It is noteworthy to mention that since $f(x)$ in Equation 2.1 is scalar, the objective function can be easily transformed from a minimization problem to a maximization problem by introducing a negative sign, as in $(-f(x))$.

Optimization problems vary based on their linearity, and convexity of the objective function and constraints. This thesis, briefly discusses some classes of optimization problems. A detailed explanation of classes of optimization problems, are found in [15] and [9].

2.2.1 Classes of optimization problems

Constrained and unconstrained optimization: Equation 2.1 is a constrained optimization problem. However, if the objective function $f(x)$ can operate freely without any restrictions under any conditions, then it's said to be relaxed from constraints and hence Equation 2.1 becomes merely:

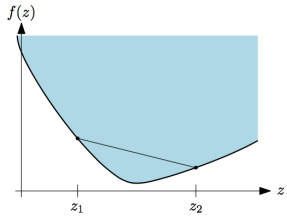
$$\min_{x} f(x) \tag{2.2}$$

Local and global optimization: Global optimization is an optimization problem that attempts to find the absolute smallest or largest point of the objective function. However, problems are mostly concerned with local minima or maxima, which is the point at which the objective function is smaller or larger than its neighboring points.

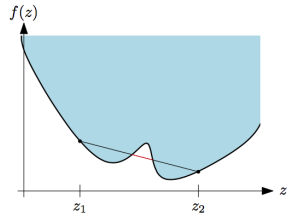
Convex and non-convex optimization: An optimization problem is convex if the objective function $f(x)$ is a convex function, and the feasible set of x is convex. Furthermore, a problem is strictly convex if there is always only one global or local solution. In Figure 2.3 convex and noncovex sets are displayed. As indicated the straight line between any point in the set must always lie within the borders of that set, if a line crosses the set then its defined as nonconvex. the latter also applies for the definition of function convexity, which is displayed in Figure 2.4.

(a) A convex set. (b) A nonconvex set.

Figure 2.3: On the left, a convex set is depicted. On the right, a nonconvex set is depicted [9].



(a) A convex function.



(b) A nonconvex function.

Figure 2.4: On the left, a convex function is depicted. On the right, a nonconvex function is depicted [9].

There are several types of of optimization problems defined by Equation 2.1. The following are the most common:

Linear Programming (LP): If the objective function $f(x)$ is linear and all constraints $g(x)$ and $h(x)$ are linear, then Equation 2.1 is a *linear program (LP)*, where d is a vector of known coefficients, which can be written as

$$\min_x \; d^T x \tag{2.3a}$$

subject to

$$g_i(x) = a_i^T x - b_i = 0, \qquad i = 1,..,m \tag{2.3b}$$

$$h_j(x) = a_j^T x - b_j \leq 0, \qquad j = 1,...,n \tag{2.3c}$$

$$x_{min} \leq x \leq x_{max} \tag{2.3d}$$

Quadratic Programming (QP): When the objective function $f(x)$ is a quadratic function and all the constraints are linear, it can be formulated as in Equation 2.4

$$\min_x \; x^T Q x + d^T x \tag{2.4a}$$

subject to

$$g_i(x) = a_i^T x - b_i = 0, \qquad i = 1,..,m \tag{2.4b}$$

$$h_j(x) = a_j^T x - b_j \leq 0, \qquad j = 1,...,n \tag{2.4c}$$

$$x_{min} \leq x \leq x_{max} \tag{2.4d}$$

QP problems can be both convex or non convex depending on the quadratic form of the objective function and the constraints. In Equation 2.4 Q is an

n-dimensional symmetric matrix and if Q is positive semidefinite, $Q \succeq 0$, then Equation 2.4 is a convex problem. $Q = 0$ is by definition positive semidefinite. Therefore Equation 2.4 turns into an LP problem which is convex if the constraints are linear. In addition, QP problems can be non convex if the Q matrix is negative definite.

Nonlinear Programming (NLP): An optimization problem defined by the system's constraints, over a set of unknown real variables, along with an objective function to be maximized or minimized, in which some of the constraints or the objective function are nonlinear. An NLP can be defined similarly by Equation 2.1.

2.2.2 Solution methods

There are various solution methods for optimization problems. There are explicit methods which can be used for simple problems, or iterative schemes which are the common approach to optimization problems (OPs).

Algorithm 1 Iterative solution procedure of OPs

```
Given initial point x_0 and stopping criteria
while stopping criteria not fulfilled do
  Compute the next iteration point
end while
```

In the iterative algorithm procedure the optimization requires an initial value x_0, which is the measured value from the plant. To solve optimization problems the most common approach is the sequential quadratic programming (SQP) method, which does not require an initial feasible point, and interior point method (IP). Furthermore, the stopping criteria include one or several of the following: (i) a maximum allowed number of iterations, (ii) progress matrices, the gradient of the objective function (∇f) or the Lagrange $\nabla \mathcal{L}$ of the objective funtion, and (iii) a characterization of the optimal point,$\|\nabla f\| < \epsilon$ or $\|\nabla \mathcal{L}\| < \epsilon$ where $\epsilon > 0$ is a chosen small value.

Interior point methods are good methods for solving nonlinear optimization problems with inequalities such as Equation 2.1. The IP penalizes inequalities by a barrier function and then solves it as in equality constrained case, see [9] and [15].

2.3 The optimal control problem

Optimal control regards the optimization of dynamic systems. Dynamic systems are processes that are evolving in time and that are charaterized by states x that allow us to predict the future behaviour of a process. These dynamic systems can be controlled by inputs u. Typically, in order to optimize some objective function these controls must be chosen optimally while respecting the constraints. The process of finding suitable inputs u for optimal control requires the discretization of the model in the predictive state.

This thesis is confronted with a problem whose dynamic system lives in continuous time and whose control inputs are a continuous profile [4]. A continuous-time optimal control problem can be described as follows.

$$\min_{x,u} E(x(T)) + \int_0^T L(x(t),u(t))dt \tag{2.5a}$$

subject to

$$x(0) - x_0 = 0, \tag{2.5b}$$
$$\dot{x}(t) - f(x(t),u(t)) = 0, \qquad t \in [0,T], \tag{2.5c}$$
$$g(x(t),u(t)) = 0, \qquad t \in [0,T], \tag{2.5d}$$
$$h(x(t),u(t)) \leq 0, \qquad t \in [0,T], \tag{2.5e}$$
$$r(x(T)) \leq 0, \tag{2.5f}$$

Problem 2.5 and its variables are visualized in Figure 2.5. In Equation 2.5 the $L(x,u)$ term is called the *Lagrange term* (not to be confused with Lagrange function) and $E(x(T))$ is called a *Mayer term*. Further, t represents the time variable $\in [0,T]$, $x(t) \in \mathbb{R}^{n_x}$ denotes a vector of state decision variables and $u(t) \in \mathbb{R}^{n_u}$ denotes a vector of control decision variables. Sub-equation 2.5c represents an equality constraint of the ODE model, Sub-equation 2.5d is an additional equality constraint which is introduced for DAE systems [4]. Furthermore, Sub-equation 2.5e represents other constraints that are usually present which are the path constraints inequalities, for example upper and lower bounds of inputs $u_{min} \leq u \leq u_{max}$. Finally, Sub-equation 2.5f describes the inequality terminal constraint, which is used if the optimization has a fixed terminal or final state that is desired.

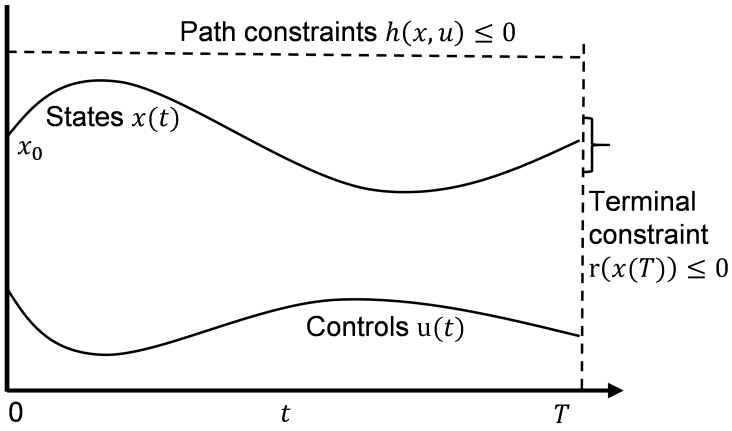


Figure 2.5: Continuous time optimization problem. Displayed Initial value of x_0, along with the states x and control u depending on time. Further, Path constraints h and terminal constraints r, where T is the prediction horizon time.

2.4 Solving optimal control problems

There are two approaches to address continuous time optimal control problems, (i) indirect approach and (ii) direct approach see Figure 2.6. This section briefly introduces the indirect and direct approaches to solve optimal control problems [4].

Indirect Methods use the optimality conditions to derive a boundary value problem (BVP) in ordinary differential equations (ODE). The BVP in the indirect method is solved by *first optimize, then discretize.* Hence the optimality conditions are described in the continuous form, and then discretized to compute a numerical solution. The major drawback in this indirect methods is the difficulty of solving differential equations due to non-linearity, [4].

Direct Methods on the contrary to indirect methods, can be used to solve optimal control problems without having to derive a BVP. Rather, direct methods transform the optimal control problem into a finite NLP which is then solved by numerical optimization that exploit the structure of the problem. Hence, direct methods *first discretize, then optimize.* To elaborate, the continuous time dynamic system is transformed into a discrete time system and then the process of optimization is deployed. Direct methods can easily treat all sort of constraints, as they are treated by well developed

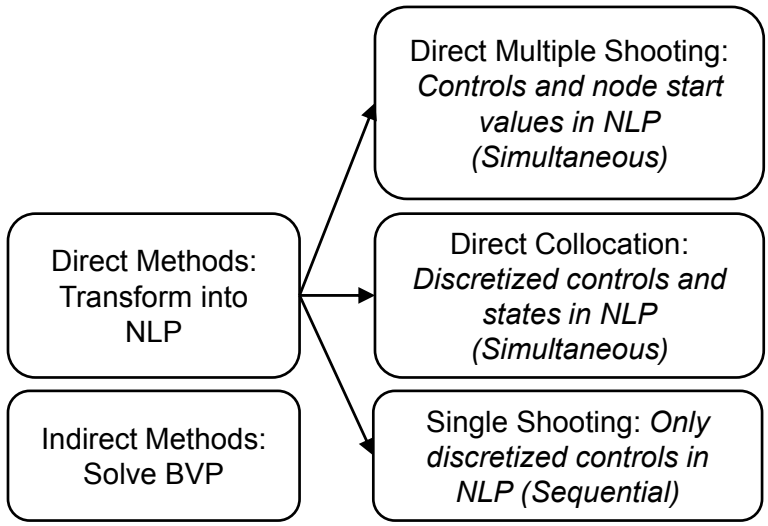


Figure 2.6: Classification of methods of optimal control problems.

NLP methods that deal with active changes [4]. There are two approaches within the group of direct methods, namely, sequential (single shooting) and simultaneous (multiple shooting and collocation) as can be seen in Figure 2.6. It is worth to note that simultaneous approaches can handle both systems of differential algebraic equations (DAE) and ordinary differential equations (ODE) [6].

Direct single shooting

In this method the model simulation and optimization are done sequentially. Here the control trajectory is approximated and parametrized using piecewise smooth approximation, while the ODEs and DAEs are solved using another integration solver.

The time horizon of the problem is divided into a set of grid-points as $t_0 = 0 < t_1 < ... < t_N = t_f$. The control parametrization is denoted by $u(t;q)$, where q is the finite control parameter which is set to $q_k = u(t)$ for $t \in [t_k,t_{k+1}]$ and $0 \leq i \leq m$, to show the dependence on each interval [4]. After the integration the states $x(t;q)$ are obtained as a function of the previous control parameters and finally the problem is moved to an NLP solver to find the optimal control trajectory/sequence. Hence, sequential approach. See Figure 2.7 for a schematic illustration of direct single shooting.

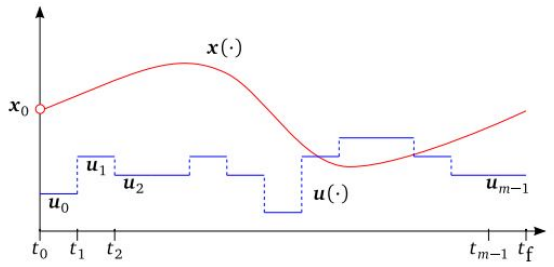


Figure 2.7: Direct Single Shooting (DSS) discretization applied to the optimal problem [14].

Direct multiple shooting

The idea behind direct multiple shooting approach is to shorten the long integration dynamics which can be counterproductive for discretizing continuous optimal control problems into NLPs. This is done by limiting the integration over short time intervals. This method performs a piecewise control discretization of the continuous control input $u(t) = q_i$ for $t \in [t_i,t_{i+1}]$, then it solves the ODE on each interval $[t_i,t_{i+1}]$ starting with artificial initial values s_i. See Figure 2.8 for an illustration of the artificial values.

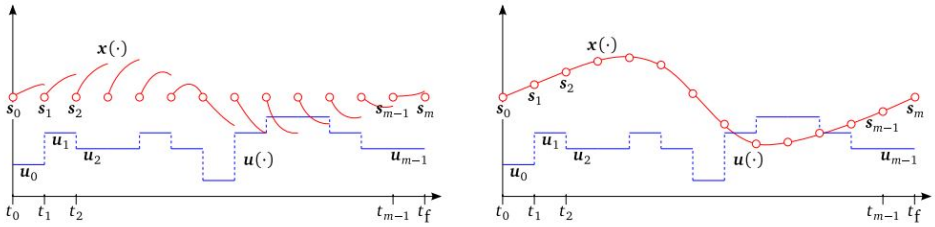


Figure 2.8: Direct Multiple Shooting (DMS) discretization applied to the optimal problem. On the left, initialized shooting nodes where solution is violating constraints. On the right, the constraints are fulfilled after NLP convergence [14].

$$\dot{x}_i(t,s_i,q_i) = f(x_i(t,s_i,q_i),q_i), \quad t \in [t_i,t_{i+1}] \tag{2.6a}$$

$$x_i(t,s_i,q_i) = s_i. \tag{2.6b}$$

In multiple shooting method the integrals in the OCP are numerically computed, and additional constraints are added to ensure the continuity condi-

tion in Equation 2.6b. Thus, OCP based on multiple shooting produces the NLP as in Equation 2.7.

$$\min_{s,q} E(s(m)) + \sum_{i=0}^{m-1} l_i(s_i,q_i) \tag{2.7a}$$

subject to

$$x_0 - s_0 = 0, \tag{2.7b}$$
$$x_i(t_{i+1},s_i,q_i) - s_{i+1} = 0, \qquad i \in [0,m-1], \tag{2.7c}$$
$$h(s_i,q_i) \leq 0, \qquad i \in [0,m], \tag{2.7d}$$
$$r(s_N) \leq 0, \tag{2.7e}$$

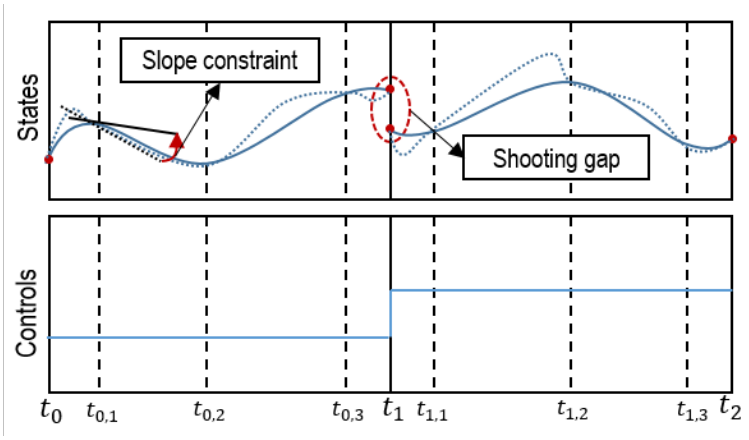


Figure 2.9: Polynomial approximation (blue curve), actual system dynamic (blue dotted curve), state derivative of polynomial (black line), derivative of the system dynamics (black dotted line) and the piecewise constant control input over the interval (blue lines). In addition, the slope constraint (red arrow) which is done t each collocation point, and the shooting gap constraint (red dashed circle) are displayed.

Direct collocation

The most relevant class of direct methods in this thesis is the direct transcription method, *direct collocation*. The main distinction in this method is that discretization occurs in both controls and states on a fixed fine grid N intervals with c intermediate collocation points in each interval. The states

in the collocation method are approximated by cubic polynomials and the controls are described as piecewise linear function. There are several common choices of the polynomials, such as, (*i*) *Gauss-Legendre scheme*, (*ii*) *Radau points* or (*iii*) *Lobatto points*. Collocation points differ in their position in the interval, for example Radau has a collocation point at the end of the interval. The collocation points are chosen based on different factors for stability, sensitivity and convergence [4].

An illustration of the direct collocation method is displayed in Figure 2.9. Figure 2.9 shows that in addition to the shooting gap constraints, a derivative equality constraint is added to the NLP. This is depicted by the red arrow which has the following formulation $\dot{x}_c - f(x_c,u(t)) = 0$.

2.5 Plantwide control

Plantwide control is concerned with the structural decisions involved in the control system design of a chemical plant. In particular, plantwide control is concerned with the questions "which variables should be controlled, measured, manipulated and what links should be made between them?" [21]. For large chemical plants such as the oil production from a multi-well system, it is important to have a suitable and stable control structure. The two main objectives of a control structure is the long term economics and the short term stability. Figure 2.10 shows the different layers constituting the selection of control structures in chemical plants. Each layer in the control hierarchy operates on a different time scale. Typically, the control structure includes (*i*) scheduling (weeks), (*ii*) site-wide (real-time) optimization (days), (*iii*) local optimization (hours), (*iv*) Supervisory or Model predictive control (MPC, minutes), and finally (*v*) Stabilizing and regulatory control (seconds). Layers (*ii*) and (*iii*) deal with the economic optimization of the system while the supervisory and regulatory layers are concerned with setpoint tracking from the layers above. Thus, these layers are interconnected through the controlled variables [21].

The supervisory layer is a slow economic layer, whose objective is to perform advanced economic control on controlled variables that are considered of high worth. In return, this layer provides setpoints for controlled variables of less economic significance in the lower regulatory layers. In this thesis the supervisory control consists firstly of a real time optimization (RTO) to find optimal steady states that can be passed further as tracking point of the second layer. The second layer is done using model predictive control

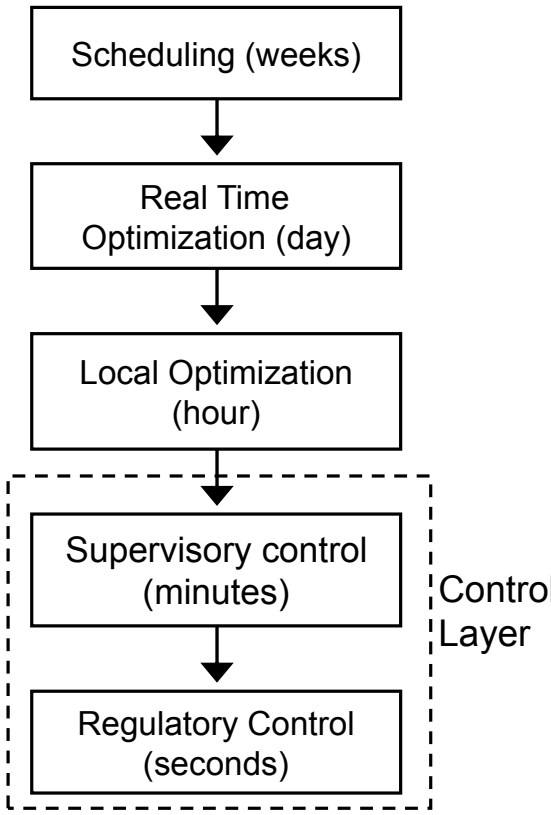


Figure 2.10: Typical control hierarchy in a chemical plant

(MPC) which attempts to track the optimal steady state point provided by the RTO. Other advanced control structures can operate in this layer, namely, cascade, ratio , decoupler valve position control, etc.

Chapter 3

System Description and Modeling

This chapter shows a modified approach for derivations of the gas lift and riser models as basis of the models previously developed in [22] and [23]. The main modifications done to the model are:

1. The use of a simplified friction equation to produce a stable system that reaches steady state.
2. The addition of water cuts which makes the gas lifted oil network a three phase system.

The simplified schematic of the gas lift is illustrated in Figure 3.2. Reservoir fluid flows through a perforated well, into the wellbore, upwards through the tubing and through a production choke, before it enters the riser. Each part of the network is treated as an independent building component of a gathering network where any of the these systems can be represented generally by the structure shown in Equations 3.1. This chapter is dedicated to introduce the reader to the gas lifted oil network models. Finally, formulation of the entire network as a system of DAEs is developed. The resulting model codes are given in Appendix C.1 and C.2.

3.1 Gas lifted oil network

This thesis connects the models developed for the gas lift and the riser to a network of three wells, three gas lifts, and a single riser. A simple

schematic of the entire system is depicted in Figure 3.1. In this gas lifted oil network, the gas injection into each gas lift is assumed to be supplied from different sources with similar properties and pressures, where these sources are independent. Usually, it is possible to assume that the gas injection is reused from top facilities such as gravity separators. Furthermore, the manifold in the system is not a tank, rather, it is thought of as a point at which all production flows from each well are gathered and jointly directed in the riser to the topside. The flows in the system described in Figure 3.1 are always travelling vertically upwards at all points, hence there is no horizontal flow movement. In addition, in Figure 3.1 all green shapes represent a flow through a valve which is controlled by a manipulated valve opening u as described in Equations 3.13 for gas injection, 3.35 for well production and 3.53 for riser production.

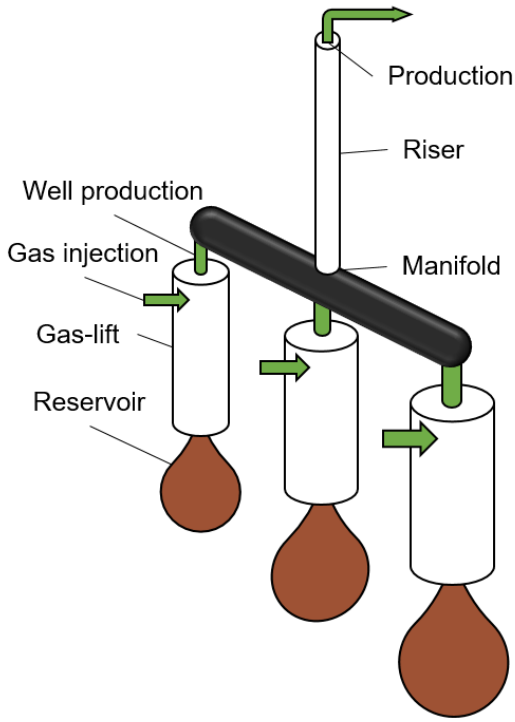


Figure 3.1: System network of oil production. Three reservoirs (brown) with three gas lifts connected by a manifold (Black) to pump oil jointly through a riser. Green shapes indicate flows through a valve, namely, gas injection into the gas lift, well production into manifold, and on the top total production choke.

As will be seen in the following sections, this system is highly coupled and very nonlinear. In order to find the well production and solve the mathematical models for the gas lifts, the manifold pressure calculated in the riser model is required. Moreover, to find the manifold pressure in the riser the production from the wells must be provided. Hence, there is the mutual dependency. However, using advanced programming tools makes this simulation possible and successful. The software used is mentioned in the final section of this chapter.

3.2 Generalized submodel

Generally, any of the subsystems of the gas lifted oil production network can be described by the ODE system in Equations 3.1, where the subscript c refers to any component of the network. In the case of this master thesis the reference is $c = \{well\ 1, well\ 2, well\ 3, riser\}$. Further, the differential states x_c represent the mass of gas and liquid in each component of the network. and can be described by the dynamic function f_c. Equation 3.1b represents algebraic variables of the DAE system by the algebraic function g_c, such as the pressures and mass flow rates of gas and liquid phases through the reservoirs and valves.

$$\dot{x}_c = f_c(x_c, u_c) \tag{3.1a}$$
$$y_c = g_c(x_c, u_c) \tag{3.1b}$$

3.3 Friction equation

This section introduces the new friction equation used in the tubing of the gas lifted oil network. This is part one of the work in this thesis, and it is one of the major changes produced in the modeling part to the models developed in [23] and [22].

In the tubing of the gas lift and riser models, calculating the pressure drop due to friction can be very challenging. Equations 14-19 in article [23] define an average flow rate from the reservoir to the tubing, $\bar{w}_{res}$, which is a constant in the model developed to avoid the circulating dependency and mutual dependence of the friction to the flow. In addition, [23] introduces superficial velocity terms which are necessary for the friction equation to

reflect the volumetric flow rate which is dependent on the mass gas oil ratio (GOR). Those superficial velocities are calculated for both the gas phase in the tubing, and the liquid phase in the tubing. Then an average mixture velocity is calculated by summing up the two superficial velocities. Finally, a friction factor from the implicit Colebrook-White equation, [10], is used to find the pressure loss due to friction from the top of the tubing down to the injection point. Hence the friction model in [23], can be considered as an estimated parameter that depends only on the injection rate in time, given that it is the only time variant variable in Equations 14-19 in article [23].

The friction model that is deployed in this thesis work, relies on mutual dependency and standard fluid dynamics notations to describe the friction. Namely the Hagen-Poiseuille equation, Equation 3.2. is used to describe the pressure drop in the flow through a long cylindrical pipe of constant cross section [16]. Previously, this equation was successfully applied to air flow through a drinking straw and a hypodermic needle. Now this thesis applies the equation to the tubing in gas lifted oil production. Because the gas lift model is a highly nonlinear and coupled system, a minor change in some of the parameters can result in different simulation run time and results. It is important to use simple versions of the pressure drop due to friction.

$$P_{fr} = 128\frac{\mu L Q}{\pi D^4} \tag{3.2}$$

$$Q = \frac{w_{out}}{\bar{\rho}} \tag{3.3}$$

In Equation 3.2 P_{fr}, is the pressure drop or pressure reduction, L is the length of the pipe, μ is the dynamic viscosity, and D is the pipe diameter. The volumetric flow rate can be calculated through Equation 3.3 where w is the mixture mass flow rate out of the pipe, and $\bar{\rho}$ is the average flow density mixture over the pipe. The Hagen-Poiseuille equation is assumed to be valid in the gas lift model due to its application on laminar flow. However, for velocities and pipe diameters above a threshold, can lead to larger pressure drop than calculated by the Hagen-Poiseuille equation. Further, the value of the pressure drop in the gas lift is considerably low. Therefore calculating pressure drop using 3.2 should be sufficient and its quantity remains well defined even in the case of varying flows.

3.4 Water cut

In addition to the new friction equation, the water cut, (WC), parameter is introduced to take into consideration the water produced during reservoir oil production. Hence, the behaviors of the three phases in the system are taken into account. It is necessary to introduce WC in the calculations to have a more accurate representation of the liquid densities in different parts of the system during the life time of the reservoir. The change in WC can be considered as a disturbance to the system as it becomes larger and it's effects become more significant. Hence, for changing WC values, the density of liquid, ρ_L and the viscosity of the fluid, μ, must be adjusted. This is done by applying the following equations:

$$\rho_L = \frac{1}{\frac{WC}{\rho_{water}} + \frac{1-WC}{\rho_{oil}}} \tag{3.4}$$

$$\mu = \frac{\mu_{oil}}{(1+WC)^{2.5}} \tag{3.5}$$

3.5 Well model

This section introduces the flow calculations of the gas lift model in the annulus and tubing parts. The derivation of the equations are done derived sequentially, while showing the necessary steps in the process. Furthermore, this section also shows how the pressures, phase fractions and the frictions are calculated, which are essential for the flow calculations. The list of symbols is presented on page xvi and the constants used in the model are shown on page 81. Finally in this section the modeling refers to Figure 3.2 which describes the gas lift design [23].

3.5.1 Basis and mass balances

The basis of the gas lift model consist of three states, (i) The mass of gas in the annulus x_1, (ii) The mass of gas in the tubing x_2, and (iii) the mass of liquid in the tubing x_3. The gas lift is modeled as a vertical cylindrical tank filled with gas at a constant temperature. The state Equations in 3.6 are simple mass balance equations for each phase (liquid and gas) in the annulus part and the tubing part of the gas lift. Each mass balance is built

by the inflow and the outflow of the respective control volume (annulus and tubing). In the model represented by Equation 3.6a, $w_{G_a,in}$ is the inlet mass flow rate of gas supplied to the gas lift through the annulus, and $w_{G_a,inj}$ is the outlet mass flow rate injected at the bottom of the annulus to the tubing, hence the subscript a denotes the annulus part. In the second part, Equations 3.6b and 3.6c describe the mass balance of gas and liquid in tubing, subscript t, respectively. Where $w_{G_t,out}$, $w_{G_{res}}$ and $w_{L_{res}}$, denote outlet gas mass flow from tubing, and inlet gas and liquid mass flow from reservoir into tubing, respectively. Equation 3.6 takes into consideration both gas and liquid present in the reservoir, hence it can describe the entire gas lift model.

$$\frac{dx_1}{dt} = \dot{x}_1 = w_{G_a,in} - w_{G_a,inj} \tag{3.6a}$$

$$\frac{dx_2}{dt} = \dot{x}_2 = w_{G_a,inj} + w_{G_{res}} - w_{G_t,out} \tag{3.6b}$$

$$\frac{dx_3}{dt} = \dot{x}_3 = w_{L_{res}} - w_{L,out} \tag{3.6c}$$

3.5.2 Flow into annulus

The modeling of the artificial gas lift presented in Figure 3.2 begins on the annulus side, where the gas flow through the gas lift choke, $w_{G,in}$ on the top left of Figure 3.2, is calculated. Through the gas lift choke and annulus, only gas is present and thus the ideal gas law is a valid assumption and hence the pressure on the top of the annulus can be derived using a constant temperature T_a over the whole annulus. The latter assumption is valid due to the slow variation in temperature over time [10]. The density of gas before the choke can be calculated by Equation 3.7, where P_{gs} is the pressure of the gas lift source and Mg is the gas molecular weight.

$$\rho_{G,in} = \frac{P_{gs}Mg}{T_aR} \tag{3.7}$$

The density of gas on the right side of the gas lift choke, annulus part, is given by Equation 3.8. Which is a simple division of the gas mass by the Total constant volume of the annulus, denoted by V_a.

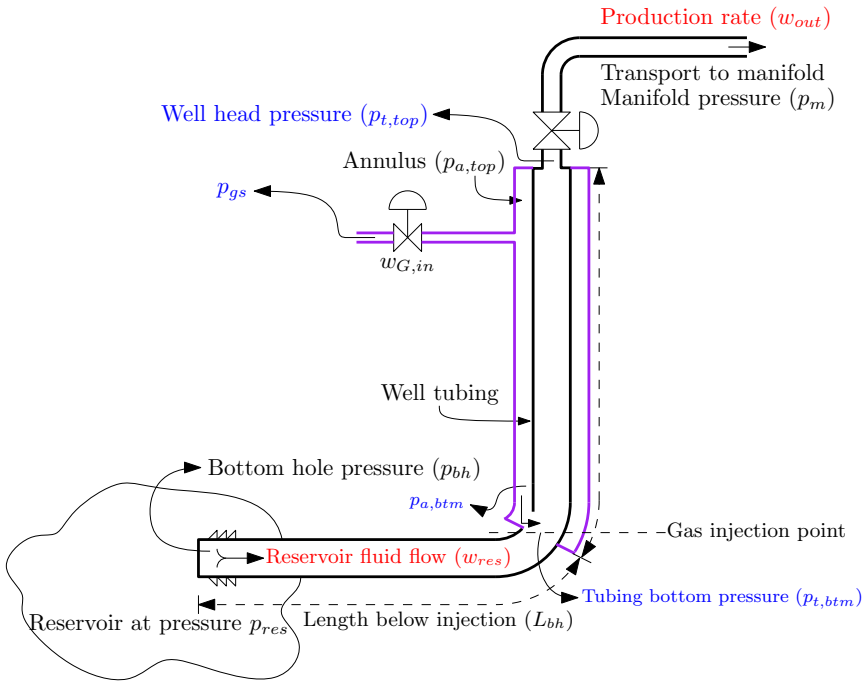


Figure 3.2: Schematic representation of the gas lift model.

$$\rho_{G,a} = \frac{x_1}{V_a} \tag{3.8}$$

Thus, the pressure at the top of the annulus $P_{a,top}$ can be written as in Equation 3.9.

$$P_{a,top} = \frac{\rho_{G,a} R T_a}{Mg} \tag{3.9}$$

Subsequently, the pressure of the annulus in the bottom near the injection point $P_{a,btm}$ can be calculated simply by adding the gravitational pressure drop.

$$\Delta P_a = \rho_{G,a} L_a g \tag{3.10}$$

$$P_{a,btm} = \Delta P_a + P_{a,top} \tag{3.11}$$

Similarily, the density of gas can be found at the bottom of the annulus by taking into consideration the ideal gas law, as follows

$$\rho_{G,a,btm} = \frac{P_{a,btm} M g}{T_a R} \tag{3.12}$$

Hence, the gas mass flow into the annulus is derived in Equation 3.13 which is based on the valve model and the pressure difference on the top before and after the gas lift choke.

$$w_{G,in} = K_{gs} u_2 \sqrt{\rho_{G,a}(P_{gs} - P_{a,top})} \tag{3.13}$$

In Equation 3.13 the pressure drop is the driving force for the flow through the gas lift choke, when P_{gs} is greater than $P_{a,top}$ the flow though the choke increases, given that P_{gs} is always greater than $P_{a,top}$. However, if P_{gs} is lower than $P_{a,top}$ there will be no flow through the choke. Usually Equation 3.13 is restricted by a max function to insure that the term under the root is always positive semi definite, however in this thesis this function was eliminated due to the use of optimizers that do not accept max or if functions during optimization. That is, because the introduction of such discontinuity into a model can have disadvantageous effects on the solver's ability to efficiently obtain an accurate solution due to the introduction of non-smooth gradients. Further, K_{gs} is a constant valve parameter and u_2 is the choke opening, otherwise know as the input or the manipulated variable.

3.5.3 Gas injection into tubing

As shown in the previous section the valve model is a good mathematical description of the flow over a point. Therefore, according to the valve model the pressures, before and after a specific point of transfer, must be found. The procedure in this section is simply to find expressions for the pressures at different points of the tubing all the way down to the injection point, once the latter is found the valve equation can be applied.

In this model the reservoir is assumed to be saturated, that is the pressure of the reservoir is at its bubble point where it is ready to vaporize. Under these circumstance the reservoir is completely in liquid phase. Subsequently, everything below the injection point is assumed to be in liquid phase, see Figure 3.2. Which implies that the length of the bottom hole, L_{bh}, is only liquid. Hence, the average density of gas above the tubing can be calculated

for the volume where gas is present, excluding the bottom hole area. This becomes, the total volume of the tubing, V_t, subtracted by the liquid volume, x_3/ρ_L, all of which are above the injection point and excluding the bottom hole volume, $V_{bh} = S_{bh}L_{bh}$. See Appendix A for constants used in the model. Thus the average density of gas in the tubing above the injection point can be formulated as in Equation 3.14.

$$\bar{\rho}_{G,t} = \frac{x_2}{V_t + V_{bh} - \frac{x_3}{\rho_L}} \tag{3.14}$$

Additionally, The average density of the entire mixture of gas and liquid in the tubing part above the injection point can be described by Equation 3.15.

$$\bar{\rho}_{mix} = \frac{x_2 + x_3 - V_{bh}\rho_L}{V_t} \tag{3.15}$$

Utilizing Equation 3.14 and the Ideal gas law assumption, the pressure at the top of the tubing can be obtained by the gas law.

$$P_{t,top} = \frac{\bar{\rho}_{G,top}RT_t}{Mg}, \tag{3.16}$$

The pressure at the bottom of the tubing near the injection point can be found by merely adding hydrostatic pressure forces to pressure loss due to friction. Thus, the pressure loss due to the gravitational force can be described by Equation 3.17, where L_t denotes the length from the top of the tubing down to the injection point. The average density of mixture in the tubing , $\bar{\rho}_{mix}$, found earlier is used to account for the gas present along the tubing.

$$\Delta P_t = \bar{\rho}_{mix}L_t g, \tag{3.17}$$

In addition, the pressure loss due to friction along the surface of the tubing, $P_{t,fr}$, is found. This is done by applying the Hagen-Poiseuille Equations 3.3 and 3.2 to the tubing section above the gas injection point. Thus, Equations 3.18 and 3.19 are found. Where w_{out} is the mass flow rate of mixture leaving the tubing to account for the entire tubing. Since w_{out} is yet to be found, mutual dependency becomes noticeable.

$$Q_t = \frac{w_{out}}{\bar{\rho}_{mix}}, \tag{3.18}$$

Further, the pressure loss due to friction and wall stress in the tubing can be formulated as in Equation 3.19. Where the average volume fraction in the tubing, $\bar{\alpha}_{L,t}$, is described in Equation 3.32.

$$P_{t,fr} = 128\frac{\bar{\alpha}_{L,t}\mu L_t Q_t}{\pi D_t^4}. \tag{3.19}$$

Finally, summing up the top pressure and the pressure losses will yield the pressure on the bottom of the tubing near the gas injection point.

$$P_{t,btm} = P_{t,fr} + \Delta P_t + P_{t,top}. \tag{3.20}$$

Given that the pressures on both ends of the injection point are available, the injected gas flow rate can be developed. This is done by combining Equations 3.11, 3.12 and 3.20 in the valve model, which yields Equation 3.21. Where K_{inj} is the constant valve parameter.

$$w_{G,inj} = K_{inj}\sqrt{\rho_{G,a,btm}(P_{a,btm} - P_{t,btm})} \tag{3.21}$$

Despite the absence of an actual valve at the injection point, Equation 3.21 is an adequate description the flow. Therefore, it is assumed that the fraction input, u, is always fully open. Hence, $u = 1$.

3.5.4 Flow from reservoir into tubing

This section shows how the mass flow rate out of the reservoir, w_{res}, is calculated.

Initially, the gas to oil ratio (GOR) is defined as the ratio between the rate of gas and liquid mass flowing from the reservoir into the tubing, $\bar{w}_{G,res,out}/\bar{w}_{L,res,out}$. GOR is an important given constant parameter in dimensionless metric units, which affects the productivity and efficiency of the oil production from reservoirs [20]. GOR is used to find the mass fraction of gas at the bottom of the tubing, through Equation 3.22.

$$\alpha^{m}_{G,btm} = \frac{GOR}{GOR+1} \tag{3.22}$$

At the bottom hole, the pressure losses are formulated using the parameters and constants for the bottom hole section. Where L_{bh} is the length of the bottom hole section, Q_{bh} is calculated using total reservoir outflow w_{res} with ρ_L since it is assumed that the flow in this section right before the exit is at very high pressure and the mixture is in the liquid phase. Finally, the diameter of the bottom hole is denoted with D_{bh}.

$$\Delta P_{bh} = \rho_L L_{bh} g \tag{3.23}$$

$$Q_{bh} = \frac{w_{res}}{\rho_L} \tag{3.24}$$

$$P_{bh,fr} = 128\frac{\mu L_{bh} Q_{bh}}{\pi D^4_{bh}} \tag{3.25}$$

Thus the pressure at the bottom hole P_{bh} is calculated by summing the hydrostatic pressure force and the pressure loss due to friction at the bottom of the tubing near the injection point, $P_{t,btm}$. Hence, Equation 3.26 is formulated.

$$P_{bh} = P_{t,btm} + \Delta P_{bh} + P_{bh,fr} \tag{3.26}$$

The mass flow rate from the reservoir w_{res} can be calculated by Equation 3.27, PI is a given parameter which describes the productivity index. PI replaces the gain and the input fraction, that is, it describes the ability of the reservoir to pump fluids from the reservoir to the tubing. In addition, the pressure of the reservoir is also a given quantity P_{res} which, in general, has a large effect on the production of oil from reservoirs.

$$w_{res} = PI\sqrt{P_{res} - P_{bh}} \tag{3.27}$$

When the fluid from the reservoir passes the bubble point part of the liquid evaporates. Thus, the gas developed must be taken into consideration. Moreover, $\alpha^{m}_{G,btm}$ can be used to find the liquid and gas mass flow from the reservoir, according to Equations 3.28 and 3.29 respectively.

$$w_{L,res} = (1 - \alpha^m_{G,btm}) w_{res} \tag{3.28}$$

$$w_{G,res} = (\alpha^m_{G,btm}) w_{res} \tag{3.29}$$

3.5.5 Flow from tubing into manifold

Here, the mass flow out of the production choke is calculated. Previously, expressions for the pressure on the top of the tubing, $P_{t,top}$, and the average density over the tubing above the injection point, $\bar{\rho}_{mix}$, have been developed. In this section a more accurate description of the mixture density on the top of the tubing is developed. This is done by taking into consideration the additional flows of liquid and gas flowing from the reservoir. Given that the reservoir flow has been exposed to an environment with lower pressure. The calculations here are based on the work of [22] and [23]. Using the pressure at the bottom of the tubing $P_{t,btm}$, a more accurate expression for the density at the same point is developed using the ideal gas law and the temperature at the bottom hole, T_{bh}. The latter will yield Equation 3.30:

$$\rho_{G,t,btm} = \frac{P_{t,btm} M g}{T_{bh} R} \tag{3.30}$$

The density, $\rho_{G,t,btm}$, is a useful quantity that is utilized to produce the volumetric fraction of liquid at the bottom of the tubing, through Equation 3.31.

$$\alpha_{L,t,btm} = \frac{w_{L,res} \rho_{G,t,btm}}{w_{L,res} \rho_{G,t,btm} + \rho_L (w_{G,res} + w_{G,inj})} \tag{3.31}$$

Further, the average volume fraction of liquid $\bar{\alpha}_{L,t}$ inside the tubing above the injection point is calculated as follows:

$$\bar{\alpha}_{L,t} = \frac{x_3 - V_{bh} \rho_L}{V_t \rho_L} \tag{3.32}$$

$\bar{\alpha}_{L,t}$ can be considered as the average liquid volume fraction at the middle of the tubing, which can be calculated from the liquid volume fractions at both ends of the tubing. This is given the assumption made by [22], which states that in a two phases vertical pipe the gradient is constant,

given that the relationship between the pressure and the volume fraction is linear. Therefore from the two quantities found above it is possible to use the simple relation through Equation 3.33:

$$\alpha_{L,t,top} = 2\bar{\alpha}_{L,t} - \alpha_{L,t,btm}. \tag{3.33}$$

Using the liquid volumetric fraction, $\alpha_{L,t,top}$, the density of mixture at the top of the tubing can be expressed as the following:

$$\rho_{mix,t,top} = \alpha_{L,t,top}\rho_L + (1 - \alpha_{L,t,top})\bar{\rho}_{G,t} \tag{3.34}$$

The valve equation can be used to find the total mass flow rate exiting the tubing through the production choke.

$$w_{out} = K_{pr}u_1\sqrt{\rho_{mix,t,top}(P_{t,top} - P_m)} \tag{3.35}$$

In Equation 3.35, u_1 is the fraction input of the production choke, K_{pr} is the production choke constant and P_m is the pressure at the manifold. P_m is a very important quantity which depends on many variables such as the top side pressure, friction pressure and the outflow through each production choke from each well in the gathering system. The manifold pressure calculations are show in Section 3.6.

Furthermore, to calculate the composition of the flow rate, the gas mass fraction on the top side of the tubing $\alpha^m_{G,t,top}$ is required. This is expressed via following

$$\alpha^m_{G,t,top} = \frac{(1 - \alpha_{L,t,top})\bar{\rho}_{G,t}}{\alpha_{L,t,top}\rho_L + (1 - \alpha_{L,t,top})\bar{\rho}_{G,t}} \tag{3.36}$$

Finally, given $\alpha^m_{G,t,top}$ and w_{out}, the gas and liquid mass outflows can be expressed as the following:

$$w_{G,out} = \alpha^m_{G,t,top}w_{out} \tag{3.37}$$

$$w_{L,out} = (1 - \alpha^m_{G,t,top})w_{out} \tag{3.38}$$

3.6 Riser model

The mathematical derivations of the flows and pressures through and in the riser model, are developed here. The riser gathers all liquid and gas outflows from the different wells through a manifold and pumps the oil to top facilities. These facilities are usually gravity separators or other advanced reservoir production equipment. The basis of the model is first presented and later the mathematical modeling equations of the riser. In this section all variables and constants are applied only to the riser, thus similar notations do not explicitly mean that they are equivalent to other variables developed earlier or after.

3.6.1 Mass balance in riser

The basis of the riser model consist of two states, (i) The mass of gas in the riser x_1, and (ii) the mass of liquid in the riser x_2. The riser, similar to the tubing in the gas lift, is modeled as a vertical cylindrical tank filled with gas at a constant temperature. The state Equations 3.39 are also a simple mass balance equations of each phase (liquid and gas) in the riser. The mass balance equations in 3.39 are built using the total inflow of gas and liquid from the wells and the outflow of gas and liquid from the riser. Note, x_1 and x_2 are not equivalent to those in Section 3.5.

$$\frac{dx_1}{dt} = \dot{x}_1 = w_{tot,G,in} - w_{G,out,r} \tag{3.39a}$$

$$\frac{dx_2}{dt} = \dot{x}_2 = w_{tot,L,in} - w_{L,out,r} \tag{3.39b}$$

In Equations 3.39, $w_{G,out,r}$ and $w_{L,out,r}$ denote gas and liquid outflow rates from the riser, respectively. Expressions for these quantities will be developed later in this part of the chapter. Moreover, $w_{tot,G,in}$ and $w_{tot,L,in}$ denote the total inflow of both gas and liquid arriving from the wells, which are the sum of the wells' outflows. Equations 3.40 show the summing operation, where n denotes the number of wells, in this case $n = 3$.

$$w_{tot,L,in} = \sum_{i=1}^{n} w_{L,out,i} \tag{3.40a}$$

$$w_{tot,G,in} = \sum_{i=1}^{n} w_{G,out,i} \tag{3.40b}$$

3.6.2 Flow through top production choke

The riser model follows a similar modeling scheme and procedure as in the tubing part of each well. The riser is simply a vertical cylindrical pipe carrying the gathered fluids and gas from each well to the top side facilities. The main driving force that pumps the fluids to the top is the pressure differences. In order to find the production outflow, the valve equation is used over the top production choke.

Initially, the manifold pressure, P_m, must be found. This is done by summing the pressure losses due to gravity and friction to the riser's pressure at the top. The density of the gas on the top side can be calculated according to the following Equation:

$$\rho_{G,top} = \frac{x_1}{V_r - \frac{x_2}{\rho_L}} \tag{3.41}$$

Where V_r is the total volume of the riser (cylindrical tank), and by subtracting the volume occupied by the liquid, $\frac{x_2}{\rho_L}$, the total volume occupied by gas in the riser is found. Hence, the pressure on the top side of the riser is found using the ideal gas law:

$$P_{r,top} = \frac{RT_r\rho_{G,top}}{Mg} \tag{3.42}$$

Further, to find the friction pressure loss the average density of the mixture over the entire riser volume is required:

$$\bar{\rho}_{mix} = \frac{x_1 + x_2}{V_r} \tag{3.43}$$

Using the average density, the total volumetric flow is constructed using Equation 3.44. Where $w_{out,r}$ is yet to be found, hence, the mutual dependency appearance.

$$Q_r = \frac{w_{out,r}}{\bar{\rho}_{mix}} \tag{3.44}$$

Using the the volumetric flow found above, the pressure loss due to friction is found by applying Hagen-Poiseuille Equation 3.2. In Equation 3.2, D_r denotes the riser's diameter, L_r denotes the length of the riser, $\bar{\alpha}_L$ is the average volume fraction of liquid in the riser given in Equation 3.48, finally μ designates the fluid's viscosity constant.

$$P_{r,fr} = 128\frac{\bar{\alpha}_L \mu L_r Q_r}{\pi D_r^4} \tag{3.45}$$

Furthermore, the pressure loss due to gravity is

$$\Delta P_r = \bar{\rho}_{mix} L_r g. \tag{3.46}$$

By summing up the pressures found above, an expression for the manifold pressure can be developed as shown in Equation 3.47.

$$P_m = P_{r,top} + \Delta P_r + P_{r,fr} \tag{3.47}$$

Next, to a find a more accurate description of how the density varies on the top side of the riser, a similar approach is done as in the previous section. The average volume fraction of the liquid, $\bar{\alpha}_L$, inside the riser is calculated as follows:

$$\bar{\alpha}_L = \frac{x_2}{V_r \rho_L} \tag{3.48}$$

Using the pressure at the manifold point, ideal gas law and the temperature T_r, will yield an expression to the density at the bottom of the riser:

$$\rho_{G,btm} = \frac{P_m M g}{R T_r} \tag{3.49}$$

Which is again a very useful quantity used to formulate a volumetric fraction of liquid at the bottom of the riser, through Equation 3.50.

$$\alpha_{L,btm} = \frac{\rho_{G,btm} w_{tot,L,in}}{\rho_{G,btm} w_{tot,L,in} + \rho_L w_{tot,G,in}} \tag{3.50}$$

Making the assumption that the volume fraction relationship is approximately linear in a vertical pipe [22], the top liquid volume fraction can be found by:

$$\alpha_{L,top} = \alpha_{L,btm} + 2\bar{\alpha}_L \tag{3.51}$$

Utilizing $\alpha_{L,top}$ found above, a more accurate description of the mixture density at the top, $\rho_{mix,top}$, is developed through:

$$\rho_{mix,top} = \alpha_{L,top}\rho_L + (1 - \alpha_{L,top})\rho_{G,top} \tag{3.52}$$

By combining the expression for $\rho_{mix,top}$ and the valve model, the total mass flow exiting the riser, $w_{r,out}$, through the production choke is described by:

$$w_{out,r} = K_{pr,r} u \sqrt{\rho_{mix,top}(P_{r,top} - P_s)} \tag{3.53}$$

In Equation 3.53, $K_{pr,r}$ is the production choke constant, P_s is the pressure in the top facilities accepting the oil, and u is the fraction input of the valve (otherwise know as the manipulated variable). The value of u here ranges between 0 and 1 and describes the opening of the valve. In all of the case studies in this thesis this valve is left fully open to maximize production.

Finally, the gas mass fraction on the top side of the riser, $\alpha^m_{G,top}$, is used to find the respective outflows of each phase. Namely the gas phase outflow, $w_{G,out,r}$, and the liquid phase outflow,$w_{L,out,r}$. These are described by the following equations respectively. The subscript L denotes total liquid or fluid which contains oil and water

$$\alpha^m_{G,top} = \frac{(1 - \alpha_{L,top})\rho_{G,top}}{\alpha_{L,top}\rho_L + (1 - \alpha_{L,top})\rho_{G,top}} \tag{3.54}$$

$$w_{G,out,r} = \alpha^m_{G,top} w_{out,r} \tag{3.55}$$

$$w_{L,out,r} = (1 - \alpha^m_{G,top}) w_{out,r} \tag{3.56}$$

3.7 Gas lifted oil network of DAEs

The mathematical models presented in this chapter are expressed by *Differential Algebraic Equations* (*DAEs*). This is presented in the following form:

$$\begin{aligned} \dot{x} &= f(x,y,u,t) \\ 0 &= g(x,y,u,t) \end{aligned} \tag{3.57}$$

Where f describes the differential part and g describes the algebraic part of the DAE. This is a semi explicit DAE, where the differential and algebraic parts are decomposed. DAEs are characterized by their index. The *index* of a DAE is the smallest number of differentiation required to obtain an ODE. The models of the system shown in Figure 3.1, can be described by a DAE system with a large index number. Generally, the higher the index, the greater the numerical difficulty becomes. The system's DAEs can be described by following:

Gas lift Differentials part

$$\dot{x}_1 = w_{G_a,in} - w_{G_a,inj} \tag{3.58a}$$

$$\dot{x}_2 = w_{G_a,inj} + w_{G_{res}} - w_{G_t,out} \tag{3.58b}$$

$$\dot{x}_3 = w_{L_{res}} - w_{L,out} \tag{3.58c}$$

Gas lift Algebraic Part

$$0 = w_{G,inj} - K_{inj}\sqrt{\rho_{G,a,btm}(P_{a,btm} - P_{t,btm})} \tag{3.58d}$$

$$0 = w_{res} - PI\sqrt{P_{res} - P_{bh}} \tag{3.58e}$$

$$0 = w_{out} - K_{pr}u_1\sqrt{\rho_{mix,t,top}(P_{t,top} - P_m)} \tag{3.58f}$$

Equations 3.58 describe how the DAE system was formulated for each of the wells. In total there are three differential variables, $\{x_1, x_2, x_3\}_i$ and three algebraic variables, $\{w_{G,inj}, w_{res}, w_{out}\}_i$, where $i = \{well\ 1, well\ 2, well\ 3\}$ denotes the well number.

Similarly, A DAE system is also formulated for the riser model which has the following form:

Riser Differentials part

$$\dot{x}_1 = w_{tot,G,in} - w_{G,out,r} \tag{3.59a}$$
$$\dot{x}_2 = w_{tot,L,in} - w_{L,out,r} \tag{3.59b}$$

Riser Algebraic Part

$$0 = w_{out,r} - K_{pr} u \sqrt{\rho_{mix,r,top}(P_{r,top} - P_s)} \tag{3.59c}$$
$$0 = w_{G,out} - \alpha^m_{G,t,top} w_{out} \tag{3.59d}$$
$$0 = w_{L,out} - (1 - \alpha^m_{G,t,top}) w_{out} \tag{3.59e}$$

Manifold Algebraic Part

$$0 = P_m - (P_{r,top} + \Delta P_r + P_{r,fr}) \tag{3.59f}$$

Equations 3.59 describe how the DAE system was formulated for riser model. In total there are two differential variables, $\{x_1, x_2\}$ and three algebraic variables, $\{w_{out}, w_{G,out}, w_{L,out}\}$, as well as the algebraic manifold pressure P_m variable.

All other quantities are not mutually dependent, therefore it is unnecessary to include them in the DAE system rather they are treated accordingly through a step by step algorithm.

3.8 Software package

For model simulation purposes the `Matlab` programming environment was used. The well and riser models were implemented separately in `Matlab` as script files, and coupled later in another script file which represents the plant. The mathematical models for the gas lifted oil network were initially solved using `fsolve` for steady state conditions, and later integrated using the `ode15s` for dynamic DAE stiff systems. The `ode15s` was used because it can solve problems with a singular mass matrix, otherwise known as differential-algebraic equations (DAEs).

Chapter 4

Formulation of Optimization Problems

In section 2.3 an OCP was developed and solved to produce a single optimal control trajectory. One might use the obtained optimal control trajectory solution to control the real process. However, because the real model typically deviates, from the model developed in the optimization problem, the precomputed control trajectory is not always satisfying [4] and [15]. The latter might lead to not reaching the points that the model trajectory predicted. Therefore, the process can be monitored for irregularities and unexpected behaviours during the time development. By being able to monitor and detect model development, the optimal control inputs or trajectory can be modified to get the best possible solution. For example, if the pressure at the manifold is not at the set point due to gas to oil ratio changes in the reservoir, the optimal control trajectory is recomputed. The latter is also described as *feedback control*.

This chapter attempts to bridge the gap between optimization and control of the oil production system shown on page 22. This is done by first presenting the concepts which merge feedback control with optimization. Namely, *Real Time Optimization (RTO)* and *Nonlinear Model Predictive Control (NMPC)*. In addition, this chapter shows the methodology used to build a nonlinear program from the collocation method and presents the software package used to solve this large NMPC system.

4.1 Real time optimization

One approach to compute the optimal feedback control mentioned earlier is in *real-time optimization* (RTO) during the run-time of the process. For example, in the case of the oil production, after the occurrence of a disturbance, the optimization is called to be solved again in order to compute the control trajectory again. However, in reality the optimal control problem might be simplified in order to speed up computation time, e.g. by predicting only a finite amount of time into the future, in addition, algorithms might need to be adapted to new tasks, thus the optimization needs to be solved again. This process is challenging and requires a long time, since numerical optimization is usually carried out on embedded hardware, [4]. The advantages of RTO approach are the flexibility provided in formulating the objective and the process model, and the capability to directly handle equality and in equality constraints of nonlinear dynamic process models.

In this thesis the RTO method is used in order to find the optimal steady state values of the input controls $\mathbf{u}_{opt,SS}$, and the differential and algebraic states $\mathbf{x}_{opt,SS}$ and $\mathbf{z}_{opt,SS}$, respectively. Hence, by solving this problem the optimal steady states can be passed to another controller, which finds suitable control trajectories. The RTO control problem can be formulated via the following:

$$\min_{u} \; L(x,z,u) \tag{4.1a}$$

subject to

$$f(x,z,u) = 0, \tag{4.1b}$$
$$g(x,z,u) = 0, \tag{4.1c}$$
$$x^{low} \leq x \leq x^{high}, \tag{4.1d}$$
$$z^{low} \leq z \leq z^{high}, \tag{4.1e}$$
$$u^{low} \leq u \leq u^{high}. \tag{4.1f}$$

In Equations 4.1, the differential variables are represented as $\mathbf{x}_c$ and the algebraic variables as $\mathbf{z}_c$, where $c = \{well\,1, well\,2, well\,3, riser\}$. These are chosen based on the DAE system developed in Equations 3.58 and 3.59. In addition, the control inputs of the system consist of 3 gas injection valves (u_2) into the wells, and 4 production valves (u_1) from the wells and the riser, see page 22 for plant visualization. Further, the objective function

is the Lagrange term in the continuous form as discussed in 2.3. In this case the RTO represents the economic objective of maximizing production $w_{L,out}$, while minimizing gas injection $w_{G,inj}$, see Equations 3.59e and 3.58d for the flow rates.

In Equations 4.1, f and g describe the differential and algebraic equations in each component of the system. See section 3.7 where the models were represented as a system of differential algebraic equations. In the RTO f is constrained to be equal to zero in order to produce the optimal steady state solutions, at which the dynamics are equal to zero.

Each of the variables $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{u}$ are subject to boundary conditions, which are indicated by the inequalities 4.1d, 4.1e and 4.1f. The lower and upper bounds are specified in Appendix C.4. The boundary conditions are essential and one of the main advantage of using the RTO method. They indicate under which conditions the system should operate to optimize a process, e.g., if the production valve of *well* 1 is desired to be open at all times, then the inequality 4.1f is specified as $\{1 \leq u_1(t) \leq 1 \Rightarrow u_1 = 1\}$. Similarly, values can be relaxed by the boundaries, e.g., if $\{0 \leq u_1(t) \leq 1\}$, then u_1 is allowed to be at any valve opening position. Hence, the RTO produces the steady state point at which the production is optimal.

In conclusion, The RTO's optimal control problem described by Equation 4.1, is solved less often in order to find the optimal steady state solutions of the decision variables $\mathbf{x}_{opt,SS}$, $\mathbf{z}_{opt,SS}$ and $\mathbf{u}_{opt,SS}$ while taking into account the disturbances that occur on the system periodically.

4.2 MPC strategy

During the operation of the real process, state variables and system parameters, such as the GOR and WC, are most likely subject to disturbances. In addition, the plant measurements could be subject to low or high noise. Hence, an optimal closed loop or feedback control approach would be much preferable, giving the optimal control for a sufficiently large range of time points t_0, initial values x_0 and parameters p_0. In other words the solution is computed at time t_0 and recomputed at every time step where the final time t_f progresses with t_0, i.e. $t_f - t_0 = T$. Here, T is the prediction horizon. This approach is called Model Predictive Control (MPC), [9], [4],[15],[5].

The MPC strategy can be formulated in the following steps:

1. Observe the system's present state $\bar{x}_0$.
2. Predict and optimize the future behaviour of the system for N steps in the prediction horizon starting with $\bar{x}_0$.
3. Apply the first control action u_{opt} to the process to minimize cost of the objective function.
4. Move the optimization horizon T one step forward and repeat the same procedure.

Step (4) in the MPC is referred to as *receding horizon control* (RHC), due to the allocation of the prediction time horizon T. See Figure 4.2 to see the strategy of the receding horizon.

Figure 4.1 illustrates the overall MPC controller strategy. As can be seen, the controller takes in initial values for the differential states, which are used to predict the future values of the output variables based on the model. Later, the optimization problem is solved using the collocation method. Finally, the manipulated variables for the first time step are implemented on the model/plant to fulfill the objectives and constraints. Here, the model represents the gas lifted oil network sketched in Figure 3.1.

The principle of *model predictive control* is presented in Figure 4.3. The limitation of the horizon to a finite length N allows to solve the problem numerically. If N is large enough, it will be a good approximation of infinite horizon problem. In Figure 4.3, the optimization is essentially solved for each time step N, where the time steps are determined by the the sampling period h. The latter yields the following: $N = \frac{T}{h}$.

To understand the different outcomes resulted by different values of N, h and T, different sampling periods were tested. Finally, it was decided that, in the case of the gas lifted oil network, the prediction horizon time is $T = 4000s$ with a sampling period $h = 200s$, which yields $N = 20$ steps. This is made by taking into consideration:

1. The time it takes for the system to reach steady state.
2. How often plant measurements are obtained.
3. How often disturbances occur.

Thereafter, the optimization problem is solved at every sampling time, to produce a complete control trajectory, as can be seen in Figure 4.3. However, only the first control action, u_{opt}, is applied to the plant. In the end

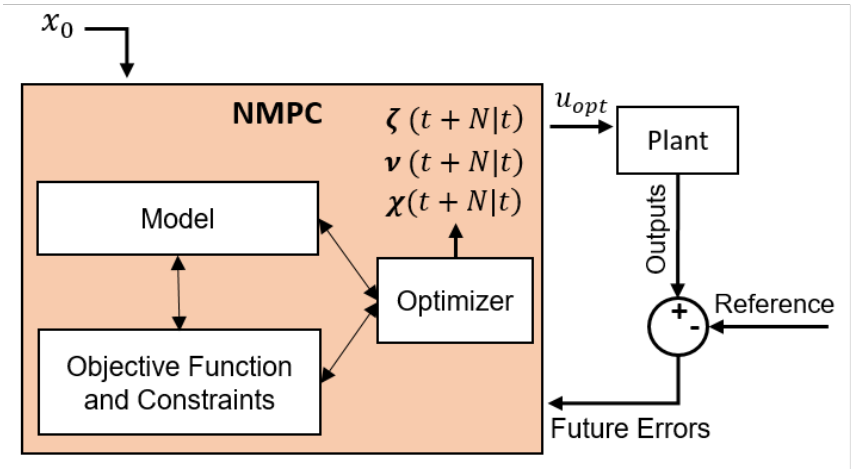


Figure 4.1: The overall strategy of the receding horizon. Displayed the initial differential states, x_0. The MPC controller box (orange) includes the all the operations within the optimization, (1) specifying objective functions and constraints, (2) setting up and formulating an optimizer, (3) optimization and predicting the model, (4) finding control trajectory, u for each time step. Later, only the the first control action is implemented u_{opt} on the plant. The measured values are compared with the reference provided by the RTO, and the future errors are returned.

of the simulation the complete set of the states responses along the control action inputs are registered.

The primary advantage of *model predictive control* over a single step optimization is that MPC couples feedback control with open loop optimization, by requiring a new solution from the open loop optimization based on the future measured errors, see the following algorithm.

Algorithm 2 State feedback MPC procedure

for t = 0,1,2,3,... **do**
  Obtain current state value from measurement data;
  Solve the dynamic OCP on the prediction horizon T from t to $t+N$;
  Apply the first control action u_t on the plant;
end for

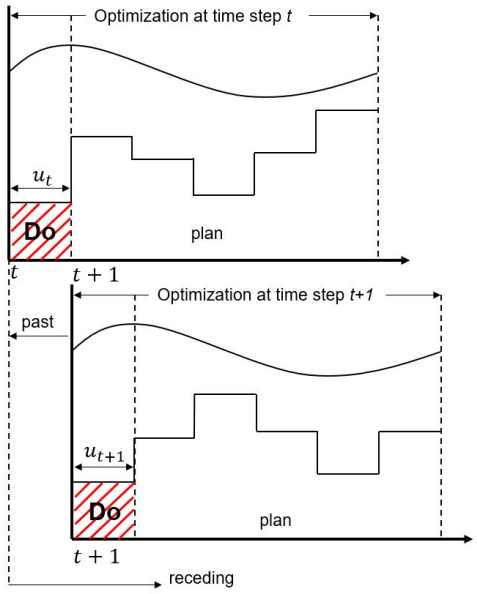


Figure 4.2: Receding horizon strategy depicted for two steps. The process is repeated N times.

4.3 Nonlinear MPC

The gas lifted well network studied in this thesis operates under stiff performance specifications due to large amount of constraints imposed on the system and demanding economical consideration to operate the system close to boundary. Because of the process model nonlinearity, the controller of this system becomes a *Nonlinear Model Predictive Control*, (NMPC). The basic principle shown in Figure 4.3 still applies.

In section 3.7 the gas lifted oil network is modelled according to the following differential algebraic equation (DAE) system:

$$\begin{aligned} \dot{x} &= f(x,y,u,t) \\ 0 &= g(x,y,u,t) \end{aligned} \tag{4.2}$$

The optimal steady state solutions of x, z, and u are supplied by the RTO through Equation 4.1. These states are assumed to be measurable. Thus, from the optimal steady state solutions and the measured states, it is possible to return future errors. The errors are returned to the NMPC through

the closed loop system, and the NMPC attempts to produce a control trajectory that minimizes the errors. Hence, a tracking NMPC. The NMPC optimization problem is formulated as follows:

$$\begin{aligned}\min_{u} \sum_{t=0}^{N-1} & \frac{1}{2}(\chi_{t+1} - x_{t+1}^{ref})^T Q_x (\chi_{t+1} - x_{t+1}^{ref}) \\ & + \frac{1}{2}(\zeta_{t+1} - z_{t+1}^{ref})^T Q_z (\zeta_{t+1} - z_{t+1}^{ref}) \\ & + \frac{1}{2}(\nu_t - u_t^{ref})^T R_t (\nu_t - u_t^{ref}) \\ & + \frac{1}{2}\Delta u_t^T R_{\Delta t} \Delta u_t \end{aligned} \tag{4.3a}$$

subject to

$$\chi_0 - x_0 = 0, \tag{4.3b}$$
$$\dot{\chi}(t) - f(\chi(t),\zeta(t),\nu(t)) = 0, \qquad t \in [t_0,t_f], \tag{4.3c}$$
$$g(\chi(t),\zeta(t),\nu(t)) = 0, \qquad t \in [t_0,t_f], \tag{4.3d}$$
$$\chi^{low} \leq \chi(t) \leq \chi^{high}, \tag{4.3e}$$
$$\zeta^{low} \leq \zeta(t) \leq \zeta^{high}, \tag{4.3f}$$
$$\nu^{low} \leq \nu(t) \leq \nu^{high}, \tag{4.3g}$$
$$-\Delta u^{high} \leq u(t) \leq \Delta u^{high}, \tag{4.3h}$$

where

$$Q_x \succ 0, \tag{4.3i}$$
$$Q_z \succ 0, \tag{4.3j}$$
$$R_t \succ 0, \tag{4.3k}$$
$$R_{\Delta t} \succ 0, \tag{4.3l}$$
$$\Delta u_t = u_t - u_{t-1}. \tag{4.3m}$$

In Equation 4.3, χ, ζ, and ν represent the predicted differential state, algebraic, and control input trajectories at time instance t for $t \in [0,...,N-1]$

In Equation 4.3 the objective function includes the minimization of the difference between system current states and their respective reference points. That is because the system needs to follow a reference trajectory that leads to optimal production, hence the terms $(\chi_{t+1} - x_{t+1}^{ref})$, $(\zeta_{t+1} - z_{t+1}^{ref})$ and $(\nu_t - u_t^{ref})$. The reason for using index $t+1$ for the states χ and ζ is that

the states of interest are χ_1,...,χ_N; ζ_1,...,ζ_N since the initial states x_0 are fixed. Similarly, a quadratic penalty on Δu_t is imposed on the optimization problem. Here, u is a parameter in the optimization problem. In this case, the previous input u_{-1} is required to calculate Δu. The terms in the objective function of the optimization problem above have weighting terms that are used for system tuning. These parameters communicate to the optimization program which variables have higher significance and are essential to reach steady state robustly. Q_x, Q_z, R_t and $R_{\Delta t}$ are always positive semidefinite matrices. Furthermore, the objective function is subject to the nonlinear differential and algebraic constraints shown in Equations 4.3c and 4.3d. The equality constraint is imposed by setting the initial predicted state χ_0 equal to the actual state x_0, which is obtained from measurement data.

Upper and lower bounds are placed on the differential and algebraic states and the control parameters. These include pressure and gas injection flow rate limits in the system's submodels. The control input constrains run from 0 to $N-1$ since u_{N-1} defines the control input on the time horizon $t \in [N-1,N]$. The change in the control input is restricted by 4.3h. This is usually an issue in optimization since valves do have limitations on their dynamic performance, e.g. the step and speed with which a control input may change.

Notice that economic measures like revenue or cost are often linear in one or several states, and can be added to the objective function in Equation 4.3 as terms for minimization $d_{xt}x_t$, $d_{zt}z_t$ and $d_{ut}u_t$, where d_{xt}, d_{zt} and d_{ut} include sales prices and the costs of input factors [19]. In this thesis they are not included in the objective function. Thus, the NMPC merely tracks optimal points of production provided by the RTO.

Equation 4.3 does not include any feedback control. Hence, it represents an open loop nonlinear DAE optimization problem. The loop can be closed by applying the the following algorithm:

Algorithm 3 NMPC procedure with state feedback

for $t = 0,1,2,3,...,N-1$ **do**
  Obtain current state of the model x_t.
  Solve the dynamic nonlinear OCP 4.3 on the prediction horizon T from t to $t+N$.
  Apply the first control action u_t on the plant.
end for

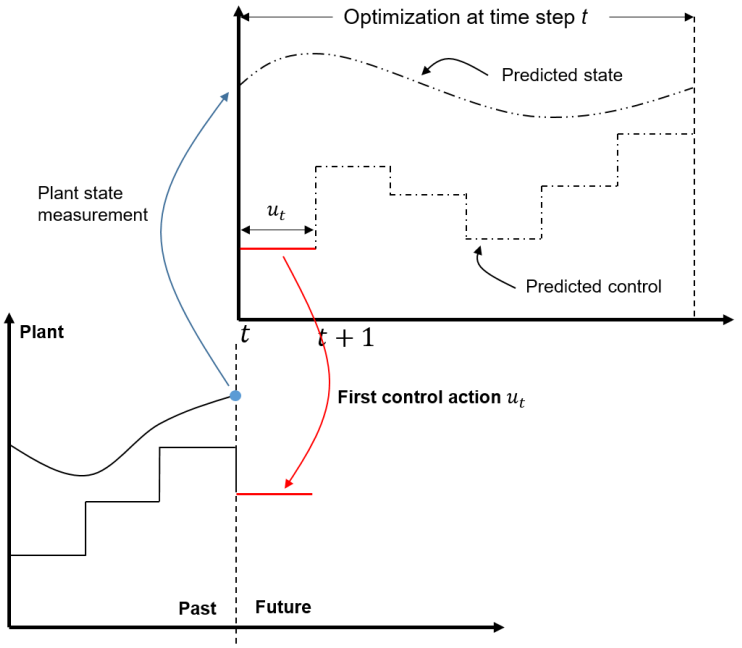


Figure 4.3: The principle of MPC. The top axis displays the open loop optimization problem, calculated for one sampling period of the model. After which only the first control action is applied to the model, and the states and inputs are registered on the bottom axis.

The algorithm above creates a NMPC system using the optimization problem developed earlier. The key addition is obtaining the current state of the system from the model plant and feeding it back to the optimization problem. Thereafter, determine whether the system is approaching the optimal steady state solutions obtained in Eq.4.1. An illustration of the algorithm is displayed in Figure 4.3. The red arrow shows that once the current states are obtained and the optimization problem is solved over T, the first optimal control action can be applied.

4.4 Discretization and transcription of OCPs

In section 2.4 the basics and background of direct collocation were presented. The discretization method applied to the gas lifted oil network model, is the direct orthogonal collocation. This section shows the discretization method of a DAE system and the transcription of the OCP [13].

The OCP in both the control and states are discretized on the collocation interval $t \in [t_k,t_{k+1}] \subseteq [t_0,t_f]$, where $k \in [0,...,N-1]$, using a K^{th}-order polynomial $p(t,v_k)$. The polynomial p depends on the coefficients x_k. The approximated polynomial is found by building Lagrange polynomials for the set of collocation times $t_{k,0},...,t_{k,d}$ which are constructed as follows:

$$P = \prod_{j=0,j\neq i}^{K} \frac{t - t_{k,j}}{t_{k,i} - t_{k,j}} \tag{4.4}$$

By combining the coefficients v and the Lagrange polynomials $l(t)$ the following is obtained:

$$x_k(t,x_k) = \sum_{i=0}^{K} x_{k,i} P_{k,i}(t) \tag{4.5}$$

Further the property of the Lagrange polynomial is as follows:

$$P_{k,i}(t_{k,j}) = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{4.6}$$

The latter property entails that for $i = 0,...,K$ the polynomial p will take a unitary value at the collocation time $t_{k,i}$ and a zero at all other times $t_{k,j\neq i}$.

Equation 4.5 is used for the simple ODE $\dot{x}(t) = f(x(t),t)$. However, the gas lifted well system is a DAE system. Thus, in order to account for the algebraic variables an addition polynomial to Equation 4.5 is added.

$$x_k(t,x_k) = \sum_{i=0}^{K} x_{k,i} P_{k,i}(t) \tag{4.7a}$$

$$z_k(t,z_k) = \sum_{i=1}^{K} z_{k,i} P_{k,i}(t) \tag{4.7b}$$

Using the polynomials in Equation 4.7, the integration over the time interval $[t_k,t_{k+1}]$ can be performed by selecting collocation variables $x_{k,i}$ and $z_{k,i}$ this is done by solving algebraic equations that ensure that the polynomial in 4.7a is a precise rendering of the states trajectories. By enforcing several

conditions, and assuming that the initial state value is available, the integration of the system over a time interval $[t_k,t_{k+1}]$ can be found via solving collocation equations.

Continuity: The initial condition must be enforced, that is, $x_k(t_k,x_k) = x_k$. Here it is assumed that the initial state x_k at the beginning of the interval $time = t_k = t_{k,0}$ is measured. Consequently, $x_k(t_k,x_k) = x_k = x_{k,0}$. Satisfying the initial condition requires simply $0 = x_{k,0} - x_k$ to hold. Notice that 4.7a has $(K+1)$ degrees of freedom (DOF), while 4.7b requires only K DOF. The reason for the additional DOF in the differential polynomial 4.7a is to ensure the continuity of the states by forcing all shooting gaps to close. This latter condition is unnecessary for the algebraic states which can have discontinuous trajectory. Moreover, the continuity boundary condition is also applied on the end of the interval, i.e. the beginning of the next interval, $0 = x_{k,K} - x_{k+1,0}$.

Dynamics: The dynamics on the remaining collocation times $t_{k,1},...,t_{k,K}$ must satisfy the dynamics of the polynomial $x_k(t,x_k)$. Hence, the following equation is imposed:

$$\frac{\delta}{\delta t}x_k(t_{k,i},x_k) = \mathbf{F}(\underbrace{x_k(t_{k,i},x_k)}_{=x_{k,i}},z_{k,i},u_k) \tag{4.8}$$

In the latter equation, $\frac{\delta}{\delta t}x_k(t_{k,i},x_k)$ denotes the time derivative of the Lagrange polynomials, and $\mathbf{F}$ denotes the model differential states as described in section 3.7.

Algebraic: The dynamics of the algebraic states on all collocation times $t_{k,1},...,t_{k,K}$ must not be subject to change. Thus they are equal to zero:

$$0 = \mathbf{G}(\underbrace{x_k(t_{k,i},x_k)}_{=x_{k,i}},z_{k,i},u_k) \tag{4.9}$$

Where $\mathbf{G}$ denotes the model's algebraic states derivatives $\frac{\delta}{\delta t}$, as described in section 3.7.

Control inputs: The control inputs are assumed to be piecewise constant in the interval $[t_k,t_{k+1}]$, which is given by:

$$u(t) = u_k \tag{4.10}$$

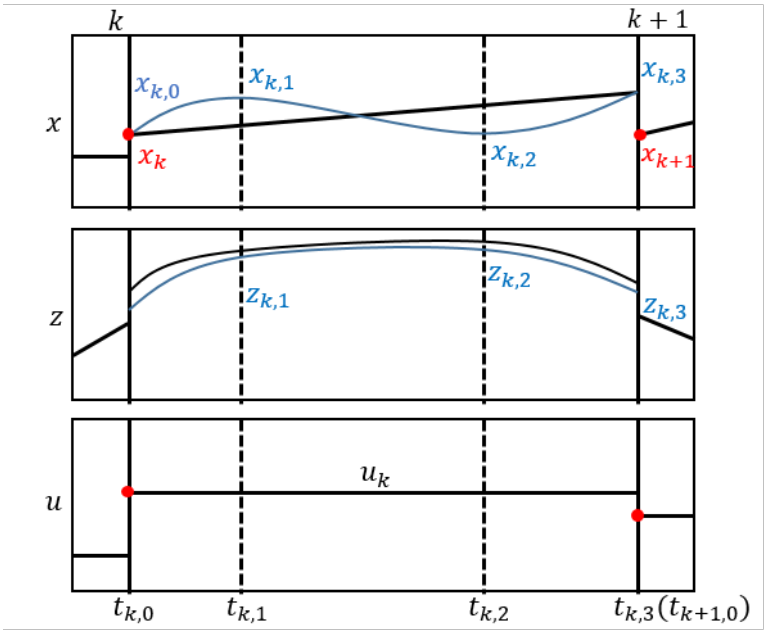


Figure 4.4: Third order direct collocation in the interval $[t_k,t_{k+1}]$. From the top, the differential states with $(K+1)$ DOF, the algebraic states with (K) DOF, and the control input.

In the gas lifted oil network, it is sufficient to use the third order polynomials $K = 3$. Moreover, by solving the collocation equations above, the differential and algebraic states can be well approximated. Figure 4.4 shows a simple and clear depiction of how the method works and how it is solved, with the blue curves indicating the solution from the collocation equations. Notice that, at the end of the interval, the continuity boundary condition also applies $0 = x_{k,K} - x_{k+1,0}$.

4.5 Nonlinear programming problem

After the discretization process, the OCP can be transcribed into a finite dimensional nonlinear programming problem (NLP). The NLP is using a software library for large scale nonlinear optimization of continuous systems. By using the third order polynomials in the discretization, the NLP problem can be formulated as in Equation 4.11, in which all the equations are discretized and optimized simultaneously in the collocation loop, using

the collocation points for the states and controls.

$$\begin{aligned}\min_{u_k} \sum_{k=0}^{N-1} & \frac{1}{2}(\chi_{k+1} - x_{k+1}^{ref})^T Q_x (\chi_{k+1} - x_{k+1}^{ref}) \\ & + \frac{1}{2}(\zeta_{k+1} - z_{k+1}^{ref})^T Q_z (\zeta_{k+1} - z_{k+1}^{ref}) \\ & + \frac{1}{2}(\nu_k - u_k^{ref})^T R (\nu_k - u_k^{ref}) \\ & + \frac{1}{2}\Delta u_k^T R_\Delta \Delta u_k \end{aligned} \tag{4.11a}$$

subject to

$$\chi_{k+1} = f(\chi_k, \zeta_k, \nu_k) + w_k \tag{4.11b}$$
$$0 = \chi_{k,0} - x_k \tag{4.11c}$$
$$0 = \dot{\chi}_k(t_{k,i}, \chi_k) - \mathbf{F}(\chi_k(t_{k,i}, \chi_k), \zeta_{k,i}, \nu_k), \tag{4.11d}$$
$$0 = \mathbf{G}(\chi_k(t_{k,i}, \chi_k), \zeta_{k,i}, \nu_k), \tag{4.11e}$$
$$0 = \chi_{k,K} - \chi_{k+1,0} \tag{4.11f}$$
$$\chi^{low} \leq \chi_{k,i} \leq \chi^{high}, \tag{4.11g}$$
$$\zeta^{low} \leq \zeta_{k,i} \leq \zeta^{high}, \tag{4.11h}$$
$$\nu^{low} \leq \nu_k \leq \nu^{high}, \tag{4.11i}$$
$$-\Delta u^{high} \leq \Delta u_k \leq \Delta u^{high}, \tag{4.11j}$$

where

$$Q_x \succ 0, \tag{4.11k}$$
$$Q_z \succ 0, \tag{4.11l}$$
$$R \succ 0, \tag{4.11m}$$
$$R_\Delta \succ 0, \tag{4.11n}$$
$$\Delta u_k = u_k - u_{k-1}, \tag{4.11o}$$

and

$$\dot{\chi}_k(t_{k,i}, \chi_k) = \frac{1}{t_{k+1} - t_k} \sum_{i=0}^{K} \chi_{k,i} \beta_{k,i}. \tag{4.11p}$$

Here, $\chi_{k+1,i} \in \mathbb{R}^{n_\chi}$, $\zeta_{k+1,i} \in \mathbb{R}^{n_\zeta}$, and $\nu_k \in \mathbb{R}^{n_\nu}$ represent the predicted differential state, algebraic, and control input at time instance k for $k \in [0,...,N-1]$ and collocation point i for $i \in [1,...,K]$. Notice that the sum of the objective function in 4.11a is taken from $i = 1,...,K$. That is justified

by the presence of algebraic variables, and the initial condition where $i = 0$ in the constraints. Furthermore, the constraints of the NLP are based on the conditions of the collocation method, namely, the *continuity*, *dynamics* and *algebraic* conditions. Finally, the time derivative of the Lagrange polynomial is defined as in 4.11p, where $\beta_{k,i}$ are the weights produced after differentiation (see [18] and [1] for detailed differentiation), and can be plugged in for the state dynamics constraints.

The NLP developed here uses the collocation points $t_{k,i}$ developed for polynomials of the third degree by *Gauss-Radau*. That is because the *Radau Collocation* chooses a set of collocation points that includes the end point of the interval. This is an advantageous property for stiff systems [15].

The NMPC formalized in Equation 4.11 can be solved in a receding horizon fashion, where at each time k, the measurement x_k is assigned as the initial state for the optimization problem. Then, the optimization problem is solved to compute the optimal states and input trajectory over the prediction horizon. Later only the first step of the optimal control sequence is implemented on the plant, $u_k = \nu_1$. Finally, at the time $k + 1$, new measurements of the state x_{k+1} are obtained and the optimization procedure is repeated, hence enabling feedback control [19].

4.6 The NMPC system and software package

The entire optimization and control system built in this thesis is described in Figure 4.5. The figure shows the complete application of the mathematical modeling of the network (Plant Model), developed in section 3. Moreover, the real time optimization (RTO) program, Equation 4.1, is developed in section 4.1. Finally, the nonlinear model predictive control (NMPC) program, Equation 4.11, is formulated in section 4.3.

1. The RTO provides the system with the optimal steady state values of x, z and u.

2. The NMPC takes in the measured states as the initial points and attempts to minimize the objective function in Equation 4.11. Thereafter, it applies the first step of the optimal control sequence $\nu_1 = u_{opt}$ on the plant.

3. The plant runs and returns measured values for the states to the NMPC.

Usually, the NMPC is used to provide set-points to a lower regulatory layer such as PI controllers (see Figure 2.10) from which the controllers manipulate the valves to keep the plant at optimal steady state operations. However, in this thesis the regulatory layer is not considered. Rather, the NMPC is used directly to manipulate the valves in the gas lifted oil network.

This thesis uses the optimization software package CasADi (Computer algebra system for Automatic Differentiation) to solve the OCP problems developed, namely, the RTO in 4.1 and NMPC in 4.3 [2]. The software uses simultaneous approach to solve the OCPs as described in Figure 2.6. CasADi is a symbolic framework for algorithmic (a.k.a. automatic) differentiation and numeric optimization, and allows users to construct scalar (SX) or sparse (MX) symbolic expressions. These expressions efficiently use the method of algorithmic differentiation in forward or reverse modes and graph coloring techniques for generating complete, large and sparse Jacobians and Hessians. CasADi makes it very simple to calculate the relevant derivatives in the DAE system, which drastically reduces the efforts for users to find derivatives (see [2] for a full manual guide and description of CasADi).

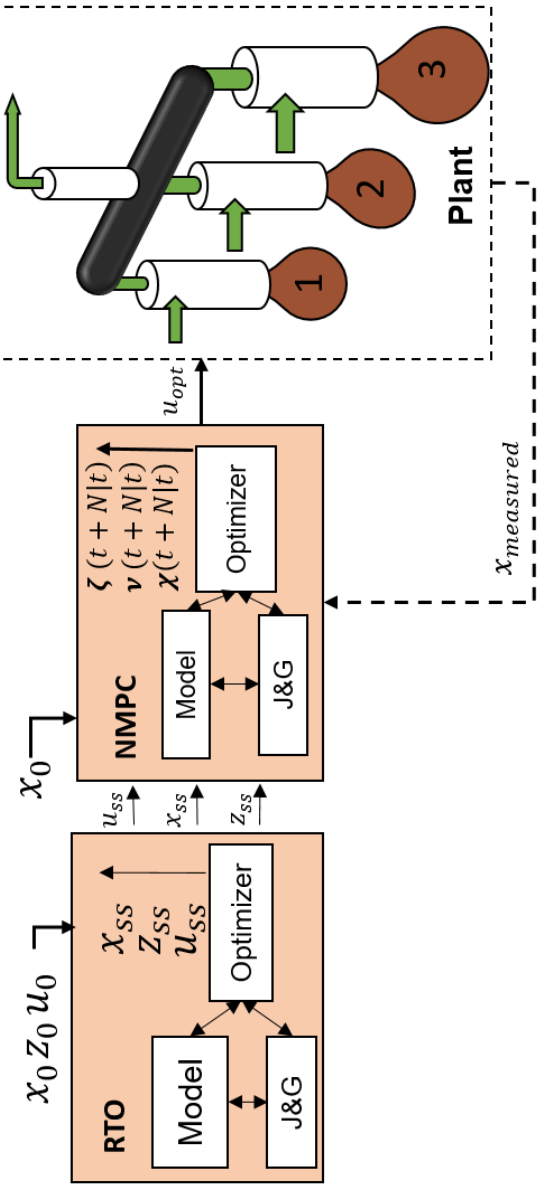


Figure 4.5: A schematic representation of the optimal control strategy and design of the gas lifted oil production network. Here, $\chi \in \mathbb{R}^{n_\chi}$, $\zeta \in \mathbb{R}^{n_\zeta}$, and $\nu \in \mathbb{R}^{n_\nu}$ represent the predicted differential state, algebraic, and control input respectively. In addition, the first control action $u_{opt} = \nu_{1,t}$ is displayed.

Chapter 5

Numerical Case Examples

This chapter investigates the following:

1. The open loop step response of the system.
2. The relationship curve between oil production and gas injection.
3. NMPC with low and high magnitude noise.
4. RTO and NMPC robustness with regard to parameter changes.

In this thesis, the plant simulations are done in Matlab, where the system of the DAEs is integrated using the function `ode15s`. For the open loop step response simulations the `ode15s` is initialized using the steady state solutions from the `fsolve` function. Further, the steady state OP in Equation 5.1 is solved using CasADi and the IPOPT plugin. Finally, the dynamic OP is transformed into a discrete NLP problem, Equation 4.11, where the model in the predictive state is integrated using the orthogonal collocation method with 3^{rd} degree radau collocation points and the NLP is solved using the IPOPT Plugin.

Initially the RTO runs to find the optimal steady state solutions that are supplied as tracking points to the NMPC. The NMPC runs to find optimal control trajectory for the gas injection, of which only the first control action is applied to the plant. Further, the plant (`ode15s`) runs to compute the states, these are sent to the NMPC as the starting points of the following iteration. In CasADi the IPOPT displayed at all times `"Optimal solution found"`.

5.1 Open loop system simulation

This section investigates the modified mathematical model introduced in Chapter 3. Initially, an open-loop response simulation is performed when the wells have different reservoir pressures to check model stability. Later, a step response simulation is done when wells have varying $GORs$ and WCs to investigate their effects. Towards the end, the relationship curve between total oil production and gas injection is displayed.

Open loop step response

The objective here is to check the stability of the modified model. The production chokes, $u_{1,c}$ for $c \in [well1, well2, well3, riser]$, of the gas lifted well network are kept fully open, given that there is no constraint on the total production of oil. Hence, the steps are applied only on the control inputs of the gas injection valves, $u_{2,n}$ in Equation 3.13, in which the subscript $n \in$ [1,2,3] denotes the well number. The steady state solutions for the states in Equations 3.58 are found using the function `Fsolve` when all injection valves are half way opened, $u_{2,n} = [50\%]$. The parameters used in the simulation are shown in Table 5.1. This choice is made to investigate the step response with varying reservoir pressures.

Table 5.1: Parameters used in open loop step response on gas injection valves for wells with varying reservoir pressure

Symb.	Description	Well 1	Well 2	Well 3	units
P_{res}	reservoir pressure	140	150	160	bar
K_{pr}	production choke constant	2.80	2.80	2.80	-
GOR	mass gas oil ratio	0.15	0.15	0.15	-
PI	productivity index	3.00	3.00	3.00	$kg/s/Pa$
WC	water cut	0.15	0.15	0.15	-

Figure 5.1 shows the open loop step response when the gas injection valves positions are adjusted to $u_{2,n} = [70\%]$. As can be seen, the system stably adjusts to new operations and reaches steady state after 2000s. This indicates that the modified model is stable. In addition, with increasing reservoir pressure the total production of liquid increases from the corresponding well. The larger the pressure differences in the wells, the larger is the driving force of the flow. Hence, more mass will escape to areas subject to low pressure.

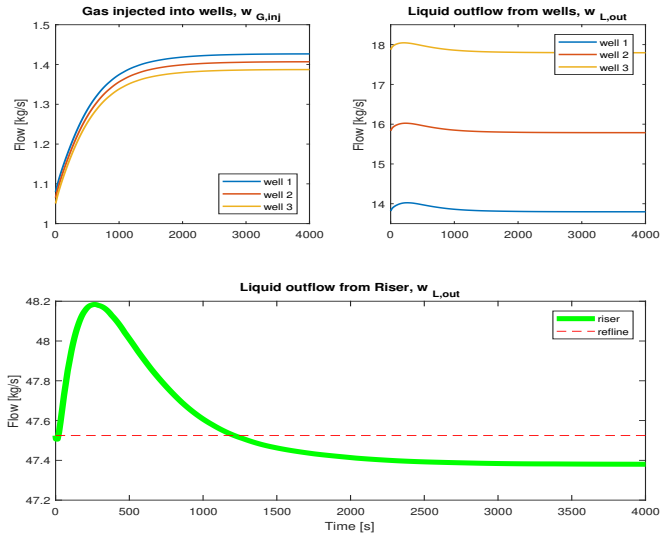


Figure 5.1: On the top left, the gas injection step increase is displayed. On the top right, the liquid outflows are displayed. On the bottom, total liquid production from the riser with $WC = 15\%$ is displayed, as well as the reference line (red) before the step response.

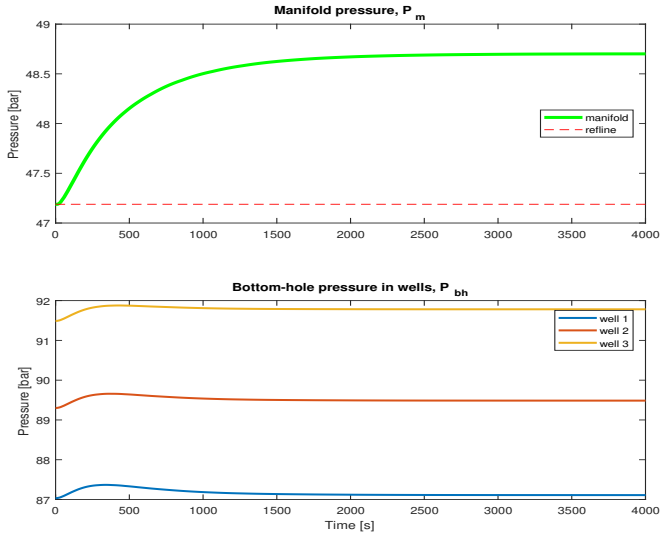


Figure 5.2: Gas injection step increase $u_{2,n} = 70\%$. On the top, the manifold pressure increase with reference line (red). On the bottom, the bottom hole pressures in the wells.

For higher gas injection rates, the overall oil production decreases and reaches a new steady state point below the previous, as indicated by the red dashed line in Figure 5.1. The system's production rates are affected by several factors in the gas lifted well network. The observation in Figure 5.1 can be justified by the trade off that is done between the pressures loss due to friction, Equation 3.19, and the hydrostatic pressure in Equation 3.17. To illustrate, the gas is injected into the wells in order to decrease the density of the fluids in the tubing, thus decreasing the bottom hole pressure. When the bottom hole pressure is decreased more oil escapes from the reservoirs as the pressure difference increases. However, this is beneficial only to some extent, given that the hydrostatic pressure drop cannot compensate the increased friction loss which is originating from increased gas flow inside the tubing.

Figure 5.2 shows the manifold and bottom hole pressures behavior when the gas injection is increased. Although this leads to decrease in the bottom hole pressure however this also leads to decline in the pressure difference between top and bottom of the tubing. As indicated, the pressure at the manifold increases when gas injection is increased. Thus, less oil is driven out of the tubing and through the riser. Hence, the high coupling of the gas lifted well network.

Effects of reservoir parameters on the system

This section briefly shows the GOR and WC effects on the total liquid and oil production of the system. These quantities are based on the nature of the reservoir and cannot be controlled or manipulated. Therefore, during the life time of the reservoir these parameters are considered as disturbances. The effects are studied by carrying an open loop step response on the gas injection for wells with varying $GORs$ and WCs. See Table 5.1.

(i) *Water cut:* The water cut was introduced to the model in Chapter 3.4. Figure 5.3 shows a step response done on the gas injection valves into the wells with varying WCs. In addition Figure 5.3 shows the oil production rates from the corresponding well. As can be seen, wells with lower WCs produce more oil, which is expected, since WC is a fraction quantity. On the other hand, the total production of liquid flow is slightly decreased from the nominal value (red dashed line). This is justified by the density and viscosity changes that have direct effects on the pressures in the tubing. The larger the WC in the well, the higher the density of the mixture

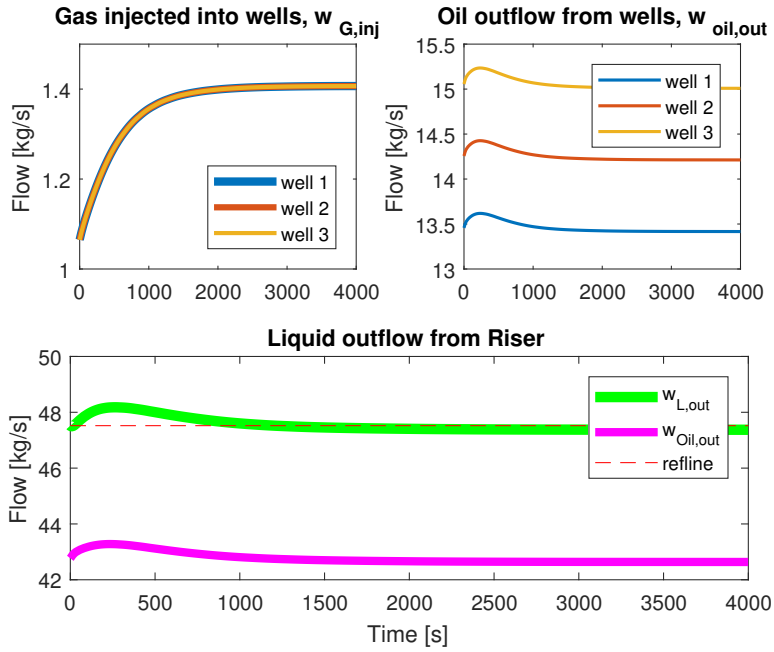


Figure 5.3: Step response simulation when $u_{2,n}$ is increased to 70% for wells with varying water cuts, $WC = 0.15, 0.10, 0.05$, for wells 1, 2 and 3 respectively. Displayed on the top left, the gas injection into wells. On the right, oil outflows from the wells. On the bottom, liquid and oil outflows from the riser.

becomes, which leads to higher bottom hole pressures and less reservoir outflow. However, Figure 5.3 indicates that the WC has insignificant effect on the total production of liquid mixture, because there is no significant increase in the density.

(*ii*) *Gas to oil ratio:* The GOR is an important parameter in dimensionless metric units, which affects the productivity and efficiency of the oil production from reservoirs. Figure 5.4 depicts liquid outflows from wells with different GOR values. On the top left of Figure 5.4, although the same gas injection step is applied to all wells, the gas entering the tubing through the gas injection point varies due to fluid densities. The higher the GOR is, the lower the density at the bottom of the tubing. On the top right, liquid production is lowest for the well with the lowest GOR, well 3. Further, for well 1 the liquid production is the highest, with slightly higher production than well 2. This shows that increasing GOR values is beneficial only to some extent. The gas lift injects gas to minimize the density of the liquid at the

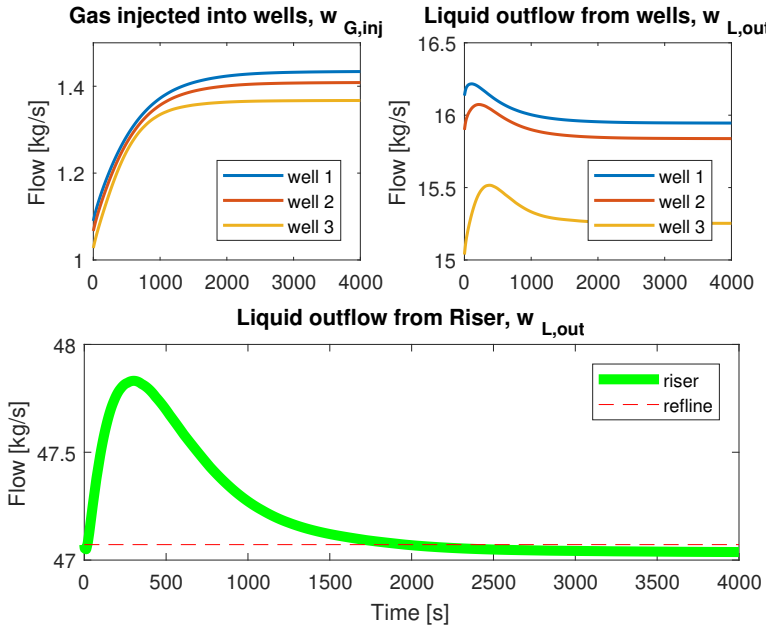


Figure 5.4: Step response simulation when $u_{2,n}$ is increased to 70% for wells with varying gas to oil ratios, $GOR = 0.2, 0.15, 0.1$, for wells 1, 2 and 3 respectively. Displayed on the top left, the gas injection into wells. On the right, liquid outflows from the wells. On the bottom, liquid outflow from the riser.

bottom of the tubing. The same phenomenon also applies to increasing gas ratios. However, the overall production from the riser is less than the nominal production point. This is caused by the trade off between hydrostatic pressure loss and pressure loss due to friction. Notice, the gas lift network in this system is highly coupled and nonlinear. This can be seen for very low values of GOR, in which the `ode15s` will display abnormal simulations and errors.

Effect of gas injection on production rates

This section attempts to justify the need for optimization in the gas lifted oil network depicted in Figure 3.1. As shown earlier, different parameters and operating conditions affect the gas lift system diversely. Thus to investigate the possible benefits of optimal control, a simulation study was performed using the parameter setup in Table 5.1.

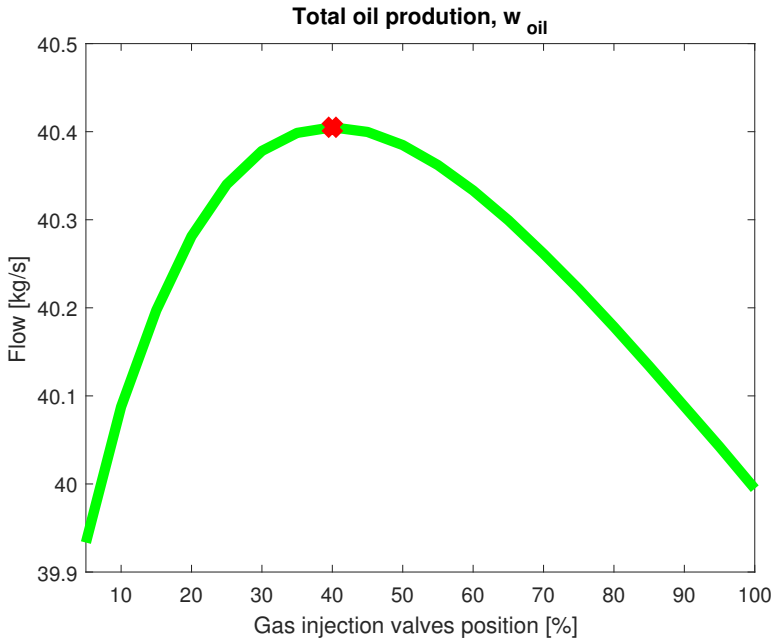


Figure 5.5: Total oil production rate $w_{oil,out}$ from the riser subject to gas injections in the three wells, where $u_{2,n}$ is the gas injection opening position. The red cross shows the point at which maximum oil production is achieved.

The relationship between gas injection and oil production is displayed in Figure 5.5. Figure 5.5 shows that the production rate of oil increases rapidly to a certain point before it starts dropping again. This implies that the injection rate of gas can be both advantageous and disadvantageous at different rates or valve opening positions. This curve is justified by the the hydrostatic pressure drop that cannot compensate the increased friction loss which is due to the increment of gas mass in the tubing parts of the wells.

Furthermore, Figure 5.6 shows the pressure difference between the bottom hole pressure and the reservoir pressure, $\Delta P = P_{res} - P_{bh}$. As indicated, the pressure difference curve has a similar behavior as in the oil production curve in Figure 5.5. This indicates that the main driver of oil production is the bottom hole pressure P_{bh}. That is because the lower the bottom hole pressure, the more oil will escape to the gas lift in pursuit of relieving conditions. The red cross in Figure 5.6 depicts the point at which the bottom hole pressure is lowest and the pressure difference is highest. This relationship is also justified by the modeling Equation 3.27, where a decrease

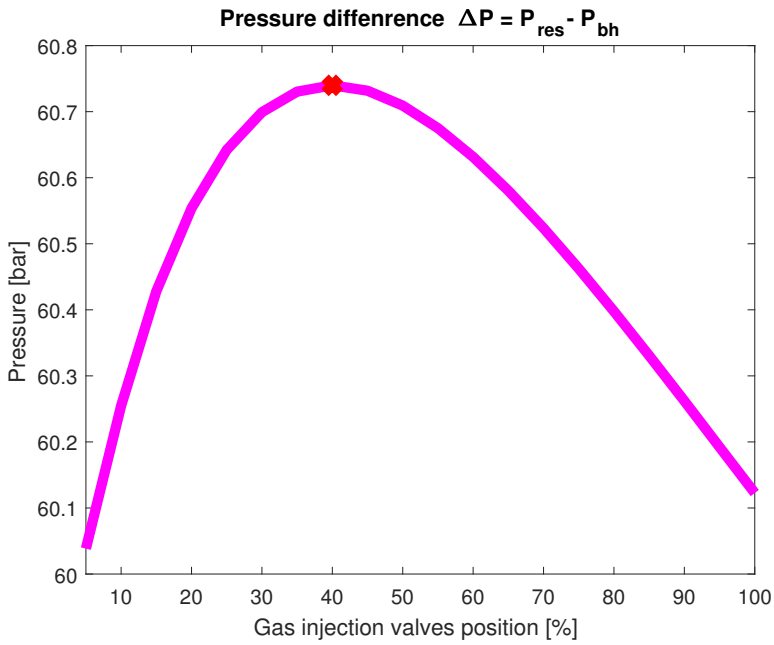


Figure 5.6: Reservoir pressure, P_{res}, and bottom hole pressure, P_{bh}, difference subject to gas injection valve opening position, $u_{2,n}$. The red cross shows the point at which ΔP is maximized.

in the bottom hole pressure is beneficial since the term under the square root increases with decreasing P_{bh} thus the reservoir production rate w_{res} increases.

5.2 Steady state optimization

The RTO is used to find the optimal steady state values of $\mathbf{x}_{opt,SS}$, $\mathbf{z}_{opt,SS}$, and $\mathbf{u}_{opt,SS}$, for differential states, algebraic, and gas injection control inputs, respectively. At all times the production chokes are left fully open assuming that there are no constraints on the total production. Hence, more liquid leads to more oil. This is done to achieve optimal production conditions while operating within bounds and constraints. The optimal steady state values are calculated less often to account for the occurrence of disturbances in the plant model, eg. WC and GOR increase in the wells. These optimal values are found by solving the optimization problem given in Equation 5.1. The objective function in the optimal control problem below

can be set based on the user's desires. In Equation 5.1, different economic objectives can be defined. In this thesis, the maximization of liquid mass rate outflow from the riser and minimization of gas injection into wells are considered as the objectives of the RTO. By making the choice of maximizing liquid, this directly leads to maximization of oil at all times regardless of the variation of the uncontrolled water cut parameter, given that there are no constraints on the production and that the production chokes are left fully open. In addition, the choice for the minimization of gas injection is merely to minimize costs of injection, assuming insufficient gas resources. However, under different circumstances the objective of the RTO can be modified to fit suitable conditions.

After setting the problem in the CasADi software, the IPOPT finds the optimal steady state solutions. These solutions are moved forward to the NMPC, see Figure 3.1 for set up and Appendix B for decision variables. The same procedure is done several times within the limits of the prediction horizon to investigate how the controller functions when disturbances occur.

$$\min_{u} \; -w_{L,out,r} + \sum_{well=1}^{3} w_{G,in,well} \tag{5.1a}$$

subject to

$$f(x,z,u) = 0, \tag{5.1b}$$
$$g(x,z,u) = 0, \tag{5.1c}$$
$$x^{low} \leq x \leq x^{high}, \tag{5.1d}$$
$$z^{low} \leq z \leq z^{high}, \tag{5.1e}$$
$$u^{low} \leq u \leq u^{high}. \tag{5.1f}$$

When plugging $\mathbf{u}_{opt,SS}$, $\mathbf{x}_{opt,SS}$, and $\mathbf{z}_{opt,SS}$ in the plant, the `ode15s` displays that the system is at steady state. Hence, the RTO solutions are valid. This simulation uses the parameters in 5.2 and the bounds in Appendix C.4. Notice that for varying combinations of parameters values it is difficult to find the optimal control solutions manually as in Figures 5.5 and 5.6. The latter is due to high nonlinearity and high coupling of the gas lifted oil network which leads to errors displayed by the `ode15s`. Thus, employing the RTO method is very beneficial, but, a control trajectory is required to control the system.

Table 5.2: Parameter values used in NMPC simulations

Symb.	Description	Well 1	Well 2	Well 3	units
P_{res}	reservoir pressure	140	150	160	bar
K_{pr}	production choke constant	2.80	3.00	3.20	-
GOR	mass gas oil ratio	0.15	0.10	0.15	-
PI	productivity index	3.00	2.50	2.00	$kg/s/Pa$
WC	water cut	0.15	0.10	0.05	-

5.3 NMPC case studies

The objective of the NMPC is to track the optimal steady state solutions given by the RTO. The discrete NLP in Equation 4.11 is solved using the IPOPT. In Equation 4.11 the outputs have to converge to the given reference points as much as possible while fulfilling the constraints. The values considered for tracking are as shown in the objectives of Equation 4.11a. Namely, the differential states, masses of gas and liquid, and algebraic variables, rates of liquid and oil and gas injection, in the gas lifted oil network. This section discusses different cases such as noisy measurements and setpoint tracking of the NMPC.

Case 1: NMPC with measurement noise

In real processes the measurements are not always accurate, thus the controller has to handle noisy measurements. It is important that the controller is not sensitive to the noise, in order to avoid exposing the valves to continuous wear and tear. In this case study, a random low and high noise is added to the differential states given in Equations 3.58 and 3.59. The noise is generated using the `randn` function in MATLAB. The function selects a random scalar from a standard normal distribution which is then multiplied by the magnitude, eg. $noise(x_1) = rand(1) * mag$, where $mag = 10^{-2},\ 10^{-3}, 10^{-4}$. The OCP is given in Equation 4.11 where $x_{ref}, z_{ref}, u_{ref}$ are the steady state optimal solutions supplied from by RTO. The RTO objectives are given in Equation 5.1 for maximization of liquid outflow and minimization of gas injections shown in 5.1a. Further, in Equation 4.11, when the measurement is corrupted by noise, the closed loop dynamics become $\chi_{k+1} = f(\chi_k, \zeta_k) + w_k$, here $w_k \in \mathbb{R}^{n_x}$.

Figure 5.7 shows the simulations done for low and high magnitude mea-

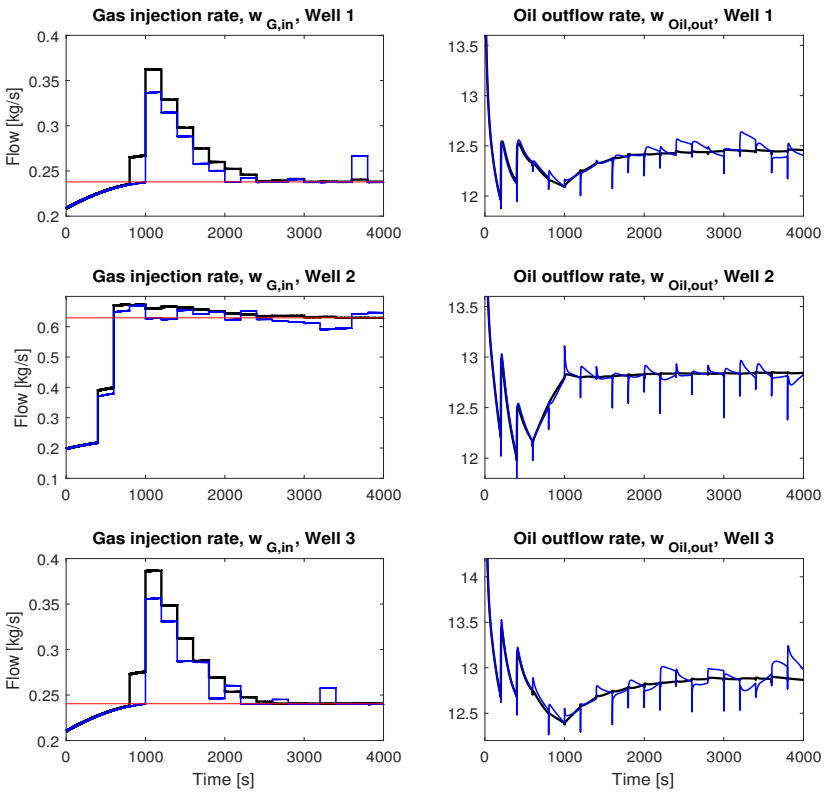


Figure 5.7: On the left, the indirectly manipulated variables for low magnitude noise (black), high magnitude noise (blue) and optimal set point from RTO (red). On the right, the oil outflows for low magnitude noise (black) and high magnitude noise (blue).

surements noise of the differential states. The simulation is done using the parameters given in Table 5.2. The low magnitude noise added to the states is of the order 10^{-4} on the gas mass in the annulus, and 10^{-3} on the gas and liquid masses in the tubing parts of the network. On the other hand, The high magnitude noise added to the states is of the order 10^{-3} on the gas mass in the annulus, and of 10^{-2} on the gas and liquid masses in the tubing of the network.

Figure 5.7 depicts the system's gas injection on the left, and well oil production on the right. The simulations indicate that the controller is more sensitive to highly noisy measurements, thereby causing the system's valves

to be noisy, which in turn leads to increased oscillations and less stability around the optimal point. For less noisy measurements (black lines), the NMPC reaches the optimal steady state after 4000s. Hence, the objective function 4.11a is satisfied. The reason why the noise measurements have wavy curves and spikes is the sampling time. In this thesis, the sampling time in the NMPC is 200s with a prediction horizon of 4000s. The choice is made based on engineering intuitions and trials so that all optimal solutions are satisfied. Further, the `ode15s` receives the first control action and runs the plant for 200s before returning a new measurement to the NMPC.

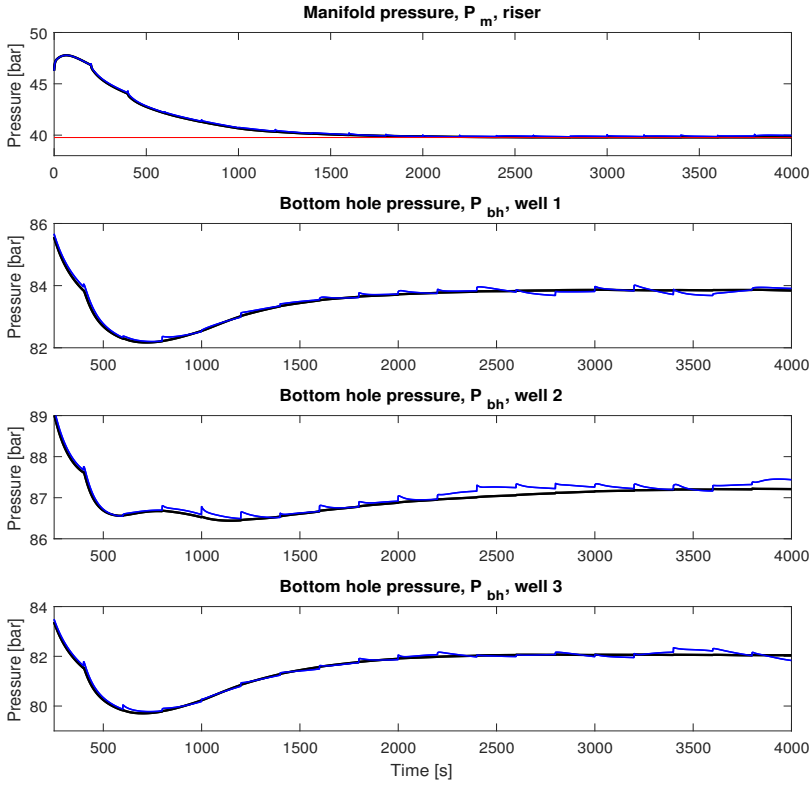


Figure 5.8: From the top: Manifold pressure with its reference point and bottom hole pressures for low magnitude noise (black) and high magnitude noise (blue).

Similar results are obtained In Figure 5.8, where the manifold pressure and the bottom hole pressures are depicted. As can be seen, the manifold

pressure does not show significant noise oscillations. Rather, it reaches its optimal steady state value after 2000s. On the contrary, the bottom hole pressures experience higher instabilities when subject to high measurement noise. Hence, for highly noisy measurements the oil production becomes noisy and less stable. This is shown in Figure 5.9, where the liquid, oil, water, and gas outflows from the riser are depicted for both low and high magnitudes of noise. As can be seen the liquid mass outflow out of the riser oscillates around the optimal steady states point (red reference line), while for low magnitudes the optimal point is fulfilled steadily.

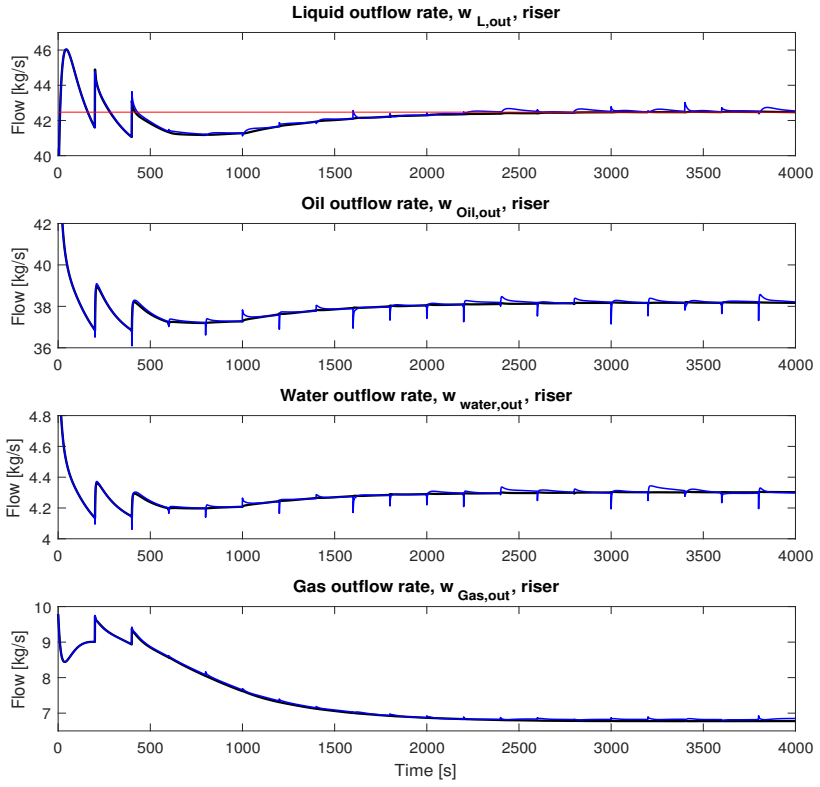


Figure 5.9: From the top: liquid, oil, water and gas outflows from the riser for low magnitude noise (black) and high magnitude noise (blue). The red reference line is for the optimal liquid mass rate.

Demonstrating high sensitivity to noise is a normal effect of the NMPC. However, due to systems nonlinearity and coupling it is extra important

to address noise issues. One possible solution is to enhance the NMPC tuning. The system can be tuned by taking into consideration the weighting parameters in the objective function in Equation 4.11. For example, the R_t and $R_{\Delta t}$ can be adjusted to counter high noise effects.

Case 2: NMPC robustness with regard to parameter changes

Typically, during the life time of the reservoir the gas to oil ratio (GOR) and water cut (WC) are subject to change. These changes can have significant effects on the reservoir oil production. As discussed in chapter 3, the oil production from the gas lifted oil network is subject to disturbances by these uncontrollable natural parameters. This section investigates the RTO responses and NMPC controller tracking ability when disturbances occur on the GOR and WC in each of the reservoirs. The latter is done by running the RTO, Equation 5.1, several times to account for these disturbances. The RTO will produce new optimal steady state solutions, which are fed to the NMPC as reference points. Further, The NMPC problem 4.11 is solved to produce a suitable control trajectory to be applied on the plant. The plant runs and returns the initial values of states for the next iteration in the NMPC. The same procedure is repeated until the end of the simulation.

In this case, the differential, algebraic and control inputs are considered for tracking to maximize oil production while minimizing gas injection, according to Equation 4.11. The parameters used in this case are displayed in Table 5.3.

Table 5.3: Parameters values used in NMPC disturbance simulations

Symb.	Description	Well 1	Well 2	Well 3	units
P_{res}	reservoir pressure	140	150	160	bar
K_{pr}	production choke constant	2.80	3.00	3.20	-
GOR	mass gas oil ratio	0.15	0.10	0.05	-
PI	productivity index	3.00	2.50	2.00	$kg/s/Pa$
WC	water cut	0.15	0.10	0.05	-

Effects of GOR

Figure 5.10 shows the gas injection inflow rates into the wells with their corresponding set points. These set points are provided by the RTO and

displayed as the red reference line. In addition, the oil outflow rates from each well are displayed on the right. The simulation is displayed when the reservoirs GOR values are subject to increase by 10% at time 2000s, 20% at time 4000s and 30% at time 6000s, from their initial values in Table 5.3. As can be seen in the simulation results, at low GOR values wells 1 and 2 require high gas injection rates. This is due to the large hydrostatic pressure in the tubing. Hence, by injecting more gas into the wells the densities are reduced which leads to the reduction of the bottom hole pressure.

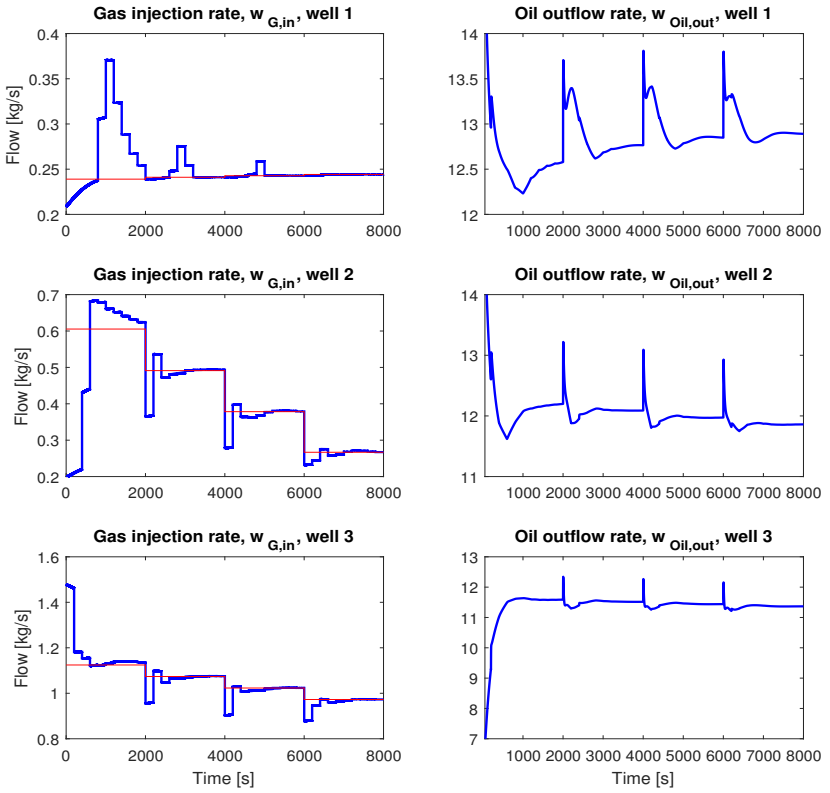


Figure 5.10: Set point tracking of gas injection in the wells when the gas lifted oil network is subject to GOR disturbances. On the left, algebraic variables $w_{G,in}$. On the right, oil outflows $w_{Oil,out}$.

Figure 5.10 shows how the NMPC controller attempts to track optimal steady state values of the gas injection, $w_{G,in}$, generated by the RTO. As can be seen the NMPC ability to track the setpoints while satisfying boundary

conditions and constraints of 4.11 is successful. On the left of Figure 5.10, every 2000s a pump occurs due the disturbance appearance. In Figure 5.10 the gas injection rate into well 1 slowly reaches the steady state point after 700s. However, because well 2 is still not at steady state the controller in well 1 is forced to find another trajectory until all wells reach their reference points.

Further, for varying $GORs$ the oil outflow from the wells slightly increases in well 1 while it decreases in well 2 and 3. The latter is justified by the bottom hole pressures in Figure 5.11.

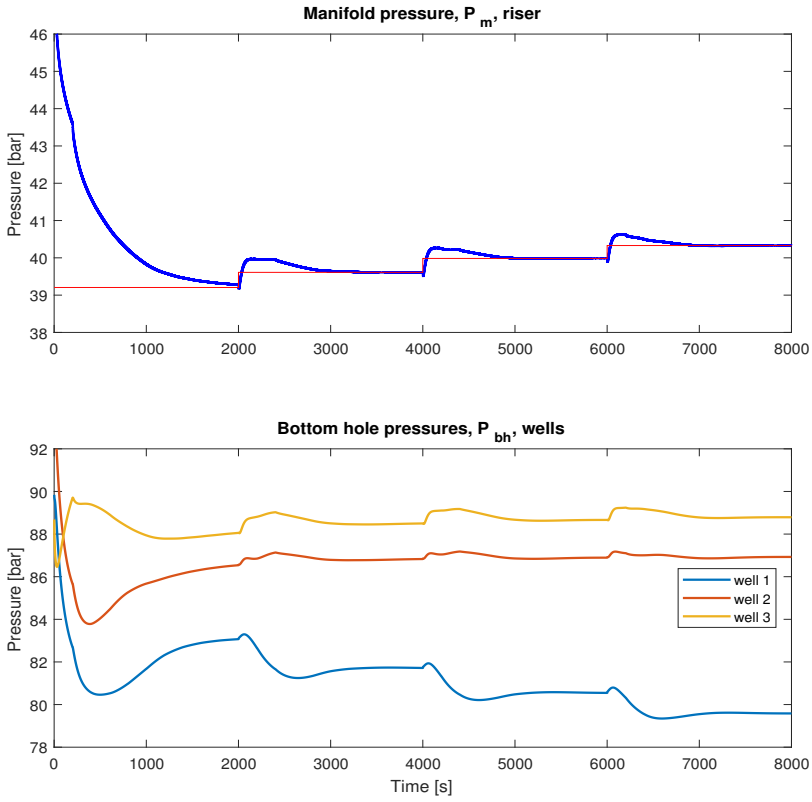


Figure 5.11: Set point tracking of the manifold pressure for new optimal steady state solution when subject to GOR disturbances every 2000s. On the bottom, bottom hole pressures P_{bh} for the wells.

In Figure 5.11 the manifold pressure is displayed, as well as its optimal

steady state setpoint. In addition, the bottom hole pressures in each well are depicted. As indicated, the bottom hole pressure in well 1 decreases every 2000s with increasing GOR. From the conclusions drawn earlier, the lower the bottom hole pressure is, the larger the oil production becomes. The latter justifies why the oil outflow from well 1 increases while the oil outflow from wells 2 and 3 decreases. Moreover, the manifold pressure P_m increases with increasing GOR values. Figure 5.11 shows that the manifold pressure steadily reaches the optimal steady state reference points every time changes are applied to the GOR. This shows that the NMPC controller is successful in tracking.

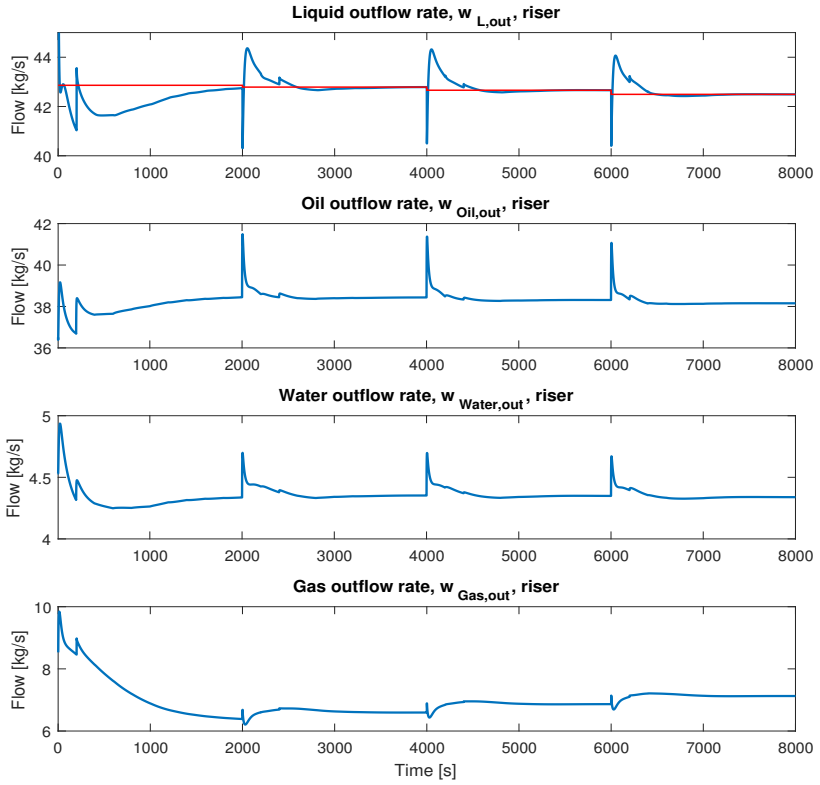


Figure 5.12: Set point tracking of liquid outflow rate for new optimal steady state solution, when subject to GOR disturbances. Additionally, oil, water and gas outflow rates from the top production choke in the riser are depicted.

Finally, the total liquid, oil, water and gas outflows rates through the riser choke are depicted in Figure 5.12. As observed, every 2000s the liquid outflow in the riser is adjusted and the NMPC controller steadily reaches the optimal solution. In general, it can be seen that, to some extent, increasing GOR values is beneficial. However, at time 4000s and 6000s it shows otherwise, as both the total liquid production and the oil production decrease.

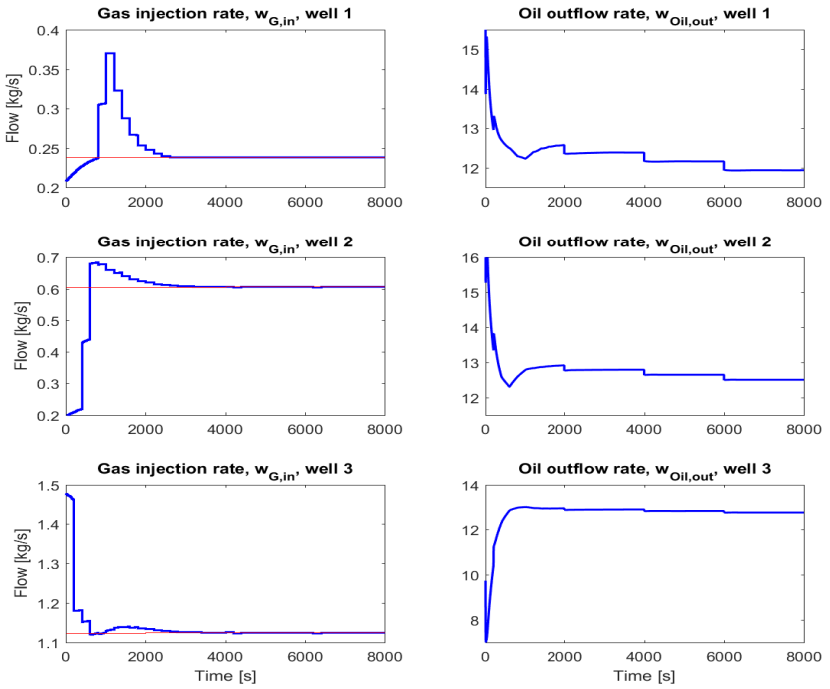


Figure 5.13: On the left set point tracking of the gas injection when subject to WC increase every 2000s. On the right, the oil outflows from each well in the gas lifted well network are depicted.

Effects of WC

The following simulations are made by assuming that the WC increases during the lifetime of the reservoir. In Figure 5.13 the gas injection rates are displayed with their corresponding set points. The red reference lines are the steady state solutions found by the RTO. The RTO runs every 2000s to provide the NMPC controller with suitable set points for tracking.

These set points are the optimal steady state solutions generated each time a disturbance occur on the gas lifted oil network. The disturbances are applied so that WC values are subject to increase by 10% at time 2000s, 20% at time 4000s, and 30% at time 6000s, from their initial values in Table 5.3. The RTO finds new optimal points which are then tracked by the NMPC. In Figure 5.13 after the occurrence of a disturbance, the RTO runs to find the optimal steady state solutions, and the NMPC successfully tracks these solutions as set points.

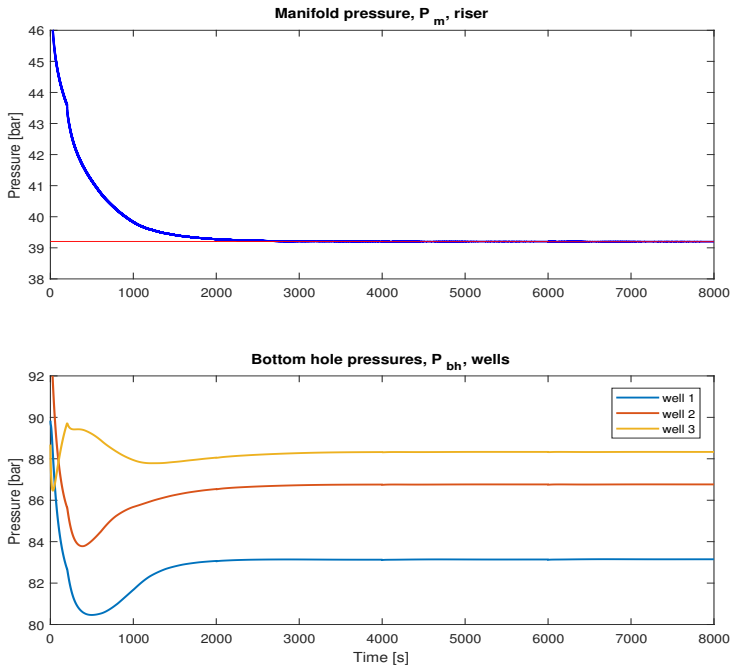


Figure 5.14: On the top, set point tracking of the manifold pressure when subject to WC increase. On the bottom, the bottom hole pressures for each well in the gas lifted oil network are displayed.

Figure 5.14 depicts the manifold pressure with its corresponding setpoint, as well as the bottom hole pressures in the gas lifts. As indicated, increasing water cuts do not affect the system significantly. However the RTO does find new steady state solutions which are insignificantly different. The idea is that increasing WC will subsequently increase the density of the fluid and the pressure at the bottom hole. Therefore for increasing WCs more gas is required for injection to increase oil production.

Figure 5.15 depicts the liquid, oil, water and gas outflow rates from the riser. Given that there are no constraints on the total production and that the quality of the product is not investigated, maximized production is desired at all times. The total liquid outflow is not affected by changing water cuts as expected, since all production chokes are left fully open and only gas injection chokes are manipulated. Hence, more liquid implies more oil.

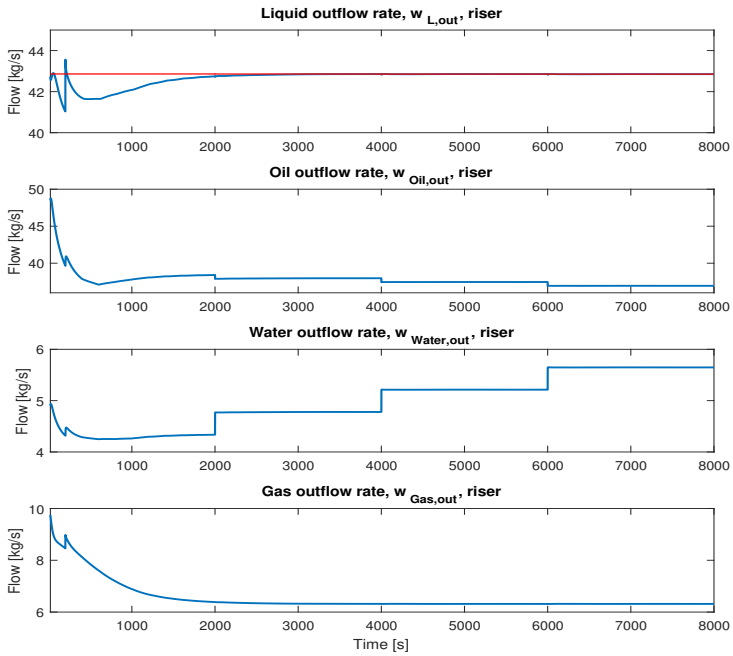


Figure 5.15: Set point tracking of liquid outflow from riser, subject to WC increase. Additionally, oil, water and gas outflow rates from the top production choke are depicted.

The RTO objectives are usually determined over a long term of case studies, eg. gas injection costs, and waste water treatment at other top facilities costs. In this thesis, the maximization of oil and minimization of gas injection are investigated. Given that there are no constraints on production, quality and that all production chokes are fully open, it is reasonable to assume that maximization of liquid leads to maximization of oil. However, if there are constraints on the quality of product, one might need to adjust the production chokes of the gas lifts so that it gives the best combination of products from each well to reduce overall water production.

Chapter 6

Conclusion

In this work, the optimization of production subject to gas injection in a gas lifted oil network consisting of three wells and a riser is taken into consideration. In this thesis, the Hagen-Poiseuille equation is proposed to represent the pressure loss due to friction in the tubing of the gas lifted oil network. In addition, the water cut is introduced to the gas lifted network model to account for a three phase system. The mathematical model is formulated as a DAE system and implemented and integrated in MATLAB using the `ode15s`. Further, a two layer control strategy is proposed in order to control the system optimally and steadily. The first layer applies RTO to produce optimal steady state solutions for maximization of liquid outflow and minimization of gas injection. The second layer employs NMPC strategy to control the flow rates and the manifold pressure. The NMPC optimization problem is formulated for a DAE system. The model inside the NMPC is solved using the direct collocation approach using CasADi within the MATLAB programming environment. Further, the simulations are conducted on the model in order to analyze the behaviour of the system under different circumstances. In addition, several numerical cases are carried out to analyze the behaviour of the RTO and NMPC control system.

Initially, the gas lifted oil network behaviour is analyzed when the system is subject to an open loop step response and parameter changes. It is observed that the gas injection into the system can be both advantageous and disadvantageous. This could lead to lower oil production and inefficient operations when dealing with multiple gas lifts in a network. In addition, it is observed that different parameter combinations such as GOR, WC, and

reservoir pressures could lead to instabilities, and less oil production. Hence, it is important to have a good control structure that takes into account the system's behavior under different conditions and constraints. Therefore, the RTO is suggested to produce the optimal steady states solutions. The RTO is tested by running the optimization problem and then plugging the results into the plant. Further, the NMPC layer is employed to produce an optimal control trajectory to track reference points. The NMPC behavior is studied to analyze the performance of the controller to sensitivity in measurement noise. It is observed that the controller is sensitive to highly noisy measurements. Thus, large measurement noise could lead to wear and tear of the injection valves. However, for low magnitudes of noise, the system performs steadily and the control structure fulfills its objectives. Finally, the control system is studied when disturbances in water cuts and gas to oil ratios are applied. In all cases, the states converged to their optimal steady state setpoints provided by the RTO, while satisfying constraints.

The simulations show that the modified model is a good representation of the gas lifted oil network. In addition, the results show that the control system built is able to effectively counter low measurement noise and produce new optimal steady state points for tracking. The results discussed in the thesis show that the simplified model and the two layer control system consisting of RTO and NMPC respectively are good modeling and control approaches for production optimization.

Future work

In this thesis, the RTO and NMPC are designed by considering flow rates, manifold pressures and the control valves of the injection of gas as the decision variables of the RTO and NMPC. However, other variables can be considered if the production quality is to be optimized, for example the bottom hole pressure or the oil cut in the riser. This must done when the production chokes of the wells are relaxed. In addition, it is not yet discussed to what degree these measurements are really available in the real facilities. Therefore, based on the actual availability of the measurements, an estimator should be introduced to the gas lifted network control structure when testing different scenarios. The choice of the estimator could be the extended kalman filter (EKF), which is one of the widely applied estimators for nonlinear systems. Finally, a better tuning scheme can be introduced to the NMPC built in this thesis.

Bibliography

[1] A. Amiraslani. "Differentiation matrices in polynomial bases". In: *Mathematical Sciences* 10 (June 2016), pp. 47–53. ISSN: 2251-7456. DOI: 10.1007/s40096-016-0177-x. URL: https://doi.org/10.1007/s40096-016-0177-x.

[2] Joel Andersson. "A General-Purpose Software Framework for Dynamic Optimization". PhD thesis. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium: Arenberg Doctoral School, KU Leuven, 2013.

[3] J.D. Clegg, S.M. Bucaram, and N.W. Hein. "Recommendations and Comparisons for Selecting Artificial-Lift Methods". In: *Journal of Petroleum Technology* (Dec. 1993).

[4] M. Diehl and S. Gros. *Numerical Optimal Control*. NOCSE. 2017.

[5] M. Diehl et al. "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations". In: *Journal of Process Control* 12 (2002), pp. 577–585.

[6] M. Dlima. "Nonlinear Model Predictive Control of Gravity Separator". MA thesis. Norwegian University of Science and Technology, 2017.

[7] G. O. Eikrem, O. Aamo, and B. Foss. "On Instability in Gas Lift Wells and Schemes for Stabilization by Automatic Control". In: *SPE Production & Operations* 23 (2008), pp. 268–279.

[8] G. Eikrem, L. Imsland, and B. Foss. "Stabilization of Gas Lifted Wells Based on State Estimation". In: *IFAC Proceedings Volumes* 37 (Jan. 2004), pp. 323–328.

[9] B. Foss and A. Heirung. *Merging Optimization and Control*. Norwegian University of Science and Technology. 2016.

[10] H. Hansen. "A Comparative Study of Control Structures Applied in Gas Lift Systems to Prevent Casing Heading". MA thesis. Norwegian University of Science and Technology, 2012.

[11] B. Hu. "Characterizing gas-lift instabilities". Master thesis. Norwegian University of Science and Technology, 2004.

[12] M. B. Jadid. "The Pressure's On: Innovations in Gas Lift". In: *Oil Field Reveiw, Schlumberger* 18 (2016).

[13] Hugo A Jakobsen. *Chemical Reactor Modeling: Multiphase Reactive Flows*. Cham: Springer International Publishing: Cham, 2014. ISBN: 3319050915. DOI: 10.1007/978-3-319-05092-8.

[14] C. Kirches. "Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control". Ph.D thesis. Heidelberg Graduate School of Mathematical and Computational Methods for Sciences, Oct. 2010.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Science+Business Media, LLC, 2006.

[16] Salvatore P. and R. Skalak. "THE HISTORY Of POISEUILLE'S LAW". In: *Annual Review of Fluid Mechanics* (1993).

[17] A. Plucenio et al. "Gas-lift Optimization and Control with Nonlinear MPC". In: *IFAC Proceedings Volumes* 42 (2009), pp. 904–909.

[18] Kody M. Powell et al. "A Continuous Formulation for Logical Decisions in Differential Algebraic Systems using Mathematical Programs with Complementarity Constraints". In: *Processes* 4 (2016). ISSN: 2227-9717. DOI: 10.3390/pr4010007. URL: http://www.mdpi.com/2227-9717/4/1/7.

[19] James B Rawlings. *Model Predictive Control:Theory and Design*. Nob Hill Publishing,LLC, Aug. 2009.

[20] A. Satter, G. M. Iqbal, and J. L. Buchwalter. *Practical Enhanced Reservoir Engineering: Assisted with Simulation Software*. PennWell Books, 2008.

[21] S. Skogestad. "Plantwide control: the search for the self-optimizing control structure". In: *Journal of Process Control* 10 (Oct. 2000), pp. 487–507.

[22] S. Skogestad and E. Jahanshahi. "Simplified Dynamical Models for Control of Severe Slugging in Multiphase Risers". In: *IFAC Proceedings Volumes* 44 (Jan. 2011), pp. 1634–1639.

[23] S. Skogestad, E. Jahanshahi, and H. Hansen. "Control structure design for stabilizing unstable gas-lift oil wells". In: *IFAC Symposium on Advaance Control of Chemical Processes* (July 2012).

Appendix A

Simulation Parameters

A.1 Subscripts

Table A.1: Subscripts

Subscripts	
Symbol	Description
a	Annulus
t	Tubing in both riser and wells
r	Riser
top	Top
btm	Bottom
bh	Bottom hole
L	Liquid
G	Gas
SS	Steady State
opt	Optimal solution

A.2 Well Model

Table A.2: Well constants

Well Constants

Symbol	Description	Value
R	Universal gas constant	8.314 $J/Kmol$
μ_{Oil}	Viscosity	3.64e-3 $Pa \cdot s$
ρ_{Oil}	Oil density	900 kg/m^3
ρ_{water}	Water density	1000 kg/m^3
g	Gravity	9.81 m/s^2
M_G	Gas molecular weight	16.7e-3 Kg/mol
T_{bh}	Temperature	400 K
T_t	Temperature	369.4 K
L_{bh}	Bottom hole length	75 m
L_t	Tubing length	2048 m
V_t	Tubing volume	25.03 m^3
S_{bh}	Bottom hole cross section	0.0141 m^2
K_{gs}	Gas lift choke constant	9.98e-5 -
K_{inj}	Gas injection choke constant	1.4e-4 -
P_{gs}	Pressure gas lift choke	140 bar

A.3 Riser Model

Table A.3: Riser constants

Riser Constants		
Symbol	Description	Value
R	Universal gas constant	8.314 $J/Kmol$
μ_{Oil}	Viscosity	3.64e-3 $Pa \cdot s$
ρ_{Oil}	Oil density	900 kg/m^3
ρ_{water}	Water density	1000 kg/m^3
g	Gravity	9.81 m/s^2
M_G	Gas molecular weight	16.7e-3 Kg/mol
T_r	Temperature	369.4 K
L_r	Riser length	250 m
S_r	riser cross section	0.0507 m^2
P_s	Pressure Gas lift choke	20 bar

Appendix B

Decision Variables in The RTO and NMPC

Table B.1: Manipulated variables in the gas lifted oil network

Symbol	Description
$u_{2,well1}$	Gas injection choke in well 1
$u_{2,well2}$	Gas injection choke in well 2
$u_{2,well3}$	Gas injection choke in well 3

Table B.2: Differential decision variables

Well 1

Symbol	Description
$x_{1,a}$	Mass of gas in annulus
$x_{2,t}$	Mass of gas in tubing
$x_{3,t}$	Mass of liquid in tubing

Well 2

Symbol	Description
$x_{1,a}$	Mass of gas in annulus
$x_{2,t}$	Mass of gas in tubing
$x_{3,t}$	Mass of liquid in tubing

Well 3

Symbol	Description
$x_{1,a}$	Mass of gas in annulus
$x_{2,t}$	Mass of gas in tubing
$x_{3,t}$	Mass of liquid in tubing

Riser

Symbol	Description
$x_{1,r}$	Mass of gas in tubing
$x_{2,r}$	Mass of liquid in tubing

Table B.3: Algebraic decision variables

Well 1

Symbol	Description
w_{inj}	Inflow of gas injection
w_{res}	Total outflow from reservoir
$w_{out,t}$	Total outflow from tubing

Well 2

Symbol	Description
w_{inj}	Inflow of gas injection
w_{res}	Total outflow from reservoir
$w_{out,t}$	Total outflow from tubing

Well 3

Symbol	Description
w_{inj}	Inflow of gas injection
w_{res}	Total outflow from reservoir
$w_{out,t}$	Total outflow from tubing

Riser

Symbol	Description
w_{out}	Total outflow
$w_{G,out}$	Gas outflow
$w_{L,out}$	Liquid outflow
P_m	Manifold pressure

Appendix C

Programming Codes

C.1 Well Code

```
function [well]= well128_model(t,x,u,Pm,P_res,PI,Kpr,GOR,WC)
% The purpose of this model is to represent mathematically a simplified
% dynamical model for a well with a gas lift . 1. we find the gas mass flow
% into the annulus thus we require to find some pressure relations . 2. we
% find the pressure on the bottom of the tubing next to the injection point
% thus finding mass gas flow injected to the tube. 3. we find bottom hole
% pressure and hence the reservoir mass flow. 4. we find the density on top
% of the tubing tto find the mass flow out and use it in the riser model.


import casadi.*

% State variables of our model are x1 mass gas in annulus, x2 mass gas in
% tubing,x3 mass liquid in tubing
% xdot1 = w_g_in - w_g_inj
% xdot2 = w_g_inj + w_g_res - w_g_out
% xdot3 = w_l_res - w_l_out

% ODE variables
x(1)      = x(1)*1000; % mass of gas in annulus [kg]
x(2)      = x(2)*1000; % mass of gas in tubing
x(3)      = x(3)*1000; % mass of liquid in tubing

% DAE vriables
w_res     = x(4);   %[kg/s]
w_g_inj   = x(5);
w_out     = x(6);
```

```
%%%%%%%%%%%%%%%%%%%%
% Call in parameters
PI     = PI*1e-6;
WC     = WC/100;
GOR    = GOR/100;
Kpr    = Kpr*1e-3;
Pm     = Pm*1e5;
P_res  = P_res*1e5;

% Parameters
Mg     = 16.7e-3;
Va     = 64.34;
Vt     = 25.03;
Ta     = 348;
R      = 8.314;
P_gs   = 140e5;
g      = 9.81;
Lt     = 2048;
La     = 2048;
Dt     = 0.134;
T_bh   = 400;
Tt     = 369.4;
Lbh    = 75;
Sbh    = 0.0141;
Vbh    = Lbh*Sbh;
Dbh    = 0.134;
Kinj   = 1.40e-4;
Kgs    = 9.98e-5;
rho_w  = 1000;
rho_o  = 900;
rho_l  = 1/(WC/rho_w + (1-WC)/rho_o);
mu_o   = 3.64e-3;
mu     = mu_o/(1+WC)^2.5;
%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%
% first step is to find gas mass flow into the annulus
rho_g_a    = x(1)/Va;
rho_g_in   = P_gs*Mg/R/Ta;
P_a_t      = rho_g_a*R*Ta/Mg;
% w_g_in     = Kgs*u(2)*sqrt(rho_g_in*max(P_gs-P_a_t,0));
% w_g_in     = Kgs*u(2)*sqrt(rho_g_in*(P_gs-P_a_t));
w_g_in     = Kgs*u(2)*(rho_g_in*(P_gs-P_a_t))^(0.5);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a_l_ave          = (x(3)-Vbh*rho_l)/Vt/rho_l;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Second step is to find gas mass flow injected into tubing
dP_a             = rho_g_a*g*La;
P_a_b            = P_a_t + dP_a;
rho_g_t_ave      = x(2)/(Vt+Vbh-x(3)/rho_l);
rho_mix_ave      = (x(3)+x(2)-Vbh*rho_l)/Vt;
P_t_t            = rho_g_t_ave*R*Tt/Mg;
Pd_t             = rho_mix_ave*g*Lt;

Q                =a_l_ave*w_out/rho_mix_ave;

% and then we can add GOR to
P_t_fr           = 128*mu*Lt*Q/(pi*Dt^4);
P_t_b            = P_t_fr + Pd_t + P_t_t;
rho_g_a_b        = P_a_b*Mg/R/Ta;
% f1               = w_g_inj-Kinj*sqrt(rho_g_a_b*max(P_a_b-P_t_b,0));
% f1               = w_g_inj-Kinj*sqrt(rho_g_a_b*(P_a_b-P_t_b));
f1               = w_g_inj-Kinj*(rho_g_a_b*(P_a_b-P_t_b))^(0.5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% now we want to find mass flow from reservoir and out of the whole system
% so we find pressure bottom hole in order to calculate the reservoir mass flow


a_g_b_m          = GOR/(GOR+1); % gas mass fraction at the bottom of tubing
dP_bh            = rho_l*g*Lbh;
Q                = w_res/rho_l;      % should we use the average wres
P_bh_fr          = 128*mu*Lbh*Q/pi/Dbh^4;
P_bh             = P_t_b+dP_bh+P_bh_fr;

% f2               = w_res -PI*max(P_res-P_bh,0); % include an fsolve
f2               = w_res -PI*(P_res-P_bh);
w_l_res          = (1-a_g_b_m)*w_res;
w_g_res          = a_g_b_m*w_res;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% after finding the reservoir mass flow we are then able to calculate the
% actual volume fractions at the bottom of the tubing when reservoir mass
```

```
% flow is included and no longer at the very hiiiigggh pressures of the
% well. using the new volume fraction we will be able to calculate the
% actual mixture density at the top of the tubing so that we have a correct
% mass flow out of the gas lift .
rho_g_t_b        = P_t_b*Mg/R/T_bh;

% we could calculate our way up
% a_l_ave          = (x(3)-Vbh*rho_l)/Vt/rho_l;
a_l_b            = w_l_res*rho_g_t_b/(rho_g_t_b*w_l_res + rho_l*(w_g_res+w_g_inj));%
    liquid volume fraction at bottom of tubing
a_l_t            = 2*a_l_ave - a_l_b ;                                            %
    liquid volume fraaction at top of the tubing Jahashahi Skogestad 2011
a_g_t_m          = (1-a_l_t)*rho_g_t_ave/( a_l_t *rho_l+(1-a_l_t)*rho_g_t_ave);       %
    gas mass fraction at top of tubing

rho_mix_t_t      = rho_l* a_l_t + rho_g_t_ave*(1-a_l_t);

% f3               = w_out -Kpr*u(1)*sqrt(rho_mix_t_t*max(P_t_t-Pm,0));
% f3               = w_out -Kpr*u(1)*sqrt(rho_mix_t_t*(P_t_t-Pm));
f3               = w_out -Kpr*u(1)*(rho_mix_t_t*(P_t_t-Pm))^(0.5);
w_g_out          = a_g_t_m*w_out;
w_l_out          = (1-a_g_t_m)*w_out;
%%%%%%%%%%%%%%%%%%%%%%%




%%%%%%%%%%%%%%%%%%%%%%%
% formatting our state equations xdot(1,2,3)

xdot1 = w_g_in - w_g_inj;
xdot2 = w_g_inj + w_g_res - w_g_out;
xdot3 = w_l_res - w_l_out;
%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%
% Defining outputs of our model
xdot          = 1e-3*[xdot1;xdot2;xdot3];
residuals     = [f1 ; f2 ; f3 ];
outflow       = [w_g_out;w_l_out;w_g_in];
y             = [1e-5*P_a_t; 1e-5*P_a_b; 1e-5*P_t_t; 1e-5*P_t_b; 1e-5*P_bh];

well          = [xdot;residuals ;outflow;y ];
%%%%%%%%%%%%%%%%%%%
```

Listing C.1: Source code for modeling the gas lift

C.2 Riser Code

```
function [dots]= Riser128_model(t,x,u,flow1,flow2,flow3,WC_r,Kp_r,P_s)


% the purpuse of this model is 1. to find the amount of oil and gas produced
% at the top of a riser before connecting it to a separator. 2. to find the manifold
    pressure at the bottom of the riser



import casadi.*

%%%%%%%%%%%%%%%%%
% the system is represented such that xdot= f(x,u) , y=h(x,u) , x
% represents the mass of the phases liquid and gas which evolves according
% to f. the function h defines the variables y which gathers the input
% pressure and output mass flow rat variables for each phase
% ODE
x(1)      =  x(1)*1000;  % check why you have to multiple by 1000
x(2)      =  x(2)*1000;

w_out     =  x(3);
w_g_out   =  x(4);
w_l_out   =  x(5);

gas_flow1 = flow1(1);          oil_flow1  = flow1(2);   % this is meant to be liquid
    flow and not oil flow
gas_flow2 = flow2(1);          oil_flow2  = flow2(2);
gas_flow3 = flow3(1);          oil_flow3  = flow3(2);

g_f_in    = [gas_flow1, gas_flow2, gas_flow3];
o_f_in    = [oil_flow1 , oil_flow2 , oil_flow3 ];
%%%%%%%%%%%%%%%%%%%%%%



% call in Parameters
WC_r   = WC_r/100;  % mass water cut [fraction]
P_s    = 20*1e5;    % Pressure after riser production choke [Pa] this can be
    adjusted later
Kp_r   = Kp_r*1e-3; % Production choke constant [-]

% given Parameters
Sr     = 0.0507;    % cross section riser [m2]
Lr     = 250;       % length riser [m]
Vr     = Sr*Lr;     % volume riser [m3]
rho_w  = 1000;      % water density [kg/m3]
rho_o  = 900;       % oil density [kg/m3]
```

```
rho_l    = 1/(WC_r/rho_w + (1-WC_r)/rho_o);        % liquid density [kg/m3]
R        = 8.314;        % universal gas constant [J/mol/K]
g        = 9.81;         % gravity [m/s2]
Tr       = 369.4;        % riser temp. [K]
Mg       = 16.7e-3;      % gas molecular weight [kg/mol]
mu_o     = 3.64e-3;
mu       = mu_o/(1+WC_r)^2.5;    % viscosity [Pa.s]
Dr       = 0.134;        % riser diameter [m]



a_l_ave        = x(2)/Vr/rho_l;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% first step is to find the manifold pressure
rho_g_t        = x(1)/(Vr-x(2)/rho_l);
rho_mix_ave    = sum(x(1:2))/Vr;        % average density around the entire riser
Q              = a_l_ave*w_out/rho_mix_ave;      % average volumetric flow
P_r_t          = R*Tr*rho_g_t/Mg;       % pressure at the top of the riser
Pd             = rho_mix_ave*g*Lr;      % pressure drop from t to b of riser
P_fr           = 128*mu*Lr*Q/(pi*Dr^4); % pressure liquid friction with inner surface
    area or riser
Pm             = P_r_t+Pd+P_fr;    % pressure at manifold, first objective of this
    model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% second step is to find mass flow rates of liquid and qas
% [SARRY] try using only average and see how results vary
% the reason for that is each well will have different densities of gas and
% liquid as they enter the bottom of the riser
% the reason we could do that is because we have Pm and this allows us to
% find the density at the maniflod aka bottom

% a_l_ave        = x(2)/Vr/rho_l;                    % average liquid volume fraction in
    riser
rho_g_b        = Pm*Mg/R/Tr;                       % Density at the bottom of riser
a_l_b          = sum(o_f_in)*rho_g_b/(sum(g_f_in)*rho_l + sum(o_f_in)*rho_g_b); % liquid
     volume fraction at the bottom
a_l_t          = 2*a_l_ave - a_l_b;                % this is a relation from Jahanshahi
    and Skogestad (2011)
rho_mix_t      = a_l_t*rho_l+(1-a_l_t)*rho_g_t;  % this is the density mix at the top
    since it will vary from bottom to top
% f              = w_out - Kp_r*u*sqrt(rho_mix_t*max(P_r_t-P_s,0)); % residule for
    fsolve of w_out
%  f              = w_out - Kp_r*u*sqrt(rho_mix_t*(P_r_t-P_s));
f              = w_out - Kp_r*u*(rho_mix_t*(P_r_t-P_s))^(0.5);
a_m_g_t        = (1-a_l_t)*rho_g_t/((1-a_l_t)*rho_g_t+a_l_t*rho_l); % gas mass fraction
    at top of the riser
```

```
f1              = w_g_out - a_m_g_t*w_out;          % residule for fsolve of w_g_out
f2              = w_l_out - (1-a_m_g_t)*w_out;      % residule for fsolve of w_l_out

xdot1           = sum(g_f_in)-w_g_out;              % gas mass flow over riser
xdot2           = sum(o_f_in)-w_l_out;              % liquid mass flow over riser

%%%%%%%%%%%%%%%%%%%%%%%%%

% the output is [dots] of size 10:
%[xdot1, xdot2, f , f1 ,f2 , Pm , mass_gass_out, mass_water_out,mass_oil_out,P_r_t]
dots=[xdot1*1e-3;xdot2*1e-3;f;f1;f2;1e-5*Pm;w_l_out];%w_g_out;WC_r*w_l_out;(1-
    WC_r)*w_l_out;P_r_t*1e-5]; % if additional outputs are desired
```

Listing C.2: Source code for modeling the riser

C.3 Network Code

```
function [y] = network_128(t,x,u,p)

%latest change i did here was that i added L as an output of this model

% in this function we combine three wells towards a manifold that enters a
% riser. each of the wells will have different properties of PI,Kpr,GOR,WC,
% and will recieve Pm from the riser. The riser in return will accept
% values of gas and liquid flows from each well and the total flow to
% complete the circuit.
import casadi.*


x_well1 = [x(1:3); x(12:14)];    % the fiirst 24 states are the ones taken into
    account
x_well2 = [x(4:6); x(15:17)];
x_well3 = [x(7:9); x(18:20)];
x_riser = [x(10:11); x(21:23)];
Pm      =  x(24);     % has been changed



%         !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% % Definig all constants to each model PI,P_res,GOR, WC,K_pr
% GOR     = [0.15 ; 0.1 ;  0.15];       % for [well1 well2 well3]
% WC      = [0.15 ; 0.1 ;  0.05];       % for [well1 well2 well3]
% Kpr     = [2.8   ; 3.0 ; 3.2  ;3.4];  % for [well1 well2 well3 riser]
%
% p=[GOR(1),WC(1),Kpr(1),...
%    GOR(2),WC(2),Kpr(2),...
```

```
%    GOR(3),WC(3),Kpr(3),...
%    Kpr(4)];
%         !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


% Definig all constants to each model PI,P_res,GOR, WC,K_pr, (Deafult)
GOR     = [p(1); p(4); p(7)]*100;          % for [well1 well2 well3]
PI      = [3     ; 2.5 ; 2];               % for [well1 well2 well3]
P_res   = [140   ; 150 ; 160];             % for [well1 well2 well3]
WC      = [p(2); p(5); p(8)]*100;          % for [well1 well2 well3]
Kpr     = [p(3); p(6); p(9); p(10)];       % for [well1 well2 well3 riser]

% % Definig all constants to each model PI,P_res,GOR, WC,K_pr
% GOR     = [0.15 ; 0.1 ;  0.15];          % for [well1 well2 well3]
% PI      = [3     ; 2.5 ; 2];             % for [well1 well2 well3]
% P_res   = [140   ; 150 ; 160];           % for [well1 well2 well3]
% WC      = [0.15 ; 0.1 ;  0.05];          % for [well1 well2 well3]
% Kpr     = [2.8   ; 3.0 ; 3.2  ;3.4];     % for [well1 well2 well3 riser]

% % Definig all parameters equal, use this only when all p are equal
% GOR     = [p(1); p(4); p(7)]*100;          % for [well1 well2 well3]
% PI      = [3     ; 3 ; 3];                 % for [well1 well2 well3]
% P_res   = [140   ; 140 ; 140];             % for [well1 well2 well3]
% WC      = [p(2); p(5); p(8)]*100;          % for [well1 well2 well3]
% Kpr     = [p(3); p(6); p(9); p(10)];       % for [well1 well2 well3 riser]

% % Definig all parameters equal,
% GOR     = [p(1); p(4); p(7)]*100;          % for [well1 well2 well3]
% PI      = [3     ; 3 ; 3];                 % for [well1 well2 well3]
% P_res   = [150   ; 150 ; 150];             % for [well1 well2 well3]
% WC      = [p(2); p(5); p(8)]*100;          % for [well1 well2 well3]
% Kpr     = [p(3); p(6); p(9); p(10)];       % for [well1 well2 well3 riser]

u_well1 = [u(1);u(2)];                     % for well 1
u_well2 = [u(3);u(4)];                     % for well 2
u_well3 = [u(5);u(6)];                     % for well 3
u_riser = u(end);                          % for riser

%%%%%%%%%%%%%%%%
%calling well model so we could find the flows and connect it to our riser
well1 = well128_model(t,x_well1,u_well1,Pm,P_res(1),PI(1),Kpr(1),GOR(1),WC(1));

well2 = well128_model(t,x_well2,u_well2,Pm,P_res(2),PI(2),Kpr(2),GOR(2),WC(2));

well3 = well128_model(t,x_well3,u_well3,Pm,P_res(3),PI(3),Kpr(3),GOR(3),WC(3));
%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%
```

```
% calling riser model to find Pm pressure and connect with well models
flow1= well1(7:8);
flow2= well2(7:8);
flow3= well3(7:8);

WC_r = (flow1(2)*WC(1) +flow2(2)*WC(2) + flow3(2)*WC(3))/ (flow1(2)+flow2(2)+
    flow3(2));

riser = Riser128_model(t,x_riser, u_riser ,flow1,flow2,flow3,WC_r,Kpr(4));

f     = riser(6)- Pm;
%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%
% Organized outputs
diff = [well1(1:3); well2(1:3); well3(1:3); riser(1:2)] ; % 11 xdots
alg  = [well1(4:6); well2(4:6); well3(4:6); riser(3:5);f] ;% 13 residuals


measurements=[riser(7);well1(7:end);well2(7:end);well3(7:end)];



% y: 24 elements consisting of all states (11), and fsolve variables (13) a simple
    vector

% measurements consisting of the following :
% 1 w_l_out                 Riser %1
% 2   w_g_out
% 3   w_l_out
% 4   w_g_in
% 5   P_a_t                 WELL 1 %8
% 6   P_a_b
% 7   P_t_t
% 8   P_t_b
% 9   P_bh
% 10    w_g_out
% 11    w_l_out
% 12    w_g_in
% 13    P_a_t             WELL 2 %8
% 14    P_a_b
% 15    P_t_t
% 16    P_t_b
% 17    P_bh
% 18        w_g_out
% 19        w_l_out
% 20        w_g_in
% 21        P_a_t         WELL 3 %8
% 22        P_a_b
```

```
% 23          P_t_t
% 24          P_t_b
% 25          P_bh


y      = [diff;alg;measurements];
end
```

Listing C.3: Source code for the gas lifted network

C.4 RTO Code

```
function [xSS,uSS]=RTO_network128(x0,u0,p)


% clc;
% clear; df
% close all;

addpath('C:\Users\sarriyh\Downloads\casadi-matlabR2014b-v3.2.3')

import casadi.*


% in this case the upper and lower bound of the production choke of the
% riser are equal such that is always open. in addition the objective
% function is to minimize injected gas and maximize oil production from
% riser.

 % defining symbolic variables

 % well 1
 x1  = MX.sym('x1');  % Mass of gas in annulus
 x2  = MX.sym('x2');  % Mass of  gas in tubing
 x3  = MX.sym('x3');  % Mass of liquid in tubing
 % well 2
 x4  = MX.sym('x4'); % Mass of gas in annulus
 x5  = MX.sym('x5'); % Mass of  gas in tubing
 x6  = MX.sym('x6'); % Mass of liquid in tubing

 % well 3
 x7  = MX.sym('x7');  % Mass of gas in annulus
 x8  = MX.sym('x8');  % Mass of  gas in tubing
 x9  = MX.sym('x9');  % Mass of liquid in tubing

 % riser
```

```
x10 = MX.sym('x10'); % Mass of gas in tubing
x11 = MX.sym('x11'); % Mass of liquid in tubing

% algebraic variables

% well 1
x12 = MX.sym('x12'); % Inflow of gas injection
x13 = MX.sym('x13'); % Total outflow from reservoir
x14 = MX.sym('x14'); % Total outflow from tubing

% well 2
x15 = MX.sym('x15'); % Inflow of gas injection
x16 = MX.sym('x16'); % Total outflow from reservoir
x17 = MX.sym('x17'); % Total outflow from tubing

% well 3
x18 = MX.sym('x18'); % Inflow of gas injection
x19 = MX.sym('x19'); % Total outflow from reservoir
x20 = MX.sym('x20'); % Total outflow from tubing

x21 = MX.sym('x21'); % Total outflow
x22 = MX.sym('x22'); % Gas outflow
x23 = MX.sym('x23'); % Liquid outflow
x24 = MX.sym('x24'); % Manifold pressure

x = [x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11;x12;x13;x14 ;...
     x15;x16;x17;x18;x19;x20;x21;x22;x23;x24];

u1 = MX.sym('u1');
u2 = MX.sym('u2');
u3 = MX.sym('u3');
u4 = MX.sym('u4');
u5 = MX.sym('u5');
u6 = MX.sym('u6');
u7 = MX.sym('u7');

u=[u1;u2;u3;u4;u5;u6;u7];

%          !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% is that even correct ?????? i get the same solution

% Definig all constants to each model PI,P_res,GOR, WC,K_pr
% GOR    = [0.15 ; 0.1 ;  0.15];      % for [well1 well2 well3]
% WC     = [0.15 ; 0.1 ;  0.05];      % for [well1 well2 well3]
% Kpr    = [2.8  ;  3.0 ;  3.2  ;3.4]; % for [well1 well2 well3 riser]
%
% p=[GOR(1),WC(1),Kpr(1),...
%    GOR(2),WC(2),Kpr(2),...
%    GOR(3),WC(3),Kpr(3),...
```

```
%    Kpr(4)];
%            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


model  = network_128(0,x,u,p);
 diff    = model(1:11);
alg      = model(12:24);
%L       = model(25) ;
% lbg  = zeros(24,1);
% ubg  = lbg;



% lower and upper bounds x(diff and alg) and u  x1..x24 represent the same
% quantities presented above

%diff wells and riser
x1_lb   = 1e-3;
x2_lb   = 1e-3;
x3_lb   = 1e-3;
x4_lb   = 1e-3;
x5_lb   = 1e-3;
x6_lb   = 1e-3;
x7_lb   = 1e-3;
x8_lb   = 1e-3;
x9_lb   = 1e-3;
x10_lb = 1e-3;
x11_lb = 1e-3;
%algebraic wells
x12_lb = 1e-2;
x13_lb = 1e-2;
x14_lb = 1e-2;
x15_lb = 1e-2;
x16_lb = 1e-2;
x17_lb = 1e-2;
x18_lb = 1e-2;
x19_lb = 1e-2;
x20_lb = 1e-2;
%algebraic riser
x21_lb = 1e-2;
x22_lb = 1e-2;
x23_lb = 1e-2;
%manifold pressure
x24_lb = 0.1;


%DOF riser and wells production chokes and gas injection chokes
u1_lb=0.1;
u2_lb=0.1;
```

```
u3_lb=0.1;
u4_lb=0.1;
u5_lb=0.1;
u6_lb=0.1;
u7_lb=1;
% setting the above in a  vertical  concatenation( a  seriec  of  inconnected
% things)
lbx = vertcat(x1_lb,x2_lb,x3_lb,x4_lb,x5_lb,x6_lb,x7_lb,x8_lb,x9_lb,x10_lb,x11_lb ,...
    x12_lb,x13_lb,x14_lb,x15_lb,x16_lb,x17_lb,x18_lb,x19_lb,x20_lb,x21_lb,x22_lb,
     x23_lb,x24_lb);

lbu = vertcat(u1_lb,u2_lb,u3_lb,u4_lb,u5_lb,u6_lb,u7_lb);


% same as above only for UPPER bounds as can be seen the production is
% relaxed by allowing the bounds to be very large
x1_ub  = 10e7;
x2_ub  = 10e7;
x3_ub  = 10e7;
x4_ub  = 10e7;
x5_ub  = 10e7;
x6_ub  = 10e7;
x7_ub  = 10e7;
x8_ub  = 10e7;
x9_ub  = 10e7;
x10_ub = 10e7;
x11_ub = 10e7;

x12_ub = 50e4;
x13_ub = 50e4;
x14_ub = 50e4;
x15_ub = 50e4;
x16_ub = 50e4;
x17_ub = 50e4;
x18_ub = 50e4;
x19_ub = 50e4;
x20_ub = 50e4;

x21_ub = 50e4;
x22_ub = 50e4;
x23_ub = 50e4;

x24_ub = 150e4;

u1_ub=1;
u2_ub=1;
u3_ub=1;
u4_ub=1;
u5_ub=1;
```

```
u6_ub=1;
u7_ub=1;

ubx = vertcat(x1_ub,x2_ub,x3_ub,x4_ub,x5_ub,x6_ub,x7_ub,x8_ub,x9_ub,x10_ub,x11_ub,
      x12_ub,x13_ub,x14_ub,x15_ub,x16_ub,x17_ub,...
     x18_ub,x19_ub,x20_ub,x21_ub,x22_ub,x23_ub,x24_ub);

ubu = vertcat(u1_ub,u2_ub,u3_ub,u4_ub,u5_ub,u6_ub,u7_ub);


% decision variables
w   = {};              % why does this work and why do we put w first what is w ? why
      do brackets differ ?
w0  = [];              % why does this work and why do we put w first what is w ?
lbw = [];
ubw = [];

% constraints
g   = {};
lbg = [];
ubg = [];


w   = {w{:},x,u};     % why does this work and why do we put w first what is w ?
lbw = [lbw;lbx;lbu];
ubw = [ubw;ubx;ubu];
w0  = [w0;x0;u0];     % why does this work and why do we put w first what is w ?

%Add the system model as constraints
g   = {g{:},vertcat( diff ,alg)};
lbg = [lbg;zeros(24,1)];
ubg = [ubg;zeros(24,1)];

%stage cost
L   = -x23 +x19 +x16 +x13;  % max oil min gas % PI used to affect results
% L   = -x23;                    % max oil
% L   = +x19 +x16 +x13;       % min gas

% Economic objective
J   = L;


nlp = struct('x',vertcat(w{:}),'f',J,'g',vertcat(g{:}));
solver = nlpsol('solver','ipopt',nlp);  % NLP solver IPOPT
sol = solver('x0',w0,'lbx',lbw,'ubx',ubw,'lbg',lbg,'ubg',ubg); % then we feed it
      values initial lower and upper bounds of x and g

% Extracting solutions
w_opt_SS = full(sol.x);
```

```
uSS=w_opt_SS(25:end);
xSS=w_opt_SS(1:24);
```

Listing C.4: Source code for RTO

C.5 Collocation Setup Code

```
function [B,C,D,d] = collocationSetup()
% Joel Andersson, joel@casadi.org, 2016
import casadi.*

% Degree of interpolating polynomial
d = 3;

% Get collocation points
%tau_root = [0 collocation_points(d, 'legendre')];
tau_root = [0 collocation_points(d, 'radau')];

% Coefficients of the collocation equation
C = zeros(d+1,d+1);

% Coefficients of the continuity equation
D = zeros(d+1, 1);

% Coefficients of the quadrature function
B = zeros(d+1, 1);

% Construct polynomial basis
for j=1:d+1
  % Construct Lagrange polynomials to get the polynomial basis at the collocation
    point
  coeff = 1;
  for r=1:d+1
    if r ~= j
      coeff = conv(coeff, [1, -tau_root(r)]);
      coeff = coeff / (tau_root(j)-tau_root(r));
    end
  end
  % Evaluate the polynomial at the final time to get the coefficients of the
    continuity equation
  D(j) = polyval(coeff, 1.0);

  % Evaluate the time derivative of the polynomial at all collocation points to get
    the coefficients of the continuity equation
  pder = polyder(coeff);
  for r=1:d+1
```

```
    C(j,r) = polyval(pder, tau_root(r));
  end

  % Evaluate the integral of the polynomial to get the coefficients of the quadrature
      function
  pint = polyint(coeff);
  B(j) = polyval(pint, 1.0);
end
```

Listing C.5: Source code for collocation setup [2]

C.6 NMPC Code

```
function [w,w0,J,u_nlp_opt,x_nlp_opt] = optProblem_net128_test0412(N,u0,p,x0,z0,
    xz0_measured,uSS,xSS)


% xz0_measured should be a column vector of length nx+xz
% N pridiction horizon
% u0 and x0 z0 are merley initial states
% this optimization problem maximizes oil production from riser, while
% tries to simultaneuously minimize the injected gas into the system. the
% production chokes of the riser and lifts are always fully opened



addpath('C:\Users\sarriyh\Downloads\casadi-matlabR2014b-v3.2.3')
import casadi.*

load('Qxzu.mat')




% Initial values these values could be changed for a better guess
% N= 20;
nx=11;
nz=13;
nu=7;
T=4000;
% T=200;

tf=T/N;


% lower and upper bounds x(diff and alg) and u
```

```
%diff wells and riser
x1_lb  = 1e-2;
x2_lb  = 1e-2;
x3_lb  = 1e-2;
x4_lb  = 1e-2;
x5_lb  = 1e-2;
x6_lb  = 1e-2;
x7_lb  = 1e-2;
x8_lb  = 1e-2;
x9_lb  = 1e-2;
x10_lb = 1e-2;
x11_lb = 1e-2;

%algebraic wells
x12_lb = 1e-2;
x13_lb = 1e-2;
x14_lb = 1e-2;
x15_lb = 1e-2;
x16_lb = 1e-2;
x17_lb = 1e-2;
x18_lb = 1e-2;
x19_lb = 1e-2;
x20_lb = 1e-2;

%algebraic riser
x21_lb = 1e-2;
x22_lb = 1e-2;
x23_lb = 1e-2;

%manifold pressure
x24_lb = 0.1;

%DOF riser and wells
u1_lb=1;
u2_lb=0.1;
u3_lb=1;
u4_lb=0.1;     % works well also with mpc
u5_lb=1;
u6_lb=0.1;
u7_lb=0.1;


% setting the above in a vertical concatenation( a seriec of inconnected
% things)
lbx = vertcat(x1_lb,x2_lb,x3_lb,x4_lb,x5_lb,x6_lb,x7_lb,x8_lb,x9_lb,x10_lb,x11_lb);

lbz = vertcat(x12_lb,x13_lb,x14_lb,x15_lb,x16_lb,x17_lb,x18_lb,x19_lb,x20_lb,x21_lb,
      x22_lb,x23_lb,x24_lb);
```

```
lbu = vertcat(u1_lb,u2_lb,u3_lb,u4_lb,u5_lb,u6_lb,u7_lb);


% same as above only for UPPER bounds
x1_ub  = 10;
x2_ub  = 10;
x3_ub  = 10;
x4_ub  = 10;
x5_ub  = 10;
x6_ub  = 10;
x7_ub  = 10;
x8_ub  = 10;
x9_ub  = 10;
x10_ub = 10;
x11_ub = 10;

x12_ub = 50;
x13_ub = 50;
x14_ub = 50;
x15_ub = 50;
x16_ub = 50;
x17_ub = 50;
x18_ub = 50;
x19_ub = 50;
x20_ub = 50;

x21_ub = 53;
x22_ub = 50;
x23_ub = 50; % this can be contrained by setting maximum amount of water however
    then production chokes of the lifts must be relaxed
%x23_ub = 50;


x24_ub = 150;

u1_ub  =1;
u2_ub  =1;
u3_ub  =1;
u4_ub  =1;
u5_ub  =1;
u6_ub  =1;
u7_ub  =1;

ubx = vertcat(x1_ub,x2_ub,x3_ub,x4_ub,x5_ub,x6_ub,x7_ub,x8_ub,x9_ub,x10_ub,x11_ub);

ubz = vertcat(x12_ub,x13_ub,x14_ub,x15_ub,x16_ub,x17_ub,x18_ub,x19_ub,x20_ub,
    x21_ub,x22_ub,x23_ub,x24_ub);

ubu = vertcat(u1_ub,u2_ub,u3_ub,u4_ub,u5_ub,u6_ub,u7_ub);
```

```
% begin p      !!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Definig all constants for each model PI,P_res,GOR, WC,K_pr
% GOR    = [0.15 ; 0.1 ;  0.15];      % for [well1 well2 well3]
% WC     = [0.15 ; 0.1  ;  0.05];      % for [well1 well2 well3]
% Kpr    = [2.8   ;  3.0  ;  3.2   ;3.4];  % for [well1 well2 well3 riser ]

% p=[GOR(1),WC(1),Kpr(1),...
%    GOR(2),WC(2),Kpr(2),...
%    GOR(3),WC(3),Kpr(3),...
%    Kpr(4)];
%End p      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!




%%%%%%%%%%%
% on this side we put the main ideas of collocation and then later add

uSS=uSS;
xSS=xSS;


% Collocation setup  double check
[B,C,D,d]= collocationSetup();
% Build NLP solver
% empty NLP
w   = {};
w0  = [];
lbw = [];
ubw = [];
J   = 0;

g   = {};
lbg = [];
ubg = [];

% initial conditions
X0  = MX.sym('X0',nx);
Z0  = MX.sym('Z0',nz);
w   = {w{:},X0,Z0};
w0  = [w0;x0;z0]; % what are those two values ? x0 z0 in nmpc gaslift defined dx0 z0
lbw = [lbw;lbx;lbz ];
ubw = [ubw;ubx;ubz];
```

```
% Begin penalty
ubd_du = vertcat  (0.1,0.1,0.1,0.1,0.1,0.1,0.1)  ;
lbd_du = vertcat(-0.1,-0.1,-0.1,-0.1,-0.1,-0.1,-0.1);
% End penalty



% formulate NLP
Xk=X0;
Zk=Z0;
Xkj={};
Zkj={};



g= {g{:},[X0;Z0]-[xz0_measured]}; % x0 closing the loop
lbg=[lbg;zeros(nx+nz,1)];
ubg=[ubg;zeros(nx+nz,1)];

for k=0:N-1
    Uk  =MX.sym(['U_' num2str(k)],nu);
    w   ={w{:},Uk};
    lbw =[lbw;lbu];
    ubw =[ubw;ubu];
    w0  =[w0;u0];

    % Regularization term for control

    % the below can be added or not
    Jcontrol = (Uk-uSS)'*diag(Qxzu(nx+nz+1:end,1))*(Uk-uSS); % Add Q or R
     weighting , uSS (reference point)

    Xkj={};
    Zkj={};

    for j=1:d % d  degree of collo.  polynomial ax^3 + bx^2 ...
        Xkj{j}=MX.sym(['X_' num2str(k) '_' num2str(j)],nx); % this looks like  (nx)x1
     casadi.MX vector in this case 11x1  []
        Zkj{j}=MX.sym(['Z_' num2str(k) '_' num2str(j)],nz);
        w   = {w{:},Xkj{j},Zkj{j}};
        w0  =[w0;x0;z0]; % initial  values only
        lbw =[lbw;lbx;lbz ];
        ubw =[ubw;ubx;ubz];
    end

    % loop over collocation  points
```

```
Xk_end=D(1)*Xk; %D(1) is the coeff of continuity eq. at polynomial at final time

for j=1:d
    % Expression for the state derivative at the collocation point
    xp = C(1,j+1)*Xk;  %coeff of collocation equation C is a 4x4 matrix


    for r=1:d
        xp=xp+C(r+1,j+1)*Xkj{r};
    end


    model   = network_128(0,[Xkj{j};Zkj{j}],Uk,p); % here p is supplied
    diff    = model(1:11);                          %hard coded
    alg     = model(12:24);

    g={g{:},tf*diff-xp,alg}; % dynamics and algebraic constraints
    lbg = [lbg;zeros(nx,1);zeros(nz,1)];
    ubg = [ubg;zeros(nx,1);zeros(nz,1)];

    % add contribution to the states
    Xk_end=Xk_end +D(j+1)*Xkj{j};


end

% add control moves and penalty term in the objective Function
% Begin Penalty
  if k>0
    % Equality Constraint
    g = {g{:},(Uk_prev - Uk)};
    lbg = [lbg;lbd_du];
    ubg = [ubg;ubd_du];

    % Include Penalty Term In Objective function with weight
    J_penalty= (Uk_prev-Uk)'*diag([ones(7,1)])*(Uk_prev-Uk); %weighting
   should be added
    J=J+J_penalty;
  end
% End Penalty

  % New NLP variable for state at end of interval
  Xk    = MX.sym(['X_' num2str(k+1)], nx);
  w     = {w{:},Xk};
  lbw   = [lbw;lbx];
  ubw   = [ubw;ubx];
  w0    = [w0; x0];

  % Shooting Gap constraint
```

```
    g    = {g{:},Xk_end-Xk};
    lbg = [lbg;zeros(nx,1)];
    ubg = [ubg;zeros(nx,1)];



    % regularization term for state variable
    Jstate =([Xk;Zkj{j}] - xSS)' *diag(Qxzu(1:nx+nz,1))* ([Xk;Zkj{j}] - xSS); % add
      a Q weighting


    % economic objective state cost if you would like to have an economic
    % NMPC
     Jecon  = -Zkj{j}(12)+Zkj{j}(8)+Zkj{j}(5)+Zkj{j}(2); % max oil min gas
%     Jecon2 = -Zkj{j}(12);                              % max oil
%     Jecon3 = Zkj{j}(8)+Zkj{j}(5)+Zkj{j}(2);            % min gas



     % stage cost Cases
% J = J + Jstate;                    % tracking only
% J = J + Jstate + Jcontrol;         % tracking and control



% if u like to have an economic NMPC

J = J + Jecon + Jstate + Jcontrol; % Mix tracking econ (stable)
% J = J + Jecon + Jcontrol;          % Mix econ control (less stable)
% J = J + Jecon;                     % Pure econ max oil min gas(unstable)

% J = J + Jecon2;                      % Pure econ max oil(Doesnt work)
% J = J + Jecon2 + Jstate + Jcontrol; % Mix tracking econ2 (stable)

% J = J + Jecon3 + Jstate + Jcontrol; % Mix tracking econ3 (stable)
% J = J + Jecon3 ;  % econ3 (doesnt work)



% save previous inputs
Uk_prev=Uk;

end


nlp = struct('x',vertcat(w{:}),'f',J,'g',vertcat(g{:})); % note i removed p since
     there is no need for it
```

```
% if you would like some extra options
%     options = struct;
%     options.ipopt.tol          = 1e-12;
%     options.acceptable_compl_inf_tol      = 1e-6;
%     solver  = nlpsol('solver', 'ipopt', nlp, options);
solver  = nlpsol('solver', 'ipopt', nlp); % i removed options for the time being
%     sol           = solver('x0', w0, 'p', paramModel.GOR, 'lbx', lbw, 'ubx', ubw, '
    lbg', lbg, 'ubg', ubg);
tic
sol  = solver('x0',w0,'lbx',lbw,'ubx',ubw,'lbg',lbg,'ubg',ubg); % then we feed it
    values initial lower and upper bounds of x and g
toc
w_opt_SS = full(sol.x); % 24 initial x0 z0 then 20 loops of 7 24 24 24 11 to get value
    take value every 90th



% the below is to stop when solver Error
    success = strcmp(solver.stats.return_status,'Infeasible_Problem_Detected');
    if (success)
        keyboard;
    end


    % this plotting function is taken from EKA
[u_nlp_opt, x_nlp_opt] = plotStatesGL_test0412(w_opt_SS, lbw, ubw, N);

end
```

Listing C.6: Source code for NMPC

C.7 Control Structure Code

```
% This script  is an MPC subject to Noise,
% noise can be simply removed by commenting it out

% This script uses optProblem_net128_test0412,
% plotStatesGL_test0412,network128...


% addpath('C:\Users\sarriyh\Downloads\casadi-matlabR2014b-v3.2.3')
% import casadi.*

clc
clear
close all
```

```
% begin              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Definig all constants for each model PI,P_res,GOR, WC,K_pr
GOR     = [0.15 ; 0.1  ;  0.15];       % for [well1 well2 well3]
WC      = [0.15 ; 0.1  ;  0.05];       % for [well1 well2 well3]
Kpr     = [2.8   ; 3.0  ; 3.2  ;3.4];  % for [well1 well2 well3 riser]

p=[GOR(1),WC(1),Kpr(1),...  % for well1
   GOR(2),WC(2),Kpr(2),...  % for well2
   GOR(3),WC(3),Kpr(3),...  % for well3
   Kpr(4)];                 % for riser
%End              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


x0 =[3.1933 0.5850 4.4132 3.3977 0.5601 5.0101 3.1600 0.5846 4.2971 0.2646  1.2903]';

z0 =[15.6427 1.0425 15 14.3927 1.1855 20 14.6892 1.0511 28 48.0037 8.5439 39.4598
      48.0963]';
u0  =[1;0.5;1;0.6;1;0.5;1];


% Steady state values from RTO
[xSS,uSS] = RTO_network128([x0;z0],u0,p);


% Pridiction horizon and run time
T          = 4000;
%  T          = 200;
Nmpc     = 20;
% Nmpc=200;
Nopt      = 20;
ode_time = T/Nopt;


% Initialization
xz_append    =[];
xz0_measured =[x0;z0];
u_opt_append =[];
outputs_append=[];


% MPC itiration
tic
for i=1:Nmpc
    fprintf('\n MPC iteration = %d \n',i);
    % run optimization
[w,w0,J,u_nlp_opt,x_nlp_opt] = optProblem_net128_test0412(Nopt,u0,p,x0,z0,
     xz0_measured,uSS,xSS);
```

```
    % extract control input
u_opt=u_nlp_opt(:,1);
u_ode= u_opt;
u_opt_append=[u_opt_append,u_ode];


    % Run Ode15s
M=diag([ones(1,11) zeros(1,13)]);
options=odeset('Mass',M);

%  begin p             !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FF= @(t,x) [eye(24,24) zeros(24,25)]*network_128(0,x,u_ode,p);
%   end  p             !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


tspan=0:1:ode_time-1;
    % [t,xz] = ode15s(FF,tspan,[x0;z0],options);
[t,xz] = ode15s(FF,tspan,xz0_measured,options);

    % extract states and time
xz_append =[xz_append;xz];



    % update xz0_measured
xz0_measured=[xz(end,:)]';

%noise              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%     % noise creation   low order
% noise=zeros(24,1);
% noise(2)=rand(1)*1e-4; %mass gass well 1
% noise(5)=rand(1)*1e-4; % "         well 2
% noise(8)=rand(1)*1e-4; % "         well 3
% noise(10)=rand(1)*1e-4; % "         riser
%
% noise(1)=rand(1)*1e-3; %
% noise(3)=rand(1)*1e-3; %
% noise(4)=rand(1)*1e-3; %
% noise(6)=rand(1)*1e-3; %
% noise(7)=rand(1)*1e-3; %
% noise(9)=rand(1)*1e-3; %
% noise(11)=rand(1)*1e-3; %

    % noise creation   high order
noise=zeros(24,1);
noise(2)=rand(1)*1e-3; %mass gass well 1
noise(5)=rand(1)*1e-3; % "         well 2
```

```
noise(8)=rand(1)*1e-3; % "          well 3
noise(10)=rand(1)*1e-3; % "          riser

noise(1)=rand(1)*1e-2; %
noise(3)=rand(1)*1e-2; %
noise(4)=rand(1)*1e-2; %
noise(6)=rand(1)*1e-2; %
noise(7)=rand(1)*1e-2; %
noise(9)=rand(1)*1e-2; %
noise(11)=rand(1)*1e-2; %
xz0_measured=xz0_measured+noise;
%noise              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

for  n=1:1:ode_time
    % Passing extra parameters
panda=network_128(0,(xz(n,:))',u_ode,p);
    % Removing [differentials; residuals]
panda(1:24)=[];
    % Appending outputs
outputs_append=[outputs_append,panda];
end
end
toc

outputs_append=outputs_append';


%save('plant_NMPC_high_noise.mat','uSS','xSS','outputs_append','xz_append','
      u_opt_append','t','p',' u0 ',' x0 ',' z0')
```

Listing C.7: Source code for entire code structure